aws

開發人員指南

Amazon CloudFront

Amazon CloudFront: 開發人員指南

Table of Contents

什麼是 Amazon CloudFront？

Amazon CloudFront 是一種網絡服務，可以加快向用戶分發靜態和動態 Web 內容（例如 .html，.css，.js 和圖像文件）的速度。 CloudFront 透過稱為節點位置的全球資料中心網路傳遞您的內容。當使用者要求您提供服務的內容時 CloudFront，會將要求路由至提供最低延遲 (時間延遲) 的節點，以便以最佳效能傳送內容。

- 如果內容已經位於延遲最低的節點位置，請立即 CloudFront 傳送。
- 如果內容不在該節點，請從您定義的來源 CloudFront 擷取內容，例如 Amazon S3 儲存貯體、MediaPackage 通道或 HTTP 伺服器 (例如 Web 伺服器)，您已將其識別為確定內容的來源。

舉個例子，假設您是從傳統的 Web 服務器提供圖像，而不是從 CloudFront. 例如，您可以使用 URL `https://example.com/sunsetphoto.png` 來提供影像 sunsetphoto.png。

您的使用者可輕鬆瀏覽至此 URL 並看到影像。但是他們可能不知道他們的請求是從一個網路路由到另一個網路，透過組成網際網路的複雜互連網路集合，直到找到映像為止。

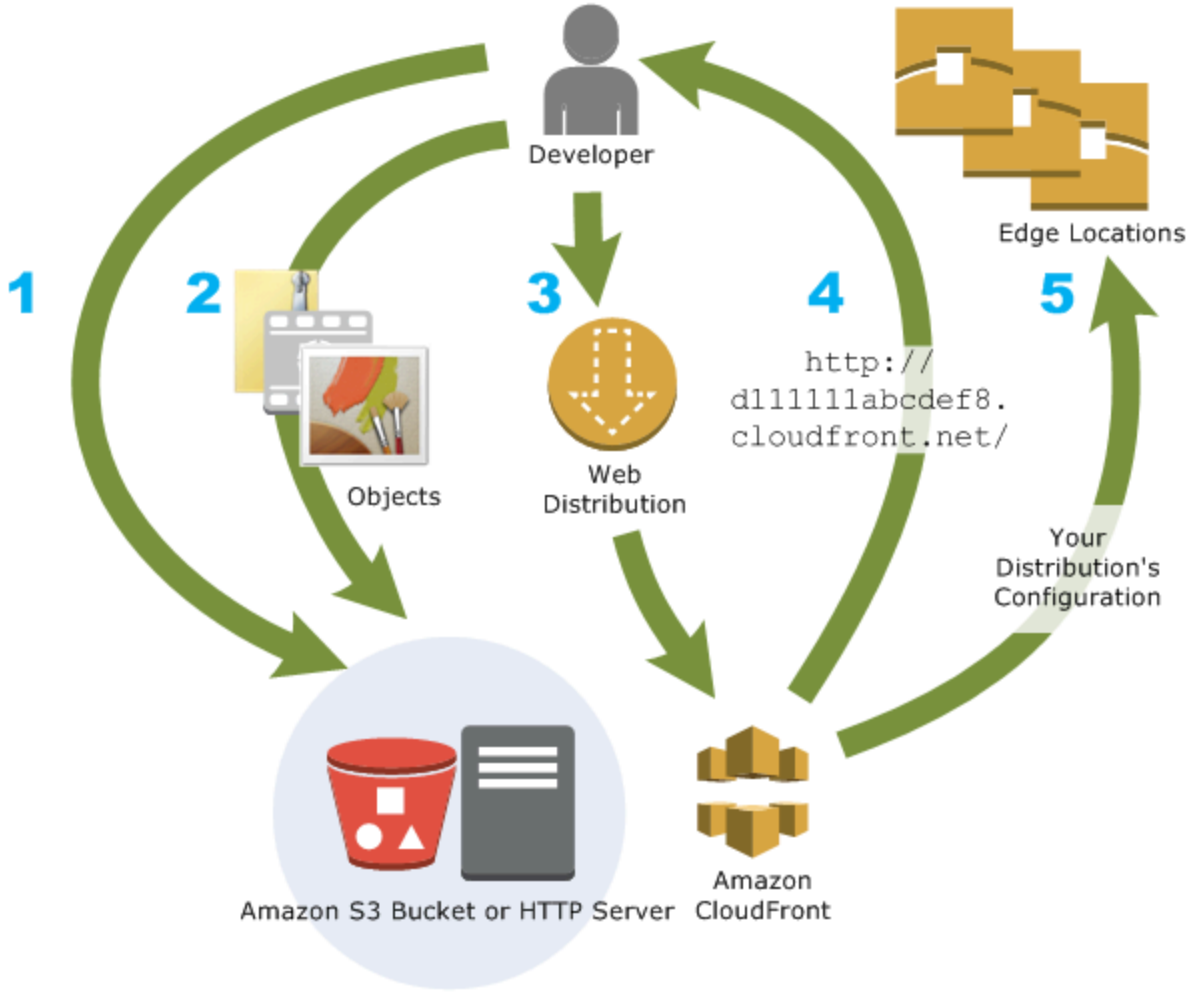
CloudFront 透過 AWS 骨幹網路將每個使用者要求路由到最能提供內容服務的節點，藉此加速內容的散佈。一般而言，這是 CloudFront 邊緣伺服器，可提供最快速的傳遞給檢視器。使用 AWS 網路可大幅減少使用者要求必須通過的網路數目，進而改善效能。使用者可以獲得更低的延遲 - 載入檔案的第一個位元所需的時間 - 以及更高的資料傳輸速率。

您也能獲得更高的可靠性和可用性，因為檔案 (也稱為物件) 的副本現在保留 (或快取) 在世界各地的多個節點。

主題

設定交付內容 CloudFront 的方式

您可以建立 CloudFront 發佈來告訴您 CloudFront 要將內容從何處傳送，以及如何追蹤和管理內容傳遞的詳細資料。然後 CloudFront 使用靠近檢視者的電腦 (邊緣伺服器)，以便在有人想要查看或使用內容時快速傳遞該內容。



如何設定 CloudFront 以傳遞內容

1. 您可以指定原始伺服器，例如 Amazon S3 儲存貯體或自己的 HTTP 伺服器，從中 CloudFront 取得檔案，然後從世界各地的 CloudFront 節點分發檔案。

   原始伺服器儲存物件的原始、最終版本。如果您是透過 HTTP 提供內容，則您的原始伺服器會是 Amazon S3 儲存貯體或 HTTP 伺服器，例如 Web 伺服器。您的 HTTP 伺服器可以在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或您管理的伺服器上執行，這些伺服器也稱為「自訂的原始伺服器」。

2. 您將檔案上傳到原始伺服器。您的檔案 (也稱為物件) 通常包含網頁、影像和媒體檔案，但也可以是能透過 HTTP 提供的任何內容。

   如果您使用 Amazon S3 儲存貯體做為原始伺服器，您可以將儲存貯體中的物件設為可公開讀取，以便知道物件 CloudFront URL 的任何人都可以存取它們。您還可以選擇保持物件的隱私以及控制誰能對其作存取。請參閱使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。

3. 您可以建立 CloudFront 散佈，告訴使用者透過網站或應用程式要求檔案時，要從 CloudFront 哪些原始伺服器取得檔案。同時，您可以指定詳細資料，例如是否 CloudFront 要記錄所有要求，以及是否要在建立分發後立即啟用。
4. CloudFront 會將網域名稱指派給您可以在 CloudFront 主控台中看到的新發行版，或在回應程式設計要求時傳回的網域名稱，例如 API 要求。如果想要，您可以改為新增要使用的替代網域名稱。
5. CloudFront 會將發行版的設定 (但不是您的內容) 傳送至其所有邊緣位置或存在點 (PoP) — 位於地理位置分散的資料中心中的伺服器集合，可 CloudFront 快取檔案複本。

當您開發網站或應用程式時，您會使用為您的 URL CloudFront 提供的網域名稱。例如，如果以分發的網域名稱 CloudFront 傳回`d111111abcdef8.cloudfront.net`，則 Amazon S3 儲存貯體 (或 HTTP 伺服器上的根目錄中) 中 logo.jpg 的 URL 為`https://d111111abcdef8.cloudfront.net/logo.jpg`。

或者，您可以設置 CloudFront 使用您自己的域名與您的分發。在這種情況下，URL 可能是 `https://www.example.com/logo.jpg`。

或者，您可以將原始伺服器設定為將標頭新增至檔案，以指出檔案在 CloudFront Edge 位置的快取中保留多長時間。在預設情況下，每個檔案在到期前的 24 小時內保持在節點。最短到期時間是 0 秒；沒有最大到期時間。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

定價

CloudFront 從其邊緣位置傳出資料的費用，以及 HTTP 或 HTTPS 要求。定價依使用類型、地理區域和功能選擇而有所不同。

使用 Amazon Simple Storage Service (Amazon S3)、Elastic Load Balancing 或 Amazon API Gateway 等來 AWS 源時，從原始端傳輸到的資料永遠 CloudFront 是免費的。使用來 AWS 源時，您只需為從檢視器傳出資料傳輸 CloudFront 到檢視器的費用。

如需詳細資訊，請參閱CloudFront 定價和計費與儲蓄組合常見問題集。

CloudFront 使用案例

使用 CloudFront 可以幫助您實現各種目標。本節僅摘列其中幾項，另提供詳細資訊的連結以讓您了解更多可能性。

主題

- 加速靜態網站內容交付
- 提供隨需視訊或即時串流視訊
- 在整個系統處理過程中加密特定欄位
- 在邊緣進行自訂
- 使用 Lambda@Edge 自訂項目提供私有內容

加速靜態網站內容交付

CloudFront 可以加快向全球觀眾傳送靜態內容 (例如影像 JavaScript、樣式表等) 的速度。通過使用 CloudFront，您可以利用 AWS 骨幹網絡和 CloudFront 邊緣服務器，在觀眾訪問您的網站時為他們提供快速，安全和可靠的體驗。

簡單存放及交付靜態內容的方法是使用 Amazon S3 儲存貯體。搭配使用 S3 CloudFront 具有許多優點，包括使用來源存取控制輕鬆限制 S3 內容存取的選項。

如需將 S3 與 CloudFront搭配使用的詳細資訊 (包括可協助您快速入門的 AWS CloudFormation 範本)，請參閱 Amazon S3 + Amazon CloudFront：在雲端進行比對。

提供隨需視訊或即時串流視訊

CloudFront 提供多種將媒體串流至全球檢視者的選項，包括預先錄製的檔案和即時活動。

- 對於點播視頻（VOD）流，您可以使用常見格式（例如 MPEG DASH，蘋果 HLS，Microsoft 流暢流媒體和 CMAF）流式傳輸 CloudFront 到任何設備。
- 對於廣播即時串流，您可在節點快取媒體片段，以便將按正確順序傳輸各片段的資訊清單檔案的多個請求相組合，從而減輕原始伺服器的負擔。

如需如何使用傳遞串流內容的詳細資訊 CloudFront，請參閱視頻點播和實時流視頻 CloudFront。

在整個系統處理過程中加密特定欄位

當您使用設定 HTTPS 時 CloudFront，您已經擁有與原始伺服器的安全 end-to-end 連線。若增設欄位層級加密，您便可在整個系統處理過程中保護特定資料並實施 HTTPS 安全性，從而只有原始伺服器端的某些應用程式才能看到該資料。

若要設定欄位層級加密，請將公開金鑰新增至 CloudFront，然後指定要使用金鑰加密的欄位集。如需詳細資訊，請參閱 使用欄位層級加密來協助保護敏感資料。

在邊緣進行自訂

透過在節點執行無伺服器程式碼，開啟了讓您為瀏覽者自訂內容與體驗的諸多可能性，同時還可減少延遲。例如，您可在原始伺服器停機進行維護時傳回自訂錯誤訊息，以免瀏覽者看到的是一般 HTTP 錯誤訊息。或者，您可以使用函數來協助授權使用者並控制對您內容的存取權，然後再將請求 CloudFront 轉寄至您的來源。

搭配使用 Lambda @Edge， CloudFront 可提供多種方式來自訂CloudFront 交付的內容。若要進一步了解 Lambda @Edge，以及如何使用建立和部署函數CloudFront，請參閱使用 Lambda @Edge 在邊緣進行自訂。如欲查看可供您針對自身解決方案進行自訂的若干程式碼範例，請參閱 Lambda@Edge 範例函數。

使用 Lambda@Edge 自訂項目提供私有內容

除了使用已簽署的 URL 或已簽署的 Cookie 之外，使用 Lambda @Edge 還可協助您設定 CloudFront 分發，以便從您自己的自訂來源提供私人內容。

若要使用提供私人內容 CloudFront，請執行下列動作：

- 要求使用者 (瀏覽者) 使用已簽章的 URL 或已簽章的 Cookie 來存取內容。
- 限制對原始伺服器的存取，以便只能從面向原點 CloudFront的伺服器使用。您可通過下列其中之一的方式執行此操作：
  - 對於 Amazon S3 原始伺服器，您可使用原始存取控制 (OAC)。
  - 對於自訂的原始伺服器，您可以執行下列操作：
    - 如果自訂來源受 Amazon VPC 安全群組保護 AWS Firewall Manager，或者您可以使用受 CloudFront 管前置詞清單，僅允許從面向原點的 IP 地址傳入流量到您 CloudFront的來源。
    - 使用自訂 HTTP 標頭來限制只存取來自的要求 CloudFront。如需更多詳細資訊，請參閱 the section called “ 在自訂原始伺服器上限制存取檔案” 及 the section called “將自訂標頭新增到原

始伺服器請求”。如需使用自訂標頭來限制 Application Load Balancer 原始伺服器的存取，請參閱 the section called “限制對 Application Load Balancers 的存取”。

- 如果自訂來源需要自訂存取控制邏輯，您可以使用 Lambda @Edge 來實作該邏輯，如本部落格文章所述：使用 Amazon CloudFront 和 Lambda @Edge 提供私有內容。

如何 CloudFront 提供內容

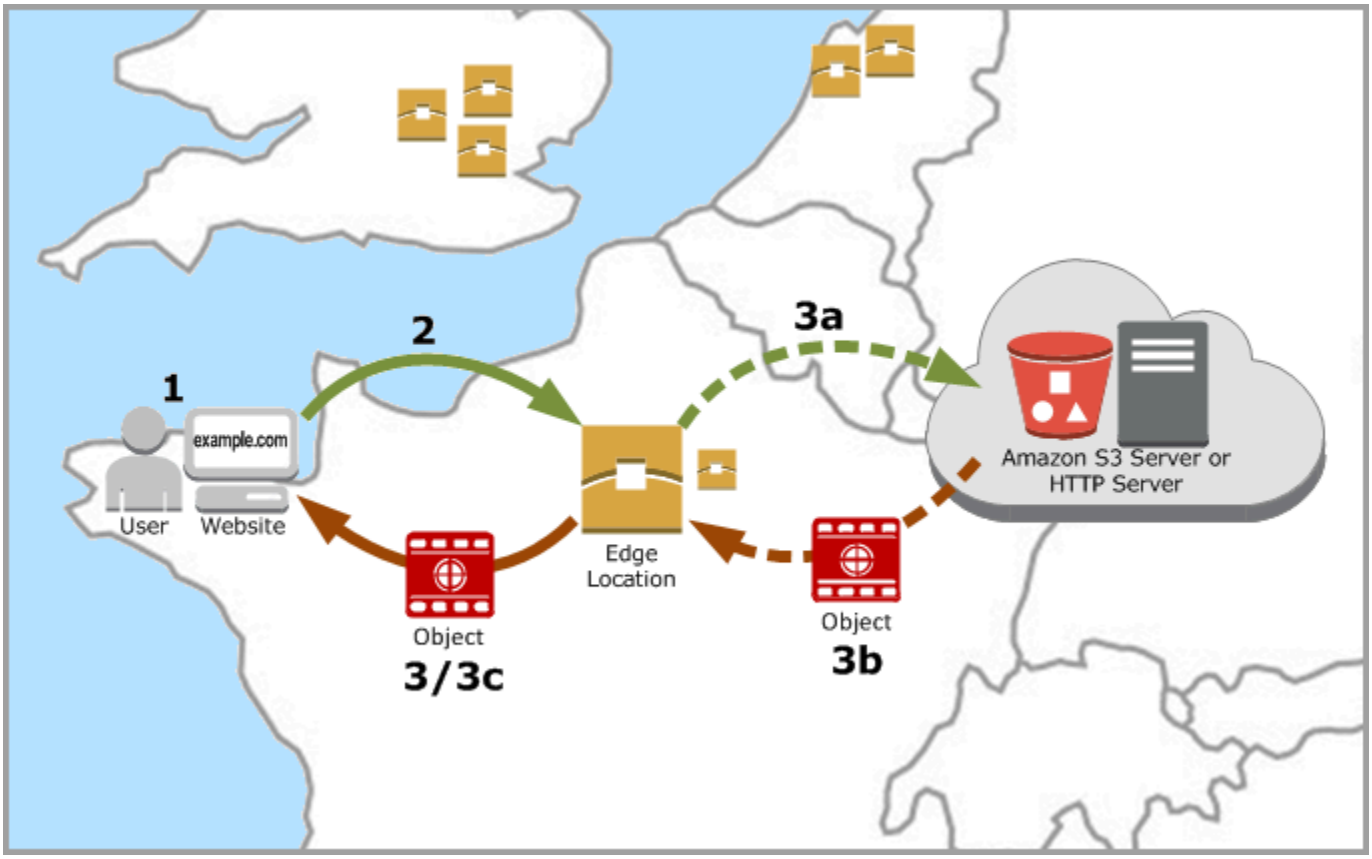
經過一些初始設置後， CloudFront 與您的網站或應用程序一起工作，並加快內容的交付速度。本節說明如何在觀眾 CloudFront 提出要求時提供您的內容。

主題

如何 CloudFront 向使用者提供內容

設定 CloudFront 為傳送內容後，使用者要求您的物件時會發生以下情況：

1. 使用者存取您的網站或應用程式並傳送一或多個物件的請求，如影像檔案或 HTML 檔案。
2. DNS 會將要求路由至 CloudFront POP (邊緣位置)，以最適合要求服務，通常是最接近的 CloudFront POP (就延遲而言)。
3. CloudFront 檢查其緩存請求的對象。如果物件位於快取中，則將其 CloudFront 傳回給使用者。如果物件不在快取中，請執 CloudFront行下列動作：
   a. CloudFront 將請求與分發中的規格進行比較，並將相應物件的請求轉送至原始伺服器，例如，至 Amazon S3 儲存貯體或 HTTP 伺服器。
   b. 原始伺服器會將物件傳送回節點。
   c. 一旦第一個字節從原點到達， CloudFront 開始將對象轉發給用戶。 CloudFront 還會將對象添加到緩存中，以便下次有人請求它時。



如何 CloudFront 與區域邊緣快取搭配使用

CloudFront 存在點 (也稱為 PoP 或邊緣位置) 確保可以快速向觀眾提供熱門內容。 CloudFront 還具有區域邊緣緩存，可以使更多內容更接近觀眾，即使內容不夠受歡迎而無法保持在 POP 中，也可以幫助提高該內容的性能。

區域節點快取有助於處理所有內容類型，特別是隨著時間推移漸漸不熱門的內容。範例包含使用者產生的內容，例如影片、照片或圖案；電子商務資產，例如產品照片和影片；以及可能因新聞和事件而知名度突然暴增的相關內容。

區域快取的工作方式

區域邊緣快取是全球部署的 CloudFront 位置，靠近檢視者。位於原始伺服器和 POP (直接向瀏覽者提供內容的全球節點) 之間。隨著物件變得較不熱門，個別 POP 可能會移除這些物件，為更熱門的內容騰出空間。區域節點快取擁有的快取比個別 POP 大，因此物件在最近的區域節點快取位置保留在快取中的時間會比較久。這有助於讓更多內容與觀眾更接近，從而減少返 CloudFront 回原始伺服器的需求，並提高觀眾的整體效能。

當瀏覽者在您的網站上或透過您的應用程式發出請求時，DNS 會將請求路由到最能滿足使用者請求的 POP。就延遲而言，此位置通常是最近的 CloudFront 節點。在 POP 中， CloudFront檢查其緩存請求

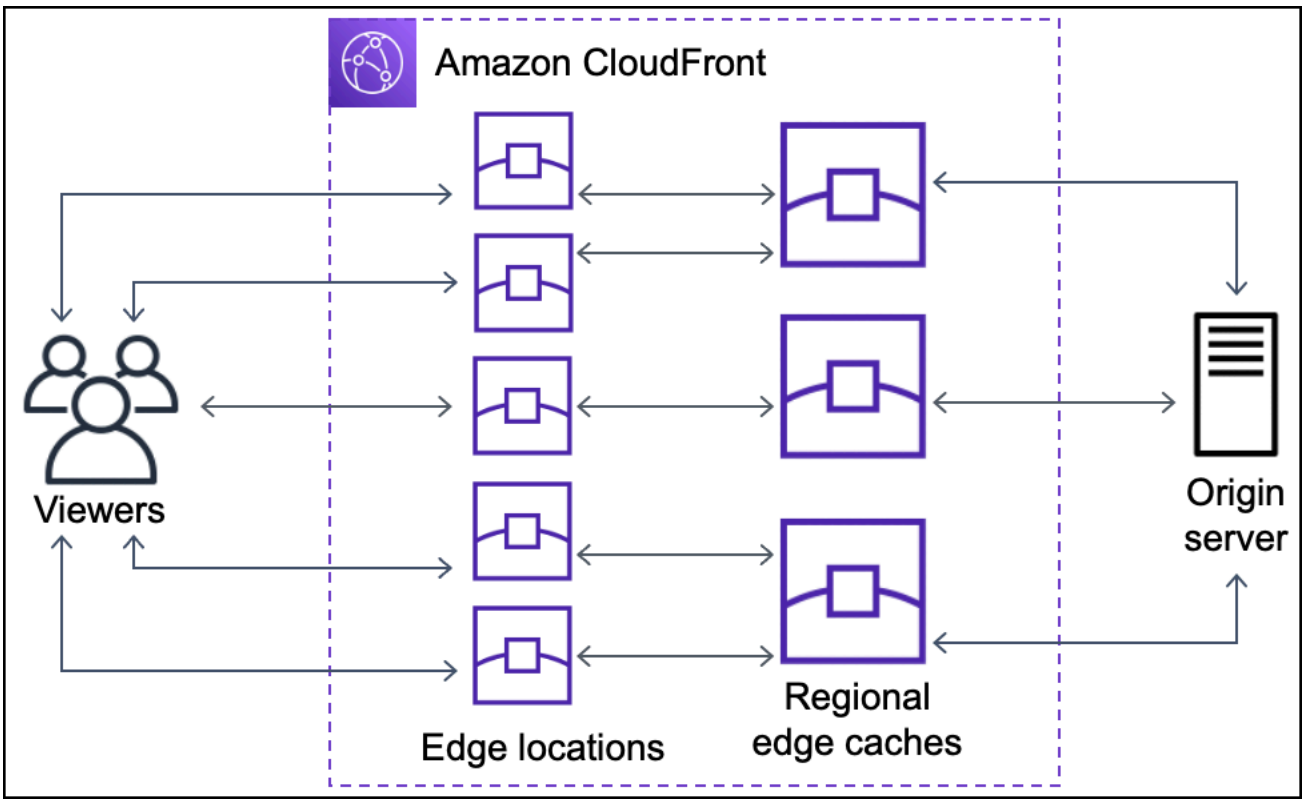
的對象。如果物件位於快取中，則將其 CloudFront傳回給使用者。如果物件不在快取中，POP 通常將移至最近的區域節點快取，以擷取物件。如需 POP 何時略過區域節點快取，並直接進入原始伺服器的相關資訊，請參閱以下備註。

在區域邊緣快取位置中， CloudFront 再次檢查其快取是否有要求的物件。如果物件位於快取中，請將其 CloudFront 轉寄至要求它的 POP。一旦第一個位元組從區域邊緣快取位置到達，就會 CloudFront 開始將物件轉寄給使用者。 CloudFront 也會將物件新增至 POP 中的快取，以供下次有人要求時使用。

對於未在 POP 或區域邊緣快取位置快取的物件，請將要求與發行版中的規格進行 CloudFront 比較，並將要求轉送至原始伺服器。原始伺服器將物件傳送回區域邊緣快取位置之後，就會將它轉寄至 POP，然後將其 CloudFront 轉寄給使用者。在這種情況下，除了 POP 之外， CloudFront 還會將物件新增至區域邊緣快取位置的快取中，以供檢視者下次要求時使用。這樣可以確保區域中的所有 PoP 共享本地緩存，從而消除了對原始服務器的多個請求。 CloudFront 也會與原始伺服器保持持續連線，以便儘快從來源擷取物件。

> **Note**
>
> - 區域節點快取具有與 POP 相同的功能。例如，快取失效請求在到期之前從 POP 快取和區域節點快取中移除物件。下次檢視者要求物件時，會 CloudFront返回原點以擷取物件的最新版本。
> - Proxy HTTP 方法 (PUT、POST、PATCH、OPTIONS 和 DELETE) 直接從 POP 進入原始伺服器，而且不透過區域節點快取進行代理。
> - 在請求時決定的動態請求不會流經區域邊緣快取，而是直接移至原始伺服器。
> - 當來源是 Amazon S3 儲存貯體，且請求的最佳區域節點快取與 S3 儲存貯體位於 AWS 區域相同時，POP 會略過區域節點快取並直接進入 S3 儲存貯體。

下圖說明要求和回應如何透過 CloudFront 節點和區域節點快取流動。



CloudFront 邊緣伺服器的位置和 IP 位址範圍

如需 CloudFront 邊緣伺服器位置的清單，請參閱 Amazon CloudFront 全球邊緣網路頁面。

Amazon Web Services (AWS) 會以 JSON 格式發佈目前的 IP 地址範圍。若要檢視目前範圍，請下載 ip-ranges.json。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的 AWS IP 地址範圍。

若要尋找與 CloudFront 邊緣伺服器相關聯的 IP 位址範圍，請在 ip-ranges.json 中搜尋下列字串：

```
"region": "GLOBAL",
"service": "CLOUDFRONT"
```

或者，您可以僅檢視位於的 CloudFront IP 範圍https://d7uri8nf7uskq.cloudfront.net/tools/list-cloudfront-ips。

使用 CloudFront 託管前綴列表

CloudFront Managed 前置詞清單包含所有全球分散式原點對向伺服器 CloudFront的 IP 位址範圍。如果您的來源託管在 Amazon VPC 安全群組上 AWS 並受到保護，您可以使用受 CloudFront 管前置詞清單，僅允許從 CloudFront面向原點的伺服器傳入流量到達來源，以防止任何非CloudFront 流量到達您的來源。 CloudFront 維護託管前綴列表，因此它始終與所有面向來源的全球起源服務器的 IP 地址保持

最新狀態。 CloudFront透過 CloudFront Managed 前置詞清單，您不需要自行讀取或維護 IP 位址範圍清單。

例如，假設您的原始伺服器是位於歐洲 (倫敦) 區域 (eu-west-2) 的 Amazon EC2 執行個體。如果執行個體位於 VPC 中，您可以建立安全群組規則，以允許 CloudFront 受管理前置詞清單中的輸入 HTTPS 存取。這可讓所有 CloudFront的全域原始伺服器連線到執行個體。如果您從安全性群組移除所有其他輸入規則，就可以防止任何非CloudFront流量到達執行個體。

CloudFront 託管前綴列表被命名為混合。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的使用 AWS受管前置詞清單。

> **Important**
>
> 受 CloudFront 管前置詞清單在套用至 Amazon VPC 配額的方式中是唯一的。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的「AWS受管字首清單權重」。

存取 CloudFront

您可以通過以下方式訪問 Amazon CloudFront ：

- AWS Management Console— 本指南中的程序說明如何使用 AWS Management Console 來執行作業。
- AWS SDK — 如果您使用的是 AWS 提供 SDK 的程式設計語言，您可以使用 SDK 來存取 CloudFront。SDK 可簡化驗證、輕鬆與您的開發環境整合，並提供 CloudFront 指令的存取權。如需詳細資訊，請參閱 Amazon Web Services 適用工具。
- CloudFront API — 如果您使用的是 SDK 無法使用的程式設計語言，請參閱 Amazon CloudFront API 參考，以取得有關 API 動作以及如何發出 API 請求的資訊。
- AWS Command Line Interface – 如需詳細資訊，請參閱 AWS Command Line Interface 使用者指南中的使用 AWS Command Line Interface完成設定。
- AWS Tools for Windows PowerShell – 如需詳細資訊，請參閱《AWS Tools for Windows PowerShell 使用者指南》中的設定 AWS Tools for Windows PowerShell。

搭 CloudFront 配 AWS SDK 使用

AWS 軟件開發套件（SDK）可用於許多流行的編程語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
AWS SDK for C++	AWS SDK for C++ 程式碼範例
AWS SDK for Go	AWS SDK for Go 程式碼範例
AWS SDK for Java	AWS SDK for Java 程式碼範例
AWS SDK for JavaScript	AWS SDK for JavaScript 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
AWS SDK for .NET	AWS SDK for .NET 程式碼範例
AWS SDK for PHP	AWS SDK for PHP 程式碼範例
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 程式碼範例
AWS SDK for Ruby	AWS SDK for Ruby 程式碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

可用性範例

找不到所需的內容嗎？ 請使用本頁面底部的提供意見回饋連結申請程式碼範例。

CloudFront 技術資源

使用下列資源取得有關技術問題的解答 CloudFront：

- AWS Re: post — 一個基於社區的問答網站，供開發人員討論相關的技術問題。 CloudFront
- AWS Support 中心 — 此網站包含您最近的支援案例、結果 AWS Trusted Advisor 和健康狀態檢查的相關資訊。它也提供討論區、技術常見問題集、服務健康狀態儀表板以及 AWS Support 計劃相關資訊的連結。

- AWS 高級 Support — 瞭解 AWS 高級 Support，這是一個 one-on-one可協助您在 AWS上建置和執行應用程式的快速回應支援管道。
- AWS IQ — 獲得 AWS 認證專業人員和專家的幫助。

開始使用 Amazon CloudFront

CloudFront 通過創建基本分發或安全的靜態網站開始提供內容的基本步驟。

主題

設定

本主題說明初步步驟，例如建立一個 AWS 帳戶，讓您準備好使用 Amazon CloudFront。

主題

註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

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

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

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

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

建立具有管理權限的使用者

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

保護您的 AWS 帳戶根使用者

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

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。
2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

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

建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱 AWS IAM Identity Center 使用者指南中的啟用 AWS IAM Identity Center。
2. 在 IAM 身分中心中，將管理存取權授予使用者。

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

以具有管理權限的使用者身分登入

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

  如需使用 IAM 身分中心使用者登入的說明，請參閱使用AWS 登入 者指南中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

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

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

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

設定 AWS Command Line Interface 或 AWS Tools for Windows PowerShell

AWS Command Line Interface (AWS CLI) 是用於管理 AWS 服務的統一工具。如需如何安裝和設定 AWS CLI的資訊，請參閱《AWS Command Line Interface 使用者指南》中的使用 AWS Command Line Interface開始設定。

如果您有使用 Windows 的經驗 PowerShell，您可能更喜歡使用 AWS Tools for Windows PowerShell。如需詳細資訊，請參閱 AWS Tools for Windows PowerShell 使用者指南中的設定 AWS Tools for Windows PowerShell。

下載 AWS 開發套件

如果您使用的程式設計語言 AWS 提供開發套件，建議您使用開發套件而非 Amazon CloudFront API。SDK 讓驗證變得更簡單、輕鬆與您的開發環境整合，並提供對 CloudFront 命令的輕鬆存取。如需詳細資訊，請參閱在 AWS上建置的工具。

開始使用基本 CloudFront發行版

本節中的程序說明如何使用 CloudFront 來設定執行下列作業的基本組態：

- 建立要用作發佈來源的值區。
- 將物件的原始版本存放在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中。
- 使用來源存取控制 (OAC) 將經過驗證的請求傳送到您的 Amazon S3 來源。OAC 會透過傳送請求， CloudFront 以防止檢視者直接存取您的 S3 儲存貯體。如需 OAC 的詳細資訊，請參閱限制對 Amazon 簡單儲存服務來源的存取。
- 在 URL 中為您的物件使用 CloudFront 網域名稱 (例如，`https://d111111abcdef8.cloudfront.net/index.html`)。
- 將物件保留在 CloudFront 邊緣位置，預設持續時間為 24 小時 (最短持續時間為 0 秒)。

其中的大多數選項均可供自訂。如需如何自訂CloudFront 發佈選項的相關資訊，請參閱建立、更新和刪除分發。

主題

必要條件

開始之前，請確定您已完成 設定 所述的步驟。

步驟 1：建立 Amazon S3 儲存貯體

Amazon S3 儲存貯體是檔案 (物件) 或資料夾的容器。 CloudFront 當 S3 儲存貯體是來源時，幾乎可以為您分發任何類型的檔案。例如， CloudFront可以分發文本，圖像和視頻。您可以在 Amazon S3 上存放的資料量沒有上限。

在本教學課程中，您會使用提供的範例`hello world`檔案建立 S3 儲存貯體，以建立基本網頁。

建立儲存貯體

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 我們建議您使用我們的 Hello World 範例為此入門。下載你好世界網頁：hello-world-html.zip。將其解壓縮並將文件`css`夾和`index`文件保存在方便的位置，例如運行瀏覽器的桌面。
3. 選擇建立儲存貯體。
4. 在 Amazon 簡單儲存服務使用者指南中，輸入符合一般用途儲存貯體命名規則的唯一值區名稱。
5. 對於「地區」，我們建議您選 AWS 區域 擇地理位置靠近您的地區。（這可以減少延遲和成本。）
   - 選擇不同的區域也有效。例如，您可以這樣做來滿足法規要求。
6. 將所有其他設定保留為預設值，然後選擇 Create bucket (建立儲存貯體)。

步驟 2：將內容上傳至儲存貯體

建立 Amazon S3 儲存貯體之後，將解壓縮`hello world`檔案的內容上傳到該儲存貯體。（您在中下載並解壓縮此文件步驟 1：建立 Amazon S3 儲存貯體。）

將內容上傳至 Amazon S3

1. 在「一般用途值區」區段中，選擇新值區的名稱。
2. 選擇上傳。
3. 在 [上傳] 頁面上，將資`css`料夾和`index`檔案拖曳至放置區域。
4. 將所有其他設定保留為預設值，然後選擇上傳。

步驟 3：建立使用具有 OAC 的 Amazon S3 原始伺服器的 CloudFront分發

在本教學課程中，您將建立使用具有來源存取控制 (OAC) 的 Amazon S3 來源的 CloudFront 分發。OAC 可協助您將經過驗證的請求安全地傳送到 Amazon S3 來源。如需 OAC 的詳細資訊，請參閱限制對 Amazon 簡單儲存服務來源的存取。

使用使用 OAC 的 Amazon S3 來源建立 CloudFront分發

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇 Create Distribution (建立分佈)。
3. 對於 Origin 、原始網域，請選擇您為本教學課程建立的 S3 儲存貯體。
4. 對於 Origin、Ori gin 存取權限，請選取 Origin 存取控制設定 (建議使用)。
5. 針對 Origin 存取控制，請選擇 [建立新的 OAC]。
6. 在 [建立新的 OAC] 窗格中，保留預設設定並選擇 [建立]。
7. 對於 Web 應用程式防火牆 (WAF)，請選取其中一個選項。
8. 對於所有其他剖面和設定，請接受預設值。如需關於這些選項的詳細資訊，請參閱 分佈設定。
9. 選擇 Create Distribution (建立分佈)。
10. 在 S3 儲存貯體政策需要更新橫幅中，閱讀訊息並選擇 [複製政策]。
11. 在同一標題中，選擇前往 S3 儲存貯體許可以更新政策的連結。這會帶您前往 Amazon S3 主控台中的儲存貯體詳細資訊頁面。）
12. 對於 Bucket policy (儲存貯體政策)，選擇 Edit (編輯)。
13. 在 「編輯陳述式」欄位中，貼上您在步驟 10 中複製的原則。

14. 選擇儲存變更。
15. 返回 CloudFront 控制台並查看新發行版的「詳細信息」部分。發佈部署完成後，[上次修改] 欄位會從 [部署] 變更為日期和時間。
16. 記錄指 CloudFront 派給您分發的網域名稱。它看起來類似下列：`d111111abcdef8.cloudfront.net`。

在生產環境中使用本教學中的散發和 S3 儲存貯體之前，請務必對其進行設定以符合您的特定需求。如需在生產環境中設定存取權的相關資訊，請參閱設定安全存取和限制對內容的存取。

步驟 4：通過訪問您的內容 CloudFront

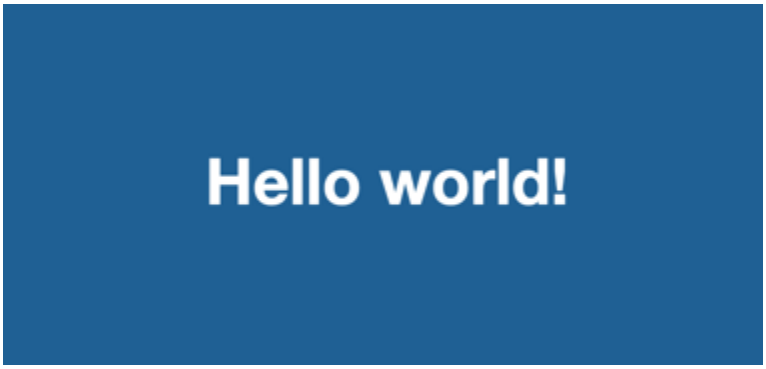
若要透過存取您的內容 CloudFront，請將您 CloudFront發佈的網域名稱與內容的主要頁面結合起來。(您已將您的分發網域名稱記錄在步驟 3：建立使用具有 OAC 的 Amazon S3 原始伺服器的 CloudFront 分發.)

- 您的分佈網域名稱可能會像這樣：`d111111abcdef8.cloudfront.net`。
- 網站主頁的路徑通常是 `/index.html`。

因此，通過訪問您的內容的 URL CloudFront 可能如下所示：

`https://d111111abcdef8.cloudfront.net/index.html`.

如果您按照前面的步驟操作並使用 hello world 網頁，則應該會看到以下內容：



當您將更多內容上傳到此 S3 儲存貯體時，您可以透CloudFront 過結合 CloudFront 分發網域名稱與 S3 儲存貯體中物件的路徑來存取內容。例如，如果您上傳名為 `new-page.html` 的新檔案到 S3 儲存貯體的根目錄，則 URL 會像這樣：

`https://d111111abcdef8.cloudfront.net/new-page.html`.

步驟 5：清除

如果您僅將分發和 S3 儲存貯體建立為學習練習，請將其刪除，以免再產生費用。首先刪除該分佈。如需詳細資訊，請參閱下列連結：

- 刪除 分發
- 刪除值區

提示

此入門教學課程提供建立發行版的最小架構。我們建議您進一步探索下列增強功能：

- 根據預設，Amazon S3 儲存貯體中的檔案 (物件) 會設定為私有。只有建立值區的使用者才有讀取或寫入檔案的權限。 AWS 帳戶 如果您想要允許任何人使用 CloudFront URL 存取 Amazon S3 儲存貯體中的檔案，則必須授與物件的公開讀取權限。
- 您可以使用 CloudFront 私有內容功能來限制對 Amazon S3 儲存貯體中內容的存取。如需有關分佈私有內容的詳細資訊，請參閱使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。
- 您可以將 CloudFront 分發配置為使用自定義域名（例如，`www.example.com`而不是`d111111abcdef8.cloudfront.net`）。如需詳細資訊，請參閱 使用自訂網址。
- 本教學使用具有來源存取控制 (OAC) 的 Amazon S3 來源。但是，如果您的來源是設定為網站端點的 S3 儲存貯體，則無法使用 OAC。在這種情況下，您必須將存儲桶設置 CloudFront 為自定義來源。如需詳細資訊，請參閱 使用設定為網站端點的 Amazon S3 儲存貯體。如需 OAC 的詳細資訊，請參閱限制對 Amazon 簡單儲存服務來源的存取。

開始使用安全的靜態網站

您可以使用本主題中描述的解決方案為您的網域名稱建立安全的靜態網站，開始使用 Amazon CloudFront 。靜態網站只會使用靜態檔案 (例如 HTML、CSS、 JavaScript影像和視訊)，不需要伺服器或伺服器端處理。有了這個解決方案，您的網站可以獲得以下好處：

- 使用 Amazon 簡易儲存服務 (Amazon S3)的耐久儲存 – 此解決方案會建立 Amazon S3 儲存貯體來託管靜態網站的內容。要更新您的網站，只需將您的新檔案上傳到 S3 儲存貯體即可。
- Amazon CloudFront 內容交付網路加速 — 此解決方案可建立 CloudFront 分發，以低延遲將您的網站提供給觀眾。該分發配置了來源訪問控制（OAC），以確保只能通過訪問該網站 CloudFront，而不是直接從 S3 訪問。

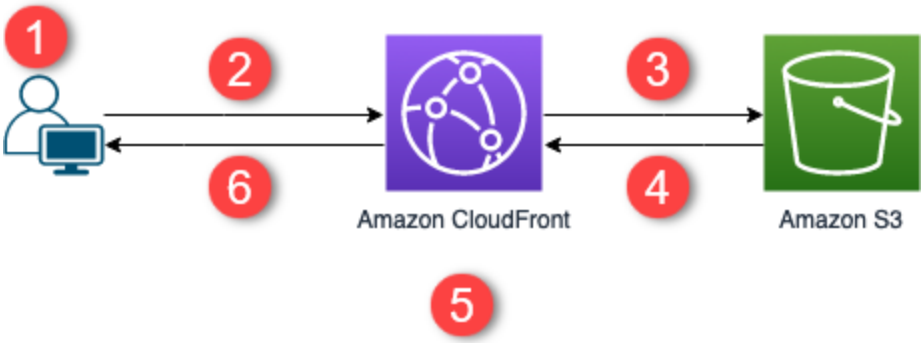
- 由 HTTPS 和安全標頭保護 — 此解決方案會在 AWS Certificate Manager (ACM) 中建立 SSL/TLS 憑證，並將其附加至發行版本。 CloudFront 此憑證可讓您透過 HTTPS 安全地為您網域的網站提供分佈。
- 配置和部署與 AWS CloudFormation— 此解決方案使用 AWS CloudFormation 模板來設置所有組件，因此您可以更專注於網站的內容，而不是在配置組件上。

這個解決方案是開源的 GitHub. 若要檢視程式碼、提交提取請求或開立問題單，請前往 https://github.com/aws-samples/amazon-cloudfront-secure-static-site。

主題

解決方案概觀

下圖顯示此靜態網站解決方案運作方式的概觀：



1. 瀏覽者在 www.example.com 請求網站。
2. 如果要求的物件已快取，則會將物件從其快取 CloudFront 傳回給檢視器。
3. 如果物件不在 CloudFront 快取中， CloudFront 請從來源 (S3 儲存貯體) 要求物件。
4. S3 將對象返回到 CloudFront。
5. CloudFront 快取物件。
6. 物件會傳回給檢視器。來到相同 CloudFront 邊緣位置之物件的後續要求會從 CloudFront 快取中提供服務。

部署解決方案

若要部署此安全靜態網站解決方案，您可以選擇下列其中一個選項：

- 使用主 AWS CloudFormation 控台部署含有預設內容的解決方案，然後將您的網站內容上傳到 Amazon S3。
- 將解決方案複製到您的電腦，以新增您的網站內容。然後，使用 AWS Command Line Interface (AWS CLI) 部署解決方案。

> **Note**
>
> 您必須使用美國東部 (維吉尼亞北部) 區域來部署 CloudFormation 範本。

主題

必要條件

若要使用此解決方案，您必須具備下列先決條件：

- 指向 Amazon Route 53 託管區域的已註冊網域名稱 (例如 example.com)。託管區域必須與您部署此解決方案的位 AWS 帳戶 置相同。如果您沒有已註冊的網域名稱，可以使用 Route 53 註冊一個網域名稱。如果您有已註冊的網域名稱，但未指向 Route 53 託管區域，請將 Route 53 配置為您的 DNS 服務。
- AWS Identity and Access Management (IAM) 許可以啟動可建立 IAM 角色的 CloudFormation 範本，以及建立解決方案中所有 AWS 資源的許可。

您必須自行負責使用此解決方案時所產生的費用。如需有關成本的詳細資訊，請參閱每個頁面的定價頁面 AWS 服務。

使用 AWS CloudFormation 主控台

使用 CloudFormation 主控台進行部署

1. 選擇 Launch on AWS (在 AWS上啟動) (啟動)，以在 AWS CloudFormation 主控台中開啟此解決方案。如有必要，請登入您的 AWS 帳戶.

2. [建立堆疊] 精靈會在 CloudFormation主控台中開啟，其中包含指定此解決方案 CloudFormation範本的預先填入欄位。

   請選擇頁面最下方的 Next (下一頁)。

3. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，輸入下列欄位的數值：

   - SubDomain— 輸入要用於您的網站的子域名。例如，如果子網域名稱為 www，表示您的網站可以在 *www*.example.com 使用。(請將 example.com 取代為您的網域名稱，如下列項目符號所述)。
   - DomainName— 輸入您的域名, #如. COM. 此網域必須指向 Route 53 託管區域。
   - HostedZoneId— 您的域名的路線 53 託管區域 ID。
   - CreateApex— (選用) 在您的組態中建立網域頂點 (example.com) 的 CloudFront別名。

4. 完成時，請選擇 Next (下一步)。
5. (選用) 在 配置堆疊選項頁面上，新增標籤和其他堆疊選項。
6. 完成時，請選擇 Next (下一步)。
7. 在 Review (檢閱) 頁面上，捲動至頁面底部，然後選取 Capabilities (功能) 區段中的兩個方塊。這些功能 CloudFormation 允許建立可存取堆疊資源的 IAM 角色，以及動態命名資源。
8. 選擇 Create Stack (建立堆疊)。
9. 等待堆疊完成建立。堆疊會建立一些巢狀堆疊，而且可能需要幾分鐘才能完成。完成時，Status (狀態) 會變更為 CREATE_COMPLETE。

   當狀態為 CREATE_COMPLETE 時，請前往 https://*www.example.com* 檢視您的網站 (將 www.example.com 取代為您在步驟 3 中指定的子網域和網域名稱)。您應該會看到網站的預設內容：

將網站的預設內容取代為您自己的內容

1. 在以下網址開啟 Amazon S3 主控台：https://console.aws.amazon.com/s3/。
2. 選擇名稱以 amazon-cloudfront-secure-static-站點-s3bucketroot- 開頭的存儲桶。

> **Note**
>
> 確定選擇名稱中包含 s3bucketroot 的儲存貯體，而不是 s3bucketlog。名稱中含有 s3bucketroot 儲存貯體包含網站內容。帶有 s3bucketlog 儲存貯體只包含日誌檔案。

3. 刪除網站的預設內容，然後上傳您自己的內容。

> **Note**
>
> 如果您使用此解決方案的預設內容檢視您的網站，則可能是某些預設內容會快取在 CloudFront 邊緣位置。為了確保檢視者能看到您更新的網站內容，請使檔案無效，以便從 CloudFront 節點移除快取的副本。如需詳細資訊，請參閱 使檔案失效。

在本機複製解決方案

先決條件

要在部署此解決方案之前新增您的網站內容，您必須在本機封裝解決方案的成品，這需要 Node.js 和 npm。如需更多詳細資訊，請參閱 https://www.npmjs.com/get-npm。

新增您的網站內容並部署解決方案

1. 從 複製或下載解決方案https://github.com/aws-samples/amazon-cloudfront-secure-static-site 複製或下載之後，請開啟命令提示字元或終端機，然後瀏覽至 `amazon-cloudfront-secure-static-site` 資料夾。
2. 執行下列命令以安裝並封裝解決方案的成品：

```
make package-static
```

3. 將網站的內容複製到 www 資料夾中，覆寫預設的網站內容。
4. 執行下列 AWS CLI 命令以建立 Amazon S3 儲存貯體來存放解決方案的成品。*example-bucket-for-artifacts*以您自己的儲存貯體名稱取代。

```
aws s3 mb s3://example-bucket-for-artifacts --region us-east-1
```

5. 執行下列 AWS CLI 命令，將解決方案的成品封裝為 CloudFormation 範本。取代*example-bucket-for-artifacts*為您在上一個步驟中建立的值區名稱。

```
aws cloudformation package \
    --region us-east-1 \
    --template-file templates/main.yaml \
    --s3-bucket example-bucket-for-artifacts \
    --output-template-file packaged.template
```

6. 執行下列命令以部署解決方案 CloudFormation，取代下列值：

   - *##CloudFormation####-* 替換為堆棧的名稱。 CloudFormation
   - *example.com* – 取代為您的網域名稱。此網域必須指向相同的 Route 53 託管區域 AWS 帳戶。
   - *www* – 取代為您的網站所使用的子網域名稱。例如，如果子網域名稱為 www，表示您的網站可以在 www.example.com 使用。
   - *#### ID* — 取代為您網域名稱的 Route 53 託管區域 ID。

```
aws cloudformation deploy \
    --region us-east-1 \
    --stack-name your-CloudFormation-stack-name \
    --template-file packaged.template \
    --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
    --parameter-overrides DomainName=example.com SubDomain=www HostedZoneId=hosted-zone-ID
```

   - (選擇性) 若要使用網域頂點部署堆疊，請改為執行下列命令。

```
aws --region us-east-1 cloudformation deploy \
    --stack-name your-CloudFormation-stack-name \
```

```
    --template-file packaged.template \
    --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
    --parameter-overrides  DomainName=example.com SubDomain=www
HostedZoneId=hosted-zone-ID CreateApex=yes
```

7. 等待 CloudFormation 堆疊完成建立。堆疊會建立一些巢狀堆疊，而且可能需要幾分鐘才能完成。完成時，Status (狀態) 會變更為 CREATE_COMPLETE。

   當狀態變更為 CREATE_COMPLETE 時，請前往 https://www.example.com 檢視您的網站 (將 www.example.com 取代為您在上一個步驟中指定的子網域和網域名稱)。您應該會看到您網站的內容。

尋找存取日誌

此解決方案會啟用 CloudFront分發的存取記錄。請完成以下步驟來尋找分佈的存取日誌。

尋找分佈的存取日誌

1. 在以下網址開啟 Amazon S3 主控台：https://console.aws.amazon.com/s3/。
2. 選擇名稱以 amazon-cloudfront-secure-static-站點-s3bucketlog- 開頭的存儲桶。

   > **Note**
   >
   > 確定選擇名稱中包含 s3bucketlogs 的儲存貯體，而不是 s3bucketroot。名稱中含有 s3bucketlog 的儲存貯體包含日誌檔案。帶有 s3bucketroot 的儲存貯體包含網站內容。

3. 名為 cdn 的文件夾包含 CloudFront 訪問日誌。

使用發行版

您可以建立 Amazon CloudFront 分發，告訴您要 CloudFront 從哪裡傳送內容，以及如何追蹤和管理內容交付的詳細資訊。

從下列組態設定中選擇：

- 您的內容來源 — Amazon S3 儲存貯體、 AWS Elemental MediaPackage 通道、 AWS Elemental MediaStore 容器、Elastic Load Balancing 負載平衡器或 HTTP 伺服器，從中CloudFront 取得要分發的檔案。您最多可以針對單一分佈指定 25 個原始伺服器的任意組合。
- 存取權限 - 您希望每個人皆可存取檔案，或者僅限部分使用者存取。
- 安全性 - 您是否要啟用 AWS WAF 保護，並請求使用者使用 HTTPS 存取您的內容。
- 快取金鑰 - 您想要包括在快取金鑰的值 (如果有的話)。快取金鑰可唯一識別指定分佈快取中的每個檔案。
- 原始請求設定 — 您是否 CloudFront 要在傳送至原始伺服器的要求中包含 HTTP 標頭、Cookie 或查詢字串。
- 地理區域限制 — 是否 CloudFront 要阻止特定國家/地區的使用者存取您的內容。
- 記錄 — 無論您是 CloudFront 要建立標準記錄還是顯示檢視者活動的即時記錄。

有關您可以為每個 AWS 帳戶創建的當前最大分配數量，請參閱分佈的一般配額。您可為每個分佈提供的檔案數量沒有上限。

您可以使用分佈透過 HTTP 或 HTTPS 提供以下內容：

- 使用 HTTP 或 HTTPS 的靜態和動態下載內容 JavaScript，例如 HTML、CSS 和影像檔案。
- 各類格式的隨需視訊，如 Apple HTTP Live Streaming (HLS) 和 Microsoft Smooth Streaming。如需詳細資訊，請參閱 提供隨選視訊 (VOD) 搭配 CloudFront。
- 即時事件，例如即時會議、即時討論會，或即時音樂會。對於即時串流，您可以使用 AWS CloudFormation 堆疊自動建立分發。如需詳細資訊，請參閱 使用 CloudFront 和AWS媒體服務提供即時串流視訊。

下列主題提供有關 CloudFront 發行版的更多詳細資訊，以及如何設定它們以符合您的業務需求。如需有關建立分佈的詳細資訊，請參閱 建立、更新和刪除分發。

主題

建立、更新和刪除分發

以下任務清單總結了建立 Amazon CloudFront 分發的程序。若要瞭解如何更新或刪除發行版本，請參閱稍後的主題。

建立 分佈

1. 建立一或多個 Amazon S3 儲存貯體或將 HTTP 伺服器設定為原始伺服器。原始伺服器是您存放內容原始版本的位置。當收 CloudFront 到文件的請求時，它會轉到原點以獲取它在邊緣位置分發的文件。您可以使用 Amazon S3 儲存貯體和 HTTP 伺服器的任意組合做為您的原始伺服器。

   如果您使用的是 Amazon S3，請注意您的儲存貯體名稱必須全部小寫和不能包含空格。

   如果您使用的是 Amazon EC2 伺服器或其他自訂原始伺服器，請檢閱 使用 Amazon EC2（或其他自定義源地）。

   如需您可為分佈建立之原始伺服器數量的目前上限，或是有關請求更高配額的詳細資訊，請參閱分佈的一般配額。
2. 將內容上傳到原始伺服器。您可以將物件設為可公開讀取，或者您可以使用 CloudFront 已簽署的 URL 來限制對您內容的存取。

   > **Important**
   >
   > 您負責確保原始伺服器的安全。您必須確定 CloudFront 具有存取伺服器的權限，而且安全性設定可保護您的內容。

3. 建立您的 CloudFront 發行版：

   - 如需使用 CloudFront 主控台建立發行版的詳細資訊，請參閱建立分發。
   - 如需使用 CloudFront API 建立分發的相關資訊，請參閱 Amazon CloudFront API 參考CreateDistribution中的。

4. (選擇性) 如果您使用主 CloudFront 控台建立發行版，請為發行版建立更多快取行為或來源。如需行為和來源的詳細資訊，請參閱 若要更新發 CloudFront 佈。
5. 測試您的分佈。如需測試的詳細資訊，請參閱 測試發行版。
6. 開發您的網站或應用程式，以便使用在步驟 3 中建立發佈後CloudFront 傳回的網域名稱來存取您的內容。例如，如果CloudFront 傳回 d1111abcdef8.cloudfront.net 做為分發的網域名稱，則 Amazon S3 儲存貯體或 HTTP 伺服器上根目錄`image.jpg`中的檔案 URL 為。`https://d111111abcdef8.cloudfront.net/image.jpg`

   如果您在建立您的分佈時指定了一或多個備用網域名稱 (CNAME)，則可以使用自己的網域名稱。在這種情況下，`image.jpg` 的 URL 可能`https://www.example.com/image.jpg`。

   注意下列事項：

   - 如果您想要使用簽章的 URL 來限制對內容的存取的詳細資訊，請參閱使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。
   - 如果您想要提供壓縮內容的詳細資訊，請參閱提供壓縮檔案。
   - 如需 Amazon S3 和自訂來源的 CloudFront 請求和回應行為的相關資訊，請參閱請求和回應行為。

主題

建立分發

本主題說明如何使用主 CloudFront 控台建立發行版。其他有用的主題包括下列內容：

- 若要了解如何建立使用具有來源存取控制 (OAC) 的 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體原點的分發，請參閱開始使用基本 CloudFront發行版。
- 如需使用 CloudFront API 建立分發的相關資訊，請參閱 Amazon CloudFront API 參考中的建立分發。

- 如需更新發行版的相關資訊 (例如，新增或變更快取行為)，請參閱更新分佈。
- 若要查看您可為每個 AWS 帳戶建立之分佈數量的目前上限，或是有關請求更高配額 (先前稱為限制) 的詳細資訊，請參閱 分佈的一般配額。

在主控台中建立 CloudFront 發行版

如要建立分佈 (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇分佈，然後選擇建立分佈。
3. 針對分佈所指定的設定。如需詳細資訊，請參閱 發佈設定參考。
4. 儲存您的變更。
5. CloudFront 建立發行版之後，發佈的 [狀態] 欄的值會從 [部署] 變更為部署發佈的日期和時間。如果您選擇啟用分發，它將可以在此時處理請求。

   CloudFront 指派給您的發行版的網域名稱會出現在發行版清單中。(它也顯示在所選用分佈的 General (一般) 索引標籤上。)

   > **Tip**
   >
   > 您可以使用替代網域名稱，而不是指派給您的名稱 CloudFront；請遵循中的步驟新增替代網域名稱 (CNAME) 以使用自訂 URL。

6. 部署分發後，請確認您可以使用新的 CloudFront URL 或 CNAME 存取您的內容。如需詳細資訊，請參閱 測試發行版。

CloudFront 顯示在主控台中的值

當您建立新的發行版或更新現有發行版時， CloudFront 會在 CloudFront 主控台中顯示下列資訊。

> **Note**
>
> 目前在 CloudFront 主控台中看不到作用中的受信任簽署者、具有作用中 CloudFront key pair 且可用來建立有效簽署 URL 的 AWS 帳戶。

分佈 ID

當您使用 CloudFront API 對分發執行動作時，您可以使用分發 ID 來指定要使用的分佈，例如，EDFDVBD6EXAMPLE。分佈的分佈 ID 不得變更。

部署和狀態

部署發行版時，您會在 [上次修改] 欄下看到 [部署] 狀態。等待發行版完成部署，並確定 [狀態] 欄顯示 [已啟用]。如需詳細資訊，請參閱 分佈狀態。

上次修改

上次修改的分佈日期和時間，使用 ISO 8601 格式，例如 2012-05-19T19:37:58Z。如需詳細資訊，請參閱 https://www.w3.org/TR/NOTE-datetime。

網域名稱

您在指向物件的連結中使用分佈的網域名稱。例如，如果分佈的網域名稱是 d111111abcdef8.cloudfront.net，則 /images/image.jpg 的連結將是 https://d111111abcdef8.cloudfront.net/images/image.jpg。您無法變更分發的 CloudFront 網域名稱。如需物件連結之 CloudFront URL 的詳細資訊，請參閱自訂中檔案的 URL 格式 CloudFront。

如果您指定了一或多個替代網域名稱 (CNAME)，您可以使用自己的網域名稱連結至物件，而不是使用 CloudFront 網域名稱。如需 CNAME 的詳細資訊，請參閱備用網域名稱 (CNAME)。

> **Note**
>
> CloudFront 網域名稱是唯一的。您的分佈的網域名稱從未用於之前的分佈，並且未來將永遠不會重複使用於另一個分佈。

發佈設定參考

當您使用主CloudFront控台建立新的發行版或更新現有發行版時，請指定下列值。

如需使用主控台建立或更新發行版的詳細資 CloudFront訊，請參閱the section called “建立分發”或the section called “更新分佈”。

主題

- 原始設定
- 快取行為設定

- 分佈設定
- 自訂錯誤頁面和錯誤快取
- 地理限制

原始設定

當您使用 CloudFront 主控台建立或更新發佈時，您會提供一或多個位置的相關資訊，也就是來源，您可以在其中儲存 Web 內容的原始版本。 CloudFront 從您的來源獲取您的 Web 內容，並通過全球邊緣服務器網絡將其提供給觀眾。

如需您可為分佈建立之原始伺服器數量的目前上限，或是有關請求更高配額的詳細資訊，請參閱 the section called “分佈的一般配額”。

如果您想要刪除原始來源，則必須先編輯或刪除與該原始來源相關聯的快取行為。

> **Important**
>
> 如果您要刪除原始來源，請確認之前由該原始來源提供的檔案，可在另一個原始來源中使用，而且您的快取行為現在正將這些檔案的請求，路由傳送到新的原始來源。

當您建立或更新分佈時，請為每個原始來源指定以下值。

主題

- 原始網域
- 通訊協定 (僅限自訂原始伺服器)
- 原始伺服器路徑
- 名稱
- 原始存取 (僅限 Amazon S3 原始伺服器)
- 新增自訂標頭
- 啟用 Origin Shield
- 連線嘗試
- 連線逾時。
- 回應逾時 (僅限自訂原始伺服器)
- 保持連線逾時 (僅限自訂原始伺服器)

- 回應和保持作用逾時配額

原始網域

原始網域是您要 CloudFront 從中取得此來源物件之 Amazon S3 儲存貯體或 HTTP 伺服器的 DNS 網域名稱，例如：

- Amazon S3 儲存貯體 – *DOC-EXAMPLE-BUCKET*.s3.*us-west-2*.amazonaws.com

> **Note**
>
> 如果您最近建立了 S3 儲存貯體，則 CloudFront 分發最多可能會傳HTTP 307 Temporary Redirect回 24 小時的回應。S3 儲存貯體名稱最多可能需要 24 小時才能傳播到所有 AWS 區域。傳播完成後，分佈會自動停止傳送這些重新引導回應；您不需要執行任何動作。如需詳細資訊，請參閱為何會收到來自 Amazon S3 的 HTTP 307 暫時重新引導回應？和暫時請求重新引導。

- 設定為網站的 Amazon S3 儲存貯體 – *DOC-EXAMPLE-BUCKET*.s3-website.*us-west-2*.amazonaws.com
- MediaStore 容器 — *examplemediastore*.data.mediastore.*us-west-1*.amazonaws.com
- MediaPackage 端點 — *examplemediapackage*.mediapackage.*us-west-1*.amazonaws.com
- Amazon EC2 執行個體 – *ec2-203-0-113-25*.compute-1.amazonaws.com
- Elastic Load Balancing 負載平衡器 – *example-load-balancer-1234567890*.*us-west-2*.elb.amazonaws.com
- 您自己的 Web 伺服器 – https://www.example.com

在 Origin domain (原始網域) 欄位中選擇網域名稱，或是輸入名稱。網域名稱不區分大小寫。

如果您的原始伺服器是 Amazon S3 儲存貯體，請注意下列事項：

- 若儲存貯體已設為網站，請輸入您儲存貯體的 Amazon S3 靜態網站託管端點；請不要從 Origin domain (原始網域) 欄位中的清單選取儲存貯體名稱。靜態網站託管端點顯示在 Amazon S3 主控台中，在 Static website hosting (靜態網站託管) 下的 Properties (屬性) 頁面上。如需詳細資訊，請參閱 the section called “使用設定為網站端點的 Amazon S3 儲存貯體”。
- 如果您為儲存貯體配置了 Amazon S3 Transfer Acceleration，請不要為 Origin domain (原始網域)指定 s3-accelerate 端點。

- 如果您使用不同 AWS 帳戶的值區，而且該值區未設定為網站，請使用下列格式輸入名稱：

  bucket-name.s3.*region*.amazonaws.com

  如果您的儲存貯體位於美國區域，而且您希望 Amazon S3 將請求路由到維吉尼亞北部的一個設施，請使用以下格式：

  bucket-name.s3.us-east-1.amazonaws.com

- 除非您使用 CloudFront 來源存取控制來保護 Amazon S3 中的內容，否則檔案必須可公開讀取。如需存取控制的詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。

> **Important**
>
> 如果該原始伺服器是 Amazon S3 儲存貯體時，則儲存貯體名稱必須符合 DNS 命名請求。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的儲存貯體限制與局限。

當您變更原點的 Origin 網域值時，會 CloudFront 立即開始將變更複製到 CloudFront 邊緣位置。在指定邊緣位置更新發佈組態之前，會 CloudFront 繼續將要求轉寄至先前的原點。一旦在該節點中更新發佈組態，就會 CloudFront開始將要求轉寄至新的原始伺服器。

變更原點不需要使用 CloudFront 來自新來源的物件重新填入 Edge 快取。只要應用程式中的檢視器要求沒有變更，就會 CloudFront 繼續提供已在邊緣快取中的物件，直到每個物件上的 TTL 過期或很少要求的物件被收回為止。

通訊協定 (僅限自訂原始伺服器)

> **Note**
>
> 這僅適用於自訂原始伺服器。

從來源擷取物件時 CloudFront 要使用的通訊協定原則。

請選擇下列其中一個值：

- 僅限 HTTP：僅 CloudFront 使用 HTTP 來訪問原點。

> **Important**
>
> 當原始伺服器是 Amazon S3 靜態網站託管端點時，HTTP only (僅限 HTTP) 是預設設定，因為 Amazon S3 不支援 HTTPS 連線的靜態網站託管端點。主 CloudFront 控台不支援針對 Amazon S3 靜態網站託管端點變更此設定。

- 僅限 HTTPS：僅 CloudFront 使用 HTTPS 來存取原始伺服器。
- 匹配查看器：根據查看者請求的協議，使用 HTTP 或 HTTPS 與您的來源 CloudFront 進行通信。CloudFront 即使檢視者同時使用 HTTP 和 HTTPS 通訊協定發出要求，也只會快取物件一次。

> **Important**
>
> 對於 CloudFront 轉寄至此原始伺服器的 HTTPS 檢視器要求，原始伺服器上 SSL/TLS 憑證中的其中一個網域名稱必須與您為原始網域指定的網域名稱相符。否則，使用 HTTP 狀態碼 502（錯誤的網關） CloudFront 響應查看器請求，而不是返回請求的對象。如需詳細資訊，請參閱 the section called “搭配使用 SSL/TLS 憑證的需求 CloudFront”。

主題

- HTTP 連接埠
- HTTPS 連接埠
- 最低來源 SSL 通訊協定

HTTP 連接埠

> **Note**
>
> 這僅適用於自訂原始伺服器。

(選擇性) 您可以指定自訂原始伺服器偵聽的 HTTP 連接埠。有效值包括連接埠 80、443，以及 1024 到 65535 之間。預設值為連接埠 80。

> **Important**
>
> 當原始伺服器是 Amazon S3 靜態網站託管端點時，連線埠 80 是預設設定，因為 Amazon S3 僅支援連線埠 80 的靜態網站託管端點。主 CloudFront 控台不支援針對 Amazon S3 靜態網站託管端點變更此設定。

HTTPS 連接埠

> **Note**
>
> 這僅適用於自訂原始伺服器。

(選擇性) 您可以指定自訂原始伺服器偵聽的 HTTPS 連接埠。有效值包括連接埠 80、443，以及 1024 到 65535 之間。預設值為連接埠 443。當 Protocol (通訊協定) 設定為 HTTP only (僅限 HTTP)，您無法指定 HTTPS port (HTTPS 連接埠) 值。

最低來源 SSL 通訊協定

> **Note**
>
> 這僅適用於自訂原始伺服器。

選擇建立與原始伺服器的 HTTPS 連線時， CloudFront 可使用的最低 TLS/SSL 通訊協定。由於較低的 TLS 通訊協定安全性較低，因此我們建議您選擇您來源支援的最新 TLS 通訊協定。當 Protocol (通訊協定) 設定為 HTTP only (僅限 HTTP)，您無法指定 Minimum origin SSL protocol (最低來源 SSL 通訊協定) 值。

如果您使用 CloudFront API 來設定要使用的 TLS/SSL 通訊協 CloudFront 定，則無法設定最小通訊協定。相反地，您 CloudFront 可以指定可與原始伺服器搭配使用的所有 TLS/SSL 通訊協定。如需詳細資訊，請參閱 Amazon CloudFront API 參考資料OriginSslProtocols中的。

原始伺服器路徑

如果您想 CloudFront 要從原始目錄中要求內容，請輸入以斜線 (/) 開頭的目錄路徑。 CloudFront 會將目錄路徑附加至 Origin 網域的值，**`cf-origin.example.com/production/images`**例如。請勿在路徑的尾端加上斜線 (/)。

例如，假設您已為您的分佈指定以下值：

- Origin domain (原始網域) – 名為 **DOC-EXAMPLE-BUCKET** 的 Amazon S3 儲存貯體
- Origin path (原始伺服器路徑) - **/production**
- Alternate domain names (CNAME) (備用網域名稱 (CNAME)) – **example.com**

當使用者在瀏覽器example.com/index.html中輸入時， CloudFront會將請求傳送至 Amazon S3 以下項目DOC-EXAMPLE-BUCKET/production/index.html。

當使用者在瀏覽器example.com/acme/index.html中輸入時，CloudFront 會將請求傳送至 Amazon S3 以下項目DOC-EXAMPLE-BUCKET/production/acme/index.html。

名稱

名稱是一個可唯一識別該分佈中此原始伺服器的字串。如果您除了預設快取行為之外建立快取行為，您可以使用您在此處指定的名稱來識別要路由 CloudFront 要求的來源，當要求符合該快取行為的路徑模式時。

原始存取 (僅限 Amazon S3 原始伺服器)

> **Note**
>
> 這僅適用於 Amazon S3 儲存貯體原始伺服器 (那些不使用 S3 靜態網站端點的原始伺服器)。

如果您想要將 Amazon S3 儲存貯體來源的存取限制為僅限特定 CloudFront 分發，請選擇來源存取控制設定 (建議使用)。

如果 Amazon S3 儲存貯體可公開存取，請選擇 public (公有)。

如需詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。

如需如何要求使用者僅使用 CloudFront URL 存取自訂來源上的物件的資訊，請參閱the section called “ 在自訂原始伺服器上限制存取檔案”。

新增自訂標頭

如果您想 CloudFront 要在傳送要求到來源時新增自訂標頭，請指定標頭名稱及其值。如需詳細資訊，請參閱 the section called “將自訂標頭新增到原始伺服器請求”。

如需可新增自訂標頭數量上限的目前上限、自訂標頭的名稱和值的最大長度，以及所有標頭名稱和值的總長度上限的詳細資訊，請參閱 配額。

啟用 Origin Shield

選擇「是」 以啟用「 CloudFront 原點護 Shield」。如需 Origin Shield 的詳細資訊，請參閱 the section called “使用 Origin Shield”。

連線嘗試

您可以設定 CloudFront 嘗試連線到原點的次數。您可以指定 1、2 或 3 做為嘗試次數。預設數字 (如果您未另外指定) 是 3。

將此設定與連線逾時一起使用，可指定在嘗試連線至次要原始伺服器或傳回錯誤回應給檢視器之前要 CloudFront 等待的時間長度。根據預設， CloudFront 在嘗試連線至次要原點或傳回錯誤回應之前，會等待 30 秒 (每次嘗試 10 秒)。您可以指定較少的嘗試次數、較短的連線逾時或兩者，以縮短此時間。

如果指定的連線嘗試次數失敗，請 CloudFront 執行下列其中一項作業：

- 如果原點是原點群組的一部分，會 CloudFront 嘗試連接至次要原點。如果指定次要原點的連線嘗試次數失敗，則會將錯誤 CloudFront 回應傳回給檢視器。
- 如果原點不是原始群組的一部分，則會 CloudFront 傳回錯誤回應給檢視器。

對於自訂來源 (包括使用靜態網站託管設定的 Amazon S3 儲存貯體)，此設定也會指定 CloudFront 嘗試從來源取得回應的次數。如需詳細資訊，請參閱 the section called “回應逾時 (僅限自訂原始伺服器)”。

連線逾時。

連線逾時是嘗試建立與原點 CloudFront 的連線時等待的秒數。您可以指定介於 1 到 10 之間的秒數 (含)。預設逾時 (如果您另外未指定) 是 10 秒。

將此設定與「連線」嘗試一起使用，可指定在嘗試連線至次要原點之前或將錯誤回應傳回給檢視器之前要 CloudFront 等待的時間長度。根據預設， CloudFront在嘗試連線至次要原點或傳回錯誤回應之前，會等待 30 秒 (每次嘗試 10 秒)。您可以指定較少的嘗試次數、較短的連線逾時或兩者，以縮短此時間。

如果 CloudFront 未在指定的秒數內建立與原點的連線，請 CloudFront 執行下列其中一項作業：

- 如果指定的連線嘗試次數超過 1 次，請再次 CloudFront 嘗試建立連線。 CloudFront 嘗試最多 3 次，由連線嘗試的值決定。

- 如果所有連線嘗試都失敗，且原點是原點群組的一部分，則會 CloudFront 嘗試連接至次要原點。如果指定次要原點的連線嘗試次數失敗，則會將錯誤 CloudFront 回應傳回給檢視器。
- 如果所有連線嘗試都失敗，且來源不是原始群組的一部分，則會將錯誤 CloudFront 回應傳回給檢視器。

回應逾時 (僅限自訂原始伺服器)

原始伺服器回應逾時，也稱為原始伺服器讀取逾時或原始伺服器請求逾時，適用於以下兩個值：

- 將請求轉發到來源後， CloudFront 等待響應的時間長度（以秒為單位）。
- 在收到來自來源的回應封包 CloudFront 之後，以及接收下一個封包之前，等待多久 (以秒為單位)。

> **Tip**
>
> 如果因為檢視器遇到 HTTP 504 狀態碼錯誤，而您想要增加逾時值，請考慮在變更逾時值之前先探索其他方式來消除這些錯誤。請查看在 the section called “HTTP 504 狀態碼 (閘道逾時)” 中的故障診斷建議。

CloudFront 行為取決於檢視器要求中的 HTTP 方法：

- `GET`和`HEAD`請求 — 如果來源沒有響應或在響應超時期間內停止響應，則中 CloudFront 斷連接。CloudFront 再次嘗試根據的值進行連線the section called “連線嘗試”。
- `DELETE`、`OPTIONS`、`PATCH`和`PUT`要求 — 如果來源在讀取逾時期間沒有回應， CloudFront 請中斷連線，而不會再次嘗試聯絡來源。用戶端可以視需要重新提交請求。

保持連線逾時 (僅限自訂原始伺服器)

保持活動超時是指在獲取響應的最後一個數據包後， CloudFront 嘗試維護與自定義來源的連接的時間（以秒為單位）。維護持久性連線可節省重新建立 TCP 連線所需的時間，並為後續請求執行另一個 TLS 交握。增加保持活動逾時有助於改善分配的 request-per-connection 量度。

> **Note**
>
> 為了讓 Keep-alive Timeout (保持連線逾時) 值有效，必須將原始伺服器設定為允許持久連線。

回應和保持作用逾時配額

> **Note**
>
> 這僅適用於自訂原始伺服器。

- 若為回應逾時，預設值為 30 秒。
- 對於保持活動超時，默認值為 5 秒。
- 對於任一配額，您都可以指定 1 到 60 秒之間的值。若要請求增加，請在 AWS Support Center Console/中建立案例。

在您要求您的逾時增加之後 AWS 帳戶，請更新您的發行版來源，讓它們具有您想要的回應逾時和保持使用中逾時值。增加帳戶的配額不會自動更新您的來源。例如，如果您使用 Lambda @Edge 函數將保持作用逾時設定為 90 秒，則您的原始伺服器必須已有 90 秒或更長時間的保持作用逾時。否則，您的 Lambda @Edge 函數可能無法執行。

如需有關發佈配額的詳細資訊，請參閱分佈的一般配額。

快取行為設定

透過設定快取行為，您可以為網站上檔案的指定 URL 路徑模式設定各種 CloudFront 功能。例如，一種快取行為可能會套用至您用作原始伺服器之網頁伺服器`images`目錄中的所有`.jpg`檔案 CloudFront。您可以為每個快取行為設定的功能包括：

- 路徑模式
- 如果您已經為您的 CloudFront 發行版設定了多個來源，那麼您 CloudFront 要轉寄請求的來源
- 是否將查詢字串轉發到您的原始伺服器。
- 存取指定的檔案是否需要已簽署的 URL
- 是否要求使用者使用 HTTPS 來存取這些檔案
- 無論您的來源新增至檔案的任何`Cache-Control`標頭值為何，這些檔案仍保留在 CloudFront 快取中的最短時間

在建立新的分佈，您可以指定預設快取行為的設定，該設定會自動將所有請求轉發到您在建立分佈時指定的原始伺服器。建立發佈之後，您可以建立額外的快取行為，以定義收到符合路徑模式之物件的要求時的 CloudFront 回應方式，例如，`*.jpg`。如果您建立額外的快取行為，則預設的快取行為一律是最

後要處理的。其他快取行為的處理順序會依照 CloudFront 主控台中的列出順序進行處理，如果您使用的是 CloudFront API，則會依照它們在發佈項`DistributionConfig`目中的列出順序進行處理。如需詳細資訊，請參閱 路徑模式。

當您建立快取行為時，您可 CloudFront 以指定要從中取得物件的一個來源。因此，如果您想 CloudFront 要從所有來源散佈物件，您至少必須擁有與起源相同數量的快取行為 (包括預設快取行為)。例如，如果您有兩個來源，而且只有預設快取行為，預設快取行為會導 CloudFront 致從其中一個來源取得物件，但是從不使用另一個來源。

如需您可為分發新增之快取行為數量的目前上限，或是有關請求更高配額 (先前稱為限制) 的詳細資訊，請參閱分佈的一般配額。

主題

- 路徑模式
- 來源或來源群組
- 檢視器通訊協定政策
- Allowed HTTP methods (允許的 HTTP 方法)
- 欄位層級加密 Config
- 快取的 HTTP 方法
- 根據選取請求標頭的快取
- 允許清單標頭
- 物件快取
- 最短 TTL
- 最長 TTL
- 預設 TTL
- 轉送 Cookie
- 允許清單 Cookie
- 查詢字串轉送和快取
- 查詢字串允許清單
- Smooth Streaming
- 限制檢視器存取 (使用已簽章的 URL 或已簽章的 Cookie)
- 可信簽署者

- AWS 帳戶 數字
- 自動壓縮物件
- CloudFront 事件
- Lambda 函數 ARN
- 包含內文

路徑模式

路徑模式 (例如，images/*.jpg) 指定您希望將此快取行為套用到哪些請求。當 CloudFront 收到一般使用者要求時，系統會將要求的路徑與路徑模式進行比較，並依照發行版中列出快取行為的順序進行比較。第一種符合決定哪個快取行為套用到該請求。例如，假設您在以下三種路徑模式下有三個快取行為模式，並按此順序排列：

- images/*.jpg
- images/*
- *.gif

> **Note**
>
> 您可以選擇性地在路徑模式的開頭加入斜線 (/)，例如/images/*.jpg。 CloudFront 行為與或沒有前導/相同。如果您未在路徑的開頭指定/，系統會自動隱含此字元；會 CloudFront 將路徑視為相同，不論是否有前導/。例如，CloudFront 對待/*product.jpg相同 *product.jpg

對檔案 images/sample.gif 的請求無法滿足第一個路徑模式，所以關聯的快取行為不能套用到該請求。該檔案確實滿足第二個路徑模式，因此即使該請求也與第三個路徑模式相符合，也會套用與第二個路徑模式相關聯的快取行為。

> **Note**
>
> 當您建立新的分佈時，預設快取行為的 Path Pattern (路徑模式) 值將設定為 * (所有檔案)，而且無法變更。此值會 CloudFront 將物件的所有要求轉寄至您在原始網域欄位中指定的來源。如果物件的要求不符合任何其他快取行為的路徑模式，會 CloudFront 套用您在預設快取行為中指定的行為。

> **Important**
>
> 請謹慎定義路徑模式及其序列，否則可能會讓使用者對您的內容作出無法預料的存取。例如，假設請求與兩個快取行為的路徑模式相符合。第一個快取行為不需要簽章的 URL 和第二個快取行為確實需要簽章的 URL。使用者可以在不使用已簽署 URL 的情況下存取物件，因為會 CloudFront 處理與第一個相符項目相關聯的快取行為。

如果您正在使用 MediaPackage 通道，則必須包含針對您為來源端點類型定義的快取行為的特定路徑模式。例如，如果是 DASH 端點，請在 路徑模式 中，輸入 `*.mpd`。如需詳細資訊和特定說明，請參閱提供以 AWS Elemental MediaPackage 格式化的即時視訊。

您指定的路徑會套用至指定目錄及指定目錄下子目錄中所有檔案的要求。 CloudFront 評估路徑模式時，不會考慮查詢字串或 Cookie。例如，如果 `images` 目錄包含 `product1` 和 `product2` 子目錄，則路徑模式 `images/*.jpg` 適用於 `images`、`images/product1` 和 `images/product2` 目錄中任何 .jpg 檔案的請求。如果要對 `images/product1` 目錄中的檔案套用與 `images` 和 `images/product2` 目錄中的套用不同的快取行為，請為 `images/product1` 建立一個單獨的快取行為，並將該快取行為移至 `images` 目錄快取行為上方 (之前) 的位置。

您可以在路徑模式中使用以下萬用字元：

- `*` 符合 0 或更多字元。
- `?` 符合完全 1 個字元。

以下範例說明萬用字元的工作方式：

路徑模式	與路徑模式相符的檔案
`*.jpg`	所有 .jpg 檔案。
`images/*.jpg`	位於 `images` 目錄和 `images` 目錄下子目錄中的所有 .jpg 檔案。
`a*.jpg`	• 檔案名稱以 a 開頭的所有 .jpg 檔案，例如 `apple.jpg` 和 `appalachian_trail_2012_05_21.jpg` 。

路徑模式	與路徑模式相符的檔案
	• 檔案路徑以 a 開頭的所有 .jpg 檔案，例如 abra/cadabra/magic.jpg 。
a??.jpg	檔案名稱以 a 開頭的所有 .jpg 檔案，後面緊接恰好兩個其他字元，例如 ant.jpg 和 abe.jpg。
.doc	檔案名稱副檔名以 .doc 開頭的所有檔案，例如 .doc、.docx 和 .docm 檔案。在這種情況下，您不能使用路徑模式*.doc?，因為該路徑模式不適用於 .doc 檔案的請求；? 萬用字元僅取代一個字元。

路徑模式 的長度上限為 255 個字元。這個值可以包含以下任何字元：

- A-Z、a-z

  途徑模式會區分大小寫，因此路徑模式 *.jpg 不適用於檔案 LOGO.JPG
- 0-9
- _ - . * $ / ~ " ' @ : +
- & 通過並以 & 傳回

路徑規範化

CloudFront 標準化與 RFC 3986 一致的 URI 路徑，然後將路徑與正確的緩存行為匹配。一旦快取行為相符，就會將原始 URI 路徑 CloudFront 傳送至原始位置。如果它們不匹配，請求將與您的默認緩存行為匹配。

某些字元會標準化並從路徑中移除，例如多條斜線 (//) 或句點 (..)。這可能會變更 CloudFront 用來符合預期快取行為的 URL。

Example 範例

您可以指定快取行為的/a/b*和/a*路徑。

- 傳送/a/b?c=1路徑的檢視器將符合/a/b*快取行為。

- 傳送/a/b/..?c=1路徑的檢視器將符合/a*快取行為。

若要解決正在規範化的路徑，您可以更新要求路徑或快取行為的路徑模式。

來源或來源群組

只有當您為現有發行版建立或更新快取行為時，才會套用此設定。

輸入現有來源或來源群組的值。這會識別要求 (例如 https://example.com/logo.jpg) 符合快取行為的路徑模式 (例如 *.jpg) 或預設快取行為 (*) 時，您要 CloudFront 將要路由傳送要求的原始或原始群組。

檢視器通訊協定政策

選擇您希望檢視者用來存取 CloudFront 節點內容的通訊協定原則：

- HTTP and HTTPS (HTTP 和 HTTPS)：檢視器可使用這兩種通訊協定。
- Redirect HTTP to HTTPS (將 HTTP 重新引導到 HTTPS)：檢視器可使用這兩種通訊協定，但是 HTTP 請求會自動重新引導到 HTTPS 請求。
- HTTPS Only (僅限 HTTPS)：檢視器只能在使用 HTTPS 的情況下存取您的內容。

如需詳細資訊，請參閱 要求使用 HTTPS 才能在檢視者和 CloudFront。

Allowed HTTP methods (允許的 HTTP 方法)

指定您要處理並轉寄 CloudFront 至原始伺服器的 HTTP 方法：

- GET，HEAD：您 CloudFront 只能使用來從您的來源獲取對象或獲取對象標題。
- GET、HEAD、OPTIONS：您CloudFront 只能使用從原始伺服器取得物件、取得物件標頭或擷取原始伺服器支援的選項清單。
- 取得、標頭、選項、PUT、POST、修補程式、刪除：您可以使用 CloudFront 來取得、新增、更新和刪除物件，以及取得物件標頭。此外，您可以執行其他 POST 操作，例如從 Web 表單提交資料。

> **Note**
>
> CloudFront 快取回應GET和要HEAD求，以及 (選擇性) OPTIONS 要求。對要OPTIONS求的回應會GET與要HEAD求的回應分開快取 (該OPTIONS方法包含在OPTIONS要求的快取金鑰中)。 CloudFront 不會快取回應至使用其他方法的要求。

> **Important**
>
> 如果您選擇 GET、HEAD、OPTIONS 或 GET、HEAD、OPTIONS、PUT、POST、PATCH、DELETE，則可能需要限制對 Amazon S3 儲存貯體或自訂原始伺服器原始伺服器的存取，以防止使用者執行不希望它們執行的操作。下列範例說明如何限制存取：
>
> - 如果您使用 Amazon S3 做為分發的來源：建立 CloudFront 來源存取控制以限制對 Amazon S3 內容的存取，並授予原始存取控制的許可。例如，如果您設定 CloudFront 為僅因為想要使用而接受和轉寄這些方法PUT，您仍然必須設定 Amazon S3 儲存貯體政策以適當處理DELETE請求。如需詳細資訊，請參閱 限制對 Amazon 簡單儲存服務來源的存取。
> - 如果您使用自訂原始伺服器：請設定您的原始伺服器以處理所有方法。例如，如果您設定 CloudFront 為僅因為想要使用而接受和轉寄這些方法POST，您仍然必須將原始伺服器設定為適當地處理要DELETE求。

欄位層級加密 Config

如果想要針對特定資料欄位強制執行欄位層級加密，請在下拉式清單中選擇欄位層級加密組態。

如需詳細資訊，請參閱 使用欄位層級加密來協助保護敏感資料。

快取的 HTTP 方法

指定當檢視者提交OPTIONS請求時，是否要 CloudFront 快取來源的回應。 CloudFront 始終緩存響應GET和HEAD請求。

根據選取請求標頭的快取

指定是否 CloudFront 要根據指定標頭的值快取物件：

- 無（改善緩存）- CloudFront 不會根據標題值緩存對象。
- 允許清單 — 僅根據指定標頭的值 CloudFront 快取物件。使用 [允許清單標題] 選擇要以快取 CloudFront 為基礎的標頭。
- 全部 — CloudFront 不會快取與此快取行為相關聯的物件。相反， CloudFront 將每個請求發送到原點。(不建議用於 Amazon S3 原始伺服器。)

無論您選擇哪個選項，都會將特定標頭 CloudFront 轉寄至您的來源，並根據您轉寄的標頭採取特定動作。如需如何 CloudFront 處理標頭轉寄的詳細資訊，請參閱HTTP 請求標頭和 CloudFront 行為 (自訂和 Amazon S3 來源)。

如需如何使用要求標頭在中設定快取 CloudFront 的詳細資訊，請參閱根據請求標頭快取內容。

允許清單標頭

只有當您根據選取的要求標頭選擇允許快取清單時，才會套用這些設定。

指定快取物件時 CloudFront 要考量的標頭。從可用標頭清單中選擇標頭，然後選擇 Add (新增)。若要轉送自訂標頭，請在欄位中輸入標頭的名稱，然後選擇 Add Custom (新增自訂)。

如需您可為每個快取行為列入允許清單之標頭數量的目前上限，或是有關請求更高配額 (先前稱為限制) 的詳細資訊，請參閱標頭的配額。

物件快取

如果您的原始伺服器要在物件中新增標`Cache-Control`頭，以控制物件在 CloudFront 快取中停留的時間長度，如果您不想變更該`Cache-Control`值，請選擇「使用原始快取標頭」。

若要指定物件停留在 CloudFront快取中的時間下限與上限 (不論`Cache-Control`標頭為何)，以及物件遺失標`Cache-Control`頭時物件停留在 CloudFront 快取中的預設時間，請選擇「自訂」。然後，在 Minimum TTL (最短 TTL)、Default TTL (預設 TTL) 及 Maximum TTL (最長 TTL) 欄位中指定值。

如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

最短 TTL

指定在 CloudFront 將另一個要求傳送至來源以判斷物件是否已更新之前，您希望物件停留在 CloudFront 快取記憶體中的時間下限 (以秒為單位)。

如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

最長 TTL

指定在 CloudFront 查詢來源以查看物件是否已更新之前，要讓物件保留在 CloudFront 快取記憶體中的時間上限 (秒)。您為 Maximum TTL (最長 TTL) 指定的值僅適用於您的原始伺服器將 HTTP 標題 (例如 `Cache-Control max-age`、`Cache-Control s-maxage` 或 `Expires`) 新增至物件時。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

若要指定 Maximum TTL (最長 TTL) 的值，您必須為 Object Caching (物件快取) 設定選擇 Customize (自訂) 選項。

Maximum TTL (最長 TTL) 的預設值為 31536000 秒 (一年)。如果將 Minimum TTL (最短 TTL) 或 Default TTL (預設 TTL) 的值變更為 31536000 秒以上，則 Maximum TTL (最長 TTL) 的預設值將變更為 Default TTL (預設 TTL) 的值。

預設 TTL

指定物件在將另一個要求 CloudFront 轉送至您的來源以判斷物件是否已更新之前，要讓物件保留在 CloudFront 快取記憶體中的預設時間 (秒)。您為 Default TTL (預設 TTL) 指定的值僅適用於您的原始伺服器不會將 HTTP 標題 (例如 `Cache-Control max-age`、`Cache-Control s-maxage` 或 `Expires`) 新增至物件時。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

若要指定 Default TTL (預設 TTL) 的值，您必須為 Object Caching (物件快取) 設定選擇 Customize (自訂) 選項。

Default TTL (預設 TTL) 的預設值為 86400 秒 (一天)。如果將 Minimum TTL (最短 TTL) 的值變更為 86400 秒以上，則 Default TTL (預設 TTL) 的預設值將變更為 Minimum TTL (最短 TTL) 的值。

轉送 Cookie

> **Note**
>
> 對於 Amazon S3 原始伺服器，此選項僅適用於配置為網站端點的儲存貯體。

指定是否要 CloudFront 將 Cookie 轉寄至您的原始伺服器，如果是，則指定要將哪些 Cookie 轉寄至原始伺服器。如果您選擇僅轉送所選取的 Cookie (Cookie 的允許清單)，請在允許清單 Cookie 欄位中輸入 Cookie 名稱。如果您選擇「全部」，則無論您的應用程式使用多少次，都會 CloudFront轉送所有 Cookie。

Amazon S3 並不處理 Cookie，而且將 Cookie 轉送到原始伺服器的動作，會降低快取的能力。對於將請求轉送到 Amazon S3 原始伺服器的快取行為，請為轉送 Cookie選擇無。

如需有關將 Cookie 轉發到原始伺服器的詳細資訊，請前往根據 Cookie 快取內容。

允許清單 Cookie

> **Note**
>
> 對於 Amazon S3 原始伺服器，此選項僅適用於配置為網站端點的儲存貯體。

如果您在 [轉寄 Cookie] 清單中選擇 [允許清單]，然後在 [允許清單 Cookie] 欄位中，輸入您要針對此快取行為轉寄 CloudFront至原始伺服器的 Cookie 名稱。在新的列上輸入每個 Cookie 名稱。

您可以指定以下萬用字元，以指定 Cookie 名稱：

- * 符合 Cookie 名稱中的 0 個或多個字元
- ? 僅符合 Cookie 名稱中的一個字元

例如，假設檢視器為包含 Cookie 的物件作請求，其 Cookie 名為：

`userid_`*`member-number`*

其中每個使用者都有*####*的唯一值。您想 CloudFront 要為每個成員快取物件的個別版本。您可以通過將所有 cookie 轉發到您的來源來完成此操作，但查看者請求包含一些您不 CloudFront 想緩存的 cookie。或者，您可以指定以下值作為 cookie 名稱，這會導 CloudFront 致所有以開頭的 cookie 轉發到原點`userid_`：

```
userid_*
```

如需您可為每個快取行為列入允許清單之 Cookie 名稱的目前上限，或是有關請求更高配額 (先前稱為限制) 的詳細資訊，請參閱Cookie 的配額 (舊版快取設定)。

查詢字串轉送和快取

CloudFront 可以根據查詢字串參數的值快取不同版本的內容。請選擇下列其中一個選項：

無 (提升快取)

如果您的原始來源傳回物件的相同版本，而不考慮查詢字串參數的值，請選擇此選項。這會增加 CloudFront 可以提供快取要求的可能性，進而改善效能並減少原始伺服器的負載。

全部轉送，依據允許清單進行快取

如果您的原始伺服器根據一或多個查詢字串參數傳回物件的不同版本，請選擇此選項。然後指定 CloudFront 要用作欄位中快取基礎的參查詢字串允許清單數。

轉發所有，根據所有快取

如果您的原始伺服器針對所有查詢字串參數傳回物件的不同版本，請選擇此選項。

如需有關根據查詢字串參數進行快取的詳細資訊，包括如何提升效能，請參閱根據查詢字串參數快取內容。

查詢字串允許清單

只有當您選擇全部轉寄，根據允許清單的快取時，才會套用此設定查詢字串轉送和快取。您可以指定要用作快取基礎 CloudFront 的查詢字串參數。

Smooth Streaming

如果您想要以 Microsoft Smooth Streaming 格式分配媒體檔案，而且您沒有 IIS 伺服器，請選擇 Yes (是)。

如果您有 Microsoft IIS 伺服器，而且想要將它當做以 Microsoft Smooth Streaming 格式分配媒體檔案的原始伺服器使用，或者您並非分配 Smooth Streaming 媒體檔案，請選擇 No (否)。

> **Note**
>
> 如果您指定 Yes (是)，而內容與 Path Pattern (路徑模式) 的值符合，您仍然可以使用此快取行為分配其他內容。

如需詳細資訊，請參閱 為 Microsoft Smooth Streaming 設定隨需視訊。

限制檢視器存取 (使用已簽章的 URL 或已簽章的 Cookie)

如果您希望與此快取行為之 `PathPattern` 相符的物件請求使用公有 URL，請選擇 No (否)。

如果您希望與此快取行為之 `PathPattern` 相符的物件請求使用已簽章的 URL，請選擇 Yes (是)。然後指定您要用來建立已簽署 URL 的 AWS 帳戶；這些帳戶稱為受信任的簽署者。

如需有關可信任簽署者的詳細資訊，請參閱指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。

可信簽署者

只有當您針對限制檢視者存取 (使用已簽署的 URL 或已簽署的 Cookie) 選擇是時，此設定才適用。

選擇您要用作此快取行為的信任簽署者的 AWS 帳戶：

- 自我：使用您目前登入的帳戶 AWS Management Console 做為信任的簽署者。如果您目前以 IAM 使用者身分登入，則相關聯的 AWS 帳戶會新增為受信任的簽署者。
- Specify Accounts (指定帳戶)：在 AWS Account Numbers (帳號) 欄位中輸入可信任簽署者的帳號。

若要建立已簽署的 URL， AWS 帳戶必須至少有一個作用中 CloudFront key pair。

> **Important**
>
> 如果您正在更新已用於分配內容的分佈，則只有在準備好開始為您的物件產生已簽章 URL 時，才會新增可信任簽署者。將可信任簽署者新增到分佈後，使用者必須使用已簽名的 URL 來存取與此 `PathPattern` 快取行為符合的物件。

AWS 帳戶 數字

只有當您選擇「指定信任簽署者的帳戶」時，才會套用此設定。

如果您要建立已簽署的 URL，而不是使用 AWS 帳戶 目前帳戶，請在此欄位中每行輸入一個 AWS 帳戶 數字。注意下列事項：

- 您指定的帳戶必須至少有一個作用中 CloudFront key pair。如需詳細資訊，請參閱 為您的簽署者建立金鑰對。
- 您無法為 IAM 使用者建立 CloudFront 金鑰配對，因此無法使用 IAM 使用者做為受信任的簽署者。
- 如需如何取得帳戶 AWS 帳戶 號AWS 帳戶 碼的詳細資訊，請參閱中的 Amazon Web Services 一般參考.
- 如果您輸入目前帳戶的帳號， CloudFront會自動勾選「自助」核取方塊，並從「帳號」清單中移除AWS 帳號。

自動壓縮物件

如果您想 CloudFront 要在檢視者支援壓縮內容時自動壓縮特定類型的檔案，請選擇 [是]。CloudFront 壓縮內容時，由於文件較小，因此下載速度更快，並且您的網頁為用戶渲染速度更快。如需詳細資訊，請參閱 提供壓縮檔案。

CloudFront 事件

此設定適用於 Lambda 函數關聯。

您可以選擇在發生下列一或多個CloudFront 事件時執行 Lambda 函數：

- CloudFront 收到來自檢視者的要求時 (檢視者要求)
- 在將請求 CloudFront 轉發到原始（原始請求）之前
- 當 CloudFront 收到來自來源的響應（原始響應）
- CloudFront 返回給查看者的響應之前（查看器響應）

如需詳細資訊，請參閱 決定要使用哪個 CloudFront 事件觸發 Lambda @Edge 函數。

Lambda 函數 ARN

此設定適用於 Lambda 函數關聯。

指定要為其新增觸發的 Lambda 函式的 Amazon 資源名稱 (ARN)。若要瞭解如何取得函數的 ARN，請參閱使用 CloudFront 主控台新增觸發程序程序的步驟 1。

包含內文

此設定適用於 Lambda 函數關聯。

如需詳細資訊，請參閱包括本體。

分佈設定

以下值適用於整個分佈。

主題

- 價格分級
- AWS WAF 網絡 ACL
- 備用網域名稱 (CNAME)
- SSL 憑證
- 自訂 SSL 用戶端支援
- 安全性政策 (最低 SSL/TLS 版本)
- 支援的 HTTP 版本
- 預設根物件
- 日誌
- 日誌儲存貯體
- 日誌字首
- Cookie 記錄
- 啟用 IPv6
- 註解
- 分佈狀態

價格分級

選擇與您要支付 CloudFront 服務的最高價格相對應的價格類別。依預設，會從所有 CloudFront 區域的邊緣位置 CloudFront 提供物件。

如需有關價格類別的詳細資訊，以及您選擇的價格等級如何影響發行版的 CloudFront 效能，請參閱CloudFront 定價。

AWS WAF 網絡 ACL

您可以使用 Web 應用程式防火牆來保護您的 CloudFront 散佈 AWS WAF，可讓您保護 Web 應用程式和 API，以便在要求到達伺服器之前封鎖要求。您可以AWS WAF 為分配啟用在建立或編輯發 CloudFront 佈時。

或者，您可以稍後在 AWS WAF 主控台中針對應用程式特定的其他安全威脅設定其他安全防護，網址為 https://console.aws.amazon.com/wafv2/。

如需詳細資訊 AWS WAF，請參閱AWS WAF 開發人員指南。

備用網域名稱 (CNAME)

選用。指定要用於物件 URL 的一或多個網域名稱，而不是在建立分發時指 CloudFront 派的網域名稱。您必須擁有網域名稱，或是擁有使用它的授權；您可以透過新增 SSL/TLS 憑證來進行驗證。

例如，若您希望物件的 URL：

```
/images/image.jpg
```

看起來像此 URL：

```
https://www.example.com/images/image.jpg
```

而不是此 URL：

```
https://d111111abcdef8.cloudfront.net/images/image.jpg
```

請新增 `www.example.com` 的 CNAME。

> **Important**
>
> 若您將 `www.example.com` 的 CNAME 新增到您的分佈，您也必須執行以下作業：
>
> - 使用您的 DNS 服務建立 (或更新) CNAME 記錄，以將 `www.example.com` 的查詢路由傳送到 `d111111abcdef8.cloudfront.net`。

- CloudFront 從受信任的憑證授權單位 (CA) 新增憑證，其中涵蓋您新增至發行版的網域名稱 (CNAME)，以驗證您使用網域名稱的授權。

  您必須擁有使用網域 DNS 服務提供者建立 CNAME 記錄的許可。通常，這表示您擁有網域，或者您正在為網域擁有者開發應用程式。

如需您可為分發新增之備用網域名稱數量的目前上限，或是有關請求更高配額 (先前稱為限制) 的詳細資訊，請參閱分佈的一般配額。

如需備用網域名稱的詳細資訊，請參閱新增替代網域名稱 (CNAME) 以使用自訂 URL。如需 CloudFront URL 的詳細資訊，請參閱自訂中檔案的 URL 格式 CloudFront。

SSL 憑證

若您指定要搭配分佈使用的備用網域名稱，請選擇 Custom SSL Certificate (自訂 SSL 憑證)，然後，為了驗證您的憑證使用其他網域名稱，請選擇涵蓋該名稱的憑證。如果您希望檢視器使用 HTTPS 存取您的物件，請選擇支援該功能的設定。

> **Note**
>
> 在您可以指定自訂 SSL 憑證之前，您必須指定有效的備用網域名稱。如需詳細資訊，請參閱使用備用網域名稱的需求 及 使用備用網域名稱和 HTTPS。

- 預設 CloudFront 憑證 (*.cloudfront.net) — 如果您想要在物件的 URL 中使用網 CloudFront 域名稱，例如，請選擇此選項。`https://d111111abcdef8.cloudfront.net/image1.jpg`
- 自訂 SSL 憑證 – 若您希望在物件的 URL 中使用您自己的網域名稱，做為備用網域名稱 (例如 `https://example.com/image1.jpg`)，請選擇此選項。然後要使用的憑證，其中涵蓋了備用網域名稱。憑證的清單可包括以下任何項目：
  - 證書提供 AWS Certificate Manager
  - 您從第三方憑證授權機構購買並上傳到 ACM 的憑證
  - 您從第三方憑證授權機構購買，並上傳到 IAM 憑證存放區的憑證

  若您選擇此設定，我們建議您僅使用您物件 URL 中的備用網域名稱 (https://example.com/logo.jpg)。如果您使用 CloudFront 散發網域名稱 (https://d111111abcdef8.cloudfront.net/logo.jpg)，而用戶端使用不支援 SNI 的舊版檢視器，檢視器的回應方式取決於您為支援的用戶端選擇的值：

- 所有用戶端：檢視器會顯示警告，因為 CloudFront 網域名稱與 SSL/TLS 憑證中的網域名稱不相符。
- 只有 Support 伺服器名稱指示 (SNI) 的用戶端：中 CloudFront 斷與檢視器的連線，而不傳回物件。

自訂 SSL 用戶端支援

只有當您為 SSL 憑證選擇自訂 SSL 憑證 (例如 .com) 時才適用。如果您為散發指定了一或多個替代網域名稱和自訂 SSL 憑證，請選擇您要如 CloudFront 何提供 HTTPS 要求：

- 支援伺服器名稱指示 (SNI) 的用戶端 - (建議) — 使用此設定，幾乎所有現代網頁瀏覽器和用戶端都可以連線到分佈，因為它們都支援 SNI。不過，有些檢視器可能會使用較舊的網頁瀏覽器或不支援 SNI 的用戶端，這表示他們無法連線至分佈。

  若要使用 CloudFront API 套用此設定，請在`SSLSupportMethod`欄位`sni-only`中指定。在中 AWS CloudFormation，欄位會命名為 `SslSupportMethod` (請注意不同的大小寫)。
- 舊式用戶端支援 — 使用此設定，不支援 SNI 的舊式網頁瀏覽器和用戶端就可以連線到分佈。不過，此設定每月會產生額外費用。有關確切價格，請轉到 Amazon CloudFront 定價頁面，然後在頁面上搜索專用 IP 自定義 SSL。

  若要使用 CloudFront API 套用此設定，請在`SSLSupportMethod`欄位`vip`中指定。在中 AWS CloudFormation，欄位會命名為 `SslSupportMethod` (請注意不同的大小寫)。

如需詳細資訊，請參閱 選擇如何 CloudFront 提供 HTTPS 要求。

安全性政策 (最低 SSL/TLS 版本)

指定您要用於與檢視者 (CloudFront 用戶端) 之間的 HTTPS 連線的安全性原則。安全政策判斷兩個設定：

- 用來與觀眾通訊的最低 SSL/ CloudFront TLS 通訊協定。
- CloudFront 可用來加密傳回給檢視者之內容的密碼。

如需安全原則 (包括每個原則所包含的通訊協定和密碼) 的詳細資訊，請參閱檢視器與之間支援的通訊協定和密碼 CloudFront。

可用的安全性原則取決於您為「SSL 憑證」和「自訂 SSL 用戶端 Support」(在 CloudFront API `SSLSupportMethod` 中稱為`CloudFrontDefaultCertificate`和) 指定的值：

- 當 SSL 憑證為預設CloudFront 憑證 (*.cloudfront.net) 時 (當在 API **`true`** 中時) 時， CloudFront 會自動將安全性原`CloudFrontDefaultCertificate`則設定為 TLSv1。
- 當 SSL Certificate (SSL 憑證) 為 Custom SSL Certificate (example.com) (自訂 SSL 憑證 (example.com))，且 Custom SSL Client Support (自訂 SSL 用戶端支援) 為 Clients that Support Server Name Indication (SNI) - (Recommended) (支援伺服器名稱指示 (SNI) 的用戶端 - (建議使用)) 時 (在 API 中則是 `CloudFrontDefaultCertificate` 為 `false` 且 `SSLSupportMethod` 為 `sni-only`)，您可以從下列安全政策中進行選擇：
  - TLSv1.2_2021
  - TLSv1.2_2019
  - TLSv1.2_2018
  - TLSv1.1_2016
  - TLSv1_2016
  - TLSv1
- 當 SSL Certificate (SSL 憑證) 為 Custom SSL Certificate (example.com) (自訂 SSL 憑證 (example.com))，且 Custom SSL Client Support (自訂 SSL 用戶端支援) 為 Legacy Clients Support (舊式用戶端支援) 時 (在 API 中則是 `CloudFrontDefaultCertificate` 為 `false` 且 `SSLSupportMethod` 為 `vip`)，您可以從下列安全政策中進行選擇：
  - TLSv1
  - SSLv3

  在此設定中，主控台或應用程式介面中無法使用安全性原則。 CloudFront 如果您想要使用這些安全政策之一，您可以選擇下列選項：
  - 評估您的分佈是否需要具有專用 IP 位址的舊式用戶端支援。如果您的檢視器支援伺服器名稱指示 (SNI)，我們建議您將分佈的 Custom SSL Client Support (自訂 SSL 用戶端支援) 設定更新為 Clients that Support Server Name Indication (SNI) (支援伺服器名稱指示 (SNI) 的用戶端) (在 API 中則是將 `SSLSupportMethod` 設為 `sni-only`)。這可讓您使用任何可用的 TLS 安全性原則，也可以降低您的 CloudFront費用。
  - 如果您必須保留具有專用 IP 地址的舊式用戶端支援，則可以在 AWS 支援中心建立案例，請求其他 TLS 安全原則之一 (TLSv1.2_2021、TLSv1.2_2019、TLSv1.2_2018、TLSv1.1_2016 或 TLSv1_2016)。

  > **Note**
  >
  > 在您聯絡 Sup AWS port 部門要求此變更之前，請考慮下列事項：

- 當您將下列其中一個安全性原則 (TLSv1.2_2021、TLSv1.2_2019、TLSv1.2_2018、TLSv1.1_2016 或 TLSv1_2016) 新增至舊版用戶端 Support 發行版時，安全性原則會套用至您帳戶中所有舊版用戶端 Support 散發的所有非 SNI 檢視器要求。 AWS 不過，當檢視器將 SNI 請求傳送至具有舊式用戶端支援的分佈時，會套用該分佈的安全政策。若要確保您想要的安全性原則適用於傳送至您 AWS 帳戶中所有舊版 Client Support 發行版的所有檢視者要求，請將所需的安全性原則個別新增至每個發行版本。
- 根據定義，新的安全政策不支援舊版的相同密碼和通訊協定。例如，如果您選擇將分佈的安全政策從 TLSv1 升級到 TLSv1.1_2016，則該分佈將不再支援 DES-CBC3-SHA 加密。如需每個安全政策所支援之密碼和通訊協定的詳細資訊，請參閱檢視器與之間支援的通訊協定和密碼 CloudFront。

支援的 HTTP 版本

選擇您希望您的發行版本在檢視者通訊時支援的 HTTP 版本 CloudFront。

若要讓檢視者和 CloudFront 使用 HTTP/2，檢視者必須支援 TLSv1.2 或更新版本，以及伺服器名稱指示 (SNI)。 CloudFront 不會透過 HTTP/2 提供 gRPC 的原生支援。

對於觀看者和使 CloudFront 用 HTTP/2，觀看者必須支持 TLSv1.3 和服務器名稱指示（SNI）。CloudFront 支持 HTTP/3 連接遷移，以允許查看者在不丟失連接的情況下切換網絡。如需連線遷移的詳細資訊，請參閱在 RFC 9000 的連線遷移。

> **Note**
>
> 如需支援 TLSv1.3 密碼的詳細資訊，請參閱 檢視器與之間支援的通訊協定和密碼 CloudFront。

預設根物件

選用。當查看者請求分發（ ）的根 URL 而不是分發（`index.html`）中的對象時，要 CloudFront 從源（例如`https://www.example.com/`）請求的對象（例如`https://www.example.com/product-description.html`）。指定預設根物件可避免暴露分佈的內容。

該名稱的長度上限為 255 個字元。該名稱可以包含以下任何字元：

- A-Z、a-z

- 0-9
- _ - . * $ / ~ " '
- & 通過並以 & 傳回

當您指定預設根物件時，只需輸入物件名稱，例如，index.html。不要在物件名稱前新增 /。

如需詳細資訊，請參閱 指定預設根物件。

日誌

是否 CloudFront 要記錄物件每個請求的相關資訊，並將日誌檔存放在 Amazon S3 儲存貯體中。您可以隨時啟用或停用記錄。如果您啟用記錄功能，則不會收取額外費用，但您可以累計在一般 Amazon S3 費用來儲存和存取 Amazon S3 儲存貯體中的檔案。您隨時都可刪除日誌。如需 CloudFront存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

日誌儲存貯體

如果您選擇開啟以進行記錄，則您 CloudFront 要存放存取日誌的 Amazon S3 儲存貯體，例如*myLogs-DOC-EXAMPLE-BUCKET*.s3.amazonaws.com。

> **Important**
>
> 請勿在下列任何區域中選擇 Amazon S3 儲存貯體，因為 CloudFront 不會將標準日誌傳遞到這些區域的儲存貯體：
>
> - 非洲 (開普敦)
> - 亞太區域 (香港)
> - 亞太區域 (海德拉巴)
> - 亞太區域 (雅加達)
> - 亞太區域 (墨爾本)
> - 加拿大西部 (卡加利)
> - 歐洲 (米蘭)
> - 歐洲 (西班牙)
> - 歐洲 (蘇黎世)
> - 以色列 (特拉維夫)
> - Middle East (Bahrain)

- 中東 (阿拉伯聯合大公國)

如果啟用記錄功能，請 CloudFront 記錄物件的每個使用者請求的相關資訊，並將檔案存放在指定的 Amazon S3 儲存貯體中。您可以隨時啟用或停用記錄。如需 CloudFront存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

> **Note**
>
> 您必須擁有取得和更新 Amazon S3 儲存貯體 ACL 所需的許可，並且該儲存貯體的 S3 ACL 必須授予您 FULL_CONTROL。這允許授 CloudFront 予awslogsdelivery帳戶在存儲桶中保存日誌文件的權限。如需詳細資訊，請參閱 設定標準記錄和存取日誌檔案所需的許可。

日誌字首

選用。如果您選擇 [開啟記錄]，請指定要CloudFront 在此發行版本之存取日誌檔名稱前置詞的字串 (如果有的話)，例如exampleprefix/。結尾的斜線) (/) 是可選的，但建議簡化瀏覽日誌檔案。如需 CloudFront 存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

Cookie 記錄

如果您想 CloudFront 要在存取記錄中包含 Cookie，請選擇 [開啟]。如果您選擇在日誌中包含 cookie，無論您如何配置此分發的緩存行為，都會 CloudFront記錄所有 cookie：轉發所有 cookie，不轉發 cookie 或將指定的 cookie 列表轉發到來源。

Amazon S3 不處理 Cookie，因此除非您的分佈還包含 Amazon EC2 或其他自訂原始伺服器原始伺服器，我們建議您選擇關閉做為 Cookie 記錄的值。

如需 Cookie 的詳細資訊，請前往根據 Cookie 快取內容。

啟用 IPv6

IPv6 是 IP 通訊協定的新版本。這是 IPv4 的最終替代品，並使用更大的地址空間。 CloudFront 始終響應 IPv4 請求。如果您想要回應 CloudFront 來自 IPv4 IP 位址 (例如：192.0.2.44) 的要求，以及來自 IPv6 位址的要求 (例如：0 資料庫 8:85:8a2e: 0370:7334)，請選取「啟用 IPv6」。

一般而言，如果在 IPv6 網路上有需要存取您的內容的使用者，則應該啟用 IPv6。不過，如果您使用簽章的 URL 或簽章的 Cookie 來限制對內容的存取，並且如果您使用自訂政策，包含 IpAddress 參數

以限制存取您的內容的 IP 地址，請不要啟用 IPv6。如果您想要透過 IP 位址限制存取一些內容，並且不限制對其他內容的存取 (或限制存取但不透過 IP 地址)，則可以建立兩個分佈。如需有關使用自訂政策建立簽章的 URL 的詳細資訊，請參閱使用自訂政策建立已簽署 URL。如需有關使用自訂政策建立簽章的 Cookie 的詳細資訊，請參閱使用自訂政策設定已簽署 Cookie。

如果您使用 Route 53 別名資源記錄集來將流量路由到您的CloudFront 發佈，則需要在下列兩個條件都成立時建立第二個別名資源記錄集：

- 您為分佈啟用 IPv6
- 您在物件的 URL 中使用備用網域名稱

如需詳細資訊，請參閱 Amazon Route 53 開發人員指南中的使用您的網域名稱將流量路由到 Amazon CloudFront 分發。

如果您透過 Route 53 或其他 DNS 服務建立了 CNAME 資源紀錄集，則不需要進行任何變更。無論檢視器請求的 IP 地址格式如何，CNAME 記錄都會把流量路由到您的分佈。

如果您啟用 IPv6 和 CloudFront 存取記錄檔，資`c-ip`料行會包含 IPv4 和 IPv6 格式的值。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌)。

> **Note**
>
> 為了保持高客戶可用性，如果我們的數據表明 IPv4 將提供更好的用戶體驗，請使用 IPv4 來 CloudFront 響應查看者請求。若要瞭解透過 IPv6 提供的要求 CloudFront 百分比，請啟用散佈的 CloudFront 記錄功能，並剖析`c-ip`資料行，其中包含提出要求的檢視器 IP 位址。這個百分比應該隨著時間的推移而增長，但它仍然是少數的流量，因為 IPv6 還沒有得到全球所有檢視器網路的支援。有些檢視器網路擁有優異的 IPv6 支援，但其他檢視器網路則完全不支援 IPv6。(檢視器網路類似於家用網路或無線電信業者)。
> 如需有關 IPv6 支援的詳細資訊，請參閱CloudFront 常見問題集。如需有關啟用存取日誌的詳細資訊，請參閱日誌、日誌儲存貯體和日誌字首欄位。

註解

選用。當您建立分佈，您可以包含高達 128 個字元的評論。您隨時都可以更新評論。

分佈狀態

表示是否要在部署後啟用或停用該分佈：

- 啟用表示一旦分佈完全部署後，您就可以部署使用分佈的網域名稱的連結，使用者可以擷取內容。每當啟用發佈時，都會 CloudFront 接受並處理任何使用與該發佈相關聯之網域名稱之內容的使用者要求。

  當您建立、修改或刪除 CloudFront 發行版時，變更會傳播至 CloudFront 資料庫需要一些時間。對分佈資訊的立即請求可能不會顯示其變更。傳輸通常可在幾分鐘內完成，但此時可能會增加高系統負載或網路分區。
- 停用表示即使分佈可能已部署並可供使用，但使用者無法使用它。每當停用散佈時，都 CloudFront 不會接受任何使用與該散發相關聯之網域名稱的使用者要求。在將分佈從停用狀態切換到啟用狀態之前 (透過更新分佈的組態)，沒有人可以使用它。

您可以根據需要隨時在停用和啟用之間切換分佈。依照更新分佈組態處理。如需詳細資訊，請參閱 更新分佈。

自訂錯誤頁面和錯誤快取

當您 CloudFront 的 Amazon S3 或自訂來源傳回 HTTP 4xx 或 5xx 狀態碼時，您可以將物件傳回給檢視器 (例如 HTML 檔案)。 CloudFront您也可以指定來自來源的錯誤回應或自訂錯誤頁面在 CloudFront Edge 快取中快取的時間長度。如需詳細資訊，請參閱 針對特定的 HTTP 狀態碼建立自訂錯誤頁面。

> **Note**
>
> 在建立分佈精靈中不包含以下值，因此只在您更新分佈時才能設定自訂錯誤頁面。

主題

- HTTP 錯誤代碼
- 回應頁面路徑
- HTTP 回應代碼
- 錯誤快取最短 TTL (秒)

HTTP 錯誤代碼

您要傳回自訂錯誤頁面 CloudFront 的 HTTP 狀態碼。您可以設定 CloudFront 為針對無、部分或所有 CloudFront 快取的 HTTP 狀態碼傳回自訂錯誤頁面。

回應頁面路徑

當您的原始伺服器傳回您為「錯誤碼」指定的 HTTP 狀態碼 (例如 403/4xx-errors/403-forbidden.html) 時，您想要 CloudFront傳回檢視器的自訂錯誤頁面路徑 (例如，)。如果您要在不同位置存放物件和自訂錯誤頁面，則您的分佈必須包含下列為屬實的快取行為：

- Path Pattern (路徑模式) 的值與自訂錯誤訊息的路徑相符。例如，假設您已在 Amazon S3 儲存貯體名為 /4xx-errors 的目錄中儲存了 4xx 錯誤的自訂錯誤頁面。您的分佈必須包含快取行為，其路徑模式會將自訂錯誤頁面的請求路由至該位置，例如 /4xx-errors/* (/4xx-errors/*)。
- Origin (原始伺服器) 的數值將 Origin ID (原始伺服器 ID) 的數值指定給包含自訂錯誤頁面的原始伺服器。

HTTP 回應代碼

您要 CloudFront 與自訂錯誤頁面一起傳回檢視器的 HTTP 狀態碼。

錯誤快取最短 TTL (秒)

您想要從原始伺服器快 CloudFront 取錯誤回應的最短時間量。

地理限制

如果您需要防止所選國家/地區的使用者存取您的內容，您可以使用允許清單或封鎖清單來設定您的 CloudFront 散佈。設定地理限制無需額外收費。如需詳細資訊，請參閱 限制您內容的地理分佈。

測試發行版

建立發行版之後， CloudFront 知道原始伺服器在哪裡，並且您知道與發行版相關聯的網域名稱。您可以使用網 CloudFront 域名稱建立物件的連結，並 CloudFront 將物件提供至您的網頁或應用程式。

> **Note**
>
> 您必須等到分佈的狀態變更為 Deployed (已部署)，然後才能測試連結。

在 Web 分佈中建立連結到物件

1. 將以下 HTML 程式碼複製到新的檔案中，以您的分佈的網域名稱取代 *domain-name*，然後以您的物件的名稱取代 *object-name*。

```
<html>
```

```
<head>My CloudFront Test</head>
<body>
<p>My text content goes here.</p>
<p><img src="https://domain-name/object-name" alt="my test image"
</body>
</html>
```

例如，如果您的網域名稱是 d111111abcdef8.cloudfront.net，物件是 image.jpg，則連結的 URL 將為：

https://d111111abcdef8.cloudfront.net/image.jpg.

如果物件位於原始伺服器的資料夾中，則該資料夾必須包含在 URL 中。例如，如果 image.jpg 位於原始伺服器上的影像資料夾中，則 URL 將為：

https://d111111abcdef8.cloudfront.net/images/image.jpg

2. 將 HTML 程式碼儲存在副檔名為 .html 的檔案中。
3. 在瀏覽器中開啟您的網頁，以確保您可以查看物件。

瀏覽器會傳回包含內嵌影像檔案的頁面，該檔案會從確 CloudFront 定適合提供物件的邊緣位置提供。

更新分佈

在 CloudFront 控制台中，您可以看到與您的 AWS 帳戶相關聯的 CloudFront 分佈，查看分發的設置，並更新大多數設置。請注意，在分佈傳播到 AWS 節點之前，您所做的更新不會生效。

若要更新發 CloudFront 佈

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選取分佈的 ID。該列表包括與您用來登錄 CloudFront 控制台的 AWS 帳戶相關聯的所有分發。
3. 若要編輯分佈的設定，請選擇 Distribution Settings (分佈設定) 索引標籤。
4. 若要更新一般設定，請選擇 Edit (編輯)。否則，請選擇您要更新之設定的索引標籤：Origins (原始伺服器) 或 Behaviors (行為)。
5. 進行更新，接著若要儲存變更，請選擇 Yes, Edit (是，編輯)。如需欄位的詳細資訊，請參閱下列主題：

   - 一般設定：分佈設定

- 原始伺服器設定：原始設定
- 快取行為設定：快取行為設定

6. 如果您想要在分佈中刪除原始伺服器，請執行下列動作：

   a. 選擇 Behaviors (行為)，然後確認您已將任何與原始伺服器相關的預設快取行為移動至另一個原始伺服器。

   b. 選擇 Origins (原始伺服器)，然後選取一個原始伺服器。

   c. 選擇 Delete (刪除)。

您也可以使用 CloudFront API 更新發行版：

- 若要更新分發，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

> **Important**
>
> 當您更新您的分佈時，請注意有多個額外的必要欄位，這些欄位在建立分佈時是非必要的。為了協助確保在您使用 CloudFront API 更新分發時包含所有必要欄位，請遵循 Amazon CloudFront API 參考UpdateDistribution中所述的步驟。

當您儲存對發佈組態的變更時，會 CloudFront 開始將變更傳播到所有邊位置。連續的組態變更依照其各自的順序傳播。在節點更新您的組態之前，會 CloudFront繼續根據先前的組態從該位置提供您的內容。在節點中更新組態後， CloudFront立即開始根據新組態從該位置提供內容。

您的變更不會立即傳播到每個節點。傳輸完成時，發佈的狀態會從變更InProgress為「已部署」。CloudFront 在傳播您的變更時，我們無法判斷指定的節點是根據先前的組態還是新設定來提供您的內容。

測試分佈

標籤是您可以用來識別和組織 AWS 資源的字詞或片語。可以新增多個標籤到每個資源，且每個標籤皆包含您所定義的金鑰和值。例如，金鑰可能是「網域」，而值可能是「example.com」。可以根據新增的標籤來搜尋與篩選資源。

您可以搭配使用標籤 CloudFront，例如下列範例：

- 在 CloudFront 分發上強制執行基於標籤的權限 如需詳細資訊，請參閱 阿巴克與 CloudFront。

- 追蹤不同類別的帳單資訊。當您將標籤套用至 CloudFront分發或其他 AWS 資源 (例如 Amazon EC2 執行個體或 Amazon S3 儲存貯體) 並啟用標籤時， AWS 會產生成本分配報告，並以逗號分隔值 (CSV 檔案) 形式產生成本分配報告，其中包含您的使用量和成本 (由作用中標籤彙總)。

  您可以套用代表業務類別 (例如成本中心、應用程式名稱或擁有者) 的標籤，來整理多個服務中的成本。如需有關使用成本分配標籤的詳細資訊，請參閱 AWS Billing 使用者指南中的使用成本分配標籤。

> **備註**
>
> - 您可以標記分佈，但無法標記原始存取身分或失效。
> - 目前不支援標籤編輯器和資源群組CloudFront。
> - 如需了解目前可新增到分發的標籤數量上限的詳細資訊，請參閱一般配額。

內容

標籤限制

以下基本限制適用於標籤：

- 如需每個分發的標籤數目上限，請參閱一般配額。
- 金鑰長度上限 - 128 個 Unicode 字元
- 數值長度上限 - 256 個 Unicode 字元
- 金鑰與值的有效值 – a-z、A-Z、0-9、空格和下列字元：_ . : / = + - 及 @
- 標記金鑰與值皆區分大小寫
- 請不要使用 `aws:` 做為索引鍵的字首。此字首已保留供 AWS 使用。

新增、編輯和刪除分發的標籤

您可以使用 CloudFront 控制台管理發行版的標籤。

若要為分佈新增標籤、編輯或刪除標籤

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您希望更新的分佈 ID。
3. 選擇 Tags (標籤) 索引標籤。
4. 選擇管理標籤。
5. 在 Manage tags (管理標籤) 頁面上，可以執行下列操作：
   - 若要加入標籤，請輸入金鑰，並選擇性地輸入標籤值。選擇「新增標籤」以新增更多標籤。
   - 若要編輯標籤，請變更標籤的金鑰或其值，或兩者均變更。可以刪除標籤的值，但仍須金鑰。
   - 若要刪除標籤，請選擇 Remove (移除)。
6. 選擇儲存變更。

相關資訊

您也可以使用 CloudFront API、 AWS Command Line Interface (AWS CLI)、 AWS SDK，以及套 AWS Tools for Windows PowerShell 用標籤。如需詳細資訊，請參閱下列主題：

- CloudFront API 作業：
  - ListTagsForResource
  - TagResource
  - UntagResource
- AWS CLI — 請參閱AWS CLI 命令參考中的雲端
- AWS SDK — 請參閱文件頁面上的適用 SDK AWS 文件
- 適用於 Windows 的工具 PowerShell — 請參閱AWS Tools for PowerShell 指令程 CloudFront式參考中的 Amazon

刪除 分發

下列程序會使用 CloudFront 主控台刪除散佈。如需使用 CloudFront API 刪除的相關資訊，請參閱 Amazon CloudFront API 參考DeleteDistribution中的。

如果您需要刪除 OAC 連接到 S3 儲存貯體的發佈，請參閱以取得重要刪除附加到 S3 儲存貯體的 OAC 的發佈詳細資訊。

> **Note**
>
> 請注意，您必須先停用分佈，然後才能刪除它，而這需要更新分佈的許可。
> 如果您 CloudFront 停用了與其相關聯的替代網域名稱的發行版，即使另一個發行版具有萬用字元 (*) 符合相同網域 (例如 *.example.com) 的替代網域名稱，也會停止接受該網域名稱的流量 (例如 *.example.com)。

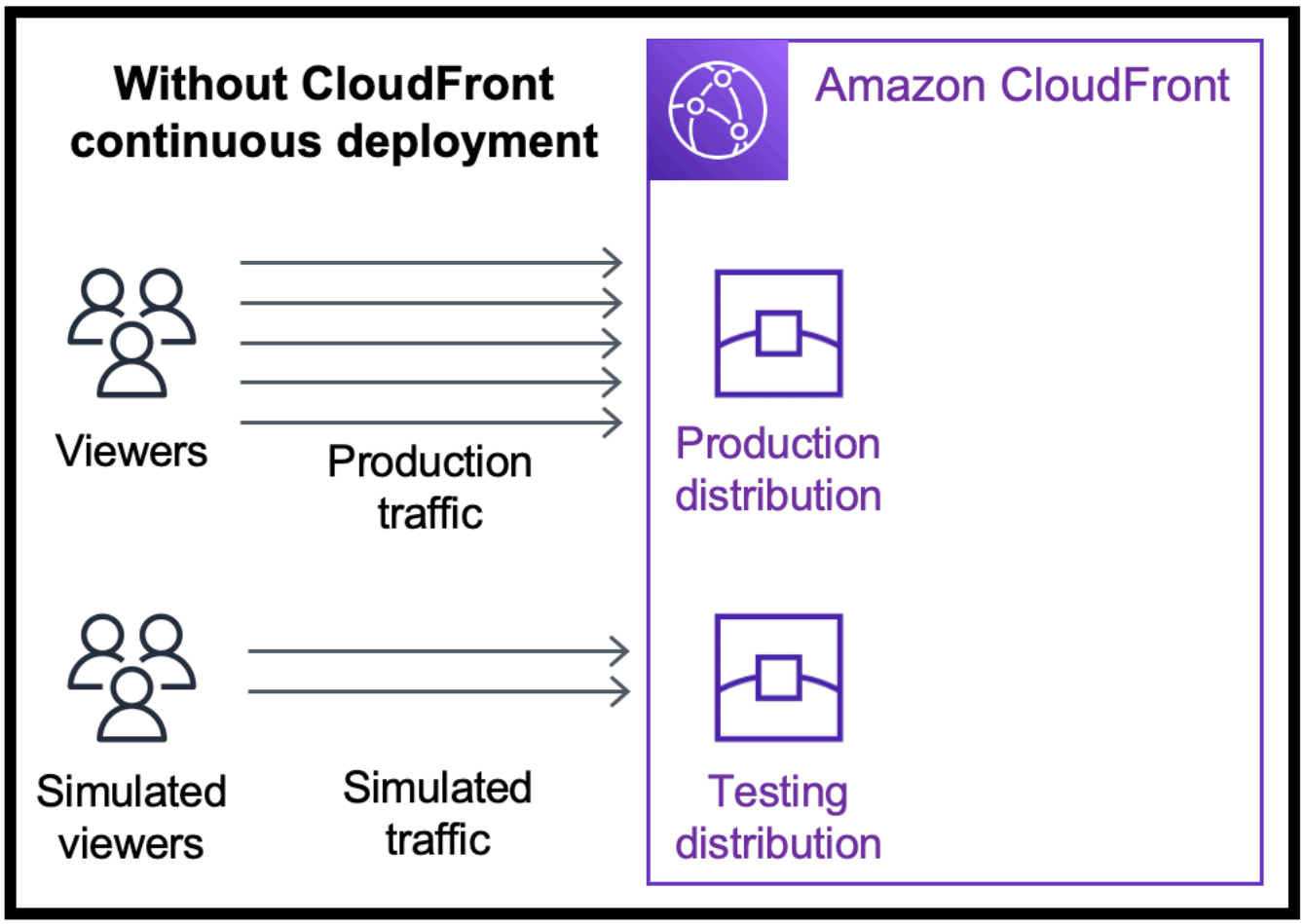
若要刪除分 CloudFront 配

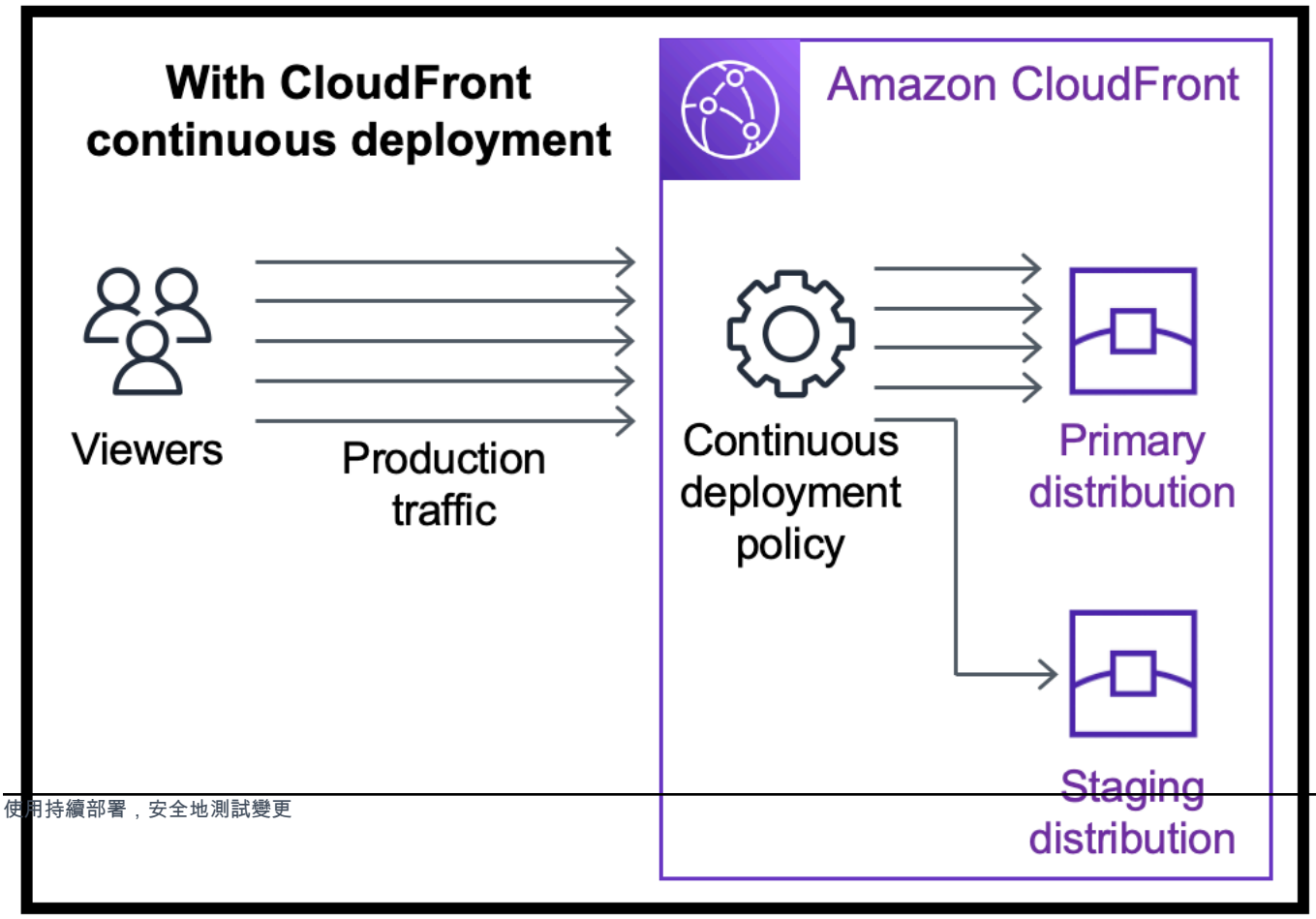
1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在 CloudFront 控制台的右窗格中，找到要刪除的發行版。
   - 如果「狀態」欄顯示「已停用」，請跳至步驟 6。
   - 如果「狀態」顯示為「已啟用」，但發行版本仍在「上次修改」欄中顯示「部署」，請等待部署完成，然後再繼續步驟 3。
3. 在 CloudFront 主控台的右窗格中，選取要刪除之發佈的核取方塊。
4. 選擇 Disable (停用) 來停用分佈，然後選擇 Yes, Disable (是，停用) 以進行確認。然後，選擇 Close (關閉)。
   - 「狀態」欄的值會立即變更為「已停用」。
5. 等到新的時間戳記出現在「上次修改」欄下。
   - 將變更傳播 CloudFront 到所有邊緣位置可能需要幾分鐘的時間。
6. 選取您要刪除之分佈的核取方塊。
7. 選擇 刪除，刪除。
   - 如果「刪除」(Delete) 選項不可用， CloudFront 則表示仍將您的變更傳播到邊位置。等到新的時間戳記出現在「上次修改」欄下，然後重複步驟 6-7。

使用 CloudFront 持續部署安全地測試 CDN 組態變更

透過 Amazon CloudFront 持續部署，您可以先使用一部分生產流量進行測試，以安全地將變更部署到 CDN 組態。您可以使用臨時分佈和持續部署政策，將真實 (生產) 檢視者的部份流量傳送至新的 CDN 組態，並驗證其是否如預期般運作。您可以即時監控新組態的效能，並在準備就緒時提升新組態，以便透過主要分佈為所有流量提供服務。

下圖顯示使用 CloudFront 持續部署的好處。若沒有此部署，您就必須使用模擬流量測試 CDN 組態變更。透過持續部署，您可以使用生產流量子集來測試變更，然後在準備就緒時將變更提升至主要分佈。


Without CloudFront continuous deployment
Amazon CloudFront
Viewers
Production traffic
Production distribution
Simulated viewers
Simulated traffic
Testing distribution


With CloudFront continuous deployment
Amazon CloudFront
Viewers
Production traffic
Continuous deployment policy
Primary distribution
Staging distribution

主題

使用 CloudFront 持續部署的工作流程

下列高階工作流程說明如何透過 CloudFront 持續部署安全地測試和部署組態變更。

1. 選擇您要做為主要分佈的分佈。主要分佈是目前為生產流量提供服務的分佈。
2. 從主要分佈中，建立臨時分佈。臨時分佈一開始會是主要分佈的副本。
3. 在持續部署政策內建立流量組態，並將其連接至主要分佈。這會決定如何將流量 CloudFront 路由至預備分發。如需詳細了解臨時分佈的路由請求，請參閱 the section called “將請求路由至臨時分佈”。
4. 更新臨時分佈的組態。如需詳細了解您可以更新的設定，請參閱 the section called “更新主要分佈或臨時分佈”。
5. 監控臨時分佈，以判斷組態變更是否如預期般運作。如需監控臨時分佈的詳細資訊，請參閱 the section called “監控臨時分佈”。

   當您監控臨時分佈時，您可以：

   - 再次更新臨時分佈的組態，以持續測試組態變更。
   - 更新持續部署政策 (流量組態)，以傳送更多或更少流量至臨時分佈。
6. 臨時分佈的效能符合您的需求時，請將臨時分佈的組態提升為主要分佈，如此會將臨時分佈的組態複製到主要分佈。這也會停用連續部署原則，這表示將所有流量 CloudFront 路由傳送至主要散發。

您可以建置自動化來監控臨時分佈的效能 (步驟 5)，並在符合特定條件時自動提升組態 (步驟 6)。

提升組態之後，您可以在下次測試組態變更時，重複使用相同的臨時分佈。

如需有關在 CloudFront 主控台、或 CloudFront API 中使用預備發行版和持續部署原則的 AWS CLI詳細資訊，請參閱下一節。

使用臨時分佈和持續部署政策

您可以使用 AWS Command Line Interface (AWS CLI) 或使用 CloudFront API 在 CloudFront 主控台中建立、更新和修改預備分發和持續部署政策。

Console

若要使用暫存發行版和連續部署原則 AWS Management Console，請使用下列程序。

建立臨時分佈和持續部署政策 (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 Distributions (分佈)。
3. 選擇您要做為主要分佈的分佈。主要分佈目前為生產流量提供服務，也就是您從中建立臨時分佈的地方。
4. 在 Continuous deployment (持續部署) 區段中，選擇 Create staging distribution (建立臨時分佈)。這項操作會開啟 Create staging distribution (建立臨時分佈) 精靈。
5. 在 Create staging distribution (建立臨時分佈) 精靈中，執行下列動作：

   a. (選用) 輸入臨時分佈的描述。

   b. 選擇下一步。

   c. 修改臨時分佈的組態。如需詳細了解您可以更新的設定，請參閱 the section called “更新主要分佈或臨時分佈”。

      修改完臨時分佈的組態後，請選擇 Next (下一步)。

   d. 使用控制台指定 Traffic configuration (流量組態)。這會決定如何將流量 CloudFront 路由至預備分發。(將流量組態CloudFront 儲存在連續部署原則中。)

      如需 Traffic configuration (流量組態) 選項的詳細資訊，請參閱 the section called “將請求路由至臨時分佈”。

      完成 Traffic configuration (流量組態) 時，請選擇 Next (下一步)。

   e. 檢閱包含流量組態的臨時分佈組態，然後選擇 Create staging distribution (建立臨時分佈)。

在 CloudFront 主控台中完成 [建立暫存分配] 精靈後，請 CloudFront 執行下列動作：

- 使用您在步驟 5c 中指定的設定，建立臨時分佈
- 使用您在步驟 5d 中指定的流量組態，建立持續部署政策
- 將持續部署政策連接至您從中建立臨時分佈的主要分佈

當主要發行版的組態 (使用連結的連續部署原則) 部署至 Edge 位置時，會 CloudFront 開始根據流量組態將指定部分的流量傳送至暫存分發。

更新臨時分佈 (主控台)

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 Distributions (分佈)。
3. 選擇主要分佈。這是目前為生產流量提供服務的分佈，也就是您從中建立臨時分佈的地方。
4. 選擇 View staging distribution (檢視臨時分佈)。
5. 使用主控台修改臨時分佈的組態。如需詳細了解您可以更新的設定，請參閱 the section called "更新主要分佈或臨時分佈"。

一旦臨時分佈的組態部署到邊緣節點，即對路由至臨時分佈的傳入流量產生效果。

更新持續部署政策 (主控台)

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 Distributions (分佈)。
3. 選擇主要分佈。這是目前為生產流量提供服務的分佈，也就是您從中建立臨時分佈的地方。
4. 在 Continuous deployment (持續部署) 區段中，選擇 Edit policy (編輯政策)。
5. 修改持續部署政策中的流量組態。完成時，請選擇 Save changes (儲存變更)。

當具有更新連續部署原則的主要發行版組態部署到邊緣位置時，會根據更新的流量組態，CloudFront 開始將流量傳送至預備分發。

提升臨時分佈組態 (主控台)

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 Distributions (分佈)。
3. 選擇主要分佈。這是目前為生產流量提供服務的分佈，也就是您從中建立臨時分佈的地方。

4. 在 Continuous deployment (持續部署) 區段中，選擇 Promote (提升)。
5. 輸入 **`confirm`**，然後選擇 Promote (提升)。

升級暫存發佈時，會將組態從暫存發佈 CloudFront 複製到主要發行版。 CloudFront 也會停用連續部署原則，並將所有流量路由傳送至主要散發。

提升組態之後，您可以在下次測試組態變更時，重複使用相同的臨時分佈。

CLI

若要使用暫存發行版和連續部署原則 AWS CLI，請使用下列程序。

建立臨時分佈 (CLI)

1. 同時使用 aws cloudfront get-distribution 和 grep 命令，針對您要做為主要分佈的分佈，取得其 ETag 值。主要分佈目前為生產流量提供服務，也就是您從中建立臨時分佈的地方。

   以下為命令的範例。在下列範例中，請將 *primary_distribution_ID* 取代為主要分佈的 ID。

   ```
   aws cloudfront get-distribution --id primary_distribution_ID | grep 'ETag'
   ```

   請複製 ETag 值，因為您會在以下步驟使用該值。
2. 使用 aws cloudfront copy-distribution 命令建立臨時分佈。下列範例命令使用逸出字元 (\) 和換行符號以提高可讀性，但您應在命令中省略這些字元。以下是範例命令：

   - 將 *primary_distribution_ID* 取代為主要分佈的 ID。
   - 將 *primary_distribution_ETag* 取代為主要分佈的 ETag 值，也就是您在上一個步驟中取得的值。
   - (可選) 將 *CLI_example* 取代為所需的呼叫者參考 ID。

   ```
   aws cloudfront copy-distribution --primary-distribution-
   id primary_distribution_ID \
                                    --if-match primary_distribution_ETag \
                                    --staging \
                                    --caller-reference 'CLI_example'
   ```

命令的輸出會顯示有關臨時分佈及其組態的資訊。複製預備分發的 CloudFront 網域名稱，因為您需要執行下列步驟。

建立持續部署政策 (包含輸入檔案的 CLI)

1. 使用下列命令建立一個名為 `continuous-deployment-policy.yaml` 的檔案，其中包含 create-continuous-deployment-policy 命令的所有輸入參數。下列命令使用逸出字元 (\) 和換行符號以提高可讀性，但您應在命令中省略這些字元。

```
aws cloudfront create-continuous-deployment-policy --generate-cli-skeleton yaml-input \
                                                    > continuous-deployment-policy.yaml
```

2. 開啟您剛才建立且命名為 `continuous-deployment-policy.yaml` 的檔案。編輯檔案以指定您想要的持續部署政策設定，然後儲存檔案。當您編輯檔案時：

   - `StagingDistributionDnsNames` 區段：
     - 將 `Quantity` 的值變更為 `1`。
     - 對於`Items`，貼上預備分發 (您從上一個步驟儲存的) 的 CloudFront 網域名稱。
   - `TrafficConfig` 區段：
     - 選擇 `Type`，可為 `SingleWeight` 或 `SingleHeader`。
     - 移除其他類型的設定。例如，若您要以權重為基礎的流量組態，請將 `Type` 設定為 `SingleWeight`，然後移除 `SingleHeaderConfig` 設定。
     - 若要使用以權重為基礎的流量組態，請將 `Weight` 值設定為介於 `.01` (百分之一) 到 `.15` (百分之十五) 間的十進位數字。

   如需 `TrafficConfig` 選項的詳細資訊，請參閱 the section called “將請求路由至臨時分佈” 和 the section called “以權重為基礎之組態的工作階段黏性”。

3. 使用下列命令，使用 `continuous-deployment-policy.yaml` 檔案中的輸入參數建立持續部署政策。

```
aws cloudfront create-continuous-deployment-policy --cli-input-yaml file://continuous-deployment-policy.yaml
```

複製命令輸出中的 Id 值。這是持續部署政策 ID，您會在接下來的步驟使用。

將持續部署政策連接至主要分佈 (包含輸入檔案的 CLI)

1. 使用下列命令，將主要分佈的組態儲存至名為 primary-distribution.yaml 的檔案。將 *primary_distribution_ID* 取代為主要分佈的 ID。

```
aws cloudfront get-distribution-config --id primary_distribution_ID --output
 yaml > primary-distribution.yaml
```

2. 開啟您剛才建立且命名為 primary-distribution.yaml 的檔案。編輯檔案，進行下列變更：

   - 將您從上一個步驟複製的持續部署政策 ID 貼到 ContinuousDeploymentPolicyId 欄位。
   - 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

   完成後儲存檔案。

3. 使用下列命令來更新主要分佈，以使用持續部署政策。將 *primary_distribution_ID* 取代為主要分佈的 ID。

```
aws cloudfront update-distribution --id primary_distribution_ID --cli-input-yaml
 file://primary-distribution.yaml
```

當主要發行版的組態 (使用連結的連續部署原則) 部署至 Edge 位置時，會 CloudFront 開始根據流量組態將指定部分的流量傳送至暫存分發。

更新臨時分佈 (包含輸入檔案的 CLI)

1. 使用下列命令，將臨時分佈的組態儲存至名為 staging-distribution.yaml 的檔案。將 *staging_distribution_ID* 取代為暫存分佈的 ID。

```
aws cloudfront get-distribution-config --id staging_distribution_ID --output
 yaml > staging-distribution.yaml
```

2. 開啟您剛才建立且命名為 staging-distribution.yaml 的檔案。編輯檔案，進行下列變更：

   - 修改臨時分佈的組態。如需詳細了解您可以更新的設定，請參閱 the section called “更新主要分佈或臨時分佈”。
   - 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

   完成後儲存檔案。

3. 使用下列命令來更新臨時分佈的組態。將 *staging_distribution_ID* 取代為暫存分佈的 ID。

```
aws cloudfront update-distribution --id staging_distribution_ID --cli-input-yaml
 file://staging-distribution.yaml
```

一旦臨時分佈的組態部署到邊緣節點，即對路由至臨時分佈的傳入流量產生效果。

更新持續部署政策 (包含輸入檔案的 CLI)

1. 使用下列命令，將持續部署政策的組態儲存至名為 continuous-deployment-policy.yaml 的檔案。將 *continuous_deployment_policy_ID* 取代為持續部署政策的 ID。下列命令使用逸出字元 (\) 和換行符號以提高可讀性，但您應在命令中省略這些字元。

```
aws cloudfront get-continuous-deployment-policy-config --
id continuous_deployment_policy_ID \
                                                    --output yaml >
 continuous-deployment-policy.yaml
```

2. 開啟您剛才建立且命名為 continuous-deployment-policy.yaml 的檔案。編輯檔案，進行下列變更：

   - 視需要修改持續部署政策的組態。例如，您可以從以標頭為基礎的流量組態，改為使用以權重為基礎的流量組態，或變更以權重為基礎的組態的流量百分比 (權重)。如需詳細資訊，請參閱 the section called “將請求路由至臨時分佈” 及 the section called “以權重為基礎之組態的工作階段黏性”。
   - 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

完成後儲存檔案。

3. 使用下列命令來更新持續部署政策。將 *continuous_deployment_policy_ID* 取代為持續部署政策的 ID。下列命令使用逸出字元 (\) 和換行符號以提高可讀性，但您應在命令中省略這些字元。

```
aws cloudfront update-continuous-deployment-policy --
id continuous_deployment_policy_ID \
                                                    --cli-input-yaml file://
continuous-deployment-policy.yaml
```

當具有更新連續部署原則的主要發行版組態部署到邊緣位置時，會根據更新的流量組態，CloudFront 開始將流量傳送至預備分發。

提升臨時分佈組態 (CLI)

- 使用 aws cloudfront update-distribution-with-staging-config 命令，將臨時分佈的組態提升為主要分佈。下列範例命令使用逸出字元 (\) 和換行符號以提高可讀性，但您應在命令中省略這些字元。以下是範例命令：
  - 將 *primary_distribution_ID* 取代為主要分佈的 ID。
  - 將 *staging_distribution_ID* 取代為臨時分佈的 ID。
  - 將 *primary_distribution_ETag* 和 *staging_distribution_ETag* 取代為主要分佈和臨時分佈的 ETag 值。請確認主要分佈的值是第一個，如範例所示。

```
aws cloudfront update-distribution-with-staging-config --
id primary_distribution_ID \
                                                    --staging-distribution-
id staging_distribution_ID \
                                                    --if-match
 'primary_distribution_ETag,staging_distribution_ETag'
```

升級暫存發佈時，會將組態從暫存發佈 CloudFront 複製到主要發行版。 CloudFront 也會停用連續部署原則，並將所有流量路由傳送至主要散發。

提升組態之後，您可以在下次測試組態變更時，重複使用相同的臨時分佈。

API

若要使用 CloudFront API 建立預備分發和持續部署原則，請使用下列 API 作業：

- CopyDistribution
- CreateContinuousDeploymentPolicy

如需詳細了解您在這些 API 呼叫中指定的欄位，請參閱以下內容：

- the section called “將請求路由至臨時分佈”
- the section called “以權重為基礎之組態的工作階段黏性”
- 您的 AWS SDK 或其他 API 用戶端的 API 參考文件

建立暫存散發和持續部署原則之後，請使用 UpdateDistribution(在主要發行版上) 將持續部署原則附加至主要發行版。

若要更新暫存發佈的組態，請使用 UpdateDistribution(在暫存發佈上) 來修改暫存發佈的組態。如需詳細了解您可以更新的設定，請參閱 the section called “更新主要分佈或臨時分佈”。

若要將暫存發行版的組態升級為主要發行版，請使用UpdateDistributionWithStagingConfig。

如需有關您在這些 API 呼叫中指定之欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

監控臨時分佈

若要監視預備分發的效能，您可以使用為所有發行版 CloudFront 提供的相同度量、記錄和報告。例如：

- 您可以在 CloudFront主控台中檢視預設 CloudFront分佈量度 (例如總要求和錯誤率)，也可以額外付費開啟其他度量 (例如按狀態碼分類的快取命中率和錯誤率)。您也可以根據這些指標建立警示。
- 您可以檢視標準日誌和即時日誌，深入了解臨時分佈接收到的請求。標準記錄檔包含下列兩個欄位，可協助您識別要求在將要求路由傳送至預備分發之前最初傳 CloudFront 送到的主要散佈：`primary-distribution-id`和`primary-distribution-dns-name`。
- 您可以在 CloudFront主控台中檢視和下載報告，例如快取統計資料報表。

了解持續部署的運作方式

下列主題說明 CloudFront 持續部署的運作方式。

主題

將請求路由至臨時分佈

當您使用 CloudFront 持續部署時，您不需要變更檢視器要求的任何相關內容。檢視者無法使用 DNS 名稱、IP 地址或 CNAME，將請求直接傳送至臨時分佈。相反地，檢視者會將要求傳送至主要 (生產) 發佈，並根據連續部署原則中的流量組態設定，將部分要求 CloudFront 路由至預備分發。有兩種類型的流量組態：

以權重為基礎的

以權重為基礎的組態會將指定百分比的檢視者請求路由至臨時分佈。當您使用權重型組態時，您也可以啟用工作階段黏著性，這有助於確保 CloudFront 將來自相同檢視器的要求視為單一工作階段的一部分。如需詳細資訊，請參閱 the section called “以權重為基礎之組態的工作階段黏性”。

以標頭為基礎的

當檢視者請求包含特定 HTTP 標頭 (您可以指定該標頭和值)，以標頭為基礎的組態會將請求路由至臨時分佈。不包含指定標頭和值的請求則會路由至主要分佈。此組態對於本地測試或您可以控制檢視者請求時非常有用。

> **Note**
>
> 路由至臨時分佈的標頭必須包含字首 `aws-cf-cd-`。

以權重為基礎之組態的工作階段黏性

當您使用權重型組態 CloudFront 將流量路由傳送至暫存分佈時，也可以啟用工作階段黏著性，這有助於確保將來自相同檢視器的要求視為單一工作階段。當您啟用工作階段黏著性時，請 CloudFront 設定 Cookie，以便單一工作階段中來自相同檢視器的所有要求都由一個發行版 (主要或暫存) 提供服務。

當您啟用工作階段黏性，您也可以指定閒置時間。如果檢視器在這段時間內處於閒置狀態 (不傳送任何要求)，工作階段就會過期，並 CloudFront 將 future 自此檢視器的要求視為新工作階段。您能以秒為單位，將閒置時間設定為 300 (5 分鐘) 到 3600 (一小時)。

在下列情況下，請 CloudFront 重設所有工作階段 (即使是作用中的工作階段)，並將所有要求視為新工作階段：

- 您停用或啟用持續部署政策
- 您停用或啟用工作階段黏性設定

更新主要分佈或臨時分佈

當主要分佈具有連接的持續部署政策時，主要分佈和臨時分佈皆可使用下列組態變更：

- 所有快取行為設定，包括預設快取行為
- 所有原始伺服器設定 (原始伺服器和原始伺服器群組)
- 自訂錯誤回應 (錯誤頁面)
- 地理限制
- 預設根物件
- 日誌設定
- 描述 (評論)

您也可以更新發佈組態中參照的外部資源，例如快取政策、回應標頭政策、函數或 Lambda @Edge CloudFront 函數。

主要分佈和臨時分佈不會共用快取

主要分佈和臨時分佈不會共用快取。將第一個要求 CloudFront 傳送至暫存散發時，其快取為空白。當請求抵達臨時分佈時，便會開始快取回應 (如果設定為如此)。

持續部署的配額和其他考量

CloudFront 持續部署需遵守下列配額和其他考量。

配額

- 每個暫存分配的最大數量 AWS 帳戶：20
- 每個連續部署原則的數目上限 AWS 帳戶：20

- 在以權重為基礎的組態中，您可以傳送至臨時分佈的流量百分比上限：15%
- 工作階段黏性閒置時間的上下限：300–3600 秒

如需詳細資訊，請參閱 配額。

> **Note**
>
> 使用持續部署且您的主要分發是使用適用於 S3 儲存貯體存取的 OAC 設定時，請更新 S3 儲存貯體政策以允許存取暫存分配。如需 S3 儲存貯體政策範例，請參閱the section called “授予原始存取控制許可，以存取 S3 儲存貯體”。

AWS WAF 網路 ACL

如果您為分發啟用連續發佈，則適用下列考量 AWS WAF：

- 您無法首次將 AWS WAF Web 存取控制清單 (ACL) 與發行版產生關聯。
- 您無法取消 AWS WAF Web ACL 與分佈的關聯。

您必須先刪除生產發行版的持續部署原則，才能執行前述工作。這也會刪除暫存分配。如需詳細資訊，請參閱 使用 AWS WAF 保護。

CloudFront 將所有請求發送到主要分發的情況

在某些情況下 (例如資源使用率高的期間) CloudFront 可能會將所有要求傳送至主要發行版，而不論連續部署原則中指定的內容為何。

CloudFront 無論連續部署原則中指定的內容為何，都會在尖峰流量時段將所有要求傳送至主要分發。尖峰流量指的是CloudFront 服務上的流量，而不是分佈中的流量。

HTTP/3

您無法搭配使用支援 HTTP/3 的分佈與持續部署。

使用各種來源與 CloudFront分佈

當您建立發佈時，您可以指定 CloudFront 傳送檔案要求的來源。您可以使用幾種不同種類的起源 CloudFront。例如，您可以使用 Amazon S3 儲存貯體、 MediaStore 容器、 MediaPackage 通道、應用程式負載平衡器或 AWS Lambda 函數 URL。

主題

使用 Amazon S3 存儲桶

下列主題說明使用 Amazon S3 儲存貯體做為 CloudFront 分發來源的不同方式。

主題

使用標準的 Amazon S3 存儲桶

當您使用 Amazon S3 做為分發的來源時，您要將 CloudFront 要交付的物件放在 Amazon S3 儲存貯體中。您可以使用 Amazon S3 支援的任何方法將物件放入 Amazon S3，例如 Amazon S3 主控台或 API，也可使用第三方工具。如同使用其他標準 Amazon S3 儲存貯體一樣，您可以在儲存貯體中建立階層結構以存放物件。

使用現有的 Amazon S3 儲存貯體做為 CloudFront 原始伺服器不會以任何方式變更儲存貯體；您仍然可以像平常一樣以標準 Amazon S3 價格存放和存取 Amazon S3 物件一樣使用它。您需要定期支付 Amazon S3 費用才能將物件存放在儲存貯體中。如需要使用之費用的詳細資訊 CloudFront，請參閱 Amazon CloudFront 定價。如需搭配現有 S3 儲存貯體使 CloudFront 用的詳細資訊，請參閱the section called “添加 CloudFront 到現有的 Amazon S3 存儲桶”。

> **Important**
>
> 若要使用值區 CloudFront，名稱必須符合 DNS 命名需求。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的儲存貯體命名規則。

當您指定 Amazon S3 儲存貯體做為原點時 CloudFront，建議您使用下列格式：

bucket-name.s3.*region*.amazonaws.com

當您以此格式指定值區名稱時，可以使用下列CloudFront 功能：

- 設定 CloudFront 為使用 SSL/TLS 與您的 Amazon S3 儲存貯體通訊。如需詳細資訊，請參閱 the section called “搭配使用 HTTPS CloudFront”。
- 使用來源存取控制來要求檢視者使用 CloudFront URL 存取您的內容，而不是使用 Amazon S3 URL。如需詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。
- 通過提交POST並PUT請求來更新值區的內容 CloudFront。如需詳細資訊，請參閱 the section called “如何 CloudFront 處理和轉送請求到您的 Amazon S3 來源” 主題中的 the section called “HTTP 方法”。

請勿使用以下格式指定儲存貯體：

- Amazon S3 路徑樣式：s3.amazonaws.com/*bucket-name*
- Amazon S3 CNAME

使用 Amazon S3 對象 Lambda

在您建立 Object Lambda 存取點時，Amazon S3 會自動為您的 Object Lambda 存取點產生唯一的別名。您可以使用此別名而不是 Amazon S3 儲存貯體名稱做為 CloudFront 分發的來源。

當您使用物件 Lambda 存取點別名作為原點時 CloudFront，建議您使用下列格式：

alias.s3.*region*.amazonaws.com

如需有關尋找 *alias* 的詳細資訊，請參閱《Amazon S3 使用者指南》中的如何為您的 S3 儲存貯體 Object Lambda 存取點使用儲存貯體式別名。

> **Important**
>
> 當您使用物件 Lambda 存取點做為的原點時 CloudFront，您必須使用原始存取控制。

如需範例使用案例，請參閱 CloudFront 將 Amazon S3 物件 Lambda 與 Amazon 搭配使用，為最終使用者量身打造內容。

CloudFront 將物件 Lambda 存取點原點視為標準 Amazon S3 儲存貯體來源相同。

如果您使用 Amazon S3 物件 Lambda 做為分發的來源，則必須設定以下四個許可。

Object Lambda Access Point

若要新增物件 Lambda 存取點的權限

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 在導覽窗格中，選擇 物件 Lambda 存取點。
3. 選擇您要使用的 Object Lambda 存取點。
4. 選擇許可索引標籤標籤。
5. 在 物件 Lambda 存取點政策 部分中選擇 編輯。
6. 將以下政策貼入 政策 欄位。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3-object-lambda:Get*",
            "Resource": "arn:aws:s3-object-lambda:region:AWS-account-
ID:accesspoint/Object-Lambda-Access-Point-name",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:cloudfront::AWS-account-
ID:distribution/CloudFront-distribution-ID"
                }
```

```
            }
        }
    ]
}
```

7. 選擇儲存變更。

Amazon S3 Access Point

若要新增 Amazon S3 存取點的許可

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 在導覽窗格中，選擇 存取點。
3. 選擇您要使用的 Amazon S3 存取點。
4. 選擇許可索引標籤標籤。
5. 在 存取點政策 部分中選擇 編輯。
6. 將以下政策貼入 政策 欄位。

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "s3objlambda",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:region:AWS-account-ID:accesspoint/Access-Point-
name",
                "arn:aws:s3:region:AWS-account-ID:accesspoint/Access-Point-name/
object/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "s3-object-lambda.amazonaws.com"
                }
```

```
            }
        }
    ]
}
```

7. 選擇儲存。

Amazon S3 bucket

將許可新增至 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 在導覽窗格中，選擇 儲存貯體。
3. 選擇您要使用的 Amazon S3 儲存貯體。
4. 選擇許可索引標籤標籤。
5. 在 儲存貯體政策 區段中，選擇 編輯。
6. 將以下政策貼入 政策 欄位。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "*",
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:DataAccessPointAccount": "AWS-account-ID"
                }
            }
        }
    ]
}
```

7. 選擇儲存變更。

AWS Lambda function

若要將權限新增至 Lambda 函數

1. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。
2. 在導覽視窗中，選擇函數。
3. 選擇您要使用的 AWS Lambda 功能。
4. 依序選擇 組態 索引標籤和 許可。
5. 在 基於資源的政策聲明 區段中選擇 新增許可。
6. 選擇 AWS 帳戶。
7. 輸入 聲明 ID 的名稱。
8. 在 主體 輸入 `cloudfront.amazonaws.com`。
9. 從 動作 下拉式選單中選擇 `lambda:InvokeFunction`。
10. 選擇儲存。

使用 Amazon S3 存取點

當您使用 S3 存取點時，Amazon S3 會自動為您產生唯一的別名。您可以使用此別名而不是 Amazon S3 儲存貯體名稱做為 CloudFront分發的來源。

當您使用 Amazon S3 存取點別名做為來源時 CloudFront，建議您使用下列格式：

alias.s3.*region*.amazonaws.com

如需有關尋找的詳細資訊*alias*，請參閱 Amazon S3 使用者指南中的為 S3 儲存貯體存取點使用儲存貯體樣式別名。

> **Important**
>
> 當您使用 Amazon S3 存取點做為來源時 CloudFront，必須使用來源存取控制。

CloudFront 將 Amazon S3 存取點來源視為標準 Amazon S3 儲存貯體來源相同。

如果您使用 Amazon S3 物件 Lambda 做為分發的來源，則必須設定以下兩個許可。

Amazon S3 Access Point

若要新增 Amazon S3 存取點的許可

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 在導覽窗格中，選擇 存取點。
3. 選擇您要使用的 Amazon S3 存取點。
4. 選擇許可索引標籤標籤。
5. 在 存取點政策 部分中選擇 編輯。
6. 將以下政策貼入 政策 欄位。

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "s3objlambda",
            "Effect": "Allow",
            "Principal": {"Service": "cloudfront.amazonaws.com"},
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:region:AWS-account-ID:accesspoint/Access-Point-
name",
                "arn:aws:s3:region:AWS-account-ID:accesspoint/Access-Point-name/
object/*"
            ],
            "Condition": {
                "StringEquals": {"aws:SourceArn": "arn:aws:cloudfront::AWS-
account-ID:distribution/CloudFront-distribution-ID"}
            }
        }
    ]
}
```

7. 選擇儲存。

Amazon S3 bucket

將許可新增至 Amazon S3 儲存貯體

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 在導覽窗格中，選擇 儲存貯體。
3. 選擇您要使用的 Amazon S3 儲存貯體。
4. 選擇許可索引標籤標籤。
5. 在 儲存貯體政策 區段中，選擇 編輯。
6. 將以下政策貼入 政策 欄位。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "*",
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:DataAccessPointAccount": "AWS-account-ID"
                }
            }
        }
    ]
}
```

7. 選擇儲存變更。

使用設定為網站端點的 Amazon S3 儲存貯體

您可以使用設定為網站端點的 Amazon S3 儲存貯體做為自訂來源CloudFront。設定 CloudFront 分發時，針對原始伺服器，輸入儲存貯體的 Amazon S3 靜態網站託管端點。這個值會顯示在 Amazon S3 主控台 Properties (屬性) 索引標籤中的 Static website hosting (靜態網站託管) 窗格裡。例如：

```
http://bucket-name.s3-website-region.amazonaws.com
```

如需指定 Amazon S3 靜態網站端點的詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的網站端點。

當您以此格式指定儲存貯體名稱做為您的原始伺服器時，可以使用 Amazon S3 重新引導和 Amazon S3 自訂錯誤文件。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的設定自訂錯誤文件和設定重新引導部分。（CloudFront還提供自定義錯誤頁面。 如需詳細資訊，請參閱the section called “針對特定的 HTTP 狀態碼建立自訂錯誤頁面”。）

使用 Amazon S3 儲存貯體做為 CloudFront 原始伺服器不會以任何方式變更儲存貯體。您可以如一般情況般繼續使用，且需支付一般 Amazon S3 費用。如需要使用之費用的詳細資訊 CloudFront，請參閱 Amazon CloudFront 定價。

> **Note**
>
> 如果您使用 CloudFront API 透過設定為網站端點的 Amazon S3 儲存貯體建立分發，則即使該網站託管於 Amazon S3 儲存貯體`CustomOriginConfig`，也必須使用以進行設定。如需有關使用 CloudFront API 建立分發的詳細資訊，請參閱 Amazon CloudFront API 參考CreateDistribution中的。

添加 CloudFront 到現有的 Amazon S3 存儲桶

如果您將物件存放在 Amazon S3 儲存貯體中，您可以讓使用者直接從 S3 取得物件，或者您可 CloudFront 以設定從 S3 取得物件，然後將它們散發給使用者。如果您的使用者經常存取物件，使用 CloudFront 可能會更具成本效益，因為資料傳輸的價格較高時， CloudFront 資料傳輸的價格低於 Amazon S3 資料傳輸的價格。此外，下載速度 CloudFront 比單獨使用 Amazon S3 更快，因為您的物件存放在更靠近使用者的位置。

> **Note**
>
> 如果您想 CloudFront 要遵守 Amazon S3 跨來源資源共用設定，請進行設定 CloudFront 以將標`Origin`頭轉寄至 Amazon S3。如需詳細資訊，請參閱 the section called “根據請求標頭快取內容”。

如果您目前使用自己的網域名稱 (例如 example.com) 直接從 Amazon S3 儲存貯體發佈內容，而不是 Amazon S3 儲存貯體的網域名稱 (例如文件範例儲存貯體 .s3.US 西部 2.Amazons.com)，您可以使用下列程序在不中斷的情況下新增內容。 CloudFront

若要新增您已經從 Amazon S3 發佈內容的 CloudFront 時間

1. 創建一個 CloudFront 分佈。如需詳細資訊，請參閱 the section called “建立、更新和刪除分發”。

   當您建立分佈時，請指定 Amazon S3 儲存貯體的名稱做為原始伺服器。

   > **Important**
   >
   > 若要使用值區 CloudFront，名稱必須符合 DNS 命名需求。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的儲存貯體命名規則。

   如果您將 Amazon S3 與 CNAME 搭配使用，也請為您的分佈指定 CNAME。
2. 建立一個在 Amazon S3 儲存貯體中包含公開可讀取物件的連結之測試 Web 頁面，並測試該連結。對於此初始測試，請在對象 URL 中使用分發的CloudFront 域名，例如，`https://d111111abcdef8.cloudfront.net/images/image.jpg`。

   如需 CloudFront URL 格式的詳細資訊，請參閱the section called “自訂檔案 URL”。
3. 如果您目前使用 Amazon S3 CNAME，您的應用程式會使用網域名稱 (例如 example.com) 參照 Amazon S3 儲存貯體中的物件，而不是使用儲存貯體的名稱 (例如 DOC-EXAMPLE-BUCKET.s3.amazonaws.com)。若要繼續使用您的網域名稱來參考物件，而不是使用網 CloudFront 域名稱進行分發 (例如 d111111abcdef8.cloudfront.net)，您必須透過 DNS 服務供應商更新您的設定。

   針對使用 Amazon S3 CNAME，您的 DNS 服務供應商必須有網域的 CNAME 資源紀錄集，該記錄目前將網域查詢路由到您的 Amazon S3 儲存貯體。例如，如果使用者請求該物件：

   `https://example.com/images/image.jpg`

該請求會自動重新路由，且使用者看到此物件：

```
https://DOC-EXAMPLE-BUCKET.s3.amazonaws.com/images/image.jpg
```

若要將查詢路由到您的 CloudFront 分發，而不是 Amazon S3 儲存貯體，您需要使用 DNS 服務供應商提供的方法來更新為網域設定的 CNAME 資源記錄。此更新的 CNAME 記錄會將 DNS 查詢從您的網域重新導向至您散發的 CloudFront 網域名稱。如需詳細資訊，請參閱 DNS 服務供應商提供的說明文件。

> **Note**
>
> 如果您使用 Route 53 做為 DNS 服務，您可以使用 CNAME 資源紀錄集或別名資源紀錄集。如需編輯資源紀錄集的詳細資訊，請參閱編輯記錄。如需別名資源紀錄集的詳細資訊，請參閱選擇別名或非別名記錄。這兩個主題都在 Amazon Route 53 開發人員指南中。

如需搭配使用 CNames 的詳細資訊 CloudFront，請參閱the section called “使用自訂網址”。

在您更新 CNAME 資源紀錄集後，傳播變更到整個 DNS 系統需要 72 小時，但通常更快。在此期間，您內容的某些請求將繼續路由到您的 Amazon S3 儲存貯體，而其他請求則會路由到 CloudFront。

將 Amazon S3 存儲桶移動到另一個 AWS 區域

如果您使用 Amazon S3 做為 CloudFront 分發的來源，而您將儲存貯體移到另一個儲存貯體 AWS 區域，則當下列兩項都成立時，最多 CloudFront 可能需要一個小時才能更新其記錄才能使用新區域：

- 您正在使用 CloudFront 來源存取身分識別 (OAI) 來限制儲存貯體的存取權。
- 您將儲存貯體移至需要簽章版本 4 進行身分驗證的 Amazon S3 區域。

當您使用 OAI 時， CloudFront 會使用 Region (以及其他值) 來計算用來從值區要求物件的簽章。如需 OAI 的詳細資訊，請參閱the section called “使用原始存取身分 (舊版，不建議使用)”。如需支援簽名版本 2 的 AWS 區域 清單，請參閱 Amazon Web Services 一般參考.

若要強制更新記錄 CloudFront的速度，您可以更新您的 CloudFront發行版，例如，透過更新 CloudFront 主控台中 [一般] 索引標籤上的 [說明] 欄位。當您更新分配時，請 CloudFront 立即檢查值區所在的區域。將變更傳播到所有節點只需幾分鐘即可完成。

使用 MediaStore 容器或通 MediaPackage 道

若要使用串流影片 CloudFront，您可以設定設定為 MediaStore容器的 Amazon S3 儲存貯體，或使用建立通道和端點 MediaPackage。然後，您可以在中創建並配置分發 CloudFront 以流式傳輸視頻。

如需詳細資訊和 step-by-step 指示，請參閱下列主題：

- the section called “使用 AWS Elemental MediaStore 做為原始伺服器來提供視訊”
- the section called “提供以 AWS Elemental MediaPackage 格式化的即時視訊”

使用應用程式負載平衡器

如果您的來源是託管在一或多個 Amazon EC2 執行個體上的一或多個 HTTP (S) 伺服器 (Web 伺服器)，則可以使用面向網際網路的 Application Load Balancer 器將流量分配到執行個體。面向網際網路的負載平衡器具有可公開解析的 DNS 名稱，並透過網際網路將來自用戶端的要求路由到目標。

如需有關使用 Application Load Balancer 器做為來源的詳細資訊 CloudFront，包括如何確保檢視者只能透過 CloudFront 而不能直接存取負載平衡器存取您的 Web 伺服器，請參閱the section called “限制對 Application Load Balancers 的存取”。

使用 Lambda 函數網址

Lambda 函數網址是 Lambda 函數的專用 HTTPS 端點。您可以使用 Lambda 函數 URL，完全在 Lambda 中建置無伺服器 Web 應用程式。您可以直接透過這個函數 URL 叫用 Lambda Web 應用程式，完全無需與 API Gateway 或 Application Load Balancer 進行整合。

如果您使用 Lambda 函數搭配函數 URL 來建置無伺服器 Web 應用程式，您可以新增 CloudFront 以取得下列優點：

- 在更接近檢視者的位置快取內容，加快應用程式的速度
- 讓您的 Web 應用程式使用自訂網域
- 使用 CloudFront 快取行為將不同 URL 路徑路由到不同的 Lambda 函數
- 使用 CloudFront 地理限制或 AWS WAF （或兩者）阻止特定請求

- AWS WAF 搭配使用可協 CloudFront 助保護您的應用程式不受惡意機器人攻擊、協助防止常見的應用程式入侵，並增強 DDoS 攻擊的防護

若要使用 Lambda 函數 URL 做為 CloudFront 發佈的來源，請指定 Lambda 函數 URL 的完整網域名稱做為原始網域。Lambda 函數 URL 網域名稱必須使用以下格式：

function-URL-ID.lambda-url.*AWS-Region*.on.aws

當您使用 Lambda 函數 URL 做為 CloudFront 發佈的來源時，函數 URL 必須可公開存取。若要這麼做，請使用下列其中一個選項：

- 如果您使用來源存取控制 (OAC)，Lambda 函數 URL 的`AuthType`參數必須使用該`AWS_IAM`值，並允許以資源為基礎的政策中的`lambda:InvokeFunctionUrl`權限。如需有關針對 OAC 使用 Lambda 函數 URL 的詳細資訊，請參閱限制對 AWS Lambda 函數 URL 來源的訪問。
- 如果您不使用 OAC，您可以將函數 URL 的`AuthType`參數設定為，`NONE`並允許以資源為基礎的原則中的`lambda:InvokeFunctionUrl`權限。

您也可以將自訂 Origin 標頭新增至 CloudFront 傳送至來源的要求，並撰寫函數程式碼以在要求中不存在標頭時傳回錯誤回應。這有助於確保使用者只能透過存取您的 Web 應用程式 CloudFront，而不能直接使用 Lambda 函數 URL 存取您的 Web 應用程式。

如需 Lambda 函數 URL 的詳細資訊，請參閱 AWS Lambda 開發人員指南：

- Lambda 函數 URL – Lambda 函數 URL 功能的一般概觀
- 叫用 Lambda 函數 URL – 針對編寫無伺服器 Web 應用程式程式碼時須用到的請求和回應承載，提供詳細資訊
- Lambda 函數 URL 的安全性和驗證模型 — 包含有關 Lambda 驗證類型的詳細資訊

使用 Amazon EC2（或其他自定義源地）

自訂來源是具有可公開解析 DNS 名稱的 HTTP (S) Web 伺服器，可透過網際網路將來自用戶端的要求路由到目標。HTTP (S) 伺服器可以託管在例如 Amazon EC2 執行個體上 AWS，或託管在其他地方。設為網站端點的 Amazon S3 原始伺服器也會被視為自訂原始伺服器。如需詳細資訊，請參閱 the section called “使用設定為網站端點的 Amazon S3 儲存貯體”。

當您使用自己的 HTTP 伺服器做為自訂原始伺服器時，必須指定伺服器的 DNS 名稱，以及 HTTP 和 HTTPS 連接埠，以及從原始擷取物件時 CloudFront 要使用的通訊協定。

當您使用自訂來源 (私人內容除外) 時，支援大多數 CloudFront 功能。雖然您可以使用已簽署的 URL 從自訂來源散發內容，但 CloudFront 若要存取自訂來源，原始伺服器必須保持可公開存取。如需詳細資訊，請參閱 the section called “使用已簽署的 URL 和已簽署的 Cookie 來限制內容”。

請遵循下列準則，將 Amazon EC2 執行個體和其他自訂來源搭配使用CloudFront。

- 在為相同 CloudFront來源提供內容的所有伺服器上託管並提供相同的內容。如需詳細資訊，請參閱 the section called “分佈設定” 主題中的 the section called “原始設定”。
- 記錄所有伺服器上的`X-Amz-Cf-Id`標頭項目，以便您需CloudFront 要 AWS Support 或使用此值進行偵錯。
- 限制對您自訂原始伺服器所監聽的 HTTP 與 HTTPS 連接埠提出請求。
- 同步實作中所有伺服器的時鐘。請注意，針對已簽署的 URL 和已簽署的 Cookie、記錄和報告 CloudFront 使用國際標準時間 (UTC)。此外，如果您使用 CloudWatch量度監視 CloudFront 活動，請注意 CloudWatch 也使用 UTC。
- 使用冗餘伺服器來處理故障。
- 如需有關使用自訂原始伺服器來提供私有內容的詳細資訊，請參閱the section called “ 在自訂原始伺服器上限制存取檔案”。
- 如需有關請求和回應行為，以及有關支援的 HTTP 狀態代碼的詳細資訊，請參閱請求和回應行為。

如果您使用 Amazon EC2 做為自訂原始伺服器，建議您執行以下操作：

- 使用 Amazon Machine Image，會自動為 Web 伺服器安裝軟體。如需詳細資訊，請參閱 Amazon EC2 說明文件。
- 使用 Elastic Load Balancing 負載平衡器來處理多個 Amazon EC2 執行個體的流量，並將應用程式與 Amazon EC2 執行個體的變更隔離。例如，如果您使用負載平衡器，則可以新增和刪除 Amazon EC2 執行個體，無需變更應用程式。如需詳細資訊，請參閱 Elastic Load Balancing 說明文件。
- 建立 CloudFront 分發時，請為原始伺服器的網域名稱指定負載平衡器的 URL。如需詳細資訊，請參閱 the section called “建立分發”。

使用 CloudFront 原始群組

例如，如果您想要針對需要高可用性的案例設定 CloudFront 原始容錯移轉，您可以為原始伺服器指定原始群組。使用原始容錯移轉來指定主要原點以 CloudFront 及第二個原點，當主要來源傳回特定的 HTTP 狀態碼失敗回應時， CloudFront 會自動切換到。

如需詳細資訊，包括原始伺服器群組的設定步驟，請參閱the section called “使用原始伺服器容錯移轉增加高可用性”。

新增替代網域名稱 (CNAME) 以使用自訂 URL

當您建立分發時，請為其 CloudFront 提供網域名稱，例如 d1111。您可以使用替代網域名稱 (也稱為 CNAME)，而不是使用此提供的網域名稱。

若要使用您自己的網域名稱，例如 www.example.com，請參閱下列章節：

主題

- 新增替代網域名稱
- 將替代網域名稱移至其他發行版
- 移除替代網域名稱
- 在替代網域名稱中使用萬用字元
- 使用備用網域名稱的需求
- 使用備用網域名稱的限制

新增替代網域名稱

下面的任務列表描述如何使用 CloudFront 控制台將替代域名添加到您的分發，以便您可以在鏈接中使用自己的域名，而不是 CloudFront 域名。如需使用 CloudFront API 更新發行版的相關資訊，請參閱使用發行版。

> **Note**
>
> 如果您要檢視器使用具有備用網域名稱的 HTTPS，請參閱使用備用網域名稱和 HTTPS。

在您開始進行前：請先確認您已執行以下作業，再更新您的分佈以新增備用網域名稱：

- 向 Route 53 或其他網域註冊商註冊網域名稱。
- 從涵蓋網域名稱的授權憑證授權機構 (CA) 取得 SSL/TLS 憑證。將憑證新增至您的分佈，來驗證您是否已獲得使用網域的授權。如需詳細資訊，請參閱 使用備用網域名稱的需求。

新增替代網域名稱

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您希望更新的分佈 ID。
3. 在 General (一般) 索引標籤上，選擇 Edit (編輯)。
4. 更新下列的值：

   備用網域名稱 (CNAME)

   新增您的備用網域名稱。使用逗號區隔網域名稱，或在新的一行輸入每個網域名稱。

   SSL 憑證

   選擇以下設定：

   - 使用 HTTPS – 請選擇自訂 SSL 憑證，然後在清單中選擇憑證。此清單包括 AWS Certificate Manager (ACM) 佈建的憑證、您從其他 CA 購買並上傳至 ACM 的憑證，以及您從其他 CA 購買並上傳至 IAM 憑證存放區的憑證。

     若您將憑證上傳至 IAM 憑證存放區，但清單中並未顯示該憑證，則請檢閱程序 匯入 SSL/TLS 憑證 以確認憑證是否已正確上傳。

     若選擇此設定，我們建議您僅使用您物件 URL 中的替代網域名稱 (https://www.example.com/logo.jpg)。如果您使用 CloudFront 散發網域名稱 (https://d111111abcdef8.cloudfront.net.cloudfront.net/logo.jpg)，檢視者的行為可能會如下，視您為支援的用戶端選擇的值而定：

     - 所有用戶端：如果檢視器不支援 SNI，就會顯示警告，因為 CloudFront 網域名稱與 TLS/SSL 憑證中的網域名稱不符。
     - 只有 Support 伺服器名稱指示 (SNI) 的用戶端：中 CloudFront 斷與檢視器的連線，而不傳回物件。

   用戶端支援

   選擇一個選項：

   - 所有用戶端：使用專用 IP 位址 CloudFront 提供 HTTPS 內容。如果您選取此選項，當您將 SSL/TLS 憑證與已啟用的分佈相關聯時，需要支付額外的費用。如需詳細資訊，請參閱 Amazon CloudFront 定價。
   - 僅支援伺服器名稱指示 (SNI) 的用戶端 (建議)：不支援 SNI 的舊版瀏覽器或其他用戶端必須使用其他方法存取內容。

如需詳細資訊，請參閱 選擇如何 CloudFront 提供 HTTPS 要求。

5. 請選擇 Yes, Edit (是，編輯)。
6. 在分佈的 General (一般) 索引標籤上，請確認 Distribution Status (分佈狀態) 是否已變更為 Deployed (已部署)。如果您在部署分佈的更新之前，嘗試使用備用網域名稱，則您在下列步驟中建立的連結可能無法運作。
7. 設定替代網域名稱的 DNS 服務 (例如 www.example.com)，將流量路由到您的分發的 CloudFront 網域名稱 (例如：您使用的方法取決於您是使用 Route 53 做為網域的 DNS 服務提供者，還是其他提供者。

> **Note**
>
> 若您 DNS 記錄指向的分佈不是您正在更新的分佈，您將只在更新您的 DNS 後，才能將備用網域名稱新增到您的分佈。如需詳細資訊，請參閱 使用備用網域名稱的限制。

Route 53

請建立別名資源紀錄集。使用別名資源紀錄集時，您不需要支付 Route 53 查詢。此外，您可以建立根網域名稱 (example.com) 的別名資源紀錄集，該 DNS 不允許 CNAME。如需詳細資訊，請參閱 Amazon Route 53 開發人員指南中的使用您的 CloudFront 網域名稱將流量路由到 Amazon 網路分發。

其他 DNS 服務供應商

使用 DNS 服務供應商提供的方法，為您的網域新增 CNAME 記錄。這個新的 CNAME 記錄會將 DNS 查詢從您的備用網域名稱重新導向至您分發的CloudFront 網域名稱 (例如：d111111abcdef8.cloudfront.net)。如需詳細資訊，請參閱 DNS 服務供應商提供的說明文件。

> **Important**
>
> 如果您已經有替代網域名稱的 CNAME 記錄，請更新該記錄，或將其取代為指向散佈 CloudFront 網域名稱的新記錄。

8. 使用 `dig` 或類似 DNS 工具，請確認您在先前步驟建立的 DNS 組態，是否指向分佈的網域名稱。

以下範例顯示了一個在 www.example.com 網域的 `dig` 請求和回應的相關部分。

```
PROMPT> dig www.example.com
```

```
; <<> DiG 9.3.3rc2 <<> www.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.com.      IN    A

;; ANSWER SECTION:
www.example.com. 10800 IN CNAME d111111abcdef8.cloudfront.net.
...
```

回答部分顯示 CNAME 記錄，該記錄會將查詢路由到CloudFront 分發網域名稱 d111111abcdef8.cloudfront.net。如果右側的名稱CNAME是您 CloudFront 分發的網域名稱，則 CNAME 記錄設定正確。如果它是任何其他值，例如，Amazon S3 儲存貯體的網域名稱，則 CNAME 紀錄配置錯誤。在這種情況下，請回到步驟 7 並更正 CNAME 紀錄以指向分佈的網域名稱。

9. 通過訪問帶有您的域名的 URL 而不是分發 CloudFront域名來測試替代域名。
10. 在您的應用程式中，將物件的 URL 變更為使用替代網域名稱，而非 CloudFront 散發的網域名稱。

將替代網域名稱移至其他發行版

如果您嘗試將替代網域名稱新增到分佈，但替代網域名稱已在不同分佈使用時，您會收到 `CNAMEAlreadyExists` 錯誤訊息 (您提供的一個或多個 CNAME 已與不同的資源相關聯)。例如，當您嘗試將 www.example.com 新增至分佈時，您會收到這個錯誤訊息，告知www.example.com 已與其他分佈相關聯。

在這種情況下，您可能想要將現有的替代網域名稱從一個分佈 (來源分佈) 移動到另一個 (目標分佈)。以下步驟是程序概觀。如需詳細資訊，請遵循概觀中每個步驟的連結。

移動替代網域名稱

1. 設定目標分佈。此分佈必須有涵蓋您移動之替代網域名稱的 SSL/TLS 憑證。如需詳細資訊，請參閱 設定目標分佈。
2. 尋找來源分佈。您可以使用 AWS Command Line Interface (AWS CLI) 來尋找替代網域名稱相關聯的散佈版本。如需詳細資訊，請參閱 尋找來源分佈。

3. 移動替代網域名稱。執行此操作的方式取決於來源分配與目標分配是否在同一 AWS 科目中。如需詳細資訊，請參閱 the section called “移動替代網域名稱”。

設定目標分佈

在移動替代網域名稱之前，必須先設定目標分佈 (要將替代網域名稱移動所至的分佈)。

設定目標分佈

1. 取得包含您移動之替代網域名稱的 SSL/TLS 憑證。如果沒有憑證，可以在 AWS Certificate Manager (ACM) 中申請一個憑證，或是從另一個憑證授權機構 (CA) 取得憑證，然後將其匯入 ACM。確定您是在美國東部 (維吉尼亞北部) (us-east-1) 區域請求或匯入憑證。
2. 如果尚未建立目標分佈，請立即建立。做為建立目標分佈的一部分，請將憑證 (來自上一個步驟) 與分佈建立關聯。如需詳細資訊，請參閱 建立分發。

   如果您已經有目標分佈，請將憑證 (來自上一個步驟) 與目標分佈建立關聯。如需詳細資訊，請參閱 更新分佈。
3. 建立 DNS TXT 紀錄，將替代網域名稱與目標分佈的分佈網域名稱建立關聯。建立 TXT 紀錄，並在替代網域名稱前加上底線 (_)。下列範例顯示 DNS 中的 TXT 紀錄：

   ```
   _www.example.com TXT d111111abcdef8.cloudfront.net
   ```

   CloudFront 使用此 TXT 記錄來驗證您對替代網域名稱的擁有權。

尋找來源分佈

在將替代網域名稱從一個分佈移至另一個分佈之前，您應該先找到來源分佈 (目前正在使用替代網域名稱的分佈)。知道來源與目標分佈兩者的 AWS 帳戶 ID 後，您便可以決定如何移動替代網域名稱。

尋找替代網域名稱的來源分佈

1. 使用 AWS Command Line Interface (AWS CLI) 中的CloudFront list-conflicting-aliases指令，如下列範例所示。以替代網域名稱取代 *www.example.com*，用您先前設定之目標分佈的 ID 取代 *EDFDVBD6EXAMPLE*。使用與目標發佈位於相同 AWS 帳戶中的認證來執行此命令。若要使用此命令，您必須擁有目標分佈的 `cloudfront:GetDistribution` 和 `cloudfront:ListConflictingAlias` 許可。

```
aws cloudfront list-conflicting-aliases --alias www.example.com --distribution-
id EDFDVBD6EXAMPLE
```

命令的輸出會顯示與所提供的網域名稱衝突或重疊的所有替代網域名稱清單。例如：

- 如果將 www.example.com 提供給命令，則命令的輸出會包含 www.example.com 以及重疊的萬用字元替代網域名稱 (*.example.com) (如果存在)。
- 如果將 *.example.com 提供給命令，則命令的輸出會包含 *.example.com 以及該萬用字元所涵蓋的任何替代網域名稱 (例如，www.example.com、test.example.com、dev.example.com 等)。

針對命令輸出中的每個替代網域名稱，您可以看到與替代網域名稱關聯之分佈的 ID，以及擁有該分佈的 AWS 帳戶 ID。分佈和帳戶 ID 會部分隱藏，這樣既可以讓您識別出您擁有的分佈和帳戶，同時又可以避免洩漏非您擁有之分佈和帳戶的資訊。

2. 在命令的輸出中，找到您要移動的替代網域名稱的分佈，並記下來源分發的 AWS 帳戶 ID。將來源分配的帳戶 ID 與您建立目標發佈的帳戶 ID 進行比較，並判斷這兩個分佈是否位於同一 AWS 帳戶中。這可協助您判斷如何移動替代網域名稱。

   若要移動替代網域名稱，請參閱下列主題。

移動替代網域名稱

根據您的狀況，從以下內容選擇移動替代網域名稱的方式：

如果來源和目標分佈位於相同的 AWS 帳戶

使用中的associate-alias指令 AWS CLI 移動替代網域名稱。此方法適用於所有相同帳戶的移動，包括當替代網域名稱是 Apex 網域時 (也稱為根網域，例如 example.com)。如需詳細資訊，請參閱 the section called “用associate-alias於移動替代網域名稱”。

如果來源分佈和目標分佈位於不同的 AWS 帳戶

如果您具有來源分佈的存取權，替代網域名稱不是 Apex 網域 (也稱為根網域，如 example.com)，而且您尚未使用與該替代網域名稱重疊的萬用字元，請使用萬用字元來移動替代網域名稱。如需詳細資訊，請參閱 the section called “使用萬用字源來移動替代網域名稱”。

如果您無法存取來源散發的 AWS 帳戶，您可以嘗試使用中的associate-alias命令 AWS CLI 來移動替代網域名稱。如果來源分佈已停用，您可以移動替代網域名稱。如需詳細資訊，請參閱 the section called “用associate-alias於移動替代網域名稱”。如果 associate-alias 命令沒有作用，

請聯絡 AWS Support。如需詳細資訊，請參閱 the section called “移動替代網域名稱 AWS Support 的聯絡人”。

用**associate-alias**於移動替代網域名稱

如果來源散佈與目標發佈位於相同的 AWS 帳戶中，或者如果它位於不同的帳戶中但已停用，則您可以使用中的CloudFront associate-alias命令 AWS CLI來移動替代網域名稱。

使用 associate-alias 來移動替代網域名稱

1. 使用執 AWS CLI 行命 CloudFront associate-alias令，如下列範例所示。以替代網域名稱取代 *www.example.com*，以具有目標分佈 ID 取代 *EDFDVBD6EXAMPLE*。使用與目標發佈位於相同 AWS 帳戶中的認證來執行此命令。請注意下列使用此命令的限制：

   - 您必須擁有目標分佈的 `cloudfront:AssociateAlias` 和 `cloudfront:UpdateDistribution` 許可。
   - 如果來源和目標分佈位於相同的 AWS 帳戶，您必須擁有來源分佈的 `cloudfront:UpdateDistribution` 許可。
   - 如果來源和目標分佈在不同的 AWS 帳戶，則必須停用來源分佈。
   - 目標分佈必須依照 the section called “設定目標分佈” 所述的方式設定。

   ```
   aws cloudfront associate-alias --alias www.example.com --target-distribution-
   id EDFDVBD6EXAMPLE
   ```

   此命令會從來源分佈中移除替代網域名稱，並將其新增至目標分佈中，藉此更新這兩個分佈。
2. 完全目標分佈部署之後，請更新 DNS 組態，將替代網域名稱的 DNS 紀錄指向目標分佈的分佈網域名稱。

使用萬用字源來移動替代網域名稱

如果來源散發位於與目標發佈不同的 AWS 帳戶中，且已啟用來源散佈，您可以使用萬用字元來移動替代網域名稱。

> **Note**
>
> 無法使用萬用字元來移動 Apex 網域 (例如 example.com)。若要在來源分佈和目標分佈位於不同的 AWS 帳戶時移動 Apex 網域，請聯絡 AWS Support。如需詳細資訊，請參閱 the section called “移動替代網域名稱 AWS Support 的聯絡人”。

使用萬用字源來移動替代網域名稱

> **Note**
>
> 此程序涉及對分佈的多次更新。等待每個分佈完整部署最新的變更後，再進行下一步。

1. 更新目標分佈，以新增涵蓋您要移動之替代網域名稱的萬用字元替代網域名稱。例如，如果您要移動的替代網域名稱為 www.example.com，請將該替代網域名稱 *.example.com 新增到目標分佈。若要執行這項操作，目標分佈上的 SSL/TLS 憑證必須包含萬用字元網域名稱。如需詳細資訊，請參閱 the section called “更新分佈”。
2. 更新替代網域名稱的 DNS 設定，以指向目標分佈的網域名稱。例如，如果您要移動的替代網域名稱是 www.example.com，請更新 www.example.com 的 DNS 紀錄，以將流量路由至目標分佈的網域名稱 (例如，d111111abcdef8.cloudfront.net)。

   > **Note**
   >
   > 在您更新 DNS 設定後，替代網域名稱仍會由來源分佈提供，因為這是替代網域名稱目前設定的所在位置。

3. 更新來源分佈來移除替代網域名稱。如需詳細資訊，請參閱 更新分佈。
4. 更新目標分佈來新增替代網域名稱。如需詳細資訊，請參閱 更新分佈。
5. 使用 dig (或類似的 DNS 查詢工具)，以驗證替代網域名稱的 DNS 紀錄是否解析為目標分佈的網域名稱。
6. (選用) 更新目標分佈來移除萬用字源替代網域名稱。

移動替代網域名稱 AWS Support 的聯絡人

如果來源和目標發佈位於不同的 AWS 帳戶中，而且您無法存取來源分發的 AWS 帳戶，或無法停用來源分發，您可以連絡 AWS Support 以移動替代網域名稱。

聯絡移 AWS Support 動替代網域名稱

1. 設定目標分佈，包括指向目標分佈的 DNS TXT 紀錄。如需詳細資訊，請參閱 設定目標分佈。
2. 聯繫 AWS Support以要求他們驗證您是否擁有該域，並將域名移至新的 CloudFront 發行版本。
3. 完全目標分佈部署之後，請更新 DNS 組態，將替代網域名稱的 DNS 紀錄指向目標分佈的分佈網域名稱。

移除替代網域名稱

如果您想要停止將網域或子網域的流量路由傳送至 CloudFront 發佈，請遵循本節中的步驟來更新 DNS 組態和 CloudFront 散發。

您務必要從分佈移除備用網域名稱，並更新您的 DNS 組態。如果您想要將網域名稱與其他 CloudFront 發行版產生關聯，這有助於避免日後發生問題。如果備用網域名稱已與一個分佈建立關聯，即無法以另一個分佈設定。

> **Note**
>
> 如果您想要從這個分發移除備用網域名稱來將該地址新增到另一個分發，請按照 將替代網域名稱移至其他發行版 中的步驟。如果您改為遵循此處的步驟 (移除網域)，然後將網域新增至另一個發行版，則會有一段時間內網域不會連結至新發佈，因 CloudFront 為會傳播至 Edge 位置的更新。

從分佈移動備用網域名稱

1. 若要開始，請將您網域的網際網路流量路由到另一個不是您 CloudFront 發佈的資源，例如 Elastic Load Balancing 負載平衡器。或者，您可以刪除將流量路由到的 DNS 記錄 CloudFront。

   執行以下其中一項，取決於您網域的 DNS 服務：

   - 如果您使用的是 Route 53，請更新或刪除別名記錄或 CNAME 記錄。如需詳細資訊，請參閱編輯記錄或刪除記錄。
   - 如果您使用其他 DNS 服務提供者，請使用 DNS 服務供應商提供的方法來更新或刪除將流量導向至 CloudFront的 CNAME 記錄。如需詳細資訊，請參閱 DNS 服務供應商提供的說明文件。
2. 在您更新網域的 DNS 紀錄，請等待變更傳播且 DNS 解析程式會路由流量到新的資源。您可以建立一些使用 URL 中網域的測試連結來確認此步驟是否完成。

3. 登入 AWS Management Console 並開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home，然後執行下列動作來更新您的 CloudFront 發行版以移除網域名稱：

   a. 選擇您希望更新的分佈 ID。

   b. 在 General (一般) 索引標籤上，選擇 Edit (編輯)。

   c. 在 Alternate Domain Names (CNAMEs) (正式名稱記錄 (CNAME)) 中，請移除您已不想再用於分佈的備用網域名稱 (或網域名稱)。

   d. 請選擇 Yes, Edit (是，編輯)。

在替代網域名稱中使用萬用字元

當您新增備用網域名稱時，可以改為在網域名稱的開頭使用 * 萬用字元，而非新增個別子網域。例如，使用替代網域名稱 *.example.com，您可以在 URL 中使用任何結尾為 example.com 的網域名稱，例如 www.example.com、product-name.example.com、marketing.product-name.example.com 等。不管網域名稱為何，物件的路徑都是一樣的，例如：

- www.example.com/images/image.jpg
- product-name.example.com/images/image.jpg
- marketing.product-name.example.com/images/image.jpg

按照這些包括萬用字元的備用網域名稱要求：

- 替代網域名稱必須以星號和點號 (*.) 開頭。
- 您不能使用萬用字元來取代子網域名稱的部分，例如：*domain.example.com。
- 您不得取代網域名稱中的子網域，例如：subdomain.*.example.com。
- 所有備用網域名稱 (包括使用萬用字元的備用網域名稱) 都必須涵蓋在憑證的主體別名 (SAN) 下。

萬用字元替代網域名稱 (例如 *.example.com) 可以包含其他正在使用中的替代網域名稱，例如 example.com。

使用備用網域名稱的需求

當您將替代網域名稱 (例如 www.example.com) 新增至 CloudFront 發行版時，需求如下：

備用網域名稱必須是小寫

所有替代網域名稱 (CNAME) 都必須是小寫。

備用網域名稱必須由有效的 SSL/TLS 憑證涵蓋

要將替代域名（CNAME）添加到 CloudFront 分發中，您必須將覆蓋備用域名的受信任有效 SSL/TLS 證書附加到您的分發。這樣可確保只有能夠存取您網域憑證的人員才能與 CloudFront 您網域相關的 CNAME 建立關聯。

受信任的憑證是由 AWS Certificate Manager (ACM) 或其他有效憑證授權單位 (CA) 所發行的憑證。您可以使用自我簽署憑證來驗證現有 CNAME，但無法驗證新 CNAME。 CloudFront 支援與 Mozilla 相同的憑證授權單位。如需目前的清單，請參閱 Mozilla 內建 CA 憑證清單。

若要使用您附加的憑證 (包括包含萬用字元的替代網域名稱) 來驗證替代網域名稱，請 CloudFront 檢查憑證上的主體別名 (SAN)。您新增的備用網域名稱必須由 SAN 涵蓋。

> **Note**
>
> 一次只能將一個憑證附加至 CloudFront 發行版本。

您可以執行以下作業，證明您已獲得授權，將特定的備用網域名稱新增到您的分佈：

- 附加包含替代網域名稱的憑證，例如 product-name.example.com。
- 連接網域名稱開頭包含 * 萬用字元的憑證，以使用單一憑證涵蓋多個子網域。當您指定萬用字元時，您可以在 CloudFront中新增多個子網域作為替代網域名稱。

下列範例說明如何在憑證中使用網域名稱中使用萬用字元來授權您在中新增特定替代網域名稱。CloudFront

- 您想要新增 marketing.example.com 做為替代網域名稱。在您的憑證中列出以下網域名稱：*.example.com。將此憑證附加至時 CloudFront，您可以為您的分發新增任何替代網域名稱，以取代該層級的萬用字元，包括 marketing.example.com。您也可以新增下列備用網域名稱 (範例)：
  - product.example.com
  - api.example.com

  但是，您無法新增比萬用字元層級高或低的備用網域名稱。例如，您無法新增替代網域名稱 example.com 或 marketing.product.example.com。

- 您想要新增 mexample.com 做為替代網域名稱。若要執行此作業，您必須在您連接到分佈的憑證上列出網域名稱 example.com 本身。
- 您想要新增 marketing.product.example.com 做為替代網域名稱。若要執行此作業，您可以在憑證上列出 *.example.com，或是在憑證上列出 marketing.product.example.com。

變更 DNS 組態的許可

當您新增替代網域名稱時，您必須建立 CNAME 記錄，以將替代網域名稱的 DNS 查詢路由傳送至您的 CloudFront 分發。若要執行此作業，您必須擁有為您正在使用備用網域名稱，使用 DNS 服務提供者建立 CNAME 記錄的許可。一般而言，這表示您擁有此網域，但您也可能是在為網域擁有者開發應用程式。

備用網域名稱和 HTTPS

若您希望檢視器搭配備用網域名稱使用 HTTPS，您必須完成額外設定。如需詳細資訊，請參閱 使用備用網域名稱和 HTTPS。

使用備用網域名稱的限制

請在使用備用網域名稱時，注意以下限制：

備用網域名稱的數量上限

如需您可為分發新增之備用網域名稱數量的目前上限，或是有關請求更高配額 (先前稱為限制) 的詳細資訊，請參閱分佈的一般配額。

重複和重疊的備用網域名稱

如果另一個 CloudFront 發行版中已經存在相同的替代網域名稱，即使您的 AWS 帳戶擁有其他 CloudFront 發行版，您也無法將備用網域名稱新增至分發。

不過，您可以新增萬用字元替代網域名稱，例如 *.example.com，其中包含 (重疊) 非萬用字元的替代網域名稱，例如 www.example.com。如果您在兩個發行版中有重疊的替代網域名稱，則無論 DNS 記錄指 CloudFront 向的分佈為何，都會使用更具體的名稱相符，將請求傳送給發佈。例如，marketing.domain.com 比 *.domain.com 更為具體。

網域 Fronting

CloudFront 包括防止跨不同 AWS 帳戶發生的網域前端的保護。網域前端是指非標準用戶端在一個帳戶中建立與網域名稱之間的 TLS/SSL 連線，然後在另一個 AWS 帳戶中針對不相關名稱發出 HTTPS 要求的案例。 AWS 例如，TLS 連線可能會連線到 www.example.com，接著傳送 www.example.org 的 HTTP 請求。

若要避免網域前端跨越不同 AWS 帳 AWS 戶的情況，請 CloudFront 確定擁有為特定連線提供之憑證的 AWS 帳戶一律符合擁有該相同連線上所處理之要求的帳戶。

如果兩個 AWS 帳戶號碼不相符， CloudFront 請以 HTTP 421 錯誤導向要求回應來回應，讓用戶端有機會使用正確的網域進行連線。

在網域的頂端節點 (Zone Apex) 新增備用網域名稱

當您將替代網域名稱新增至分發時，通常會在 DNS 組態中建立 CNAME 記錄，以將網域名稱的 DNS 查詢路由到您的 CloudFront 分發。不過，您不能為 DNS 命名空間，也稱為 Zone Apex 的頂端節點建立 CNAME 記錄；DNS 通訊協定不允許此操作。例如，如果您註冊 DNS 名稱 example.com，Zone Apex 就是 example.com。您無法為 example.com 建立 CNAME 紀錄，但可以為 www.example.com、newproduct.example.com 等建立 CNAME 紀錄。

如果您使用 Route 53 做為您的 DNS 服務，則可以建立別名資源紀錄集，其與 CNAME 記錄相比，有兩項優勢。您可以在頂端節點 (example.com) 建立網域名稱的別名資源紀錄集。此外，使用別名資源紀錄集時，您不需要支付 Route 53 查詢。

> **Note**
>
> 如果您啟用 IPv6，必須建立兩個別名資源紀錄集：一個給路由 IPv4 流量 (A 紀錄)，一個給路由 IPv6 流量 (AAAA 紀錄)。如需詳細資訊，請參閱 發佈設定參考 主題中的 啟用 IPv6。

如需詳細資訊，請參閱 Amazon Route 53 開發人員指南中的使用您的 CloudFront 網域名稱將流量路由到 Amazon 網路分發。

WebSockets 搭 CloudFront配發行版使用

Amazon CloudFront 支援使用以 TCP 為基礎的通訊協定 WebSocket，當您需要用戶端和伺服器之間的長期雙向連線時非常有用。持久性連線通常是即時應用程式的一項要求。您可能使用的場景 WebSockets 包括社交聊天平台，在線協作工作區，多玩家遊戲以及提供諸如金融交易平台之類的實時數據源的服務。通過 WebSocket 連接的數據可以雙向流動，以實現全雙工通信。

WebSocket 功能會自動啟用，以便與任何發行版搭配使用。若要使用 WebSockets，請在附加至發行版的快取行為中設定下列其中一項：

- 將所有查看者請求標頭轉發到您的來源。(您可以使用AllViewer 受管理的來源要求原則。)
- 特別在您的原始`Sec-WebSocket-Version`請求策略中轉發`Sec-WebSocket-Key`和請求標頭。

WebSocket 協議的工作原理

該 WebSocket 協議是一種獨立的基於 TCP 的協議，可讓您避免 HTTP 的某些超額和可能增加的延遲。

若要建立 WebSocket 連線，用戶端會傳送一般 HTTP 要求，該要求使用 HTTP 的升級語意來變更通訊協定。然後，伺服器便可以完成交握。 WebSocket 連線會保持開啟狀態，而且用戶端或伺服器可以彼此傳送資料框架，而不必每次都建立新的連線。

根據預設， WebSocket 通訊協定使用連接埠 80 進行一般 WebSocket 連線，而連接埠 443 則用於透過 TLS/SSL 的 WebSocket 連線。您為您選擇 CloudFront 檢視器通訊協定政策並通訊協定 (僅限自訂原始伺服器)套用至 WebSocket連線以及 HTTP 流量的選項。

WebSocket要求

WebSocket 請求必須符合下列標準格式的 RFC 6455。

範例用戶端請求：

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: https://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

範例伺服器回應：

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

如果用戶端或伺服器中斷 WebSocket 連線，或網路中斷，用戶端應用程式應用程式應該會重新起始與伺服器的連線。

建議設定

為了避免在使用時發生非預期的壓縮相關問題 WebSockets，我們建議您在原始請求策略中包含以下標頭：

- `Sec-WebSocket-Key`
- `Sec-WebSocket-Version`
- `Sec-WebSocket-Protocol`
- `Sec-WebSocket-Accept`
- `Sec-WebSocket-Extensions`

使用政策

Amazon CloudFront 提供三種不同類型的政策，您可以透過下列方 CloudFront 式自訂這些政策：

指定快取和壓縮設定

使用 CloudFront 快取政策，您可以指定 CloudFront 包含在快取金鑰中的 HTTP 標頭、Cookie 和查詢字串。快取索引鍵可決定檢視者的 HTTP 要求是否產生快取命中 (物件會從 CloudFront 快取提供給檢視器)。在快取金鑰中包含較少的值會增加快取命中的可能性。

您也可以使用快取原則，為快取中的物件指定 CloudFront 存留時間 (TTL) 設定值，並啟用 CloudFront 要求和快取壓縮物件。

指定值以包含在原始伺服器請求中 (但不會包含在快取金鑰中)。

透過 CloudFront 原始要求原則，您可以指定 CloudFront 包含在原始要求中的 HTTP 標頭、Cookie 和查詢字串。這些是當有快取未命中時 CloudFront 傳送至原點的要求。

快取政策中的所有值都會自動包含在原始伺服器請求中，但是藉助原始伺服器請求政策，您可以在原始伺服器請求中包含其他值，而不需將其包含在快取金鑰中。

指定要移除或新增至檢視者回應的 HTTP 標頭

使用回 CloudFront 應標頭原則，您可以控制 HTTP 標頭，該標頭 CloudFront 包含在傳送給檢視者 (網頁瀏覽器或其他用戶端) 的 HTTP 回應中。您可以從來源的 HTTP 回應中移除標頭，或在 CloudFront 傳送給檢視者的回應中新增 HTTP 標頭，而無需對來源進行任何變更或撰寫任何程式碼。

如需詳細資訊，請參閱下列主題。

主題

- the section called “控制快取金鑰”
- the section called “控制原始伺服器請求”
- 新增或移除回應標頭

控制快取金鑰

使用 Amazon CloudFront，您可以控制在 CloudFront 節點快取之物件的快取金鑰。快取金鑰是快取中每個物件的唯一 ID，它會決定檢視器請求是否會導致快取命中。當檢視者請求產生與先前請求相同的

快取金鑰，且該快取金鑰的物件位於節點的快取中且有效時，就會發生快取命中。當有快取命中時，會從 CloudFront 邊緣位置將物件提供給檢視者，這具有下列優點：

- 降低原始伺服器的負載
- 減少檢視器的延遲

當您擁有較高的快取命中率 (較高的瀏覽者請求會導致快取命中) 時，您可以從網站或應用程式獲得較佳的效能。改善快取命中率的一種方法是，只在快取金鑰中包含必要的最小值。如需詳細資訊，請參閱瞭解快取金鑰。

若要控制快取金鑰，請使用 CloudFront 快取政策。您可以將快取原則附加至 CloudFront 散發中的一或多個快取行為。

主題

- 建立快取政策
- 瞭解快取政策
- 使用受管快取政策
- 瞭解快取金鑰

建立快取政策

您可以使用快取政策藉由控制快取金鑰中包含的值 (URL 查詢字串、HTTP 標頭和 Cookie) 來改善快取命中率。您可以使用 AWS Command Line Interface (AWS CLI) 或使用 CloudFront API 在 CloudFront 主控台中建立快取政策。

建立快取政策之後，您可以將它附加到 CloudFront 分佈中的一或多個快取行為。

Console

建立快取政策 (主控台)

1. 登入AWS Management Console並開啟 CloudFront 主控台中的 [原則] 頁面，位於https://console.aws.amazon.com/cloudfront/v4/home?#/policies。
2. 選擇建立快取政策。
3. 選擇此快取政策所需的設定。如需詳細資訊，請參閱 瞭解快取政策。
4. 完成時，請選擇 Create (建立)。

建立快取政策之後，您可以將其附加到快取行為。

若要將快取政策附加至現有分佈 (主控台)

1. 在主控台中開啟 [發行版] 頁 CloudFront 面，位於https://console.aws.amazon.com/cloudfront/v4/home#/distributions。
2. 選擇要更新的分佈，然後選擇行為索引標籤。
3. 選擇要更新的快取行為，然後選擇編輯。

   或者，若要建立新的快取行為，請選擇 Create behavior (建立行為)。
4. 在 Cache key and origin requests (快取金鑰和原始伺服器請求) 一節中，請確定已選擇 Cache policy and origin request policy (快取政策和原始伺服器請求政策)。
5. 在 Cache policy (快取政策) 中，選擇要連接到此快取行為的快取政策。
6. 請在頁面底部選擇 Save changes (儲存變更)。

若要將快取政策附加至新分佈 (主控台)

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇 Create Distribution (建立分佈)。
3. 在 Cache key and origin requests (快取金鑰和原始伺服器請求) 一節中，請確定已選擇 Cache policy and origin request policy (快取政策和原始伺服器請求政策)。
4. 在 Cache policy (快取政策) 中，選擇要連接到此分佈預設快取行為的快取政策。
5. 為原始伺服器、預設快取行為和其他分佈設定選擇所需的設定。如需詳細資訊，請參閱 發佈設定參考。
6. 完成後，請選擇 Create distribution (建立分佈)。

CLI

若要使用 AWS Command Line Interface (AWS CLI) 建立快取政策，請使用 aws cloudfront create-cache-policy 命令。您可以使用輸入檔案來提供命令的輸入參數，而不是將每個個別參數指定為命令列輸入。

建立快取政策 (包含輸入檔案的 CLI)

1. 使用下列命令建立一個名為 `cache-policy.yaml` 的檔案，其中包含 create-cache-policy 命令的所有輸入參數。

```
aws cloudfront create-cache-policy --generate-cli-skeleton yaml-input > cache-
policy.yaml
```

2. 開啟您剛才建立且命名為 `cache-policy.yaml` 的檔案。編輯檔案以指定您想要的快取政策設定，然後儲存檔案。您可以從檔案中移除選用欄位，但不要移除必要欄位。

   如需有關快取政策設定的詳細資訊，請參閱 瞭解快取政策。

3. 使用下列命令，使用 `cache-policy.yaml` 檔案中的輸入參數建立快取政策。

```
aws cloudfront create-cache-policy --cli-input-yaml file://cache-policy.yaml
```

   記下命令輸出中的 `Id` 值。這是快取原則識別碼，您需要它將快取原則附加至 CloudFront 發行版本的快取行為。

將快取政策附加至現有分佈 (包含輸入檔案的 CLI)

1. 使用下列命令來儲存您要更新之 CloudFront 發行版的發佈組態。將 *`distribution_ID`* 取代為分佈的 ID。

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml >
 dist-config.yaml
```

2. 開啟您剛才建立且命名為 `dist-config.yaml` 的檔案。編輯檔案，對您要更新為使用快取政策的每個快取行為進行下列變更。

   - 在快取行為中，新增名為 `CachePolicyId` 的欄位。對於欄位值，請使用您在建立政策後記下的快取政策 ID。
   - 從快取行為中移除 `MinTTL`、`MaxTTL`、`DefaultTTL` 和 `ForwardedValues` 欄位。這些設定是在快取政策中指定的，因此您無法在相同的快取行為中包含這些欄位和快取政策。
   - 將 `ETag` 欄位重新命名為 `IfMatch`，但不要變更欄位的值。

   完成後儲存檔案。

3. 使用下列命令來更新分佈以使用快取政策。將 *`distribution_ID`* 取代為分佈的 ID。

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

若要將快取政策附加至新分佈 (包含輸入檔案的 CLI)

1. 使用下列命令建立一個名為 `distribution.yaml` 的檔案，其中包含 create-distribution 命令的所有輸入參數。

   ```
   aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
    distribution.yaml
   ```

2. 開啟您剛才建立且命名為 `distribution.yaml` 的檔案。在預設快取行為的 `CachePolicyId` 欄位中，輸入您在建立政策後記下的快取政策 ID。繼續編輯檔案以指定所需的分佈設定，然後在完成後儲存檔案。

   如需有關分佈設定的詳細資訊，請參閱 發佈設定參考。

3. 使用下列命令，使用 `distribution.yaml` 檔案中的輸入參數建立分佈。

   ```
   aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
   ```

API

若要使用 CloudFront API 建立快取政策，請使用 CreateCachePolicy. 如需有關您在此 API 呼叫中指定欄位的詳細資訊，請參閱 瞭解快取政策 和 AWS 開發套件或其他 API 用戶端的 API 參考文件。

建立快取政策之後，您可以使用下列其中一個 API 呼叫，將其附加至快取行為：

- 若要將其附加至現有發行版中的快取行為，請使用UpdateDistribution。
- 若要將其附加至新發行版中的快取行為，請使用CreateDistribution。

對於這兩個 API 呼叫，請在快取行為中的 `CachePolicyId` 欄位中提供快取請求政策的 ID。如需您在這些 API 呼叫中指定其他欄位的詳細資訊，請參閱 發佈設定參考 和 AWS 開發套件或其他 API 用戶端的 API 參考說明文件。

瞭解快取政策

您可以使用快取政策藉由控制快取金鑰中包含的值 (URL 查詢字串、HTTP 標頭和 Cookie) 來改善快取命中率。 CloudFront 針對常見使用案例提供一些預先定義的快取政策 (稱為受管政策)。您可以使用這些受管政策，也可以建立專屬於您需求的專屬快取政策。如需有關受管政策的詳細資訊，請參閱 使用受管快取政策。

快取政策包含下列設定，這些設定分類為政策資訊、存留時間 (TTL) 設定，以及快取金鑰設定。

政策資訊

名稱

用來識別快取政策的唯一名稱。在主控台中，您可以使用名稱將快取政策附加到快取行為。

描述

用來描述快取政策的備註。這是選用的，但它可以協助您識別快取政策的目的。

存活期 (TTL) 設定

存留時間（TTL）設置與`Cache-Control`和 `Expires` HTTP 標頭（如果它們在源響應中）一起使用，以確定 CloudFront 緩存中的對象保持有效的時間長度。

最短 TTL

在檢查來源以 CloudFront 查看物件是否已更新之前，您希望物件保留在 CloudFront 快取記憶體中的最短時間 (以秒為單位)。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

最長 TTL

物件在檢查來源以 CloudFront 查看物件是否已更新之前停留在 CloudFront快取記憶體中的時間上限 (以秒為單位)。 CloudFront 只有在原點傳送物件`Cache-Control`或`Expires`標頭時，才會使用此設定。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

預設 TTL

在檢查來源以 CloudFront 查看物件是否已更新之前，您希望物件保留在 CloudFront 快取記憶體中的預設時間 (秒)。 CloudFront 只有在原點未傳送`Cache-Control`或`Expires`標頭隨物件一起傳送時，才會使用此設定值作為物件的 TTL。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

> **Note**
>
> 如果 [最小 TTL]、[最大 TTL] 和 [預設 TTL] 設定都設為 0，則會停用快取。 CloudFront

快取金鑰設定

快取金鑰設定會指定檢視器要求中 CloudFront 包含在快取金鑰中的值。這些值可以包括 URL 查詢字串、HTTP 標頭和 Cookie。您包含在快取金鑰中的值會自動包含在 CloudFront 傳送至原始伺服器的請求中，稱為原始伺服器請求。如需在不影響快取金鑰的情況下控制原始伺服器請求的相關資訊，請參閱控制原始伺服器請求。

快取金鑰設定包括：

- 標頭
- Cookie
- 查詢字串
- 壓縮支援

標頭

檢視器要求中 CloudFront 包含在快取金鑰和原始要求中的 HTTP 標頭。針對標頭，您可以選擇下列設定之一：

- 無 - 檢視器請求中的 HTTP 標頭不會包含在快取金鑰中，也不會自動包含在原始伺服器請求中。
- Include the following headers (包含以下標頭) - 您可以指定檢視器請求中的哪些 HTTP 標頭會包含在快取金鑰中，並自動包含在原始伺服器請求中。

當您使用 Include the following headers (包含以下標頭) 設定時，您可以指定 HTTP 標頭的名稱，而不是它們的值。例如，請參閱下列 HTTP 標頭：

```
Accept-Language: en-US,en;q=0.5
```

在這種情況下，您可以將標頭指定為 `Accept-Language`，而不是 `Accept-Language: en-US,en;q=0.5`。但是，在快取金鑰和原始伺服器請求中， CloudFront 會包含完整的標頭，包括其值。

您也可以在快取金鑰中包含由產生 CloudFront 的特定標頭。如需詳細資訊，請參閱 the section called “新增 CloudFront 要求標頭”。

Cookie

檢視器要求中 CloudFront 包含在快取金鑰和原始要求中的 Cookie。針對 Cookie，您可以選擇下列設定之一：。

- 無 - 檢視器請求中的 Cookie 不會包含在快取金鑰中，也不會自動包含在原始伺服器請求中。
- 所有 - 檢視器請求中的所有 Cookie 都包含在快取金鑰中，並自動包含在原始伺服器請求中。
- Include specified cookies (包含指定的 Cookie) - 您可以指定檢視器請求中的哪些 Cookie 會包含在快取金鑰中，並自動包含在原始伺服器請求中。
- Include all cookies except (包含所有 Cookie，除了：) - 您可以指定檢視器請求中的哪些 Cookie 不會包含在快取金鑰中，且不會自動包含在原始伺服器請求中。所有其他 Cookie、您預期指定的 Cookie，都會包含在快取金鑰中，並自動包含在原始伺服器請求中。

當您使用 Include specified cookies (包含指定 Cookie) 或 Include all cookies except (包含所有 Cookie，除了：) 設定時，您可以依名稱指定 Cookie，而不是依值來指定 Cookie。例如，請參閱下列 `Cookie` 標頭：

```
Cookie: session_ID=abcd1234
```

在這種情況下，您可以將 Cookie 指定為 `session_ID`，而不是 `session_ID=abcd1234`。不過，在快取金鑰和原始要求中 CloudFront 包含完整 Cookie (包括其值)。

查詢字串

檢視器要求中 CloudFront 包含在快取金鑰和原始要求中的 URL 查詢字串。對於查詢字串，您可以選擇下列其中一個設定：

- 無 – 檢視器請求中的查詢字串不會包含在快取金鑰中，也不會自動包含在原始伺服器的請求中。
- 所有 – 檢視器請求中的所有查詢字串都會包含在快取金鑰中，並且也會自動包含在原始伺服器請求中。
- Include specified query strings (包含指定的查詢字串) – 您可以指定檢視器請求中的哪些查詢字串會包含在快取金鑰中，並自動包含在原始伺服器請求中。
- Include all query strings except (包含所有查詢字串，除了：) – 您可以指定檢視器請求中的哪些查詢字串不會包含在快取金鑰中，且不會自動包含在原始伺服器請求中。所有其他查詢字串 (您指定的查詢字串) 都會包含在快取金鑰中，並自動包含在原始伺服器請求中。

當您使用 Include specified query strings (包含指定的查詢字串) 或 Include all query strings except (包含所有查詢字串，除了：) 設定時，您可以依名稱來指定查詢字串，而不是依值來指定查詢字串。例如，請參閱下列 URL 路徑。

```
/content/stories/example-story.html?split-pages=false
```

在這種情況下，您可以將查詢字串指定為 `split-pages`，而不是 `split-pages=false`。不過，在快取金鑰和原始要求中包 CloudFront 含完整的查詢字串 (包括其值)。

壓縮支援

這些設 CloudFront 定可在檢視器支援時，要求並快取以 Gzip 或 Brotli 壓縮格式壓縮的物件。這些設定也允許 CloudFront 壓縮運作。檢視器透過 `Accept-Encoding` HTTP 標頭表示他們支援這些壓縮格式。

> **Note**
>
> 只有在使用 HTTPS 發送請求時，Chrome 和 Firefox 網頁瀏覽器才支援 Brotli 壓縮。這些瀏覽器不支援使用 HTTP 請求的 Brotli。

當下列任一條件成立時，請啟用這些設定：

- 當查看者支援它們時，您的原始伺服器返回 Gzip 壓縮物件 (請求包含帶有值 `gzip` 的 `Accept-Encoding` HTTP 標頭)。在這種情況下，請使用啟用 Gzip 的設置 (`true`在 CloudFront APIAWS CLI，AWSSDK 中設置`EnableAcceptEncodingGzip`為) 。AWS CloudFormation
- 當查看者支援它們時，您的原始伺服器返回 Brotli 壓縮物件 (請求包含帶有值 `br` 的 `Accept-Encoding` HTTP 標頭)。在此情況下，請使用已啟用 Brotli 的設定 (`true`在 CloudFront API、AWS AWS CLI SDK 中設定`EnableAcceptEncodingBrotli`為或)。AWS CloudFormation
- 此快取政策連接到的快取行為會設定 CloudFront 壓縮。在這種情況下，您可以啟用 Gzip 或 Brotli 的快取，或兩者的快取。啟用 CloudFront 壓縮後，啟用兩種格式的快取可協助降低資料傳出至網際網路的成本。

> **Note**
>
> 如果您為其中一種或兩種壓縮格式啟用快取，請勿在與相同快取行為相關聯的原始要求原則中包含Accept-Encoding標頭。 CloudFront當為這些格式啟用快取時，始終在原始請求中包含此標頭，因此包括Accept-Encoding在原始請求策略中沒有任何作用。

如果您的原始伺服器未傳回 Gzip 或 Brotli 壓縮物件，或是快取行為未設定壓縮，請勿啟用 CloudFront 壓縮物件的快取功能。如果您這樣做，可能會導致快取命中率降低。

以下說明這些設定如何影響 CloudFront 發佈。下列所有案例假設檢視器請求會包含 Accept-Encoding 標頭。當查看者請求不包含Accept-Encoding標頭時， CloudFront 不會在緩存鍵中包含此標頭，並且不會將其包含在相應的源請求中。

針對兩種壓縮格式啟用快取壓縮物件時

如果檢視器同時支援 Gzip 和 broTLI — 也就是說，如果gzip和br值同時位於檢視器要求的Accept-Encoding標頭中，則會執行下列動作：CloudFront

- 將標頭標準化為 Accept-Encoding: br,gzip，並在快取金鑰中包含標準化標頭。快取金鑰不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。
- 如果節點位置在快取中有符合請求且未過期的 Brotli 或 Gzip 壓縮物件，則節點位置會將物件傳回給檢視器。
- 如果邊緣位置在快取中沒有符合要求且未過期的 Brotli 或 Gzip 壓縮物件，則會在對應的原始要求中 CloudFront 包含標準化的 header (Accept-Encoding: br,gzip)。原始伺服器請求不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。

如果檢視器支援一種壓縮格式，但不支援另一種壓縮格式 (例如，if gzip 是檢視器要求Accept-Encoding標頭中的值，但 br is not — 則CloudFront 會執行下列動作：

- 將標頭標準化為 Accept-Encoding: gzip，並在快取金鑰中包含標準化標頭。快取金鑰不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。
- 如果節點位置在快取中有符合請求且未過期的 Gzip 壓縮物件，則節點位置會將物件傳回給檢視器。
- 如果邊緣位置在快取中沒有符合要求且未過期的 Gzip 壓縮物件，請在對應的原始要求中加 CloudFront 入標準化的 header (Accept-Encoding: gzip)。原始伺服器請求不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。

若要瞭解 CloudFront 如果檢視器支援 Brotli 但不支援 Gzip，請在上述範例中彼此取代兩種壓縮格式。

如果檢視器不支援 Brotli 或 GZIP (也就是說，檢視器要求中的Accept-Encoding標頭不包含br或作為值)：gzipCloudFront

- 不包含在快取金鑰中的 Accept-Encoding 標頭。
- 包含 Accept-Encoding: identity 在相應的原始伺服器請求中。原始伺服器請求不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。

針對某種壓縮格式啟用快取壓縮物件時，但不啟用另一種壓縮格式

如果檢視器支援啟用快取的格式 (例如，啟用了 Gzip 的快取壓縮物件，且檢視器支援 Gzip (gzip是檢視器要求中Accept-Encoding標頭中的值之一)，則會執行下列動作：CloudFront

- 將標頭標準化為 Accept-Encoding: gzip，並在快取金鑰中包含標準化標頭。
- 如果節點位置在快取中有符合請求且未過期的 Gzip 壓縮物件，則節點位置會將物件傳回給檢視器。
- 如果邊緣位置在快取中沒有符合要求且未過期的 Gzip 壓縮物件，請在對應的原始要求中加 CloudFront 入標準化的 header (Accept-Encoding: gzip)。原始伺服器請求不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。

當檢視器同時支援 Gzip 和 Brotli (檢視器請求中的 Accept-Encoding 標頭同時包含 gzip 和 br 值) 時，這種行為是相同的，因為在這種情況下，不會啟用快取 Brotli 的壓縮物件。

若要瞭解 CloudFront 如果針對 Brotli 啟用快取壓縮物件而不是 Gzip，請在上述範例中彼此取代兩種壓縮格式。

如果檢視器不支援啟用快取的壓縮格式 (檢視器要求中的Accept-Encoding標頭不包含該格式的值)，請執行下列動作 CloudFront：

- 不包含在快取金鑰中的 Accept-Encoding 標頭。
- 包含 Accept-Encoding: identity 在相應的原始伺服器請求中。原始伺服器請求不包含檢視者傳送的 Accept-Encoding 標頭中的其他值。

針對兩種壓縮格式停用快取壓縮物件時

停用這兩種壓縮格式的快取壓縮物件時，會 CloudFront 將標Accept-Encoding頭視為檢視器要求中的任何其他 HTTP 標頭相同。預設情況下，系統不會將其不會包含在快取金鑰中，也不包含在原始伺服器請求中。您可以將其包含在快取政策中的標頭清單或原始伺服器請求政策中，如同任何其他 HTTP 標頭。

使用受管快取政策

CloudFront 提供一組受管理的快取政策，您可以將其附加到發行版本的任何快取行為。使用受管快取政策，您不需要撰寫或維護自己的快取政策。受管政策會使用針對特定使用案例最佳化的設定。

主題

附加受管快取政策

若要使用受管快取政策，請將其附加至分佈中的快取行為。此程序與您建立快取政策時的程序相同，但您只要附加其中一個受管快取政策，而不是建立新政策。您可以依名稱 (使用主控台) 或 ID (使用 AWS CLI 或軟體開發套件) 附加政策。名稱和 ID 會列在下一節中。

如需詳細資訊，請參閱 建立快取政策。

可用的受管快取政策

下列主題說明您可以使用的受管快取政策。

主題

Amplify

在 CloudFront 主控台中檢視此原則

此政策是專為與 AWS Amplify Web 應用程式的原始伺服器搭配使用而設計。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

```
2e54312d-136d-493c-8eb9-b001f22f67d2
```

此政策包括以下設定：

- 最小 TTL：2 秒
- 最長 TTL：600 秒 (10 分鐘)
- 預設 TTL：2 秒
- 包含在快取金鑰中的標頭：
  - `Authorization`
  - `CloudFront-Viewer-Country`
  - `Host`

  由於已啟用快取壓縮物件設定，因此也會包含標準化的 `Accept-Encoding` 標頭。如需詳細資訊，請參閱壓縮支援。
- 快取金鑰中包含的 Cookie：所有 Cookie 都包含在內。
- 快取金鑰中包含的查詢字串：包含所有查詢字串。
- 快取壓縮物件設定：已啟用。如需詳細資訊，請參閱壓縮支援。

CachingDisabled

在 CloudFront 主控台中檢視此原則

此政策會停用快取。此政策對於動態內容和無法快取的請求非常有用。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`4135ea2d-6df8-44a3-9df3-4b5a84be39ad`

此政策包括以下設定：

- 最小 TTL：0 秒
- 最長 TTL：0 秒
- 預設 TTL：0 秒
- 快取金鑰中包含的標頭：無
- 快取金鑰中包含的 Cookie：無
- 快取金鑰中包含的查詢字串：無
- 快取壓縮物件設定：停用

CachingOptimized

在 CloudFront 主控台中檢視此原則

此原則的設計目的是將快取金鑰中 CloudFront 包含的值最小化，以最佳化快取效率。 CloudFront 在快取鍵中不包含任何查詢字串或 Cookie，而且只包含標準化`Accept-Encoding`標頭。這可 CloudFront 讓您在原點傳回或啟用CloudFront 邊緣壓縮時，分別快取 Gzip 和 Brotli 壓縮格式中的物件。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`658327ea-f89d-4fab-a63d-7e88639e58f6`

此政策包括以下設定：

- 最小 TTL：1 秒。
- 最長 TTL：31,536,000 秒 (365 天)。
- 預設 TTL：86,400 秒 (24 小時)。
- 快取金鑰中包含的標頭：沒有明確包含任何標頭。包含標準化的 `Accept-Encoding` 標頭，因為已啟用快取壓縮物件設定。如需詳細資訊，請參閱壓縮支援。
- 快取金鑰中包含的 Cookie：無。
- 快取金鑰中包含的查詢字串：無。
- 快取壓縮物件設定：已啟用。如需詳細資訊，請參閱壓縮支援。

CachingOptimizedForUncompressedObjects

在 CloudFront 主控台中檢視此原則

此政策的設計目的是將快取金鑰中包含的值降至最低，以最佳化快取效率。不包括查詢字串、標頭或 Cookie。此政策與前一個政策相同，但會停用快取壓縮物件設定。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`b2884449-e4de-46a7-ac36-70bc7f1ddd6d`

此政策包括以下設定：

- 最小 TTL：1 秒
- 最長 TTL：31,536,000 秒 (365 天)
- 預設 TTL：86,400 秒 (24 小時)
- 快取金鑰中包含的標頭：無

- 快取金鑰中包含的 Cookie：無
- 快取金鑰中包含的查詢字串：無
- 快取壓縮物件設定：停用

元素 MediaPackage

在 CloudFront 主控台中檢視此原則

此政策是專為與 AWS Elemental MediaPackage 端點的原始伺服器搭配使用而設計。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`08627262-05a9-4f76-9ded-b50ca2e3a84f`

此政策包括以下設定：

- 最小 TTL：0 秒
- 最長 TTL：31,536,000 秒 (365 天)
- 預設 TTL：86,400 秒 (24 小時)
- 快取金鑰中包含的標頭：
  - `Origin`

  由於已為 Gzip 啟用快取壓縮物件設定，所以也會包含標準化的 `Accept-Encoding` 標頭。如需詳細資訊，請參閱壓縮支援。
- 快取金鑰中包含的 Cookie：無
- 快取金鑰中包含的查詢字串：
  - `aws.manifestfilter`
  - `start`
  - `end`
  - `m`
- 快取壓縮物件設定：已為 Gzip 啟用。如需詳細資訊，請參閱壓縮支援。

瞭解快取金鑰

快取索引鍵可決定檢視器對 CloudFront 邊緣位置的要求是否會導致快取命中。快取金鑰是快取中物件的唯一識別碼。快取中的每個物件都有唯一的快取金鑰。

當檢視器請求產生與先前請求相同的快取金鑰，且該快取金鑰的物件位於節點的快取中且有效時，就會發生快取命中。當有快取命中時，要求的物件會從 CloudFront 邊緣位置提供給檢視者，這具有下列優點：

- 降低原始伺服器的負載
- 減少檢視器的延遲

當您的快取命中率 較高 (造成快取命中率較高的檢視器請求比例較高時)，您可以從網站或應用程式獲得較佳的效能。改善快取命中率的一種方法是，只在快取金鑰中包含必要的最小值。如需詳細資訊，請參閱下列區段。

您可以使用快取政策來修改快取金鑰中的值 (URL 查詢字串、HTTP 標頭和 Cookie)。(您也可以使用 Lambda @Edge 函數修改快取金鑰)。在修改快取金鑰之前，請務必瞭解應用程式的設計方式，以及根據檢視器請求的特性提供不同的回應的時間和方式。當查看器請求中的值確定您的來源返回的回應時，您應該在快取金鑰中包含該值。但是，如果您在快取金鑰中包含一個不影響您來源返回的回應值，則最終可能會快取重複的物件。

預設快取金鑰

根據預設， CloudFront 散發的快取金鑰包含下列資訊：

- CloudFront 發行版的網域名稱 (例如，網域名稱)
- 請求物件的 URL 路徑 (例如 `/content/stories/example-story.html`)

> **Note**
>
> `OPTIONS` 方法中包含在 `OPTIONS` 請求的快取金鑰中。這表示，`OPTIONS` 請求的回應會與 `GET` 和 `HEAD` 請求的回應分開快取。

依預設，來自檢視者請求的其他值不會包含在快取金鑰中。考慮從 Web 瀏覽器下面的 HTTP 請求。

```
            GET /content/stories/example-story.html?ref=0123abc&split-pages=false
HTTP/1.1
            Host: d111111abcdef8.cloudfront.net
            User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/68.0
            Accept: text/html,*/*
```

```
            Accept-Language: en-US,en
            Cookie: session_id=01234abcd
            Referer: https://news.example.com/
```

當像這個範例這樣的檢視器要求進入 CloudFront 邊緣位置時， CloudFront 會使用快取金鑰來判斷是否有快取命中。根據預設，只有下列請求元件包含在快取金鑰中：`/content/stories/example-story.html` 和 `d111111abcdef8.cloudfront.net`。如果請求的對象不在緩存中（緩存未命中），則 CloudFront 向原點發送請求以獲取對象。獲取對象後，將其 CloudFront 返回給查看器並將其存儲在邊緣位置的緩存中。

當 CloudFront 收到由快取鍵所決定的相同物件的另一個要求時，會立即將快取的物件 CloudFront 提供給檢視者，而不傳送要求至來源。例如，請考慮下列的 HTTP 請求在先前的請求之後進入。

```
            GET /content/stories/example-story.html?ref=xyz987&split-pages=true
HTTP/1.1
            Host: d111111abcdef8.cloudfront.net
            User-Agent: Mozilla/5.0 AppleWebKit/537.36 Chrome/83.0.4103.116
            Accept: text/html,*/*
            Accept-Language: en-US,en
            Cookie: session_id=wxyz9876
            Referer: https://rss.news.example.net/
```

此請求適用於與先前請求相同的物件，但與先前請求不同。它具有不同的 URL 查詢字串，不同的 `User-Agent` 和 `Referer` 標頭，以及不同的 `session_id` Cookie。但是，根據預設，這些值都不是快取金鑰的一部分，因此第二個請求會導致快取命中。

自訂快取金鑰

在某些情況下，您可能想要在快取金鑰中包含更多資訊，即使這樣做可能會導致較少的快取點擊次數。您可以使用快取政策指定要包含在快取金鑰中的內容。

例如，如果您的原始伺服器使用檢視器請求中的 `Accept-Language` HTTP 標頭，根據檢視器的語言傳回不同的內容，您可能需要將此標頭包含在快取金鑰中。當您這樣做時， CloudFront 使用此標頭來確定緩存命中，並在原始請求中包含標頭（發生緩存未命中時 CloudFront 發送給原始請求的請求）。

在快取索引鍵中包含其他值的一個潛在後果是，由於檢視器要求中可 CloudFront能發生的變化，最終可能會快取重複的物件。例如，檢視者可能會針對 `Accept-Language` 標頭傳送下列任何值：

- en-US,en
- en,en-US
- en-US, en
- en-US

所有這些不同的值都表示檢視器的語言是英文，但這種變化可能會導 CloudFront 致多次快取相同的物件。這可以減少快取命中並增加原始伺服器請求的數目。您可以避免這種重複，方法是不要在快取金鑰中包含 Accept-Language 標頭，而是將網站或應用程式設定為針對不同語言的內容使用不同的 URL (例如 /en-US/content/stories/example-story.html)。

對於您想要包含在快取金鑰中的任何指定值，您應該確定您瞭解該值的多少種不同變化可能會出現在檢視器請求中。對於某些請求值，將它們包含在快取金鑰中很少具有意義。例如，User-Agent 標頭可以有數千個獨特的變化，所以通常不會是要包含在快取金鑰中的候選項。具有使用者特定或工作階段特定值，以及在數千個 (甚至數百萬個) 請求中是唯一的 Cookie，也不適合包含快取金鑰的候選項。如果您在快取金鑰中包含這些值，每個唯一變化都會產生快取中物件的另一個副本。如果物件的這些副本不是唯一的，或者如果您最終得到如此大量稍微不同的物件，每個物件只會得到少量的快取命中，您可能需要考慮不同的方法。您可以從快取金鑰中排除這些高變數值，也可以將物件標示為不可快取。

自訂快取金鑰時請小心。有時候這是可取的，但它可能會產生意想不到的後果，例如快取重複的物件，降低快取命中率以及增加原始伺服器請求的數量。如果您的來源網站或應用程式需要從檢視器請求接收分析、遙測或其他用途的特定值，但這些值不會變更來源傳回的物件，請使用原始伺服器請求政策將這些值包含在原始伺服器請求中，但不會將它們包含在快取金鑰中。

控制原始伺服器請求

當檢視器要求 CloudFront 導致快取未命中 (要求的物件未在邊緣位置快取) 時， CloudFront 會將要求傳送至原始位置以擷取物件。這就是所謂的原始伺服器請求。原始伺服器請求會永遠包含來自檢視器請求的以下資訊：

- URL 路徑 (僅路徑，不含 URL 查詢字串或網域名稱)
- 請求內文 (如果有)
- CloudFront 自動包含在每個原始請求中的 HTTP 標頭Host，包括User-Agent、和 X-Amz-Cf-Id

依預設，來自檢視器請求的其他資訊，例如 URL 查詢字串、HTTP 標頭和 Cookie，不會包含在原始伺服器請求中。（例外：使用舊版緩存設置，默認情況下將標題 CloudFront 轉發到您的來源。）但是，

您可能想要在原始伺服器接收其中一些其他資訊，例如收集資料以供分析或遙測。您可以使用原始伺服器請求政策來控制原始伺服器請求中包含的資訊。

原始伺服器請求政策與控制快取金鑰的快取政策分開。這種分離可讓您在原始位置接收其他資訊，並保持良好的快取命中率 (造成快取命中的檢視器請求比例)。您可以單獨控制原始伺服器請求中包含哪些資訊 (使用原始伺服器請求政策) 以及包含在快取金鑰中 (使用快取政策) 來執行此操作。

雖然這兩種政策彼此獨立，但實際上彼此相關聯。您包含在快取金鑰 (使用快取政策) 中的所有 URL 查詢字串、HTTP 標頭和 Cookie 都會自動包含在原始伺服器請求中。使用原始伺服器請求政策，指定您要包含在原始伺服器請求中，但不包含在快取金鑰中的資訊。就像快取原則一樣，您可以將原始要求原則附加至 CloudFront 散發中的一或多個快取行為。

您也可以使用原始伺服器請求政策，將其他 HTTP 標頭新增至未包含在檢視器請求中的原始伺服器請求。這些額外的標頭是在發送源請求 CloudFront 之前添加的，標題值是根據查看者請求自動確定的。如需詳細資訊，請參閱 the section called “新增 CloudFront 要求標頭”。

主題

建立原始伺服器請求政策

您可以使用原始要求原則來控制 CloudFront 傳送至原始伺服器的要求中包含的值 (URL 查詢字串、HTTP 標頭和 Cookie)。您可以使用 AWS Command Line Interface (AWS CLI) 或使用 CloudFront API 在 CloudFront 控制台中創建源請求策略。

建立原始要求原則之後，您可以將它附加至 CloudFront 散發中的一或多個快取行為。

不需要原始伺服器請求政策。當快取行為未附加原始伺服器請求政策時，原始伺服器請求會包含快取政策中指定的所有值，但僅此而已。

> **Note**
>
> 若要使用原始伺服器請求政策，快取行為也必須使用快取政策。在沒有快取政策的情況下，您無法在快取行為中使用原始伺服器請求政策。

Console

建立原始伺服器請求政策 (主控台)

1. 登入AWS Management Console並開啟 CloudFront 主控台中的 [原則] 頁面，位於https://console.aws.amazon.com/cloudfront/v4/home?#/policies。
2. 選擇 Origin request (原始伺服器請求)，然後選擇 Create origin request policy (建立原始伺服器請求政策)。
3. 選擇此原始伺服器請求政策所需的設定。如需詳細資訊，請參閱 瞭解原始伺服器請求政策。
4. 完成時，請選擇 Create (建立)。

建立原始伺服器請求政策後，您可以將其附加至快取行為。

將原始伺服器請求政策附加到現有分佈 (主控台)

1. 在主控台中開啟 [發行版] 頁 CloudFront 面，位於https://console.aws.amazon.com/cloudfront/v4/home#/distributions。
2. 選擇要更新的分佈，然後選擇行為索引標籤。
3. 選擇要更新的快取行為，然後選擇編輯。

   或者，若要建立新的快取行為，請選擇 Create behavior (建立行為)。
4. 在 Cache key and origin requests (快取金鑰和原始伺服器請求) 一節中，請確定已選擇 Cache policy and origin request policy (快取政策和原始伺服器請求政策)。
5. 針對 Origin request policy (原始伺服器請求政策)，請選擇要連接至此快取行為的原始伺服器請求政策。
6. 請在頁面底部選擇 Save changes (儲存變更)。

將原始伺服器請求政策附加到新分佈 (主控台)

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇 Create Distribution (建立分佈)。
3. 在 Cache key and origin requests (快取金鑰和原始伺服器請求) 一節中，請確定已選擇 Cache policy and origin request policy (快取政策和原始伺服器請求政策)。
4. 對於 Origin request policy (原始伺服器請求政策)，請選擇要連接至此分佈預設快取行為的原始伺服器請求政策。

5. 為原始伺服器、預設快取行為和其他分佈設定選擇所需的設定。如需詳細資訊，請參閱 發佈設定參考。
6. 完成後，請選擇 Create distribution (建立分佈)。

CLI

若要使用 AWS Command Line Interface (AWS CLI) 建立原始伺服器請求原則，請使用 aws cloudfront create-origin-request-policy 命令。您可以使用輸入檔案來提供命令的輸入參數，而不是將每個個別參數指定為命令列輸入。

建立原始伺服器請求政策 (包含輸入檔案的 CLI)

1. 使用下列命令建立一個名為 `origin-request-policy.yaml` 的檔案，其中包含 create-origin-request-policy 命令的所有輸入參數。

```
aws cloudfront create-origin-request-policy --generate-cli-skeleton yaml-input >
 origin-request-policy.yaml
```

2. 開啟您剛才建立且命名為 `origin-request-policy.yaml` 的檔案。編輯檔案以指定您想要的原始伺服器請求政策設定，然後儲存檔案。您可以從檔案中移除選用欄位，但不要移除必要欄位。

   如需有關原始伺服器請求原則設定的詳細資訊，請參閱 瞭解原始伺服器請求政策。
3. 使用下列命令，使用 `origin-request-policy.yaml` 檔案中的輸入參數建立原始伺服器請求政策。

```
aws cloudfront create-origin-request-policy --cli-input-yaml file://origin-
request-policy.yaml
```

   記下命令輸出中的 `Id` 值。這是原始要求原則識別碼，您需要它將原始要求原則附加至 CloudFront 散發的快取行為。

若要將原始伺服器請求政策附加至現有分佈 (包含輸入檔案的 CLI)

1. 使用下列命令來儲存您要更新之 CloudFront 發行版的發佈組態。將 *`distribution_ID`* 取代為分佈的 ID。

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml >
 dist-config.yaml
```

2. 開啟您剛才建立且命名為 dist-config.yaml 的檔案。編輯檔案，對您要更新的每個快取行為進行下列變更，以使用原始伺服器請求政策。

   - 在快取行為中，新增名為 OriginRequestPolicyId 的欄位。對於欄位值，請使用您在建立政策後記下的原始伺服器請求政策 ID。
   - 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

   完成後儲存檔案。

3. 使用下列命令來更新分佈，以使用原始伺服器請求政策。將 *distribution_ID* 取代為分佈的 ID。

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

若要將原始伺服器請求政策連接至新分佈 (包含輸入檔案的 CLI)

1. 使用下列命令建立一個命名為 distribution.yaml 的檔案，其中包含 create-distribution 命令的所有輸入參數。

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
 distribution.yaml
```

2. 開啟您剛才建立且命名為 distribution.yaml 的檔案。在預設快取行為的 OriginRequestPolicyId 欄位中，輸入您在建立政策後記下的原始伺服器請求政策 ID。繼續編輯檔案以指定所需的分佈設定，然後在完成後儲存檔案。

   如需有關分佈設定的詳細資訊，請參閱 發佈設定參考。

3. 使用下列命令，使用 distribution.yaml 檔案中的輸入參數建立分佈。

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

若要使用 CloudFront API 建立原始要求政策，請使用 CreateOriginRequestPolicy. 如需有關您在此 API 呼叫中指定欄位的詳細資訊，請參閱 瞭解原始伺服器請求政策 和 AWS 開發套件或其他 API 用戶端的 API 參考文件。

建立原始伺服器請求政策後，您可以使用下列其中一個 API 呼叫，將其附加至快取行為：

- 若要將其附加至現有發行版中的快取行為，請使用UpdateDistribution。
- 若要將其附加至新發行版中的快取行為，請使用CreateDistribution。

對於這兩個 API 呼叫，請在快取行為中的 `OriginRequestPolicyId` 欄位中提供原始伺服器請求政策的 ID。如需您在這些 API 呼叫中指定其他欄位的詳細資訊，請參閱 發佈設定參考 和 AWS 開發套件或其他 API 用戶端的 API 參考說明文件。

瞭解原始伺服器請求政策

CloudFront 針對一般使用案例，提供一些預先定義的原始要求原則 (稱為受管理原則)。您可以使用這些受管政策，也可以根據您的需求建立自己的原始伺服器請求政策。如需有關受管政策的詳細資訊，請參閱 使用受管原始伺服器請求政策。

原始伺服器請求政策包含下列設定，這些設定分類為政策資訊和原始伺服器請求設定。

政策資訊

名稱

用來識別原始伺服器請求政策的唯一名稱。在主控台中，您可以使用名稱將原始伺服器請求政策附加到快取行為。

描述

描述原始伺服器請求政策的備註。這是選用的。

原始伺服器請求設定

Origin 請求設定會指定檢視器要求中包含的值，這些要求包含在 CloudFront 傳送至原始要求 (稱為來源要求) 的要求中。這些值可以包括 URL 查詢字串、HTTP 標頭和 Cookie。您指定的值會包含在原始伺服器請求中，但不會包含在快取金鑰中。如需控制快取金鑰的資訊，請參閱 控制快取金鑰。

標頭

檢視器要求中 CloudFront 包含在原始要求中的 HTTP 標頭。針對標頭，您可以選擇下列設定之一：

- 無 – 檢視器請求中的 HTTP 標頭不包含在原始伺服器請求中。
- 所有檢視器標頭 – 檢視器請求中的所有 HTTP 標頭都會包含在原始伺服器請求中。
- 所有檢視器標頭和下列 CloudFront標頭 — 檢視器要求中的所有 HTTP 標頭都包含在原始請求中。此外，您可以指定要新增至原始要求的 CloudFront 標頭。如需 CloudFront 標頭的詳細資訊，請參閱the section called “新增 CloudFront 要求標頭”。
- Include the following headers (包含下列標頭) – 您可以指定原始伺服器請求中要包含哪些 HTTP 標頭。

  > **Note**
  >
  > 請勿指定已包含在 原始伺服器自訂標頭 設定中的標頭。如需詳細資訊，請參閱 設定 CloudFront 為將自訂標頭新增至原始請求。

- 所有檢視器標頭，除了..... – 您可以指定原始伺服器請求中不包含哪些 HTTP 標頭。檢視器要求中的所有其他 HTTP 標頭都會包含在內，但指定的標頭除外。

當您使用 [全部檢視器標頭] 和 [下列 CloudFront 標頭]、[包含下列標頭] 或 [除設定以外的所有檢視器標頭] 時，您只能依標頭名稱指定 HTTP 標頭。 CloudFront 在原始請求中包含完整標頭，包括其值。

> **Note**
>
> 當您使用「除了設定以外的所有檢視器標頭」移除檢視器標`Host`頭時，會 CloudFront 將包含來源網域名稱的新標`Host`頭新增至原始請求。

Cookie

檢視器要求中 CloudFront 包含在原始要求中的 Cookie。針對 Cookie，您可以選擇下列設定之一：。

- 無 – 檢視器請求中的 Cookie 不包含在原始伺服器請求中。
- 所有 – 檢視器請求中的所有 Cookie 都會包含在原始伺服器請求中。
- 包含指定的 Cookie – 您可以指定在原始伺服器請求中納入哪些檢視器請求的 Cookie。

- 所有 Cookie，除了..... – 您可以指定原始伺服器請求中不要納入哪些檢視器請求中的 Cookie。檢視器請求中的所有其他 Cookie 都會包含在內。

當您使用「包含下列 Cookie」或「除了所有 Cookie」設定時，您只能依其名稱指定 Cookie。CloudFront 在原始請求中包含完整的 Cookie，包括其值。

查詢字串

檢視器要求中 CloudFront 包含在原始要求中的 URL 查詢字串。對於查詢字串，您可以選擇下列其中一個設定：

- 無 – 檢視器請求中的查詢字串不包含在原始伺服器請求中。
- 所有 – 檢視器請求中的查詢字串都包含在原始伺服器請求中。
- 包含指定的查詢字串 – 您可以指定原始伺服器請求中要納入檢視器請求中的哪些查詢字串。
- 所有查詢字串，除了..... – 您指定檢視器請求中的哪些查詢字符串不包含在原始伺服器請求中。包括所有其他查詢字串。

當您使用 [包含下列查詢字串] 或 [除了所有查詢字串以外的查詢字串] 設定時，您只能依名稱指定查詢字串。 CloudFront 在原始請求中包含完整的查詢字串 (包括其值)。

使用受管原始伺服器請求政策

CloudFront 提供一組受管理的來源請求策略，您可以將其附加到發行版的任何緩存行為。使用受管原始伺服器請求政策，您不需要編寫或維護自己的原始伺服器請求政策。受管政策會使用針對特定使用案例最佳化的設定。

主題

附加受管原始伺服器請求政策

若要使用受管原始伺服器請求政策，請將其附加到分佈中的快取行為。此程序與您建立原始伺服器請求政策時的程序相同，但您只需附加其中一個受管原始伺服器請求政策，而不是建立新的原始伺服器請求政策。您可以依名稱 (使用主控台) 或 ID (使用 AWS CLI 或軟體開發套件) 附加政策。名稱和 ID 會列在下一節中。

如需詳細資訊，請參閱 建立原始伺服器請求政策。

可用的受管原始伺服器請求政策

下列主題說明您可以使用的受管原始伺服器請求政策。

主題

AllViewer

在 CloudFront 主控台中檢視此原則

此政策包含檢視器請求中的所有值 (標頭、Cookie 和查詢字串)。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

```
216adef6-5c7f-47e4-b989-5492eafa07d3
```

此政策包括以下設定：

- 原始伺服器請求中包含的標頭：檢視器請求中的所有標頭
- 原始伺服器請求中包含的 Cookie：所有
- 原始伺服器請求中包含的查詢字串：全部

AllViewerAndCloudFrontHeaders-2022-06

在 CloudFront 主控台中檢視此原則

此政策包括來自檢視者要求的所有值 (標頭、Cookie 和查詢字串)，以及到 2022 年 6 月發行的所有 CloudFront CloudFront 標頭 (不包括 2022 年 6 月之後發行的標頭)。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

```
33f36d7e-f396-46d9-90e0-52428a34d9dc
```

此政策包括以下設定：

- 原始要求中包含的標頭：檢視器要求中的所有標頭，以及下列 CloudFront 標頭：
  - CloudFront-Forwarded-Proto
  - CloudFront-Is-Android-Viewer
  - CloudFront-Is-Desktop-Viewer
  - CloudFront-Is-IOS-Viewer
  - CloudFront-Is-Mobile-Viewer
  - CloudFront-Is-SmartTV-Viewer
  - CloudFront-Is-Tablet-Viewer
  - CloudFront-Viewer-Address
  - CloudFront-Viewer-ASN
  - CloudFront-Viewer-City
  - CloudFront-Viewer-Country
  - CloudFront-Viewer-Country-Name
  - CloudFront-Viewer-Country-Region
  - CloudFront-Viewer-Country-Region-Name
  - CloudFront-Viewer-Http-Version
  - CloudFront-Viewer-Latitude
  - CloudFront-Viewer-Longitude
  - CloudFront-Viewer-Metro-Code
  - CloudFront-Viewer-Postal-Code
  - CloudFront-Viewer-Time-Zone
  - CloudFront-Viewer-TLS
- 原始伺服器請求中包含的 Cookie：所有
- 原始伺服器請求中包含的查詢字串：全部

AllViewerExceptHostHeader

在 CloudFront 主控台中檢視此原則

此政策不包含來自檢視者請求的 Host 標頭，但會包含檢視器請求中的所有其他值 (標頭、Cookie 和查詢字串)。

此原則也包含 HTTP 通訊協定、HTTP 版本、TLS 版本以及所有裝置類型和檢視器位置標頭的其他CloudFront 要求標頭。

此政策旨在與 Amazon API Gateway 和 AWS Lambda 函數 URL 原始伺服器搭配使用。這些來源期望Host標頭包含原始域名，而不是 CloudFront 分發的域名。將檢視者請求中的 Host 標頭轉寄至這些原始伺服器可能會導致它們無法正常運作。

> **Note**
>
> 當您使用此受管理的來源要求原則移除檢視者的Host標頭時，會 CloudFront 將包含來源網域名稱的新Host標頭新增至原始請求。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`b689b0a8-53d0-40ab-baf2-68738e2966ac`

此政策包括以下設定：

- 原始伺服器請求中包含的標頭：檢視器請求中的所有標頭 (Host 標頭除外)
- 原始伺服器請求中包含的 Cookie：所有
- 原始伺服器請求中包含的查詢字串：全部

科斯-CustomOrigin

在 CloudFront 主控台中檢視此原則

此政策包含在原始伺服器為自訂原始伺服器時，啟用跨原始伺服器資源共用 (CORS) 請求的標頭。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`59781a5b-3903-41f3-afcb-af62929ccde1`

此政策包括以下設定：

- 原始伺服器請求中包含的標頭：
  - `Origin`
- 原始伺服器請求中包含的 Cookie：無
- 原始伺服器請求中包含的查詢字串：無

CORS-S3Origin

在 CloudFront 主控台中檢視此原則

此原則包括當原始伺服器資源 Amazon S3 儲存貯體時，啟用跨原始伺服器資源共用 (CORS) 請求的標頭。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`88a5eaf4-2fd4-4709-b370-b4c650ea3fcf`

此政策包括以下設定：

- 原始伺服器請求中包含的標頭：
  - `Origin`
  - `Access-Control-Request-Headers`
  - `Access-Control-Request-Method`
- 原始伺服器請求中包含的 Cookie：無
- 原始伺服器請求中包含的查詢字串：無

元素-MediaTailor-PersonalizedManifests

在 CloudFront 主控台中檢視此原則

此政策專為搭配本身是 AWS Elemental MediaTailor 端點的原始伺服器使用而設計。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`775133bc-15f2-49f9-abea-afb2e0bf67d2`

此政策包括以下設定：

- 原始伺服器請求中包含的標頭：
  - `Origin`
  - `Access-Control-Request-Headers`
  - `Access-Control-Request-Method`
  - `User-Agent`
  - `X-Forwarded-For`
- 原始伺服器請求中包含的 Cookie：無

- 原始伺服器請求中包含的查詢字串：全部

UserAgentRefererHeaders

在 CloudFront 主控台中檢視此原則

此政策只包含 `User-Agent` 和 `Referer` 標頭。它不包括任何查詢字串或 Cookie。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`acba4595-bd28-49b8-b9fe-13317c0390fa`

此政策包括以下設定：

- 原始伺服器請求中包含的標頭：
  - `User-Agent`
  - `Referer`
- 原始伺服器請求中包含的 Cookie：無
- 原始伺服器請求中包含的查詢字串：無

新增 CloudFront 要求標頭

您可以設定 CloudFront 為將特定的 HTTP 標頭新增至從檢視者 CloudFront 接收的要求，並轉送至您的來源或邊緣函式。這些 HTTP 標頭的值皆基於檢視者請求的特性。標頭提供有關檢視者的裝置類型、IP 地址、地理位置、請求通訊協定 (HTTP 或 HTTPS)、HTTP 版本、TLS 連線詳細內容以及 JA3 指紋。

有了這些標題，您的原始伺服器或邊緣函數即可接收有關檢視器的資訊，而不需要您編寫自己的程式碼來判斷此資訊。如果您的來源根據這些標頭中的信息返回不同的響應，則可以將它們包含在緩存密鑰中，以便單獨 CloudFront 緩存響應。例如，您的原始伺服器可能會根據檢視器所在國家/地區使用特定語言的內容進行回應，或者使用針對特定裝置類型訂製的內容。您的原始伺服器還可能會將這些標頭寫入日誌檔案，您可以使用這些檔案來確定有關檢視器所在位置、檢視器所在的裝置類型等資訊。

如果您想要在快取金鑰中包含標頭，請使用快取政策。如需更多詳細資訊，請參閱 the section called “控制快取金鑰” 及 the section called “瞭解快取金鑰”。

要在原始伺服器中接收這些表頭，但不將它們包含在快取金鑰中，請使用原始伺服器請求政策。如需詳細資訊，請參閱 the section called “控制原始伺服器請求”。

主題

用於判斷檢視器裝置類型的標頭

可以新增下列標頭來判斷檢視者的裝置類型。根據標`User-Agent`頭的值，將這些標頭的值 CloudFront 設定為`true`或`false`。如果裝置屬於多個類別，一個以上的值可以是 `true`。例如，對於某些平板電腦裝置，會`CloudFront-Is-Tablet-Viewer`將`CloudFront-Is-Mobile-Viewer`和 CloudFront設定為`true`。

- `CloudFront-Is-Android-Viewer`— 設定為何`true`時 CloudFront 判斷檢視器是配備 Android 作業系統的裝置。
- `CloudFront-Is-Desktop-Viewer`— 設定為何`true`時 CloudFront 判斷檢視器是桌面裝置。
- `CloudFront-Is-IOS-Viewer`— 設定為何`true`時 CloudFront 判斷檢視器是配備蘋果行動裝置作業系統 (例如 iPhone、iPod 觸控和部分 iPad 裝置) 的裝置。
- `CloudFront-Is-Mobile-Viewer`— 設定為何`true`時 CloudFront 判斷檢視器是行動裝置。
- `CloudFront-Is-SmartTV-Viewer`— 設定為何`true`時 CloudFront 判斷檢視器是智慧型電視。
- `CloudFront-Is-Tablet-Viewer`— 設定為何`true`時 CloudFront 判斷檢視器是平板電腦。

用於判斷檢視器位置的標頭

您可以新增下列標頭來決定檢視者的位置。 CloudFront 根據檢視者的 IP 位址決定這些標頭的值。對於這些標題值中的非 ASCII 字元，請根據 CloudFront RFC 3986 的 1.2 節對字元進行百分比編碼。

- `CloudFront-Viewer-Address` - 包含檢視者 IP 地址以及請求的來源連接埠，例如標頭值 `198.51.100.10:46532` 表示瀏覽器的 IP 地址是 198.51.100.10，請求來源連接埠是 46532。
- `CloudFront-Viewer-ASN` - 包含檢視器的自治系統編號 (ASN)。

> **Note**
>
> `CloudFront-Viewer-Address` 和 `CloudFront-Viewer-ASN` 可以在原始伺服器請求政策中新增，而不是在快取政策中新增。

- `CloudFront-Viewer-Country` – 包含檢視器國家/地區的兩個字母國家/地區代碼。如需國家/地區代碼的清單，請參閱 ISO 3166-1 alpha-2。

當您新增下列標頭時，會將它們 CloudFront 套用至所有要求，但來自AWS網路的要求除外：

- `CloudFront-Viewer-City` - 包含檢視器所在城市的名稱。
- `CloudFront-Viewer-Country-Name` – 包含檢視器所在國家/地區的名稱。
- `CloudFront-Viewer-Country-Region` – 包含代表檢視器區域的代碼 (最多三個字元)。該區域是 ISO 3166-2 代碼的第一層細分 (即最廣泛或最不具體的一層)。
- `CloudFront-Viewer-Country-Region-Name` – 包含檢視器區域的名稱。該區域是 ISO 3166-2 代碼的第一層細分 (即最廣泛或最不具體的一層)。
- `CloudFront-Viewer-Latitude` – 包含檢視器的約略緯度。
- `CloudFront-Viewer-Longitude` – 包含檢視器的約略經度。
- `CloudFront-Viewer-Metro-Code` – 包含檢視器的地鐵代碼。只有當檢視器在美國時，才會出現此問題。
- `CloudFront-Viewer-Postal-Code` – 包含檢視器的郵遞區號。
- `CloudFront-Viewer-Time-Zone` 包含檢視器的時區，採用 IANA 時區資料庫格式 (例如，`America/Los_Angeles`)。

用於判斷檢視者標頭結構的標頭

您可以新增下列標頭，協助根據檢視者傳送的標頭來識別檢視者。例如，不同的瀏覽器可能會以特定順序傳送 HTTP 標頭。若 `User-Agent` 標頭中指定的瀏覽器與該瀏覽器的預期標頭順序不同，您可以拒絕該請求。此外，若 `CloudFront-Viewer-Header-Count` 值與 `CloudFront-Viewer-Header-Order` 中的標頭數量不同，您也可以拒絕該請求。

- `CloudFront-Viewer-Header-Order` – 按要求的順序包含檢視者的標頭名稱，並以冒號分隔。例如：`CloudFront-Viewer-Header-Order: Host:User-Agent:Accept:Accept-Encoding`。超出 7,680 字元限制的標頭會被截斷。
- `CloudFront-Viewer-Header-Count` – 包含檢視者標頭的總數。

其他 CloudFront 標題

您可以新增下列標頭，判斷檢視者的通訊協定、版本、JA3 指紋和 TLS 連線詳細資料：

- `CloudFront-Forwarded-Proto` – 包含檢視器請求的通訊協定 (HTTP 或 HTTPS)。
- `CloudFront-Viewer-Http-Version` – 包含檢視器請求的 HTTP 版本。
- `CloudFront-Viewer-JA3-Fingerprint` – 包含檢視者的 JA3 指紋。JA3 指紋可協助您判斷請求是來自已知用戶端、惡意軟體或惡意機器人，或預期的 (允許清單中的) 應用程式。此標頭倚賴檢視者的 SSL/TLS `Client Hello` 封包，且僅適用於 HTTPS 請求。

  > **Note**
  >
  > 您可以在原始伺服器請求政策中新增 `CloudFront-Viewer-JA3-Fingerprint`，而不需要在快取政策中新增。

- `CloudFront-Viewer-TLS`— 包含用於檢視器與之間連線之 SSL/TLS 握手的 SSL/TLS 交握的 SSL/TLS 版本、加密和相關資訊。 CloudFront標頭值的格式如下：

  ```
  SSL/TLS_version:cipher:handshake_information
  ```

  對於 *`handshake_information`*，標頭可包含下列值：

  - `fullHandshake` – 已針對 SSL/TLS 工作階段進行完整交握。
  - `sessionResumed` – 之前的 SSL/TLS 工作階段已恢復。
  - `connectionReused` – 之前的 SSL/TLS 連線已重複使用。

  下列是此標頭的一些範例值：

  ```
  TLSv1.3:TLS_AES_128_GCM_SHA256:sessionResumed
  ```

  ```
  TLSv1.2:ECDHE-ECDSA-AES128-GCM-SHA256:connectionReused
  ```

  ```
  TLSv1.1:ECDHE-RSA-AES128-SHA256:fullHandshake
  ```

  ```
  TLSv1:ECDHE-RSA-AES256-SHA:fullHandshake
  ```

  有關此標頭值中可能存在的 SSL/TLS 版本和密碼的完整列表，請參閱 the section called “檢視器與之間支援的通訊協定和密碼 CloudFront”。

> **Note**
>
> 您可以在原始伺服器請求政策中新增 `CloudFront-Viewer-TLS`，而不需要在快取政策中新增。

了解原始伺服器請求政策和快取政策如何協同運作

您可以使用 CloudFront 原始請求策略來控制 CloudFront 發送到原始請求的請求，這些請求稱為原始請求。若要使用原始伺服器請求政策，您必須連接快取政策至相同的快取行為。在沒有快取政策的情況下，您無法在快取行為中使用原始伺服器請求政策。如需詳細資訊，請參閱 the section called “控制原始伺服器請求”。

原始請求策略和緩存策略共同運作，以確定原始請求中 CloudFront包含的值。您包含在快取金鑰 (使用快取政策) 中的所有 URL 查詢字串、HTTP 標頭和 Cookie 都會自動包含在原始伺服器請求中。您在原始伺服器請求政策中指定的任何其他查詢字串、標頭和 Cookie 都會包含在原始伺服器請求中 (但不會包含在快取金鑰中)。

原始請求政策和快取政策具有可能彼此衝突的設定。例如，一個政策可能允許某些值，而另一個政策會封鎖它們。下表說明當您同時使用原始要求原則和快取原則的設定時，原始要求中 CloudFront 包含哪些值。這些設定通常適用於所有類型的值 (查詢字串、標頭和 Cookie)，但您無法在快取政策中指定所有標頭或使用標頭封鎖清單的例外。

	原始伺服器請求政策			
	無	全部	允許清單	封鎖清單
快取政策				
無	除了每個原始伺服器請求中包含的預設值以外，來源請求中不會包含任何來自檢視器要求的值。如需詳細資訊，請參閱 the section called “控	來自檢視器請求的所有值都包含在原始伺服器請求中。	只有在原始伺服器請求政策中指定的值才會包含在原始伺服器請求中。	除了原始伺服器請求政策中指定的值以外，所有來自檢視器請求的值都包含在原始伺服器請求中。

	原始伺服器請求政策			
	無	全部	允許清單	封鎖清單
	制原始伺服器請求”。			
全部 注意：您無法在快取政策中指定所有標頭。	來自檢視器要求的所有查詢字串和 Cookie 都包含在原始伺服器請求中。	來自檢視器請求的所有值都包含在原始伺服器請求中。	來自檢視器要求的所有查詢字串和 Cookie，以及原始伺服器請求政策中指定的任何標頭，都包含在原始伺服器請求中。	來自檢視器要求的所有查詢字串和 Cookie 都包含在原始伺服器請求中，且包含在原始伺服器請求政策封鎖清單中指定的字串和 Cookie。快取政策設定會覆寫原始伺服器請求政策封鎖清單。
允許清單	原始伺服器請求中只包含來自檢視器請求的指定值。	來自檢視器請求的所有值都包含在原始伺服器請求中。	快取政策或原始伺服器請求政策中指定的所有值都包含在原始伺服器請求中。	即使原始伺服器請求政策封鎖清單中指定了相同的值，快取政策中指定的值也會包含在原始伺服器請求中。快取政策允許清單會覆寫原始要求政策封鎖清單。

	原始伺服器請求政策			
	無	全部	允許清單	封鎖清單
封鎖清單 注意：您無法在快取政策封鎖清單中指定標頭。	來自檢視器要求的所有查詢字串和 Cookie (指定的字串除外) 都包含在原始伺服器請求中。	來自檢視器請求的所有值都包含在原始伺服器請求中。	即使在快取政策封鎖清單中指定了相同的值，原始伺服器請求政策中指定的值也會包含在原始伺服器請求中。原始伺服器請求政策允許清單會覆寫快取政策封鎖清單。	來自檢視器請求的所有值 (快取政策或原始伺服器請求政策中指定的值除外) 都會包含在原始伺服器請求中。

在 CloudFront回應中新增或移除 HTTP 標頭

您可以設定 CloudFront 為修改傳送給檢視者之回應中的 HTTP 標頭。 CloudFront 在將回應傳送給檢視者之前，可以移除從來源收到的標頭，或在回應中新增標頭。且不須編寫程式碼或變更來源，即可進行這些變更。

例如，您可以移除標頭 (例如`X-Powered-By`和)，這`Vary`樣就 CloudFront 不會在傳送給檢視者的回覆中包含這些標題。或者，您可以新增如下所示的 HTTP 標頭：

- 新增 `Cache-Control` 標頭以控制瀏覽器快取。
- 新增 `Access-Control-Allow-Origin` 標頭以啟用跨原始來源資源分享 (CORS)。您也可以新增其他 CORS 標頭。
- 新增一組常見的安全性標頭，例如 `Strict-Transport-Security`、`Content-Security-Policy`、`X-Frame-Options` 等。
- 一個`Server-Timing`頭，以查看與性能和通過的請求和響應的路由相關的信息 CloudFront。

若要指定在 HTTP 回應中 CloudFront 新增或移除的標頭，請使用回應標頭原則。您可以將回應標頭原則附加至另一個快取行為，並 CloudFront 修改其傳送至符合快取行為之要求的回應中的標頭。CloudFront 修改它從緩存中提供的響應以及它從原點轉發的響應中的頭文件。如果來源回應包含在回

應標頭原則中新增的一或多個標頭，則原則可以指定是否 CloudFront 使用從來源接收到的標頭，還是使用回應標頭原則中的標頭覆寫該標頭。

CloudFront 針對一般使用案例，提供預先定義的回應標頭原則 (稱為受管理政策)。您可以使用這些受管政策，也可以建立您專屬的政策。您可以將單一回應標頭原則附加至您的 AWS 帳戶.

如需詳細資訊，請參閱下列主題。

主題

- 建立回應標頭政策
- 使用受管回應標頭政策
- 瞭解回應標頭政策

建立回應標頭政策

您可以使用回應標頭政策來指定 Amazon 在 HTTP 回應中 CloudFront 新增或移除的 HTTP 標頭。如需回應標頭政策及其使用原因的詳細資訊，請參閱the section called “新增或移除回應標頭”。

您可以在 CloudFront 主控台中建立回應標頭政策。或者，您可以使用AWS CloudFormation、AWS Command Line Interface (AWS CLI) 或 CloudFront API 來建立一個。建立回應標頭原則之後，您可以將它附加至 CloudFront 散發中的一或多個快取行為。

建立自訂回應標頭政策之前，請先確認是否有任何受管回應標頭政策適合您的使用案例；如果有，您就可以將該政策連接到您的快取行為，而不需自行建立或管理回應標頭政策。

Console

建立回應標頭政策 (主控台)

1. 登入AWS Management Console，然後移至 CloudFront 主控台中 [原則] 頁面上的 [回應標頭] 索引標籤https://console.aws.amazon.com/cloudfront/v4/home#/policies/responseHeaders。
2. 選擇 Create response headers policy (建立回應標頭政策)。
3. 在 Create response headers policy (建立回應標頭政策) 表單中執行下列動作：

   a. 在 Details (詳細資訊) 面板中，輸入回應標頭政策的 Name (名稱) 和 (選用) 解釋政策用途的 Description (描述)。

   b. 在 Cross-origin resource sharing (CORS) (跨來源資源分享 (CORS)) 面板中，選擇 Configure CORS (設定 CORS) 切換開關，並設定您要新增至政策的任何 CORS 標頭。如果您希望設定的標頭取代從原點 CloudFront 接收的標頭，請選取「原點取代」勾選方塊。

如需 CORS 標頭設定的詳細資訊，請參閱 the section called “CORS 標頭”。

c. 在 Security headers (安全性標頭) 面板中，選擇切換開關並設定您要新增至政策的每個安全性標頭。

如需安全性標頭設定的詳細資訊，請參閱 the section called “安全性標頭”。

d. 在 Custom headers (自訂標頭) 面板中，新增您想要納入政策的任何自訂標頭。

如需自訂標頭設定的詳細資訊，請參閱 the section called “自訂標頭”。

e. 在「移除標題」面板中，新增您要 CloudFront 從來源回應中移除的任何標題名稱，而不要包含在 CloudFront傳送給檢視者的回應中。

如需移除標頭設定的詳細資訊，請參閱 the section called “移除標頭”。

f. 在 Server-Timing header (Server-Timing 標頭) 面板中，選擇 Enable (啟用) 切換，然後輸入採樣率 (須為介於 0 到 100 之間的數值，包括 0 與 100)。

如需有關 `Server-Timing` 標頭的詳細資訊，請參閱the section called “Server-Timing 標頭”。

4. 選擇 Create (建立) 以建立政策。

建立回應標頭原則之後，您可以將其附加至 CloudFront 散發中的快取行為。

若要將回應標頭政策連接至現有分佈 (主控台)

1. 在主控台中開啟 [發行版] 頁 CloudFront 面，位於https://console.aws.amazon.com/cloudfront/v4/home#/distributions。
2. 選擇要更新的分佈，然後選擇 Behaviors (行為) 索引標籤。
3. 選擇要更新的快取行為，然後選擇 Edit (編輯)。

   或者，若要建立新的快取行為，請選擇 Create behavior (建立行為)。
4. 對於 Response headers policy (回應標頭政策)，選擇要新增至快取行為的政策。
5. 選擇 Save changes (儲存變更) 以更新快取行為。如果要建立新的快取行為，請選擇 Create behavior (建立行為)。

若要將回應標頭政策連接至新分佈 (主控台)

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。

2. 選擇 Create Distribution (建立分佈)。
3. 對於 Response headers policy (回應標頭政策)，選擇要新增至快取行為的政策。
4. 為您的分佈選擇其他設定。如需詳細資訊，請參閱 the section called “分佈設定”。
5. 選擇 Create distribution (建立分佈) 以建立分佈。

AWS CloudFormation

若要以 AWS CloudFormation 建立回應標頭政策，請使用 `AWS::CloudFront::ResponseHeadersPolicy` 資源類型。下列範例顯示用於建立回應標頭政策的 AWS CloudFormation 範本語法 (YAML 格式)。

```
Type: AWS::CloudFront::ResponseHeadersPolicy
Properties:
  ResponseHeadersPolicyConfig:
    Name: EXAMPLE-Response-Headers-Policy
    Comment: Example response headers policy for the documentation
    CorsConfig:
      AccessControlAllowCredentials: false
      AccessControlAllowHeaders:
        Items:
          - '*'
      AccessControlAllowMethods:
        Items:
          - GET
          - OPTIONS
      AccessControlAllowOrigins:
        Items:
          - https://example.com
          - https://docs.example.com
      AccessControlExposeHeaders:
        Items:
          - '*'
      AccessControlMaxAgeSec: 600
      OriginOverride: false
    CustomHeadersConfig:
      Items:
        - Header: Example-Custom-Header-1
          Value: value-1
          Override: true
        - Header: Example-Custom-Header-2
          Value: value-2
```

```
          Override: true
    SecurityHeadersConfig:
      ContentSecurityPolicy:
        ContentSecurityPolicy: default-src 'none'; img-src 'self'; script-src
 'self'; style-src 'self'; object-src 'none'; frame-ancestors 'none'
        Override: false
      ContentTypeOptions: # You don't need to specify a value for 'X-Content-Type-
Options'.
                          # Simply including it in the template sets its value to
 'nosniff'.
        Override: false
      FrameOptions:
        FrameOption: DENY
        Override: false
      ReferrerPolicy:
        ReferrerPolicy: same-origin
        Override: false
      StrictTransportSecurity:
        AccessControlMaxAgeSec: 63072000
        IncludeSubdomains: true
        Preload: true
        Override: false
      XSSProtection:
        ModeBlock: true # You can set ModeBlock to 'true' OR set a value for
 ReportUri, but not both
        Protection: true
        Override: false
    ServerTimingHeadersConfig:
      Enabled: true
      SamplingRate: 50
    RemoveHeadersConfig:
      Items:
        - Header: Vary
        - Header: X-Powered-By
```

如需詳細資訊，請參閱《AWS CloudFormation使用指南》中的「AWS::CloudFront::ResponseHeaders策略」。

CLI

若要以 AWS Command Line Interface (AWS CLI) 建立回應標頭政策，請使用 aws cloudfront create-response-headers-policy 命令。您可以使用輸入檔案來提供命令的輸入參數，而不必分別將每個個別參數指定為命令列輸入。

建立回應標頭政策 (包含輸入檔案的 CLI)

1. 使用下列命令建立名為 response-headers-policy.yaml 的檔案。這個檔案中包含 create-response-headers-policy 命令的所有輸入參數。

```
aws cloudfront create-response-headers-policy --generate-cli-skeleton yaml-input
 > response-headers-policy.yaml
```

2. 開啟您剛才建立的 response-headers-policy.yaml 檔案。編輯檔案以指定政策名稱和所需的回應標頭政策組態，然後儲存檔案。

   如需有關回應標頭政策設定的詳細資訊，請參閱 the section called “瞭解回應標頭政策”。

3. 使用下列命令建立回應標頭政策。您所建立的政策會使用 response-headers-policy.yaml 檔案中的輸入參數。

```
aws cloudfront create-response-headers-policy --cli-input-yaml file://response-
headers-policy.yaml
```

   記下命令輸出中的 Id 值，這是回應標頭政策的 ID，您需要它將原則附加至 CloudFront 散發的快取行為。

將回應標頭政策連接至現有分佈 (包含輸入檔案的 CLI)

1. 使用下列命令來儲存您要更新之 CloudFront 發行版的發佈組態。將 *distribution_ID* 改為分佈的 ID。

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml >
 dist-config.yaml
```

2. 開啟您剛才建立且命名為 dist-config.yaml 的檔案。編輯檔案，對快取行為進行下列變更，以便讓該行為使用此回應標頭政策。

   - 在快取行為中新增名為 ResponseHeadersPolicyId 的欄位。對於欄位值，請使用您在建立政策後記下的回應標頭政策 ID。
   - 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

完成後儲存檔案。

3. 使用下列命令來更新分佈以使用回應標頭政策。將 *distribution_ID* 改為分佈的 ID。

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

若要將回應標頭政策連接至新分佈 (包含輸入檔案的 CLI)

1. 使用下列命令建立名為 `distribution.yaml` 的檔案。這個檔案中包含 create-distribution 命令的所有輸入參數。

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
 distribution.yaml
```

2. 開啟您剛才建立的 `distribution.yaml` 檔案。在預設快取行為的 `ResponseHeadersPolicyId` 欄位中，輸入您在建立政策後記下的回應標頭政策 ID。繼續編輯檔案以指定所需的分佈設定，然後在完成後儲存檔案。

   如需有關分佈設定的詳細資訊，請參閱 發佈設定參考。

3. 使用下列命令，使用 `distribution.yaml` 檔案中的輸入參數建立分佈。

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

若要使用 CloudFront API 建立回應標頭政策，請使用 CreateResponseHeadersPolicy. 如需有關您在此 API 呼叫中指定欄位的詳細資訊，請參閱 the section called “瞭解回應標頭政策” 和 AWS 開發套件或其他 API 用戶端的 API 參考文件。

建立回應標頭政策之後，您可以使用下列其中一個 API 呼叫，將其連接至快取行為：

- 若要將其附加至現有發行版中的快取行為，請使用UpdateDistribution。
- 若要將其附加至新發行版中的快取行為，請使用CreateDistribution。

對於這兩個 API 呼叫，請在快取行為中的 `ResponseHeadersPolicyId` 欄位中提供回應標頭政策的 ID。如需您在這些 API 呼叫中指定其他欄位的詳細資訊，請參閱 發佈設定參考 和 AWS 開發套件或其他 API 用戶端的 API 參考說明文件。

使用受管回應標頭政策

使用回 CloudFront 應標頭政策，您可以指定 Amazon 在傳送給檢視者的回應中 CloudFront移除或新增的 HTTP 標頭。如需回應標頭政策及其使用原因的詳細資訊，請參閱the section called “新增或移除回應標頭”。

CloudFront 提供受管回應標頭原則，您可以將這些原則附加至發行 CloudFront 版中的快取行為。使用受管回應標頭政策，您不需要撰寫或維護自己的政策。受管政策包含適用於常見使用案例的 HTTP 回應標頭集。

主題

- 連接受管回應標頭政策
- 可用的受管回應標頭政策

連接受管回應標頭政策

若要使用受管回應標頭政策，請將其連接至分佈中的快取行為。這項程序與建立自訂回應標頭政策的程序相同，只不過您並不會建立新政策，而是連接其中一項受管政策。您可以依名稱 (使用主控台) 或 ID (使用 AWS CloudFormation、AWS CLI 或 AWS 開發套件) 連接政策。名稱和 ID 會列在下一節中。

如需詳細資訊，請參閱 the section called “建立回應標頭政策”。

可用的受管回應標頭政策

下列主題說明您可以使用的受管回應標頭政策。

主題

- 科斯-和-SecurityHeadersPolicy
- CORS-With-Preflight
- 科斯-with-preflight-and SecurityHeadersPolicy
- SecurityHeadersPolicy
- SimpleCORS

科斯-和-SecurityHeadersPolicy

在 CloudFront 主控台中檢視此原則

使用此受管政策允許來自任何原始伺服器的簡單 CORS 請求。此原則也會在 CloudFront 傳送給檢視者的所有回覆中新增一組安全性標頭。這項政策將 the section called “SimpleCORS” 和 the section called “SecurityHeadersPolicy” 政策結合為一個政策。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

`e61eb60c-9c35-4d20-a928-2b84e02af89c`

政策設定

	標頭名稱	標頭值	是否覆寫原始伺服器？
CORS 標頭：	`Access-Control-Allow-Origin`	`*`	否
安全性標頭：	`Referrer-Policy`	`strict-origin-when-cross-origin`	否
	`Strict-Transport-Security`	`max-age=31536000`	否
	`X-Content-Type-Options`	`nosniff`	是
	`X-Frame-Options`	`SAMEORIGIN`	否
	`X-XSS-Protection`	`1; mode=block`	否

CORS-With-Preflight

在 CloudFront 主控台中檢視此原則

使用此受管政策可允許來自任何原始伺服器的 CORS 請求，包括預檢請求。針對預檢要求 (使用 HTTP 方`OPTIONS`法)，會 CloudFront 將下列所有三個標頭新增至回應。對於簡單的 CORS 請求，只 CloudFront 添加標`Access-Control-Allow-Origin`頭。

如果從來源 CloudFront 接收的響應包含任何這些標頭，則在對查看器的響應中 CloudFront 使用接收的頭（及其值）。 CloudFront不使用此原則中的標頭。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

```
5cc3b908-e619-4b99-88e5-2cf7f45965bd
```

政策設定

	標頭名稱	標頭值	是否覆寫原始伺服器？
CORS 標頭：	`Access-Control-Allow-Methods`	`DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT`	否
	`Access-Control-Allow-Origin`	`*`	
	`Access-Control-Expose-Headers`	`*`	

科斯-with-preflight-and SecurityHeadersPolicy

在 CloudFront 主控台中檢視此原則

使用此受管政策可允許來自任何原始伺服器的 CORS 請求，包含預檢請求。此原則也會在 CloudFront 傳送給檢視者的所有回覆中新增一組安全性標頭。這項政策將 the section called “CORS-With-Preflight” 和 the section called “SecurityHeadersPolicy” 政策結合為一個政策。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

```
eaab4381-ed33-4a86-88ca-d9558dc6cd63
```

政策設定

	標頭名稱	標頭值	是否覆寫原始伺服器？
CORS 標頭：	Access-Control-Allow-Methods	DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT	否
	Access-Control-Allow-Origin	*	
	Access-Control-Expose-Headers	*	
安全性標頭：	Referrer-Policy	strict-origin-when-cross-origin	否
	Strict-Transport-Security	max-age=31536000	否
	X-Content-Type-Options	nosniff	是
	X-Frame-Options	SAMEORIGIN	否
	X-XSS-Protection	1; mode=block	否

SecurityHeadersPolicy

在 CloudFront 主控台中檢視此原則

使用此受管理原則可將一組安全性標頭新增至 CloudFront 傳送給檢視者的所有回應。如需這些安全性標頭的詳細資訊，請參閱 Mozilla 的 Web 安全指南。

使用此響應標頭策略， CloudFront 添加X-Content-Type-Options: nosniff到所有響應。從來源 CloudFront收到的響應包含此頭文件以及未包含此標題時，就是這種情況。針對此原則中的所有其他標頭，如果從來源 CloudFront 接收的回應包含標頭，則會在回應檢視器時 CloudFront 使用接收到的標頭 (及其值)。而不是此政策中的標頭。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

67f7725c-6f97-4210-82d7-5512b31e9d03

政策設定

	標頭名稱	標頭值	是否覆寫原始伺服器？
安全性標頭：	Referrer-Policy	strict-origin-when-cross-origin	否
	Strict-Transport-Security	max-age=31536000	否
	X-Content-Type-Options	nosniff	是
	X-Frame-Options	SAMEORIGIN	否
	X-XSS-Protection	1; mode=block	否

SimpleCORS

在 CloudFront 主控台中檢視此原則

使用此受管政策允許來自任何原始伺服器的簡單 CORS 請求。使用此原則， CloudFront 將標頭新增Access-Control-Allow-Origin: *至簡單 CORS 要求的所有回應。

如果從原點 CloudFront 接收到的響應包含標Access-Control-Allow-Origin題，則在對查看器的響應中 CloudFront 使用該標頭（及其值）。 CloudFront 不使用此原則中的標頭。

使用AWS CloudFormationAWS CLI、或 CloudFront API 時，此原則的識別碼為：

60669652-455b-4ae9-85a4-c4c02393f86c

政策設定

	標頭名稱	標頭值	是否覆寫原始伺服器？
CORS 標頭：	`Access-Control-Allow-Origin`	*	否

瞭解回應標頭政策

您可以使用回應標頭政策來指定 Amazon 在傳送給檢視者的回應中 CloudFront 移除或新增的 HTTP 標頭。如需回應標頭政策及其使用原因的詳細資訊，請參閱the section called “新增或移除回應標頭”。

下列主題說明回應標頭政策中的設定。這些設定分為多個類別，下列主題中依次介紹這些類別。

主題

政策詳細資訊 (中繼資料)

政策詳細資訊設定包含回應標頭政策的中繼資料。

- 名稱 - 用來識別回應標頭政策的名稱。在主控台中，您可以使用名稱將政策連接至快取行為。
- 描述(選用) - 用來描述回應標頭政策的註解。這是選用的設定，但它可以協助您識別政策的目的。

CORS 標頭

跨來源資源分享 (CORS) 設定可讓您在回應標頭政策中新增和設定 CORS 標頭。

此清單著重於如何在回應標頭政策中指定設定和有效值。如需進一步瞭解這些標頭以及如何實際將標頭用於 CORS 請求和回應，請參閱 MDN Web Docs 網站上的跨原始來源資源分享與 CORS 通訊協定規格。

Access-Control-Allow-Credentials

這是一個布爾設置（`true`或`false`），用於確定是否在 CORS 請求的響應中 CloudFront 添加`Access-Control-Allow-Credentials`標題。當此設定設定為時`true`，會在 CORS 要求的回應中 CloudFront 新增`Access-Control-Allow-Credentials: true`標頭。否 CloudFront 則，請勿將此標頭新增至回應。

Access-Control-Allow-Headers

指定 CloudFront 用作 CORS 預檢要求回應標`Access-Control-Allow-Headers`頭值的標頭名稱。此設定的有效值包括 HTTP 標頭名稱或萬用字元 (*)，用以表示所有標頭皆受允許。

> **Note**
>
> 標`Authorization`頭不能使用通配符，必須明確列出。

萬用字元的有效使用範例

範例	相符	不相符
`x-amz-*`	`x-amz-test` `x-amz-`	`x-amz`
`x-*-amz`	`x-test-amz` `x--amz`	
`*`	除了 `Authorization` 之外的所有標頭	`Authorization`

Access-Control-Allow-Methods

指定 CloudFront 用來做為 CORS 預檢要求回應`Access-Control-Allow-Methods`標頭值的 HTTP 方法。有效值包含 `GET`、`DELETE`、`HEAD`、`OPTIONS`、`PATCH`、`POST`、`PUT`、`ALL`。`ALL` 是包含所有已列出 HTTP 方法的特殊值。

Access-Control-Allow-Origin

指定 CloudFront 可在`Access-Control-Allow-Origin`回應標頭中使用的值。此設定的有效值包括特定原始伺服器 (例如 `http://www.example.com`)，或者代表允許所有原始伺服器的萬用字元 (*)。請參見下列表格中的範例。

> **Note**
>
> 允許在網域最左邊 (`*.example.org`) 使用萬用字元 (*)。
> 下列位置不允許使用萬用字元 (*)：
>
> - 頂層網域 (`example.*`)
> - 子網域右側 (`test.*.example.org`)
> - 條款內部 (`exa*mple.org`)

此資料表會顯示萬用字元的有效使用範例：

範例	相符	不相符

`http://*.example.org`	`http://www.example.org`
	
	`http://test.example.org`
	
	`http://test.example.org:123`
		`https``://test.example.org`
		
		`https``://test.example.org:123`

`*.example.org`	`test.example.org`
	
	`test.test.example.org`
	
	`.example.org`
	
	`http://test.example.org`	

範例	相符	不相符
	https://test.example.org http://test.example.org:123 https://test.example.org:123	
example.org	http://example.org https://example.org	
http://example.org		**https**://example.org http://example.org:123
http://example.org:*	http://example.org:123 http://example.org	
http://example.org:1*3	http://example.org:123 http://example.org:1893 http://example.org:13	
.example.org:1	test.example.org:123	

Access-Control-Expose-Headers

指定 CloudFront 用作 CORS 要求回應標`Access-Control-Expose-Headers`頭值的標頭名稱。此設定的有效值包括 HTTP 標頭名稱或萬用字元 (*)。

Access-Control-Max-Age

秒數，CloudFront 用作 CORS 預檢要求回應`Access-Control-Max-Age`標頭的值。

覆寫原始伺服器

Boolean 設定，可決定來源回應包含原則中其中一個 CORS 標頭時的 CloudFront 行為方式。

- 當設定為`true`且原始回應包含也位於原則中的 CORS 標頭時，會將原則中的 CORS 標頭 CloudFront 新增至回應。 CloudFront 然後將該響應發送給查看者。 CloudFront 忽略它從原點接收到的標題。
- 當設定為`false`且原始回應包含 CORS 標頭 (無論 CORS 標頭是否在原則中) 時，會 CloudFront 包含它從來源接收到回應的 CORS 標頭。 CloudFront 不會將原則中的任何 CORS 標頭新增至傳送給檢視器的回應。

安全性標頭

安全性標頭設定可讓您在回應標頭政策中新增及設定數個與安全性相關的 HTTP 回應標頭。

這份清單會說明如何在回應標頭政策中指定設定和有效值，如需進一步瞭解各個標頭以及如何實際將標頭用於 HTTP 回應，請參閱 MDN Web Docs 網站上的相關連結。

Content-Security-Policy

指定用來做為`Content-Security-Policy`回應標頭值的 CloudFront 內容安全性原則指示詞。

如需此標頭和有效政策指令的詳細資訊，請參閱 MDN Web 文件中的 Content-Security-Policy。

> **Note**
>
> `Content-Security-Policy` 標頭值限制為 1783 個字元。

推薦網站政策

指定 CloudFront 用作`Referrer-Policy`回應標頭值的反向連結原則指示詞。此設定的有效值為 `no-referrer`、`no-referrer-when-downgrade`、`origin`、`origin-when-cross-origin`、`same-origin`、`strict-origin`、`strict-origin-when-cross-origin`、`unsafe-url`。

如需此標頭和這些指令的詳細資訊，請參閱 MDN Web 文件中的 Referrer-Policy。

Strict-Transport-Security

指定 CloudFront 用來做為`Strict-Transport-Security`回應標頭值的指令和設定。對於此設定，您可以分別指定：

- 秒數，它 CloudFront 使用作為此標頭的`max-age`指令的值
- Boolean 設定 (`true`或`false`)`preload`，用於決定是否在此標頭的值中 CloudFront 包含`preload`指示詞
- Boolean 設定 (`true`或`false`)`includeSubDomains`，用於決定是否在此標頭的值中 CloudFront包含`includeSubDomains`指示詞

如需此標頭和這些指令的詳細資訊，請參閱 MDN Web 文件中的 Strict-Transport-Security。

X-Content-Type-Options

這是一個布爾設置（`true`或`false`），用於確定是否 CloudFront 將標`X-Content-Type-Options`題添加到響應。當此設定為時`true`，會 CloudFront 將標`X-Content-Type-Options: nosniff`頭新增至回應。否則 CloudFront 不會添加此標頭。

如需此標頭的詳細資訊，請參閱 MDN Web 文件中的 X-Content-Type-Options。

X-Frame-Options

指定 CloudFront 用作`X-Frame-Options`回應標頭值的指示詞。此設定的有效值為 `DENY` 或 `SAMEORIGIN`。

如需此標頭和這些指令的詳細資訊，請參閱 MDN Web 文件中的 X-Frame-Options。

X-XSS-Protection

指定 CloudFront 用來做為`X-XSS-Protection`回應標頭值的指令和設定。對於此設定，您可以分別指定：

- `0` (禁用 XSS 篩選) 或 `1` (啟用 XSS 篩選) 的 `X-XSS-Protection` 設定
- Boolean 設定 (`true`或`false`)`block`，可決定此標頭的值中是否 CloudFront 包含`mode=block`指示詞
- 報告 URI，決定是否在此標頭的值中 CloudFront 包含`report=`*`reporting URI`*指令

您可以將 `block` 指定為 `true`，也可以指定報告 URI，但不能同時指定兩者。如需此標頭和這些指令的詳細資訊，請參閱 MDN Web 文件中的 X-XSS-Protection。

覆寫原始伺服器

這些安全標頭設置中的每一個都包含一個布爾設置（`true`或`false`），該設置可確定來自來源的響應包含該標頭時的 CloudFront 行為方式。

當此設定設為`true`且來源回應包含標頭時，會將原則中的標頭 CloudFront 新增至其傳送給檢視器的回應。同時忽略從原始伺服器收到的標頭。

當此設定設定為`false`且來源回應 CloudFront 包含標頭時，會在傳送給檢視器的回應中包含從來源接收的標頭。

當來源回應不包含標頭時，會將原則中的標頭 CloudFront 新增至傳送給檢視者的回應中。CloudFront 當此設置設置為`true`或時執行此操作`false`。

自訂標頭

您可以使用自訂標頭設定，在回應標頭原則中新增和設定自訂 HTTP 標頭。 CloudFront 將這些標題添加到返回給查看者的每個響應中。您可以為每個自訂標頭指定值，但不指定也沒關係，這是因為 CloudFront 可以添加一個沒有值的響應頭。

每個自訂標頭也有自己的原始伺服器覆寫設定：

- 當此設定設為`true`且來源回應包含原則中的自訂標頭時，會將原則中的自訂標頭 CloudFront 新增至傳送給檢視器的回應。同時忽略從原始伺服器收到的標頭。
- 當此設定為`false`且來源回應包含原則中的自訂標頭時，會在傳送給檢視器的回應中 CloudFront 包含從來源接收的自訂標頭。
- 當原始回應不包含原則中的自訂標頭時，會將原則中的自訂標頭 CloudFront 新增至傳送給檢視者的回應。 CloudFront 當此設置設置為`true`或時執行此操作`false`。

移除標頭

您可以指定要 CloudFront 從來源收到的回覆中移除的標頭，這樣標頭就不會包含在 CloudFront 傳送給檢視者的回覆中。 CloudFront 從發送給查看器的每個響應中刪除標題，無論對象 CloudFront是從緩存還是從源提供。例如，您可以移除不適用於瀏覽器的標頭，例如`X-Powered-By`或`Vary`，以便從傳送給檢視者的回應中 CloudFront 移除這些標頭。

當您使用回應標頭原則指定要 CloudFront 移除的標頭時，請先移除標頭，然後新增在回應標頭原則 (CORS 標頭、安全性標頭、自訂標頭等) 中指定的任何標頭。如果您指定要移除的標頭，但同時在原則的另一個區段中新增相同的標頭， CloudFront 請在傳送給檢視者的回應中加入標頭。

> **Note**
>
> 您可以使用回應標頭政策來移除從來源 CloudFront 接收的`Server`和`Date`標頭，這樣這些標頭 (從來源接收) 就不會包含在 CloudFront 傳送給檢視者的回應中。不過，如果您這麼做，請在傳

送給檢視者的回覆中 CloudFront 新增自己的這些標頭版本。對於 CloudFront添加的`Server`標題，標題的值是`CloudFront`。

您無法移除的標頭

您無法使用回應標頭政策移除下列標頭。如果您在回應標頭政策 (API 中的 `ResponseHeadersPolicyRemoveHeadersConfig`) 的 Remove headers (移除標頭) 區段中指定這些標頭，就會收到錯誤訊息。

- `Connection`
- `Content-Encoding`
- `Content-Length`
- `Expect`
- `Host`
- `Keep-Alive`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `Proxy-Connection`
- `Trailer`
- `Transfer-Encoding`
- `Upgrade`
- `Via`
- `Warning`
- `X-Accel-Buffering`
- `X-Accel-Charset`
- `X-Accel-Limit-Rate`
- `X-Accel-Redirect`
- `X-Amz-Cf-.*`
- `X-Amzn-Auth`
- `X-Amzn-Cf-Billing`
- `X-Amzn-Cf-Id`

- X-Amzn-Cf-Xff
- X-Amzn-ErrorType
- X-Amzn-Fle-Profile
- X-Amzn-Header-Count
- X-Amzn-Header-Order
- X-Amzn-Lambda-Integration-Tag
- X-Amzn-RequestId
- X-Cache
- X-Edge-.*
- X-Forwarded-Proto
- X-Real-Ip

Server-Timing 標頭

使用標Server-Timing頭設定來啟用從傳送的 Server-Timing HTTP 回應中的標頭 CloudFront。您可以使用此標頭來查看指標，以幫助您獲得有關以及來源行為 CloudFront 和性能的見解。例如，您可以查看哪個快取層出現了快取命中，或是在發生快取遺漏時，查看來自原始伺服器的第一個位元組延遲時間。Server-Timing標頭中的指標可協助您疑難排解問題或測試 CloudFront 或來源組態的效率。

如需將Server-Timing標頭搭配使用的詳細資訊 CloudFront，請參閱下列主題。

如要啟用 Server-Timing 標頭，請建立 (或編輯) 回應標頭政策。

主題

- 取樣率和 Pragma 請求標頭
- 來自原始伺服器的 Server-Timing 標頭
- Server-Timing 標頭指標
- Server-Timing 標頭範例

取樣率和 Pragma 請求標頭

您在回應標頭政策中啟用 Server-Timing 標頭後，還可以指定抽樣率。採樣率是 0-100（含）的數字，它指定了要 CloudFront 添加Server-Timing標題的響應百分比。當您將取樣率設定為 100 時，會針對符合回應標Server-Timing頭原則所附 CloudFront 加之快取行為的每個要求，將標頭新增至

HTTP 回應。當您將其設定為 50 時，會 CloudFront 將標頭新增至符合快取行為之要求的 50% 回應。您可以將抽樣率設定為 0 至 100 之間的任何數值，最多可精確到小數點後四位數字。

當採樣率設定為小於 100 的數字時，您無法控制要將Server-Timing標頭 CloudFront 新增至哪些回應，只能控制百分比。不過，假如您為 HTTP 請求新增 Pragma 標頭，並將值設定為 server-timing，那麼該請求的回應便會一定收到 Server-Timing 標頭，無論抽樣率設為多少都一樣；即使採樣率設置為零（0），如果請求包含標Server-Timing頭，則將標Pragma: server-timing頭 CloudFront 添加到響應中。

來自原始伺服器的 Server-Timing 標頭

當存在快取未命中並將要求 CloudFront 轉送至原始位置時，來源可能會在其回應中包含Server-Timing標頭。 CloudFront在此情況下，會 CloudFront 將其度量新增至從來源接收的Server-Timing標頭。 CloudFront 傳送給檢視器的回應包含單一Server-Timing標頭，其中包含來自來源的值以及 CloudFront 新增的量度。來自來源的標頭值可能位於結尾，或介於兩組 CloudFront 新增至標頭的量度之間。

當有快取命中時， CloudFront 傳送給檢視器的回應會包含單一Server-Timing標頭，其中只包含標頭值中的 CloudFront 度量 (不包括來源的值)。

Server-Timing 標頭指標

將Server-Timing標頭 CloudFront 新增至 HTTP 回應時，標頭的值會包含一或多個量度，可協助您深入瞭解以及來源的行為 CloudFront 和效能。以下清單包含所有指標及其可能的值；標Server-Timing頭僅包含其中一些指標，具體取決於請求和響應的性質 CloudFront。

Server-Timing 標頭中的部分指標僅有名稱 (沒有值)，有些指標則有名稱和一個值。如果指標具有值，名稱和值之間會以半形分號分隔 (;)；假如標頭包含多個指標，則各個指標之間會以半形逗號分隔 (,)。

cdn-cache-hit

CloudFront 提供了來自緩存的響應，而不向原點發出請求。

cdn-cache-refresh

CloudFront 在向原點發送請求後，提供了緩存中的響應，以驗證緩存對象是否仍然有效。在這種情況下， CloudFront 沒有從原點檢索完整的對象。

cdn-cache-miss

CloudFront 沒有提供來自緩存的響應。在這種情況下，在返回響應之前從來源 CloudFront請求完整對象。

cdn-pop

包含描述哪個存在 CloudFront 點 (POP) 處理要求的值。

cdn-rid

包含具有請求 CloudFront 唯一識別碼的值。透過 AWS Support排解問題時，您可以使用這個請求標識符 (RID)

cdn-hit-layer

當 CloudFront 提供快取回應而未向來源提出要求時，就會顯示此測量結果。包含以下其中一個值：

- 邊緣- CloudFront 提供了來自 POP 位置的緩存響應。
- REC — CloudFront 提供來自區域邊緣快取 (REC) 位置的快取回應。
- 起源 Shield 牌- CloudFront 提供了來自 REC 的緩存響應，這是作為起源護 Shield。

cdn-upstream-layer

當從來源 CloudFront 要求完整物件時，會顯示此測量結果，且包含下列其中一個值：

- EDGE – POP 位置將請求直接傳送給了原始伺服器。
- REC – REC 位置將請求直接傳送給了原始伺服器。
- Origin Shield – 做為 Origin Shield 的 REC 將請求直接傳送給了原始伺服器。

cdn-upstream-dns

包含一個值，用於表示從原始伺服器擷取 DNS 記錄所花費的毫秒數。值為零 (0) 表示 CloudFront 使用快取的 DNS 結果或重複使用現有的連線。

cdn-upstream-connect

包含一個值，用於表示完成原始伺服器 DNS 請求以及 TCP (與 TLS，如有) 完成原始伺服器連線之間所花費的毫秒數。值為零 (0) 表示已 CloudFront 重複使用現有連線。

cdn-upstream-fbl

包含一個值，用於表示完成原始伺服器 HTTP 請求以及來自原始伺服器的回應收到第一個位元組之間所花費的毫秒數 (第一個位元組延遲時間)。

cdn-downstream-fbl

包含一個值，用於表示邊緣節點完成收到請求以及將回應的第一個位元組傳送給檢視者之間所花費的毫秒數。

Server-Timing 標頭範例

以下是啟用Server-Timing標頭設定 CloudFront 時，檢視者可能會收到的Server-Timing標頭範例。

Example – 快取遺漏

下列範例會顯示當要求的物件不在 CloudFront 快取中時，檢視者可能會收到的Server-Timing標頭。

```
Server-Timing: cdn-upstream-layer;desc="EDGE",cdn-upstream-dns;dur=0,cdn-upstream-
connect;dur=114,cdn-upstream-fbl;dur=177,cdn-cache-miss,cdn-pop;desc="PHX50-C2",cdn-
rid;desc="yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQUe9H1ifslzWhb0w7aLbFvGg==",cdn-downstream-
fbl;dur=436
```

此 Server-Timing 標頭表示以下內容：

- 原始請求是從存在 CloudFront 點 (POP) 位置 (cdn-upstream-layer;desc="EDGE") 傳送的。
- CloudFront 使用了來源（cdn-upstream-dns;dur=0）的緩存 DNS 結果。
- 完成與來源 () 的 TCP（和 TLS，如果適用）連接需 CloudFront 要 114 毫秒。cdn-upstream-connect;dur=114
- 完成 request (cdn-upstream-fbl;dur=177) 後，從來源接收響應的第一個字節需 CloudFront 要 177 毫秒。
- 請求的對象不在 CloudFront的緩存（cdn-cache-miss）中。
- 請求是由 PHX50-C2 代碼識別的節點接收的 (cdn-pop;desc="PHX50-C2")。
- 此要求的 CloudFront 唯一識別碼為 yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQUe9H1ifslzWhb0w7aLbFvGg== (cdn-rid;desc="yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQUe9H1ifslzWhb0w7aLbFvGg==")。
- 收 CloudFront 到查看器請求（cdn-downstream-fbl;dur=436）後，將響應的第一個字節發送給查看器花了 436 毫秒。

Example – 快取命中

下列範例會顯示當要求的物件位於快取中時，檢視者可能會收到 CloudFront的Server-Timing標頭。

```
Server-Timing: cdn-cache-hit,cdn-pop;desc="SEA19-C1",cdn-
rid;desc="nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9diOpeVc7xsrLKj-g==",cdn-hit-
layer;desc="REC",cdn-downstream-fbl;dur=137
```

此 Server-Timing 標頭表示以下內容：

- 快取中含有請求的物件 (cdn-cache-hit)。
- 請求是由 SEA19-C1 代碼識別的節點接收的 (cdn-pop;desc="SEA19-C1")。
- 此要求的 CloudFront 唯一識別碼為 nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9diOpeVc7xsrLKj-g== (cdn-rid;desc="nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9diOpeVc7xsrLKj-g==")。
- 請求的物件是在區域節點快取 (REC) 位置進行快取的 (cdn-hit-layer;desc="REC")。
- 收 CloudFront 到查看器請求（ cdn-downstream-fbl;dur=137 ）後，將響應的第一個字節發送給查看器花費 137 毫秒。

新增、移除或取代 CloudFront散佈的內容

本節說明如何確定 CloudFront 可以存取您要提供給檢視者的內容、如何在網站或應用程式中指定物件，以及如何移除或取代內容。

主題

新增和存取 CloudFront 散佈的內容

當您 CloudFront 要分發內容 (物件) 時，可以將檔案加入至您為發佈指定的其中一個來源，然後公開檔案的 CloudFront 連結。 CloudFront 邊緣位置不會從原始位置擷取新檔案，直到邊緣位置收到檢視者的要求。如需詳細資訊，請參閱 如何 CloudFront 提供內容。

當您新增要 CloudFront 分發的檔案時，請確定將其新增到分發中指定的其中一個 Amazon S3 儲存貯體，或針對自訂原始檔案，新增至指定網域中的目錄。此外，請確認適用的快取行為中路徑模式是否傳送請求到正確原始伺服器。

例如，假設快取行為的路徑模式是 `*.html`。如果您沒有設定任何其他快取行為來將要求轉寄至該來源，則只 CloudFront 會轉寄`*.html`檔案。例如，在這個案例中，永遠不 CloudFront 會散佈您上傳至原始檔的 .jpg 檔案，因為您尚未建立包含 .jpg 檔案的快取行為。

CloudFront 服務器不確定它們服務的對象的 MIME 類型。當您上傳檔案到原始伺服器時，建議您為該檔案設定 `Content-Type` 標頭欄位。

使用 CloudFront 發行版更新現有內容

有兩種方法可以更新已設定 CloudFront 為您分發的現有內容：

- 使用相同的名稱更新檔案
- 在檔案名稱中使用版本識別符進行更新

我們建議您在檔案名稱或資料夾名稱中使用版本識別碼，以協助您更好地控制所 CloudFront 提供內容的管理。

使用版本控制的檔案名稱更新現有檔案

當您更新 CloudFront 發行版中的現有檔案時，我們建議您在檔案名稱或目錄名稱中加入某種版本識別碼，以便讓自己更好地控制您的內容。此識別碼可能是日期時間戳記、序號或區分兩個相同物件版本的其他方法。

例如，您可能會命名圖形檔案為 image_1.jpg，而不是命名為 image.jpg。當您想要開始提供檔案的新版本，則命名新檔案名稱為 image_2.jpg 並更新在 Web 應用程式或網站的連結以指向 image_2.jpg。或者，您可能會將所有圖形放在 images_v1 目錄中，且當您想要開始提供一或多個圖形的新版本時，您可建立新的 images_v2 目錄，並更新連結以指向該目錄。使用版本控制，您不必等待物件過期，才能 CloudFront 開始提供新版本的物件，而且您不必為物件失效付費。

即使您對檔案進行版本控制，仍建議您設定過期日期。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

> **Note**
>
> 指定版本控制的檔案名稱或目錄名稱與 Amazon S3 物件版本控制無關。

使用相同的檔案名稱更新現有內容

雖然您可以更新 CloudFront 發行版中的現有檔案並使用相同的檔案名稱，但我們不建議您這麼做。CloudFront 只有在要求檔案時，才會將檔案分散到邊緣位置，而不是在您將新檔案或更新的檔案放在原始檔案時。若您使用同名的較新版本更新原始伺服器中的現有檔案，節點並不會從原始伺服器取得新版本，除非以下兩種情況皆發生：

- 快取中的舊版本檔案過期。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。
- 在該節點有使用者請求取得檔案。

如果您在取代檔案時使用相同的名稱，則無法控制何時 CloudFront 開始提供新檔案。依預設，會在節點中 CloudFront 快取檔案 24 小時。(如需詳細資訊，請參閱「管理內容保持在快取中達多久時間 (過期)」。) 例如，若您取代了整個網站上的所有檔案：

- 較不熱門的頁面其檔案可能不會在任何節點中。下次請求時將開始提供這些檔案的新版本。
- 某些頁面的檔案可能會在一些節點中但其他節點並沒有，所以最終使用者會看到不同的版本，具體取決於是從哪個節點提供。
- 最受歡迎頁面的檔案新版本可能無法提供長達 24 小時，因為您 CloudFront 可能已經擷取了這些頁面的檔案，就在您以新版本取代檔案之前。

刪除內容，因此 CloudFront 不會分發

您可以從來源移除不想再包含在 CloudFront發行版中的檔案。不過， CloudFront 會繼續顯示邊緣快取中的檢視者內容，直到檔案過期為止。

若您想要立即移除檔案，就必須執行以下其中一項：

- 使該檔案失效。如需詳細資訊，請參閱 使檔案失效。
- 使用檔案版本控制。當您使用版本控制時，不同版本的檔案會有不同的名稱，您可以在 CloudFront 發行版本中使用這些名稱，以變更傳回給檢視者的檔案。如需詳細資訊，請參閱 使用版本控制的檔案名稱更新現有檔案。

自訂中檔案的 URL 格式 CloudFront

使用想要提供 CloudFront 給檢視者的物件 (內容) 來設定來源之後，您必須使用正確的 URL 來參照網站或應用程式程式碼中的這些物件， CloudFront以便提供服務。

針對網頁或 Web 應用程式中的物件，您在 URLs 中使用的網域名稱可以是下列任一種：

- 建立分發時 CloudFront 自動指派的網域名稱`d111111abcdef8.cloudfront.net`，例如
- 您自己的網域名稱，例如 `example.com`

例如，您可以使用下列其中一個 URLs 傳回檔案 `image.jpg`：

```
https://d111111abcdef8.cloudfront.net/images/image.jpg
```

```
https://example.com/images/image.jpg
```

可以使用相同的 URL 格式無論在 Amazon S3 儲存貯體或自訂原始伺服器存放內容，像是自己的 Web 伺服器其中之一。

> **Note**
>
> URL 格式部分取決於您在分佈中為 Origin Path (原始伺服器路徑) 指定的值。這個值會提 CloudFront 供物件的頂層目錄路徑。如需有關建立分佈時設定原始伺服器路徑的詳細資訊，請參閱原始伺服器路徑。

如需 URL 格式的詳細資訊，請參閱下列各節：

使用您自己的網域名稱 (example.com)

您可以新增更容易使用的替代網域名稱，而不是使用在建立分發時為您 CloudFront 指派的預設網域名稱`example.com`。通過使用設置您自己的域名 CloudFront，您可以將這樣的 URL 用於分發中的對象：

```
https://example.com/images/image.jpg
```

如果您打算在檢視者和之間使用 HTTPS CloudFront，請參閱使用備用網域名稱和 HTTPS。

在 URL 中使用結尾斜線 (/)

當您為 CloudFront 發行版中的目錄指定 URL 時，請選擇永遠使用尾隨斜線，或永遠不要使用尾隨斜線。例如，只選擇下列其中一種格式來適用於所有 URL：

```
https://d111111abcdef8.cloudfront.net/images/
```

```
https://d111111abcdef8.cloudfront.net/images
```

它為什麼重要？

這兩種格式都可以連結至 CloudFront 物件，但保持一致有助於避免在您稍後想要使目錄失效時發生問題。CloudFront 完全按照定義的方式儲存 URL，包括尾端斜線。因此，如果您的格式不一致，則需要使用和不使用斜線使目錄 URL 無效，以確保 CloudFront 刪除目錄。

必須讓兩種 URL 格式都失效是麻煩事，且還會導致額外成本。這是因為如果您必須加倍失效處理以涵蓋這兩種類型的 URL，則可能會超出該月允許的免費失效處理數上限。如果發生這種情況，您必須支付所有無效的費用，即使每個目錄 URL 中只有一種格式存在。 CloudFront

建立適用於受限制內容的簽署 URL

如果您想要限制存取內容，可以建立簽署 URL。例如，如果只想分配您的內容給已授權的使用者，可以建立僅適用於特定時段內有效的 URL，或只從指定 IP 地址的 URL。如需詳細資訊，請參閱 使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。

指定預設根物件

您可以設 CloudFront 定為在使用者要求發佈的根 URL 而非要求散佈中的物件時，傳回特定物件 (預設根物件)。指定預設根物件能讓您避免暴露分佈的內容。

主題

- 如何指定預設根物件
- 預設根物件的運作方式
- 如果您沒有定義根對象，該如何 CloudFront 工作

如何指定預設根物件

為了避免暴露分佈的內容或傳回錯誤，請完成以下步驟為您的分佈指定預設根物件。

指定分佈的預設根物件

1. 上傳預設根物件到您分佈指向的原始伺服器。

   該文件可以是支持的任何類型 CloudFront。如需檔案名稱的約束清單，請參閱中`DefaultRootObject`元素的說明DistributionConfig。

   > **Note**
   >
   > 如果預設根物件的檔案名稱太長或包含無效字元，則會 CloudFront 傳回錯誤`HTTP 400 Bad Request - InvalidDefaultRootObject`。此外， CloudFront 快取程式碼 10 秒 (依預設)，並將結果寫入存取記錄。

2. 確認物件的權限至 CloudFront 少授與`read`存取權。

   如需 Amazon S3 許可的詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的 Amazon S3 中的身分和存取管理。

3. 使用CloudFront 控制台或 CloudFront API 更新您的發行版以引用默認根對象。

   若要使用 CloudFront 主控台指定預設根物件：

   a. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
   b. 在上方窗格中的分佈清單裡，請選取分佈來更新。
   c. 在 Settings (設定) 窗格中的 General (一般) 標籤，選擇 Edit (編輯)。
   d. 在 Edit settings (編輯設定) 對話方塊的 Default root object (預設根物件) 欄位中，輸入預設根物件的檔案名稱。

      只輸入物件名稱，例如，`index.html`。不要在物件名稱前新增 `/`。
   e. 選擇儲存變更。

   若要使用 CloudFront API 更新組態，您可以為發行版中的`DefaultRootObject`元素指定值。如需使用 CloudFront API 指定預設根物件的相關資訊，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。
4. 確認您已藉由請求根 URL 來啟用預設根物件。如果您的瀏覽器不會顯示預設根物件，請執行以下步驟：

   a. 透過在 CloudFront 主控台中檢視發行版的狀態，確認您的發行版已完全部署。
   b. 重複步驟 2 和 3 來驗證您是否已授予正確的許可和驗證您是否正確地更新分佈的組態來指定預設根物件。

預設根物件的運作方式

假設以下請求指向物件 `image.jpg`：

```
https://d111111abcdef8.cloudfront.net/image.jpg
```

相對地，以下請求指向同一分佈的根 URL，而非如前述範例指向特定物件：

```
https://d111111abcdef8.cloudfront.net/
```

當您定義預設根物件時，呼叫分佈根的最終使用者請求會傳回預設的根物件。例如，如果您指定檔案 `index.html` 為預設根物件，則請求：

```
https://d111111abcdef8.cloudfront.net/
```

傳回：

```
https://d111111abcdef8.cloudfront.net/index.html
```

> **Note**
>
> CloudFront 不確定具有多個尾隨斜線 (`https://d111111abcdef8.cloudfront.net///`) 的 URL 是否等於`https://d111111abcdef8.cloudfront.net/`。您的原始伺服器會進行這項比較。

如果您定義預設根物件時，適用於分佈子目錄的最終使用者請求不會傳回預設的根物件。例如，假設`index.html`是您的預設根物件，且會 CloudFront 接收發 CloudFront佈下`install`目錄的最終使用者要求：

```
https://d111111abcdef8.cloudfront.net/install/
```

CloudFront 即使目錄中`index.html`出現的複本，也不會傳回預設的根物`install`件。

如果您將發行版設定為允許所有 CloudFront 支援的 HTTP 方法，則預設根物件會套用至所有方法。例如，如果您的預設根物件是 index.php，而您撰寫應用程式以將`POST`要求提交至網域的根目錄 (https://example.com)，則會將要求 CloudFront 傳送至 https://example.com/index.php。

CloudFront 預設根物件的行為與 Amazon S3 索引文件的行為不同。當您配置 Amazon S3 儲存貯體做為網站，並指定索引文件時，Amazon S3 傳回索引文件，即使使用者在儲存貯體中請求子目錄。(索引文件的副本必須出現在每個子目錄)。有關將 Amazon S3 儲存貯體設定為網站以及索引文件的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 AmazonS3 上的託管網站一章。

> **Important**
>
> 請記住，預設根物件僅適用於您的 CloudFront 發佈。您仍然需要為原始伺服器管理安全性。例如，如果您使用 Amazon S3 原始伺服器，您仍然需要設定適當的 Amazon S3 儲存貯體 ACL，以確保在儲存貯體上您想要的存取層級。

如果您沒有定義根對象，該如何 CloudFront 工作

如果您不定義預設根物件，則分佈根的請求會通過您的原始伺服器。如果您使用 Amazon S3，有可能傳回以下任何情況：

- Amazon S3 儲存貯體的內容清單 — 在下列任何一種情況下，使用存取分發的任何人都可以看 CloudFront 到您的來源內容：
  - 您的儲存貯體未設定正確。
  - 儲存貯體上的 Amazon S3 許可與您的分佈相關連，且在儲存貯體中物件上把存取權限給予每個人。
  - 最終使用者使用原始伺服器根 URL 存取原始伺服器。
- 來源的私有內容清單 — 如果您將來源設定為私有分發 (只有您並 CloudFront 有權存取)，任何擁有登入資料可透過存取分發的人，都可以看到與您分發相關聯的 Amazon S3 儲存貯體的內容 CloudFront。在這種情況下，使用者無法透過您的原始伺服器根 URL 來存取您的內容。如需有關分佈私有內容的詳細資訊，請參閱the section called “使用已簽署的 URL 和已簽署的 Cookie 來限制內容”。
- `Error 403 Forbidden`— 如果與您的分發相關聯的 Amazon S3 儲存貯體上的許可或該儲存貯體中物件的許可拒絕對所有人的存取，則會CloudFront 傳回 CloudFront 此錯誤。

使檔案失效

如果您需要在檔案到期前將檔案從 CloudFront Edge 快取移除，可以執行下列其中一項作業：

- 透過節點快取使該檔案失效。下次檢視者要求檔案時，會 CloudFront 返回原始檔案以擷取最新版本的檔案。
- 使用檔案版本控制以提供該檔案具有不同名稱的不同版本。如需詳細資訊，請參閱 使用版本控制的檔案名稱更新現有檔案。

主題

在無效文件和使用版本化文件名之間進行選擇

若要控制從分佈中提供的檔案版本，您可使檔案失效或以版本控制的檔案名稱為其命名。如果您想經常更新檔案，建議您主要使用檔案版本控制，原因如下：

- 版本控制可讓您控制請求傳回哪些檔案，甚至是使用者什麼時候在本地或在企業快取代理之後快取一個版本。如果您使該檔案失效，使用者可能會繼續看到舊版本，直到檔案從這些快取中過期。
- CloudFront 存取記錄包含檔案的名稱，因此版本控制可讓您更輕鬆地分析檔案變更的結果。
- 版本控制讓您可向不同的使用者提供不同版本的檔案。
- 版本控制簡化了檔案修訂版本之間的往返復原。
- 版本控制更便宜。您仍然需要付費 CloudFront 才能將新版本的文件傳輸到節點，但是您不必為無效的文件付費。

如需檔案版本控制的詳細資訊，請參閱使用版本控制的檔案名稱更新現有檔案。

決定要使哪些檔案無效

若您要使多個檔案失效，例如某個目錄中的所有檔案，或者檔名以相同字元開頭的所有檔案，則可在失效路徑的結尾包含 * 萬用字元。如需有關使用 * 萬用字元的詳細資訊，請參閱Invalidation paths。

若要使檔案失效，您可以指定個別檔案的路徑或以 * 萬用字元結尾的路徑，其可能會套用到一個或多個檔案，如以下範例所示：

- `/images/image1.jpg`
- `/images/image*`
- `/images/*`

如果您想要使選取的檔案無效，但使用者不一定要存取原始檔案上的每個檔案，您可以判斷檢視者向哪些檔案請求，CloudFront 並僅使這些檔案失效。若要判斷檢視者已要求哪些檔案，請啟用 CloudFront 存取記錄。如需存取日誌的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

指定要使其無效的檔案

當您指定要使檔案無效時，請參閱下列資訊：

區分大小寫

無效驗證路徑是區分大小寫的。例如，`/images/image.jpg`並`/images/Image.jpg`指定兩個不同的檔案。

使用 Lambda 函數變更 URI

如果您的發行 CloudFront 版在檢視器要求事件上觸發 Lambda 函數，並且該函數變更了要求檔案的 URI，建議您將這兩個 URI 無效，以便從 CloudFront邊緣快取中移除檔案：

- 在檢視器請求中的 URI
- 函數予以變更之後的 URI

Example 範例

假設您的 Lambda 函數從以下位置更改文件的 URI：

`https://d111111abcdef8.cloudfront.net/index.html`

對於包含語言目錄的 URI：

`https://d111111abcdef8.cloudfront.net/en/index.html`

若要使該檔案失效，您必須指定以下路徑：

- `/index.html`
- `/en/index.html`

如需詳細資訊，請參閱 Invalidation paths。

預設根物件

若要使預設根物件 (檔案) 失效，請以您為任何其他檔案指定路徑的相同方式指定其路徑。如需詳細資訊，請參閱 預設根物件的運作方式。

轉送 Cookie

如果您設定 CloudFront 將 Cookie 轉寄至您的來源， CloudFront 邊緣快取可能包含檔案的多個版本。當您使檔案無效時，會使檔案的每個快取版本 CloudFront 無效，而不論其關聯的 Cookie 為何。您無法根據關聯的 Cookie 選擇性地使某些版本而不是其他版本失效。如需詳細資訊，請參閱 根據 Cookie 快取內容。

Forwarding headers (轉送標頭)

如果您設定 CloudFront 將標頭清單轉寄至來源，並根據標頭的值進行快取， CloudFront Edge 快取可能會包含檔案的數個版本。當您使檔案無 CloudFront效時，無論標頭值為何，都會使檔案的

每個快取版本無效。您無法根據標頭值選擇性地使某些版本而不是其他版本失效。（如果您配置 CloudFront 為將所有標頭轉發到您的來源，則 CloudFront 不會緩存文件。）如需詳細資訊，請參閱 根據請求標頭快取內容。

轉送查詢字串

如果您設定 CloudFront 為將查詢字串轉寄至您的來源，則必須在使檔案無效時包含查詢字串，如下列範例所示：

- `/images/image.jpg?parameter1=a`
- `/images/image.jpg?parameter1=b`

如果用戶端請求包含針對同一檔案的五種不同查詢字串，則您可以使該檔案失效五次，每一查詢字串一次，或者也可在失效路徑中使用 * 萬用字元，如以下範例所示：

```
/images/image.jpg*
```

如需有關在失效路徑使用萬用字元的詳細資訊，請參閱Invalidation paths。

如需查詢字串的詳細資訊，請參閱根據查詢字串參數快取內容。

若要判斷哪些查詢字串正在使用中，您可以啟用 CloudFront記錄功能。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌)。

允許的上限

如需允許的最大無效驗證數目的詳細資訊，請參閱。並行失效請求上限

Microsoft Smooth Streaming file (Microsoft Smooth Streaming 檔案)

如果您已針對對應的快取行為啟用「平滑串流」，就無法使用 Microsoft 流暢串流格式的媒體檔案失效。

Non-ASCII or unsafe characters in the path (路徑中非 ASCII 或不安全字元)

如果路徑包含非 ASCII 字元或 RFC 1738 中定義的不安全字元，請將這些字元進行 URL 編碼。不要對路徑中的任何其他字符進行 URL 編碼，否 CloudFront 則不會使更新文件的舊版本無效。

Invalidation paths (失效路徑)

路徑與分佈有關。例如，若要使檔案失效`https://d111111abcdef8.cloudfront.net/images/image2.jpg`，您可以指`/images/image2.jpg`定。

> **Note**
>
> 在CloudFront控制台中，您可以省略路徑中的前導斜杠，如下所示：`images/image2.jpg`。當您直接使用 CloudFront API 時，無效路徑必須以前導斜線開頭。

您也可以使用 * 萬用字元使多個檔案同時失效。取代 0 或多個字元的 *，必須是失效路徑的最後一個字元。

如果您使用 AWS Command Line Interface (AWS CLI) 使檔案無效，並指定包含*萬用字元的路徑，則必須在路徑周圍使用引號 (")，例"/*"如。

Example 範例：無效驗證路徑

- 使目錄中的所有檔案無效：

  */directory-path/** (目錄路徑)
- 若要使目錄、其所有子目錄以及目錄和子目錄中的所有檔案無效：

  */directory-path** (目錄路徑)
- 為了使所有名稱相同，但不同的檔案名稱延伸的檔案失效，例如 logo.jpg、logo.png 和 logo.gif：

  /directory-path (目錄路徑) */file-name* (檔案名稱) **.***
- 無論檔案名稱延伸如何，若要使檔案名稱以相同字元開頭的目錄中的所有檔案（例如 HLS 格式中影片的所有檔案）失效：

  */##### / initial-characters-in-file**
- 當您配置 CloudFront 為基於查詢字符串參數緩存，並且想要使文件的每個版本無效時：

  /directory-path (目錄路徑) */file-name* (檔案名稱) *.file-name-extension**
- 要使發行版中的所有文件無效：

  /

路徑的長度上限為 4,000 個字元。您不能在路徑中使用萬用字元。它只能在路徑的末尾加入。

如需使用 Lambda 函式變更 URI 的情況下使檔案失效的詳細資訊，請參閱Changing the URI Using a Lambda Function。

如果失效路徑是一個目錄且如果您尚未標準化指定目錄的方法 (包含或不含結尾斜線 (/))，我們建議您，讓包含及不包含結尾斜線的目錄皆失效，例如 `/images` 和 `/images/`。

Signed URLs (簽署的 URL)

如果您使用已簽章的 URL，則透過僅包含問號 (?) 前面的 URL 部分以使檔案失效。

使檔案無效 (主控台)

您可以使用 CloudFront 主控台來建立和執行無效驗證、顯示先前提交的無效驗證清單，以及顯示有關個別無效驗證的詳細資訊。您也可以複製現有的失效、編輯檔案路徑的清單和執行已編輯的失效。您可以從清單移除失效。

內容

使檔案無效

若要使用 CloudFront 主控台使檔案無效，請執行下列動作。

使檔案無效 (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您欲使其檔案失效的分佈。
3. 請選擇 Invalidations (失效) 索引標籤。
4. 選擇「建立無效驗證」。
5. 針對您要使其失效的檔案，每行輸入一個失效路徑。如需有關指定失效路徑的詳細資訊，請參閱指定要使其無效的檔案。

   > **Important**
   >
   > 請謹慎指定檔案路徑。開始後便無法取消失效請求。

6. 選擇「建立無效驗證」。

複製、編輯和重新執行現有的無效驗證

您可以複製您之前建立的失效、更新失效路徑的清單，並執行已更新的失效。您無法複製現有的無效驗證、更新失效路徑，然後儲存更新的無效驗證而不執行它。

> **Important**
>
> 如果您複製仍在進行中的無效驗證，請更新無效驗證路徑清單，然後執行更新的無效驗證，CloudFront 不會停止或刪除您複製的失效驗證。如果有任何無效驗證路徑出現在原始路徑和副本中，CloudFront 將嘗試使檔案失效兩次，而這兩項無效的路徑都會計入您當月免費無效的最大數量。如果您已經達到免費無效驗證的最大數量，我們將會向您收取每個檔案的兩項無效驗證費用。如需詳細資訊，請參閱 並行失效請求上限。

複製、編輯和重新執行現有的失效

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 請選取包含您欲複製的失效的分佈。
3. 請選擇 Invalidations (失效) 索引標籤。
4. 選擇您想要複製的失效。

   如果您不確定要複製哪一種無效驗證，可以選擇無效驗證，然後選擇「檢視詳細資訊」，以顯示該失效的詳細資訊。
5. 選擇「複製到新的」。
6. 如果適用，請更新失效路徑清單。
7. 選擇「建立無效驗證」。

取消無效

當您將無效驗證要求提交給時 CloudFront，會在幾秒鐘內將要求 CloudFront 轉寄至所有節點，而且每個節點都會立即開始處理失效。因此，提交後便無法取消失效。

列表無效

您可以使用主控台顯示您為發行版建立和執行的最後 100 項無效驗證清單。 CloudFront 如果您想要取得超過 100 項無效的清單，請使用 `ListInvalidations` API 作業。如需詳細資訊，請參閱 Amazon CloudFront API 參考資料ListInvalidations中的。

列出失效

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 請選取欲顯示失效清單的分佈。
3. 請選擇 Invalidations (失效) 索引標籤。

> **Note**
>
> 您無法從清單移除失效。

顯示關於失效驗證的資訊

您可以顯示有關失效的詳細資訊，包括分佈 ID、失效 ID、失效的狀態、建立失效的日期和時間，以及失效路徑的完整清單。

顯示有關失效的資訊

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 請選取包含您欲顯示詳細資訊的失效的分佈。
3. 請選擇 Invalidations (失效) 索引標籤。
4. 選擇適用的無效驗證 ID 或選取無效驗證 ID，然後選擇「檢視詳細資訊」。

使檔案無效 (CloudFront API)

如需有關使物件無效並顯示無效驗證相關資訊的詳細資訊，請參閱 Amazon API 參考中的以下主題：CloudFront

- CreateInvalidation
- ListInvalidations
- GetInvalidation

> **Note**
>
> 如果您使用 AWS Command Line Interface (AWS CLI) 使檔案無效，並指定包含*萬用字元的路徑，則必須在路徑周圍使用引號 (")，例如下列範例：
>
> ```
> aws cloudfront create-invalidation --distribution-id distribution_ID --paths "/*"
> ```

並行失效請求上限

如果您使檔案逐個失效，則每個進行中的分佈可同時請求最多 3,000 個檔案失效。一次失效請求最多可達 3,000 個檔案，或針對單一檔案請求 3,000 次，或者總計不超過 3,000 個檔案的任何組合方式。例如，您可以提交 30 次失效請求，每次使 100 個檔案失效。只要 30 個失效請求都仍在進行中，您就無法提交任何更多的失效請求。如果超過最大值，則會 CloudFront 傳回錯誤訊息。

如果您使用 * 萬用字元，則每次進行中都可以請求最多 15 個失效路徑。您也可以就每一進行中的分佈同時請求失效最多 3,000 個檔案；萬用字元失效請求允許的上限與檔案逐個失效的上限無關。

支付檔案失效

您每月提交的前 1,000 個失效路徑免費；您只需支付每月超過 1,000 個的每一失效路徑。每一失效路徑可適用於單一檔案 (如 `/images/logo.jpg`) 或多個檔案 (如 `/images/*`)。包含*萬用字元的路徑會計為一個路徑，即使它會造 CloudFront 成數千個檔案失效。

每月 1,000 個免費失效路徑的上限適用於您建立一個 AWS 帳戶中所有分佈的失效路徑總數。例如，如果您使用 AWS 帳戶 `john@example.com`來建立三個分配，並針對指定月份中的每個分配提交 600 個無效驗證路徑 (總共 1,800 個無效驗證路徑)，則該月 AWS 會向您收取 800 個無效驗證路徑的費用。

無論您要使之失效的檔案有多少，單一檔案 (`/images/logo.jpg`) 還是所有與分佈關聯的檔案 (`/*`)，提交失效路徑的費用皆相同。因為無效驗證要求中的每個路徑都會向您收費，因此即使您將多個路徑併入單一要求中，每個路徑仍會針對計費目的個別計算。

如需無效驗證定價的詳細資訊，請參閱 Amazon CloudFront 定價。如需有關失效路徑的詳細資訊，請參閱Invalidation paths。

提供壓縮檔案

您可 CloudFront 以使用自動壓縮某些類型的物件 (檔案)，並在檢視器 (網頁瀏覽器或其他用戶端) 支援壓縮物件時提供壓縮物件。如果檢視器具有 `Accept-Encoding` HTTP 標頭，表示他們支援壓縮物件。

CloudFront 可以使用 Gzip 和 Brotli 壓縮格式壓縮物件。當查看器支持兩種格式，並且兩者都存在於到達的緩存服務器中時，則 CloudFront 更喜歡 Brotli。如果快取伺服器中只有一種壓縮格式，則 CloudFront 傳回它。

> **Note**
>
> 只有在使用 HTTPS 發送請求時，Chrome 和 Firefox 網頁瀏覽器才支援 Brotli 壓縮。這些瀏覽器不支援使用 HTTP 請求的 Brotli。

請求的物件壓縮後物件較小，下載更快 – 在某些情況下，壓縮後還不到原始檔案的四分之一大小。特別是對於JavaScript 和 CSS 文件，更快的下載可以導致更快的網頁呈現為您的用戶。此外，由於 CloudFront 資料傳輸的成本是根據所提供的資料總量計算，因此提供壓縮物件的成本可能會比未壓縮的物件提供服務便宜。

某些自訂原始伺服器也可以壓縮物件。您的來源可能可以壓縮 CloudFront 未壓縮的物件 (請參閱 CloudFront 壓縮的檔案類型)。如果您的 origin 將壓縮物件傳回至 CloudFront，則會 CloudFront 偵測該物件是否根據`Content-Encoding`標頭的存在進行壓縮，而且不會再次壓縮物件。

配置 CloudFront 壓縮物件

若 CloudFront 要設定為壓縮物件，請執行下列所有動作，更新您要提供壓縮物件的快取行為：

1. 確定 Compress Objects Automatically (自動壓縮物件) 設定為 Yes (是)。(在AWS CloudFormation 或 CloudFront API 中，設定`Compress`為`true`。)
2. 使用快取政策來指定快取設定，並確保 Gzip 和 Brotli 設定皆已啟用。(在 AWS CloudFormation 或 CloudFront API 中，將 `EnableAcceptEncodingGzip` 和 `EnableAcceptEncodingBrotli` 設為 `true`。)
3. 請確定快取政策中的 TTL 值已設為大於零的值。當您將 TTL 值設定為零時，會停用快取，而且 CloudFront 不會壓縮物件。

若要更新快取行為，您可以使用下列任何工具：

- CloudFront 控制台
- AWS CloudFormation
- AWS 軟體開發套件和命令列工具

CloudFront 壓縮的工作原理

當您設定 CloudFront 為壓縮物件 (請參閱上一節) 時，它的運作方式如下：

1. 一個檢視器請求了一個物件。檢視器會在請求中包含 `Accept-Encoding` HTTP 標頭，而標頭值則包含 `gzip`、`br` 或兩者。這表示檢視器支援壓縮物件。當查看器同時支持 Gzip 和布羅特利時，更喜歡布羅特利 CloudFront 。

   > **Note**
   >
   > 只有在使用 HTTPS 發送請求時，Chrome 和 Firefox 網頁瀏覽器才支援 Brotli 壓縮。這些瀏覽器不支援使用 HTTP 請求的 Brotli。

2. 在邊緣位置， CloudFront 檢查緩存中是否有請求對象的壓縮副本。
3. 如果壓縮的物件已經在快取中，請 CloudFront 將它傳送給檢視器，並略過剩下的步驟。

   如果壓縮的物件不在快取中，請將要求 CloudFront 轉寄至來源。

   > **Note**
   >
   > 如果物件的未壓縮副本已經在快取中， CloudFront 可能會將它傳送給檢視器，而不需要將要求轉送至原始位置。例如， CloudFront 先前略過壓縮時可能會發生這種情況。發生這種情況時，會 CloudFront 快取未壓縮的物件並繼續提供服務，直到物件過期、收回或無效為止。

4. 如果來源傳回壓縮物件 (如 HTTP 回應中存在`Content-Encoding`標頭所指出)，則會將壓縮的物件 CloudFront 傳送給檢視器、將它新增至快取，然後略過剩餘步驟。 CloudFront 不會再壓縮物件。

   如果 origin 將未壓縮的對象返回到 CloudFront （ HTTP 響應中沒有`Content-Encoding`標題），則 CloudFront 確定該對象是否可壓縮。如需有關如何 CloudFront判斷物件是否可壓縮的詳細資訊，請參閱下一節。

5. 如果該對象是可壓縮的，那么請將其 CloudFront 壓縮，將其發送到查看器，然后將其添加到緩存中。（在極少數情況下， CloudFront 可能會跳過壓縮並將未壓縮的對象發送給查看器。）

CloudFront 壓縮的注意事項

下列清單提供有關何時 CloudFront 壓縮物件的詳細資訊。

請求使用 HTTP 1.0

如果請求 CloudFront 使用 HTTP 1.0，則 CloudFront 刪除標`Accept-Encoding`頭並且不壓縮響應中的對象。

`Accept-Encoding` 請求標頭

如果查看器請求中缺少`Accept-Encoding`標頭，或者它不包含`gzip`或`br`作為值，則 CloudFront 不會在響應中壓縮對象。如果標`Accept-Encoding`頭包含其他值，例如`deflate`，請在將請求轉發到原點之前將其 CloudFront 刪除。

當設定 CloudFront 為壓縮物件時，它會在快取金鑰和原始要求中自動包含`Accept-Encoding`標頭。

動態內容

CloudFront 並不總是壓縮動態內容。有時候動態內容的回應會被壓縮，有時候則不會被壓縮。

當您設定 CloudFront 為壓縮物件時，內容已快取

CloudFront 當物件從原點取得物件時，會壓縮物件。當您設定 CloudFront 為壓縮物件時，CloudFront 不會壓縮已在邊緣位置快取的物件。此外，當快取的物件在邊緣位置過期，並將物件的另一個要求 CloudFront 轉送至您的來源時，當您的 origin 傳回 HTTP 狀態碼 304 時，CloudFront 不會壓縮物件，也就是說邊緣位置已經擁有該物件的最新版本。如果您 CloudFront 要壓縮已快取在邊緣位置的物件，則需要使這些物件無效。如需詳細資訊，請參閱 使檔案失效。

已配置原始伺服器來壓縮物件

如果您設定 CloudFront 為壓縮物件，而原點也會壓縮物件，則原點應包含`Content-Encoding`標頭，表示物件已經壓縮。 CloudFront 當來自來源的響應包含標`Content-Encoding`題時，CloudFront 不會壓縮對象，而不管標題的值如何。 CloudFront 將回應傳送給檢視器，並將物件快取至節點位置。

CloudFront 壓縮的檔案類型

如需 CloudFront 壓縮檔案類型的完整清單，請參閱 CloudFront 壓縮的檔案類型。

CloudFront 壓縮物件的大小

CloudFront 壓縮大小介於 1,000 個位元組到 1 萬位元組之間的物件。

`Content-Length` 標頭

原點必須在響應中包含一個`Content-Length`標頭，該標題 CloudFront用於確定對象的大小是否在 CloudFront壓縮的範圍內。如果標`Content-Length`頭遺失、包含無效值，或包含超出壓縮大小範圍的值，則 CloudFront 不會壓縮物件。 CloudFront

回應的 HTTP 狀態碼

CloudFront 只有當回應的 HTTP 狀態碼為、或`404`時`200`，`403`才會壓縮物件。

回應沒有內文

當來源的 HTTP 響應沒有主體時，沒有任何壓縮 CloudFront 的內容。

`ETag` 標頭

CloudFront 壓縮物件時，有時會修改 HTTP 回應中的`ETag`標頭。如需詳細資訊，請參閱 the section called “ETag 標頭轉換”。

CloudFront 跳過壓縮

CloudFront 以最大努力的方式壓縮物件。在極少數情況下，會 CloudFront 略過壓縮。CloudFront 根據各種因素 (包括主機容量) 做出此決定。如果 CloudFront 略過某個物件的壓縮，它會快取未壓縮的物件，並繼續提供給檢視者，直到物件過期、撤銷或失效為止。

CloudFront 壓縮的檔案類型

如果您設定 CloudFront 為壓縮物件，則 CloudFront 只會壓縮`Content-Type`回應標頭中具有下列其中一個值的物件：

- `application/dash+xml`
- `application/eot`
- `application/font`
- `application/font-sfnt`
- `application/javascript`
- `application/json`
- `application/opentype`
- `application/otf`
- `application/pdf`
- `application/pkcs7-mime`

- application/protobuf
- application/rss+xml
- application/truetype
- application/ttf
- application/vnd.apple.mpegurl
- application/vnd.mapbox-vector-tile
- application/vnd.ms-fontobject
- application/wasm
- application/xhtml+xml
- application/xml
- application/x-font-opentype
- application/x-font-truetype
- application/x-font-ttf
- application/x-httpd-cgi
- application/x-javascript
- application/x-mpegurl
- application/x-opentype
- application/x-otf
- application/x-perl
- application/x-ttf
- font/eot
- font/opentype
- font/otf
- font/ttf
- image/svg+xml
- text/css
- text/csv
- text/html
- text/javascript
- text/js

- text/plain
- text/richtext
- text/tab-separated-values
- text/xml
- text/x-component
- text/x-java-source
- text/x-script
- vnd.apple.mpegurl

ETag 標頭轉換

當來自來源的未壓縮物件包含有效的強式 ETag HTTP 標頭並 CloudFront 壓縮物件時， CloudFront 也會將強標ETag頭值轉換為 weakETag，並將弱ETag值傳回給檢視器。檢視器可以儲存弱值 ETag，並使用它來傳送具有 If-None-Match HTTP 標頭的條件式請求。這可讓檢視者和來源將物件的壓縮和未壓縮版本視為語意等效，進而減少不必要的資料傳輸。 CloudFront

有效的強式 ETag 標頭值以雙引號字元 (") 開頭。要將強ETag值轉換為弱值，請將字符 CloudFront 添加W/到強ETag值的開頭。

當來自 origin 的物件包含弱ETag標頭值 (以字元開頭的值W/) 時， CloudFront 不會修改此值，並將它傳回給檢視器，如同從原點接收的那樣。

當來自 origin 的物件包含無效的ETag標頭值 (該值開頭不是"或以W/) 時，會 CloudFront 移除ETag標頭並將物件傳回給檢視器，而不含回ETag應標頭。

如需詳細資訊，請參閱 MDN Web 文件中的下列頁面：

- 指示詞 (ETag HTTP 標頭)
- 弱驗證 (HTTP 條件式請求)
- If-None-Match HTTP 標頭

產生自訂錯誤回應

如果您正在通過 CloudFront 某個對象由於某種原因無法使用，則您的 Web 服務器通常會返回相關的 HTTP 狀態代碼 CloudFront 來指示這一點。例如，如果檢視者要求無效的 URL，您的 Web 伺服器會傳回 HTTP 404 (找不到) 狀態碼 CloudFront，並將該狀態碼 CloudFront 傳回給檢視者。

如果您願意，您可 CloudFront 以配置為將自定義錯誤響應返回給查看器。您也有數個選項可用來管理發生錯誤時的 CloudFront 回應方式。若要指定自訂錯誤訊息的選項，請更新 CloudFront 發佈以指定這些值。如需詳細資訊，請參閱 設定錯誤回應行為。

如果您設定 CloudFront 為傳回 HTTP 狀態碼的自訂錯誤頁面，但是自訂錯誤頁面無法使用，則會將從包含自訂錯誤頁面的來源 CloudFront 接收到的狀態碼 CloudFront 傳回給檢視器。例如，假設您的自訂來源傳回 500 狀態碼，而且您已設定 CloudFront 為從 Amazon S3 儲存貯體取得 500 個狀態碼的自訂錯誤頁面。不過，有人不小心從您的值區刪除了自訂錯誤頁面。 CloudFront 會傳回 HTTP 404 狀態碼 (未找到) 給要求物件的檢視器。

將自訂錯誤頁面 CloudFront 傳回給檢視器時，您需要支付自訂錯誤頁面的標準 CloudFront 費用，而不是要求物件的費用。如需有關CloudFront 費用的詳細資訊，請參閱 Amazon CloudFront 定價。

主題

設定錯誤回應行為

若要設定自訂錯誤回應，您可以使用 CloudFront 主控台、 CloudFront API 或 AWS CloudFormation。無論您選擇如何更新組態，請考慮下列提示和建議：

- 將您的自訂錯誤頁面儲存在可存取的位置 CloudFront。我們建議您將它們存放在 Amazon S3 儲存貯體中，並且不要將它們儲存在與網站或應用程式內容之其餘部分相同的位置。如果您將自訂錯誤頁面儲存在與您的網站或應用程式相同的原始位置，且來源開始傳回 5xx 錯誤，則 CloudFront 無法取得自訂錯誤頁面，因為原始伺服器無法使用。如需詳細資訊，請參閱 將物件和自訂錯誤頁面存放在不同位置。
- 確保 CloudFront 具有獲取自定義錯誤頁面的權限。如果自訂錯誤頁面存放在 Amazon S3 中，頁面必須可公開存取，或者您必須設定 CloudFront 來源存取控制 (OAC)。如果自訂錯誤頁面存放在自訂原始伺服器中，則必須可以公開存取頁面。
- (選用) 如果需要，請設定您的原始伺服器以新增 `Cache-Control` 或 `Expires` 標頭以及自訂錯誤頁面。您也可以使用「錯誤快取下限 TTL」設定來控制自訂錯誤頁面 CloudFront 快取的時間長度。如需詳細資訊，請參閱 控制 CloudFront 快取錯誤的時間長度。

設定自訂錯誤回應 (CloudFront主控台)

要在 CloudFront 控制台中配置自定義錯誤響應，您必須具有 CloudFront分發。在主控台中，自訂錯誤回應的組態設定僅適用於現有分發。若要了解如何建立分發，請參閱開始使用基本 CloudFront發行版。

若要設定自訂錯誤回應 (主控台)

1. 登入 AWS Management Console 並在 CloudFront 主控台中開啟 [發行版] 頁面，位於https://console.aws.amazon.com/cloudfront/v4/home#distributions。
2. 在分發清單中，選擇要更新的分發。
3. 選擇 Error Pages (錯誤頁面) 標籤，然後選擇 Create Custom Error Response (建立自訂錯誤回應)。
4. 輸入適用的值。如需詳細資訊，請參閱 自訂錯誤頁面和錯誤快取。
5. 輸入所需的值後，選擇 Create (建立)。

設定自訂錯誤回應 (CloudFrontAPI 或 AWS CloudFormation)

若要使用 CloudFront API 設定自訂錯誤回應 AWS CloudFormation，或使用散發中的`CustomErrorResponse`類型。如需詳細資訊，請參閱下列內容：

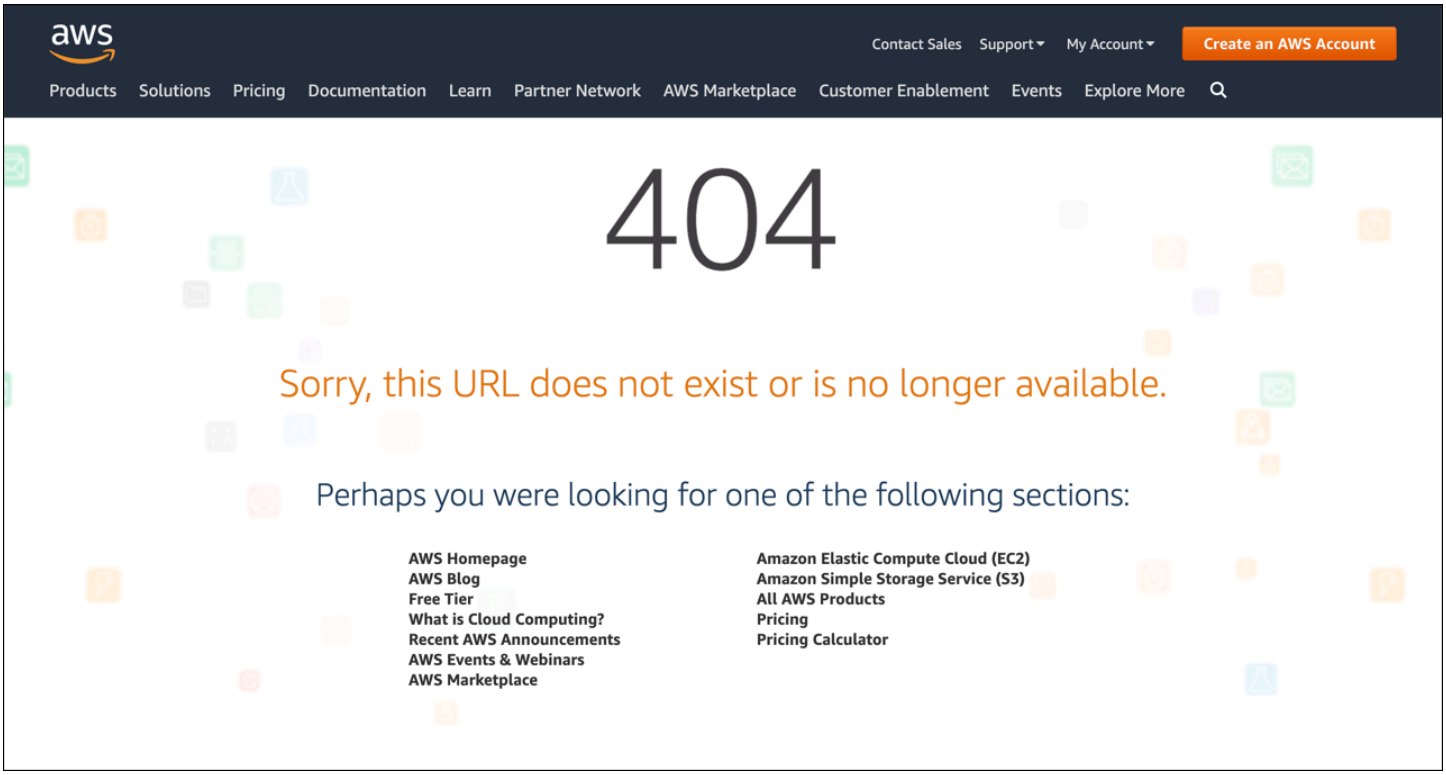
- AWS::CloudFront::Distribution CustomErrorResponse《AWS CloudFormation 使用者指南》中的
- CustomErrorResponse在 Amazon CloudFront API 參考

針對特定的 HTTP 狀態碼建立自訂錯誤頁面

如果您想要顯示自訂錯誤訊息而不是預設訊息 (例如，使用與網站其他部分相同格式的網頁)，您可以將包含您自訂錯誤訊息的物件 (例如 HTML 檔案) CloudFront 傳回給檢視者。

若要指定您要傳回的檔案以及應傳回檔案的錯誤，請更新您的 CloudFront 發行版以指定這些值。如需詳細資訊，請參閱 設定錯誤回應行為。

例如，以下是自訂錯誤頁面：



您可以為每個支援的 HTTP 狀態碼指定不同的物件，或您可以為所有支援的狀態碼使用相同的物件。您可以選擇為某些狀態碼指定自訂錯誤頁面，而不是為其他狀態碼指定。

您通過提供的對象 CloudFront 可能由於各種原因而無法使用。這些分為兩大類：

- 用戶端錯誤顯示請求有問題。例如，有指定名稱的物件不可用，或使用者沒有在 Amazon S3 儲存貯體中取得物件所需的許可。發生用戶端錯誤時，來源會將 4xx 範圍內的 HTTP 狀態碼傳回至 CloudFront。
- 伺服器錯誤顯示原始伺服器有問題。例如，HTTP 伺服器是忙碌或不可用。發生伺服器錯誤時，原始伺服器會傳回 5xx 範圍內的 HTTP 狀態碼 CloudFront，或在一段時間內 CloudFront 未收到來源伺服器的回應，並假設 504 狀態碼 (閘道逾時)。

CloudFront 可以傳回自訂錯誤頁面的 HTTP 狀態碼包括下列項目：

- 400, 403, 404, 405, 414, 416

> **備註**
>
> - 如果 CloudFront 偵測到要求可能不安全，則會 CloudFront 傳回 400 (錯誤要求) 錯誤，而非自訂錯誤頁面。

  - 您可以為 HTTP 狀態碼 416 (要求的範圍不符合) 建立自訂錯誤頁面，也可以變更原始碼 416 CloudFront 傳回狀態碼時傳回給檢視者的 HTTP 狀態碼。 CloudFront(如需詳細資訊，請參閱 變更傳回的回應碼 CloudFront。) 不過，CloudFront 不會快取狀態碼 416 回應，因此即使您指定狀態碼 416 的 [錯誤快取最小 TTL] 值，也 CloudFront 不會使用它。

- 500、501、502、503、504

> **Note**
>
> 在某些情況下，即使您設 CloudFront 定這麼做，也 CloudFront 不會傳回 HTTP 503 狀態碼的自訂錯誤頁面。如果 CloudFront 錯誤碼為`Capacity Exceeded`或`Limit Exceeded`，則會將 503 狀態碼 CloudFront 傳回給檢視器，而不使用自訂錯誤頁面。

有關如何 CloudFront 處理來源錯誤響應的詳細說明，請參閱如何從您的來源 CloudFront 處理和緩存 HTTP 4xx 和 5xx 狀態碼。

將物件和自訂錯誤頁面存放在不同位置

如果您要在不同位置存放物件和自訂錯誤頁面，則您的分佈必須包含下列為屬實的快取行為：

- Path Pattern (路徑模式) 的值與自訂錯誤訊息的路徑相符。例如，假設您已在 Amazon S3 儲存貯體名為 `/4xx-errors` 的目錄中儲存了 4xx 錯誤的自訂錯誤頁面。您的分佈必須包含路徑模式路由自訂錯誤頁面請求至該位置的快取行為，例如 `/4xx-errors/*`。
- Origin (原始伺服器) 的數值將 Origin ID (原始伺服器 ID) 的數值指定給包含自訂錯誤頁面的原始伺服器。

如需詳細資訊，請參閱 快取行為設定。

變更傳回的回應碼 CloudFront

您可以設定 CloudFront 為將不同於從來源 CloudFront接收到的 HTTP 狀態碼傳回給檢視器。例如，如果您的來源傳回 500 狀態碼至 CloudFront，您可能想 CloudFront 要將自訂錯誤頁面和 200 狀態碼 (OK) 傳回給檢視器。您可能希望 CloudFront 將狀態碼傳回給檢視器的原因有很多種，這與原因不同於原始伺服器返回的狀態碼 CloudFront：

- 有些網際網路裝置 (例如，一些防火牆和公司代理) 會攔截 HTTP 4xx 和 5xx 狀態碼並禁止回應傳回給檢視器。在此案例中，如果您替換 `200`，回應不會攔截。

- 如果您不關心區分不同的客戶端錯誤或服務器錯誤，則可以指定400或500作為 CloudFront返回所有 4xx 或 5xx 狀態碼的值。
- 您會希望傳回 200 狀態碼 (OK) 與靜態網站，如此您的客戶便不會知道您網站已關閉。

如果您啟用CloudFront 標準記錄檔，並設定 CloudFront為變更回應中的 HTTP 狀態碼，記錄檔中的sc-status欄值會包含您指定的狀態碼。不過，x-edge-result-type 資料行的值不會受到影響。它包含來自原始伺服器回應的結果類型。例如，假設您設定 CloudFront 為在原始位置傳回 404 (未找到) 時，200將的狀態碼傳回給檢視器 CloudFront。當原始伺服器回應 404 狀態碼的請求時，在日誌中 sc-status 資料行的值會是 200，但 x-edge-result-type 資料行的值則會是 Error。

您可以設定 CloudFront 為傳回下列任何 HTTP 狀態碼以及自訂錯誤頁面：

- 200
- 400, 403, 404, 405, 414, 416
- 500、501、502、503、504

控制 CloudFront 快取錯誤的時間長度

CloudFront 快取錯誤回應，預設持續時間為 10 秒。 CloudFront 然後將對象的下一個請求提交給您的來源，以查看導致錯誤的問題是否已解決，並且所請求的對象是否可用。

您可以為每個快取的 4xx 和 5xx 狀態碼指定錯誤快取持續時間 (錯誤快取下限 TTL)。 CloudFront (如需詳細資訊，請參閱 可快取的狀態碼 CloudFront。) 當您指定持續時間時，請注意以下事項：

- 如果您指定的錯誤快取持續時間較短，則會將更多要求 CloudFront 轉送至來源，而不是指定的持續時間較長。針對 5xx 錯誤，這可能會讓原本導致原始伺服器傳回錯誤的問題加劇。
- 當您的來源傳回物件的錯誤時，會以錯誤 CloudFront 回應或自訂錯誤頁面回應物件的要求，直到錯誤快取持續時間過去為止。如果您指定較長的錯誤快取持續時間， CloudFront 可能會在物件再次變為可用後很長一段時間內，以錯誤回應或您的自訂錯誤頁面繼續回應要求。

> **Note**
>
> 您可以為 HTTP 狀態碼 416 (要求的範圍不符合) 建立自訂錯誤頁面，也可以變更原始碼 416 CloudFront 傳回狀態碼時傳回給檢視者的 HTTP 狀態碼。 CloudFront(如需詳細資訊，請參閱變更傳回的回應碼 CloudFront。) 不過， CloudFront 不會快取狀態碼 416 回應，因此即使您指定狀態碼 416 的 [錯誤快取最小 TTL] 值，也 CloudFront不會使用它。

如果您想要控制個別物件 CloudFront 快取錯誤的時間長度，您可以設定原始伺服器，將適用的標頭新增至該物件的錯誤回應。

如果來源新增`Cache-Control: max-age`或`Cache-Control: s-maxage`指示詞或`Expires`標頭，會 CloudFront 快取標頭中較大值或錯誤快取下限 TTL 的錯誤回應。

> **Note**
>
> `Cache-Control: max-age` 和 `Cache-Control: s-maxage` 值不得大於為擷取的錯誤頁面所設定之快取行為的 Maximum TTL (最大 TTL) 值。

如果來源新增其他`Cache-Control`指示詞或未新增標頭，請CloudFront 快取錯誤回應的錯誤快取下限 TTL 值。

如果物件的 4xx 或 5xx 狀態碼的到期時間超過您想要的時間，且物件可以再次使用，則您可以使用請求的物件 URL 使快取的錯誤碼失效。如果原始伺服器傳回多個物件的錯誤回應，則您需要個別使每一個物件失效。如需有關使物件失效的詳細資訊，請參閱使檔案失效。

使用 AWS WAF 保護

您可以用AWS WAF來保護您的 CloudFront 發行版和原始伺服器。 AWS WAF 是一種 Web 應用程式防火牆，可在要求到達您的伺服器之前封鎖要求，以協助保護 Web 應用程式和 API 的安全 如需詳細資訊，請參閱使用 CloudFront 和加速和保護您的網站 AWS WAF。

若要啟用 AWS WAF 保護，您可以：

- 在 CloudFront 主控台中使用一鍵式保護。一鍵式防護可建立 AWS WAF Web 存取控制清單 (Web ACL)、設定規則以保護伺服器免於遭受常見網路安全威脅的侵害，以及為您附加 Web ACL 至 CloudFront 散佈。本節中的主題假設使用一鍵式保護。
- 使用您在 AWS WAF 主控台或使用 AWS WAF API 建立的預先設定 Web ACL (存取控制清單)。如需詳細資訊，請參閱AWS WAF 開發人員指南中的 Web 存取控制清單 (AssociateWebACL) 和 AWS WAF API 參考資料中的 ACL

您可以在以下 AWS WAF 情況啟用：

- 建立分發
- 使用安全性儀表板編輯現有分佈的安全性設定

當您使用一鍵式保護時，會 CloudFront 套用一組 AWS 建議的保護：

- 根據 Amazon 內部威脅情報，封鎖 IP 地址免於遭受潛在威脅。
- 如 OWASP 前 10 所述，防範 Web 應用程式中發現的最常見漏洞。
- 抵禦發現應用程式漏洞的惡意行為者。

> **Important**
>
> AWS WAF 如果您想要在「 CloudFront安全性」儀表板中檢視安全指標，則必須啟用此功能。如果未啟用 AWS WAF，則您只能使用 [安全性] 儀表板來啟用 AWS WAF 或設定 CloudFront 地理限制。如需更多儀表版的相關資訊，請參閱此章節後續的 使用 CloudFront 安全儀表板。

主題

- AWS WAF 為分配啟用
- 停用 AWS WAF 安全性保護
- 設定速率限制
- 使用 CloudFront 安全儀表板

AWS WAF 為分配啟用

建立發佈時，您可以啟用 AWS WAF 和使用現有的 ACL。

若要為新 AWS WAF 的發行版啟用

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [分配]，然後選擇 [建立分配]。
3. 如有需要，請遵循 建立分發 中的步驟。
4. 在「Web 應用程式防火牆」區段中，選擇 「編輯」，然後選擇「啟動安全保護」。
5. 完成下列欄位：
   - [使用監視模式] — 當您想要先收集資料以測試防護的運作方式時，啟用監視模式。在您啟用監控模式時，如果保護處於作用中狀態，則不會封鎖請求。而是監控模式會收集請求相關資料，如果保護處於作用中狀態，則會封鎖這些請求。當您準備好開始封鎖時，您可以在 [安全性] 頁面上啟用封鎖功能。
   - 其他保護 — 選擇您要啟用的任何選項。如果您啟用速率限制，請參閱the section called “設定速率限制”取得詳細資訊。
   - 估價 — 您可以打開該部分以顯示一個字段，您可以在該字段中輸入不同的請求/月數並查看新的估計值。
6. 檢閱剩餘的分佈設定，然後選擇 [建立分發]。

建立發行版後，建 CloudFront 立安全性儀表板。您可以使用此儀表板來停用或啟用 AWS WAF。如果 AWS WAF 尚未啟用，儀表板中的圖表和圖形將保持空白。

使用現有的 Web ACL

如果您有 Web ACL，則可以使用它來代替由提供的保護 AWS WAF。

使用現有的 AWS WAF 模型組態

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 執行以下任意一項：
   a. 選擇建立分佈並依照 建立分發 中的步驟執行，然後返回本主題。
   b. 選擇現有的組態，然後選擇 [安全性] 索引標籤。
3. 在「Web 應用程式防火牆 (WAF)」區段中，選擇 「編輯」，然後選擇「啟用安全性保護」。
4. 選擇 使用現有 WAF 組態。只有在您已設定 Web ACL 時，才會顯示此選項。
5. 從 選擇 Web ACL 表格中選擇您現有的 Web ACL。
6. 檢閱剩餘的分佈設定，然後選擇 [建立發佈]。

停用 AWS WAF 安全性保護

如果您的發行版不需要 AWS WAF 安全性保護，您可以使用 CloudFront 主控台停用此功能。

如果您之前已啟用 AWS WAF 保護，但並未選擇現有的 WAF 設定 (也稱為一鍵式保護)，則 CloudFront 會自動為您建立 Web ACL。對於以這種方式建立的 Web ACL， CloudFront 主控台會取消資源的關聯，並刪除 Web ACL。

取消 Web ACL 的關聯與刪除它不同。取消關聯會從您的發行版中移除 Web ACL，但不會從您的. AWS 帳戶如需詳細資訊，請參閱、和 AWS Shield Advanced 開發人員指南中的建立 Web ACL 與 AWS 資源的關聯或取AWS WAF消關聯。 AWS Firewall Manager

請參閱下列程序，以停用 AWS WAF 保護並取消 Web ACL 與散佈的關聯。

若要停用中的 AWS WAF 安全性保護 CloudFront

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [發佈]，然後選擇您要變更的分佈。
3. 選擇「安全性」標籤，然後選擇「編輯」。
4. 在 [Web 應用程式防火牆 (WAF)] 區段中，選擇 [停用防 AWS WAF 護]。
5. 選擇儲存變更。

> **備註**
>
> - 如果您停用 AWS WAF 安全防護，但仍想要從中刪除 Web ACL AWS 帳戶，您可以手動將其刪除。按照以下程序刪除網 ACL。在 [AWS WAF 護 Shield] 主控台中，對於 [Web ACL] 頁面，您必須選擇全域 (CloudFront) 清單來尋找網路 ACL。
> - 當您從 CloudFront 主控台刪除發行版時，如果您選擇了一鍵式保護，也 CloudFront 會嘗試刪除 Web ACL。這是最大的努力，並不總是保證。如需詳細資訊，請參閱 刪除 分發。

設定速率限制

速率限制是您在設定安全保護時可能會收到的其中一項建議。

CloudFront 始終在監視器模式下啟用速率限制。啟用監視模式時，會 CloudFront擷取測量結果，告訴您是否已超過速率限制欄位中設定的速率、頻率以及多少。

儲存分配之後，會 CloudFront 開始根據「速率限制」欄位中的數字收集資料。

您可以在任何 CloudFront發行版的 [安全性] 索引標籤上的 [安全性-Web 應用程式防火牆 (WAF)] 區段中管理速率限制設定。

若要設定速率限制

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [發佈]，然後選擇您要變更的分佈。
3. 選擇 Security (安全) 標籤。
4. 在 [Web 應用程式防火牆 (WAF)] 區段中，選擇 [速率限制] 旁邊的 [監視模式訊息] 以顯示對話方塊，其中包含所收集資料的詳細資訊。您可以選擇性地變更速率限制。微調速率後，您可以選擇 [啟用封鎖] (在對話方塊上) 以停用監視器模式。 CloudFront 將開始阻止超過指定速率限制的請求。

使用 CloudFront 安全儀表板

CloudFront 為您的每個發行版建立安全性儀表板。您可以在 CloudFront 主控台中使用儀表板。透過儀表板，您可以在單一位置使用 CloudFront 並 AWS WAF 一起使用，以監控和管理 Web 應用程式的常見安全防護。儀表板提供下列任務和資料：

- 安全性設定：您可以啟用和停用 AWS WAF 保護，並查看任何應用程式特定的保護，例如保護。WordPress
- 安全性趨勢：包括允許和封鎖的請求、挑戰和 CAPTCHA 請求，以及主要攻擊類型。
- 機器人請求：您可以查看有多少流量來自機器人、有哪些類型的機器人 (已驗證與未驗證)，以及機器人類型的百分比分配 (已驗證與未驗證) 如何隨時間變化。
- 請求日誌：日誌資料可有助於回答有關安全性趨勢或機器人請求的問題。您可以在不撰寫查詢的情況下搜尋日誌，並檢視彙整圖表，以協助判斷篩選的日誌集是否主要由 HTTP 方法、IP 地址、URI 路徑或國家/地區的子集來驅動。您可以將滑鼠游標暫留在圖表中的值上，並封鎖 IP 地址和國家/地區。
- 地理限制管理

> **Note**
>
> AWS WAF 如果您想要在「 CloudFront 安全性」儀表板中檢視安全指標，則必須啟用此功能。若未 AWS WAF 啟用，您只能使用 [安全性] 儀表板來啟用 AWS WAF 或設定 CloudFront 地理限制。
> 如需啟用的更多資訊， AWS WAF 請參閱AWS WAF 為分配啟用。

以下各節說明如何使用儀表板。

主題

了解趨勢資料

儀表板中指定時間範圍的安全性趨勢區段會顯示指定時間段內流量的摘要指標。儀表板會顯示「已允許」、「已封鎖」、「挑戰」和「CAPTCHA」請求的資料。您可以查看流量比例以及它們隨時間變化的情況。例如，如果所有請求提升 3%，但允許的請求增加了 14%，這代表您在目前期間允許更大部分流量通過。

此區段提供三種長條圖：請求、熱門攻擊類型和熱門國家/地區。您可以使用熱門國家/地區圖表來封鎖國家/地區。

如要使用圖表

- 使用圖表上方的日期範圍、規則動作和精細程度控制項來設定時間範圍並篩選資料。
- 將滑鼠游標暫留在任何長條上，即可查看指定時段內的請求、攻擊或國家/地區資料。
- 若要封鎖某個國家/地區，請將滑鼠游標移到該列上，並將封鎖國家/地區名稱滑桿移至開啟位置。

> **Note**
>
> 如果您先前在 CloudFront 主控台外建立自訂 AWS WAF 規則來封鎖國家/地區，則可能無法使用 [封鎖] 選項。

啟用機器人控制功能

特定時間範圍的機器人請求區段會顯示機器人請求資料。您可以啟用或停用 AWS WAF 機器人控制功能。啟用機器人控制功能時會產生費用，且儀表板會提供成本估計。

如果您啟用機器人控制功能，儀表板圖表會顯示來自每種類型和機器人類別的流量。如果您停用機器人控制功能，圖表會根據請求取樣顯示流量。

此區段提供兩種長條圖：依機器人類型的請求和依機器人類別的請求。圖表之差異，取決您是否啟用機器人控制功能。

啟用機器人控制功能

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇分佈，然後選擇您要變更的分佈。
3. 選擇 Security (安全) 標籤。
4. 向下捲動至指定時間範圍的機器人請求區段，然後選擇啟用機器人控制功能。
5. 在「機器人控制」對話方塊的「組態」下，選取「為一般機器人啟用機器人控制」核取方塊。
6. 選擇儲存變更。

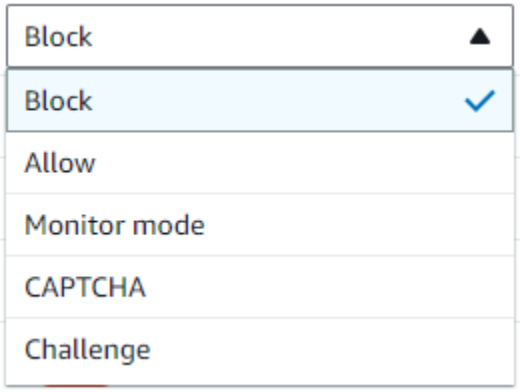
當您啟用機器人保護時，您可以選擇設定每個未經驗證的機器人如何依據機器人類別進行處理。例如，您可以將 HTTP 程式庫機器人設定為監控模式，並將挑戰指派給連結檢查程式。

已知為常見且可驗證 AWS 的機器人 (例如已知的搜尋引擎搜尋器) 不受您在此處設定的動作所限制。機器人控制功能會在將經過驗證的機器人標記為已驗證之前，確認其來源是否來自其聲明的來源。

若要設定類別的保護

1. 重複上述步驟中的 1 和 2，以啟動安全性儀表板。
2. 在按機器人分類請求圖表中，指向未驗證機器人動作欄中的任何項目，然後選擇編輯圖示。

3. 在導覽窗格清單中，請選擇下列其中一個：
   - 封鎖
   - Allow
   - 監控模式
   - CAPTCHA
   - 挑戰



4. 選取清單旁邊的勾選標記以確認變更。

如要使用圖表

- 在指定時間範圍的安全性趨勢區段中，使用日期範圍、規則動作，和精細程度控制項設定時間範圍並篩選機器人資料。

- 在依機器人類型的請求圖表中，將滑鼠游標暫留在任何長條上，即可查看依機器人類型分類的請求數目。
- 在依機器人類別的請求圖表中，將滑鼠游標暫留在任何長條上，即可查看依機器人類別分類的請求數目。

了解日誌

日誌資料可協助您隔離特定的流量模式。例如，日誌可以顯示特定流量來自何處或其作用。

啟用日誌

1. 在每月請求數方塊中輸入預期的請求數量，以預估啟用日誌的成本。
2. 選取 [啟用 AWS WAF 記錄檔] 核取方塊。
3. 選擇 啟用 。

CloudFront 會建立記 CloudWatch 錄群組，並更新您的組 AWS WAF 態以開始記錄 CloudWatch。首次使用時，日誌資料可能需要幾分鐘才會出現。圖表的請求區段會列出每個請求。在個別請求下方，長條圖會依據 HTTP 方法、熱門 URI 路徑、熱門 IP 地址和熱門國家/地區彙總資料。圖表可以幫助您找出模式。例如，您可能會看到來自單一 IP 地址的請求數量不成比例，或是您先前在日誌中未看到的國家/地區的資料。您可以依據國家/地區、主機標頭和其他屬性篩選請求，以協助尋找不想要的流量。找出該流量後，請將滑鼠游標暫留在個別請求或圖表項目上，並封鎖 IP 地址或國家/地區。

如要使用圖表

- 使用指定時間範圍的安全性趨勢區段中的日期範圍、規則動作和精細程度控制來設定時間範圍並篩選資料。
- 將滑鼠游標暫留在任何長條上，即可查看指定時段內的 URI 路徑、IP 地址或國家/地區資料。
- 若要封鎖 IP 地址或國家/地區，請將滑鼠游標暫留在該圖條上，然後將封鎖項目名稱滑桿移至開啟位置。

> **Note**
>
> 如果您先前在 CloudFront 主控台外建立自訂 AWS WAF 規則來封鎖國家/地區或 IP 位址，則可能無法使用 [封鎖] 選項。

> **Note**
>
> 顯示的量度是以 Web 存取控制清單 (ACL) 為基礎。因此，如果您將同一個 Web ACL 與多個發行版產生關聯，您將會看到 Web ACL 的所有量度，而不僅是針對該分發處理的 AWS WAF 請求。

管理 CloudFront 地理限制

您可隨時管理地理限制。

若要管理地理限制

1. 向下捲動至地理限制區段。
2. 選擇編輯。
3. 選取允許清單將國家/地區新增至您允許的國家/地區清單，或選取封鎖清單，將國家/地區新增至您的封鎖國家/地區清單。
4. 將所需的國家/地區新增至清單中，然後選擇儲存變更。

CloudFront 並 AWS WAF 提供地理限制功能。 CloudFront 免費提供地理限制，但您的儀表板不會顯示被封鎖國家/地區的指標。相反地，當您將游標暫留在安全性儀表板中的國家/地區列上並封鎖某個國家/地區時，您會使用 AWS WAF 地理限制。它們也會封鎖國家/地區，但您的儀表板會顯示已封鎖請求的請求指標。

安全性儀表板定價

如果您啟用 Amazon 的 AWS WAF 記錄功能 CloudWatch， CloudFront 安全儀表板會查詢、彙總和顯示來自 CloudWatch 日誌的見解。使用安全儀表板不收取費用，但 Amazon CloudWatch 定價適用於透過儀表板查詢的日誌。如需詳細資訊，請參閱 Amazon CloudWatch 定價。

設定安全存取和限制對內容的存取

CloudFront 提供數個選項來保護其提供的內容。以下是您可以用 CloudFront 來保護和限制內容存取的一些方法：

- 設定 HTTPS 連線
- 防止特定地理位置的使用者存取內容
- 要求使用者使用 CloudFront 已簽署的網址或已簽署的 Cookie 存取內容
- 設定特定內容欄位的欄位層級加密
- 用 AWS WAF 於控制對內容的存取

主題

搭配使用 HTTPS CloudFront

您可以設定 CloudFront 為要求檢視者使用 HTTPS，以便在與檢視者 CloudFront 通訊時加密連線。您也可以設定 CloudFront 為在原始伺服器上使用 HTTPS，以便在與原始伺服器 CloudFront 通訊時加密連線。

如果您設定 CloudFront 為要求 HTTPS 與檢視者通訊並與來源通訊，則 CloudFront 收到請求時會發生以下情況：

1. 檢視者將 CloudFront HTTPS 要求提交給。這裡有一些 SSL/TLS 協商之間的檢視器和. CloudFront 最後，檢視器以加密格式提交請求。
2. 如果邊 CloudFront 緣位置包含快取的回應，則會 CloudFront 加密回應並將其傳回給檢視器，然後檢視者將其解密。

3. 如果邊 CloudFront 緣位置不包含快取的回應，請與原始伺服器 CloudFront 執行 SSL/TLS 交涉，並在交涉完成時，以加密格式將要求轉送至您的來源。
4. 您的來源會解密請求，處理它（生成響應），加密響應並將響應返回給。 CloudFront
5. CloudFront 解密響應，重新加密，然後將其轉發給查看器。 CloudFront還將響應緩存在節點位置，以便在下次請求時可以使用響應。
6. 檢視器解密回應。

無論您的來源是 Amazon S3 儲存貯體還是自訂來源 (例如 HTTP/S 伺服器) MediaStore，此程序的運作方式基本上都相同。

> **Note**
>
> 為協助阻止 SSL 重新交涉類型的攻擊， CloudFront 不支援檢視器和來源要求的重新交涉。

如需有關如何在檢視者和來源之間以及 CloudFront在檢視者之間 CloudFront 要求 HTTPS 的詳細資訊，請參閱下列主題。

主題

- 要求使用 HTTPS 才能在檢視者和 CloudFront
- CloudFront 與您的自訂原始伺服器之間的通訊需要 HTTPS
- 需要 HTTPS 進行 CloudFront與您的 Amazon S3 來源之間的通訊
- 檢視器與之間支援的通訊協定和密碼 CloudFront
- 與來源之間支援的通訊協定 CloudFront 和密碼

要求使用 HTTPS 才能在檢視者和 CloudFront

您可以在 CloudFront 發行版中設定一或多個快取行為，使其在檢視器和 CloudFront. 您也可以設定一或多個快取行為，以允許 HTTP 和 HTTPS，如此一來，某些物件 CloudFront 需要 HTTPS，但其他物件則不需要 HTTPS。組態步驟取決於您在物件 URL 中使用哪個網域名稱：

- 如果您使用 CloudFront 指派給分發的網域名稱，例如 d1111abcdef8.cloudfront.net，您可以變更一或多個快取行為的檢視器通訊協定原則設定，以需要進行 HTTPS 通訊。在該組態中，CloudFront 提供 SSL/TLS 憑證。

  若要使用 CloudFront 主控台變更檢視器通訊協定原則的值，請參閱本節稍後的程序。

如需如何使用 CloudFront API 變更`ViewerProtocolPolicy`元素值的詳細資訊，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

- 如果您使用自己的網域名稱，例如 example.com，則需要變更數個 CloudFront設定。您也需要使用 AWS Certificate Manager (ACM) 提供的 SSL/TLS 憑證，或是從第三方憑證授權機構或 IAM 憑證存放區將憑證匯入至 ACM。如需詳細資訊，請參閱 使用備用網域名稱和 HTTPS。

> **Note**
>
> 如果您想確保檢視者從中取得的物件在從您的 CloudFront 來源取 CloudFront 得物件時已加密，請務必在 CloudFront 與原始伺服器之間使用 HTTPS。如果您最近在 CloudFront 和原始位置之間從 HTTP 變更為 HTTPS，建議您將 CloudFront 邊緣位置中的物件無效。 CloudFront 無論檢視器 (HTTP 或 HTTPS) 使用的通訊協定是否符合用來取得物件的通訊協定，都會將物件傳回給檢視器。 CloudFront 如需有關移除或替換分佈中物件的詳細資訊，請參閱新增、移除或取代 CloudFront散佈的內容。

若要在檢視器之間需要 HTTPS，以及 CloudFront 一或多個快取行為，請執行下列程序。

若要設定 CloudFront 為在檢視者之間需要 HTTPS 和 CloudFront

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在 CloudFront 主控台的上方窗格中，選擇您要更新之發行版的 ID。
3. 在 [行為] 索引標籤上，選取您要更新的快取行為，然後選擇 [編輯]。
4. 為檢視器通訊協定原則指定下列其中一個值：

   重新導向 HTTP 到 HTTPS

   檢視器可以使用這兩種通訊協定。HTTP `GET` 和`HEAD`要求會自動重新導向至 HTTPS 要求。 CloudFront 返回 HTTP 狀態碼 301（永久移動）以及新的 HTTPS 網址。檢視器接著會 CloudFront 使用 HTTPS URL 將要求重新提交至。

   > **Important**
   >
   > 如果您透`PATCH`過 HTTP 至 HTTPS 快取行為和 HTTP 1.1 或更新版本的要求通訊協定版本傳送`POST`、、或透過 HTTP 傳送、、或透過 HTTP，請將要求重新

導 CloudFront 向至具有 HTTP 狀態碼 307 (暫時重新導向) 的 HTTPS 位置。PUT DELETE OPTIONS這可確保再次將請求傳送到使用相同方法與內容承載的新位置。如果您使用低於 HTTP 1.1 的要PATCH求通訊協定版本OPTIONS，透過 HTTP 傳送POST、、、或要求至 HTTPS 快取行為，則會傳 CloudFront 回 HTTP 狀態碼 403 (禁止)。PUT DELETE

當檢視者發出重新導向至 HTTPS 要求的 HTTP 要求時，兩個要求都會 CloudFront收取費用。對於 HTTP 請求，費用僅適用於請求以及 CloudFront 返回給檢視器的標頭。針對 HTTPS 請求，會收取該請求與原始伺服器傳回的標頭及物件之費用。

僅限使用 HTTPS

檢視器只能在使用 HTTPS 的情況下存取您的內容。如果檢視器傳送 HTTP 要求而非 HTTPS 要求，則會傳 CloudFront 回 HTTP 狀態碼 403 (禁止)，而且不會傳回物件。

5. 選擇儲存變更。
6. 針對您想要在檢視器和檢視器之間需要 HTTPS 的每個額外快取行為，重複步驟 3 到 5 CloudFront。
7. 您使用生產環境中已更新的組態之前，請先確認以下項目：
   - 每個快取行為中的路徑模式僅適用於您想要檢視器使用 HTTPS 的請求。
   - 快取行為會以您要評估 CloudFront 的順序列出。如需詳細資訊，請參閱 路徑模式。
   - 快取行為會將請求路由到正確的原始伺服器。

CloudFront 與您的自訂原始伺服器之間的通訊需要 HTTPS

您可以要求 HTTPS 在與原始伺服器 CloudFront 之間進行通訊。

> **Note**
>
> 如果您的原始伺服器是設定為網站端點的 Amazon S3 儲存貯體，則無法設定 CloudFront 為在原始伺服器上使用 HTTPS，因為 Amazon S3 不支援網站端點的 HTTPS。

若要在 CloudFront 與原始伺服器之間要求 HTTPS，請遵循本主題中的程序執行下列作業：

1. 在您的分佈中，針對原始伺服器變更其 Origin Protocol Policy (原始伺服器通訊協定政策) 設定。

2. 在原始伺服器上安裝 SSL/TLS 憑證 (當您使用 Amazon S3 來源或某些其他來 AWS 源時不需要此憑證)。

主題

變更 CloudFront設定

下列程序說明如何設定 CloudFront 為使用 HTTPS 與 Elastic Load Balancing 負載平衡器、Amazon EC2 執行個體或其他自訂來源通訊。如需使用 CloudFront API 更新分發的相關資訊，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

設定 CloudFront 為在 CloudFront 與您的自訂原始伺服器之間需要 HTTPS

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在 CloudFront 主控台的上方窗格中，選擇您要更新之發行版的 ID。
3. 在「行為」頁籤上，選取您要更新的原點，然後選擇「編輯」。
4. 更新下列設定：

   原始伺服器通訊協定政策

   針對您分佈中適用的原始伺服器，變更其 Origin Protocol Policy (原始伺服器通訊協定政策)：

   - 僅限 HTTPS — 僅 CloudFront 使用 HTTPS 與您的自訂原始伺服器通訊。
   - 比對檢視器 — 根據檢視器要求的 CloudFront 通訊協定，使用 HTTP 或 HTTPS 與您的自訂來源進行通訊。例如，如果您選擇符合原始通訊協定原則的檢視器，而檢視者使用 HTTPS 向其要求物件 CloudFront， CloudFront 也會使用 HTTPS 將請求轉寄至您的來源。

     只有針對 Viewer Protocol Policy (檢視器通訊協定政策) 指定 Redirect HTTP to HTTPS (將 HTTP 重新導向至 HTTPS) 或 HTTPS Only (僅限 HTTPS) 時，才能選擇 Match Viewer (配合檢視器)。

     CloudFront 即使檢視者同時使用 HTTP 和 HTTPS 通訊協定發出要求，也只會快取物件一次。

原始伺服器 SSL 通訊協定

針對分佈中適用的原始伺服器，選擇 Origin SSL Protocols (原始伺服器 SSL 通訊協定)。SSLv3 通訊協定較不安全，因此我們建議您只有在原始伺服器不支援 TLSv1 或新版本時，選擇 SSLv3。TLSv1 交握與 SSLv3 有回溯相容與正向相容，但 TLSv1.1 和 TLSv1.2 則無。當您選擇 SSLv3 時， CloudFront 只會傳送 SSLv3 交握要求。

5. 選擇儲存變更。
6. 針對您要在自訂原點之間 CloudFront 需要 HTTPS 的每個額外來源重複步驟 3 到 5。
7. 您使用生產環境中已更新的組態之前，請先確認以下項目：

   - 每個快取行為中的路徑模式僅適用於您想要檢視器使用 HTTPS 的請求。
   - 快取行為會以您要評估 CloudFront 的順序列出。如需詳細資訊，請參閱 路徑模式。
   - 快取行為會將請求轉傳到您已變更其 Origin Protocol Policy (原始伺服器通訊協定政策) 的原始伺服器。

在您的自訂原始伺服器上安裝 SSL/TLS 憑證

在自訂原始伺服器上，您可以從以下來源使用 SSL/TLS 憑證：

- 如果您的原始伺服器是 Elastic Load Balancing 負載平衡器，則可以使用 AWS Certificate Manager (ACM) 提供的憑證。您也可以使用信任第三方憑證授權單位簽署的憑證並匯入 ACM。
- 對於 Elastic Load Balancing 負載平衡器以外的來源，您必須使用由受信任的協力廠商憑證授權單位 (CA) 簽署的憑證，例如 Comodo 或賽門鐵克。DigiCert

從原始伺服器傳回的憑證必須包含下列其中一個網域名稱：

- 來源網域欄位中的網域名稱 (CloudFront API 中的`DomainName`欄位)。
- 如果快取行為已設定為轉發 `Host` 標頭至原始伺服器，則為 `Host` 標頭中的網域名稱。

CloudFront 使用 HTTPS 與您的來源通訊時，請 CloudFront 確認憑證是由受信任的憑證授權單位所簽發。 CloudFront 支援與 Mozilla 相同的憑證授權單位。如需目前的清單，請參閱 Mozilla 內建 CA 憑證清單。您無法使用自我簽署憑證在 CloudFront 與原始伺服器之間進行 HTTPS 通訊。

> **Important**
>
> 如果原始伺服器傳回過期的憑證、無效的憑證或自我簽署的憑證，或者原始伺服器以錯誤的順序傳回憑證鏈結， CloudFront 請中斷 TCP 連線，將 HTTP 狀態碼 502 (Bad Gateway) 傳回給檢視器，並將`X-Cache`標頭設定為`Error from cloudfront`。此外，如果完整的憑證鏈結 (包括中繼憑證) 不存在， CloudFront 則會中斷 TCP 連線。

需要 HTTPS 進行 CloudFront與您的 Amazon S3 來源之間的通訊

當您的來源是 Amazon S3 儲存貯體時，使用 HTTPS 進行通訊的選項 CloudFront 取決於您使用儲存貯體的方式。如果您的 Amazon S3 儲存貯體設定為網站端點，則無法設定 CloudFront 為使用 HTTPS 與原始伺服器通訊，因為 Amazon S3 在該組態中不支援 HTTPS 連線。

當您的原始伺服器是支援 HTTPS 通訊的 Amazon S3 儲存貯體時，請務 CloudFront 必使用檢視者用來提交請求的通訊協定將請求轉送至 S3。通訊協定 (僅限自訂原始伺服器) 設定的預設為比對檢視器，而且無法變更。

如果您想要使用 HTTPS 進行 CloudFront 與 Amazon S3 之間的通訊，您必須將檢視器通訊協定政策的值變更為將 HTTP 重新導向至 HTTPS 或僅限 HTTPS。本節稍後的程序說明如何使用 CloudFront 主控台變更檢視器通訊協定原則。如需使用 CloudFront API 更新分發`ViewerProtocolPolicy`元素的相關資訊，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

當您搭配支援 HTTPS 通訊的 Amazon S3 儲存貯體使用 HTTPS 時，Amazon S3 會提供 SSL/TLS 憑證，您則不需要。

設定為 CloudFront 要求使用 HTTPS 至您的 Amazon S3 原始伺服器

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在 CloudFront 主控台的上方窗格中，選擇您要更新之發行版的 ID。
3. 在 Behaviors (行為) 索引標籤中，選擇您要更新的快取行為，然後選擇 Edit (編輯)。
4. 請針對 Viewer Protocol Policy (檢視器通訊協定政策) 指定下列其中一個值：

   重新引導 HTTP 到 HTTPS

   檢視者可以使用這兩種通訊協定，但 HTTP 要求會自動重新導向至 HTTPS 要求。 CloudFront 返回 HTTP 狀態碼 301（永久移動）以及新的 HTTPS 網址。檢視器接著會 CloudFront 使用 HTTPS URL 將要求重新提交至。

> **Important**
>
> CloudFront 不會將DELETE、、OPTIONSPATCHPOST、或PUT要求從 HTTP 重新導向至 HTTPS。如果您將快取行為設定為重新導向至 HTTPS，請使用 HTTP DELETE 狀態碼 403 (禁止) CloudFront 回應該快取行為的 HTTP POST、、、或PUT要求。OPTIONS PATCH

當檢視者發出重新導向至 HTTPS 要求的 HTTP 要求時，兩個要求都會 CloudFront收取費用。對於 HTTP 請求，費用僅適用於請求以及 CloudFront 返回給檢視器的標頭。針對 HTTPS 請求，會收取該請求與標頭及原始伺服器傳回的物件之費用。

僅限 HTTPS

檢視器只能在使用 HTTPS 的情況下存取您的內容。如果檢視器傳送 HTTP 要求而非 HTTPS 要求，則會傳 CloudFront 回 HTTP 狀態碼 403 (禁止)，而且不會傳回物件。

5. 請選擇 Yes, Edit (是，編輯)。
6. 針對您想要在檢視器和和 S3 之間需要 HTTPS 的每個額外快取行為 CloudFront，重複步驟 3 到 5。 CloudFront
7. 您使用生產環境中已更新的組態之前，請先確認以下項目：
   - 每個快取行為中的路徑模式僅適用於您想要檢視器使用 HTTPS 的請求。
   - 快取行為會以您要評估 CloudFront 的順序列出。如需詳細資訊，請參閱 路徑模式。
   - 快取行為會將請求路由到正確的原始伺服器。

檢視器與之間支援的通訊協定和密碼 CloudFront

當您在檢視者和 CloudFront 發行版之間需要 HTTPS 時，您必須選擇安全性原則來決定下列設定：

- 用來與檢視者通訊的最低 SSL/ CloudFront TLS 通訊協定。
- CloudFront 可用來加密與檢視者通訊的密碼。

若要選擇安全政策，請指定適用於 安全性政策 (最低 SSL/TLS 版本) 的適用值。下表列出 CloudFront 可用於每個安全性原則的通訊協定和密碼。

檢視器必須支援至少一個受支援的密碼，才能與之建立 HTTPS 連線。 CloudFront CloudFront 從檢視器支援的密碼中，依列出的順序選擇密碼。另請參閱OpenSSL、S2n 和 RFC 密碼名稱。

	安全政策						
	SSLv3	TLSv1	TLSv1_26	TLSv1.1_016	TLSv1.2_018	TLSv1.2_019	TLSv1.2_2021
支援的 SSL/TLS 通訊協定							
TLSv1.3	◆	◆	◆	◆	◆	◆	◆
TLSv1.2	◆	◆	◆	◆	◆	◆	◆
TLSv1.1	◆	◆	◆	◆			
TLSv1	◆	◆	◆				
SSLv3	◆						
支援的 TLSv1.3 密碼							
TLS_AES_128_GCM_SHA256	◆	◆	◆	◆	◆	◆	◆
TLS_AES_256_GCM_SHA384	◆	◆	◆	◆	◆	◆	◆
TLS_CHACHA20_POLY1305_SHA256	◆	◆	◆	◆	◆	◆	◆
支援的 ECDSA 密碼							
ECDHE-ECDSA-AES128-GCM-SHA256	◆	◆	◆	◆	◆	◆	◆
ECDHE-ECDSA-AES128-SHA256	◆	◆	◆	◆	◆	◆	
ECDHE-ECDSA-AES128-SHA	◆	◆	◆	◆			

	安全政策						
	SSLv3	TLSv1	TLSv1_26	TLSv1.1_016	TLSv1.2_018	TLSv1.2_019	TLSv1.2_2021
ECDHE-ECDSA-AES256-GCM-SHA384	♦	♦	♦	♦	♦	♦	♦
ECDHE-ECDSA-CHACHA20-POLY1305	♦	♦	♦	♦	♦	♦	♦
ECDHE-ECDSA-AES256-SHA384	♦	♦	♦	♦	♦	♦	
ECDHE-ECDSA-AES256-SHA	♦	♦	♦	♦			
支援的 RSA 密碼							
ECDHE-RSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦	♦
ECDHE-RSA-AES128-SHA256	♦	♦	♦	♦	♦	♦	
ECDHE-RSA-AES128-SHA	♦	♦	♦	♦			
ECDHE-RSA-AES256-GCM-SHA384	♦	♦	♦	♦	♦	♦	♦
ECDHE-RSA-CHACHA20-POLY1305	♦	♦	♦	♦	♦	♦	♦
ECDHE-RSA-AES256-SHA384	♦	♦	♦	♦	♦	♦	
ECDHE-RSA-AES256-SHA	♦	♦	♦	♦			

	安全政策						
	SSLv3	TLSv1	TLSv1_2 6	TLSv1.1_ 016	TLSv1.2_ 018	TLSv1.2_ 019	TLSv1.2_2 021
AES128-GCM-SHA256	♦	♦	♦	♦	♦		
AES256-GCM-SHA384	♦	♦	♦	♦	♦		
AES128-SHA256	♦	♦	♦	♦	♦		
AES256-SHA	♦	♦	♦	♦			
AES128-SHA	♦	♦	♦	♦			
DES-CBC3-SHA	♦	♦					
RC4-MD5	♦						

OpenSSL、S2n 和 RFC 密碼名稱

penSSL and s2n 會使用與 TLS 標準不同的密碼名稱 (RFC 2246、RFC 4346、RFC 5246 和 RFC 8446)。下表將 OpenSSL 和 s2n 名稱對應到每個密碼的 RFC 名稱。

對於具有橢圓曲線金鑰交換演算法的密碼， CloudFront 支援下列橢圓形曲線：

- prime256v1
- secp384r1
- X25519

OpenSSL 和 s2n 密碼名稱	RFC 密碼名稱
支援的 TLSv1.3 密碼	
TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256	TLS_CHACHA20_POLY1305_SHA256

OpenSSL 和 s2n 密碼名稱	RFC 密碼名稱
支援的 ECDSA 密碼	
ECDHE-ECDSA-AES128-GCM-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
ECDHE-ECDSA-AES128-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
ECDHE-ECDSA-AES128-SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
ECDHE-ECDSA-AES256-GCM-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE-ECDSA-CHACHA20-POLY1305	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
ECDHE-ECDSA-AES256-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
ECDHE-ECDSA-AES256-SHA	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
支援的 RSA 密碼	
ECDHE-RSA-AES128-GCM-SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ECDHE-RSA-AES128-SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
ECDHE-RSA-AES128-SHA	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
ECDHE-RSA-AES256-GCM-SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

OpenSSL 和 s2n 密碼名稱	RFC 密碼名稱
ECDHE-RSA-CHACHA20-POLY1305	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
ECDHE-RSA-AES256-SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE-RSA-AES256-SHA	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
AES128-GCM-SHA256	TLS_RSA_WITH_AES_128_GCM_SHA256
AES256-GCM-SHA384	TLS_RSA_WITH_AES_256_GCM_SHA384
AES128-SHA256	TLS_RSA_WITH_AES_128_CBC_SHA256
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES-CBC3-SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
RC4-MD5	TLS_RSA_WITH_RC4_128_MD5

檢視器與之間支援的簽名配置 CloudFront

CloudFront 支援下列檢視器與之間的連線簽名配置CloudFront。

- TLS_SIGNATURE_SCHEME_RSA_PSS_PSS_SHA256
- TLS_SIGNATURE_SCHEME_RSA_PSS_PSS_SHA384
- TLS_SIGNATURE_SCHEME_RSA_PSS_PSS_SHA512
- TLS_SIGNATURE_SCHEME_RSA_PSS_RSAE_SHA256
- TLS_SIGNATURE_SCHEME_RSA_PSS_RSAE_SHA384
- TLS_SIGNATURE_SCHEME_RSA_PSS_RSAE_SHA512
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA256
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA384
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA512

- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA224
- TLS_SIGNATURE_SCHEME_ECDSA_SHA256
- TLS_SIGNATURE_SCHEME_ECDSA_SHA384
- TLS_SIGNATURE_SCHEME_ECDSA_SHA512
- TLS_SIGNATURE_SCHEME_ECDSA_SHA224
- TLS_SIGNATURE_SCHEME_ECDSA_SECP256R1_SHA256
- TLS_SIGNATURE_SCHEME_ECDSA_SECP384R1_SHA384
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA1
- TLS_SIGNATURE_SCHEME_ECDSA_SHA1

與來源之間支援的通訊協定 CloudFront 和密碼

如果您選擇在 CloudFront 與原始伺服器之間需要 HTTPS，您可以決定要允許哪個 SSL/TLS 通訊協定進行安全連線，並 CloudFront 可以使用下表列出的任何 ECDSA 或 RSA 密碼來連線至原始伺服器。您的來源必須至少支援其中一個密碼，才能建立與您的 CloudFront 來源的 HTTPS 連線。

penSSL and s2n 會使用與 TLS 標準不同的密碼名稱 (RFC 2246、RFC 4346、RFC 5246 和 RFC 8446)。下表將包含每種密碼的 OpenSSL 和 s2n 名稱以及 RFC 名稱。

對於具有橢圓曲線金鑰交換演算法的密碼， CloudFront 支援下列橢圓形曲線：

- prime256v1
- secp384r1
- X25519

OpenSSL 和 s2n 密碼名稱	RFC 密碼名稱
支援的 ECDSA 密碼	
ECDHE-ECDSA-AES256-GCM-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE-ECDSA-AES256-SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

OpenSSL 和 s2n 密碼名稱	RFC 密碼名稱
ECDHE-ECDSA-AES256-SHA	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ECDHE-ECDSA-AES128-GCM-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
ECDHE-ECDSA-AES128-SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
ECDHE-ECDSA-AES128-SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
支援的 RSA 密碼	
ECDHE-RSA-AES256-GCM-SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE-RSA-AES256-SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE-RSA-AES256-SHA	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ECDHE-RSA-AES128-GCM-SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ECDHE-RSA-AES128-SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
ECDHE-RSA-AES128-SHA	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES-CBC3-SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA

OpenSSL 和 s2n 密碼名稱	RFC 密碼名稱
RC4-MD5	TLS_RSA_WITH_RC4_128_MD5

CloudFront 與原點之間支援的簽章配置

CloudFront 支援下列 CloudFront 與原點之間的連線簽章配置。

- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA256
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA384
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA512
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA224
- TLS_SIGNATURE_SCHEME_ECDSA_SHA256
- TLS_SIGNATURE_SCHEME_ECDSA_SHA384
- TLS_SIGNATURE_SCHEME_ECDSA_SHA512
- TLS_SIGNATURE_SCHEME_ECDSA_SHA224
- TLS_SIGNATURE_SCHEME_RSA_PKCS1_SHA1
- TLS_SIGNATURE_SCHEME_ECDSA_SHA1

使用備用網域名稱和 HTTPS

若您想要在檔案的 URL 中使用自己的網域名稱 (例如 `https://www.example.com/image.jpg`)，並希望您的檢視器使用 HTTPS，則必須完成下列主題的步驟。(例如，如果您在 URL 中使用預設 CloudFront 分發網域名稱`https://d111111abcdef8.cloudfront.net/image.jpg`，請改為遵循下列主題中的指引：要求使用 HTTPS 才能在檢視者和 CloudFront)

> **Important**
>
> 當您將憑證新增至散發時，會 CloudFront 立即將憑證傳播到其所有邊緣位置。當新的節點可用時，也會將憑證CloudFront 傳播到這些位置。您無法限制將憑證 CloudFront 傳播到的 Edge 位置。

主題

選擇如何 CloudFront 提供 HTTPS 要求

如果您希望檢視者使用 HTTPS 並為您的檔案使用替代網域名稱，請選擇下列其中一個選項來處 CloudFront 理 HTTPS 要求的方式：

- 使用伺服器名稱指示 (SNI) – 推薦
- 在每個節點使用專用的 IP 地址

本節說明每個選項的運作方式。

使用 SNI 提供 HTTPS 請求 (適用於大部分用戶端)

伺服器名稱指示 (SNI) 是 TLS 通訊協定的延伸，2010 年之後推出的瀏覽器和用戶端支援此選項。如果您設定 CloudFront 為使用 SNI 提供 HTTPS 要求，請 CloudFront 將您的替代網域名稱與每個節點的 IP 位址建立關聯。當檢視器提交內容的 HTTPS 請求時，DNS 會將請求路由到正確節點的 IP 地址。您網域名稱的 IP 地址在 SSL/TLS 交握溝通期間決定；IP 地址並非專用於您的分佈。

SSL/TLS 溝通發生在建立 HTTPS 連線的初期。如果 CloudFront 無法立即判斷要求的網域，就會中斷連線。當 SNI 支援的檢視器提交內容的 HTTPS 請求時，情況如下：

1. 檢視器會自動從要求 URL 取得網域名稱，並將其新增至 TLS 用戶端 Hello 訊息的 SNI 延伸。
2. 當 CloudFront 收到 TLS 用戶端 Hello 時，它會使用 SNI 延伸模組中的網域名稱來尋找相符的 CloudFront 散佈，並傳回相關聯的 TLS 憑證。
3. 檢視器並 CloudFront 執行 SSL/TLS 交涉。
4. CloudFront 將要求的內容傳回給檢視者。

關於支援 SNI 的瀏覽器，最新的清單請參閱 Wikipedia 條目 Server Name Indication (伺服器名稱指示)。

如果您想要使用 SNI，但某些使用者的瀏覽器不支援 SNI，則您有幾個選項：

- 設定 CloudFront 為使用專用 IP 位址而非 SNI 來提供 HTTPS 要求。如需詳細資訊，請參閱 使用專用 IP 地址提供 HTTPS 請求 (適用於所有用戶端)。
- 使用 CloudFront SSL/TLS 憑證而非自訂憑證。這需要您在檔案的 URL 中使用散佈的 CloudFront 網域名稱，例如`https://d111111abcdef8.cloudfront.net/logo.png`。

  如果您使用預設 CloudFront 憑證，檢視者必須支援 SSL 通訊協定 TLSv1 或更新版本。 CloudFront 不支援使用預設CloudFront 憑證的 SSLv3。

  您也必須將使用的 SSL/TLS 憑證 CloudFront 從自訂憑證變更為預設 CloudFront 憑證：
  - 如果您尚未使用分佈來分配內容，可以只變更組態。如需詳細資訊，請參閱 更新分佈。
  - 如果您已經使用分發來散佈內容，則必須建立新的 CloudFront 分發並變更檔案的 URL，以減少或減少內容無法使用的時間。如需詳細資訊，請參閱 從自訂 SSL/TLS 憑證還原為預設憑證 CloudFront 。
- 如果您可以控制您的使用者使用哪些瀏覽器，請讓他們升級到支援 SNI 的瀏覽器。
- 使用 HTTP，而非 HTTPS。

使用專用 IP 地址提供 HTTPS 請求 (適用於所有用戶端)

伺服器名稱指示 (SNI) 是將請求關聯至網域的一個方法。另一個方式是使用專用 IP 地址。如果您有使用者無法升級到 2010 年後推出的瀏覽器或用戶端，您可以使用專用 IP 地址來提供 HTTPS 請求。關於支援 SNI 的瀏覽器，最新的清單請參閱 Wikipedia 條目 Server Name Indication (伺服器名稱指示)。

> **Important**
>
> 如果您設定 CloudFront 為使用專用 IP 位址提供 HTTPS 要求，則需支付每月額外費用。當您把 SSL/TLS 憑證與分佈相關聯並啟用此分佈時，即開始收費。如需有關 CloudFront 定價的詳細資訊，請參閱 Amazon CloudFront 定價。除此之外：請參閱 Using the Same Certificate for Multiple CloudFront Distributions。

當您設定 CloudFront 使用專用 IP 位址提供 HTTPS 要求時，請CloudFront將您的憑證與每個節 CloudFront 點中的專用 IP 位址建立關聯。當檢視器提交內容的 HTTPS 請求時，情況如下：

1. DNS 將請求路由到適用節點中分佈的 IP 地址。
2. 如果用戶端要求在`ClientHello`郵件中提供 SNI 延伸模組，則會 CloudFront 搜尋與該 SNI 相關聯的發佈。
   - 如果有相符項目，請使用 SSL/TLS 憑證 CloudFront 回應要求。
   - 如果沒有相符項目，請改 CloudFront 用 IP 位址來識別您的散佈，並決定要傳回給檢視器的 SSL/TLS 憑證。
3. 檢視器並使用您的 SSL/TLS 憑證 CloudFront 執行 SSL/TLS 交涉。
4. CloudFront 將要求的內容傳回給檢視者。

這種方式適用於每個 HTTPS 請求，無論使用者使用的是瀏覽器或其他檢視器。

請求使用三個或更多專用 IP SSL/TLS 憑證的許可

如果您需要將三個或更多 SSL/TLS 專用 IP 憑證與永久關聯的權限 CloudFront，請執行下列程序。如需 HTTPS 請求的詳細資訊，請參閱 選擇如何 CloudFront 提供 HTTPS 要求。

> **Note**
>
> 此程序適用於在您的 CloudFront 發行版中使用三個以上的專用 IP 憑證。預設值為 2。請注意，您不可以繫結超過 1 個 SSL 憑證到分佈。
> 您一次只能將單一 SSL/TLS 憑證關聯至一個 CloudFront 發行版。此數字代表您可以在所有 CloudFront 發行版中使用的專用 IP SSL 憑證總數。

要求使用三個或更多憑證與 CloudFront散發的權限

1. 請前往支援中心並建立案例。
2. 請指出您需要許可才能使用的憑證有多少，以及說明請求中的情況。我們會儘快更新您的帳戶。
3. 請繼續下一個程序。

搭配使用 SSL/TLS 憑證的需求 CloudFront

本主題描述 SSL/TLS 憑證的需求。它們適用於以下兩種情況 (除非另有說明)：

- 在檢視者和之間使用 HTTPS 的憑證 CloudFront
- 在 CloudFront 與您的來源之間使用 HTTPS 的憑證

主題

憑證發行者

我們建議您使用由 AWS Certificate Manager (ACM) 出具的憑證。如需有關如何從 ACM 取得憑證的資訊，請參閱 AWS Certificate Manager 使用者指南。若要在中使用 ACM 憑證 CloudFront，請務必在美國東部 (維吉尼亞北部) 區域 () 申請 (或匯入) 憑證。`us-east-1`

CloudFront 支援與 Mozilla 相同的憑證授權單位 (CA)，因此如果您不使用 ACM，請在 Mozilla 內含的 CA 憑證清單上使用 CA 核發的憑證。如需有關如何取得和安裝憑證的詳細資訊，請參閱 HTTP 伺服器軟體的文件和 CA 的文件。

AWS 區域 為 AWS Certificate Manager

若要在 AWS Certificate Manager (ACM) 中使用憑證以在檢視者之間要求 HTTPSCloudFront，請確定您在美國東部 (維吉尼亞北部) 區域 () 要求 (或匯入) 憑證。`us-east-1`

如果您想要在原始伺服器之間 CloudFront 需要 HTTPS，並且在 Elastic Load Balancing 中使用負載平衡器作為來源，則可以在 any 中要求或匯入憑證 AWS 區域。

憑證格式

憑證必須為 X.509 PEM 格式。如果您使用 AWS Certificate Manager，這是預設格式。

中繼憑證

如果您使用第三方憑證授權機構 (CA)，請在列出位於 .pem 檔案的憑證鏈中的所有中繼憑證，從為您網域簽屬憑證的 CA 開始。一般而言，您可以在 CA 網站上找到以適當鏈結順序列出中繼和根憑證的檔案。

> **Important**
>
> 不包括下列項目：根憑證、不在信任路徑的中繼憑證，或您 CA 的公有金鑰憑證。

範例如下：

```
-----BEGIN CERTIFICATE-----
Intermediate certificate 2
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Intermediate certificate 1
-----END CERTIFICATE-----
```

Key type

CloudFront 支援 RSA 和 ECDSA 的公開私密金鑰配對。

CloudFront 支援使用 RSA 和 ECDSA 憑證與檢視器和來源的 HTTPS 連線。使用 AWS Certificate Manager (ACM)，您可以要求並匯入 RSA 或 ECDSA 憑證，然後將它們與您的散發產生關聯。CloudFront

如需您可以在 HTTPS 連線中進行交涉所支援的 RSA 和 ECDSA 密碼清單 CloudFront ，請參閱和。the section called “檢視器與之間支援的通訊協定和密碼 CloudFront” the section called “與來源之間支援的通訊協定 CloudFront 和密碼”

私有金鑰

如果您從第三方憑證授權單位 (CA) 使用憑證，請注意以下事項：

- 私有金鑰必須符合憑證中的公有金鑰。
- 私有金鑰必須是 PEM 格式。
- 無法用密碼加密私有金鑰。

如果 AWS Certificate Manager (ACM) 提供憑證，ACM 就不會釋放私密金鑰。私密金鑰會儲存在 ACM 中，供與 ACM 整合的 AWS 服務使用。

許可

您必須具有使用和匯入 SSL/TLS 憑證的許可。如果您使用的是 AWS Certificate Manager (ACM)，建議您使用 AWS Identity and Access Management 權限來限制憑證的存取權。如需詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的 Identity and Access Management。

憑證金鑰的大小

CloudFront 支援的憑證金鑰大小取決於金鑰和憑證的類型。

對於 RSA 憑證：

CloudFront 支援 1024 位元、2048 位元和 3072 位元和 4096 位元 RSA 金鑰。搭配使用之 RSA 憑證的金鑰長度上限為 4096 CloudFront 位元。

請注意，ACM 會發出具有多達 2048 位元金鑰的 RSA 憑證。若要使用 3072 位元或 4096 位元的 RSA 憑證，您需要從外部取得憑證並將其匯入 ACM，之後就可以與您搭配使用。 CloudFront

如需有關如何判斷 RSA 金鑰大小的詳細資訊，請參閱判斷 SSL/TLS RSA 憑證中公有金鑰的大小。

對於 ECDSA 憑證：

CloudFront 支援 256 位元金鑰。若要在 ACM 中使用 ECDSA 憑證以在檢視器之間要求以及之間使用 HTTPS CloudFront，請使用最初的 256v1 橢圓曲線。

支援的憑證類型

CloudFront 支援受信任憑證授權單位所發行的所有憑證類型。

憑證過期日期和續約

如果您使用的是從協力廠商憑證授權單位 (CA) 取得的憑證，則必須監控憑證到期日期，並更新您匯入 AWS Certificate Manager (ACM) 的憑證，或在憑證存放區到期前上傳到 AWS Identity and Access Management 憑證存放區。

如果使用 ACM 提供的憑證，ACM 會為您管理憑證續約。如需詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的受管續約。

CloudFront 發行版和憑證中的網域名稱

當您使用自訂原始伺服器時，原始伺服器上的 SSL/TLS 憑證包含 Common Name (通用名稱) 欄位中的網域名稱，且在 Subject Alternative Names (主體別名) 欄位中可能還有更多網域名稱。(在憑證網域名稱中CloudFront 支援萬用字元。)

憑證中其中一個網域名稱必須符合您指定給原始網域名稱的網域名稱。如果沒有相符的網域名稱，則會將 HTTP 狀態碼 CloudFront 傳回`502 (Bad Gateway)`給檢視器。

> **Important**
>
> 當您將替代網域名稱新增至分發時，請 CloudFront 檢查您所附加的憑證是否涵蓋替代網域名稱。憑證必須涵蓋憑證主體別名 (SAN) 欄位中的備用網域名稱。這表示 SAN 欄位必須包含完全相符的替代網域名稱，或在您要新增之替代網域名稱的相同層級包含萬用字元。
> 如需詳細資訊，請參閱 使用備用網域名稱的需求。

最低 SSL/TLS 通訊協定版本

如果您使用專用 IP 位址，請為檢視者之間的連線設定最低 SSL/TLS 通訊協定版本，並 CloudFront 選擇安全性原則。

如需詳細資訊，請參閱 發佈設定參考 主題中的 安全性政策 (最低 SSL/TLS 版本)。

支援的 HTTP 版本

如果您將一個憑證與多個 CloudFront 發行版產生關聯，則與憑證相關聯的所有發行版都必須使用相同的選項支援的 HTTP 版本。您可以在建立或更新 CloudFront 發佈時指定此選項。

搭配使用 SSL/TLS 憑證的配額 CloudFront (檢視者與僅限檢視者之間使用 HTTPS) CloudFront

請注意下列配額 (先前稱為限制) 搭配使用 SSL/TLS 憑證。CloudFront這些配額僅適用於您使用 AWS Certificate Manager (ACM) 佈建的 SSL/TLS 憑證、匯入 ACM，或上傳至 IAM 憑證存放區，以便在檢視者與之間進行 HTTPS 通訊。 CloudFront

每個 CloudFront 散發的最大憑證數目

您最多可以將一個 SSL/TLS 憑證與每個 CloudFront發行版產生關聯。

您可以匯入 ACM 或上傳至 IAM 憑證存放區的憑證上限數量

如果您從第三方 CA 取得 SSL/TLS 憑證，您必須在以下其中一個位置存放憑證：

- AWS Certificate Manager – 如需 ACM 憑證數量的目前配額，請參閱 AWS Certificate Manager 使用者指南中的配額。列出的配額是包含您使用 ACM 佈建的憑證與匯入 ACM 的憑證之總和。
- IAM 憑證存放區 — 有關可上傳到 AWS 帳戶 IAM 憑證存放區的憑證數目的目前配額 (先前稱為限制)，請參閱 IAM 使用者指南中的 IAM 和 STS 限制。您可以在 AWS Management Console 中請求更高的配額。

每個 AWS 帳戶的最大憑證數目 (僅限專用 IP 位址)

如果您想要使用專用 IP 地址提供 HTTPS 請求，請注意以下事項：

- 默認情況下，允許 CloudFront 您在您的 AWS 帳戶中使用兩個證書，一個用於日常使用，另一個用於需要輪換多個發行版的證書時。
- 如果您的 AWS 帳戶需要兩個以上的自訂 SSL/TLS 憑證，請前往 Sup port 中心並建立案例。請指出您需要許可才能使用的憑證有多少，以及說明請求中的情況。我們會儘快更新您的帳戶。

針對使用不同 AWS 帳戶建立的 CloudFront 發行版使用相同的憑證

如果您使用的是第三方 CA，並且想要將相同的憑證與使用不同 AWS 帳戶建立的多個CloudFront 發行版搭配使用，則必須將憑證匯入 ACM，或將其上傳至每個 AWS 帳戶一次 IAM 憑證存放區。

如果您使用 ACM 提供的憑證，則無法設定CloudFront 為使用由其他 AWS 帳戶建立的憑證。

針 CloudFront 對其他 AWS 服務使用相同的憑證

如果您從受信任的憑證授權單位 (例如 Comodo DigiCert、或賽門鐵克) 購買憑證，您可以針 CloudFront 對其他 AWS 服務使用相同的憑證。如果將憑證匯入到 ACM，則只需要匯入一次，即可將憑證用於多個 AWS 服務。

如果您使用 ACM 提供的憑證，則在 ACM 中存放憑證。

針對多個 CloudFront 發行版使用相同的憑證

您可以將相同的憑證用於服務 HTTPS 要求的任何或所有 CloudFront發行版。注意下列事項：

- 您可以使用相同的憑證提供使用專用 IP 地址的請求，與提供使用 SNI 的請求。
- 您僅能把一個 SSL/TLS 憑證與每個 &CF; 分佈相關聯。
- 每個分佈必須包含一或多個也出現在憑證中 Common Name (通用名稱) 欄位或 Subject Alternative Names (主體別名) 欄位的備用網域名稱。

- 如果您使用專用 IP 地址提供 HTTPS 請求，並且使用相同的 AWS 帳戶創建了所有分發，則可以通過對所有發行版使用相同的證書來顯著降低成本。 CloudFront 每個憑證的費用，而非每個發行版本的費用。

  例如，假設您使用相同的 AWS 帳戶建立三個發行版，並且對所有三個發行版都使用相同的憑證。您只需支付使用專用 IP 地址的費用一次。

  但是，如果您使用專用 IP 地址提供 HTTPS 請求，並且使用相同的證書在不同的 AWS 帳戶中創建 CloudFront 分發，則每個帳戶都需要支付使用專用 IP 地址的費用。例如，如果您使用三個不同的 AWS 帳戶建立三個發行版，並且對所有三個發行版都使用相同的憑證，則每個帳戶都會針對使用專用 IP 位址收取全額費用。

設定備用網域名稱和 HTTPS

若要在檔案的 URL 中使用替代網域名稱，以及在檢視者之間使用 HTTPS CloudFront，請執行適用的程序。

主題

取得 SSL/TLS 憑證

如果您尚未擁有憑證，請取得一個 SSL/TLS 憑證。如需詳細資訊，請參閱適用的文件：

- 若要使用 AWS Certificate Manager (ACM) 提供的憑證，請參閱使用AWS Certificate Manager 者指南。然後跳至 更新您的 CloudFront 發行版。

  > **Note**
  >
  > 我們建議您使用 ACM 在 AWS 受管資源上佈建、管理和部署 SSL/TLS 憑證。您必須在美國東部 (維吉尼亞北部) 區域請求 ACM 憑證。

- 若要從第三方憑證授權單位 (CA) 取得憑證，請參閱憑證授權單位提供的文件。當您有憑證時，請繼續進行下一個程序。

匯入 SSL/TLS 憑證

如果您從第三方 CA 取得憑證，請將憑證匯入到 ACM 或上傳至 IAM 憑證存放區：

ACM (推薦)

ACM 讓您從 ACM 主控台以及透過程式設計方式匯入第三方憑證。如需將憑證匯入 ACM 的詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的將憑證匯入 AWS Certificate Manager。您必須在美國東部 (維吉尼亞北部) 區域匯入憑證。

IAM 憑證存放區

(不建議使用) 使用下列 AWS CLI 命令將第三方憑證上傳至 IAM 憑證存放區。

```
aws iam upload-server-certificate \
        --server-certificate-name CertificateName \
        --certificate-body file://public_key_certificate_file \
        --private-key file://privatekey.pem \
        --certificate-chain file://certificate_chain_file \
        --path /cloudfront/path/
```

注意下列事項：

- AWS 帳戶 — 您必須使用您用來建立 CloudFront 分發的相同 AWS 帳戶，將憑證上傳至 IAM 憑證存放區。
- --path 參數 – 當您將憑證上傳到 IAM 時，`--path` 參數 (憑證路徑) 的值必須以 `/cloudfront/` 開頭，例如 `/cloudfront/production/` 或 `/cloudfront/test/`。路徑必須以 / 結尾。
- 現有的憑證 – 您必須指定 `--server-certificate-name` 和 `--path` 參數的值 (不同於與現有憑證相關聯的值)。
- 使用 CloudFront 主控台 — 您在中為`--server-certificate-name`參數指定的值 (例如) 會顯示在 CloudFront主控台的「SSL 憑證」清單中。 AWS CLImyServerCertificate
- 使用 CloudFront API — 記下 AWS CLI 傳回的英數字串，例如`AS1A2M3P4L5E67SIIXR3J`。這是您會在 `IAMCertificateId` 元素中指定的值。您不需要也由 CLI 傳回的 IAM ARN。

若要取得有關的更多資訊 AWS CLI，請參閱《AWS Command Line Interface 使用指南》和《指AWS CLI令參考》。

更新您的 CloudFront 發行版

若要更新分佈的設定，請執行以下程序：

若要設定替代網域名稱的 CloudFront 散佈

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您希望更新的分佈 ID。
3. 在 General (一般) 索引標籤上，選擇 Edit (編輯)。
4. 更新下列的值：

   備用網域名稱 (CNAME)

   選擇 [新增項目] 以新增適用的替代網域名稱。使用逗號區隔網域名稱，或在新的一行輸入每個網域名稱。

   自訂 SSL 憑證

   從下拉式清單中選取憑證。

   此處列出高達 100 個憑證。如果您有超過 100 個憑證，而您沒有看到想要新增的憑證，可以在欄位中輸入憑證 ARN 來選取。

   如果您將憑證上傳至 IAM 憑證存放區但它沒有被列出，而您無法在欄位中輸入名稱來選取，請檢閱此程序 匯入 SSL/TLS 憑證 以確認您是否正確上傳憑證。

   > **Important**
   >
   > 將 SSL/TLS 憑證與CloudFront 分發產生關聯後，請勿從 ACM 或 IAM 憑證存放區刪除憑證，直到您從所有發行版中移除憑證並部署所有發行版。

5. 選擇儲存變更。
6. 設定 CloudFront 為在檢視者之間需要 HTTPS，以及 CloudFront：

   a. 在 Behaviors (行為) 索引標籤中，選擇您想要更新的快取行為，然後選擇 Edit (編輯)。

   b. 請針對 Viewer Protocol Policy (檢視器通訊協定政策) 指定下列其中一個值：

      重新引導 HTTP 到 HTTPS

      檢視者可以使用這兩種通訊協定，但 HTTP 要求會自動重新導向至 HTTPS 要求。CloudFront會傳回 HTTP 狀態碼以`301 (Moved Permanently)`及新的 HTTPS 網址。檢視器接著會 CloudFront 使用 HTTPS URL 將要求重新提交至。

> **Important**
>
> CloudFront 不會將DELETE、、OPTIONSPATCHPOST、或PUT要求從 HTTP 重新導向至 HTTPS。如果您將快取行為設定為重新導向至 HTTPS，請使用 HTTP DELETE 狀態碼 CloudFront 回應該快取行為的 HTTP POST、、、或PUT要求403 (Forbidden)。OPTIONS PATCH

當檢視者發出重新導向至 HTTPS 要求的 HTTP 要求時，兩個要求都會 CloudFront 收取費用。對於 HTTP 請求，費用僅適用於請求以及 CloudFront返回給檢視器的標頭。針對 HTTPS 請求，會收取該請求與標頭及原始伺服器傳回的檔案之費用。

僅限 HTTPS

檢視器只能在使用 HTTPS 的情況下存取您的內容。如果檢視器傳送 HTTP 要求而非 HTTPS 要求，則會傳 CloudFront 回 HTTP 狀態碼，403 (Forbidden)而不會傳回檔案。

c. 請選擇 Yes, Edit (是，編輯)。

d. 針對您想要在檢視器和檢視器之間需要 HTTPS 的每個額外快取行為，重複步驟 a 到 c CloudFront。

7. 您使用生產環境中已更新的組態之前，請先確認以下項目：

- 每個快取行為中的路徑模式僅適用於您想要檢視器使用 HTTPS 的請求。
- 快取行為會以您要評估 CloudFront的順序列出。如需詳細資訊，請參閱 路徑模式。
- 快取行為會將請求路由到正確的原始伺服器。

判斷 SSL/TLS RSA 憑證中公有金鑰的大小

當您使用 CloudFront 替代網域名稱和 HTTPS 時，SSL/TLS RSA 憑證中公開金鑰的大小上限為 4096 位元。(這是金鑰大小，而不是公有金鑰中字元的數量)。如果您用 AWS Certificate Manager 於憑證，雖然 ACM 支援較大的 RSA 金鑰，但您無法搭配使用較大的金鑰。 CloudFront

您可以透過執行以下 OpenSSL 命令判斷 RSA 公有金鑰的大小：

```
openssl x509 -in path and filename of SSL/TLS certificate -text -noout
```

其中：

- `-in` 會指定 SSL/TLS RSA 憑證的路徑和檔案名稱。
- `-text` 會讓 OpenSSL 以位元為單位顯示 RSA 公有金鑰的長度。
- `-noout` 避免 OpenSSL 顯示公有金鑰。

輸出範例：

```
Public-Key: (2048 bit)
```

增加 SSL/TLS 憑證的配額

您可以匯入 AWS Certificate Manager 或上傳到 AWS Identity and Access Management 的 SSL/TLS 憑證，有數量上的配額 (先前稱為限制)。當您設定使用專用 IP 位址 CloudFront 來提供 HTTPS 要求時，可與 AWS 帳戶搭配使用的 SSL/TLS 憑證數量也會有配額。不過，您可以請求提高配額。

主題

- 您可以匯入至 ACM 的憑證
- 您可以上傳至 IAM 的憑證
- 您可以與專用 IP 地址搭配使用的憑證

您可以匯入至 ACM 的憑證

如需可匯入 ACM 之憑證數量的配額，請參閱 AWS Certificate Manager 使用者指南中的配額。

若要請求提高配額，請在支援中心主控台建立案例。指定下列值：

- 接受 Service limit increase (提高服務配額) 的預設值。
- 在 Limit type (配額類型) 中，選擇 Certificate Manager。
- 在「區域」中，選擇您要匯入憑證的 AWS 區域。
- 對於 Limit (限制)，選擇 Number of ACM certificates (ACM 憑證的數量)。

然後填寫表單的其餘部分，並提交表單。

您可以上傳至 IAM 的憑證

如需您可以上傳到 IAM 的憑證數量配額 (先前稱為限制)，請參閱 IAM 使用者指南中的 IAM 和 STS 限制。

若要請求提高配額，請在支援中心主控台建立案例。指定下列值：

- 接受 Service limit increase (提高服務配額) 的預設值。
- 在 Limit type (配額類型) 中，選擇 Certificate Manager。
- 在「區域」中，選擇您要匯入憑證的 AWS 區域。
- 對於 Limit (限制)，選擇 Server Certificate Limit (IAM) (伺服器憑證限制 (IAM))。

然後填寫表單的其餘部分，並提交表單。

您可以與專用 IP 地址搭配使用的憑證

如需使用專用 IP 位址提供 HTTPS 要求時可用於每個 AWS 帳戶的 SSL 憑證數量配額 (先前稱為限制)，請參閱SSL 憑證的配額。

若要請求提高配額，請在支援中心主控台建立案例。指定下列值：

- 接受 Service limit increase (提高服務配額) 的預設值。
- 針對「限制型態」，選擇 CloudFront「分配」
- 對於 Limit (限制)，選擇 Dedicated IP SSL Certificate Limit per Account (每個帳戶的專用 IP SSL 憑證限制)。

然後填寫表單的其餘部分，並提交表單。

輪換 SSL/TLS 憑證

如果您使用 AWS Certificate Manager (ACM) 提供的憑證，則不需要輪換 SSL/TLS 憑證。ACM 會為您管理憑證續約。如需詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的受管續約。

> **Note**
>
> ACM 不會為您從第三方憑證授權機構取得並匯入到 ACM 的憑證管理憑證續約。

如果您使用第三方憑證機構，而且已將憑證匯入 ACM (建議) 或上傳到 IAM 憑證存放區，則必須偶爾使用另一個憑證來替換某個憑證。例如，您必須在憑證過期日期接近時更換憑證。

> **Important**
>
> 如果您設定 CloudFront 使用專用 IP 位址來提供 HTTPS 要求，則在輪換憑證時使用一或多個其他憑證可能會產生額外的按比例計費。我們建議您盡快更新分佈，把額外的費用降到最低。

若要更換憑證，請執行以下程序。在您更換憑證的同時以及處理完成後，檢視器皆能持續存取您的內容。

若要輪換 SSL/TLS 憑證

1. 增加 SSL/TLS 憑證的配額 判斷您是否需要使用更多 SSL 憑證的許可。若是如此，請請求許可並等到授予許可時，再繼續步驟 2。
2. 將新憑證匯入到 ACM 或上傳至 IAM。如需詳細資訊，請參閱 Amazon CloudFront 開發人員指南中的匯入 SSL/TLS 憑證。
3. 一次更新一個分佈，以使用新的憑證。如需詳細資訊，請參閱 Amazon CloudFront 開發人員指南中的列出、檢視和更新發 CloudFront 佈。
4. (選擇性) 更新所有 CloudFront 發行版後，您可以從 ACM 或 IAM 刪除舊憑證。

> **Important**
>
> 請勿刪除 SSL/TLS 憑證，直到您將其從所有分佈中移除，並直到您已更新的分佈已變更為 `Deployed`。

從自訂 SSL/TLS 憑證還原為預設憑證 CloudFront

如果您設定 CloudFront 為在檢視器和之間使用 HTTPS CloudFront，並且設定CloudFront 為使用自訂 SSL/TLS 憑證，則可以變更組態以使用預設 CloudFront SSL/TLS 憑證。此程序取決於您是否已使用分佈來分配您的內容：

- 如果您尚未使用分佈來分配內容，可以只變更組態。如需詳細資訊，請參閱 更新分佈。
- 如果您已經使用分發來散佈內容，則必須建立新的 CloudFront分發並變更檔案的 URL，以減少或減少內容無法使用的時間。若要執行此操作，請執行以下程序。

回復為預設 CloudFront 憑證

1. 使用所需的 CloudFront 配置創建一個新的發行版。針對「SSL 憑證」，請選擇「預設 CloudFront 憑證 (*.cloudfron t.net)」。

   如需詳細資訊，請參閱 建立、更新和刪除分發。

2. 對於您正在使用發佈的檔案 CloudFront，請更新應用程式中的 URL，以使用 CloudFront 指派給新發行版的網域名稱。例如，請將 `https://www.example.com/images/logo.png` 變更為 `https://d111111abcdef8.cloudfront.net/images/logo.png`。

3. 刪除與自訂 SSL/TLS 憑證相關聯的散佈，或更新散佈以將 SSL 憑證的值變更為預設 CloudFront 憑證 (*. cloudfront.net)。如需詳細資訊，請參閱 更新分佈。

   > **Important**
   >
   > 在您完成此步驟之前，會 AWS 繼續向您收取使用自訂 SSL/TLS 憑證的費用。

4. (選用) 刪除您的自訂 SSL/TLS 憑證。

   a. 執行 AWS CLI 命令`list-server-certificates`以取得您要刪除之憑證的憑證 ID。若要取得更多資訊，請參閱《指AWS CLI 令參考》list-server-certificates中的。

   b. 執行 AWS CLI 命令`delete-server-certificate`以刪除憑證。若要取得更多資訊，請參閱《指AWS CLI 令參考》delete-server-certificate中的。

從具有專用 IP 地址的自訂 SSL/TLS 憑證切換到 SNI

如果您設定 CloudFront 使用具有專用 IP 位址的自訂 SSL/TLS 憑證，則可以改用搭配 SNI 使用自訂 SSL/TLS 憑證，並免除與專用 IP 位址相關聯的費用。下列程序將告訴您如何做到。

> **Important**
>
> 此 CloudFront 設定更新不會影響支援 SNI 的檢視器。檢視者可以在變更前後存取您的內容，也可以在變更傳播至節 CloudFront 點時存取您的內容。不支援 SNI 的檢視器，在變更之後不能存取您的內容。如需詳細資訊，請參閱 選擇如何 CloudFront 提供 HTTPS 要求。

若要從具有專用 IP 地址的自訂 SSL/TLS 憑證切換到 SNI

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇欲檢閱或更新的分佈 ID。
3. 請選擇 Distribution Settings (分佈設定)。
4. 在 General (一般) 索引標籤上，選擇 Edit (編輯)。
5. 將 Custom SSL Client Support (自訂 SSL 用戶端支援) 的設定，變更為 Only Clients that Support Server Name Indication (SNI) (只限支援伺服器名稱指示 (SNI) 的用戶端)。
6. 請選擇 Yes, Edit (是，編輯)。

使用已簽署的 URL 和已簽署的 Cookie 提供私有內容

許多透過網際網路分配內容的公司想要限制對所選使用者的文件、業務資料、媒體串流，或內容的存取許可，例如已付費的使用者。若要使用安全地提供此私人內容 CloudFront，您可以執行下列動作：

- 要求您的使用者使用特殊 CloudFront 簽署的 URL 或已簽署的 Cookie 存取您的私人內容。
- 要求使用者使用 CloudFront URL 存取您的內容，而不是直接存取原始伺服器 (例如 Amazon S3 或私有 HTTP 伺服器) 上內容的 URL。不需要 CloudFront URL，但我們建議您避免使用者略過您在已簽署 URL 或已簽署 Cookie 中指定的限制。

主題

提供私有內容服務的概觀

您可以透過兩種方式控制使用者對私人內容的存取權：

- 限制對 CloudFront 緩存中文件的訪問。
- 執行以下其中一項以在您的原始伺服器中限制存取檔案：
  - 為您的 Amazon S3 儲存貯體設定原始存取控制 (OAC)。
  - 為私有 HTTP 伺服器 (自訂原始伺服器) 設定自訂標頭。

限制對快取中檔案的 CloudFront存取

您可以設定 CloudFront 為要求使用者使用已簽署的 URL 或已簽署的 Cookie 存取您的檔案。然後，您可以開發應用程式，以建立和分配已簽署的 URL 給驗證過的使用者，或傳送 `Set-Cookie` 標頭，這是經驗證之使用者設定的已簽署 Cookie。(若要提供一些使用者長期存取少量檔案，您還可以手動建立簽章的 URL。)

當您建立已簽章的 URL 或已簽章的 Cookie，以控制存取您的檔案時，可以指定下列限制：

- 結束日期和時間，之後 URL 不再有效。
- (選用) URL 生效的日期和時間。
- (選用) 可用於存取內容的電腦的 IP 地址或地址範圍。

使用公有-私有金鑰對中的私有金鑰對已簽署的 URL 或已簽署的 Cookie 的一部分進行雜湊和簽名。當有人使用已簽署的 URL 或已簽署的 Cookie 存取檔案時，請 CloudFront 比較 URL 或 Cookie 的已簽署和未簽署部分。如果它們不匹配，則 CloudFront 不提供文件。

您必須使用 RSA-SHA1 來簽署網址或餅乾。 CloudFront 不接受其他算法。

限制對 Amazon S3 儲存貯體中檔案的存取

您可以選擇性地保護 Amazon S3 儲存貯體中的內容，以便使用者可以透過指定的 CloudFront 分發存取內容，但無法使用 Amazon S3 URL 直接存取內容。這樣可以防止某人繞過 CloudFront 並使用 Amazon S3 URL 來取得您想要限制存取的內容。使用簽章 URL 不需要此步驟，但我們建議執行此作業。

若要求使用者透過 CloudFront URL 存取您的內容，請執行下列工作：

- 授予 CloudFront 原始存取控制權限，以讀取 S3 儲存貯體中的檔案。

- 建立來源存取控制，並將其與您的 CloudFront 發行版產生關聯。
- 移除其他人使用 Amazon S3 URL 讀取檔案的許可。

如需詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。

在自訂原始伺服器上限制存取檔案

如果您使用自訂原始伺服器，您可以選擇性設定自訂標頭來限制存取。 CloudFront 若要從自訂原始檔案取得檔案，必須CloudFront 使用標準 HTTP (或 HTTPS) 要求來存取檔案。但是，通過使用自定義標題，您可以進一步限制對內容的訪問，以便用戶只能通過訪問它CloudFront，而不是直接訪問它。使用簽章 URL 不需要此步驟，但我們建議執行此作業。

若要要求使用者透過存取內容 CloudFront，請變更 CloudFront 發行版中的下列設定：

原始伺服器自訂標頭

設定 CloudFront 為將自訂標頭轉寄至您的來源。請參閱 設定 CloudFront 為將自訂標頭新增至原始請求。

檢視器通訊協定政策

將您的散佈設定為要求檢視者使用 HTTPS 存取 CloudFront。請參閱 檢視器通訊協定政策。

原始伺服器通訊協定政策

將您的發行版設定為 CloudFront 要求使用與檢視者相同的通訊協定，將請求轉寄至來源。請參閱通訊協定 (僅限自訂原始伺服器)。

完成這些變更後，請在自訂來源上更新您的應用程式，以僅接受包含您設定 CloudFront 要傳送之自訂標頭的要求。

檢視器通訊協定政策與原始伺服器通訊協定政策的組合可確保自訂標頭在傳輸過程中加密。但是，我們建議您定期執行以下操作，以旋轉轉 CloudFront 發到您的來源的自定義標題：

1. 更新您的 CloudFront 發行版以開始將新標題轉發到您的自定義來源。
2. 更新您的應用程式以接受新的標頭，以確認要求來自 CloudFront。
3. 當要求不再包含您要取代的標頭時，請將應用程式更新為不再接受舊標頭，以確認要求來源 CloudFront。

任務清單：提供私有內容服務

若要設定 CloudFront 為提供私人內容，請執行下列工作：

1. （可選，但建議使用）要求您的用戶只能通過訪問您的內容 CloudFront。您使用的方法取決於您使用的是 Amazon S3 或自訂原始伺服器：

   - Amazon S3 – 請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。
   - 自訂原始伺服器 – 請參閱 在自訂原始伺服器上限制存取檔案。

   自訂原始伺服器包括 Amazon EC2、設定為網站端點的 Amazon S3 儲存貯體、Elastic Load Balancing，以及您自己的 HTTP Web 伺服器。
2. 指定您要用來建立已簽署 URL 或已簽署 Cookie 的信任金鑰群組或可信簽署者。建議您使用信任的金鑰群組。如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。
3. 撰寫您的應用程式，使用已簽章的 URL或使用已設定簽章的 Cookie 的 `Set-Cookie` 標頭，以回應來自已獲授權使用者的請求。請遵循下列其中一個主題中的步驟進行：

   - 使用已簽署 URL
   - 使用已簽署 Cookie

   如果您不確定要使用哪個方法，請參閱 在已簽署 URL 和已簽署 Cookie 之間進行選擇。

指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者

主題

- 在信任的金鑰群組 (建議使用) 和 AWS 帳戶之間選擇
- 為您的簽署者建立金鑰對
- 重新格式化私有金鑰 (僅限 .NET 和 Java)
- 將簽署者新增至分佈
- 輪換金鑰對

若要建立已簽署 URL 或已簽署 Cookie，您需要 簽署者。簽署者可以是您在其中建立的受信任金鑰群組 CloudFront，或是包含 CloudFront key pair 組的 AWS 帳戶。建議您使用具有已簽署 URL 和已簽

署 Cookie 信任的金鑰群組。如需詳細資訊，請參閱 在信任的金鑰群組 (建議使用) 和 AWS 帳戶之間選擇。

簽署者有兩個用途：

- 一旦您將簽署者新增至您的發佈，就會 CloudFront 開始要求檢視者使用已簽署的 URL 或已簽署的 Cookie 來存取您的檔案。
- 當您建立已簽署 URL 或已簽署 Cookie 時，您可以使用簽署者金鑰對中的私有金鑰對 URL 或 Cookie 的一部分進行簽署。當有人要求受限制的檔案時，請 CloudFront 將 URL 或 Cookie 中的簽名與未簽署的 URL 或 Cookie 進行比較，以確認檔案沒有遭到竄改。 CloudFront 還會驗證 URL 或 Cookie 是否有效，這意味著到期日期和時間尚未超過。

當您指定簽署者時，您也可以透過將簽署者新增至快取行為，以間接指定需要簽署 URL 或簽署 Cookie 的檔案。如果分佈只有一個快取行為，則使用者必須使用已簽署 URL 或已簽署 Cookie 來存取該分佈中的任何檔案。如果您建立多個快取行為，並將簽署者新增到一些快取行為，而非其他快取行為，則可以請求檢視器使用已簽署 URL 或已簽署 Cookie 來存取這些物件，而非其他檔案。

若要指定允許建立已簽署 URL 或已簽署 Cookie 的簽署者 (私密金鑰)，並將簽署者新增至您的 CloudFront 分發，請執行下列工作：

1. 決定要使用受信任的金鑰群組還是 AWS 帳戶做為簽署者。建議您使用信任的金鑰群組。如需詳細資訊，請參閱 在信任的金鑰群組 (建議使用) 和 AWS 帳戶之間選擇。
2. 針對您在步驟 1 中選擇的簽署者，建立公有-私有金鑰對。如需詳細資訊，請參閱 為您的簽署者建立金鑰對。
3. 如果您使用 .NET 或 Java 建立已簽署 URL 或已簽署 Cookie，請重新格式化私有金鑰。如需詳細資訊，請參閱 重新格式化私有金鑰 (僅限 .NET 和 Java)。
4. 在您要為其建立已簽署 URL 或已簽署 Cookie 的分佈中，指定簽署者。如需詳細資訊，請參閱 將簽署者新增至分佈。

在信任的金鑰群組 (建議使用) 和 AWS 帳戶之間選擇

若要使用已簽署 URL 或已簽署 Cookie，您需要 簽署者。簽署者可以是您在其中建立的受信任金鑰群組 CloudFront，或是包含 CloudFront key pair 組的 AWS 帳戶。基於下列原因，建議您使用信任的金鑰群組：

- 使用金 CloudFront 鑰群組時，您不需要使用 AWS 帳戶根使用者來管理 CloudFront 已簽署 URL 和已簽署 Cookie 的公開金鑰。 AWS 最佳做法建議您不要在不需要時使用 root 使用者。

- 透過金 CloudFront 鑰群組，您可以使用 CloudFront API 管理公開金鑰、金鑰群組和受信任的簽署者。您可以使用 API 自動化金鑰建立和金鑰輪換。當您使用 AWS root 使用者時，您必須使用 AWS Management Console 來管理CloudFront 金鑰配對，因此您無法自動化處理程序。
- 由於您可以使用 CloudFront API 管理金鑰群組，因此也可以使用 AWS Identity and Access Management (IAM) 許可政策來限制不同使用者可執行的動作。例如，您可以允許使用者上傳公有金鑰，但不能刪除公有金鑰。或者，您可以允許使用者刪除公有金鑰，但只有在符合特定條件時，例如使用多因素驗證、從特定網路傳送請求，或是在特定日期和時間範圍內傳送請求。
- 透過金 CloudFront 鑰群組，您可以將更多數目的公開金鑰與 CloudFront 發行版產生關聯，讓您在使用和管理公開金鑰方面有更大的彈性。根據預設，您最多可以將四個金鑰群組與單一分佈產生關聯，而且一個金鑰群組中最多可以有五個公有金鑰。

  當您使用 AWS 帳號 root 使用者管理 CloudFront 金鑰配對時，每個帳 AWS 戶最多只能有兩個作用中 CloudFront 金鑰配對。

為您的簽署者建立金鑰對

您用來建立已 CloudFront 簽署網址或已簽署 Cookie 的每位簽署者都必須有公開私密 key pair。簽署者會使用其私密金鑰來簽署 URL 或 Cookie，並 CloudFront 使用公開金鑰驗證簽章。

建立金鑰組的方式取決於您使用受信任的金鑰群組做為簽署者 (建議使用)，還是使用 CloudFront key pair。如需詳細資訊，請參閱下列區段。您建立的金鑰對必須符合下列需求：

- 它必須是一個 SSH-2 RSA 金鑰對。
- 它必須是 base64 編碼的 PEM 格式。
- 它必須是 2048 位元金鑰對。

為了協助保護應用程式的安全，建議您定期輪換金鑰對。如需詳細資訊，請參閱 輪換金鑰對。

為信任的金鑰群組建立金鑰對 (建議使用)

若要為信任的金鑰群組建立金鑰對，請執行下列步驟：

1. 建立公有-私有金鑰對。
2. 將公開金鑰上傳至 CloudFront。
3. 將公開金鑰新增至 CloudFront 金鑰群組。

如需詳細資訊，請參閱下列程序。

建立一組金鑰對

> **Note**
>
> 下列步驟會使用 OpenSSL 做為建立金鑰對的一種方式來示範。還有許多其他方法可以建立 RSA 金鑰對。

1. 下列範例命令會使用 OpenSSL 產生長度為 2048 位元的 RSA 金鑰對，並儲存至名為 `private_key.pem` 的檔案。

   ```
   openssl genrsa -out private_key.pem 2048
   ```

2. 產生的檔案同時包含公有和私有金鑰。下列範例命令會從名為 `private_key.pem` 的檔案擷取公有金鑰。

   ```
   openssl rsa -pubout -in private_key.pem -out public_key.pem
   ```

   您稍後會在下列程序中上傳公有金鑰 (在 `public_key.pem` 檔案中)。

將公開金鑰上傳至 CloudFront

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽功能表中，選擇公有金鑰。
3. 選擇 [建立公開金鑰]。
4. 在「建立公用金鑰」視窗中，執行下列動作：

   a. 在金鑰名稱中，輸入識別公有金鑰的名稱。

   b. 對於 Key value (鍵值)，貼上公有金鑰。如果您遵循上述程序中的步驟，則公有金鑰位於名為 `public_key.pem` 的檔案中。若要複製並貼上公有金鑰的內容，您可以：

      - 在 macOS 或 Linux cat 命令列上使用這個命令，如下所示：

        ```
        cat public_key.pem
        ```

複製該命令的輸出，然後將其貼 Key value (鍵值) 欄位中。

- 使用純`public_key.pem`文字編輯器開啟檔案，例如記事本 (在 Windows 上) 或 TextEdit (在 macOS 上)。複製檔案內容，然後將其貼到 Key value (鍵值) 欄位中。

c. (選用) 對於註解，請新增註解以描述公有金鑰。

完成時，請選擇新增。

5. 記錄公有金鑰 ID。您稍後在建立已簽署 URL 或已簽署 Cookie 時使用它，以做為 `Key-Pair-Id` 欄位的值。

將公有金鑰新增至金鑰群組

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽功能表中，選擇金鑰群組。
3. 選擇新增金鑰群組。
4. 在建立金鑰群組頁面上，執行下列動作：

   a. 在金鑰群組名稱中，輸入識別金鑰群組的名稱。

   b. (選用) 在註解中，輸入註解以描述金鑰群組。

   c. 在公有金鑰中，選取要新增至金鑰群組的公有金鑰，然後選擇新增。針對您要新增至金鑰群組的每個公有金鑰重複此步驟。

5. 選擇建立金鑰對。
6. 記錄金鑰群組名稱。稍後您可以使用它來將金鑰群組與 CloudFront 散發中的快取行為建立關聯。在 CloudFront API 中，您可以使用金鑰群組 ID 將金鑰群組與快取行為建立關聯。)

建立 CloudFront key pair (不建議使用，需要 AWS 帳號 root 使用者)

> **Important**
>
> 建議您建立信任金鑰群組的公有金鑰，而不是遵循下列步驟。如需建立已簽署 URL 和已簽署 Cookie 之公有金鑰的建議方式，請參閱為信任的金鑰群組建立金鑰對 (建議使用)。

您可以使用下列方式建立 CloudFront key pair：

- 在中建立 key pair AWS Management Console 並下載私密金鑰。請參見下列步驟：
- 使用諸如 OpenSSL 的應用程式建立 RSA 金鑰對，並將公有金鑰上傳到 AWS Management Console。如需有關建立 RSA 金鑰對的詳細資訊，請參閱 為信任的金鑰群組建立金鑰對 (建議使用)。

若要在中建立 CloudFront 金鑰配對 AWS Management Console

1. 使用 AWS 帳戶根 AWS Management Console 使用者的認證登入。

   > **Important**
   >
   > IAM 使用者無法建立 CloudFront 金鑰配對。您必須使用根使用者登入資料登入，才能建立金鑰對。

2. 選擇您的帳戶名稱，然後選擇我的安全登入資料。
3. 選擇CloudFront 金鑰配對。
4. 確認您沒有多個作用中的金鑰對。如果您已經有兩個使用中的金鑰對，則無法建立金鑰對。
5. 選擇建立新的金鑰對。

   > **Note**
   >
   > 您也可以選擇建立自己的 key pair 並上傳公開金鑰。 CloudFront 金鑰配對支援 1024、2048 位元或 4096 位元金鑰。

6. 在建立金鑰對的對話方塊中，選擇下載私有金鑰檔案，然後將檔案儲存在電腦上。

   > **Important**
   >
   > 將 key pair 的私密金 CloudFront 鑰儲存在安全位置，並設定檔案的權限，以便只有所需的管理員才能讀取。如果有人取得您的私有金鑰，他們可以產生有效已簽章的 URL 和已簽章的 Cookie 並下載您的內容。您無法再次取得私密金鑰，因此如果遺失或刪除私密金鑰，您必須建立新的 CloudFront key pair。

7. 記錄您的金鑰對的金鑰對 ID。(在中 AWS Management Console，這稱為「存取金鑰 ID」。) 您將在建立已簽署 URL 或已簽署 Cookie 時使用它。

重新格式化私有金鑰 (僅限 .NET 和 Java)

如果您正使用 .NET 或 Java 以建立已簽署 URL 或已簽署 Cookie，則無法在預設 PEM 格式中，使用金鑰對的私有金鑰來建立簽章。相反地，請執行下列動作：

- .NET 框架 – 將私有金鑰轉換為 .NET 框架使用的 XML 格式。有幾種工具可用。
- Java – 將私有金鑰轉換為 DER 格式。執行此操作的一種方法是使用以下 OpenSSL 命令。在下列命令中，`private_key.pem` 是包含 PEM 格式化的私有金鑰的檔案名稱，而且 `private_key.der` 是執行命令之後包含 DER 格式化的私有金鑰的檔案名稱。

```
openssl pkcs8 -topk8 -nocrypt -in private_key.pem -inform PEM -out private_key.der -
outform DER
```

  為了確保編碼器能正確運作，請將 Bouncy Castle Java 的密碼編譯 API 的 JAR 新增到專案中，然後再新增 Bouncy Castle 供應商。

將簽署者新增至分佈

簽署者是可信任的金鑰群組 (建議使用) 或 CloudFront key pair 組，可以建立已簽署的 URL 和已簽署的 Cookie 以供分發使用。若要在 CloudFront 分發中使用已簽署的 URL 或已簽署的 Cookie，您必須指定簽署者。

簽署者與快取行為相關聯。在某些物件或不用於相同分佈之其他檔案下，這允許您要求簽章 URL 或簽章 Cookie。只有和對應快取行為相關聯的檔案，分佈才會要求已簽署的 URL 或 Cookie。

同樣地，簽署者只能為與對應快取行為相關聯的檔案簽署 URL 或 Cookie。例如，如果您有一個快取行為下的簽署者和不同快取行為下的不同簽署者，則這兩個簽署者都不能為與其他快取行為的相關檔案建立簽署 URL 或 Cookie。

> **Important**
>
> 在您將簽署者新增至分佈之前，請執行下列動作：
>
> - 請仔細定義快取行為中的路徑模式，以及快取行為順序，這樣您就不會讓使用者意外地存取您的內容，也不會讓他們存取您希望所有人都能使用的內容。

例如，假設請求與兩個快取行為的路徑模式相符合。第一個快取行為不需要已簽章的 URL 或已簽章的 Cookie，而第二個快取行為則需要。因為CloudFront 處理與第一個相符項目相關聯的快取行為，因此使用者無需使用已簽署 URL 或已簽署的 Cookie 即可存取檔案。

如需喲歐冠路徑模式的詳細資訊，請參閱 路徑模式。

- 對於您已經用來分佈內容的分配內容，請在新增簽署者之前，確定您已準備好開始產生已簽署 URL 和已簽署 Cookie。當您新增簽署者時， CloudFront 會拒絕不包含有效簽署網址或已簽署 Cookie 的要求。

您可以使用 CloudFront 控制台或 CloudFront API 將簽署者添加到您的分發中。

主題

- 使用主控台將簽署者新增至發行 CloudFront 版
- 使用 API 將簽署者新增至發行 CloudFront 版

使用主控台將簽署者新增至發行 CloudFront 版

下列步驟說明如何將信任的金鑰群組新增為簽署者。您也可以將 AWS 帳戶新增為受信任的簽署者，但不建議這麼做。

使用主控台將簽署者新增至分佈

1. 記錄要用作信任的簽署者之金鑰群組的金鑰群組 ID。如需詳細資訊，請參閱 為信任的金鑰群組建立金鑰對 (建議使用)。
2. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
3. 使用已簽署 URL 或已簽署 Cookie，選擇要保護其檔案的分佈。

   > **Note**
   >
   > 若要將簽署者新增至新的發行套件，您可以指定建立發行套件時步驟 6 所述的相同設定。

4. 選擇 Behaviors (動作) 索引標籤。
5. 選取快取行為，其路徑模式符合您要使用已簽署 URL 或已簽署 Cookie 保護的檔案，然後選擇編輯。
6. 在編輯行為頁面上，執行下列動作：

a. 針對限制檢視器存取 (使用已簽署 URL 或已簽署 Cookie)，按一下是。
b. 針對 Trusted Key Groups or Trusted Signer (信任的金鑰群組或可信簽署者)，選擇 Trusted Key Groups (信任的金鑰群組)。
c. 在信任的金鑰群組中，選擇要新增的金鑰群組，然後選擇新增。如果您要新增多個金鑰群組，請重複此步驟。

7. 選擇是，編輯以更新快取行為。

使用 API 將簽署者新增至發行 CloudFront 版

您可以使用 CloudFront API 將受信任的金鑰群組新增為簽署者。您可以將簽署者新增至現有分佈或新分佈。在任一情況下，請在 `TrustedKeyGroups` 元素中指定值。

您也可以將 AWS 帳戶新增為受信任的簽署者，但不建議這麼做。

請參閱 Amazon CloudFront API 參考資料中的下列主題：

- 更新現有的發行版 — UpdateDistribution
- 建立新的發行版本 — CreateDistribution

輪換金鑰對

建議您定期輪換 (變更) 已簽署網址和已簽署 Cookie 的金鑰對。若要輪換您用來建立已簽署 URL 或已簽署 Cookie 的金鑰對，而不會使尚未到期的無效 URL 或 Cookie，請執行以下任務：

1. 建立新的金鑰對，並將公有金鑰新增至金鑰群組。如需詳細資訊，請參閱 為信任的金鑰群組建立金鑰對 (建議使用)。
2. 如果您在上一個步驟中建立了新的金鑰群組，請以簽署者的身分將金鑰群組新增到分佈中。

> **Important**
>
> 請勿從金鑰群組中移除任何現有的公有金鑰，或從分佈中移除任何金鑰群組。請僅加入新的。

3. 使用新金鑰對中的私有金鑰來更新應用程式以建立簽章。確認已簽署 URL 或使用新私有金鑰簽署的 Cookie 正在運作。
4. 等待直到已簽署 URL 或 Cookie 過期，而使用先前的金鑰對。然後從金鑰群組中移除舊的公有金鑰。如果您在步驟 2 中建立新的金鑰群組，請從分佈中移除舊的金鑰群組。

在已簽署 URL 和已簽署 Cookie 之間進行選擇

CloudFront 已簽署的網址和已簽署的 Cookie 提供相同的基本功能：它們可讓您控制哪些人可以存取您的內容。如果您想透過以下方式提供私人內容， CloudFront 並嘗試決定是否使用已簽署的 URL 或已簽署的 Cookie，請考慮下列事項。

在以下案例使用已簽章的 URL：

- 您想要限制對個別檔案的存取，例如，適用於您的應用程式安裝下載。
- 您的使用者正在使用不支援 Cookie 的用戶端 (例如，自訂 HTTP 用戶端)。

在以下案例使用已簽章的 Cookie：

- 您想要提供對多個限制檔案的存取，例如，HLS 格式視訊的所有檔案或網站中訂閱者區域的所有檔案。
- 您不想變更目前的 URL。

如果您目前未使用簽章的 URL，並且您的 (未簽署) URL 包含以下任何查詢字串參數，則不能使用已簽章的 URL 或已簽章的 Cookie：

- `Expires`
- `Policy`
- `Signature`
- `Key-Pair-Id`

CloudFront 假設包含任何這些查詢字串參數的 URL 是已簽署的 URL，因此不會查看已簽署的 Cookie。

使用已簽署 URL 和已簽署 Cookie

已簽署的網址優先於已簽署的 Cookie。如果您同時使用已簽署的 URL 和已簽署的 Cookie 來控制對相同檔案的存取，且檢視者使用已簽署的 URL 要求檔案，請 CloudFront 決定是否只根據已簽署的 URL 將檔案傳回給檢視者。

使用已簽署 URL

主題

- 在已簽署 URL 的標準和自訂原則之間進行選擇
- 已簽署 URL 的工作方式
- 選擇已簽署 URL 的有效時間
- 何時 CloudFront 檢查簽名 URL 中的到期日期和時間？
- 範例程式碼和第三方工具
- 使用標準政策建立簽署的 URL
- 使用自訂政策建立已簽署 URL

已簽章的 URL 包含附加資訊，例如到期日期和時間，以便您更有效地控制對內容的存取。此附加資訊顯示在政策聲明中，該政策聲明基於標準政策或自訂政策。標準和自訂政策之間的差異將在接下來的兩節中說明。

> **Note**
>
> 您可使用標準政策建立一些簽章的 URL，並使用自訂政策為相同的分佈建立一些簽章的 URL。

在已簽署 URL 的標準和自訂原則之間進行選擇

當您建立已簽章的 URL 時，您將編寫一個 JSON 格式的政策聲明來指定對已簽章的 URL 的限制，例如 URL 的有效時間。您可以使用標準政策或自訂政策。以下是標準和自訂政策的比較：

描述	標準政策	自訂政策
您可以重複使用多個檔案的政策聲明。要重複使用政策聲明，您必須在 `Resource` 物件中使用萬用字元。如需詳細資訊，請參閱 您在使用自訂政策的已簽署 URL 政策陳述式中指定的值)。	否	是
您可以指定使用者可以開始存取您的內容的日期和時間。	否	是 (選用)
您可以指定使用者無法再存取您的內容的日期和時間。	是	是

描述	標準政策	自訂政策
您可以指定可以存取您的內容的使用者的 IP 地址或 IP 地址範圍。	否	是 (選用)
該已簽章的 URL 包含該政策的 base64 編碼版本，這會導致較長的 URL。	否	是

如需使用標準政策建立已簽署的 URL 的詳細資訊，請參閱 使用標準政策建立簽署的 URL。

如需使用自訂政策建立已簽署的 URL 的詳細資訊，請參閱 使用自訂政策建立已簽署 URL。

已簽署 URL 的工作方式

以下概述如何針對已簽署的 URL 設定 CloudFront 和 Amazon S3，以及使用者使用已簽署的 URL 要求檔案時如何 CloudFront 回應。

1. 在您的 CloudFront 散發中，指定一或多個受信任的金鑰群組，其中包含 CloudFront 可用來驗證 URL 簽章的公開金鑰。您可以使用對應的私有金鑰來簽署 URL。

   如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。
2. 開發應用程式以判斷使用者是否應該有權存取您的內容，並為您所要限制存取的應用程式的檔案或某些部分建立的簽署 URL。如需詳細資訊，請參閱下列主題：

   - 使用標準政策建立簽署的 URL
   - 使用自訂政策建立已簽署 URL
3. 使用者請求檔案用於所要請求的簽章 URL。
4. 您的應用程式會驗證使用者是否有權存取檔案：他們已經登入，他們已經支付存取內容的費用，或者他們已經滿足其他一些存取要求。
5. 您的應用程式會建立和傳回已簽章的 URL 給使用者。
6. 已簽章的 URL 允許使用者下載或串流內容。

   這個步驟是自動的；使用者通常不需要執行任何額外操作來存取內容。例如，如果使用者在 Web 瀏覽器l中存取您的內容時，則應用程式會將已簽章的 URL 傳回到瀏覽器。瀏覽器會立即使用已簽署的 URL 存取 CloudFront Edge 快取中的檔案，而無需使用者進行任何操作。
7. CloudFront 使用公開金鑰來驗證簽章，並確認 URL 沒有遭到竄改。如果簽章無效，請求會遭到拒絕。

如果簽名有效，請查 CloudFront 看 URL 中的策略語句（如果您使用固定策略，則構建一個策略）以確認請求仍然有效。例如，如果您指定 URL 的開始和結束日期和時間，請 CloudFront 確認使用者在您要允許存取的期間內嘗試存取您的內容。

如果要求符合原則陳述式中的需求， CloudFront 會執行標準作業：判斷檔案是否已在 Edge 快取中，視需要將要求轉送至原始位置，然後將檔案傳回給使用者。

> **Note**
>
> 如果不帶正負號的 URL 包含查詢字串參數，請務必將這些參數包含在您已簽署 URL 部分中。如果在簽署已簽署 URL 之後，將查詢字串新增此 URL，則 URL 會傳回 HTTP 403 狀態。

選擇已簽署 URL 的有效時間

您可以分配私有內容以僅在短時間內使用已簽章的有效 URL (可能只有區區幾分鐘)。在短時間內有效的簽署網址適合用於on-the-fly 將內容發佈給使用者特定用途，例如依需求將電影租借或音樂下載內容發佈給客戶。如果已簽章的 URL 僅在短時間內有效，您可能會希望使用您開發的應用程式自動產生他們。當使用者開始下載檔案或開始播放媒體檔案時，請 CloudFront 比較 URL 中的到期時間與目前時間，以判斷 URL 是否仍然有效。

您還可以使用有效期較長的 (可能會持續數年) 已簽章的 URL 分配私有內容。有效期較長的已簽章的 URL 對於向已知使用者分佈私有內容非常有用，例如向投資者分佈業務計畫或向員工分佈培訓教材。您可以開發應用程式，為您產生這些長期簽署的 URL。

何時 CloudFront 檢查簽名 URL 中的到期日期和時間？

CloudFront 在發出 HTTP 要求時，檢查已簽署 URL 中的到期日期和時間。如果用戶端在到期前一刻才開始下載大型檔案，則即使在下載期間過期了，下載也應該要完成。如果 TCP 連線中斷並且用戶端在到期時間過後嘗試重新啟動下載，則下載將失敗。

如果用戶端使用範圍 GET 以取得較小型的檔案，則到期時間過後發生的任何 GET 請求都將失敗。如需範圍 GET 的詳細資訊，請參閱 如何 CloudFront 處理對象的部分請求（範圍 GET）。

範例程式碼和第三方工具

如需建立已簽署 URL 的雜湊和已簽署部分的範例程式碼，請參閱下列主題：

- 使用 Perl 建立 URL 簽章

- 使用 PHP 建立 URL 簽章
- 使用 C# 和 .NET 架構建立 URL 簽章
- 使用 Java 建立 URL 簽章

使用標準政策建立簽署的 URL

若要使用標準政策建立已簽署 URL，請完成以下步驟。

使用標準政策建立簽章的 URL

1. 如果您使用 .NET 或 Java 建立簽章的 URL，並且您還沒有將金鑰對的私有金鑰從預設的 .pem 格式重新格式化為與 .NET 或 Java 相容的格式，則現在執行此操作。如需詳細資訊，請參閱 重新格式化私有金鑰 (僅限 .NET 和 Java)。
2. 以指定的順序串連以下值，並刪除各部分之間的空格 (包括標籤和新行字元)。您可能必須在應用程式的程式碼的字串中包含逸出字元。所有值都有一個字串類型。每個部分都按編號 (1) 輸入下列兩個範例。

   (1)

   ##### URL

   如果您沒有使用已簽署的 CloudFront URL (包括您自己的查詢字串參數 (如果有的話)，則基本 URL 是用來存取檔案的 URL。如需有關分佈的 URL 格式的詳細資訊，請參閱 自訂中檔案的 URL 格式 CloudFront。

   - 下列 CloudFront URL 適用於發行版中的影像檔案 (使用 CloudFront 網域名稱)。請注意，`image.jpg` 位於 `images` 目錄中。在 URL 中檔案的路徑必須與 HTTP 伺服器或 Amazon S3 儲存貯體中的檔案的路徑相符。

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg
     ```

   - 下列 CloudFront URL 包含查詢字串：

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
     ```

   - 下列 CloudFront URL 適用於發行版中的影像檔案。兩種都使用備用網域名稱，第二個包含查詢字串：

     ```
     https://www.example.com/images/image.jpg
     ```

```
https://www.example.com/images/image.jpg?color=red
```

- 以下 CloudFront URL 適用於發行版中使用替代網域名稱和 HTTPS 通訊協定的映像檔：

```
https://www.example.com/images/image.jpg
```

2

?

? 代表基本 URL 後面所接的查詢字串參數。即使您沒有自己的查詢字串參數，也請包含 ?。

3

######## (#####)&

此值是選用的。如果您希望新增自己的查詢字串參數，例如：

```
color=red&size=medium
```

接著，請在 ? 之後 (請參閱

2

)

和 Expires 參數之前加入參數。在極少數情況下，您可能需要將查詢字串參數放在 Key-Pair-Id 之後。

> **Important**
>
> 您的參數不能被命名為 Expires、Signature 或 Key-Pair-Id。

如果加入自己的參數，請在每個參數的後面附加一個 &，包括最後一個參數。

4

Expires=*Unix ########## (#####) ####### (UTC)*

您希望 URL 停止允許存取檔案的日期和時間。

以 Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 指定過期日期和時間。例如，2013 年 1 月 1 日上午 10:00 UTC 以 Unix 時間格式轉換為 1357034400。若要使用 epoch 時間，請針對不能比 2147483647 (世界協調時間 2038 年 1 月 19 日 03:14:07) 晚的日期使用 32 位元整數。如需世界協調時間的詳細資訊，請參閱 RFC 3339、網際網路上的日期和時間：時間戳記，https://tools.ietf.org/html/rfc3339。

5 **&Signature=###############**

JSON 政策聲明的雜湊、簽章和 base64-encoded 版本。如需詳細資訊，請參閱 為使用標準政策的已簽署 URL 建立簽章。

6 **&Key-Pair-Id=#####CloudFront #### ID###############**

CloudFront 公開金鑰的識別碼，例如K2JCJMDEHXQW5F。公開金鑰 ID 會告訴要使用 CloudFront 哪個公開金鑰來驗證已簽署的 URL。 CloudFront 將簽章中的資訊與原則陳述式中的資訊進行比較，以確認 URL 未遭竄改。

此公有金鑰必須屬於分佈中信任的簽署者金鑰群組。如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。

已簽署 URL 的範例：

1 **https d111111abcdef8.cloudfront.net/image.jpg**
2 **?**
3 **color=red&size=medium&**
4 **Expires=1357034400**
5 **&Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZrvDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVY**
6 **&Key-Pair-Id=K2JCJMDEHXQW5F**

為使用標準政策的已簽署 URL 建立簽章

若要使用標準政策的已簽章的 URL 建立簽章，請執行下列程序：

1. 建立政策聲明。請參閱 為使用標準政策的已簽署 URL 建立政策陳述式。

2. 簽署政策聲明來建立簽章。請參閱 為使用標準政策的已簽署 URL 建立簽章。

為使用標準政策的已簽署 URL 建立政策陳述式

當您使用標準政策建立已簽章的 URL 時，Signature 參數是政策聲明的雜湊和簽章版本。對於使用標準政策的簽章的 URL，您不會將政策聲明包括在 URL 中，就像您使用自訂政策的已簽章的 URL 一樣。若要建立政策聲明，請執行下列程序。

為使用標準政策的已簽章的 URL 建立政策聲明

1. 建構政策聲明，使用下列 JSON 格式，並使用 UTF-8 字元編碼。完全按照規定包含所有標點符號和其他常值。如需有關 Resource 和 DateLessThan 參數的詳細資訊，請參閱 您在使用標準政策的已簽署 URL 的政策陳述式中指定的值。

```
{
    "Statement": [
        {
            "Resource": "base URL or stream name",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": ending date and time in Unix time format and
 UTC
                }
            }
        }
    ]
}
```

2. 從政策陳述式中刪除所有空格 (包括標籤和新行字元)。您可能必須在應用程式的程式碼的字串中包含逸出字元。

您在使用標準政策的已簽署 URL 的政策陳述式中指定的值

當您為標準政策建立政策聲明時，您可以指定以下值。

資源

> **Note**
>
> 您只能為 Resource 指定一個值。

包含查詢字串的基本 URL (如果有的話)，但不包括 CloudFront `Expires`、`Signature`、和`Key-Pair-Id`參數，例如：

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

注意下列事項：

- 通訊協定 – 此值必須以 `http://` 或 `https://` 開頭。
- 查詢字串參數 – 如果沒有查詢字串參數，請省略問號。
- 替代網域名稱 – 如果在 URL 中指定了替代網域名稱 (CNAME)，則在引用網頁或應用程式中的檔案時，必須指定替代網域名稱。請勿為物件指定 Amazon S3 URL。

DateLessThan

Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 的 URL 的到期日期和時間。例如，2013 年 1 月 1 日上午 10:00 UTC 以 Unix 時間格式轉換為 1357034400。

此值必須與已簽章的 URL 中的 `Expires` 查詢字串參數的值相符。不要將值括在引號中。

如需詳細資訊，請參閱 何時 CloudFront 檢查簽名 URL 中的到期日期和時間？。

使用標準政策的已簽署 URL 範例政策陳述式

當您在已簽署的 URL 中使用以下範例政策聲明時，使用者在到 UTC 2013 年 1 月 1 日上午 10:00 之前可以存取檔案 `https://d111111abcdef8.cloudfront.net/horizon.jpg`：

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/horizon.jpg?
size=large&license=yes",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": 1357034400
                }
            }
        }
    ]
}
```

為使用標準政策的已簽署 URL 建立簽章

若要在已簽章的 URL 中建立 Signature 參數的值，請對您在 為使用標準政策的已簽署 URL 建立政策陳述式 中建立的政策聲明進行雜湊和簽署。

如需有關如何對政策聲明進行雜湊、簽章和編碼的詳細資訊和範例，請參閱：

- 使用 Linux 命令和 OpenSSL 進行 Base64 編碼和加密
- 為已簽署 URL 建立簽章的程式碼範例

選項 1：使用標準政策建立簽章

1. 使用 SHA-1 雜湊函數和 RSA 對您在為使用標準政策的已簽章的 URL 建立政策聲明程序中建立的政策聲明進行雜湊和簽署。使用不再包含空格之政策陳述式的版本。

   對於雜湊函數所需的私有金鑰，使用其公有金鑰在活動的信任金鑰組中的私有金鑰進行分佈。

   > **Note**
   >
   > 用於雜湊和簽名政策聲明的方法取決於您的程式設計語言和平台。如需程式碼範例，請參閱 為已簽署 URL 建立簽章的程式碼範例。

2. 從雜湊和已簽署字串中移除所有空格 (包括索引標籤和新行字元)。
3. 使用 MIME base64 編碼的 Base64-encode 字串。如需詳細資訊，請參閱 RFC 2045，MIME (多用途網際網路郵件延伸) 第一部分：網際網路訊息內文的格式中的第 6.8 節：Base64 Content-Transfer-Encoding。
4. 將 URL 查詢字串中無效的字元替換為有效的字元。下表列出無效和有效的字元。

取代這些無效的字元	有了這些有效的字元
+	- (連字號)
=	_ (底線)
/	~ (波狀符號)

5. 在 &Signature= 之後，將結果值附加到已簽章的 URL，然後返回到使用標準政策建立簽章的 URL以完成已簽章的 URL 的各個部分的連接。

使用自訂政策建立已簽署 URL

主題

若要使用自訂政策建立簽章的 URL，請執行以下程序。

使用自訂政策建立簽章的 URL

1. 如果您使用 .NET 或 Java 建立簽章的 URL，並且您還沒有將金鑰對的私有金鑰從預設的 .pem 格式重新格式化為與 .NET 或 Java 相容的格式，則現在執行此操作。如需詳細資訊，請參閱 重新格式化私有金鑰 (僅限 .NET 和 Java)。
2. 以指定的順序串連以下值，並刪除各部分之間的空格 (包括標籤和新行字元)。您可能必須在應用程式的程式碼的字串中包含逸出字元。所有值都有一個字串類型。每個部分都按編號 (1) 輸入下列兩個範例。

   (1)

   ##### URL

   如果您沒有使用已簽署的 CloudFront URL (包括您自己的查詢字串參數 (如果有的話)，則基本 URL 是用來存取檔案的 URL。如需有關分佈的 URL 格式的詳細資訊，請參閱 自訂中檔案的 URL 格式 CloudFront。

   以下範例說明您為分佈指定的值。

   - 下列 CloudFront URL 適用於發行版中的影像檔案 (使用 CloudFront 網域名稱)。請注意，`image.jpg` 位於 `images` 目錄中。在 URL 中檔案的路徑必須與 HTTP 伺服器或 Amazon S3 儲存貯體中的檔案的路徑相符。

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg
     ```

   - 下列 CloudFront URL 包含查詢字串：

     ```
     https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
     ```

   - 下列 CloudFront URL 適用於發行版中的影像檔案。兩種都使用備用網域名稱，第二個包含查詢字串：

```
https://www.example.com/images/image.jpg

https://www.example.com/images/image.jpg?color=red
```

- 以下 CloudFront URL 適用於發行版中使用替代網域名稱和 HTTPS 通訊協定的映像檔：

```
https://www.example.com/images/image.jpg
```

2

?

? 代表基本 URL 後面所接的查詢字串參數。即使您沒有自己的查詢字串參數，也請包含 ?。

3

######## (#####)&

此值是選用的。如果您希望新增自己的查詢字串參數，例如：

```
color=red&size=medium
```

接著，請在 ? 之後 (請參閱

2

)

和 Policy 參數之前加入。在極少數情況下，您可能需要將查詢字串參數放在 Key-Pair-Id 之後。

> **Important**
>
> 您的參數不能被命名為 Policy、Signature 或 Key-Pair-Id。

如果加入自己的參數，請在每個參數的後面附加一個 &，包括最後一個參數。

4

Policy=*###### base64 ####*

您的政策聲明採用 JSON 格式，刪除了空格，然後使用 base64 編碼。如需詳細資訊，請參閱為使用自訂政策的已簽署 URL 建立政策陳述式。

政策陳述式可控制簽署的 URL 授予使用者的存取權限。它包括檔案的 URL、到期日期和時間、URL 變成有效的選擇性日期和時間，以及允許存取檔案的選用 IP 地址或 IP 地址範圍。

5 **&Signature=###############**

JSON 政策聲明的雜湊、簽章和 base64-encoded 版本。如需詳細資訊，請參閱 為使用自訂準政策的已簽署 URL 建立簽章。

6 **&Key-Pair-Id=#####CloudFront #### ID###############**

CloudFront 公開金鑰的識別碼，例如K2JCJMDEHXQW5F。公開金鑰 ID 會告訴要使用 CloudFront 哪個公開金鑰來驗證已簽署的 URL。 CloudFront 將簽章中的資訊與原則陳述式中的資訊進行比較，以確認 URL 未遭竄改。

此公有金鑰必須屬於分佈中信任的簽署者金鑰群組。如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。

已簽署 URL 的範例：

1 **https**
d111111abcdef8.cloudfront.net/image.jpg
2 **?**
3 **color**
4 **Polic**
6Ly9kemJlc3FtN3VuMW0wLmNsb3VkZnJvbnQubmV0L2RlbW8ucGhwIiwgDQogICAgICAiQ
29uZGl0aW9uIjp7IA0KICAgICAgICAgIklwQWRkcmVzcyI6eyJBV1M6U291cmNlSXAiOiI
yMDcuMTcxLjE4MC4xMDEvMzIifSwNCiAgICAgICAgICJEYXRlR3JlYXRlclRoYW4iOnsiQ
VdTOkVwb2NoVGltZSI6MTI5Njg2MDE3Nn0sDQogICAgICAgICAiRGF0ZUxlc3NUaGFuIjp
7IkFXUzpFcG9jaFRpbWUiOjEyOTY4NjAyMjZ9DQogICAgICB9IA0KICAgfV0gDQp9DQo
5 **&Sign**
j19DzZrvDh6hQ73lDx~ -ar3UocvvRQVw6EkC~GdpGQyyOSKQim-
TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmat
EXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6
6 **&Key-Pair-Id=K2JCJMDEHXQW5F**

為使用自訂政策的已簽署 URL 建立政策陳述式

請完成以下步驟，以為使用自訂政策之簽署的 URL 建立政策陳述式。

如需以各種方式控制對檔案存取的政策聲明範例，請參閱 the section called “使用自訂政策的已簽署 URL 範例政策陳述式”。

為使用自訂政策的已簽章的 URL 建立政策聲明

1. 使用下列 JSON 格式建構政策聲明。使用您自己的值取代小於 (<) 和大於 (>) 符號及其中的描述。如需詳細資訊，請參閱 the section called “您在使用自訂政策的已簽署 URL 政策陳述式中指定的值”。

```
{
    "Statement": [
        {
            "Resource": "<Optional but recommended: URL of the file>",
            "Condition": {
                "DateLessThan": {
                 "AWS:EpochTime": <Required: ending date and time in Unix time
 format and UTC>
                },
                "DateGreaterThan": {
                 "AWS:EpochTime": <Optional: beginning date and time in Unix time
 format and UTC>
                },
                "IpAddress": {
                 "AWS:SourceIp": "<Optional: IP address>"
                }
            }
        }
    ]
}
```

   注意下列事項：

   - 您只能在政策中包含一個陳述式。
   - 使用 UTF-8 字元編碼。
   - 完全按照規定包含所有標點符號和參數名稱。不接受參數名稱的縮寫。
   - Condition 部分的參數順序不重要。
   - 如需有關 Resource、DateLessThan、DateGreaterThan 和 IpAddress 值的詳細資訊，請參閱 the section called “您在使用自訂政策的已簽署 URL 政策陳述式中指定的值”。
2. 從政策陳述式中刪除所有空格 (包括標籤和新行字元)。您可能必須在應用程式的程式碼的字串中包含逸出字元。

3. 使用 MIME base64 編碼 Base64-encode 政策聲明。如需詳細資訊，請參閱 RFC 2045，MIME (多用途網際網路郵件延伸) 第一部分：網際網路訊息內文的格式中的第 6.8 節：Base64 Content-Transfer-Encoding。
4. 將 URL 查詢字串中無效的字元替換為有效的字元。下表列出無效和有效的字元。

取代這些無效的字元	有了這些有效的字元
+	- (連字號)
=	_ (底線)
/	~ (波狀符號)

5. 在 `Policy=` 之後，將結果值附加到已簽章的 URL。
6. 透過政策聲明進行雜湊、簽名和 base64-encoding 來建立用於簽章 URL 的簽章。如需詳細資訊，請參閱 the section called “為使用自訂準政策的已簽署 URL 建立簽章”。

您在使用自訂政策的已簽署 URL 政策陳述式中指定的值

當您為自訂政策建立政策聲明時，您可以指定以下值。

資源

URL，包括任何查詢字串，但不包括 CloudFront `PolicySignature`、和`Key-Pair-Id`參數。例如：

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg\?
size=large&license=yes
```

您只能為 `Resource` 指定一個 URL 值。

> **Important**
>
> 您可以省略政策中的 `Resource` 參數，但如此便代表任何擁有已簽署 URL 的人，都可以存取與您建立已簽署 URL 的金鑰對關聯的任何分發中的所有檔案。

注意下列事項：

- 通訊協定 – 此值必須以 http://、https:// 或 *:// 開頭。
- 查詢字串參數 – 如果 URL 擁有查詢字串參數，請使用反斜線字元 (\) 來逸出開始查詢字串的問號字元 (?)。例如：

  ```
  https://d111111abcdef8.cloudfront.net/images/horizon.jpg\?
  size=large&license=yes
  ```

- 萬用字元 – 您可以在政策中的 URL 中使用萬用字元。支援下列萬用字元：
  - 星號 (*)，符合零或更多字元
  - 問號 (?)，剛好符合一個字元

  當原則中的 URL 與 HTTP 要求中的 URL CloudFront 相符時，原則中的 URL 會分為四個區段 (通訊協定、網域、路徑和查詢字串)，如下所示：

  ```
  [protocol]://[domain]/[path]\?[query string]
  ```

  當您在政策的 URL 中使用萬用字元時，萬用字元比對只會套用在包含萬用字元的區段邊界內。例如，在政策中考量此 URL：

  ```
  https://www.example.com/hello*world
  ```

  在此範例中，星號萬用字元 (*) 僅適用於路徑區段中，因此它符合 URL https://www.example.com/helloworld 和 https://www.example.com/hello-world，但與 URL https://www.example.net/hello?world 不相符。

  下列例外適用於萬用字元相符的區段邊界：

  - 路徑區段中的結尾星號表示查詢字串區段中的星號。例如，http://example.com/hello* 等同於 http://example.com/hello*\?*。
  - 網域區段域部分中的結尾星號表示路徑和查詢字串部分中都有星號。例如，http://example.com* 等同於 http://example.com*/*\?*。
  - 政策中的 URL 可以省略通訊協定區段，並在網域區段中以星號開頭。在這種情況下，通訊協定部分暗中設為星號。例如，策略 *example.com 中的 URL 等同於 *://*example.com/。
  - 星號本身 ("Resource": "*") 符合任何 URL。

  例如，策略 https://d111111abcdef8.cloudfront.net/*game_download.zip* 中的值:符合下列所有 URL：

  - https://d111111abcdef8.cloudfront.net/game_download.zip

  - `https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes`
  - `https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp`
- 替代網域名稱 – 如果在 URL 中指定了替代網域名稱 (CNAME)，則在引用網頁或應用程式中的檔案時，必須指定替代網域名稱。請勿在政策中為檔案指定 Amazon S3 URL。

DateLessThan

Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 的 URL 的到期日期和時間。在政策中，不要將值括在引號中。如需世界協調時間的詳細資訊，請參閱網際網路上的日期和時間：時間戳記。

例如，2023 年 1 月 31 日上午 10:00 UTC 以 Unix 時間格式轉換為 1675159200。

這是區段中唯一必要的`Condition`參數。 CloudFront 需要此值以防止用戶永久訪問您的私人內容。

如需更多資訊，請參閱the section called “何時 CloudFront 檢查簽名 URL 中的到期日期和時間？”

DateGreaterThan (選擇性)

Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 的 URL 的選用日期和時間。不允許用戶在指定的日期和時間或之前訪問文件。不要將值括在引號中。

IpAddress (選擇性)

提出 HTTP 請求的用戶端 IP 地址。注意下列事項：

- 要允許任何 IP 位址存取該檔案，請省略 `IpAddress` 參數。
- 您可以指定一個 IP 地址或一個 IP 地址範圍。如果用戶端的 IP 地址位於兩個不同的範圍之一，則無法設定政策以允許存取。
- 若要允許從單一 IP 地址存取，您需要指定：

  "*IPv4 IP ##*/32"
- 必須以標準 IPv4 CIDR 格式指定 IP 地址範圍 (例如 `192.0.2.0/24`)。如需詳細資訊，請參閱《無類別域間路由 (CIDR)：網際網路地址指派和彙總計劃》。

> **Important**
>
> 不支援 IPv6 格式的 IP 地址，例如 2001:0db8:85a3::8a2e:0370:7334。

如果您使用的是包括 IpAddress 的自訂政策，請不要為分佈啟用 IPv6。如果您希望透過 IP 地址限制對某些內容的存取，並支援對其他內容的 IPv6 請求，則可以建立兩個分佈。如需詳細資訊，請參閱 the section called “分佈設定” 主題中的 the section called “啟用 IPv6”。

使用自訂政策的已簽署 URL 範例政策陳述式

以下範例說明政策聲明說明了如何控制對特定檔案的存取、目錄中的所有檔案，或與金鑰對 ID 相關聯的所有檔案。此範例還會說明了如何控制來自個別 IP 地址或各種 IP 地址的存取，以及如何防止使用者在指定的日期和時間過期後使用簽章 URL。

如果複製並貼上這些範例中的任何一項，請刪除所有空格 (包括索引標籤和換行符號字元)、用您自我的值取代該值，並在大括弧 (}) 的後面加上換行字元。

如需詳細資訊，請參閱 the section called “您在使用自訂政策的已簽署 URL 政策陳述式中指定的值”。

主題

- 範例政策陳述式：從一個 IP 地址範圍存取一個檔案
- 範例政策陳述式：從一個 IP 地址範圍存取目錄中的所有檔案
- 範例政策陳述式：從一個 IP 地址存取與金鑰對 ID 相關的所有檔案

範例政策陳述式：從一個 IP 地址範圍存取一個檔案

以下範例自訂政策在已簽署的 URL 中指定使用者在到 UTC 2023 年 1 月 31 日上午 10:00 之前可以從 192.0.2.0/24 範圍內的 IP 地址存取檔案 https://d111111abcdef8.cloudfront.net/game_download.zip：

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/game_download.zip",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1675159200
                }
```

```
            }
        }
    ]
}
```

範例政策陳述式：從一個 IP 地址範圍存取目錄中的所有檔案

以下範例自訂政策允許您為 `training` 目錄中的任何檔案建立簽章的 URL，如 `Resource` 參數中的 (*) 萬用字元所示。使用者在到 UTC 2023 年 1 月 31 日上午 10:00 之前可以從 `192.0.2.0/24` 範圍內的 IP 地址存取該檔案：

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/training/*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1675159200
                }
            }
        }
    ]
}
```

您使用此政策的每個已簽章的 URL 都包括一個識別特定的檔案的基本 URL，例如：

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

範例政策陳述式：從一個 IP 地址存取與金鑰對 ID 相關的所有檔案

以下範例自訂政策允許您為與任何分佈關聯的任何檔案建立簽章的 URL，如 `Resource` 參數中的 (*) 萬用字元所示。已簽署的 URL 必須使用 `https://` 通訊協定，而不是 `http://`。使用者必須使用 IP 地址 `192.0.2.10/32`。(CIDR 表示法中的 `192.0.2.10/32` 值是指單個 IP 地址 `192.0.2.10`。) 這些檔案只能在 2023 年 1 月 31 日上午 10:00 UTC 至 2023 年 2 月 2 日上午 10:00 UTC 之間使用：

```
{
    "Statement": [
       {
```

```
            "Resource": "https://*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.10/32"
                },
                "DateGreaterThan": {
                    "AWS:EpochTime": 1675159200
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1675332000
                }
            }
        }
    ]
}
```

您使用此原則的每個已簽署 URL 都有一個可識別特定 CloudFront 發行版中特定檔案的 URL，例如：

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

該已簽署 URL 還包括金鑰對 ID，必須與在基本 URL 中指定的分佈 (d111111abcdef8.cloudfront.net) 中的可信金鑰群組相關聯。

為使用自訂準政策的已簽署 URL 建立簽章

使用自訂政策的已簽章的 URL 的簽章是政策聲明的雜湊、簽章和 base64-encoded 版本。若要建立自訂政策的簽章，請完成以下步驟。

如需有關如何對政策聲明進行雜湊、簽章和編碼的詳細資訊和範例，請參閱：

- 使用 Linux 命令和 OpenSSL 進行 Base64 編碼和加密
- 為已簽署 URL 建立簽章的程式碼範例

選項 1：使用自訂政策建立簽章

1. 使用 SHA-1 雜湊函數和 RSA 對您在為使用自訂政策的已簽章的 URL 建立政策聲明程序中建立的 JSON 政策聲明進行雜湊和簽署。使用不再包含空格但尚未進行 base64 編碼的政策陳述式版本。

   對於雜湊函數所需的私有金鑰，使用其公有金鑰在活動的信任金鑰組中的私有金鑰進行分佈。

> **Note**
>
> 用於雜湊和簽名政策聲明的方法取決於您的程式設計語言和平台。如需程式碼範例，請參閱 為已簽署 URL 建立簽章的程式碼範例。

2. 從雜湊和已簽署字串中移除所有空格 (包括索引標籤和新行字元)。
3. 使用 MIME base64 編碼的 Base64-encode 字串。如需詳細資訊，請參閱 RFC 2045，MIME (多用途網際網路郵件延伸) 第一部分：網際網路訊息內文的格式中的第 6.8 節：Base64 Content-Transfer-Encoding。
4. 將 URL 查詢字串中無效的字元替換為有效的字元。下表列出無效和有效的字元。

取代這些無效的字元	有了這些有效的字元
+	- (連字號)
=	_ (底線)
/	~ (波狀符號)

5. 在 `&Signature=` 之後，將結果值附加到已簽章的 URL，然後返回到使用自訂政策建立簽章的 URL以完成已簽章的 URL 的各個部分的連接。

使用已簽署 Cookie

CloudFront 簽署的 Cookie 可讓您控制當您不想變更您目前的 URL，或是想要提供存取多個受限檔案的存取權時 (例如網站訂閱者區域中的所有檔案) 時，誰可以存取您的內容。本主題說明使用已簽章的 Cookie 時的注意事項，並介紹如何使用標準和自訂政策設定已簽章的 Cookie。

主題

- 在已簽署 Cookie 的標準和自訂原則之間進行選擇
- 已簽署 Cookie 的工作方式
- 防止濫用已簽署 Cookie
- 何時 CloudFront 檢查簽名 cookie 中的到期日期和時間？
- 範例程式碼和第三方工具
- 使用標準政策設定已簽署 Cookie

- 使用自訂政策設定已簽署 Cookie

在已簽署 Cookie 的標準和自訂原則之間進行選擇

當您建立已簽章的 Cookie 時，您將編寫一個 JSON 格式的政策聲明來指定對已簽章的 Cookie 的限制，例如 Cookie 的有效時間。您可以使用標準政策或自訂政策。下表比較了標準和自訂政策：

描述	標準政策	自訂政策
您可以重複使用多個檔案的政策聲明。要重複使用政策聲明，您必須在 `Resource` 物件中使用萬用字元。如需詳細資訊，請參閱 您在政策陳述式中為已簽章 Cookie 的自訂政策指定的值)。	否	是
您可以指定使用者可以開始存取您的內容的日期和時間	否	是 (選用)
您可以指定使用者無法再存取您的內容的日期和時間	是	是
您可以指定可以存取您的內容的使用者的 IP 地址或 IP 地址範圍	否	是 (選用)

如需有關使用標準政策建立已簽章的 Cookie 的詳細資訊，請參閱 使用標準政策設定已簽署 Cookie。

如需有關使用自訂政策建立已簽章的 Cookie 的詳細資訊，請參閱 使用自訂政策設定已簽署 Cookie。

已簽署 Cookie 的工作方式

以下是您如何設定 CloudFront 已簽署 Cookie 的概觀，以及使用者提交包含已簽署 Cookie 的要求時如何 CloudFront 回應。

1. 在您的 CloudFront 散發中，指定一或多個受信任的金鑰群組，其中包含 CloudFront 可用來驗證 URL 簽章的公開金鑰。您可以使用對應的私有金鑰來簽署 URL。

   如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。
2. 您開發應用程式，以判斷使用者是否應該存取您的內容，如果是，則向檢視器傳送三個 `Set-Cookie` 標頭。（每個`Set-Cookie`標頭只能包含一個名稱-值對，一個帶 CloudFront 符號的 cookie 需要三個名稱-值對。）在瀏覽者請求您的私有內容之前，您必須將 `Set-Cookie` 標頭傳

送到檢視器。如果您在 Cookie 上設定了較短的到期時間，則可能還需要傳送三個 Set-Cookie 標頭以回應後續請求，以便使用者持續存取。

通常，您的 CloudFront 發行版至少會有兩種快取行為，一種不需要驗證，另一種則需要驗證。網站安全部分的錯誤頁面包含重定向器或指向登入頁面的連結。

如果您將分發配置為基於 Cookie 緩存文件，則 CloudFront 不會根據簽名 cookie 中的屬性緩存單獨的文件。

3. 使用者登入您的網站，以及付費內容或滿足其他存取需求。
4. 您的應用程式傳回回應中的 Set-Cookie 標頭，並且檢視器會儲存名稱值組。
5. 使用者請求了檔案。

   使用者的瀏覽器或其他檢視器從步驟 4 取得名稱值組，並將它們新增至 Cookie 標頭中的請求中。這是已簽章的 Cookie。
6. CloudFront 使用公開金鑰來驗證已簽署 Cookie 中的簽章，並確認 Cookie 沒有遭到竄改。如果簽章無效，請求會遭到拒絕。

   如果 cookie 中的簽名有效，請查 CloudFront 看 cookie 中的策略聲明（如果您使用固定策略，則構建一個策略）以確認請求仍然有效。例如，如果您為 Cookie 指定了開始和結束日期和時間，請 CloudFront 確認使用者在您要允許存取的期間內嘗試存取您的內容。

   如果要求符合原則陳述式中的需求，就像針對未受限制的內容一樣 CloudFront 提供您的內容：它會判斷檔案是否已在 Edge 快取中，視需要將要求轉送至原始位置，然後將檔案傳回給使用者。

防止濫用已簽署 Cookie

如果您在 Domain 標頭中指定 Set-Cookie 參數，請指定可能的最精確值，以降低具有相同根網域名稱之人員的存取可能性。例如，app.example.com 優於 example.com，尤其是當您無法控制 example.com 時。這有助於防止他人從 www.example.com 存取您的內容。

要協助避免發生這種類型的攻擊，請執行下列動作：

- 排除 Expires 和 Max-Age 的屬性，以便 Set-Cookie 標頭建立工作階段 Cookie。工作階段 Cookie 會在使用者關閉瀏覽器時自動刪除，以降低有人未經授權存取您的內容的可能性。
- 包含 Secure 屬性，以便檢視器在請求中包含該 Cookie 時，將對其進行加密。
- 如果可能，請使用自訂政策，並包含檢視器的 IP 地址。

- 在 CloudFront-Expires 屬性中，根據您希望使用者訪問您的內容的時間長度，指定最短的合理到期時間。

何時 CloudFront 檢查簽名 cookie 中的到期日期和時間？

要確定簽署的 cookie 是否仍然有效，請在 HTTP 請求時 CloudFront 檢查 cookie 中的到期日期和時間。如果用戶端在到期前一刻才開始下載大型檔案，則即使在下載期間過期了，下載也應該要完成。如果 TCP 連線中斷並且用戶端在到期時間過後嘗試重新啟動下載，則下載將失敗。

如果用戶端使用範圍 GET 以取得較小型的檔案，則到期時間過後發生的任何 GET 請求都將失敗。如需範圍 GET 的詳細資訊，請參閱 如何 CloudFront 處理對象的部分請求（範圍 GET）。

範例程式碼和第三方工具

私有內容的範本程式碼只說明如何為已簽章的 URL 建立簽章。但是，針對已簽章的 Cookie 建立簽章的程序非常類似，因此大部分範本程式碼仍然相關。如需詳細資訊，請參閱下列主題：

- 使用 Perl 建立 URL 簽章
- 使用 PHP 建立 URL 簽章
- 使用 C# 和 .NET 架構建立 URL 簽章
- 使用 Java 建立 URL 簽章

使用標準政策設定已簽署 Cookie

若要使用標準政策設定已簽章的 Cookie，請完成以下步驟。若要建立簽章，請參閱 為使用標準政策的已簽署 Cookie 建立簽章。

使用標準政策設定已簽章的 Cookie

1. 如果您使用 .NET 或 Java 建立已簽章的 Cookie，而且尚未將金鑰對的私有金鑰從預設的 .pem 格式重新格式化為與 .NET 或 Java 相容的格式，請現在執行此操作。如需詳細資訊，請參閱 重新格式化私有金鑰 (僅限 .NET 和 Java)。
2. 請編寫您的應用程式以發送三個 Set-Cookie 標頭給已核准的瀏覽者。您需要三個Set-Cookie標頭，因為每個標Set-Cookie頭只能包含一個名稱-值對，並且一個帶CloudFront符號的 cookie 需要三個名稱-值對。名稱值組是：CloudFront-Expires、CloudFront-Signature 和 CloudFront-Key-Pair-Id。在使用者對要控制存取的檔案發出第一次請求之前，這些值必須出現在檢視器。

> **Note**
>
> 一般而言，建議您排除 Expires 和 Max-Age 屬性。排除這些屬性會導致瀏覽器在使用者關閉瀏覽器時刪除 Cookie，從而降低有人未經授權存取您的內容的可能性。如需詳細資訊，請參閱 防止濫用已簽署 Cookie。

Cookie 屬性的名稱區分大小寫。

只包含分行符號，以使屬性更易讀。

```
Set-Cookie:
CloudFront-Expires=date and time in Unix time format (in seconds) and Coordinated
 Universal Time (UTC);
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Signature=hashed and signed version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Key-Pair-Id=public key ID for the CloudFront public key whose
 corresponding private key you're using to generate the signature;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly
```

(選用) **Domain**

請求檔案的網域名稱。如果不指定 Domain 屬性，則預設值是 URL 中的網域名稱，並且它僅適用於指定的網域名稱，而不適用於子網域名稱。如果指定 Domain 屬性，它也適用於子網域名稱。網域名稱中的前導點 (例如 Domain=.example.com) 是可選的。此外，如果指定 Domain 屬性，則 URL 中的網域名稱和 Domain 屬性的值必須相符。

您可以指定指 CloudFront 派給分發的網域名稱，例如 d1111abcdef8.cloudfront.net，但您無法為網域名稱指定 *.cloudfront.net。

如果要在 URL 中使用備用網域名稱 (如 example.com)，則無論您是否指定了 `Domain` 屬性，都必須將備用網域名稱新增到分佈中。如需詳細資訊，請參閱 發佈設定參考 主題中的 備用網域名稱 (CNAME)。

(選用) **`Path`**

請求檔案的路徑。如果未指定 `Path` 屬性，則預設值為 URL 中的路徑。

`Secure`

在檢視器傳送請求之前，需要對 Cookie 進行加密。我們建議您透過 HTTPS 連線傳送`Set-Cookie`標頭，以確保 Cookie 屬性不受 man-in-the-middle 攻擊。

`HttpOnly`

定義瀏覽器（在支持的情況下）如何與 cookie 值進行交互。使用時`HttpOnly`，Cookie 值無法存取 JavaScript。此預防措施有助於緩解跨網站指令碼 (XSS) 攻擊。如需詳細資訊，請參閱使用 HTTP Cookie。

`CloudFront-Expires`

以 Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 指定過期日期和時間。例如，2013 年 1 月 1 日上午 10:00 UTC 以 Unix 時間格式轉換為 1357034400。若要使用 epoch 時間，請針對不能比 2147483647 (世界協調時間 2038 年 1 月 19 日 03:14:07) 晚的日期使用 32 位元整數。如需世界協調時間的詳細資訊，請參閱 RFC 3339、網際網路上的日期和時間：時間戳記，https://tools.ietf.org/html/rfc3339。

`CloudFront-Signature`

JSON 政策聲明的雜湊、簽章和 base64-encoded 版本。如需詳細資訊，請參閱 為使用標準政策的已簽署 Cookie 建立簽章。

`CloudFront-Key-Pair-Id`

CloudFront 公開金鑰的識別碼，例如`K2JCJMDEHXQW5F`。公開金鑰 ID 會告訴要使用 CloudFront 哪個公開金鑰來驗證已簽署的 URL。 CloudFront 將簽章中的資訊與原則陳述式中的資訊進行比較，以確認 URL 未遭竄改。

此公有金鑰必須屬於分佈中信任的簽署者金鑰群組。如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。

以下範例顯示當您使用與檔案的 URL 中的分佈關聯的網域名稱時，一個已簽章的 Cookie 的 Set-Cookie 標頭：

```
Set-Cookie: CloudFront-Expires=1426500000; Domain=d111111abcdef8.cloudfront.net; Path=/
images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=yXrSIgyQoeE4FBI4eMKF6ho~CA8_;
 Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
 Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly
```

以下範例顯示當您在檔案的 URL 中使用備用網域名稱 example.org 時，一個已簽章的 Cookie 的 Set-Cookie 標頭：

```
Set-Cookie: CloudFront-Expires=1426500000; Domain=example.org; Path=/images/*; Secure;
 HttpOnly
Set-Cookie: CloudFront-Signature=yXrSIgyQoeE4FBI4eMKF6ho~CA8_; Domain=example.org;
 Path=/images/*; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/images/*;
 Secure; HttpOnly
```

如果要在 URL 中使用備用網域名稱 (如 example.com)，則無論您是否指定了 Domain 屬性，都必須將備用網域名稱新增到分佈中。如需詳細資訊，請參閱 發佈設定參考 主題中的 備用網域名稱 (CNAME)。

為使用標準政策的已簽署 Cookie 建立簽章

若要使用標準政策的已簽章的 Cookie 建立簽章，請執行下列動作：

1. 建立政策聲明。請參閱 為使用標準政策的已簽署 Cookie 建立政策陳述式。
2. 簽署政策聲明來建立簽章。請參閱 簽署政策陳述式以為使用標準政策的已簽署 Cookie 建立簽章。

為使用標準政策的已簽署 Cookie 建立政策陳述式

當您設定使用標準政策的簽章 Cookie 時，該 CloudFront-Signature 屬性是政策聲明的雜湊及已簽章的版本。對於使用標準政策的已簽章的 Cookie，您不會將政策聲明包括在 Set-Cookie 標頭中，就像使用自訂政策的已簽章的 Cookie 一樣。若要建立政策陳述式，請完成下列步驟。

為使用標準政策的已簽章的 Cookie 建立政策聲明

1. 建構政策聲明，使用下列 JSON 格式，並使用 UTF-8 字元編碼。完全按照規定包含所有標點符號和其他常值。如需有關 Resource 和 DateLessThan 參數的詳細資訊，請參閱 您在政策陳述式中為已簽署 Cookie 標準政策所指定的值。

```
{
    "Statement": [
        {
            "Resource": "base URL or stream name",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": ending date and time in Unix time format and
 UTC
                }
            }
        }
    ]
}
```

2. 從政策陳述式中刪除所有空格 (包括標籤和新行字元)。您可能必須在應用程式的程式碼的字串中包含逸出字元。

您在政策陳述式中為已簽署 Cookie 標準政策所指定的值

當您為標準政策建立政策聲明時，您可以指定以下值：

資源

包含查詢字串的基本 URL (如果有)，例如：

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

您只能為 Resource 指定一個值。

注意下列事項：

- 通訊協定 – 此值必須以 http:// 或 https:// 開頭。
- 查詢字串參數 – 如果沒有查詢字串參數，請省略問號。
- 替代網域名稱 – 如果在 URL 中指定了替代網域名稱 (CNAME)，則在引用網頁或應用程式中的檔案時，必須指定替代網域名稱。請勿為檔案指定 Amazon S3 URL。

DateLessThan

Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 的 URL 的到期日期和時間。不要將值括在引號中。

例如，2015 年 3 月 16 日上午 10:00 UTC 以 Unix 時間格式轉換為 1426500000。

這個值必須符合 `CloudFront-Expires` 標頭中 `Set-Cookie` 屬性的值。不要將值括在引號中。

如需詳細資訊，請參閱 何時 CloudFront 檢查簽名 cookie 中的到期日期和時間？。

標準政策的範例政策陳述式

當您在已簽章的 Cookie 中使用以下範例政策聲明時，使用者在到 UTC 2015 年 3 月 16 日上午 10:00 之前可以存取檔案 `https://d111111abcdef8.cloudfront.net/horizon.jpg`：

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/horizon.jpg?
size=large&license=yes",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime": 1426500000
                }
            }
        }
    ]
}
```

簽署政策陳述式以為使用標準政策的已簽署 Cookie 建立簽章

要為 `CloudFront-Signature` 標頭中的 `Set-Cookie` 屬性建立值，請對您在 為使用標準政策的已簽章的 Cookie 建立政策聲明 中建立的政策聲明進行雜湊和簽署。

如需有關如何對政策聲明進行雜湊、簽章和編碼的詳細資訊和範例，請參閱以下主題：

- 使用 Linux 命令和 OpenSSL 進行 Base64 編碼和加密
- 為已簽署 URL 建立簽章的程式碼範例

使用標準政策為已簽章的 Cookie 建立簽章

1. 使用 SHA-1 雜湊函數和 RSA 對您在為使用標準政策的已簽章的 Cookie 建立政策聲明程序中建立的政策聲明進行雜湊和簽署。使用不再包含空格之政策陳述式的版本。

   對於雜湊函數所需的私有金鑰，使用其公有金鑰在活動的信任金鑰組中的私有金鑰進行分佈。

   > **Note**
   >
   > 用於雜湊和簽名政策聲明的方法取決於您的程式設計語言和平台。如需程式碼範例，請參閱 為已簽署 URL 建立簽章的程式碼範例。

2. 從雜湊和已簽署字串中移除所有空格 (包括索引標籤和新行字元)。
3. 使用 MIME base64 編碼的 Base64-encode 字串。如需詳細資訊，請參閱 RFC 2045，MIME (多用途網際網路郵件延伸) 第一部分：網際網路訊息內文的格式中的第 6.8 節：Base64 Content-Transfer-Encoding。
4. 將 URL 查詢字串中無效的字元替換為有效的字元。下表列出無效和有效的字元。

取代這些無效的字元	有了這些有效的字元
+	- (連字號)
=	_ (底線)
/	~ (波狀符號)

5. 將結果值包含在 `Set-Cookie` 名稱值組的 `CloudFront-Signature` 標頭中。然後返回到 使用標準政策設定已簽章的 Cookie 為 `Set-Cookie` 新增 `CloudFront-Key-Pair-Id` 標頭。

使用自訂政策設定已簽署 Cookie

主題

- 自訂原則的範例Set-Cookie標題
- 為使用自訂政策的已簽署 Cookie 建立政策陳述式
- 使用自訂政策的已簽署 Cookie 範例政策陳述式
- 為使用自訂政策的已簽署 Cookie 建立簽章

若要設定使用標準政策的已簽署 Cookie，請完成以下步驟。

使用自訂政策設定已簽章的 Cookie

1. 如果您使用 .NET 或 Java 建立簽章的 URL，並且您還沒有將金鑰對的私有金鑰從預設的 .pem 格式重新格式化為與 .NET 或 Java 相容的格式，則現在執行此操作。如需詳細資訊，請參閱 重新格式化私有金鑰 (僅限 .NET 和 Java)。
2. 請編寫您的應用程式以發送三個 `Set-Cookie` 標頭給已核准的瀏覽者。您需要三個`Set-Cookie`標頭，因為每個標`Set-Cookie`頭只能包含一個名稱-值對，並且一個帶 CloudFront 符號的 cookie 需要三個名稱-值對。名稱值組是：`CloudFront-Policy`、`CloudFront-Signature` 和 `CloudFront-Key-Pair-Id`。在使用者對要控制存取的檔案發出第一次請求之前，這些值必須出現在檢視器。

   > **Note**
   >
   > 一般而言，建議您排除 `Expires` 和 `Max-Age` 屬性。這會導致瀏覽器在使用者關閉瀏覽器時刪除 Cookie，從而降低有人未經授權存取您的內容的可能性。如需詳細資訊，請參閱防止濫用已簽署 Cookie。

   Cookie 屬性的名稱區分大小寫。

   只包含分行符號，以使屬性更易讀。

```
Set-Cookie:
CloudFront-Policy=base64 encoded version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly


Set-Cookie:
CloudFront-Signature=hashed and signed version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
```

```
CloudFront-Key-Pair-Id=public key ID for the CloudFront public key whose
 corresponding private key you're using to generate the signature;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly
```

(選用) **Domain**

請求檔案的網域名稱。如果不指定 Domain 屬性，則預設值是 URL 中的網域名稱，並且它僅適用於指定的網域名稱，而不適用於子網域名稱。如果指定 Domain 屬性，它也適用於子網域名稱。網域名稱中的前導點 (例如 Domain=.example.com) 是可選的。此外，如果指定 Domain 屬性，則 URL 中的網域名稱和 Domain 屬性的值必須相符。

您可以指定指 CloudFront 派給分發的網域名稱，例如 d1111abcdef8.cloudfront.net，但您無法為網域名稱指定 *.cloudfront.net。

如果要在 URL 中使用備用網域名稱 (如 example.com)，則無論您是否指定了 Domain 屬性，都必須將備用網域名稱新增到分佈中。如需詳細資訊，請參閱 發佈設定參考 主題中的 備用網域名稱 (CNAME)。

(選用) **Path**

請求檔案的路徑。如果未指定 Path 屬性，則預設值為 URL 中的路徑。

Secure

在檢視器傳送請求之前，需要對 Cookie 進行加密。我們建議您透過 HTTPS 連線傳送Set-Cookie標頭，以確保 Cookie 屬性不受 man-in-the-middle 攻擊。

HttpOnly

要求檢視器僅在 HTTP 或 HTTPS 請求中傳送 Cookie。

CloudFront-Policy

您的政策聲明採用 JSON 格式，刪除了空格，然後使用 base64 編碼。如需詳細資訊，請參閱 為使用自訂政策的已簽署 Cookie 建立簽章。

政策陳述式可控制已簽署 Cookie 授予使用者的存取權限。它包括使用者可以存取的檔案、到期日期和時間、URL 變成有效的選擇性日期和時間，以及允許存取檔案的選用 IP 地址或 IP 地址範圍。

CloudFront-Signature

JSON 政策聲明的雜湊、簽章和 base64-encoded 版本。如需詳細資訊，請參閱 為使用自訂政策的已簽署 Cookie 建立簽章。

CloudFront-Key-Pair-Id

CloudFront 公開金鑰的識別碼，例如K2JCJMDEHXQW5F。公開金鑰 ID 會告訴要使用 CloudFront哪個公開金鑰來驗證已簽署的 URL。 CloudFront將簽章中的資訊與原則陳述式中的資訊進行比較，以確認 URL 未遭竄改。

此公有金鑰必須屬於分佈中信任的簽署者金鑰群組。如需詳細資訊，請參閱 指定可以建立已簽署 URL 和已簽署 Cookie 的簽署者。

自訂原則的範例**Set-Cookie**標題

請參閱標Set-Cookie題對的下面的例子。

如果您想要在 URL 中使用替代網域名稱 (例如 example.org)，則無論您是否指定屬性，都必須將替代網域名稱新增至您的分發中Domain。如需詳細資訊，請參閱 發佈設定參考 主題中的 備用網域名稱 (CNAME)。

Example 範例 1

當您在檔案的 URL 中使用與分發相關聯的網域名稱時，您可以將Set-Cookie標題用於一個已簽署的 Cookie。

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_;
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
```

Example 範例 2

當您在檔案的網址中使用替代網域名稱 (example.org) 時，您可以針對一個已簽署的 Cookie 使用Set-Cookie標頭。

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=example.org; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_; Domain=example.org;
 Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/; Secure;
 HttpOnly
```

Example 範例 3

當您在檔案的 URL 中使用與分發相關聯的網域名稱時，您可以針對已簽署的要求使用`Set-Cookie`標頭配對。

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_;
 Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
 Domain=dd111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
```

Example 範例 4

如果您使用的是替代網域名稱 (example.org)，而該網域名稱與檔案 URL 中的分佈相關聯，您可以針對一個已簽署的請求使用`Set-Cookie`標頭配對。

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
 Domain=example.org; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_; Domain=example.org;
 Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/; Secure;
 HttpOnly
```

為使用自訂政策的已簽署 Cookie 建立政策陳述式

若要為自訂政策建立政策陳述式，請執行以下步驟。如需以各種方式控制對檔案存取的幾個政策聲明範例，請參閱 使用自訂政策的已簽署 Cookie 範例政策陳述式。

為使用自訂政策的簽章的 Cookie 建立政策聲明

1. 使用下列 JSON 格式建構政策聲明。

```
{
    "Statement": [
        {
            "Resource": "URL of the file",
            "Condition": {
                "DateLessThan": {
                    "AWS:EpochTime":required ending date and time in Unix time
 format and UTC
                },
                "DateGreaterThan": {
                    "AWS:EpochTime":optional beginning date and time in Unix time
 format and UTC
                },
                "IpAddress": {
                    "AWS:SourceIp": "optional IP address"
                }
            }
        }
    ]
}
```

注意下列事項：

- 您只能包含一個聲明。
- 使用 UTF-8 字元編碼。
- 完全按照規定包含所有標點符號和參數名稱。不接受參數名稱的縮寫。
- Condition 部分的參數順序不重要。
- 如需有關 Resource、DateLessThan、DateGreaterThan 和 IpAddress 值的詳細資訊，請參閱 您在政策陳述式中為已簽章 Cookie 的自訂政策指定的值。

2. 從政策陳述式中刪除所有空格 (包括標籤和新行字元)。您可能必須在應用程式的程式碼的字串中包含逸出字元。
3. 使用 MIME base64 編碼 Base64-encode 政策聲明。如需詳細資訊，請參閱 RFC 2045，MIME (多用途網際網路郵件延伸) 第一部分：網際網路訊息內文的格式中的第 6.8 節：Base64 Content-Transfer-Encoding。
4. 將 URL 查詢字串中無效的字元替換為有效的字元。下表列出無效和有效的字元。

取代這些無效的字元	有了這些有效的字元
+	- (連字號)
=	_ (底線)
/	~ (波狀符號)

5. 將結果值包含在您的 Set-Cookie 標頭後的 CloudFront-Policy=。
6. 透過進行雜湊、簽章和 base64 編碼政策聲明來為 Set-Cookie 建立CloudFront-Signature 標頭的簽章。如需詳細資訊，請參閱 為使用自訂政策的已簽署 Cookie 建立簽章。

您在政策陳述式中為已簽章 Cookie 的自訂政策指定的值

當您為自訂政策建立政策聲明時，您可以指定以下值。

資源

包含查詢字串的基本 URL (如果有)：

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

> **Important**
>
> 如果省略 Resource 參數，則使用者可以存取與用於建立已簽章的 URL 的金鑰對相關聯的任何分佈的所有檔案。

您只能為 Resource 指定一個值。

注意下列事項：

- 通訊協定 – 此值必須以 http:// 或 https:// 開頭。
- 查詢字串參數 – 如果沒有查詢字串參數，請省略問號。
- 萬用字元 – 可以使用比對零或多個字元 (*) 的萬用字元，或比對字串中任意位置單一字元 (?) 的萬用字元。例如，值為：

```
https://d111111abcdef8.cloudfront.net/*game_download.zip*
```

將包括 (例如) 下列檔案：

- `https://d111111abcdef8.cloudfront.net/game_download.zip`
- `https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes`
- `https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp`

- 替代網域名稱 – 如果在 URL 中指定了替代網域名稱 (CNAME)，則在引用網頁或應用程式中的檔案時，必須指定替代網域名稱。請勿為檔案指定 Amazon S3 URL。

DateLessThan

Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 的 URL 的到期日期和時間。不要將值括在引號中。

例如，2015 年 3 月 16 日上午 10:00 UTC 以 Unix 時間格式轉換為 1426500000。

如需詳細資訊，請參閱 何時 CloudFront 檢查簽名 cookie 中的到期日期和時間？。

DateGreaterThan (選擇性)

Unix 時間格式 (以秒為單位) 和國際標準時間 (UTC) 的 URL 的選用日期和時間。不允許用戶在指定的日期和時間或之前訪問文件。不要將值括在引號中。

IpAddress (選擇性)

提出 GET 請求的用戶端 IP 地址。注意下列事項：

- 要允許任何 IP 位址存取該檔案，請省略 `IpAddress` 參數。
- 您可以指定一個 IP 地址或一個 IP 地址範圍。例如，如果用戶端的 IP 位址位於兩個不同的範圍之一，則無法設定政策以允許存取。
- 若要允許從單一 IP 地址存取，您需要指定：

  `"IPv4 IP ##/32"`

- 必須以標準 IPv4 CIDR 格式指定 IP 地址範圍 (例如 `192.0.2.0/24`)。如需詳細資訊，請前往 RFC 4632，無類別網域間路由選擇 (CIDR)：網際網路地址指派和彙總計劃，https://tools.ietf.org/html/rfc4632。

> **Important**
>
> 不支援 IPv6 格式的 IP 地址，例如 2001:0db8:85a3::8a2e:0370:7334。

如果您使用的是包括 `IpAddress` 的自訂政策，請不要為分佈啟用 IPv6。如果您希望透過 IP 地址限制對某些內容的存取，並支援對其他內容的 IPv6 請求，則可以建立兩個分佈。如需詳細資訊，請參閱 發佈設定參考 主題中的 啟用 IPv6。

使用自訂政策的已簽署 Cookie 範例政策陳述式

以下範例說明政策聲明說明了如何控制對特定檔案的存取、目錄中的所有檔案，或與金鑰對 ID 相關聯的所有檔案。此範例還會說明如何控制來自個別的 IP 地址或各種 IP 地址的存取，以及如何防止使用者在指定的日期和時間之後使用已簽章的 Cookie。

如果複製並貼上這些範例中的任何一項，請刪除所有空格 (包括索引標籤和換行符號字元)、用您自我的值取代該值，並在大括弧 (}) 的後面加上換行字元。

如需詳細資訊，請參閱 您在政策陳述式中為已簽章 Cookie 的自訂政策指定的值。

主題

- 範例政策陳述式：從一個 IP 地址範圍存取一個檔案
- 範例政策陳述式：從一個 IP 地址範圍存取目錄中的所有檔案
- 範例政策陳述式：從一個 IP 地址存取與金鑰對 ID 相關的所有檔案

範例政策陳述式：從一個 IP 地址範圍存取一個檔案

以下範例自訂政策在已簽署的 Cookie 中，其指定使用者在 2023 年 1 月 1 日上午 10:00 (UTC) 之前，可以從 `192.0.2.0/24` 範圍內的 IP 地址存取檔案 `https://d111111abcdef8.cloudfront.net/game_download.zip`：

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/game_download.zip",
            "Condition": {
                "IpAddress": {
```

```
                    "AWS:SourceIp": "192.0.2.0/24"
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1357034400
                }
            }
        }
    ]
}
```

範例政策陳述式：從一個 IP 地址範圍存取目錄中的所有檔案

以下範例自訂政策允許您為 `training` 目錄中的任何檔案建立已簽章的 Cookie，如 `Resource` 參數中的 * 萬用字元所示。使用者在到 UTC 2013 年 1 月 1 日上午 10:00 之前可以從 `192.0.2.0/24` 範圍內的 IP 位址存取該檔案：

```
{
    "Statement": [
        {
            "Resource": "https://d111111abcdef8.cloudfront.net/training/*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.0/24"
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1357034400
                }
            }
        }
    ]
}
```

您使用此政策的每個已簽章的 Cookie 都包括一個識別特定的檔案的基本 URL，例如：

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

範例政策陳述式：從一個 IP 地址存取與金鑰對 ID 相關的所有檔案

以下範例自訂政策允許您為與任何分佈關聯的任何檔案設定已簽章的 Cookie，如 `Resource` 參數中的 * 萬用字元所示。使用者必須使用 IP 地址 `192.0.2.10/32`。(CIDR 表示法中的 `192.0.2.10/32` 值是指單個 IP 地址 `192.0.2.10`。) 這些檔案只能在 2013 年 1 月 1 日上午 10:00 UTC 至 2013 年 1 月 2 日上午 10:00 UTC 之間可用：

```
{
    "Statement": [
        {
            "Resource": "https://*",
            "Condition": {
                "IpAddress": {
                    "AWS:SourceIp": "192.0.2.10/32"
                },
                "DateGreaterThan": {
                    "AWS:EpochTime": 1357034400
                },
                "DateLessThan": {
                    "AWS:EpochTime": 1357120800
                }
            }
        }
    ]
}
```

您使用此政策的每個已簽署 Cookie 都包含一個基礎 URL，用於識別特定 CloudFront 發行版中的特定檔案，例如：

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

該已簽署 Cookie 還包括金鑰對 ID，必須與在基本 URL 中指定的分佈 (d111111abcdef8.cloudfront.net) 內信任的簽署者相關聯。

為使用自訂政策的已簽署 Cookie 建立簽章

使用自訂政策的已簽章的 Cookie 的簽章是政策聲明的雜湊、簽章和 base64-encoded 版本。

如需有關如何對政策聲明進行雜湊、簽章和編碼的詳細資訊和範例，請參閱：

- 使用 Linux 命令和 OpenSSL 進行 Base64 編碼和加密
- 為已簽署 URL 建立簽章的程式碼範例

使用自訂政策為已簽章的 Cookie 建立簽章

1. 使用 SHA-1 雜湊函數和 RSA 對您在為使用自訂政策的已簽章的 URL 建立政策聲明程序中建立的 JSON 政策聲明進行雜湊和簽署。使用不再包含空格但尚未進行 base64 編碼的政策陳述式版本。

   對於雜湊函數所需的私有金鑰，使用其公有金鑰在活動的信任金鑰組中的私有金鑰進行分佈。

> **Note**
>
> 用於雜湊和簽名政策聲明的方法取決於您的程式設計語言和平台。如需程式碼範例，請參閱 為已簽署 URL 建立簽章的程式碼範例。

2. 從雜湊和已簽署字串中移除所有空格 (包括索引標籤和新行字元)。
3. 使用 MIME base64 編碼的 Base64-encode 字串。如需詳細資訊，請參閱 RFC 2045，MIME (多用途網際網路郵件延伸) 第一部分：網際網路訊息內文的格式中的第 6.8 節：Base64 Content-Transfer-Encoding。
4. 將 URL 查詢字串中無效的字元替換為有效的字元。下表列出無效和有效的字元。

取代這些無效的字元	有了這些有效的字元
+	- (連字號)
=	_ (底線)
/	~ (波狀符號)

5. 將結果值包含在 `Set-Cookie` 名稱值組的 `CloudFront-Signature=` 標頭中，並返回到 使用自訂政策設定已簽章的 Cookie 為 `Set-Cookie` 新增 `CloudFront-Key-Pair-Id` 標頭。

使用 Linux 命令和 OpenSSL 進行 Base64 編碼和加密

您可以使用以下 Linux 命令列命令和 OpenSSL 來雜湊和簽署政策聲明、base64 編碼簽章，並將 URL 查詢字串參數中無效的字元替換為有效的字元。

如需 OpenSSL 的詳細資訊，請前往 https://www.openssl.org。

```
cat policy | tr -d "\n" | tr -d " \t\n\r" | openssl sha1 -sign private_key.pem |
 openssl base64 -A | tr -- '+=/' '-_~'
```

在上述命令中：

- `cat`讀取`policy`檔案

- `tr -d "\n" | tr -d " \t\n\r"`刪除由添加的空格和換行符 `cat`
- OpenSSL 會使用 SHA-1 對檔案進行雜湊處理，並使用 RSA 和私密金鑰檔來簽署檔案 `private_key.pem`
- OpenSSL 雜湊和簽署的政策聲明進行編碼
- `tr`以有效字元取代 URL 查詢字串參數中無效的字元

如需示範建立簽名的更多程式碼範例，請參閱為已簽署 URL 建立簽章的程式碼範例。

為已簽署 URL 建立簽章的程式碼範例

本節包含可下載的應用程式範例，示範如何為已簽章的 URL 建立簽章。範例可在 Perl、PHP、C# 和 Java 中找到。您可以使用任何範例來建立簽章的 URL。Perl 指令碼在 Linux 和 macOS 平台上執行。PHP 範例可以在任何執行 PHP 的伺服器上執行。C# 範例使用 .NET 架構。

如需 JavaScript (Node.js) 中的範例程式碼，請參閱 AWS 開發人員部落格上的在 Node.js 中建立 Amazon CloudFront 簽署的網址。

如需 Python 中的範例程式碼，請參閱在適用於 Python 的AWS 開發套件 (Boto3) API 參考 CloudFront 中為 Amazon 產生已簽署的網址，以及 Boto3 儲存庫中的這個範例程式碼。 GitHub

主題

- 使用 Perl 建立 URL 簽章
- 使用 PHP 建立 URL 簽章
- 使用 C# 和 .NET 架構建立 URL 簽章
- 使用 Java 建立 URL 簽章

使用 Perl 建立 URL 簽章

本節包含適用於 Linux/Mac 平台的 Perl 指令碼，可用來建立私有內容的簽章。若要建立簽章，請使用指定 CloudFront URL 的命令列引數執行指令碼、簽署者私密金鑰的路徑、金鑰 ID，以及 URL 的到期日。此工具也可以解碼簽章的 URL。

> **Note**
>
> 建立 URL 簽章只是私有內容提供服務的程序的一部分，以使用簽章 URL。若要取得有關 end-to-end 程序的更多資訊，請參閱使用已簽署 URL。

主題

用於建立已簽署 URL 的 Perl 指令碼來源

下面的 Perl 源代碼可用於創建一個簽名的 URL CloudFront。程式碼中的註解包含有關命令列參數和工具功能的詳細資訊。

```
#!/usr/bin/perl -w

# Copyright 2008 Amazon Technologies, Inc.  Licensed under the Apache License, Version
 2.0 (the "License");
# you may not use this file except in compliance with the License. You may obtain a
 copy of the License at:
#
# https://aws.amazon.com/apache2.0
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 KIND, either express or implied.
# See the License for the specific language governing permissions and limitations under
 the License.

=head1 cfsign.pl

cfsign.pl - A tool to generate and verify Amazon CloudFront signed URLs

=head1 SYNOPSIS

This script uses an existing RSA key pair to sign and verify Amazon CloudFront signed
 URLs

View the script source for details as to which CPAN packages are required beforehand.

For help, try:

cfsign.pl --help

URL signing examples:

cfsign.pl --action encode --url https://images.my-website.com/gallery1.zip --policy
 sample_policy.json --private-key privkey.pem --key-pair-id mykey
```

```
cfsign.pl --action encode --url https://images.my-website.com/gallery1.zip --expires
 1257439868 --private-key privkey.pem --key-pair-id mykey

URL decode example:

cfsign.pl --action decode --url "http//mydist.cloudfront.net/?Signature=AGO-
PgxkYo99MkJFHvjfGXjG1QDEXeaDb4Qtzmy85wqyJjK7eKojQWa4BCRcow__&Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc29
Pair-Id=mykey"


To generate an RSA key pair, you can use openssl and the following commands:

# Generate a 2048 bit key pair
openssl genrsa -out private-key.pem 2048
openssl rsa -in private-key.pem -pubout -out public-key.pem


=head1 OPTIONS

=over 8

=item B<--help>

Print a help message and exits.

=item B<--action> [action]

The action to execute.  action can be one of:

  encode - Generate a signed URL (using a canned policy or a user policy)
  decode - Decode a signed URL

=item B<--url>

The URL to en/decode

=item B<--stream>

The stream to en/decode

=item B<--private-key>

The path to your private key.
```

```
=item B<--key-pair-id>

The key pair identifier.

=item B<--policy>

The CloudFront policy document.

=item B<--expires>

The Unix epoch time when the URL is to expire. If both this option and
the --policy option are specified, --policy will be used. Otherwise, this
option alone will use a canned policy.

=back

=cut

use strict;
use warnings;

# you might need to use CPAN to get these modules.
# run perl -MCPAN -e "install <module>" to get them.
# The openssl command line will also need to be in your $PATH.
use File::Temp qw/tempfile/;
use File::Slurp;
use Getopt::Long;
use IPC::Open2;
use MIME::Base64 qw(encode_base64 decode_base64);
use Pod::Usage;
use URI;

my $CANNED_POLICY
    = '{"Statement":[{"Resource":"<RESOURCE>","Condition":{"DateLessThan":
{"AWS:EpochTime":<EXPIRES>}}}]}';

my $POLICY_PARAM      = "Policy";
my $EXPIRES_PARAM     = "Expires";
my $SIGNATURE_PARAM   = "Signature";
my $KEY_PAIR_ID_PARAM = "Key-Pair-Id";

my $verbose = 0;
my $policy_filename = "";
my $expires_epoch = 0;
```

```
my $action = "";
my $help = 0;
my $key_pair_id = "";
my $url = "";
my $stream = "";
my $private_key_filename = "";

my $result = GetOptions("action=s"      => \$action,
                        "policy=s"      => \$policy_filename,
                        "expires=i"     => \$expires_epoch,
                        "private-key=s" => \$private_key_filename,
                        "key-pair-id=s" => \$key_pair_id,
                        "verbose"       => \$verbose,
                        "help"          => \$help,
                        "url=s"         => \$url,
                        "stream=s"      => \$stream,
                    );

if ($help or !$result) {
    pod2usage(1);
    exit;
}

if ($url eq "" and $stream eq "") {
    print STDERR "Must include a stream or a URL to encode or decode with the --stream
 or --url option\n";
    exit;
}

if ($url ne "" and $stream ne "") {
    print STDERR "Only one of --url and --stream may be specified\n";
    exit;
}

if ($url ne "" and !is_url_valid($url)) {
    exit;
}

if ($stream ne "") {
    exit unless is_stream_valid($stream);

    # The signing mechanism is identical, so from here on just pretend we're
    # dealing with a URL
    $url = $stream;
```

```
}

if ($action eq "encode") {
    # The encode action will generate a private content URL given a base URL,
    # a policy file (or an expires timestamp) and a key pair id parameter
    my $private_key;
    my $public_key;
    my $public_key_file;

    my $policy;
    if ($policy_filename eq "") {
        if ($expires_epoch == 0) {
            print STDERR "Must include policy filename with --policy argument or an
 expires" .
                         "time using --expires\n";
        }

        $policy = $CANNED_POLICY;
        $policy =~ s/<EXPIRES>/$expires_epoch/g;
        $policy =~ s/<RESOURCE>/$url/g;
    } else {
        if (! -e $policy_filename) {
            print STDERR "Policy file $policy_filename does not exist\n";
            exit;
        }
        $expires_epoch = 0; # ignore if set
        $policy = read_file($policy_filename);
    }

    if ($private_key_filename eq "") {
        print STDERR "You must specific the path to your private key file with --
private-key\n";
        exit;
    }

    if (! -e $private_key_filename) {
        print STDERR "Private key file $private_key_filename does not exist\n";
        exit;
    }

    if ($key_pair_id eq "") {
        print STDERR "You must specify a key pair id with --key-pair-id\n";
        exit;
    }
```

```
    my $encoded_policy = url_safe_base64_encode($policy);
    my $signature = rsa_sha1_sign($policy, $private_key_filename);
    my $encoded_signature = url_safe_base64_encode($signature);

    my $generated_url = create_url($url, $encoded_policy, $encoded_signature,
 $key_pair_id, $expires_epoch);


    if ($stream ne "") {
        print "Encoded stream (for use within a swf):\n" . $generated_url . "\n";
        print "Encoded and escaped stream (for use on a webpage):\n" .
 escape_url_for_webpage($generated_url) . "\n";
    } else {
        print "Encoded URL:\n" . $generated_url . "\n";
    }
} elsif ($action eq "decode") {
    my $decoded = decode_url($url);
    if (!$decoded) {
        print STDERR "Improperly formed URL\n";
        exit;
    }

    print_decoded_url($decoded);
} else {
    # No action specified, print help.  But only if this is run as a program (caller
 will be empty)
    pod2usage(1) unless caller();
}

# Decode a private content URL into its component parts
sub decode_url {
    my $url = shift;

    if ($url =~ /(.*)\?(.*)/) {
        my $base_url = $1;
        my $params = $2;

        my @unparsed_params = split(/&/, $params);
        my %params = ();
        foreach my $param (@unparsed_params) {
            my ($key, $val) = split(/=/, $param);
            $params{$key} = $val;
        }
```

```
        my $encoded_signature = "";
        if (exists $params{$SIGNATURE_PARAM}) {
            $encoded_signature = $params{"Signature"};
        } else {
            print STDERR "Missing Signature URL parameter\n";
            return 0;
        }

        my $encoded_policy = "";
        if (exists $params{$POLICY_PARAM}) {
            $encoded_policy = $params{$POLICY_PARAM};
        } else {
            if (!exists $params{$EXPIRES_PARAM}) {
                print STDERR "Either the Policy or Expires URL parameter needs to be
specified\n";
                return 0;
            }

            my $expires = $params{$EXPIRES_PARAM};

            my $policy = $CANNED_POLICY;
            $policy =~ s/<EXPIRES>/$expires/g;

            my $url_without_cf_params = $url;
            $url_without_cf_params =~ s/$SIGNATURE_PARAM=[^&]*&?//g;
            $url_without_cf_params =~ s/$POLICY_PARAM=[^&]*&?//g;
            $url_without_cf_params =~ s/$EXPIRES_PARAM=[^&]*&?//g;
            $url_without_cf_params =~ s/$KEY_PAIR_ID_PARAM=[^&]*&?//g;

            if ($url_without_cf_params =~ /(.*)\?$/) {
                $url_without_cf_params = $1;
            }

            $policy =~ s/<RESOURCE>/$url_without_cf_params/g;

            $encoded_policy = url_safe_base64_encode($policy);
        }

        my $key = "";
        if (exists $params{$KEY_PAIR_ID_PARAM}) {
            $key = $params{$KEY_PAIR_ID_PARAM};
        } else {
            print STDERR "Missing $KEY_PAIR_ID_PARAM parameter\n";
```

```
            return 0;
        }

        my $policy = url_safe_base64_decode($encoded_policy);

        my %ret = ();
        $ret{"base_url"} = $base_url;
        $ret{"policy"} = $policy;
        $ret{"key"} = $key;

        return \%ret;
    } else {
        return 0;
    }
}

# Print a decoded URL out
sub print_decoded_url {
    my $decoded = shift;

    print "Base URL: \n" . $decoded->{"base_url"} . "\n";
    print "Policy: \n" . $decoded->{"policy"} . "\n";
    print "Key: \n" . $decoded->{"key"} . "\n";
}

# Encode a string with base 64 encoding and replace some invalid URL characters
sub url_safe_base64_encode {
    my ($value) = @_;

    my $result = encode_base64($value);
    $result =~ tr|+=/|-_~|;

    return $result;
}

# Decode a string with base 64 encoding.  URL-decode the string first
# followed by reversing any special character ("+=/") translation.
sub url_safe_base64_decode {
    my ($value) = @_;

    $value =~ s/%([0-9A-Fa-f]{2})/chr(hex($1))/eg;
    $value =~ tr|-_~|+=/|;

    my $result = decode_base64($value);
```

```
    return $result;
}

# Create a private content URL
sub create_url {
    my ($path, $policy, $signature, $key_pair_id, $expires) = @_;

    my $result;
    my $separator = $path =~ /\?/ ? '&' : '?';
    if ($expires) {
        $result = "$path$separator$EXPIRES_PARAM=$expires&$SIGNATURE_PARAM=$signature&
$KEY_PAIR_ID_PARAM=$key_pair_id";
    } else {
        $result = "$path$separator$POLICY_PARAM=$policy&$SIGNATURE_PARAM=$signature&
$KEY_PAIR_ID_PARAM=$key_pair_id";
    }
    $result =~ s/\n//g;

    return $result;
}

# Sign a document with given private key file.
# The first argument is the document to sign
# The second argument is the name of the private key file
sub rsa_sha1_sign {
    my ($to_sign, $pvkFile) = @_;
    print "openssl sha1 -sign $pvkFile $to_sign\n";

    return write_to_program($pvkFile, $to_sign);
}

# Helper function to write data to a program
sub write_to_program {
my ($keyfile, $data) = @_;
unlink "temp_policy.dat" if (-e "temp_policy.dat");
unlink "temp_sign.dat" if (-e "temp_sign.dat");

write_file("temp_policy.dat", $data);

system("openssl dgst -sha1 -sign \"$keyfile\" -out temp_sign.dat temp_policy.dat");

my $output = read_file("temp_sign.dat");
```

```
    return $output;
}

# Read a file into a string and return the string
sub read_file {
    my ($file) = @_;

    open(INFILE, "<$file") or die("Failed to open $file: $!");
    my $str = join('', <INFILE>);
    close INFILE;

    return $str;
}

sub is_url_valid {
    my ($url) = @_;

    # HTTP distributions start with http[s]:// and are the correct thing to sign
    if ($url =~ /^https?:\/\//) {
        return 1;
    } else {
        print STDERR "CloudFront requires absolute URLs for HTTP distributions\n";
        return 0;
    }
}

sub is_stream_valid {
    my ($stream) = @_;

    if ($stream =~ /^rtmp:\/\// or $stream =~ /^\/?cfx\/st/) {
        print STDERR "Streaming distributions require that only the stream name is
 signed.\n";
        print STDERR "The stream name is everything after, but not including, cfx/st/
\n";
        return 0;
    } else {
        return 1;
    }
}

# flash requires that the query parameters in the stream name are url
# encoded when passed in through javascript, etc.  This sub handles the minimal
# required url encoding.
sub escape_url_for_webpage {
```

```
    my ($url) = @_;

    $url =~ s/\?/%3F/g;
    $url =~ s/=/%3D/g;
    $url =~ s/&/%26/g;

    return $url;
}

1;
```

使用 PHP 建立 URL 簽章

任何運行 PHP 的 Web 服務器都可以使用此 PHP 示例代碼為私有發行 CloudFront 版創建策略語句和簽名。完整範例會建立具有已簽署 URL 連結的功能正常運作的網頁，並使用 CloudFront 串流播放影片串流。您可以在以下位置下載完整的示例AmazonCloudFront。DeveloperGuide https://docs.aws.amazon.com/ /samples/demo-php.zip

您還可以使用 AWS SDK for PHP中的 `UrlSigner` 類別建立簽章的 URL。如需詳細資訊，請參閱 AWS SDK for PHP API 參考資料 UrlSigner中的類別。

> **Note**
>
> 建立 URL 簽章只是私有內容提供服務的程序的一部分，以使用簽章 URL。如需有關整個程序的詳細資訊，請參閱 使用已簽署 URL。

主題

- 範例：RSA SHA-1 簽章
- 範例：建立標準政策
- 範例：建立自訂政策
- 完整程式碼範例

範例：RSA SHA-1 簽章

在以下的程式碼範例中，函數 `rsa_sha1_sign` 雜湊並簽署政策陳述式。所需的引數為政策陳述式，以及與您分佈之信任金鑰群組中公有金鑰對應的私有金鑰。接著，該 `url_safe_base64_encode` 函數會建立已簽章的 URL 安全版本。

```
function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";

    // load the private key
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);

    // compute signature
    openssl_sign($policy, $signature, $pkeyid);

    // free the key from memory
    openssl_free_key($pkeyid);

    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with
    // the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}
```

範例：建立標準政策

下列範例程式碼會建構簽章的標準政策陳述式。如需有關標準政策的詳細資訊，請參閱 使用標準政策建立簽署的 URL。

> **Note**
>
> `$expires` 變數是日期/時間戳記，必須整數，而不是字串。

```
function get_canned_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $expires) {
    // this policy is well known by CloudFront, but you still need to sign it,
    // since it contains your parameters
```

```
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '","Condition":
{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';

    // sign the canned policy
    $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
    // make the signature safe to be included in a url
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, null, $encoded_signature,
 $key_pair_id, $expires);
    // url-encode the query string characters to work around a flash player bug
    return encode_query_params($stream_name);
}
```

範例：建立自訂政策

下列範例程式碼會建構簽章的自訂政策陳述式。如需有關自訂政策的詳細資訊，請參閱 使用自訂政策建立已簽署 URL。

```
function get_custom_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $policy) {
    // sign the policy
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a url
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_signature,
 $key_pair_id, null);
    // url-encode the query string characters to work around a flash player bug
    return encode_query_params($stream_name);
}
```

完整程式碼範例

下列範例程式碼提供使用 PHP 建立 CloudFront 已簽署 URL 的完整示範。您可以在以下位置下載這個完整的範例：AmazonCloudFrontDeveloperGuidehttps://docs.aws.amazon.com/ /samples/demo-php.zip

在下列範例中，您可以修改$policyCondition元素以允許 IPv4 和 IPv6 位址範圍。如需範例，請參閱 Amazon 簡單儲存服務使用者指南中的 IAM 政策中的使用 IPv6 地址。

```
<?php

function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";

    // load the private key
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);

    // compute signature
    openssl_sign($policy, $signature, $pkeyid);

    // free the key from memory
    openssl_free_key($pkeyid);

    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

function create_stream_name($stream, $policy, $signature, $key_pair_id, $expires) {
    $result = $stream;
    // if the stream already contains query parameters, attach the new query parameters
 to the end
    // otherwise, add the query parameters
    $separator = strpos($stream, '?') == FALSE ? '?' : '&';
    // the presence of an expires time means we're using a canned policy
    if($expires) {
        $result .= $path . $separator . "Expires=" . $expires . "&Signature=" .
 $signature . "&Key-Pair-Id=" . $key_pair_id;
    }
    // not using a canned policy, include the policy itself in the stream name
    else {
```

```
        $result .= $path . $separator . "Policy=" . $policy . "&Signature=" .
 $signature . "&Key-Pair-Id=" . $key_pair_id;
    }

    // new lines would break us, so remove them
    return str_replace('\n', '', $result);
}

function encode_query_params($stream_name) {
    // Adobe Flash Player has trouble with query parameters being passed into it,
    // so replace the bad characters with their URL-encoded forms
    return str_replace(
        array('?', '=', '&'),
        array('%3F', '%3D', '%26'),
        $stream_name);
}

function get_canned_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $expires) {
    // this policy is well known by CloudFront, but you still need to sign it, since it
 contains your parameters
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '","Condition":
{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';
    // the policy contains characters that cannot be part of a URL, so we base64 encode
 it
    $encoded_policy = url_safe_base64_encode($canned_policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, null, $encoded_signature,
 $key_pair_id, $expires);
    // URL-encode the query string characters to support Flash Player
    return encode_query_params($stream_name);
}

function get_custom_policy_stream_name($video_path, $private_key_filename,
 $key_pair_id, $policy) {
    // the policy contains characters that cannot be part of a URL, so we base64 encode
 it
    $encoded_policy = url_safe_base64_encode($policy);
    // sign the original policy, not the encoded version
```

```
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_signature,
 $key_pair_id, null);
    // URL-encode the query string characters to support Flash Player
    return encode_query_params($stream_name);
}


// Path to your private key.  Be very careful that this file is not accessible
// from the web!

$private_key_filename = '/home/test/secure/example-priv-key.pem';
$key_pair_id = 'K2JCJMDEHXQW5F';

$video_path = 'example.mp4';

$expires = time() + 300; // 5 min from now
$canned_policy_stream_name = get_canned_policy_stream_name($video_path,
 $private_key_filename, $key_pair_id, $expires);

$client_ip = $_SERVER['REMOTE_ADDR'];
$policy =
'{'.
    '"Statement":['.
        '{'.
            '"Resource":"'. $video_path . '",'.
            '"Condition":{'.
                '"IpAddress":{"AWS:SourceIp":"' . $client_ip . '/32"},'.
                '"DateLessThan":{"AWS:EpochTime":' . $expires . '}'.
            '}'.
        '}'.
    ']' .
    '}';
$custom_policy_stream_name = get_custom_policy_stream_name($video_path,
 $private_key_filename, $key_pair_id, $policy);

?>

<html>
```

```
<head>
    <title>CloudFront</title>
<script type='text/javascript' src='https://example.cloudfront.net/player/
swfobject.js'></script>
</head>

<body>
    <h1>Amazon CloudFront</h1>
    <h2>Canned Policy</h2>
    <h3>Expires at <?= gmdate('Y-m-d H:i:s T', $expires) ?></h3>
    <br />

    <div id='canned'>The canned policy video will be here</div>

    <h2>Custom Policy</h2>
    <h3>Expires at <?= gmdate('Y-m-d H:i:s T', $expires) ?> only viewable by IP <?=
 $client_ip ?></h3>
    <div id='custom'>The custom policy video will be here</div>

    <!-- ************** Have to update the player.swf path to a real JWPlayer instance.
    The fake one means that external people cannot watch the video right now -->
    <script type='text/javascript'>
  var so_canned = new SWFObject('https://files.example.com/
player.swf','mpl','640','360','9');
  so_canned.addParam('allowfullscreen','true');
  so_canned.addParam('allowscriptaccess','always');
  so_canned.addParam('wmode','opaque');
  so_canned.addVariable('file','<?= $canned_policy_stream_name ?>');
  so_canned.addVariable('streamer','rtmp://example.cloudfront.net/cfx/st');
        so_canned.write('canned');

  var so_custom = new SWFObject('https://files.example.com/
player.swf','mpl','640','360','9');
  so_custom.addParam('allowfullscreen','true');
  so_custom.addParam('allowscriptaccess','always');
  so_custom.addParam('wmode','opaque');
  so_custom.addVariable('file','<?= $custom_policy_stream_name ?>');
  so_custom.addVariable('streamer','rtmp://example.cloudfront.net/cfx/st');
  so_custom.write('custom');
    </script>
</body>

</html>
```

另請參閱:

- 使用 Perl 建立 URL 簽章
- 使用 C# 和 .NET 架構建立 URL 簽章
- 使用 Java 建立 URL 簽章

使用 C# 和 .NET 架構建立 URL 簽章

本節中的 C# 範例會實作範例應用程式，示範如何使用固定和自訂原則陳述式建立 CloudFront 私有發行版的簽章。此範例包含使用 AWS SDK for .NET 的公用程式函數，這些函數在 .NET 應用程式中非常實用。

您也可以使用 AWS SDK for .NET建立已簽章的 URL 和已簽章的 Cookie。在 AWS SDK for .NET API 參考中，請參閱下列主題：

- 簽名網址 — Amazon。 CloudFront > AmazonCloudFrontUrlSigner
- 簽名餅乾-Amazon。 CloudFront > AmazonCloudFrontCookieSigner

> **Note**
>
> 建立 URL 簽章只是私有內容提供服務的程序的一部分，以使用簽章 URL。如需有關整個程序的詳細資訊，請參閱 使用已簽署 URL。

若要下載程式碼，請前往使用 C# 的簽章程式碼。

若要在 .NET 架構中使用 RSA 金鑰，您必須將 AWS 提供的 .pem 檔案轉換為 .NET 架構使用的 XML 格式。

轉換後，RSA 私有金鑰檔案的格式如下：

Example XML .NET 架構格式的 RSA 私有金鑰

```
<RSAKeyValue>
  <Modulus>
    wO5IvYCP5UcoCKDo1dcspoMehWBZcyfs9QEzGi6Oe5y+ewGr1oW+vB2GPB
    ANBiVPcUHTFWhwaIBd3oglmF0lGQljP/jOfmXHUK2kUUnLnJp+oOBL2NiuFtqcW6h/L5lIpD8Yq+NRHg
    Ty4zDsyr2880MvXv88yEFURCkqEXAMPLE=
  </Modulus>
```

```
  <Exponent>AQAB</Exponent>
  <P>
    5bmKDaTz
    npENGVqz4Cea8XPH+sxt+2VaAwYnsarVUoSBeVt8WLloVuZGG9IZYmH5KteXEu7fZveYd9UEXAMPLE==
  </P>
  <Q>
    1v9l/WN1a1N3rOK4VGoCokx7kR2SyTMSbZgF9IWJNOugR/WZw7HTnjipO3c9dy1Ms9pUKwUF4
    6d7049EXAMPLE==
  </Q>
  <DP>
    RgrSKuLWXMyBH+/l1Dx/I4tXuAJIrlPyo+VmiOc7b5NzHptkSHEPfR9s1
    OK0VqjknclqCJ3Ig86OMEtEXAMPLE==
  </DP>
  <DQ>
    pjPjvSFw+RoaTu0pgCA/jwW/FGyfN6iim1RFbkT4
    z49DZb2IM885f3vf35eLTaEYRYUHQgZtChNEV0TEXAMPLE==
  </DQ>
  <InverseQ>
    nkvOJTg5QtGNgWb9i
    cVtzrL/1pFEOHbJXwEJdU99N+7sMK+1066DL/HSBUCD63qD4USpnf0myc24in0EXAMPLE==</InverseQ>
  <D>
      Bc7mp7XYHynuPZxChjWNJZIq+A73gm0ASDv6At7F8Vi9r0xUlQe/v0AQS3ycN8QlyR4XMbzMLYk
      3yjxFDXo4ZKQtOGzLGteCU2srANiLv26/imXA8FVidZftTAtLviWQZBVPTeYIA69ATUYPEq0a5u5wjGy
      UOij9OWyuEXAMPLE=
   </D>
</RSAKeyValue>
```

以下 C# 程式碼會透過執行以下程序建立使用標準政策簽章的 URL：

- 建立政策聲明。
- 使用 SHA1 對政策陳述式進行雜湊處理，並使用 RSA 和其對應的公有金鑰在信任金鑰組中的私有金鑰對結果進行簽署。
- Base64 編碼雜湊和簽章的政策聲明，並取代特殊字元，以使字串安全地用作 URL 請求參數。
- 串連值。

如需完整的實作，請參閱使用 C# 的簽章程式碼範例。

Example C# 的標準政策簽章方式

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
```

```
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '_')
        .Replace('/', '~');
}

public static string CreateCannedPrivateURL(string urlString,
    string durationUnits, string durationNumber, string pathToPolicyStmnt,
    string pathToPrivateKey, string privateKeyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-pathToPolicyStmnt,
    // 5-pathToPrivateKey, 6-PrivateKeyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);

    // Create the policy statement.
    string strPolicy = CreatePolicyStatement(pathToPolicyStmnt,
        urlString,
        DateTime.Now,
        DateTime.Now.Add(timeSpanInterval),
        "0.0.0.0/0");
    if ("Error!" == strPolicy) return "Invalid time frame." +
        "Start time cannot be greater than end time.";

    // Copy the expiration time defined by policy statement.
    string strExpiration = CopyExpirationTimeFromPolicy(strPolicy);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.
    using (SHA1CryptoServiceProvider
        cryptoSHA1 = new SHA1CryptoServiceProvider())
    {
        bufferPolicy = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();
        XmlDocument xmlPrivateKey = new XmlDocument();

        // Load your private key, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available.
```

```
        xmlPrivateKey.Load(pathToPrivateKey);

        // Format the RSACryptoServiceProvider providerRSA and
        // create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter rsaFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        rsaFormatter.SetHashAlgorithm("SHA1");
        byte[] signedPolicyHash = rsaFormatter.CreateSignature(bufferPolicy);

        // Convert the signed policy to URL-safe base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedPolicyHash);

        // Concatenate the URL, the timestamp, the signature,
        // and the key pair ID to form the signed URL.
        return urlString +
            "?Expires=" +
            strExpiration +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            privateKeyId;
    }
}
```

以下 C# 程式碼透過執行以下步驟建立使用自訂政策簽章的 URL：

1. 建立政策聲明。
2. Base64 編碼政策聲明，並取代特殊字元，以使字串安全地用作 URL 請求參數。
3. 使用 SHA1 對政策陳述式進行雜湊處理，並使用 RSA 和其對應的公有金鑰在信任金鑰組中的私有金鑰對結果進行加密。
4. Base64 編碼雜湊政策聲明，並取代特殊字元，以使字串安全地用作 URL 請求參數。
5. 串連值。

如需完整的實作，請參閱使用 C# 的簽章程式碼範例。

Example C# 的自訂政策簽章方式

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
```

```
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '_')
        .Replace('/', '~');
}

public static string CreateCustomPrivateURL(string urlString,
    string durationUnits, string durationNumber, string startIntervalFromNow,
    string ipaddress, string pathToPolicyStmnt, string pathToPrivateKey,
    string PrivateKeyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-starttimeFromNow,
    // 5-ip_address, 6-pathToPolicyStmt, 7-pathToPrivateKey, 8-privateKeyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);
    TimeSpan timeSpanToStart = GetDurationByUnits(durationUnits,
        startIntervalFromNow);
    if (null == timeSpanToStart)
        return "Invalid duration units." +
            "Valid options: seconds, minutes, hours, or days";

    string strPolicy = CreatePolicyStatement(
        pathToPolicyStmnt, urlString, DateTime.Now.Add(timeSpanToStart),
        DateTime.Now.Add(timeSpanInterval), ipaddress);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Convert the policy statement to URL-safe base64 encoding and
    // replace unsafe characters + = / with the safe characters - _ ~

    string urlSafePolicy = ToUrlSafeBase64String(bufferPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.
    byte[] bufferPolicyHash;
    using (SHA1CryptoServiceProvider cryptoSHA1 =
        new SHA1CryptoServiceProvider())
    {
        bufferPolicyHash = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();
        XmlDocument xmlPrivateKey = new XmlDocument();
```

```
        // Load your private key, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available.
        xmlPrivateKey.Load(pathToPrivateKey);

        // Format the RSACryptoServiceProvider providerRSA
        // and create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter RSAFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        RSAFormatter.SetHashAlgorithm("SHA1");
        byte[] signedHash = RSAFormatter.CreateSignature(bufferPolicyHash);

        // Convert the signed policy to URL-safe base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedHash);

        return urlString +
            "?Policy=" +
            urlSafePolicy +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            PrivateKeyId;
    }
}
```

Example 適用於簽章產生的公用方法

以下方法從檔案取得政策聲明和剖析簽章產生的時間間隔。

```
public static string CreatePolicyStatement(string policyStmnt,
   string resourceUrl,
   DateTime startTime,
   DateTime endTime,
   string ipAddress)

{
   // Create the policy statement.
   FileStream streamPolicy = new FileStream(policyStmnt, FileMode.Open,
 FileAccess.Read);
   using (StreamReader reader = new StreamReader(streamPolicy))
   {
```

```
      string strPolicy = reader.ReadToEnd();

      TimeSpan startTimeSpanFromNow = (startTime - DateTime.Now);
      TimeSpan endTimeSpanFromNow = (endTime - DateTime.Now);
      TimeSpan intervalStart =
         (DateTime.UtcNow.Add(startTimeSpanFromNow)) -
         new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
      TimeSpan intervalEnd =
         (DateTime.UtcNow.Add(endTimeSpanFromNow)) -
         new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);

      int startTimestamp = (int)intervalStart.TotalSeconds; // START_TIME
      int endTimestamp = (int)intervalEnd.TotalSeconds;  // END_TIME

      if (startTimestamp > endTimestamp)
         return "Error!";

      // Replace variables in the policy statement.
      strPolicy = strPolicy.Replace("RESOURCE", resourceUrl);
      strPolicy = strPolicy.Replace("START_TIME", startTimestamp.ToString());
      strPolicy = strPolicy.Replace("END_TIME", endTimestamp.ToString());
      strPolicy = strPolicy.Replace("IP_ADDRESS", ipAddress);
      strPolicy = strPolicy.Replace("EXPIRES", endTimestamp.ToString());
      return strPolicy;
   }
}

public static TimeSpan GetDuration(string units, string numUnits)
{
   TimeSpan timeSpanInterval = new TimeSpan();
   switch (units)
   {
      case "seconds":
         timeSpanInterval = new TimeSpan(0, 0, 0, int.Parse(numUnits));
         break;
      case "minutes":
         timeSpanInterval = new TimeSpan(0, 0, int.Parse(numUnits), 0);
         break;
      case "hours":
         timeSpanInterval = new TimeSpan(0, int.Parse(numUnits), 0 ,0);
         break;
      case "days":
         timeSpanInterval = new TimeSpan(int.Parse(numUnits),0 ,0 ,0);
         break;
```

```
      default:
         Console.WriteLine("Invalid time units;" +
            "use seconds, minutes, hours, or days");
         break;
   }
   return timeSpanInterval;
}

private static TimeSpan GetDurationByUnits(string durationUnits,
   string startIntervalFromNow)
{
   switch (durationUnits)
   {
      case "seconds":
         return new TimeSpan(0, 0, int.Parse(startIntervalFromNow));
      case "minutes":
         return new TimeSpan(0, int.Parse(startIntervalFromNow), 0);
      case "hours":
         return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0);
      case "days":
         return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0, 0);
      default:
         return new TimeSpan(0, 0, 0, 0);
   }
}

public static string CopyExpirationTimeFromPolicy(string policyStatement)
{
   int startExpiration = policyStatement.IndexOf("EpochTime");
   string strExpirationRough = policyStatement.Substring(startExpiration +
      "EpochTime".Length);
   char[] digits = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

   List<char> listDigits = new List<char>(digits);
   StringBuilder buildExpiration = new StringBuilder(20);

   foreach (char c in strExpirationRough)
   {
      if (listDigits.Contains(c))
         buildExpiration.Append(c);
   }
   return buildExpiration.ToString();
}
```

另請參閱

- 使用 Perl 建立 URL 簽章
- 使用 PHP 建立 URL 簽章
- 使用 Java 建立 URL 簽章

使用 Java 建立 URL 簽章

除了下列程式碼範例之外，您還可以使用 AWS SDK for Java (版本 1) 中的公用CloudFrontUrlSigner程式類別來建立CloudFront 已簽署的 URL。

如需更多範例，請參閱使用 AWS SDK 程式碼範例程式碼庫中的AWS SDK 建立已簽署的網址和 Cookie。

> **Note**
>
> 建立已簽署的 URL 只是提供私人內容的其中一部分 CloudFront。如需有關整個程序的詳細資訊，請參閱 使用已簽署 URL。

下列範例顯示如何建立 CloudFront 已簽署的 URL。

Example Java 政策和簽章加密方法

```
package org.example;

import java.time.Instant;
import java.time.temporal.ChronoUnit;
import software.amazon.awssdk.services.cloudfront.CloudFrontUtilities;
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;
import software.amazon.awssdk.services.cloudfront.url.SignedUrl;

public class Main {

    public static void main(String[] args) throws Exception {
        CloudFrontUtilities cloudFrontUtilities = CloudFrontUtilities.create();
        Instant expirationDate = Instant.now().plus(7, ChronoUnit.DAYS);
        String resourceUrl = "https://a1b2c3d4e5f6g7.cloudfront.net";
        String keyPairId = "K1UA3WV15I7JSD";
        CannedSignerRequest cannedRequest = CannedSignerRequest.builder()
```

```
                .resourceUrl(resourceUrl)
                .privateKey(new java.io.File("/path/to/private_key.pem").toPath())
                .keyPairId(keyPairId)
                .expirationDate(expirationDate)
                .build();
        SignedUrl signedUrl =
 cloudFrontUtilities.getSignedUrlWithCannedPolicy(cannedRequest);
        String url = signedUrl.url();
        System.out.println(url);

    }
}
```

另請參閱：

- 使用 Perl 建立 URL 簽章
- 使用 PHP 建立 URL 簽章
- 使用 C# 和 .NET 架構建立 URL 簽章

限制對 AWS 原始伺服器的存取

您可以以 CloudFront 提供以下好處的方式配置和某些 AWS 起源：

- 限制對 AWS 來源的訪問，使其不可公開訪問
- 確保檢視者 (使用者) 只能透過指定的 CloudFront 發行版本存取 AWS 來源中的內容，防止他們直接從值區存取內容，或透過非預期的發佈存取內容 CloudFront

若要這麼做，請設定 CloudFront 為將已驗證的要求傳送至您的 AWS 來源，並將 AWS 來源設定為僅允許存取來自的已驗證要求 CloudFront。如需詳細資訊，請參閱下列主題以取得相容類型的 AWS 原點。

主題

- 限制對 AWS Elemental MediaPackage v2 來源的訪問
- 限制對 AWS Elemental MediaStore 原始伺服器的存取
- 限制對 AWS Lambda 函數 URL 來源的訪問
- 限制對 Amazon 簡單儲存服務來源的存取

限制對 AWS Elemental MediaPackage v2 來源的訪問

CloudFront 提供原始存取控制 (OAC)，以限制對 MediaPackage v2 來源的存取。

> **Note**
>
> CloudFront OAC 僅支援 MediaPackage V2。 MediaPackage 不支援 v1。

主題

建立新的 OAC

完成下列主題中描述的步驟，以在中 CloudFront設定新的 OAC。

主題

必要條件

在您建立和設定 OAC 之前，您必須具有 MediaPackage v2 原點的 CloudFront 發佈。如需詳細資訊，請參閱 使用 MediaStore 容器或通 MediaPackage 道。

授予 OAC 訪問 MediaPackage v2 原點的權限

在您建立 OAC 或在 CloudFront 發行版中進行設定之前，請確定 OAC 具有存取 MediaPackage v2 來源的權限。在建立發行版之後，但在您將 OAC 新增至 CloudFront 散發組態中的 MediaPackage v2 原點之前，請執行此動作。

若要授與 OAC 存取 MediaPackage v2 來源的權限，請使用 IAM 政策允許 CloudFront 服務主體 (`cloudfront.amazonaws.com`) 存取來源。只有當要求代表包含 MediaPackage v2 來源的 CloudFront 發行版時，原則中的`Condition`元素才允許 CloudFront 存取 MediaPackage v2 原點。

Example：允許對 CloudFront 分配進行唯讀存取的 IAM 政策

下列原則允許 CloudFront 散發 (*E1PDK09ESKHJWT*) 存取 MediaPackage v2 來源。原點是為`Resource`元素指定的 ARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipal",
            "Effect": "Allow",
            "Principal": {"Service": "cloudfront.amazonaws.com"},
            "Action": "mediapackagev2:GetObject",
            "Resource": "arn:aws:mediapackagev2:us-
east-1:123456789012:channelGroup/channel-group-name/channel/channel-name/
originEndpoint/origin_endpoint_name",
            "Condition": {
                "StringEquals": {"AWS:SourceArn":
 "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT"}
            }
        }
    ]
}
```

> **Note**
>
> 如果您建立的散佈版本不具備 MediaPackage v2 來源的權限，您可以從 CloudFront 主控台選擇 [複製原則]，然後選擇 [更新端點權限]。然後，您可以將複製的權限附加到端點。如需詳細資訊，請參閱AWS Elemental MediaPackage 使用指南中的端點策略欄位。

建立 OAC

若要建立 OAC，您可以使用 AWS Management Console AWS CloudFormation、 AWS CLI、或 CloudFront API。

Console

若要建立 OAC

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。

2. 於左側導覽窗格中，選擇 Origin access (原始存取)。
3. 選擇 Create control setting (建立控制設定)。
4. 在 [建立新的 OAC] 表單上，執行下列動作：
   a. 輸入 OAC 的「名稱」與 (選擇性)「說明」。
   b. 對於簽署行為，建議您保留預設設定 (簽署要求 (建議))。如需詳細資訊，請參閱 the section called “原始存取控制的進階設定”。
5. 針對「原點」類型，選擇 MediaPackage V2。
6. 選擇建立。

> **Tip**
> 建立 OAC 之後，請記下 [名稱]。您需要於下列程序中進行使用。

若要將 OAC 新增至發行版中的 MediaPackage v2 原點

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您要新增 OAC 之 MediaPackage V2 來源的發佈，然後選擇「起源」頁籤。
3. 選取您要新增 OAC 的 MediaPackage v2 原點，然後選擇編輯。
4. 在原始伺服器的Protocol (通訊協定) 選取 HTTPS only (僅限 HTTPS)。
5. 從 Origin 存取控制下拉式清單中，選擇您要使用的 OAC 名稱。
6. 選擇儲存變更。

發行版會開始部署到所有邊 CloudFront 緣位置。當節點接收到新組態時，它會簽署傳送至 MediaPackage v2 原點的所有要求。

CloudFormation

若要使用建立 OAC AWS CloudFormation，請使用資`AWS::CloudFront::OriginAccessControl`源類型。下列範例會顯示建立 OAC 的 AWS CloudFormation 範本語法 (YAML 格式)。

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
```

```
      Name: ExampleOAC
      OriginAccessControlOriginType: mediapackagev2
      SigningBehavior: always
      SigningProtocol: sigv4
```

若要取得更多資訊，請參閱AWS CloudFormation 使用指南中的〈AWS::CloudFront::OriginAccess控制〉。

CLI

若要使用 AWS Command Line Interface (AWS CLI) 建立原始存取控制，請使用aws cloudfront create-origin-access-control指令。您可以使用輸入檔案來提供命令的輸入參數，而不必分別將每個個別參數指定為命令列輸入。

如要建立原始存取控制 (包含輸入檔案的 CLI)

1. 使用下列命令建立名為 `origin-access-control.yaml` 的檔案。這個檔案中包含 create-origin-access-control 命令的所有輸入參數。

   ```
   aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
    origin-access-control.yaml
   ```

2. 開啟您剛才建立的 `origin-access-control.yaml` 檔案。編輯檔案以新增 OAC 的名稱、說明 (選用)，並將 `SigningBehavior` 變更為 `always`。接著儲存檔案。

   如需其他 OAC 設定的相關資訊，請參閱 the section called “原始存取控制的進階設定”。

3. 使用下列命令，利用 `origin-access-control.yaml` 檔案中的輸入參數建立原始存取控制。

   ```
   aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
   access-control.yaml
   ```

   記下命令輸出中的 `Id` 值，您需要它將 OAC 添加到 CloudFront 發行版中的 MediaPackage v2 原點。

將 OAC 附加至現有發行版中的 MediaPackage v2 原點 (包含輸入檔的 CLI)

1. 使用下列命令來儲存您要新增 OAC 的 CloudFront發佈組態。該發行版必須具有 MediaPackage v2 來源。

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. 開啟您剛才建立且命名為 `dist-config.yaml` 的檔案。編輯檔案，進行下列變更：

   - 於 `Origins` 物件中，將 OAC 的 ID 新增至名為 `OriginAccessControlId` 的欄位。
   - 從名為 `OriginAccessIdentity` 的欄位中移除值(如果存在)。
   - 將 `ETag` 欄位重新命名為 `IfMatch`，但不要變更欄位的值。

   完成後儲存檔案。

3. 使用下列命令來更新分佈，以使用原始存取控制。

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

發行版會開始部署到所有邊 CloudFront 緣位置。當節點接收到新組態時，它會簽署傳送至 MediaPackage v2 原點的所有要求。

API

若要使用 CloudFront API 建立 OAC，請使用 CreateOriginAccessControl. 如需有關您在此 API 呼叫中指定之欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

建立 OAC 之後，您可以使用下列其中一個 API 呼叫，將其附加至發行版中的 MediaPackage v2 來源：

- 若要將其附加至現有發行版，請使用UpdateDistribution。
- 要將其附加到新的發行版本，請使用CreateDistribution。

對於這兩個 API 呼叫，請在來源內的`OriginAccessControlId`欄位中提供 OAC ID。如需有關您在這些 API 呼叫中指定的其他欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件發佈設定參考和說明文件。

原始存取控制的進階設定

CloudFront OAC 功能包含僅適用於特定使用案例的進階設定。除非您對進階設定有特定需求，否則請使用建議的設定。

OAC 包含名為簽署行為 (在主控台中) 或 `SigningBehavior` (在 API、CLI 和 AWS CloudFormation) 的設定。此設定提供下列選項：

永遠簽署原始請求 (建議設定)

我們建議使用此設定，於主控台中名為Sign requests (recommended) (簽署請求 (建議使用))，或於 API、CLI 和 AWS CloudFormation中的 `always`。使用此設定 CloudFront 時，永遠會簽署傳送至 MediaPackage v2 來源的所有要求。

絕不簽署原始伺服器請求

此設定於主控台中命名為 Do not sign requests (請勿簽署請求)，或 API、CLI 和 AWS CloudFormation中的 `never`。使用此設定可關閉使用此 OAC 之所有發行版中所有來源的 OAC。與從使用它的所有來源和發行版中逐個移除 OAC 相比，這可以節省時間和精力。使用此設定時，CloudFront 不會簽署傳送至 MediaPackage v2 原點的任何要求。

> **Warning**
>
> 若要使用此設定， MediaPackage v2 來源必須可公開存取。如果您將此設定與無法公開存取的 MediaPackage v2 來源搭配使用，則 CloudFront 無法存取原點。 MediaPackage v2 來源返回錯誤， CloudFront 並將這些錯誤傳 CloudFront 遞給查看者。如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中 MediaPackage的 [原則和權限] 範例 MediaPackage v2 原則。

請勿覆寫檢視器 (用戶端) **`Authorization`** 標題

此設定於主控台中命名為 Do not override authorization header (請勿覆寫授權標頭)，或於 API、CLI 和 AWS CloudFormation中的 `no-override`。如果您只想在對應的檢視器 CloudFront 要求不包含`Authorization`標頭時簽署原始請求，請使用此設定。使用此設定時，會在檢視器

要求存在時 CloudFront 傳遞來自檢視器要求的Authorization標頭，但是當檢視器要求未包含Authorization標頭時，會簽署原始要求 (新增其自己的Authorization標頭)。

> **Warning**
>
> 若要從檢視器要求傳遞Authorization標頭，您必須針對使用與此原始存取控制相關聯的 MediaPackage v2 來源的所有快取行為，將Authorization標頭新增至快取原則。

限制對 AWS Elemental MediaStore 原始伺服器的存取

CloudFront 提供原始存取控制 (OAC) 來限制對來源的存取 AWS Elemental MediaStore 。

主題

- 建立新的原始存取控制
- 原始存取控制的進階設定

建立新的原始存取控制

完成下列主題中描述的步驟，以在中設定新的來源存取控制 CloudFront。

主題

- 必要條件
- 授予原始訪問控制訪問 MediaStore 來源的權限
- 建立原始存取控制

必要條件

在建立和設定原始存取控制之前，您必須擁有具有 MediaStore 來源的 CloudFront 分佈。

授予原始訪問控制訪問 MediaStore 來源的權限

在您建立原始存取控制或在 CloudFront發行版中進行設定之前，請確定 OAC 具有存取來 MediaStore 源的權限。在建立 CloudFront 發行版之後，但在將 OAC 新增至散發組態中的 MediaStore原點之前，請執行此動作。

若要授與 OAC 存取來 MediaStore 源的權限，請使用 MediaStore 容器原則允許 CloudFront 服務主體 (`cloudfront.amazonaws.com`) 存取來源。使用原則中的`Condition`元素，只有在要求代表包含 MediaStore 原始位置的 CloudFront 發佈時，才允許 CloudFront 存取 MediaStore 容器。

以下是允許 CloudFront OAC 存取來源的 MediaStore 容器 MediaStore 原則範例。

Example MediaStore 允許 CloudFront OAC 唯讀存取的容器原則

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "AllowCloudFrontServicePrincipalReadOnly",
                "Effect": "Allow",
                "Principal": {
                  "Service": "cloudfront.amazonaws.com"
                },
                "Action": [
                  "mediastore:GetObject"
                ],
                "Resource":
 "arn:aws:mediastore:<region>:111122223333:container/<container name>/*",
                "Condition": {
                    "StringEquals": {
                      "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
                    },
                    "Bool": {
                      "aws:SecureTransport": "true"
                    }
                }
            }
        ]
}
```

Example MediaStore 允許 CloudFront OAC 讀取和寫入存取權的容器原則

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "AllowCloudFrontServicePrincipalReadWrite",
                "Effect": "Allow",
```

```
                "Principal": {
                  "Service": "cloudfront.amazonaws.com"
                },
                "Action": [
                  "mediastore:GetObject",
                  "mediastore:PutObject"
                ],
                "Resource":
 "arn:aws:mediastore:<region>:111122223333:container/<container name>/*",
                "Condition": {
                    "StringEquals": {
                      "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
                    },
                    "Bool": {
                      "aws:SecureTransport": "true"
                    }
                }
            }
        ]
}
```

> **Note**
>
> 若要允許寫入存取權，您必須設定允許的 HTTP 方法，以包含PUT在 CloudFront發行版的行為設定中。

建立原始存取控制

若要建立 OAC，您可以使用 AWS Management Console AWS CloudFormation、 AWS CLI、或 CloudFront API。

Console

如要建立原始存取控制

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 於左側導覽窗格中，選擇 Origin access (原始存取)。
3. 選擇 Create control setting (建立控制設定)。

4. 在 Create control setting (建立控制設定) 表單上，執行下列動作：

   a. 於 Details (詳細資訊) 窗格中，輸入 Name (名稱) 和 (選用) Description (描述)，以用於原始存取控制。

   b. 於 Settings (設定) 窗格中，建議您保留預設設定 (Sign requests (recommended)) (簽署請求 (建議使用))。如需詳細資訊，請參閱 the section called “原始存取控制的進階設定”。

5. MediaStore 從「原點類型」下拉清單中選擇。

6. 選擇建立。

   建立 OAC 之後，請記下 Name (名稱)。您需要於下列程序中進行使用。

若要將原始存取控制新增至發佈中的 MediaStore 來源

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您要新增 OAC 的 MediaStore 來源的分佈，然後選擇「起源」索引標籤。
3. 選取您要新增 OAC 的 MediaStore 原點，然後選擇 [編輯]。
4. 在原始伺服器的Protocol (通訊協定) 選取 HTTPS only (僅限 HTTPS)。
5. 在 Origin access control (原始存取控制) 下拉式功能表中，選擇您想要使用的 OAC。
6. 選擇儲存變更。

發行版會開始部署到所有邊 CloudFront 緣位置。當節點接收到新設定時，會簽署傳送至 MediaStore儲存貯體原點的所有要求。

CloudFormation

若要使用建立原始存取控制 (OAC) AWS CloudFormation，請使用資`AWS::CloudFront::OriginAccessControl`源類型。下列範例顯示建立原始存取控制的範 AWS CloudFormation 本語法 (YAML 格式)。

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: mediastore
      SigningBehavior: always
      SigningProtocol: sigv4
```

若要取得更多資訊，請參閱AWS CloudFormation 使用指南中的〈AWS::CloudFront::OriginAccessControl〉。

CLI

若要使用 AWS Command Line Interface (AWS CLI) 建立原始存取控制，請使用aws cloudfront create-origin-access-control指令。您可以使用輸入檔案來提供命令的輸入參數，而不必分別將每個個別參數指定為命令列輸入。

如要建立原始存取控制 (包含輸入檔案的 CLI)

1. 使用下列命令建立名為 `origin-access-control.yaml` 的檔案。這個檔案中包含 create-origin-access-control 命令的所有輸入參數。

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
 origin-access-control.yaml
```

2. 開啟您剛才建立的 `origin-access-control.yaml` 檔案。編輯檔案以新增 OAC 的名稱、說明 (選用)，並將 `SigningBehavior` 變更為 `always`。接著儲存檔案。

   如需其他 OAC 設定的相關資訊，請參閱 the section called “原始存取控制的進階設定”。

3. 使用下列命令，利用 `origin-access-control.yaml` 檔案中的輸入參數建立原始存取控制。

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

   記下命令輸出中的 `Id` 值，您需要將 OAC 新增至 CloudFront 發行版中的 MediaStore 原點。

將 OAC 附加到現有發行版中的 MediaStore 原點 (包含輸入檔的 CLI)

1. 使用下列命令來儲存您要新增 OAC 的 CloudFront發佈組態。分佈必須具有 MediaStore 來源。

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. 開啟您剛才建立且命名為 `dist-config.yaml` 的檔案。編輯檔案，進行下列變更：

- 於 Origins 物件中，將 OAC 的 ID 新增至名為 OriginAccessControlId 的欄位。
- 從名為 OriginAccessIdentity 的欄位中移除值(如果存在)。
- 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

完成後儲存檔案。

3. 使用下列命令來更新分佈，以使用原始存取控制。

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

發行版會開始部署到所有邊 CloudFront 緣位置。當節點接收到新組態時，它會簽署傳送至 MediaStore原點的所有要求。

API

若要使用 CloudFront API 建立原始存取控制，請使用 CreateOriginAccessControl. 如需有關您在此 API 呼叫中指定之欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

建立原始存取控制之後，您可以使用下列其中一個 API 呼叫將其附加至發佈中的 MediaStore 來源：

- 若要將其附加至現有發行版，請使用UpdateDistribution。
- 要將其附加到新的發行版，請使用CreateDistribution。

對於這兩個 API 呼叫，請於原始伺服器內部的 OriginAccessControlId 欄位中提供原始存取控制 ID。如需有關您在這些 API 呼叫中指定的其他欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件發佈設定參考和說明文件。

原始存取控制的進階設定

CloudFront 原始存取控制功能包含僅適用於特定使用案例的進階設定。除非您對進階設定有特定需求，否則請使用建議的設定。

原始存取控制包含名為簽署行為 (在主控台中) 或 SigningBehavior (在 API、CLI 和 AWS CloudFormation) 的設定。此設定提供下列選項：

永遠簽署原始請求 (建議設定)

我們建議使用此設定，於主控台中名為Sign requests (recommended) (簽署請求 (建議使用))，或於 API、CLI 和 AWS CloudFormation中的 `always`。使用此設定 CloudFront時，永遠會簽署傳送至 MediaStore 原始伺服器的所有要求。

絕不簽署原始伺服器請求

此設定於主控台中命名為 Do not sign requests (請勿簽署請求)，或 API、CLI 和 AWS CloudFormation中的 `never`。使用此設定，關閉使用此原始存取控制之所有分佈中的所有原始伺服器的原始存取控制。與從所有使用其原始伺服器和分佈中逐一移除原始存取控制相比，此可節省時間和精力。使用此設定時， CloudFront 不會簽署傳送至 MediaStore 原始伺服器的任何要求。

> **Warning**
>
> 若要使用此設定， MediaStore 原點必須可公開存取。如果您將此設定與不可公開存取的 MediaStore 來源搭配使用，則 CloudFront無法存取原點。 MediaStore 原點會傳回錯誤，CloudFront並將這些錯誤傳 CloudFront 遞給檢視者。如需詳細資訊，請參閱透過 HTTPS 進行公用讀取存取權的範例 MediaStore 容器原則。

請勿覆寫檢視器 (用戶端) **`Authorization`** 標題

此設定於主控台中命名為 Do not override authorization header (請勿覆寫授權標頭)，或於 API、CLI 和 AWS CloudFormation中的 `no-override`。如果您只想在對應的檢視器 CloudFront 要求不包含`Authorization`標頭時簽署原始請求，請使用此設定。使用此設定時，會在檢視器要求存在時 CloudFront 傳遞來自檢視器要求的`Authorization`標頭，但是當檢視器要求未包含`Authorization`標頭時，會簽署原始要求 (新增其自己的`Authorization`標頭)。

> **Warning**
>
> 若要從檢視器要求傳遞`Authorization`標頭，您必須針對使用與此原始存取控制相關聯的來 MediaStore 源的所有快取行為，將`Authorization`標頭新增至快取原則。

限制對 AWS Lambda 函數 URL 來源的訪問

CloudFront 提供來源存取控制 (OAC)，以限制對 Lambda 函數 URL 來源的存取。

主題

- 建立新的 OAC
- 原始存取控制的進階設定

建立新的 OAC

完成下列主題中描述的步驟，以在中 CloudFront設定新的 OAC。

> **Note**
>
> 如果您使用PUT或POST方法搭配 Lambda 函數 URL，您的使用者必須提供已簽署的承載給 CloudFront。Lambda 不支持未簽名的有效載荷。

主題

- 必要條件
- 授予 OAC 存取 Lambda 函數 URL 的權限
- 建立 OAC

必要條件

在建立和設定 OAC 之前，您必須擁有一個以 Lambda 函數 URL 作為來源的 CloudFront 發佈。如需詳細資訊，請參閱 使用 Lambda 函數網址。

授予 OAC 存取 Lambda 函數 URL 的權限

在您建立 OAC 或在 CloudFront 發佈中進行設定之前，請確定 OAC 具有存取 Lambda 函數 URL 的權限。在建立分發之後，但在您將 OAC 新增至 CloudFront發佈組態中的 Lambda 函數 URL 之前，請執行此動作。

> **Note**
>
> 若要更新 Lambda 函數 URL 的 IAM 政策，您必須使用 AWS Command Line Interface (AWS CLI)。目前不支援在 Lambda 主控台中編輯身分與存取權管理政策。

下列 AWS CLI 命令會授與 CloudFront 服務主體 (`cloudfront.amazonaws.com`) 存取您的 Lambda 函數 URL。只有當請求代表包含 Lambda 函數 URL 的 CloudFront 發佈時，政策中的`Condition`元素才允許 CloudFront 存取 Lambda。

Example：更新原則以允許 CloudFront OAC 唯讀存取的 AWS CLI 命令

以下 AWS CLI 命令允許 CloudFront 分發（*E1PDK09ESKHJWT*）訪問您的 Lambda *FUNCTION_URL_NAME*。

```
aws lambda add-permission \
--statement-id "AllowCloudFrontServicePrincipal" \
--action "lambda:InvokeFunctionUrl" \
--principal "cloudfront.amazonaws.com" \
--source-arn "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT" \
--function-name FUNCTION_URL_NAME
```

> **Note**
>
> 如果您建立發行版，但它沒有 Lambda 函數 URL 的權限，您可以從 CloudFront主控台選擇複製 CLI 命令，然後從命令列終端機輸入此命令。如需詳細資訊，請參閱AWS Lambda 開發人員指南 AWS 服務中的授予函數存取權。

建立 OAC

若要建立 OAC，您可以使用 AWS Management Console AWS CloudFormation、 AWS CLI、或 CloudFront API。

Console

若要建立 OAC

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 於左側導覽窗格中，選擇 Origin access (原始存取)。
3. 選擇 Create control setting (建立控制設定)。
4. 在 [建立新的 OAC] 表單上，執行下列動作：
   a. 輸入 OAC 的「名稱」與 (選擇性)「說明」。
   b. 對於簽署行為，建議您保留預設設定 (簽署要求 (建議))。如需詳細資訊，請參閱 the section called “原始存取控制的進階設定”。
5. 針對原點類型，選擇 Lambda。

6. 選擇建立。

> **Tip**
>
> 建立 OAC 之後，請記下「名稱」。您需要於下列程序中進行使用。

若要將來源存取控制新增至發佈中的 Lambda 函數 URL

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您想要將 OAC 新增至的 Lambda 函數 URL 的發佈，然後選擇「起源」索引標籤。
3. 選取您要新增 OAC 的 Lambda 函數 URL，然後選擇 [編輯]。
4. 在原始伺服器的Protocol (通訊協定) 選取 HTTPS only (僅限 HTTPS)。
5. 從 Origin 存取控制下拉式清單中，選擇您要使用的 OAC 名稱。
6. 選擇儲存變更。

發行版會開始部署到所有邊 CloudFront 緣位置。當節點接收到新組態時，它會簽署傳送至 Lambda 函數 URL 的所有請求。

CloudFormation

若要使用建立 OAC AWS CloudFormation，請使用資`AWS::CloudFront::OriginAccessControl`源類型。下列範例會顯示建立 OAC 的 AWS CloudFormation 範本語法 (YAML 格式)。

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: lambda
      SigningBehavior: always
      SigningProtocol: sigv4
```

若要取得更多資訊，請參閱AWS CloudFormation 使用指南中的〈AWS::CloudFront::OriginAccess控制〉。

CLI

若要使用 AWS Command Line Interface (AWS CLI) 建立原始存取控制，請使用aws cloudfront create-origin-access-control指令。您可以使用輸入檔案來提供命令的輸入參數，而不必分別將每個個別參數指定為命令列輸入。

如要建立原始存取控制 (包含輸入檔案的 CLI)

1. 使用下列命令建立名為 `origin-access-control.yaml` 的檔案。這個檔案中包含 create-origin-access-control 命令的所有輸入參數。

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
 origin-access-control.yaml
```

2. 開啟您剛才建立的 `origin-access-control.yaml` 檔案。編輯檔案以新增 OAC 的名稱、說明 (選用)，並將 `SigningBehavior` 變更為 `always`。接著儲存檔案。

   如需其他 OAC 設定的相關資訊，請參閱 the section called “原始存取控制的進階設定”。

3. 使用下列命令，利用 `origin-access-control.yaml` 檔案中的輸入參數建立原始存取控制。

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

   記下命令輸出中的 `Id` 值，您需要將 OAC 新增至 CloudFront 分發中的 Lambda 函數 URL。

若要將 OAC 附加至現有發行版中的 Lambda 函數 URL (包含輸入檔案的 CLI)

1. 使用下列命令來儲存您要新增 OAC 的 CloudFront發佈組態。該發行版必須有一個 Lambda 函數 URL 作為來源。

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. 開啟您剛才建立且命名為 `dist-config.yaml` 的檔案。編輯檔案，進行下列變更：

   - 於 `Origins` 物件中，將 OAC 的 ID 新增至名為 `OriginAccessControlId` 的欄位。

- 從名為 `OriginAccessIdentity` 的欄位中移除值(如果存在)。
- 將 `ETag` 欄位重新命名為 `IfMatch`，但不要變更欄位的值。

完成後儲存檔案。

3. 使用下列命令來更新分佈，以使用原始存取控制。

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

發行版會開始部署到所有邊 CloudFront 緣位置。當節點接收到新組態時，它會簽署傳送至 Lambda 函數 URL 的所有請求。

API

若要使用 CloudFront API 建立 OAC，請使用 CreateOriginAccessControl. 如需有關您在此 API 呼叫中指定之欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

建立 OAC 之後，您可以使用下列其中一個 API 呼叫，將其附加至分發中的 Lambda 函數 URL：

- 若要將其附加至現有發行版，請使用UpdateDistribution。
- 要將其附加到新的發行版本，請使用CreateDistribution。

對於這兩個 API 呼叫，請在來源內的`OriginAccessControlId`欄位中提供 OAC ID。如需有關您在這些 API 呼叫中指定的其他欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件和說明文件。

原始存取控制的進階設定

CloudFront OAC 功能包含僅適用於特定使用案例的進階設定。除非您對進階設定有特定需求，否則請使用建議的設定。

OAC 包含名為簽署行為 (在主控台中) 或 `SigningBehavior` (在 API、CLI 和 AWS CloudFormation) 的設定。此設定提供下列選項：

永遠簽署原始請求 (建議設定)

我們建議使用此設定，於主控台中名為Sign requests (recommended) (簽署請求 (建議使用))，或於 API、CLI 和 AWS CloudFormation中的 `always`。使用此設定 CloudFront 時，永遠會簽署傳送至 Lambda 函數 URL 的所有要求。

絕不簽署原始伺服器請求

此設定於主控台中命名為 Do not sign requests (請勿簽署請求)，或 API、CLI 和 AWS CloudFormation中的 `never`。使用此設定可關閉使用此 OAC 之所有發行版中所有來源的 OAC。與從使用它的所有來源和發行版中逐個移除 OAC 相比，這可以節省時間和精力。使用此設定時，CloudFront 不會簽署傳送至 Lambda 函數 URL 的任何要求。

> **Warning**
>
> 若要使用此設定，Lambda 函數 URL 必須可公開存取。如果您將此設定與無法公開存取的 Lambda 函數 URL 搭配使用，則 CloudFront 無法存取來源。Lambda 函數 URL 會傳回錯誤， CloudFront 並將這些錯誤傳 CloudFront 送給檢視者。如需詳細資訊，請參閱AWS Lambda 使用者指南中 Lambda 中針對政策和許可的範例 Lambda 政策。

請勿覆寫檢視器 (用戶端) **`Authorization`** 標題

此設定於主控台中命名為 Do not override authorization header (請勿覆寫授權標頭)，或於 API、CLI 和 AWS CloudFormation中的 `no-override`。如果您只想在對應的檢視器 CloudFront 要求不包含`Authorization`標頭時簽署原始請求，請使用此設定。使用此設定時，會在檢視器要求存在時 CloudFront 傳遞來自檢視器要求的`Authorization`標頭，但是當檢視器要求未包含`Authorization`標頭時，會簽署原始要求 (新增其自己的`Authorization`標頭)。

> **Warning**
>
> 若要從檢視器要求傳遞`Authorization`標頭，您必須針對使用與此原始存取控制相關聯的 Lambda 函數 URL 的所有快取行為，將`Authorization`標頭新增至快取政策。

限制對 Amazon 簡單儲存服務來源的存取

CloudFront 提供兩種將經過驗證的請求傳送至 Amazon S3 來源的方式：原始存取控制 (OAC) 和來源存取身分識別 (OAI)。OAC 可協助您保護來源安全，例如 Amazon S3。我們建議使用 OAC，因為其支援：

- 所有 Amazon S3 儲存貯體 AWS 區域，包括 2022 年 12 月後推出的選擇加入區域
- 使用 AWS KMS的 Amazon S3 伺服器端加密 (SSE-KMS)
- 對 Amazon S3 的動態請求 (PUT 和 DELETE)

原始存取身分 (OAI) 不適用於上述清單中的情境，或者在這些情境中需要額外的因應措施。下列主題說明如何將原始存取控制 (OAC) 與 Amazon S3 原始伺服器搭配使用。如需如何從原始存取身分 (OAI) 遷移至原始存取控制 (OAC) 的相關資訊，請參閱 the section called “從原始存取身分 (OAI) 遷移至原始存取控制 (OAC)”。

> **備註**
>
> - 當您將 CloudFront OAC 與 Amazon S3 儲存貯體起源搭配使用時，必須將 Amazon S3 物件擁有權設定為強制執行儲存貯體擁有者，這是新 Amazon S3 儲存貯體的預設值。如果您需要 ACL，請使用值區擁有者偏好設定來維持對透過 CloudFront上傳之物件的控制權。
> - 如果您的來源是設定為網站端點的 Amazon S3 儲存貯體，則必須將其設定 CloudFront 為自訂來源。這表示您無法使用 OAC (或 OAI)。OAC 不支援使用 Lambda @Edge 來重新導向來源。

主題

- the section called “建立新的原始存取控制”
- the section called “刪除附加到 S3 儲存貯體的 OAC 的發佈”
- the section called “從原始存取身分 (OAI) 遷移至原始存取控制 (OAC)”
- the section called “原始存取控制的進階設定”

建立新的原始存取控制

完成下列主題中描述的步驟，以在中設定新的來源存取控制 CloudFront。

主題

- 必要條件
- 授予原始存取控制許可，以存取 S3 儲存貯體
- 建立原始存取控制

必要條件

在建立和設定來源存取控制 (OAC) 之前，您必須具有 Amazon S3 儲存貯體來源的 CloudFront分發。此原始伺服器必須是一般 S3 儲存貯體，而非設定為網站端點的儲存貯體。如需使用 S3 儲存貯體來源設定 CloudFront 分發的詳細資訊，請參閱the section called “開始使用基本發行版”。

> **Note**
>
> 當您使用 OAC 來保護 S3 儲存貯體來源時， CloudFront 與 Amazon S3 之間的通訊一律會透過 HTTPS 進行，無論您的特定設定為何。

授予原始存取控制許可，以存取 S3 儲存貯體

在建立來源存取控制 (OAC) 或在 CloudFront 發佈中設定之前，請確定 OAC 具有存取 S3 儲存貯體來源的權限。在建立 CloudFront分發之後，但在將 OAC 新增至散發組態中的 S3 來源之前，請執行此操作。

若要授與 OAC 存取 S3 儲存貯體的權限，請使用 S3 儲存貯體政策允許 CloudFront服務主體 (`cloudfront.amazonaws.com`) 存取儲存貯體。使用政策中的`Condition`元素，只有在請求代表包含 S3 來源的 CloudFront 發佈時才允許 CloudFront 存取儲存貯體。

如需新增或修改儲存貯體政策的相關資訊，請參閱 Amazon S3 使用者指南中的使用 Amazon S3 主控台新增儲存貯體政策。

以下是允許 CloudFront OAC 存取 S3 來源的 S3 儲存貯體政策範例。

Example 允許 CloudFront OAC 唯讀存取的 S3 儲存貯體政策

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "AllowCloudFrontServicePrincipalReadOnly",
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudfront.amazonaws.com"
        },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::<S3 bucket name>/*",
        "Condition": {
            "StringEquals": {
```

```
                "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
            }
        }
    }
}
```

Example 允許 CloudFront OAC 讀取和寫入存取的 S3 儲存貯體政策

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "AllowCloudFrontServicePrincipalReadWrite",
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudfront.amazonaws.com"
        },
        "Action": [
            "s3:GetObject",
            "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::<S3 bucket name>/*",
        "Condition": {
            "StringEquals": {
                "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
            }
        }
    }
}
```

SSE-KMS

如果 S3 儲存貯體來源中的物件使用伺服器端加密 AWS Key Management Service (SSE-KMS) 進行加密，您必須確定 OAC 具有使用金鑰的權限。 AWS KMS 如要授予 OAC 使用 KMS 金鑰的許可，請將陳述式新增至 KMS 金鑰政策。如需如何修改金鑰政策的相關資訊，請參閱《AWS Key Management Service 開發人員指南》中的變更金鑰政策。

下列範例顯示允許 OAC 使用 KMS 金鑰的 KMS 金鑰政策陳述式。

Example KMS 金鑰原則陳述式，可讓 CloudFront OAC 存取 SSE-KMS 的 KMS 金鑰

```
{
```

```
    "Sid": "AllowCloudFrontServicePrincipalSSE-KMS",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "cloudfront.amazonaws.com"
        ]
     },
    "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
            "StringEquals": {
                "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
            }
        }
}
```

建立原始存取控制

若要建立原始存取控制 (OAC)，您可以使用 AWS Management Console AWS CloudFormation AWS CLI、、或 CloudFront API。

Console

如要建立原始存取控制

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 於左側導覽窗格中，選擇 Origin access (原始存取)。
3. 選擇 Create control setting (建立控制設定)。
4. 在 Create control setting (建立控制設定) 表單上，執行下列動作：
   a. 於 Details (詳細資訊) 窗格中，輸入 Name (名稱) 和 (選用) Description (描述)，以用於原始存取控制。
   b. 於 Settings (設定) 窗格中，建議您保留預設設定 (Sign requests (recommended)) (簽署請求 (建議使用))。如需詳細資訊，請參閱 the section called “原始存取控制的進階設定”。
5. 從 Origin type (原始伺服器類型) 下拉式功能表中選擇 S3。

6. 選擇建立。

   建立 OAC 之後，請記下 Name (名稱)。您需要於下列程序中進行使用。

如要將原始存取控制新增至分佈中的 S3 原始伺服器

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇一個您想要新增 OAC 之具 S3 原始伺服器的分佈，然後選擇 Origins (原始伺服器) 標籤。
3. 選取您想要將 OAC 新增至的 S3 原始伺服器，然後選擇 Edit (編輯)。
4. 對於 Origin 存取權限，請選擇 Origin 存取控制設定 (建議使用)。
5. 在 Origin access control (原始存取控制) 下拉式功能表中，選擇您想要使用的 OAC。
6. 選擇儲存變更。

發行版會開始部署到所有邊 CloudFront 緣位置。當邊緣節點接收到新組態時，其會簽署傳送至 S3 儲存貯體原始伺服器的所有請求。

CloudFormation

若要使用建立原始存取控制 (OAC) AWS CloudFormation，請使用資`AWS::CloudFront::OriginAccessControl`源類型。下列範例顯示建立原始存取控制的範 AWS CloudFormation 本語法 (YAML 格式)。

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
      Description: An optional description for the origin access control
      Name: ExampleOAC
      OriginAccessControlOriginType: s3
      SigningBehavior: always
      SigningProtocol: sigv4
```

若要取得更多資訊，請參閱AWS CloudFormation 使用指南中的〈AWS::CloudFront::OriginAccess控制〉。

CLI

若要使用 AWS Command Line Interface (AWS CLI) 建立原始存取控制，請使用aws cloudfront create-origin-access-control指令。您可以使用輸入檔案來提供命令的輸入參數，而不必分別將每個個別參數指定為命令列輸入。

如要建立原始存取控制 (包含輸入檔案的 CLI)

1. 使用下列命令建立名為 `origin-access-control.yaml` 的檔案。這個檔案中包含 create-origin-access-control 命令的所有輸入參數。

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
 origin-access-control.yaml
```

2. 開啟您剛才建立的 `origin-access-control.yaml` 檔案。編輯檔案以新增 OAC 的名稱、說明 (選用)，並將 `SigningBehavior` 變更為 `always`。接著儲存檔案。

   如需其他 OAC 設定的相關資訊，請參閱 the section called “原始存取控制的進階設定”。

3. 使用下列命令，利用 `origin-access-control.yaml` 檔案中的輸入參數建立原始存取控制。

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

   記下命令輸出中的 `Id` 值，您需要將 OAC 新增至 CloudFront 散發中的 S3 儲存貯體來源。

如要將 OAC 附加至現有分佈 (包含輸入檔案的 CLI) 中的 S3 儲存貯體原始伺服器

1. 使用下列命令來儲存您要新增 OAC 的 CloudFront發佈組態。分佈必須具有 S3 儲存貯體原始伺服器。

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --
output yaml > dist-config.yaml
```

2. 開啟您剛才建立且命名為 `dist-config.yaml` 的檔案。編輯檔案，進行下列變更：

   - 於 `Origins` 物件中，將 OAC 的 ID 新增至名為 `OriginAccessControlId` 的欄位。
   - 從名為 `OriginAccessIdentity` 的欄位中移除值(如果存在)。
   - 將 `ETag` 欄位重新命名為 `IfMatch`，但不要變更欄位的值。

   完成後儲存檔案。

3. 使用下列命令來更新分佈，以使用原始存取控制。

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-
input-yaml file://dist-config.yaml
```

發行版會開始部署到所有邊 CloudFront 緣位置。當邊緣節點接收到新組態時，其會簽署傳送至 S3 儲存貯體原始伺服器的所有請求。

API

若要使用 CloudFront API 建立原始存取控制，請使用 CreateOriginAccessControl. 如需有關您在此 API 呼叫中指定之欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

建立原始存取控制之後，您可以使用下列其中一個 API 呼叫，將其連接至分佈中的 S3 儲存貯體原始伺服器：

- 若要將其附加至現有發行版，請使用UpdateDistribution。
- 要將其附加到新的發行版，請使用CreateDistribution。

對於這兩個 API 呼叫，請於原始伺服器內部的 `OriginAccessControlId` 欄位中提供原始存取控制 ID。如需有關您在這些 API 呼叫中指定的其他欄位的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件發佈設定參考和說明文件。

刪除附加到 S3 儲存貯體的 OAC 的發佈

如果您需要刪除 OAC 連接到 S3 儲存貯體的分發，您應該先刪除該分發，然後再刪除 S3 儲存貯體來源。或者，在原始網域名稱中加入「地區」。如果這是不可能的，您可以在刪除之前切換到公用，從發行版本中移除 OAC。如需詳細資訊，請參閱 刪除 分發。

從原始存取身分 (OAI) 遷移至原始存取控制 (OAC)

如要從舊式原始存取身分 (OAI) 遷移至原始存取控制 (OAC)，請先更新 S3 儲存貯體原始伺服器，以允許 OAI 和 OAC 存取儲存貯體的內容。這可確保在轉換期間 CloudFront 永遠不會失去對值區的存取權。如要允許 OAI 和 OAC 存取 S3 儲存貯體，請更新儲存貯體政策，以包括兩個陳述式，每種主體各一個。

下列範例 S3 儲存貯體政策允許 OAI 和 OAC 存取 S3 原始伺服器。

Example 允許唯讀存取 OAI 和 OAC 的 S3 儲存貯體政策

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFrontServicePrincipalReadOnly",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
                }
            }
        },
        {
            "Sid": "AllowLegacyOAIReadOnly",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity <origin access identity ID>"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}
```

更新 S3 原始伺服器的儲存貯體政策以允許存取 OAI 和 OAC 之後，您可更新發佈組態以使用 OAC 而非 OAI。如需詳細資訊，請參閱 the section called “建立新的原始存取控制”。

完整部署分佈之後，您可移除儲存貯體政策中允許存取 OAI 的陳述式。如需詳細資訊，請參閱 the section called “授予原始存取控制許可，以存取 S3 儲存貯體”。

原始存取控制的進階設定

CloudFront 原始存取控制功能包含僅適用於特定使用案例的進階設定。除非您對進階設定有特定需求，否則請使用建議的設定。

原始存取控制包含名為簽署行為 (在主控台中) 或 `SigningBehavior` (在 API、CLI 和 AWS CloudFormation) 的設定。此設定提供下列選項：

永遠簽署原始請求 (建議設定)

我們建議使用此設定，於主控台中名為Sign requests (recommended) (簽署請求 (建議使用))，或於 API、CLI 和 AWS CloudFormation中的 `always`。使用此設定， CloudFront永遠會簽署傳送至 S3 儲存貯體來源的所有請求。

絕不簽署原始伺服器請求

此設定於主控台中命名為 Do not sign requests (請勿簽署請求)，或 API、CLI 和 AWS CloudFormation中的 `never`。使用此設定，關閉使用此原始存取控制之所有分佈中的所有原始伺服器的原始存取控制。與從所有使用其原始伺服器和分佈中逐一移除原始存取控制相比，此可節省時間和精力。使用此設定， CloudFront 不會簽署傳送至 S3 儲存貯體來源的任何請求。

> **Warning**
>
> 如要使用此設定，S3 儲存貯體原始伺服器必須可公開存取。如果您對不可公開存取的 S3 儲存貯體來源使用此設定，則 CloudFront 無法存取原始碼。S3 儲存貯體來源會傳回錯誤，CloudFront 並將這些錯誤傳 CloudFront 送給檢視器。

請勿覆寫檢視器 (用戶端) **`Authorization`** 標題

此設定於主控台中命名為 Do not override authorization header (請勿覆寫授權標頭)，或於 API、CLI 和 AWS CloudFormation中的 `no-override`。如果您只想在對應的檢視器 CloudFront 要求不包含`Authorization`標頭時簽署原始請求，請使用此設定。使用此設定時，會在檢視器要求存在時 CloudFront 傳遞來自檢視器要求的`Authorization`標頭，但是當檢視器要求未包含`Authorization`標頭時，會簽署原始要求 (新增其自己的`Authorization`標頭)。

> **Warning**
>
> 如要從檢視器請求傳遞 `Authorization` 標題，您必須將 `Authorization` 標題新增至快取政策中，適用於使用與此原始存取控制相關聯之 S3 儲存貯體原始伺服器的所有快取行為。

使用原始存取身分 (舊版，不建議使用)

原始存取身分概觀

CloudFront 原始存取身分識別 (OAI) 提供與原始存取控制 (OAC) 類似的功能，但不適用於所有案例。這就是為什麼我們建議改用 OAC。具體而言，OAI 不支援：

- 所有 Amazon S3 儲存貯體 AWS 區域，包括選擇加入區域
- 使用 AWS KMS的 Amazon S3 伺服器端加密 (SSE-KMS)
- 對 Amazon S3 的動態請求 (PUT、POST 或 DELETE)
- 2022 年 12 月後 AWS 區域 推出的新功能

如需從 OAI 遷移至 OAC 的相關資訊，請參閱 the section called “從原始存取身分 (OAI) 遷移至原始存取控制 (OAC)”。

授予原始存取身分的許可，以讀取 Amazon S3 儲存貯體中的檔案

當您使用 CloudFront 主控台建立 OAI 或將其新增至分發時，可以自動更新 Amazon S3 儲存貯體政策，以授予 OAI 存取儲存貯體的權限。或者，您可以選擇手動建立或更新儲存貯體政策。無論您使用哪種方法，您仍應檢閱許可以確認下列事項：

- 您的 CloudFront OAI 可以代表透 CloudFront過要求檔案的檢視者存取值區中的檔案。
- 檢視者無法使用 Amazon S3 網址在外部存取您的檔案 CloudFront。

> **Important**
>
> 如果您設定 CloudFront 為接受和轉寄所有 CloudFront 支援的 HTTP 方法，請務必將所需的權限授與 CloudFront OAI。例如，如果您設定為 CloudFront 接受和轉寄使用該DELETE方法的要求，請將儲存貯體原則設定為適當地處理要DELETE求，以便檢視者只能刪除您想要的檔案。

使用 Amazon S3 儲存貯體原則

您可以透過下列方式建立或更新儲存貯體政策，讓 CloudFront OAI 存取權存取 Amazon S3 儲存貯體中的檔案：

- 使用 Amazon S3 主控台中的 Amazon S3 儲存貯體的Permissions (許可) 標籤。
- PutBucketPolicy在 Amazon S3 API 中使用。

- 使用 CloudFront 主控台。當您在 CloudFront 主控台中將 OAI 新增至原始設定時，可以選擇 [是，更新值區政策]，告知 CloudFront 您代表您更新值區政策。

如果您手動更新儲存貯體政策，請務必：

- 在政策中指定正確的 OAI 做為 `Principal`。
- 授予 OAI 代表檢視器存取物件所需的許可。

如需詳細資訊，請參閱下列區段。

在儲存貯體政策中指定 OAI 做為 **`Principal`**

若要於 Amazon S3 儲存貯體政策中指定 OAI 做為 `Principal`，請使用包含 OAI ID 的 OAI Amazon Resource Name (ARN)。例如：

```
"Principal": {
    "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity <origin
 access identity ID>"
}
```

在「安全性」、「來源存取」、「身分識別 (舊版)」下方的 CloudFront 主控台中找到 OAI ID。或者，在 CloudFront API ListCloudFrontOriginAccessIdentities中使用。

授予權限給一個 OAI

如要授予 OAI 存取 Amazon S3 儲存貯體中物件的許可，請使用與特定 Amazon S3 API 作業相關政策中的動作。例如，`s3:GetObject` 動作可讓 OAI 讀取儲存貯體中的物件。如需詳細資訊，請參閱以下章節中的範例，或參閱 Amazon Simple Storage Service 使用者指南中的 Amazon S3 動作。

Amazon S3 儲存貯體政策範例

下列範例顯示允許 CloudFront OAI 存取 S3 儲存貯體的 Amazon S3 儲存貯體政策。

在「安全性」、「來源存取」、「身分識別 (舊版)」下方的 CloudFront 主控台中找到 OAI ID。或者，在 CloudFront API ListCloudFrontOriginAccessIdentities中使用。

Example 授予 OAI 讀取存取權限的 Amazon S3 儲存貯體政策

下列範例可讓 OAI 讀取指定儲存貯體中的物件 (`s3:GetObject`)。

```
{
    "Version": "2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity <origin access identity ID>"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}
```

Example 授予 OAI 讀取和寫入存取權限的 Amazon S3 儲存貯體政策

下列範例可讓 OAI 讀取和寫入指定儲存貯體中的物件 (`s3:GetObject` 和 `s3:PutObject`)。這可讓檢視者透過將檔案上傳到您的 Amazon S3 儲存貯體 CloudFront。

```
{
    "Version": "2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity <origin access identity ID>"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}
```

使用 Amazon S3 物件 ACL (不建議)

> **Important**
>
> 我們建議使用 Amazon S3 儲存貯體政策 向 OAI 提供 S3 儲存貯體的存取權。您可以使用存取控制清單 (ACL)，如本節中所述，但我們不建議此方法。
> Amazon S3 建議設定 S3 物件擁有權為已強制執行儲存貯體擁有者，這意味著儲存貯體及其中物件的 ACL 被停用。當您將此設定套用至「物件擁有權」 時，您必須使用儲存貯體政策來授與 OAI 的存取權 (請參閱上一節)。
> 本節僅適用於需要 ACL 的舊式使用案例。

您可以透過下列方式建立或更新檔案的 ACL，讓 CloudFront OAI 存取權存取 Amazon S3 儲存貯體中的檔案：

- 使用 Amazon S3 主控台中 Amazon S3 物件的 Permissions (許可) 標籤。
- PutObjectAcl在 Amazon S3 API 中使用。

當您使用 ACL 授予對 OAI 的存取權限時，您必須使用 OAI 的 Amazon S3 正式使用者 ID 來指定 OAI。在 CloudFront 主控台中，您可以在 [安全性]、[原始存取]、[身分識別 (舊版)] 下找到此 ID。如果您使用 CloudFront API，請使用建立 OAI 時傳回的`S3CanonicalUserId`元素值，或在 CloudFront API ListCloudFrontOriginAccessIdentities中呼叫。

在僅支援簽章版本 4 驗證的 Amazon S3 區域中使用原始存取身分

較新 Amazon S3 區域請求您使用簽章版本 4 進行驗證請求 (如需每個 Amazon S3 區域支援的簽章版本，請參閱《AWS 一般參考》中的 Amazon Simple Storage Service 端點和配額。) 如果您使用的是原始存取身分，且您的儲存貯體位於需要簽章版本 4 的其中一個區域，請注意下列事項：

- `DELETE`、`GET`、`HEAD`、`OPTIONS` 和 `PATCH` 請求可以在沒有授權的情況下得到支援。
- 不支援 `POST` 請求。

限制對 Application Load Balancers 的存取

對於在 Elastic Load Balancing 中由面向網際網路的應用程式負載平衡器提供的 Web 應用程式或其他內容， CloudFront 可以快取物件並直接提供給使用者 (檢視者)，從而降低應用程式負載平衡器的負載。面向網際網路的負載平衡器具有可公開解析的 DNS 名稱，並透過網際網路將來自用戶端的要求路由到目標。

CloudFront還可以幫助減少延遲，甚至吸收一些分佈式拒絕服務（DDoS）攻擊。

但是，如果使用者可以直接略過 CloudFront 並存取您的 Application Load Balancer，您將無法獲得這些好處。但是您可以設定 Amazon CloudFront 和 Application Load Balancer，以防止使用者直接存取應用程式負載平衡器。這讓使用者只能透過以下方式存取 Application Load Balancer CloudFront，確保您獲得使用的好處 CloudFront。

若要防止使用者直接存取應用程式負載平衡器並僅允許透過存取 CloudFront，請完成下列高階步驟：

1. 設定 CloudFront 以將自訂 HTTP 標頭新增至傳送至 Application Load Balancer 的要求。
2. 將 Application Load Balancer 設定為僅轉寄包含自訂 HTTP 標頭的請求。
3. (選用) 需要 HTTPS 來改善此解決方案的安全性。

如需詳細資訊，請參閱下列主題。完成這些步驟後，使用者只能透過存取您的 Application Load Balancer CloudFront。

主題

設定 CloudFront 為將自訂 HTTP 標頭新增至要求

您可以設定 CloudFront 為將自訂 HTTP 標頭新增至傳送至原始伺服器的要求 (在本例中為 Application Load Balancer)。

> **Important**
>
> 此用例依賴於保留自訂標頭名稱和值密碼。如果標頭名稱和值不是密碼，其他 HTTP 用戶端可能會將它們包含在直接傳送至 Application Load Balancer 的請求中。這可能會導致 Application Load Balancer 的運作方式，就像要求來自其他 CloudFront 時間一樣。若要防止這種情況，請保留自訂標頭名稱和值密碼。

您可以設 CloudFront 定使用 CloudFront主控台或 CloudFront API 將自訂 HTTP 標頭新增至原始請求。 AWS CloudFormation

若要新增自訂 HTTP 標頭 (CloudFront 主控台)

在 CloudFront 主控台中，使用 Origin 設定中的 Origin 自訂標題設定。輸入標頭名稱及其值，如下列範例所示。

> **Note**
>
> 此範例中的標頭名稱和值僅為演示。在生產中，使用隨機產生的值。將標頭名稱和值視為安全登入資料，如使用者名稱和密碼。

當您為現有發行版建立或編輯原點時，以及建立新 CloudFront 發佈時，您可以編輯 Origin 自訂標題設定。如需詳細資訊，請參閱 更新分佈 及 建立分發。

新增自訂 HTTP 標頭 (AWS CloudFormation)

在 AWS CloudFormation 範本中，使用`OriginCustomHeaders`屬性，如下列範例所示。

> **Note**
>
> 此範例中的標頭名稱和值僅為演示。在生產中，使用隨機產生的值。將標頭名稱和值視為安全登入資料，如使用者名稱和密碼。

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  TestDistribution:
    Type: 'AWS::CloudFront::Distribution'
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: app-load-balancer.example.com
            Id: Example-ALB
            CustomOriginConfig:
              OriginProtocolPolicy: https-only
              OriginSSLProtocols:
```

```
            - TLSv1.2
        OriginCustomHeaders:
           - HeaderName: X-Custom-Header
             HeaderValue: random-value-1234567890
    Enabled: 'true'
    DefaultCacheBehavior:
      TargetOriginId: Example-ALB
      ViewerProtocolPolicy: allow-all
      CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6
    PriceClass: PriceClass_All
    ViewerCertificate:
      CloudFrontDefaultCertificate: 'true'
```

若要取得更多資訊，請參閱《AWS CloudFormation 使用指南》中的原點和OriginCustomHeader性質。

若要新增自訂 HTTP 標頭 (CloudFront API)

在 CloudFront API 中，使用裡面的`CustomHeaders`對象`Origin`。如需詳細資訊，請參閱CreateDistributionAmazon CloudFront API 參考UpdateDistribution中的和，以及您的開發套件或其他 API 用戶端的說明文件。

有一些標頭名稱不能指定為來源自訂標頭。如需詳細資訊，請參閱 無法新增至原始請求的 CloudFront 自訂標頭。

將 Application Load Balancer 設定為僅轉寄包含特定標頭的請求

設定為將自訂 HTTP 標頭新增 CloudFront 至傳送至 Application Load Balancer 的要求後 (請參閱上一節)，您可以將負載平衡器設定為僅轉寄包含此自訂標頭的要求。您可以透過在負載平衡器的接聽程式中新增規則並修改預設規則來執行此動作。

必要條件

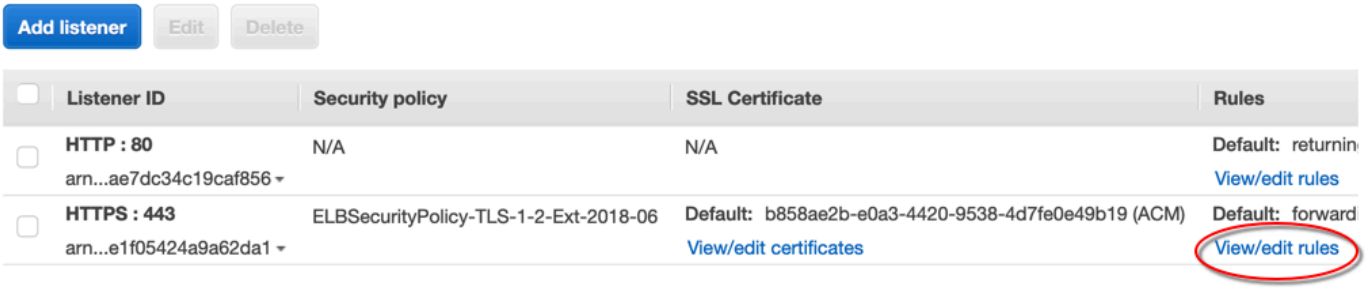
若要使用下列程序，您必須具有至少一個接聽程式的 Application Load Balancer。如果您尚未建立應用程式負載平衡器，請參閱應用程式負載平衡器使用者指南中的建立應用程式負載平衡器。

下列程序會修改 HTTPS 接聽程式。您可以使用相同的程序來修改 HTTP 接聽程式。

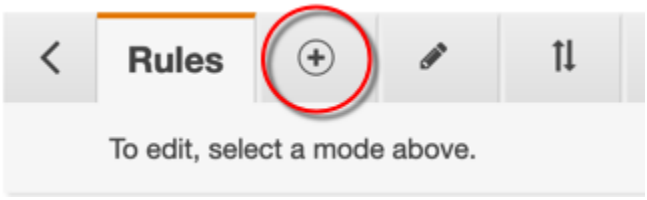
更新 Application Load Balancer 接聽程式中的規則

1. 在 Amazon EC2 主控台中開啟負載平衡器頁面。

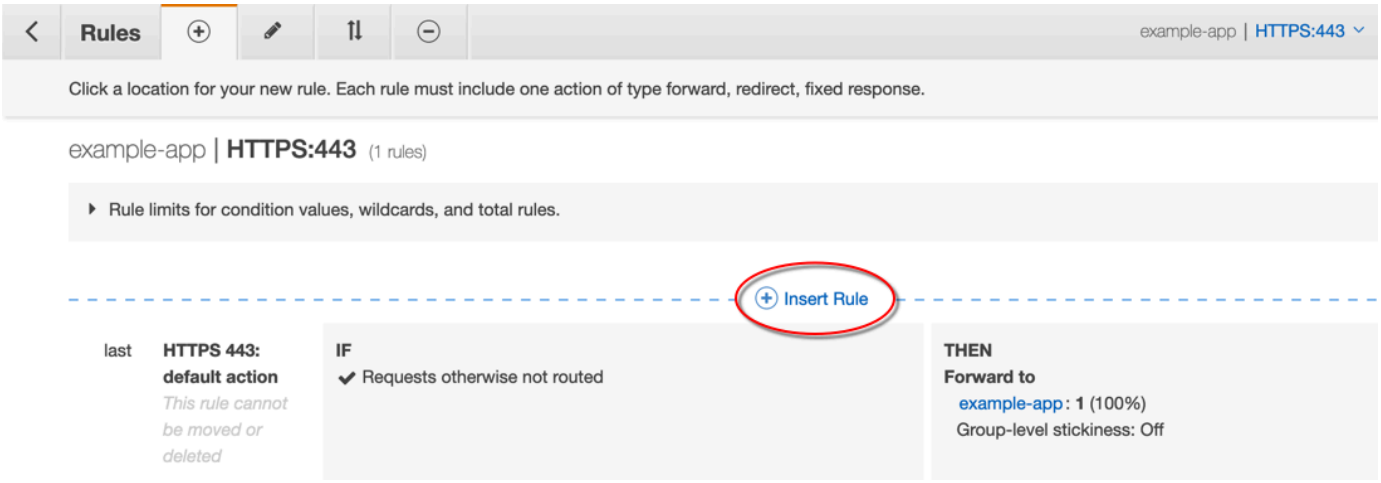
2. 選擇作為 CloudFront 發行版本來源的負載平衡器，然後選擇「接聽程式」索引標籤。
3. 針對您要修改的接聽程式，請選擇檢視/編輯規則。



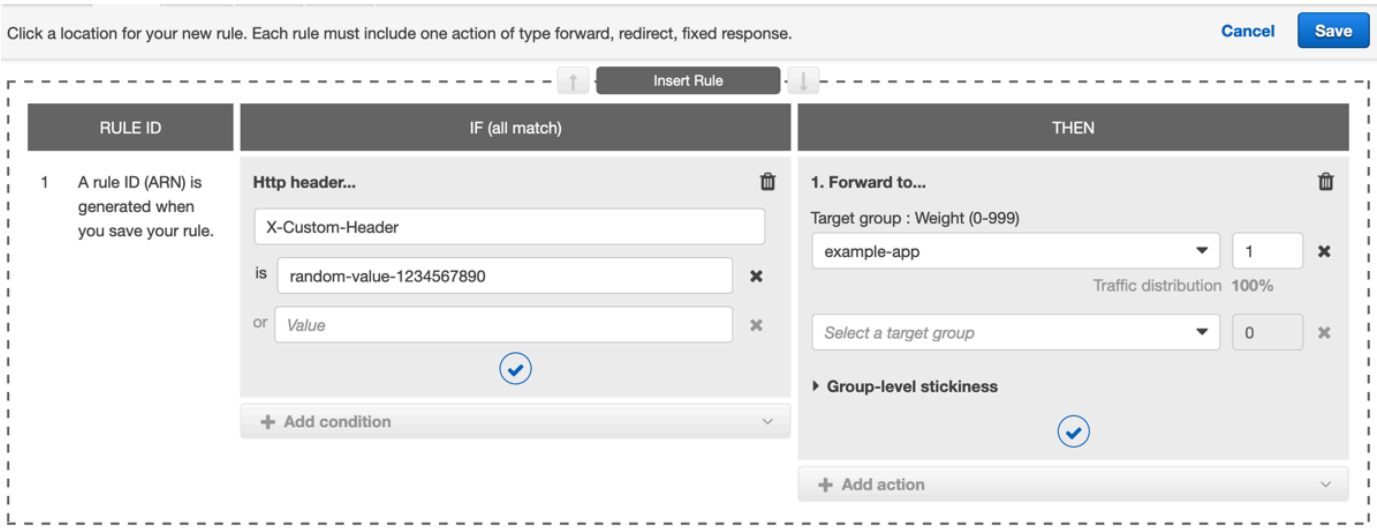
4. 選擇圖示以新增規則。



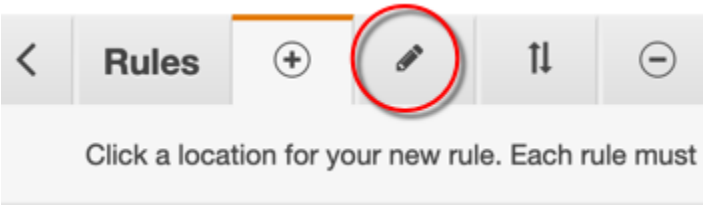
5. 選擇 Insert Rule (插入規則)。



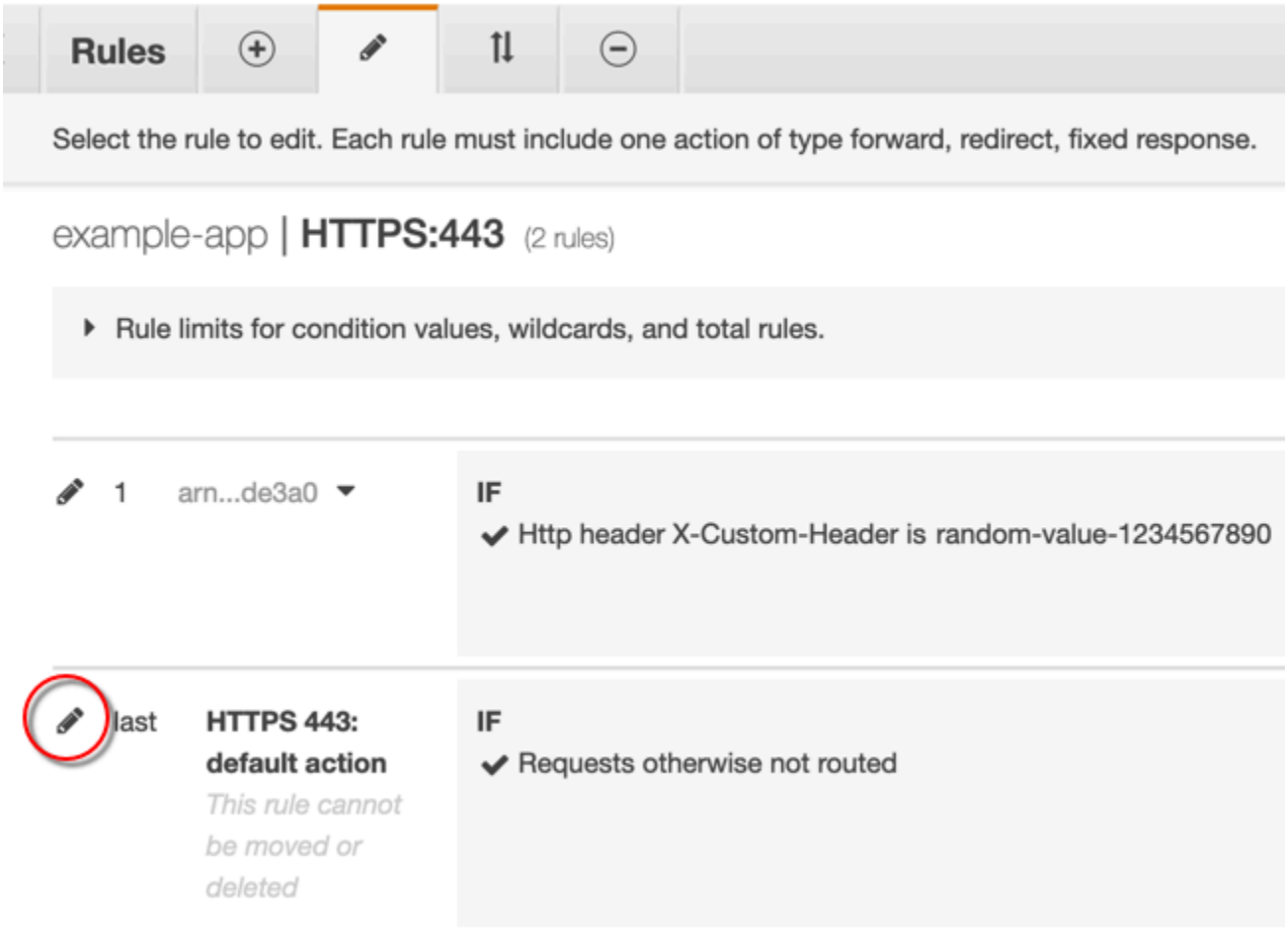
6. 對於新規則，請執行下列動作：
   a. 選擇新增條件，然後選擇 HTTP 標頭。指定您在中新增為原始自訂標頭的 HTTP 標頭名稱和值 CloudFront。
   b. 選擇新增動作，然後選擇轉寄至。選擇要轉寄請求的目標群組。
   c. 選擇儲存以建立新規則。



7. 選擇圖示以編輯規則。



8. 選擇預設規則的編輯圖示。



9. 針對預設規則，請執行下列動作：

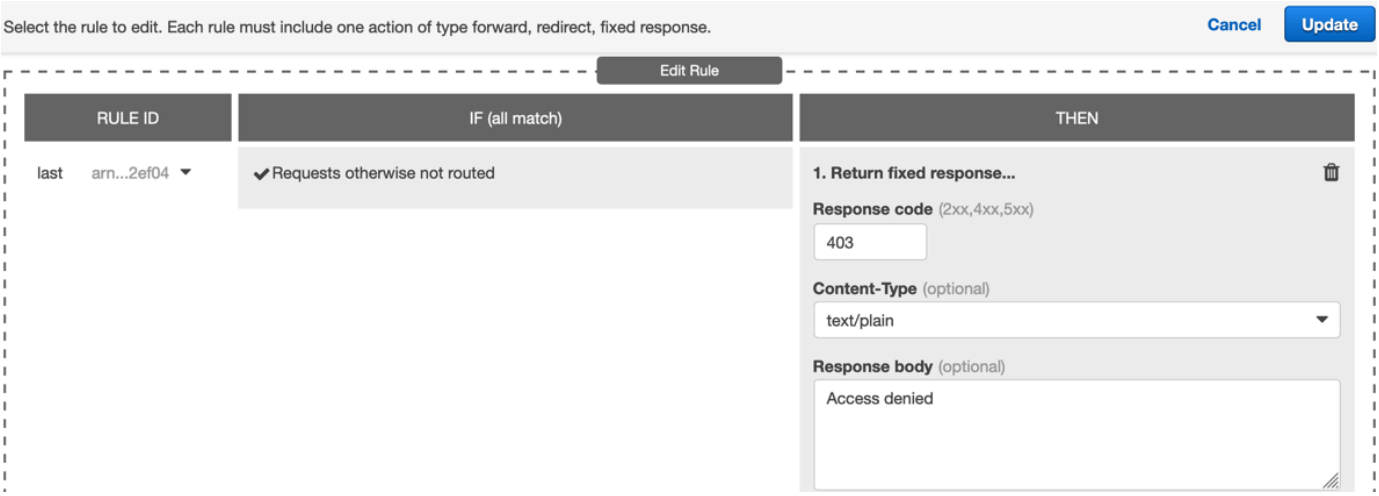
   a. 刪除預設動作。

   b. 選擇新增動作，然後選擇傳回固定回應。

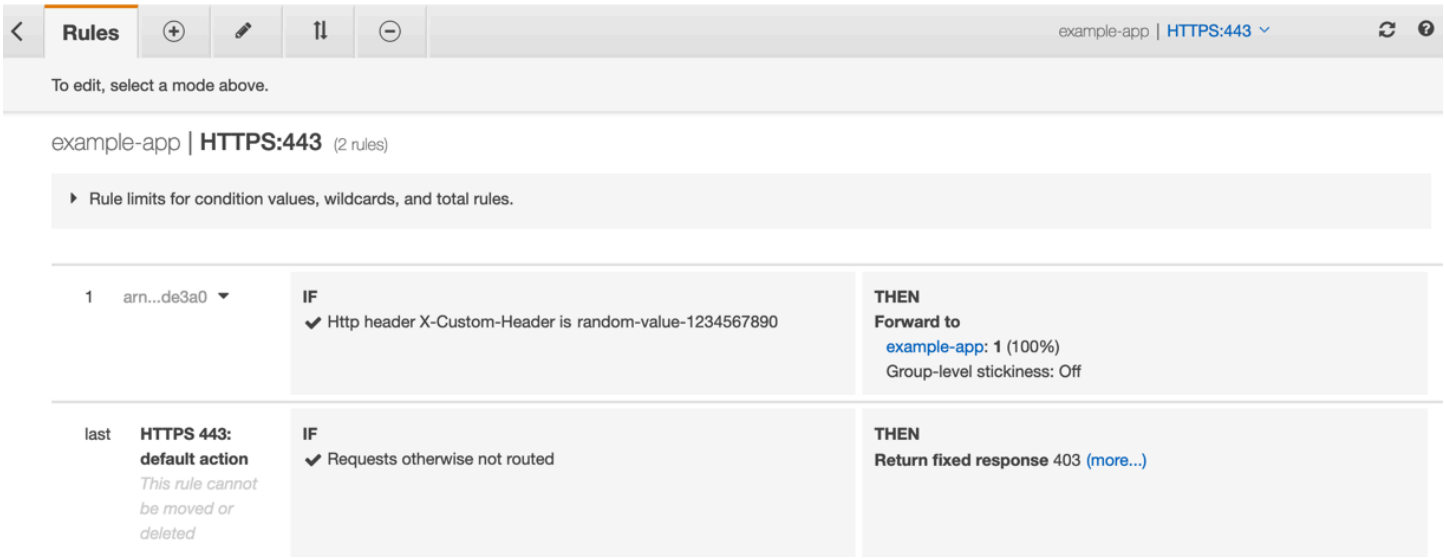
   c. 在回應代碼中，輸入 **403**。

   d. 在回應主體中，輸入 **Access denied**。

   e. 選擇更新以更新預設規則。



完成這些步驟後，負載平衡器接聽程式會有兩個規則，如下圖所示。第一個規則會轉寄包含 HTTP 標頭 (來自的要求 CloudFront) 的要求。第二個規則會對所有其他要求 (不來自的要求 CloudFront) 傳送固定回應。



您可以將要求傳送至您的 CloudFront 散發，並傳送至您的 Application Load Balancer，以驗證解決方案是否有效。傳 CloudFront 回 Web 應用程式或內容的要求，而直接傳送至 Application Load Balancer 的要求會傳回含純文字訊息的回403應`Access denied`。

(選用) 改善此解決方案的安全性

為了提高此解決方案的安全性，您可以將分發 CloudFront 配置為在向應用 Application Load Balancer 發送請求時始終使用 HTTPS。請記住，此解決方案僅適用於保留自訂標頭名稱和值密碼的情況下。使用 HTTPS 有助於防止竊聽者發現標頭名稱和值。我們也建議定期輪換標頭名稱和值。

針對來源請求使用 HTTPS

若要設定為針對原始要求使用 HTTPS，請 CloudFront 將「原始通訊協定原則」設定設定為「僅限 HTTPS」。此設定可在主 CloudFront 控台 AWS CloudFormation和 CloudFront API 中使用。如需詳細資訊，請參閱 通訊協定 (僅限自訂原始伺服器)。

當您設定 CloudFront 為針對原始要求使用 HTTPS 時，下列條件也適用：

- 您必須設定 CloudFront 使用原始要求原則將`Host`標頭轉寄至原始位置。您可以使用AllViewer 受管理的來源請求策略。
- 請確定您的 Application Load Balancer 具有 HTTPS 接聽程式 (如前一節所示)。如需詳細資訊，請參閱應用程式負載平衡器使用者指南中的建立 HTTPS 接聽程式。使用 HTTPS 接聽程式時，您必須擁有與路由到 Application Load Balancer 的網域名稱相符的 SSL/TLS 憑證。
- 的 SSL/TLS 憑證只 CloudFront 能在中 (ACM) 中要求 AWS Certificate Manager (或匯入)。`us-east-1` AWS 區域 由於 CloudFront 是全球服務，因此它會自動將憑證從「`us-east-1`區域」散發到與您的散 CloudFront發相關聯的所有區域。
  - 例如，如果您在區域中有 Application Load Balancer (ALB)，則必須同時在`ap-southeast-2`區域 (用於在和 ALB 來源之間使用 HTTPS) 和`ap-southeast-2`區域 (用於在檢視器 CloudFront 和之間使用 HTTPS) 中`us-east-1`設定 SSL/TLS 憑證。 CloudFront這兩個憑證都應與路由到應 Application Load Balancer 的網域名稱相符。如需詳細資訊，請參閱 AWS 區域 為 AWS Certificate Manager。
- 如果 Web 應用程式的一般使用者 (也稱為檢視者或用戶端) 可以使用 HTTPS，您也可以設定 CloudFront 為偏好 (甚至需要) 使用者的 HTTPS 連線。若要這樣做，請使用檢視器協定政策設定。您可以將它設定為將最終使用者從 HTTP 重新導向至 HTTPS，或拒絕使用 HTTP 的請求。此設定可在主 CloudFront 控台 AWS CloudFormation和 CloudFront API 中使用。如需詳細資訊，請參閱 檢視器通訊協定政策。

輪換標頭名稱和值

除了使用 HTTPS 之外，我們還建議定期輪換標頭名稱和值。執行這項操作的高階步驟如下：

1. 設定 CloudFront 以將其他自訂 HTTP 標頭新增至傳送至 Application Load Balancer 的要求。

2. 更新 Application Load Balancer 接聽程式規則，以轉寄包含此其他自訂 HTTP 標頭的請求。
3. 設定 CloudFront 以停止將原始自訂 HTTP 標頭新增至傳送至 Application Load Balancer 的要求。
4. 更新 Application Load Balancer 接聽程式規則，以停止轉寄包含來源自訂 HTTP 標頭的請求。

如需完成這些步驟的詳細資訊，請參閱前一章節。

(選擇性) 透過使用 AWS-managed 前置詞清單來限制對原點的存取 CloudFront

若要進一步限制對 Application Load Balancer 的存取，您可以設定與 Application Load Balancer 相關聯的安全性群組，以便僅接受服務使用 AWS-managed 前置詞清單 CloudFront 時來自的流量。如此可防止非源 CloudFront 自網路層 (第 3 層) 或傳輸層 (第 4 層) 的流量到達您的 Application Load Balancer。

如需詳細資訊，請參閱使用 Amazon CloudFront 部落格文章的 AWS-managed 前置詞清單限制對您來源的存取。

限制您內容的地理分佈

您可以使用地理限制 (有時稱為地理區域封鎖)，防止特定地理位置的使用者存取您透過 Amazon 分發發 CloudFront 佈的內容。若要使用地理限制，您有兩個選擇：

- 使用地 CloudFront 理限制功能。使用此選項可限制對與分佈相關聯的所有檔案的存取，並限制在國家/地區層級的存取。
- 使用第三方地理位置服務。使用此選項來針對與分佈相關聯檔案的子集作限制存取，或在比國家/地區層級更細的層級作限制存取。

主題

- 使用 CloudFront 地理限制
- 使用第三方地理位置服務

使用 CloudFront 地理限制

當用戶請求您的內容時，無論用戶位於何處， CloudFront 通常都會提供所請求的內容。如果您需要防止特定國家/地區的使用者存取您的內容，您可以使用 CloudFront 地理限制功能執行下列其中一項作業：

- 只有當使用者位於您允許清單上其中一個核准國家/地區時，才能存取您的內容。
- 如果使用者位於您封鎖清單上的被禁國家/地區，則阻止其存取您的內容。

例如，如果要求來自您未獲授權散佈內容的國家/地區，您可以使用 CloudFront 地理限制來封鎖要求。

> **Note**
>
> CloudFront 使用協力廠商資料庫來決定使用者的位置。IP 地址與國家/地區之間的映射的準確性因區域而異。根據最近的測試，整總體準確性為 99.8%。如果 CloudFront 無法判斷使用者的位置，請 CloudFront 提供使用者要求的內容。

以下是地理限制的運作方式：

1. 假設您有權僅在列支敦斯登分配您的內容。您可以更新您的CloudFront 發行版，以新增僅包含列支敦士登的允許清單。(或者，您可以新增包含除了列支敦斯登以外的每個國家/地區的封鎖清單。)
2. 摩納哥的使用者會要求您的內容，而 DNS 會將要求路由至義大利米蘭的 CloudFront 節點。
3. 米蘭的邊緣節點會查詢您的分佈，並確定摩納哥的使用者沒有下載您的內容的許可。
4. CloudFront 將 HTTP 狀態碼傳回`403 (Forbidden)`給使用者。

您可以選擇配置 CloudFront 為將自定義錯誤消息返回給用戶，並且可以指定 CloudFront 要為請求的文件緩存錯誤響應的時間長度。預設值為 10 秒。如需詳細資訊，請參閱 針對特定的 HTTP 狀態碼建立自訂錯誤頁面。

地理限制適用於整個分佈。如果您需要對部分內容套用一項限制，而對內容的另一部分套用不同的限制(或無限制)，您必須建立個別的 CloudFront 散佈或使用第三方地理定位服務。

如果您啟用 CloudFront 標準記錄檔 (存取記錄)，您可以搜尋值所在的記錄項目 `sc-status` (HTTP 狀態碼)，以識別 CloudFront 拒絕的要求`403`。不過，只使用標準記錄檔時，您無法根據使用者的位置來區分 CloudFront拒絕的要求與 CloudFront 拒絕的要求，因為使用者因為其他原因沒有存取檔案的權限。如果您有第三方地理位置服務 (例如 Digital Element) MaxMind，或者，您可以根據存取記錄中 `c-ip` (用戶端 IP) 欄中的 IP 位址來識別要求的位置。如需 CloudFront 標準記錄檔的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

下列程序說明如何使用 CloudFront 主控台將地理限制新增至現有發行版。如需有關如何使用主控台建立分佈的詳細資訊，請參閱建立分發。

若要將地理限制新增至 CloudFront Web 分發 (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇分佈，然後選擇您要更新的分佈。
3. 選擇安全性分頁，然後選擇地理限制。
4. 選擇編輯。
5. 選擇 Allow list (允許清單) 建立允許的國家/地區清單，或 Block list (封鎖清單) 建立封鎖的國家/地區清單。
6. 將所需的國家/地區新增至清單中，然後選擇 Save changes (儲存變更)。

使用第三方地理位置服務

使用 CloudFront 地理限制功能，您可以在國家/地區層級控制您使用指定網頁發佈的所有檔案的內容發佈。如果您有地理限制不符合國家/地區界限的使用案例，或者如果您想要限制只存取您透過特定發行版所提供的某些檔案，您可以結CloudFront 合第三方地理定位服務。這讓您不僅可以根據國家/地區控制對內容的存取，還可以根據城市、郵遞區號，甚至是緯度和經度來控制對內容的存取。

當您使用第三方地理定位服務時，我們建議您使用 CloudFront 已簽署的 URL，您可以使用該 URL 指定到期日期和時間，此時間之後該 URL 不再有效。此外，我們建議您使用 Amazon S3 儲存貯體做為來源，因為之後您可以使用 CloudFront 來源存取控制來防止使用者直接從來源存取您的內容。如需有關已簽署的 URL 和原始存取控制的詳細資訊，請參閱 使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。

以下步驟解釋如何使用第三方地理位置服務來控制對檔案的存取。

使用協力廠商地理定位服務來限制對發佈中檔案的存取 CloudFront

1. 使用地理位置服務取得帳戶。
2. 將內容上傳至 Amazon S3 儲存貯體。
3. 設定 Amazon CloudFront 和 Amazon S3 以提供私有內容。如需詳細資訊，請參閱 使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。
4. 寫入您的 Web 應用程式，以執行下列動作：
   - 將每個使用者請求的 IP 地址傳送到地理位置服務。
   - 評估地理位置服務的傳回值，以判斷使用者是否位於您 CloudFront 要發佈內容的位置。

- 如果您想要將內容發佈到使用者的位置，請為您的 CloudFront 內容產生已簽署的 URL。如果您不想將內容分配到該位置，請將 HTTP 狀態碼 403 (Forbidden) 傳回給使用者。或者，您可以配置 CloudFront 為返回自定義錯誤消息。如需詳細資訊，請參閱 the section called “針對特定的 HTTP 狀態碼建立自訂錯誤頁面”。

  如需詳細資訊，請參閱您所使用的地理位置服務的文件。

您可以使用 Web 伺服器變動變數來獲得瀏覽您的網站的使用者的 IP 位址。請注意以下警告：

- 如果您的 Web 伺服器透過負載平衡器無法連線至網際網路，則可以使用 Web 伺服器變數來獲得遠端 IP 地址。不過，此 IP 地址不一定是使用者的 IP 地址。它也可以是代理伺服器的 IP 地址，取決於使用者如何連接到網際網路。
- 如果您的 Web 伺服器透過負載平衡器連線至網際網路，則 Web 伺服器變數可能包含負載平衡器的 IP 地址，而非使用者的 IP 地址。在這個組態中，我們建議您使用 `X-Forwarded-For` HTTP 標頭中的最後一個 IP 地址。此標頭通常包含多個 IP 地址，其中大部分用於代理或負載平衡器。清單中的最後一個 IP 地址最有可能與使用者的地理位置相關聯。

如果您的 Web 伺服器無法連接到負載平衡器，我們建議您使用 Web 伺服器變數而非 `X-Forwarded-For` 標頭，以避免 IP 地址詐騙。

使用欄位層級加密來協助保護敏感資料

使用 Amazon CloudFront，您可以使用 end-to-end HTTPS 強制執行與原始伺服器的安全連線。欄位層級加密可新增額外的安全層，可讓您在整個系統處理過程中保護特定的資料，以便只有特定應用程式才能看到它。

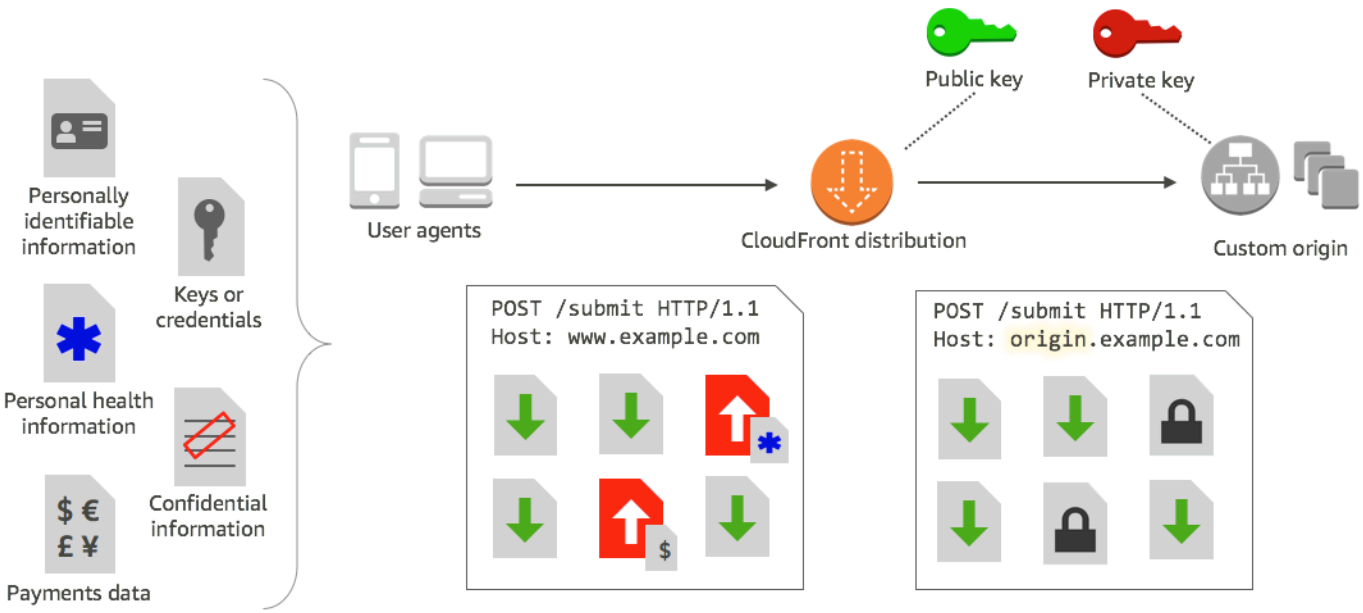
欄位層級加密可讓您的使用者安全地將敏感資訊上傳到您的 Web 伺服器。使用者提供的敏感資訊會在邊緣、靠近使用者處加密，並在整個應用程式堆疊中保持加密。此加密可確保只有需要資料的應用程式 (並具有可解密的登入資料) 才能執行這項操作。

要使用字段級加密，請在 CloudFront 配置分發時，請在 POST 請求中指定要加密的字段集以及用於加密它們的公鑰。您可以在請求中加密多達 10 個資料欄位。(您無法使用欄位層級加密對請求中的所有資料進行加密；您必須指定要加密的各個欄位。)

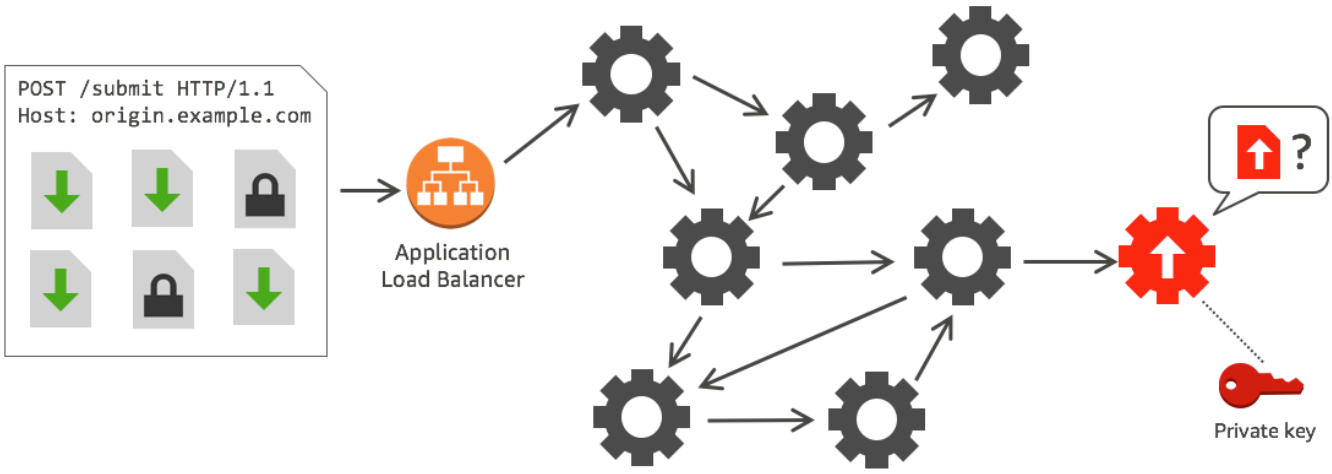
當將帶有欄位層級加密的 HTTPS 請求轉發到原始伺服器，並且該請求被路由到原始應用程式或子系統中時，敏感資料仍然被加密，從而降低了敏感資料遭受洩露或意外遺失資料的風險。出於業務考量而需

對敏感資料存取的元件，例如需要存取信用卡號碼的付款系統，可以使用適當的私有金鑰來解密和存取該資料。

> **Note**
>
> 為了使用欄位層級加密，您的原始伺服器必須支援區塊編碼。



CloudFront 欄位層級加密使用非對稱加密，也稱為公開金鑰加密。您提供公開金鑰 CloudFront，而您指定的所有敏感資料都會自動加密。您提供的金鑰 CloudFront 無法用於解密加密值；只有您的私密金鑰才能執行此動作。



主題

- 欄位層級加密概觀
- 設定欄位層級加密
- 在您的原始伺服器解密資料欄位

欄位層級加密概觀

以下步驟概述了設定欄位層級加密。對於特定的步驟，請參閱設定欄位層級加密。

1. 取得公有金鑰/私有金鑰對。在 CloudFront中開始設定欄位層級加密之前，您必須取得並新增公開金鑰。
2. 建立欄位層級加密的設定檔。您在中 CloudFront建立的欄位層級加密設定檔會定義您要加密的欄位。
3. 建立欄位層級加密的組態。組態會指定要使用的設定檔 (根據請求的內容類型或查詢引數)，用來將特定資料欄位加密。您也可以針對不同案例選擇所需的請求轉發行為選項。例如，您可以設定要求 URL 中的 query 引數所指定的描述檔名稱不存在於中的行為 CloudFront。
4. 至快取行為的連結。將組態連結至發行版的快取行為，以指定何時 CloudFront應該加密資料。

設定欄位層級加密

依照以下步驟開始使用欄位層級加密。若要了解欄位層級加密的配額 (先前稱為限制)，請參閱 配額。

- 步驟 1：建立 RSA 金鑰對
- 步驟 2：將您的公鑰添加到 CloudFront
- 步驟 3：建立欄位層級加密的設定檔。
- 步驟 4：建立組態
- 步驟 5：將組態新增到快取行為

步驟 1：建立 RSA 金鑰對

若要開始使用，您必須建立包含公有金鑰和私有金鑰的 RSA 金鑰對。公開金鑰可 CloudFront 以加密資料，而私密金鑰可讓原始元件解密已加密的欄位。您可以使用 OpenSSL 或其他工具來建立金鑰對。金鑰大小必須為 2048 個位元。

例如，如果您使用的是 OpenSSL，您可以使用以下命令來產生長度為 2048 個位元組的金鑰對，並將其儲存在檔案 `private_key.pem` 中：

```
openssl genrsa -out private_key.pem 2048
```

產生的檔案同時包含公有和私有金鑰。要從該檔案中擷取公有金鑰，請執行以下命令：

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

公有金鑰檔案 (`public_key.pem`) 包含您在下列步驟中貼上的編碼鍵值。

步驟 2：將您的公鑰添加到 CloudFront

取得 RSA key pair 後，請將您的公開金鑰新增至 CloudFront。

將您的公開金鑰新增至 CloudFront (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 Public key (公有金鑰)。
3. 選擇 Add public key (新增公有金鑰)。
4. 在 Key name (金鑰名稱) 中，輸入金鑰的獨特名稱。名稱不能有空格，並且只能包含英數字元、底線 (_) 和 連字號 (-)。最大字元數為 128。

5. 針對 Key value (鍵值)，貼上公有金鑰的編碼鍵值 (包括 `-----BEGIN PUBLIC KEY-----` 和 `-----END PUBLIC KEY-----` 行)。
6. 針對 Comment (註解)，加入選擇性的註解。例如，您可以包含公有金鑰的到期日期。
7. 選擇 Add (新增)。

您可以重複程序中的步驟來新增更多要與之 CloudFront 搭配使用的金鑰。

步驟 3：建立欄位層級加密的設定檔。

在您新增至少一個公開金鑰之後 CloudFront，請建立描述檔，告知要加密CloudFront 哪些欄位。

建立欄位層級加密的設定檔 (主控台)

1. 在導覽窗格中，選擇 Field-level encryption (欄位層級加密)。
2. 選擇 Create profile (建立設定檔)。
3. 填寫下列欄位：

   設定檔名稱

   請輸入設定檔的專屬名稱。名稱不能有空格，並且只能包含英數字元、底線 (_) 和 連字號 (-)。最大字元數為 128。

   公有金鑰名稱

   在下拉式清單中，選擇您在步驟 2 CloudFront 中新增至的公開金鑰名稱。 CloudFront 使用金鑰來加密您在此設定檔中指定的欄位。

   供應商名稱

   輸入片語以協助識別您的金鑰，例如您取得到金鑰對的供應商。當應用程式解密資料欄位時，將需要此資訊以及私有金鑰。供應商名稱不能有空格，並且只能包含英數字元、冒號 (:)、底線 (_) 和 連字號 (-)。最大字元數為 128。

   欄位名稱模式符合

   輸入要加密的資料欄位名稱或識別要 CloudFront求中資料欄位名稱的模式。選擇 + 選項來新增您想要使用此金鑰加密的所有欄位。

   對於欄位名稱模式，您可以輸入資料欄位的完整名稱 DateOfBirth，例如，或僅輸入萬用字元 (*) 的名稱的第一部分，例如 CreditCard *。除了可選的萬用字元 (*) 外，欄位名稱模式必須只包含英數字元、方括號 ([和])、句號 (.)、底線 (_) 和連字號 (-)。

請確定不要為不同的欄位名稱模式使用重疊的字元。例如，如果您有 ABC* 的欄位名稱模式，則不能新增另一個 AB* 的欄位名稱模式。此外，欄位名稱區分大小寫，可以使用的字元數上限為 128。

註解

(選用) 輸入有關此設定檔的評論。最多可使用 128 個字元。

4. 填寫完欄位後，選擇 Create profile (建立設定檔)。
5. 如果您想新增更多設定檔，請選擇 Add profile (新增設定檔)。

步驟 4：建立組態

建立一或多個欄位層級加密設定檔之後，請建立組態，以指定要求的內容類型，其中包括要加密的資料、用於加密的設定檔，以及其他指定您 CloudFront 要如何處理加密的選項。

例如，當 CloudFront 無法加密資料時，您可以在下列情況中指定是否 CloudFront要封鎖或轉寄要求至您的來源：

- 當請求的內容類型不在配置中時 — 如果您尚未將內容類型添加到配置中，則可以指定是否 CloudFront 應將具有該內容類型的請求轉發到來源，而不加密數據字段，或阻止請求並返回錯誤。

> **Note**
>
> 如果您將內容類型新增至設定，但尚未指定要搭配該類型使用的描述檔，請務 CloudFront 必將具有該內容類型的要求轉送至來源。

- 當查詢引數中提供的配置文件名稱未知時 — 當您指定具有配置文件名稱的**`fle-profile`**查詢參數時，您可以指定是否 CloudFront 應在不加密數據字段的情況下將請求發送到原點，還是阻止請求並返回錯誤。

在組態中，還可以指定一個在 URL 中做為查詢參數所提供的設定檔是否要覆寫了一個已對應到該查詢內容類型的設定檔。根據預設， CloudFront 會使用已對應至內容類型的描述檔 (如果您指定)。這可讓您擁有一個預設情況下使用的設定檔，但也會決定您所希望強制執行不同設定檔的特定請求。

因此，例如，您可以指定 (在您的組態中) **`SampleProfile`** 做為要使用的查詢參引數設定檔。然後`https://d1234.cloudfront.net`，您可以使用 URL `https://d1234.cloudfront.net?fle-profile=SampleProfile` 而不是 CloudFront 使**`SampleProfile`**用此請求，而不是為請求的內容類型設置的配置文件。

您最多可以為單一帳戶建立 10 個組態，然後將其中一個組態與的該帳戶的任何分佈的快取行為相關聯。

建立欄位層級加密的組態 (主控台)

1. 在 Field-level encryption (欄位層級加密) 頁面，選擇 Create configuration (建立組態)。

   注意：如果您尚未建立至少一個設定檔，則不會看到用於建立組態的選項。
2. 請填寫以下欄位指定要使用的設定檔。(有些欄位無法變更。)

   內容類型 (無法變更)

   內容類型設定為 `application/x-www-form-urlencoded`，無法變更。

   預設設定檔 ID (選用)

   在下拉式清單中選擇設定檔，此設定檔會對應到 Content type (內容類型) 欄位中的內容類型。

   內容格式 (無法變更)

   內容格式設定為 `URLencoded`，無法變更。
3. 如果您要變更下列選項的 CloudFront 預設行為，請選取適當的核取方塊。

   當請求的內容類型未作設定時，請轉發請求到原始來源

   如果您尚未指定用於請求內容類型的設定檔，而要允許請求轉傳到您的原始伺服器，請勾選此核取方塊。

   使用提供的查詢參數覆寫內容類型的設定檔

   如果您要允許查詢引數中所提供的設定檔，覆寫您針對內容類型指定的設定檔，請勾選此核取方塊。
4. 如果您選取核取方塊，以允許查詢參數來覆寫預設的設定檔，則必須完成組態的下列其他欄位。在這些查詢參數對應中，您最多可以建立五個，以便與查詢一起使用。

   查詢參數

   輸入要包含在 URL 中以用於 `fle-profile` 查詢參數的值。此值指示使CloudFront 用與此查詢引數相關聯的設定檔 ID (您在下一個欄位中指定)，以進行此查詢的欄位層級加密。

   最多可使用 128 個字元。該值不能包含空格，而且必須使用英數字元或以下字元：破折號 (-)、句點 (.)、底線 (_)、星號 (*)、加號 (+)、百分比 (%)。

設定檔 ID

在下拉式清單中選擇設定檔，您要將此設定檔與您針對 Query argument (查詢引數) 輸入的值建立關聯。

當查詢參數中指定的設定檔不存在時，將請求轉發到原始來源

如果未在中定義查詢引數中指定的描述檔，請選取此核取方塊，允許要求移至您的來源 CloudFront。

步驟 5：將組態新增到快取行為

若要使用欄位層級加密，請透過將組態 ID 新增為分佈的值，將組態連結到分佈的快取行為。

> **Important**
>
> 若要將欄位層級的加密設定連結至快取行為，必須將分佈設定為永遠使用 HTTPS，並接受來自瀏覽者的 HTTP POST 和 PUT 請求。也就是說，下列條件必須為真：
>
> - 快取行為的 Viewer Protocol 政策 (檢視器通訊協定政策) 必須設定為將 Redirect HTTP to HTTPS (HTTP 重新引導至 HTTPS) 或 HTTPS Only (僅 HTTPS)。(在 AWS CloudFormation 或 CloudFront API 中，`ViewerProtocolPolicy`必須設定為`redirect-to-https`或`https-only`。)
> - 快取行為的允許的 HTTP 方法必須設為 GET、HEAD、OPTIONS、PUT、POST、PATCH、DELETE。(在 AWS CloudFormation 或 CloudFront API 中，`AllowedMethods`必須設定為`GET、HEAD、OPTIONS、PUT、POST、PATCH、DELETE`。 這些可以以任何順序指定。)
> - 原始伺服器設定的 Origin Protocol 政策 (原始伺服器通訊協定政策) 必須設定為 Match Viewer (符合檢視器) 或 HTTPS Only (僅 HTTPS)。(在 AWS CloudFormation 或 CloudFront API 中，`OriginProtocolPolicy`必須設定為`match-viewer`或`https-only`。)

如需詳細資訊，請參閱 發佈設定參考。

在您的原始伺服器解密資料欄位

CloudFront 使用加密資料欄位。AWS Encryption SDK資料在整個應用程式堆疊中保持加密狀態，只能由具有解密憑證的應用程式存取。

加密後，加密文字是 base64 編碼的。當您的應用程式在原始伺服器解密文字時，必須先對加密文字解碼，然後使用 AWS 加密開發套件來解密資料。

以下程式碼範例說明應用程式如何在原始伺服器中解密資料。注意下列事項：

- 為了簡化範例，本範例從工作目錄中的檔案載入公有金鑰和私有金鑰 (以 DER 格式)。在實務上，您可以將私有金鑰存放在安全的離線位置，例如離線硬體安全模組中，並將公有金鑰分佈到您的開發團隊。
- CloudFront 在加密數據時使用特定信息，並且應在原始位置使用相同的參數集來對其進行解密。初始化時 CloudFront使用的參數 MasterKey 包括以下內容：
  - PROVIDER_NAME：當您建立欄位層級加密設定檔時指定了這個值。在這裡使用相同的值。
  - KEY_NAME：您在上傳公開金鑰時建立了一個名稱 CloudFront，然後在設定檔中指定金鑰名稱。在這裡使用相同的值。
  - 算法： CloudFront 用`RSA/ECB/OAEPWithSHA-256AndMGF1Padding`作加密算法，因此您必須使用相同的算法來解密數據。
- 如果您以加密文字做為輸入來執行以下範例程式時，則解密的資料將輸出到您的主控台。如需詳細資訊，請參閱 AWS 加密 SDK 中的 Java 範例程式碼。

範本程式碼

```
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.KeyFactory;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

import org.apache.commons.codec.binary.Base64;

import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoResult;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;

/**
 * Sample example of decrypting data that has been encrypted by CloudFront field-level
 encryption.
 */
```

```
public class DecryptExample {

    private static final String PRIVATE_KEY_FILENAME = "private_key.der";
    private static final String PUBLIC_KEY_FILENAME = "public_key.der";
    private static PublicKey publicKey;
    private static PrivateKey privateKey;

    // CloudFront uses the following values to encrypt data, and your origin must use
 same values to decrypt it.
    // In your own code, for PROVIDER_NAME, use the provider name that you specified
 when you created your field-level
    // encryption profile. This sample uses 'DEMO' for the value.
    private static final String PROVIDER_NAME = "DEMO";
    // In your own code, use the key name that you specified when you added your public
 key to CloudFront. This sample
    // uses 'DEMOKEY' for the key name.
    private static final String KEY_NAME = "DEMOKEY";
    // CloudFront uses this algorithm when encrypting data.
    private static final String ALGORITHM = "RSA/ECB/OAEPWithSHA-256AndMGF1Padding";

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

        final String dataToDecrypt = args[0];

        // This sample uses files to get public and private keys.
        // In practice, you should distribute the public key and save the private key
 in secure storage.
        populateKeyPair();

        System.out.println(decrypt(debase64(dataToDecrypt)));
    }

    private static String decrypt(final byte[] bytesToDecrypt) throws Exception {
        // You can decrypt the stream only by using the private key.

        // 1. Instantiate the SDK
        final AwsCrypto crypto = new AwsCrypto();

        // 2. Instantiate a JCE master key
        final JceMasterKey masterKey = JceMasterKey.getInstance(
                publicKey,
                privateKey,
                PROVIDER_NAME,
                KEY_NAME,
```

```
                ALGORITHM);

        // 3. Decrypt the data
        final CryptoResult <byte[], ? > result = crypto.decryptData(masterKey,
 bytesToDecrypt);
        return new String(result.getResult());
    }

    // Function to decode base64 cipher text.
    private static byte[] debase64(final String value) {
        return Base64.decodeBase64(value.getBytes());
    }

    private static void populateKeyPair() throws Exception {
        final byte[] PublicKeyBytes =
 Files.readAllBytes(Paths.get(PUBLIC_KEY_FILENAME));
        final byte[] privateKeyBytes =
 Files.readAllBytes(Paths.get(PRIVATE_KEY_FILENAME));
        publicKey = KeyFactory.getInstance("RSA").generatePublic(new
 X509EncodedKeySpec(PublicKeyBytes));
        privateKey = KeyFactory.getInstance("RSA").generatePrivate(new
 PKCS8EncodedKeySpec(privateKeyBytes));
    }
}
```

最佳化快取和可用性

本節說明如何設定和管理物件快取方式以提升效能並符合您的業務需求。

若要瞭解如何新增和移除您 CloudFront 要提供的內容，請參閱新增、移除或取代 CloudFront散佈的內容。

主題

- 快取如何與 CloudFront 邊緣位置搭配運作
- 增加直接從快取提供的要求比例 (CloudFront 快取命中率)
- 使用 Amazon CloudFront 起源 Shield
- 透過 CloudFront 原始容錯移轉將高可用性
- 管理內容保持在快取中達多久時間 (過期)
- 根據查詢字串參數快取內容
- 根據 Cookie 快取內容
- 根據請求標頭快取內容

快取如何與 CloudFront 邊緣位置搭配運作

使用的目的之一 CloudFront 是減少原始伺服器必須直接回應的要求數目。透過 CloudFront 快取，會從更接近使用者的 CloudFront邊緣位置提供更多物件。這樣可以降低原始伺服器上的負載並降低延遲。

CloudFront 可從 Edge 快取提供服務的要求越多， CloudFront必須轉送至原始伺服器才能取得物件的最新版本或唯一版本的檢視器要求就越少。若 CloudFront 要最佳化以盡可能少地向您的來源提出要求，請考慮使用 CloudFront Origin Shield 牌。如需詳細資訊，請參閱 使用 Amazon CloudFront 起源 Shield。

與所有要求相比，直接從 CloudFront 快取提供的要求比例稱為快取命中率。您可以在 CloudFront 主控台中檢視點擊、遺漏和錯誤的檢視器要求百分比。如需詳細資訊，請參閱 檢視 CloudFront 快取統計報告。

諸多因素皆會影響快取命中率。您可以依照中的指示調整 CloudFront 散發組態，以改善快取命中率增加直接從快取提供的要求比例 (CloudFront 快取命中率)。

增加直接從快取提供的要求比例 (CloudFront 快取命中率)

您可以增加直接從快取提供的檢視者要求比例，而不是前往原始伺服器 CloudFront 取得內容，藉此提升效能。這就是所謂的改善快取命中率。

下列各節說明如何提高您的快取命中率。

主題

指定物件 CloudFront 快取的時間長度

若要提高快取命中率，您可以設定原始伺服器，將 Cache-Control max-age 指令新增至物件，並為 `max-age` 指定最長的實際值。快取持續時間越短，就越頻繁地 CloudFront 將要求傳送到您的來源，以判斷物件是否已變更並取得最新版本。您可以以 `stale-while-revalidate` 和 `stale-if-error` 指令補充 `max-age`，以在特定情況下進一步改善快取命中率。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

使用 Origin Shield

CloudFront Origin Shield 可以幫助您提高 CloudFront 發行版的緩存命中率，因為它在您的來源前提供了一層額外的緩存。當您使用 Origin Shield 時，來自所有來源快取層的 CloudFront所有要求都來自單一位置。 CloudFront 可以使用來自 Origin Shield 的單一來源要求擷取每個物件，而 CloudFront快取的所有其他層 (節點位置和區域節點快取) 都可以從 Origin Shield 擷取物件。

如需詳細資訊，請參閱 使用 Amazon CloudFront 起源 Shield。

根據查詢字串參數快取

如果您設定 CloudFront 為根據查詢字串參數進行快取，您可以在執行下列動作時改善快取：

- 設定 CloudFront 為僅轉寄您的來源將會傳回唯一物件的查詢字串參數。

- 對所有相同參數的執行個體使用相同的大小寫 (大寫或小寫)。例如，如果一個請求包含`parameter1=A`而另一個請求包含`parameter1=a`，則當請求包含`parameter1=A`並在請求包含時將單獨的請求 CloudFront 轉發到您的來源。`parameter1=a` CloudFront 然後分別快取原始碼傳回的對應物件，即使物件相同也是如此。如果您只使用`A`或`a`，則將較少的請求 CloudFront 轉發到您的來源。
- 以相同順序列出參數。與大小寫不同的情況一樣，如果一個對象的請求包含查詢字符串，`parameter1=a&parameter2=b`而同一對象的另一個請求包含`parameter2=b&parameter1=a`，則將這兩個請求CloudFront 轉發到您的來源，並單獨緩存相應的對象，即使它們是相同的。如果您始終對參數使用相同的順序，則將較少的請求 CloudFront轉發到您的來源。

如需詳細資訊，請參閱 根據查詢字串參數快取內容。如果您要檢閱 CloudFront 轉寄至來源的查詢字串，請參閱 CloudFront 記錄檔案`cs-uri-query`欄中的值。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌)。

根據 Cookie 值快取

如果您設定 CloudFront 為根據 Cookie 值進行快取，則可以在執行下列動作時改善快取：

- 配置 CloudFront 為僅轉發指定的 cookie，而不是轉發所有 cookie。對於您配置轉發 CloudFront 到您的來源的 cookie，請 CloudFront 轉發 cookie 名稱和值的每個組合。然後它分別快取原始伺服器傳回的物件，即使它們都是相同的。

  例如，假設檢視者在每個請求中包含兩個 Cookie，每個 Cookie 都有三個可能的值，並且所有 Cookie 值的組合都是可能的。 CloudFront 針對每個物件，將最多六個不同的要求轉寄至您的來源。如果您的來源僅基於其中一個 cookie 返回對象的不同版本，那 CloudFront 麼將比必要的更多請求轉發到您的來源，並且不必要地緩存多個相同版本的對象。
- 為靜態和動態內容創建單獨的緩存行為，並配置 CloudFront為僅將動態內容的 cookie 轉發到您的來源。

  例如，假設您的發行版只有一個快取行為，而且您正在針對動態內容 (例如`.js`檔案) 和很少變更的`.css`檔案使用散佈。 CloudFront 根據 Cookie 值快取`.css`檔案的個別版本，因此每個 CloudFront 節點都會針對每個新 Cookie 值或 Cookie 值組合，將要求轉送至您的來源。

  如果您建立路徑模式所屬的快取行為，`*.css`且 CloudFront 不會根據 Cookie 值快取，則只會將邊緣位置為指定檔`.css`案收到的第一個要求，以及`.css`檔案到期後的第一個要求，將檔案要求 CloudFront 轉送至您的`.css`來源。

- 可能的話，每個使用者的 Cookie 值是唯一的 (例如使用者 ID) 時，為動態內容建立個別的快取行為，且動態內容根據較少數量的唯一值而有所不同。

如需詳細資訊，請參閱 根據 Cookie 快取內容。如果您想查看 CloudFront 轉發到您的來源的 Cookie，請查看 CloudFront 日誌文件`cs(Cookie)`列中的值。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌)。

根據請求標頭快取

如果您設定 CloudFront 為根據要求標頭快取，您可以在執行下列動作時改善快取：

- 配置 CloudFront 為僅基於指定的標頭轉發和緩存，而不是基於所有標頭轉發和緩存。對於您指定的標頭， CloudFront轉發標頭名稱和值的每個組合。然後它分別快取原始伺服器傳回的物件，即使它們都是相同的。

  > **Note**
  >
  > CloudFront 始終將以下主題中指定的標題轉發到您的來源：
  >
  > - 如何 CloudFront 處理並將請求轉送到您的 Amazon S3 原始伺服器 > CloudFront 移除或更新的 HTTP 要求標頭
  > - 如何 CloudFront 處理和轉寄請求至您的自訂原始伺服器 > HTTP 請求標頭和 CloudFront 行為 (自訂和 Amazon S3 來源)

  當您設定 CloudFront 為根據要求標頭快取時，您不會變更 CloudFront 轉寄的標頭，只有是否根據標頭值 CloudFront 快取物件。
- 請嘗試避免根據有大量唯一值的請求標頭來快取。

  例如，如果您希望根據用戶的設備提供不同大小的圖像，則不 CloudFront 要根據標題配置緩存，該`User-Agent`標題具有大量可能的值。而是根據 CloudFront 裝置 CloudFront 類型的標頭`CloudFront-Is-Desktop-Viewer`、`CloudFront-Is-Mobile-Viewer`、`CloudFront-Is-SmartTV-Viewer`和設定快取。`CloudFront-Is-Tablet-Viewer`此外，如果您是傳回適用於平板電腦和桌面的影像相同版本，則僅轉送 `CloudFront-Is-Tablet-Viewer` 標頭，而非 `CloudFront-Is-Desktop-Viewer` 標頭。

如需詳細資訊，請參閱 根據請求標頭快取內容。

不需要壓縮時，移除 Accept-Encoding 標頭

如果未啟用壓縮 (因為來源不支援壓縮、 CloudFront 不支援壓縮或內容無法壓縮)，您可以將發行版中的快取行為與設定如下的來源建立關聯，藉此提高快取命中率：Custom Origin Header

- Header name (標頭名稱)：`Accept-Encoding`
- Header value (標頭值)：(保留空白)

使用此配置時，請從緩存鍵中 CloudFront 刪除`Accept-Encoding`標頭，並且不會在原始請求中包含標頭。此配置適用於與該來源發行版一起 CloudFront 提供服務的所有內容。

使用 HTTP 來提供媒體內容

如需如何最佳化隨需視訊 (VOD) 及串流視訊內容的相關資訊，請參閱 視頻點播和實時流視頻 CloudFront。

使用 Amazon CloudFront 起源 Shield

CloudFront Origin Shield 是 CloudFront 快取基礎架構中的另一層，有助於將原始伺服器的負載降到最低、改善其可用性，並降低其營運成本。使用 CloudFront 起源 Shield 牌，您可以獲得以下好處：

緩衝區快取命中率

Origin Shield 可以幫助您提高 CloudFront 發行版本的緩存命中率，因為它在您的來源前提供了一層額外的緩存。當您使用 Origin Shield 時，來自所有快取層對您來源的 CloudFront所有要求都會透過 Origin Shield 進行，從而增加快取命中的可能性。 CloudFront可以使用來自 Origin Shield 的單一來源請求擷取每個物件至您的來源，而 CloudFront 快取的所有其他層 (邊緣位置和區域節點快取) 都可以從 Origin Shield 擷取物件。

減少原始伺服器負載

Origin Shield 可以進一步減少針對相同物件傳送到原始伺服器的同時請求數量。對於不在 Origin Shield 快取中的內容請求會與相同物件的其他請求整合，因此只需要一個請求，即可傳送到您的原始伺服器。在來源處理較少的請求可以在尖峰負載或意外流量尖峰期間保留原始伺服器的可用性，並且可以降低 just-in-time封裝、映像轉換和資料傳出 (DTO) 等成本。

更好的網路效能

當您在與原始伺服器延遲最低的 AWS 地區啟用 Origin Shield 時，您可以獲得更好的網路效能。對於某個 AWS 區域中的來源， CloudFront 網路流量會一直保留在您的來源的高輸送量 CloudFront

網路上。對於以外的來源 AWS， CloudFront 網路流量會一直保留在 CloudFront 網路上，直到 Origin Shield，因為它與您的原始伺服器具有低延遲連線。

使用 Origin Shield 會產生額外費用。如需詳細資訊，請參閱CloudFront 定價。

主題

Origin Shield 的使用案例

CloudFront 起源 Shield 對許多使用案例都有益，包括以下內容：

- 分佈在不同地理區域的檢視器
- 為即時串流或 on-the-fly影像處理提供 just-in-time 封裝的來源
- 具有容量或頻寬限制的內部部署原始伺服器
- 使用多個內容交付網路 (CDN) 的工作負載

Origin Shield 在其他情況下可能不太適合，例如代理至原始伺服器的動態內容、快取性低的內容，或很少請求的內容。

以下各節說明 Origin Shield 對於下列使用案例的好處。

使用案例

不同地理區域的檢視器

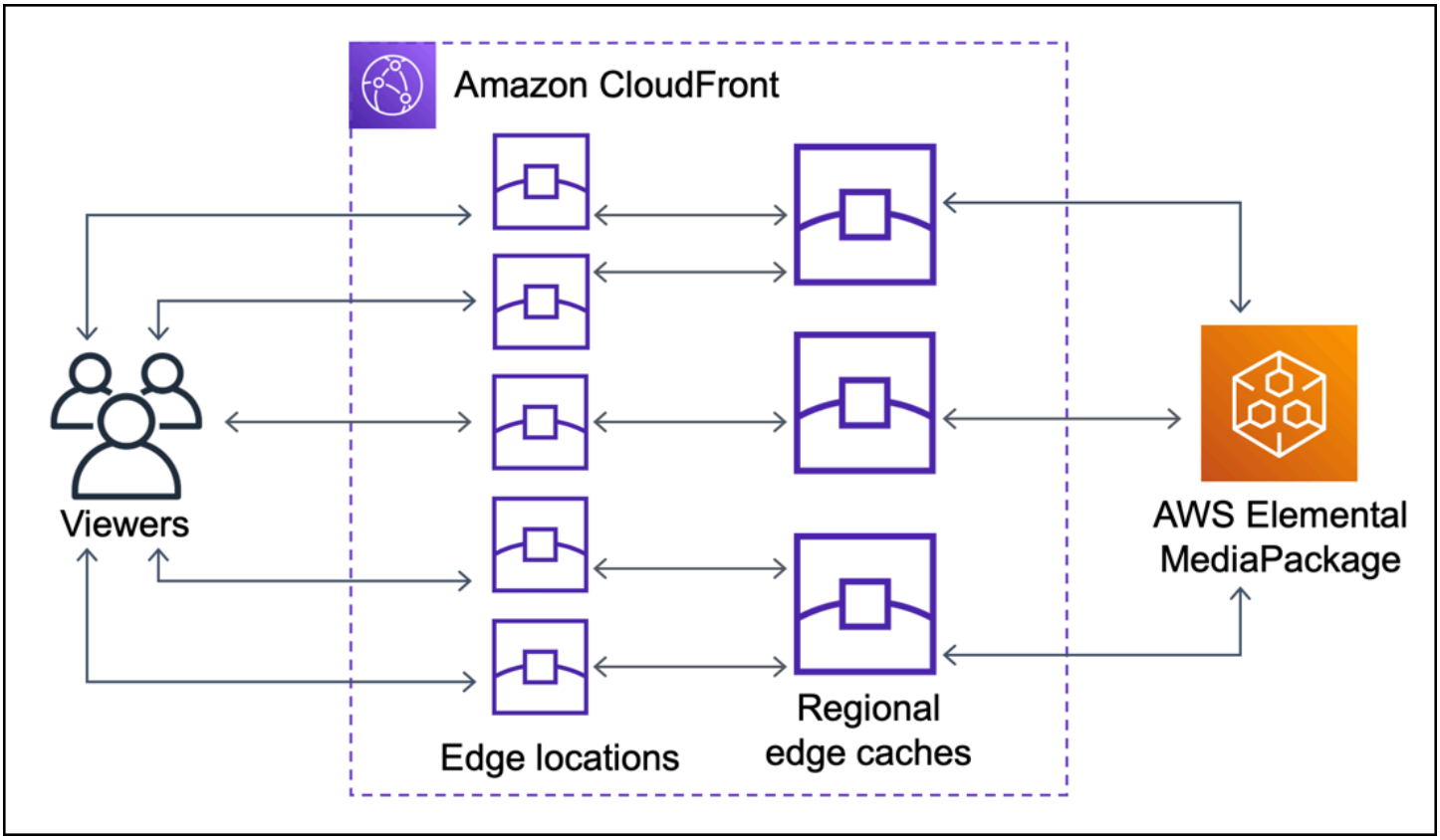
使用 Amazon CloudFront，您本質上可以減少來源的負載，因為 CloudFront 可以從緩存提供的請求不會轉到您的來源。除了全球 CloudFront節點網路之外，區域節點快取還可做為中間層快取層，以提

供快取命中，並整合附近地理區域中檢視者的原始要求。檢視器要求會先路由到附近的 CloudFront 節點，如果物件未快取在該位置，則會將要求傳送至地區邊緣快取。

當檢視器位於不同的地理區域時，可以透過不同的區域邊緣快取路由請求，每個請求都可以將相同內容的請求傳送至您的原始伺服器。但是，有了 Origin Shield，您可以在區域邊緣快取和您的原始伺服器之間獲得額外的一層快取。來自所有區域邊緣快取的所有請求都會經過 Origin Shield，進一步減少原始伺服器的負載。下圖說明此概念。在下圖中，原點是 AWS Elemental MediaPackage。

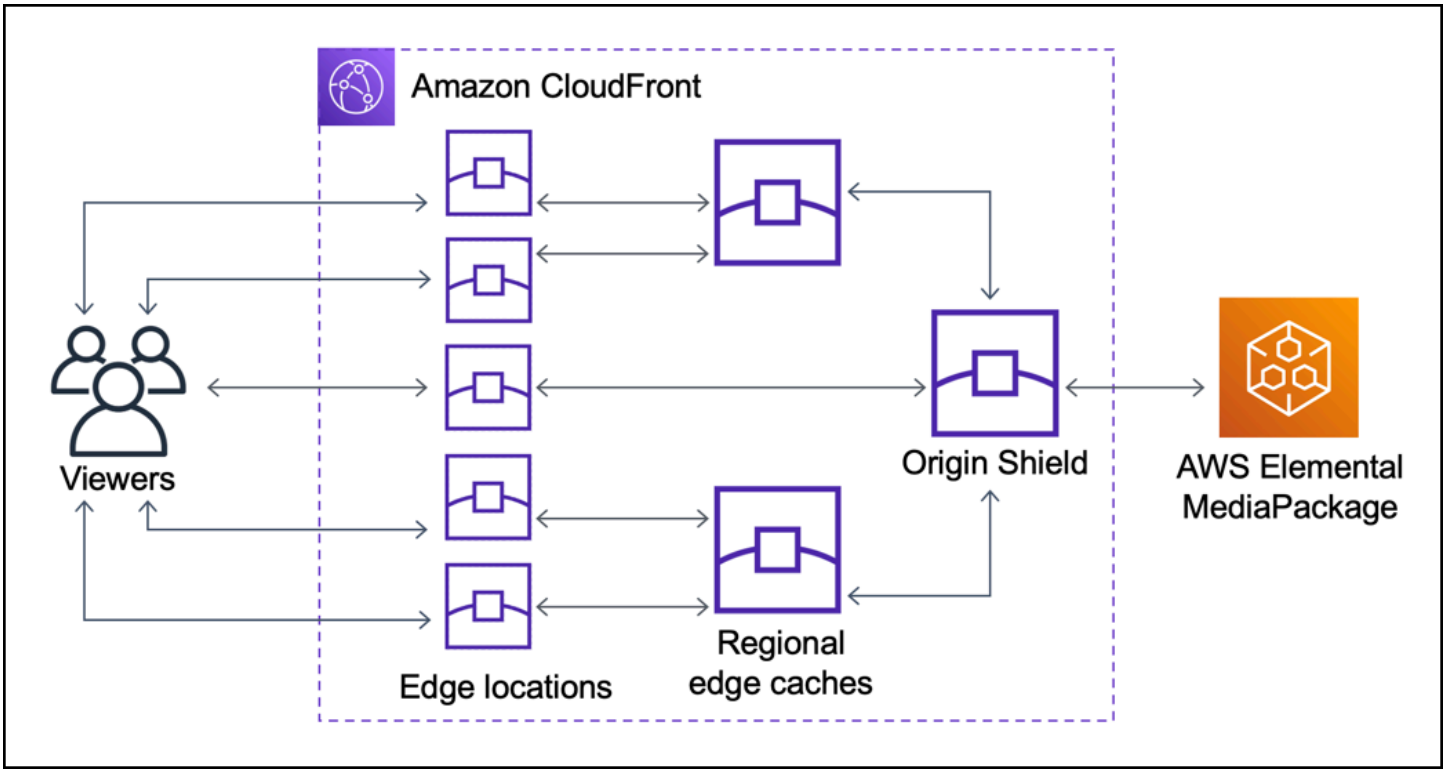
沒有 Origin Shield

如果沒有 Origin Shield，您的原始伺服器可能會收到相同內容的重複請求，如下圖所示。



具備 Origin Shield

使用 Origin Shield 有助於減少原始伺服器的負載，如下圖所示。



多個 CDN

若要提供即時視訊活動或熱門隨需內容，您可以使用多個內容交付網路 (CDN)。使用多個 CDN 可以提供某些好處，但這也表示您的原始伺服器可能會收到許多相同內容的重複請求，每個請求都來自不同的 CDN 或相同 CDN 中的不同位置。這些冗餘要求可能會對原始伺服器的可用性造成不利影響，或對 just-in-time 封裝或資料傳出 (DTO) 到網際網路等程序造成額外的作業成本。

當您將 Origin Shield 與使用您的 CloudFront 發行版作為其他 CDN 的來源結合使用時，您可以獲得以下好處：

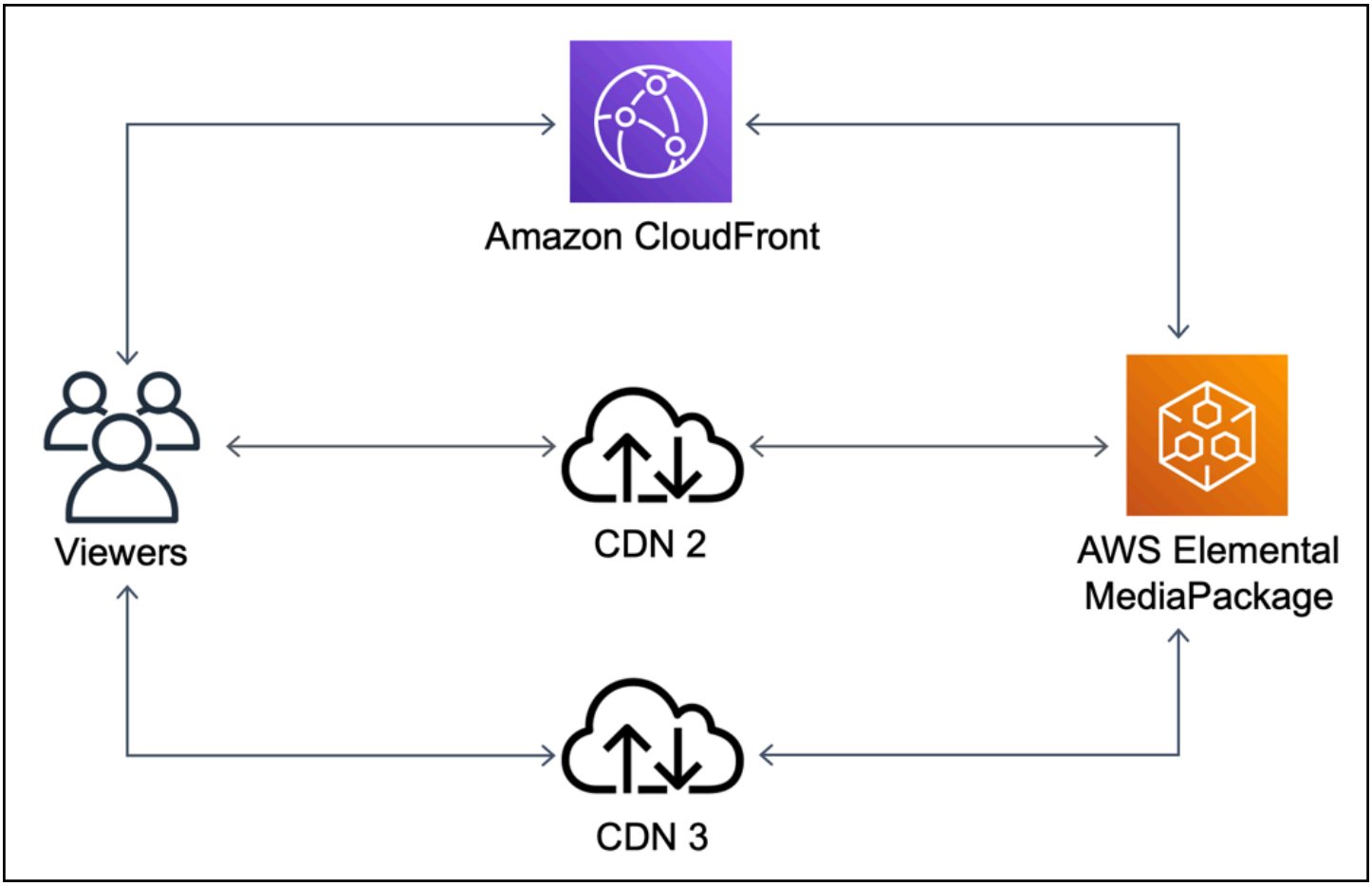
- 減少原始伺服器收到的冗餘請求，這有助於降低使用多個 CDN 的負面影響。
- 跨 CDN 的常見快取金鑰 ，以及集中式管理原始伺服器面相的功能。
- 改善的網路效能。來自其他 CDN 的網路流量會在附近的 CloudFront 邊緣位置終止，這可能會提供來自本機快取的點擊。如果要求的物件不在節點位置快取中，則對來源的要求會一直保留在 CloudFront 網路上，一直到 Origin Shield，這可為來源提供高輸送量和低延遲。如果請求的物件位於 Origin Shield 的快取中，可完全避免原始伺服器的請求。

> **Important**
>
> 如果您有興趣在多 CDN 架構中使用 Origin Shield，並且有折扣價格，請聯絡我們或您的 AWS 銷售代表以取得詳細資訊。可能需支付額外費用。

下圖顯示當您使用多個 CDN 提供熱門即時視訊活動時，此組態如何幫助將原始伺服器的負載降至最低。在下圖中，原點是 AWS Elemental MediaPackage。

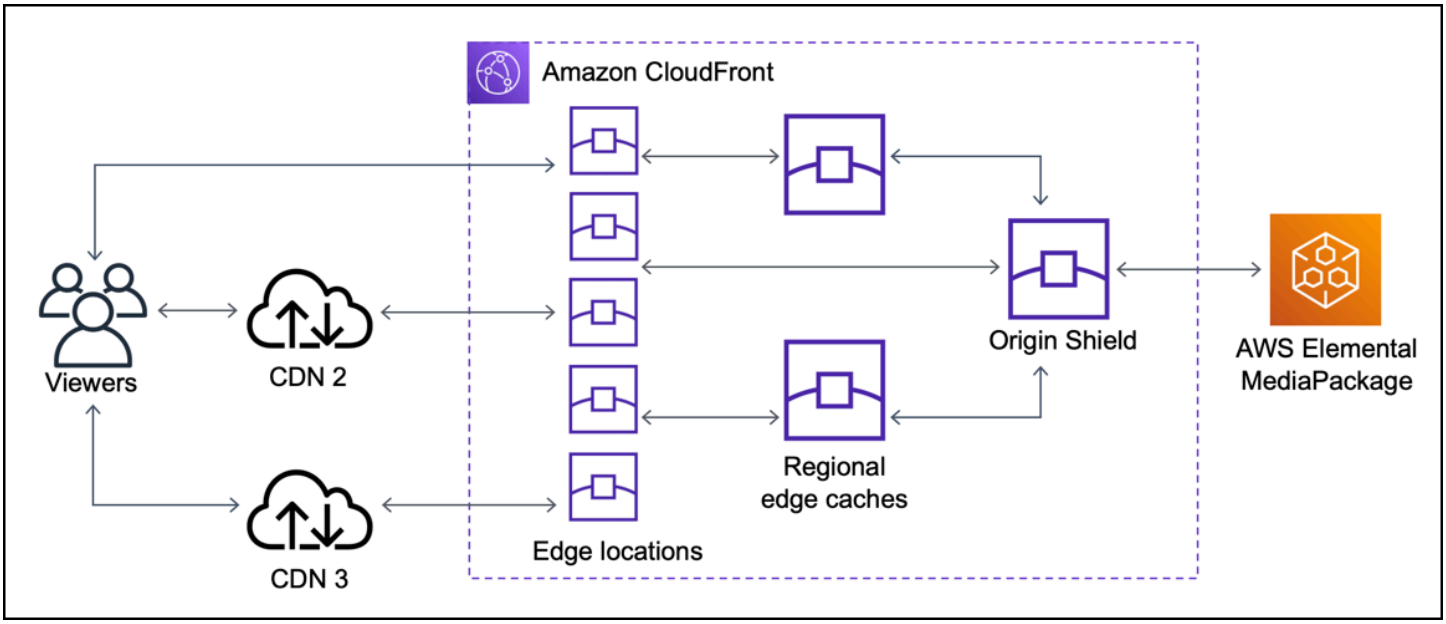
沒有 Origin Shield (多個 CDN)

沒有 Origin Shield，您的原始伺服器可能會收到相同內容的重複請求，每個請求都來自不同的 CDN，如下圖所示。



具備 Origin Shield (多個 CDN)

使用 Origin Shield CloudFront 作為其他 CDN 的來源，可協助減少來源的負載，如下圖所示。



選擇原點護 Shield 的 AWS 區域

Amazon 在具有區域邊緣快取的 AWS 區域 CloudFront 提 CloudFront 供原始 Shield。當您啟用原始護 Shield 時，您可以選擇原始護 Shield 的 AWS 區域。您應該為原始伺服器選擇最低延遲的 AWS 區域。您可以對 AWS 區域中的原點使用原點護 Shield，以及原點不在 AWS。

對於 AWS 區域中的原始伺服器

如果您的出發地位於某個 AWS 地區，請先確定您的出發地是否位於 CloudFront 提供 Origin 護 Shield 的地區。 CloudFront 在以下 AWS 區域提供原點護 Shield。

- 美國東部 (俄亥俄) – us-east-2
- 美國東部 (維吉尼亞北部) – us-east-1
- 美國西部 (奧勒岡) – us-west-2
- 亞太區域 (孟買) – ap-south-1
- 亞太區域 (首爾) – ap-northeast-2
- 亞太區域 (新加坡) – ap-southeast-1
- 亞太區域 (雪梨) – ap-southeast-2
- 亞太區域 (東京) – ap-northeast-1
- 歐洲 (法蘭克福) – eu-central-1
- 歐洲 (愛爾蘭) – eu-west-1
- 歐洲 (倫敦) – eu-west-2

- 南美洲 (聖保羅) – sa-east-1

如果您的出發地位於提 CloudFront 供起源護 Shield 的 AWS 地區

如果您的出發地位於 CloudFront 提供 Origin 護 Shield 的 AWS 地區（請參閱前面的清單），請在與您的出發地相同的地區啟用 Origin 護 Shield。

如果您的出發地不在提 CloudFront供起源護 Shield 的 AWS 地區

如果您的出發地不在提 CloudFront 供起源護 Shield 的 AWS 地區，請參閱下表以確定要在哪個區域中啟用起源護 Shield。

如果您的原始伺服器在...	Origin Shield 啟用於...
美國西部 (加利佛尼亞北部) – us-west-1	美國西部 (奧勒岡) – us-west-2
非洲 (開普敦) – af-south-1	歐洲 (愛爾蘭) – eu-west-1
亞太區域 (香港) – ap-east-1	亞太區域 (新加坡) – ap-southeast-1
加拿大 (中部) – ca-central-1	美國東部 (維吉尼亞北部) – us-east-1
歐洲 (米蘭) – eu-south-1	歐洲 (法蘭克福) – eu-central-1
歐洲 (巴黎) – eu-west-3	歐洲 (倫敦) – eu-west-2
歐洲 (斯德哥爾摩) – eu-north-1	歐洲 (倫敦) – eu-west-2
中東 (巴林) – me-south-1	亞太區域 (孟買) – ap-south-1

對於 外部的原始伺服器 AWS

您可以將 Origin Shield 與內部部署的原始伺服器或不在 AWS 區域中的原始伺服器搭配使用。在這種情況下，請在與原始伺服器延遲最低的 AWS 地區啟用 Origin Shield 牌。如果您不確定哪個 AWS 地區對您的來源延遲最低，可以使用以下建議來幫助您做出決定。

- 您可以參閱上表，根據原始伺服器的地理區域，了解哪個 AWS 區域對您的原始伺服器有最低延遲。
- 您可以在幾個地理位置靠近原始地點的不同 AWS 區域啟動 Amazon EC2 執行個體，並使用執行一些測試，ping以測量這些區域和您的來源之間的一般網路延遲。

啟用 Origin Shield

您可以啟用 Origin Shield 來改善快取命中率、降低原始伺服器的負載，並協助改善效能。若要啟用原點護 Shield，請變更 CloudFront 分佈中的原點設定。Origin Shield 是原始伺服器的屬性。針對您 CloudFront 發行版中的每個來源，您可以在任何 AWS 地區分別啟用 Origin Shield，為該來源提供最佳效能。

您可以使用或使 AWS CloudFormation用 CloudFront API 在 CloudFront 主控台中啟用原始 Shield 牌。

Console

若要為現有的原始伺服器啟用 Origin Shield (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇包含您要更新之原始伺服器的分佈。
3. 選擇 Origins and Origin Groups (原始伺服器和原始伺服器群組) 索引標籤。
4. 選擇要更新的原始伺服器，然後選擇 Edit (編輯)。
5. 對於 Enable Origin Shield (啟用 Origin Shield)，選擇 Yes (是)。
6. 在「原點護 Shield 區域」中，選擇您要啟用原始護 Shield 的 AWS 區域。如需選擇區域的說明，請參閱選擇原點護 Shield 的 AWS 區域。
7. 在頁面底部，選擇執行。

當您的分佈狀態為 Deployed (已部署)，Origin Shield 便已就緒。這需要幾分鐘的時間。

為新的原始伺服器啟用 Origin Shield (主控台)

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 若要在現有分佈中建立新的原始伺服器，請執行下列動作：

   1. 選擇您要在其中建立原始伺服器的分佈。
   2. 選擇 Create Origin (建立原始伺服器)，然後繼續步驟 3。

   若要在新分佈中建立新的原始伺服器，請執行下列動作：

1. 選擇 Create Distribution (建立分佈)。
2. 在 Web 區段中，選擇 Get Started (開始使用)。在 Origin Settings (原始伺服器設定) 區段中，從步驟 3 開始，完成以下步驟。

3. 對於 Enable Origin Shield (啟用 Origin Shield)，選擇 Yes (是)。
4. 在「原點護 Shield 區域」中，選擇您要啟用原始護 Shield 的 AWS 區域。如需選擇區域的說明，請參閱選擇原點護 Shield 的 AWS 區域。

   如果您要建立新分佈，請使用頁面上的其他設定，繼續設定您的分佈。如需更多詳細資訊，請參閱 發佈設定參考。
5. 請務必選擇 Create (建立) (適用於現有分佈中的新原始伺服器) 或 Create Distribution (建立分佈) (適用於新分佈中的新原始伺服器)。

當您的分佈狀態為 Deployed (已部署)，Origin Shield 便已就緒。這需要幾分鐘的時間。

AWS CloudFormation

若要啟用原始 Shield AWS CloudFormation，請在`OriginShield`AWS::CloudFront::Distribution資源中使用`Origin`屬性類型中的屬性。您可以將 `OriginShield` 屬性新增至現有的 `Origin`，或在建立新的 `Origin` 時，加入該屬性。

下列範例顯示 YAML 格式的語法，適用於在美國西部 (奧勒岡) 區域 (`OriginShield`) 中啟用 `us-west-2`。如需選擇區域的說明，請參閱the section called “選擇原點護 Shield 的 AWS 區域”。此範例僅顯示 `Origin` 屬性類型，而不是整個 `AWS::CloudFront::Distribution` 資源。

```
Origins:
- DomainName: 3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com
  Id: Example-EMP-3ae97e9482b0d011
  OriginShield:
    Enabled: true
    OriginShieldRegion: us-west-2
  CustomOriginConfig:
    OriginProtocolPolicy: match-viewer
    OriginSSLProtocols: TLSv1
```

如需詳細資訊，請參閱《AWS CloudFormation 使用指南》中的資源和屬性參考一節中的 AWS::CloudFront::Distribution Ori gin。

API

若要使用 AWS SDK 或 AWS Command Line Interface (AWS CLI) 透過 CloudFront API 啟用原始 Shield，請使用`OriginShield`類型。在 `OriginShield` 中，指定 `Origin` 中的 `DistributionConfig`，。如需有關`OriginShield`類型的資訊，請參閱 Amazon CloudFront API 參考中的下列資訊。

- OriginShield(類型)
- Origin (類型)
- DistributionConfig(類型)
- UpdateDistribution（操作）
- CreateDistribution（操作）

使用這些類型和操作的特定語法會根據軟體開發套件、CLI 或 API 用戶端而有所不同。如需詳細資訊，請參閱軟體開發套件、CLI 或用戶端的參考文件。

預估 Origin Shield 成本

您可以根據進入 Origin Shield 做為增量層的請求數量來累算 Origin Shield 的費用。

對於代理至原始伺服器的動態 (非可快取) 請求，Origin Shield 一律為增量層。動態請求使用 HTTP 方法`PUT`、`POST`、`PATCH`、和`DELETE`。

`GET`而且存留時間 (TTL) 設定小於 3600 秒的`HEAD`要求會被視為動態要求。此外，`GET`和已禁用緩存的`HEAD`請求也被視為動態請求。

若要針對動態請求預估 Origin Shield 的費用，請使用下列公式：

動態請求總數 x 每 10,000 個請求的 Origin Shield 費用 / 10,000

對於使用 HTTP 方法`GET`、和的非動態要求 `HEAD`、`OPTIONS`，來源 Shield 有時是增量層。當您啟用起源護 Shield 時，您可以選 AWS 區域 擇原點護 Shield。對於自然會移至與原始護 Shield 相同區域中的區域邊緣快取的要求，Origin Shield 並不是增量層。您不會為這些要求累積起源護 Shield 費用。如果要求移至與原始護 Shield 不同區域的區域邊緣快取，然後移至「原始護 Shield」，「起源護 Shield」是一個增量層。您會對這些請求累算 Origin Shield 費用。

若要針對可快取請求預估 Origin Shield 的費用，請使用下列公式：

可快取請求的總數 x (1 – 快取命中率) x 不同區域中從區域節點快取移至 Origin Shield 的請求百分比 x 每 10,000 個請求的 Origin Shield 費用 / 10,000

如需有關原始 Shield 牌每 10,000 個請求收費的詳細資訊，請參閱CloudFront 定價。

Origin Shield 高可用性

起源 Shield 牌利用 CloudFront 區域邊緣快取功能。這些邊緣快取中的每一個都建立在一個 AWS 區域中，至少使用三個可用區域，其中包含 auto-scaling 的 Amazon EC2 執行個體叢集。從位 CloudFront 置到 Origin Shield 的連線也會針對每個請求使用作用中的錯誤追蹤，以便在主要原點護 Shield 位置無法使用時，自動將請求路由至次要的 Origin Shield 位置。

起源 Shield 如何與其他功能互 CloudFront 動

以下各節將說明起源護 Shield 如何與其他 CloudFront功能互動。

起源 Shield 牌和 CloudFront 記錄

若要查看 Origin Shield 何時處理請求，您必須啟用下列其中一項：

- CloudFront 標準日誌（訪問日誌）。標準日誌是免費提供的。
- CloudFront 即時記錄檔。使用即時日誌會產生額外費用。請參閱 Amazon CloudFront 定價。

來自 Origin Shield 的快取命中會顯示`OriginShieldHit`在 CloudFront 記錄檔中的`x-edge-detailed-result-type`欄位中。起源 Shield 利用 Amazon CloudFront 的區域邊緣緩存。如果要求從 CloudFront 節點路由至作為 Origin Shield 的地區邊緣快取，則會在記錄檔`Hit`中報告為一個，而不是`OriginShieldHit`。

Origin Shield 和原始伺服器群組

原點 Shield 牌與CloudFront 原始群組相容。由於 Origin Shield 是原始伺服器的屬性，因此，即使原始伺服器是原始伺服器群組的一部分，請求一律會針對每個原始伺服器透過 Origin Shield 來傳輸。針對指定的要求，會透過主要來源的 Origin Shield 將請求 CloudFront 路由至原始群組中的主要來源。如果該要求失敗 (根據原始群組容錯移轉準則)，請透過次要原點的 Origin Shield 將要求 CloudFront 路由至次要原點。

Origin Shield 和 Lambda@Edge

Origin Shield 不會影響 Lambda@Edge 函數的功能，但它會影響這些函數執行所在的 AWS 區域。

當您將原始 Shield 牌與 Lambda @Edge 搭配使用時，面向原點的觸發程序 (原始請求和來源回應) 會在啟用原始護 Shield 的 AWS 區域中執行。如果主要的起點護 Shield 位置無法使用，並且將請求 CloudFront 路由到次要的起點護 Shield 位置，Lambda @Edge 面向起點的觸發器也會轉換為使用次要原點護 Shield 位置。

檢視器面向觸發程序不會受到影響。

透過 CloudFront 原始容錯移轉將高可用性

您可以針對需要高可用性的案例設定來源容錯移轉。 CloudFront 如要開始使用，您可以使用兩個原始伺服器建立原始伺服器群組：主要及次要。如果主要原點無法使用，或傳回指示失敗的特定 HTTP 回應狀態碼，則 CloudFront 會自動切換至次要來源。

若要設定原始伺服器容錯移轉，您必須至少分佈兩部原始伺服器。接著，您可以為您的分佈建立原始伺服器群組，其中包含兩個原始伺服器，並將其中一個設為主要原始伺服器。最後，您可以建立或更新快取行為，以使用原始伺服器群組。

如要查看設定原始伺服器群組和設定特定原始伺服器容錯移轉選項的步驟，請參閱 建立原始伺服器群組。

設定快取行為的來源容錯移轉之後， CloudFront 請針對檢視器要求執行下列動作：

- 當有緩存命中時， CloudFront 返回請求的對象。
- 發生快取未命中時，會將要求 CloudFront 路由至原始群組中的主要原點。
- 當主要來源傳回未設定用於容錯移轉的狀態碼 (例如 HTTP 2xx 或 3xx 狀態碼) 時，會將要求的物件 CloudFront 提供給檢視者。
- 發生下列任何一種情況時：
  - 主要原始伺服器會傳回您為容錯移轉設定的 HTTP 狀態碼
  - CloudFront 無法連接到主要原點
  - 主要原始伺服器的回應花費時間太長 (逾時)

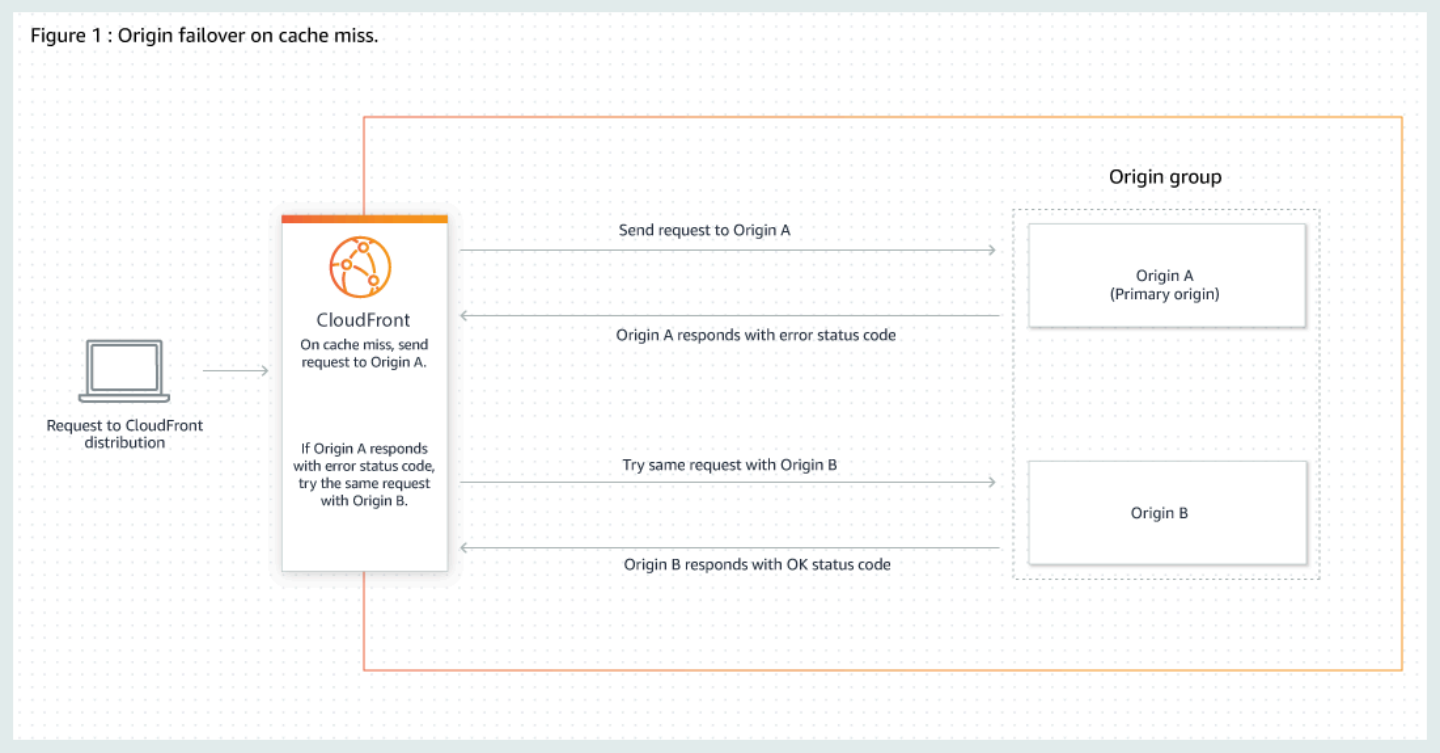
  然後將請求 CloudFront 路由至原始群組中的次要原點。

> **Note**
>
> 對於某些使用案例 (例如串流視訊內容)，您可能會想 CloudFront 要快速容錯移轉至次要來源。若要調整容 CloudFront 錯移轉至次要原點的速度，請參閱控制原始伺服器逾時和嘗試次數。

CloudFront 將所有傳入的要求路由至主要來源，即使先前的要求失敗移轉至次要來源也是如此。CloudFront 只有在對主要來源的要求失敗後，才會傳送要求至次要來源。

CloudFront 只有當檢視器要求的 HTTP 方法為GET、HEAD或時，才容錯移轉至次要原點OPTIONS。CloudFront 當檢視器傳送不同的 HTTP 方法 (例如POST，等等) 時PUT，不會容錯移轉。

下圖說明原始伺服器容錯移轉的運作方式。



主題

- 建立原始伺服器群組
- 控制原始伺服器逾時和嘗試次數
- 使用原始伺服器容錯移轉與 Lambda@Edge 函數搭配
- 搭配原始伺服器容錯移轉使用自訂錯誤頁面

建立原始伺服器群組

建立來源群組

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇您要為原始伺服器群組建立的分佈。

3. 選擇 Origins (原始伺服器) 索引標籤。
4. 請確定分佈有多個原始伺服器。如果沒有多個原始伺服器，請新增第二個原始伺服器。
5. 在 Origins (原始伺服器) 索引標籤上的 Origin groups (原始伺服器群組) 窗格中，選擇 Create Origin Group (建立原始伺服器群組)。
6. 選擇原始伺服器群組的原始伺服器。新增原始伺服器後，使用箭號來設定優先順序，即哪一個原始伺服器為主要原始伺服器，哪一個為輔助。
7. 輸入原始伺服器群組的名稱。
8. 選擇要用來做為容錯移轉條件的 HTTP 代碼。您可以選擇以下狀態碼的任意組合：400、403、404、416、500、502、503 或 504。當 CloudFront 收到包含您指定的其中一個狀態碼的回應時，會容錯移轉至次要原點。

   > **Note**
   >
   > CloudFront 只有當檢視器要求的 HTTP 方法為GET、HEAD或時，才容錯移轉至次要原點OPTIONS。 CloudFront 當檢視器傳送不同的 HTTP 方法 (例如POST，等等) 時PUT，不會容錯移轉。

9. 選擇 Create origin group (建立原始伺服器群組)。

請務必將您的原始群組指派為發行版快取行為的來源。如需詳細資訊，請參閱 名稱。

控制原始伺服器逾時和嘗試次數

預設情況下， CloudFront 會嘗試連線至原始群組中的主要原點，長達 30 秒 (3 次連線嘗試，每次嘗試 10 秒)，然後再容錯移轉至次要原點。對於某些使用案例 (例如串流視訊內容)，您可能會想 CloudFront 要更快速地容錯移轉至次要來源。您可以調整以下設定，以影響容 CloudFront 錯移轉至次要原點的速度。如果來源是次要原點或不屬於原始群組的來源，則這些設定會影響將 HTTP 504 CloudFront 回應傳回給檢視器的速度。

若要更快地容錯移轉，請指定較短的連線逾時、較少的連線嘗試次數或兩者。對於自訂原始伺服器 (包括使用靜態網站託管所設定的 Amazon S3 儲存貯體)，您也可以調整原始伺服器回應逾時。

原始伺服器連線逾時

原始連線逾時設定會影響嘗試建立與原始連線時的 CloudFront 等待時間長度。依預設，會 CloudFront 等待 10 秒來建立連線，但您可以指定 1—10 秒 (含)。如需詳細資訊，請參閱 連線逾時。。

原始伺服器連線嘗試次數

原始連線嘗試次數設定會影響 CloudFront 嘗試連線至原點的次數。依預設， CloudFront 嘗試連線 3 次，但您可以指定 1—3 (含)。如需詳細資訊，請參閱 連線嘗試。

對於自訂來源 (包括使用靜態網站託管設定的 Amazon S3 儲存貯體)，此設定也會影響在原始回應逾時的情況下 CloudFront 嘗試從來源取得回應的次數。

原始伺服器回應逾時

> **Note**
>
> 這僅適用於自訂原始伺服器。

來源響應超時設置會影響從源接收響應（或接收完整響應）的 CloudFront 等待時間。依預設，會 CloudFront 等待 30 秒，但您可以指定 1—60 秒 (含)。如需詳細資訊，請參閱 回應逾時 (僅限自訂原始伺服器)。

如何變更這些設定

在CloudFront 主控台中變更這些設定

- 對於新原始伺服器或新的分佈，您可以在建立資源時指定這些數值。
- 對於現有分佈中的現有原始伺服器，您可以在編輯原始伺服器時指定這些數值。

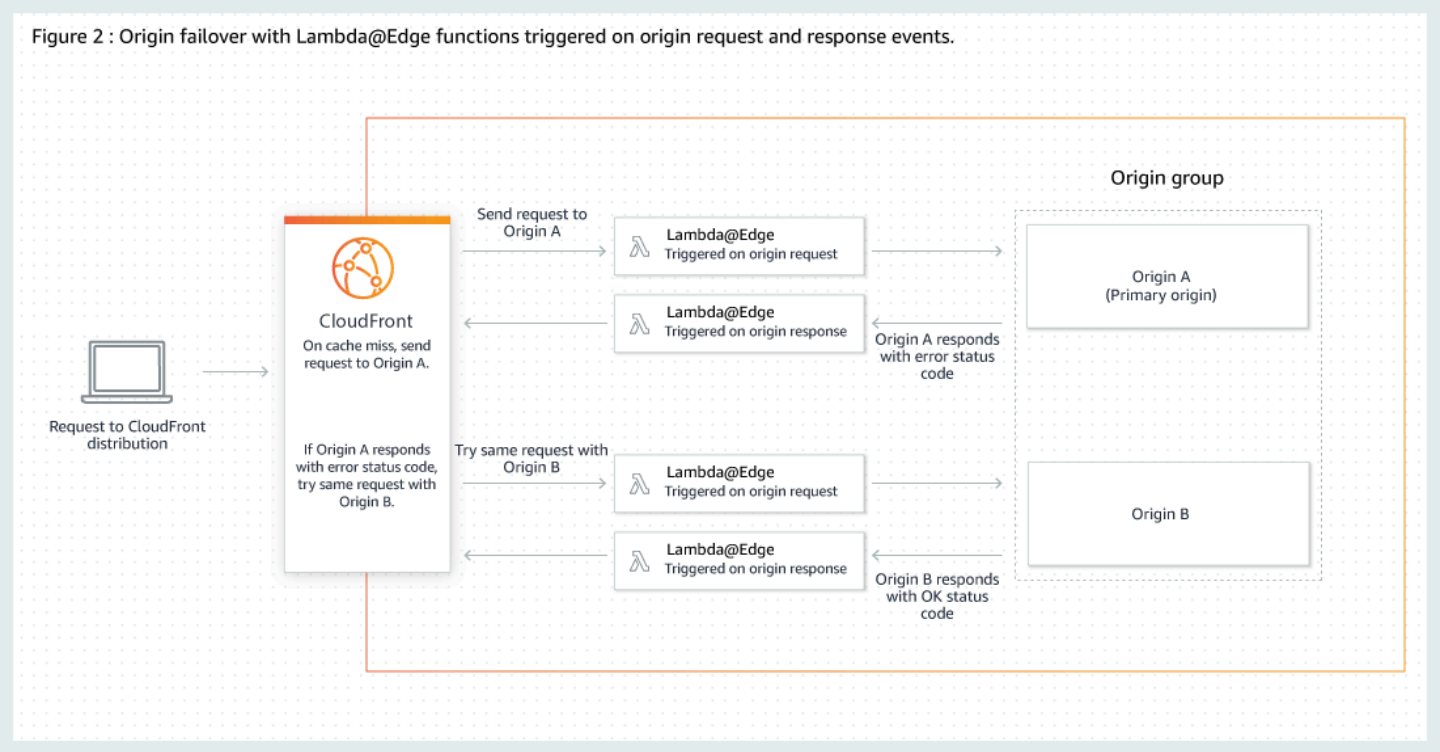
如需更多詳細資訊，請參閱 發佈設定參考。

使用原始伺服器容錯移轉與 Lambda@Edge 函數搭配

您可以將 Lambda @Edge 函數與您使用來源群組設定的發行版搭 CloudFront 配使用。如要使用 Lambda 函數，請在建立快取行為時，於原始伺服器群組的原始伺服器請求或原始伺服器回應觸發條件中進行指定。當您搭配原始伺服器群組使用 Lambda@Edge 函數時，可能會針對單一瀏覽者請求觸發函數兩次。例如，考量以下情境：

1. 您可以使用原始伺服器請求觸發條件建立 Lambda@Edge 函數。
2. CloudFront 傳送要求至主要來源 (快取未命中時) 時，Lambda 函數會觸發一次。
3. 主要原始伺服器會回應為容錯移轉設定的 HTTP 狀態碼。
4. 將相同的請求發 CloudFront 送到次要原點時，Lambda 函數會再次觸發。

下圖說明當您在原始伺服器請求或回應觸發中包括 Lambda@Edge 函數時，原始伺服器容錯移轉的運作方式。



如需使用 Lambda@Edge 觸發的詳細資訊，請參閱the section called “為 Lambda @Edge 函數新增觸發程序”。

如需管理 DNS 容錯移轉的詳細資訊，請參閱 Amazon Route 53 開發人員指南中的設定 DNS 容錯移轉。

搭配原始伺服器容錯移轉使用自訂錯誤頁面

您可以使用自訂錯誤頁面與原始伺服器群組搭配，其方式與您將它們與未針對原始伺服器容錯移轉設定之原始伺服器搭配使用的方式類似。

當您使用來源容錯移轉時，您可 CloudFront 以設定為傳回主要或次要來源 (或兩者) 的自訂錯誤頁面：

- 傳回主要來源的自訂錯誤頁面 — 如果主要來源傳回未設定用於容錯移轉的 HTTP 狀態碼，則會將自訂錯誤頁面 CloudFront 傳回給檢視者。
- 傳回次要來源的自訂錯誤頁面 — 如果 CloudFront 收到次要來源的失敗狀態碼，則會 CloudFront 傳回自訂錯誤頁面。

如需搭配使用自訂錯誤頁的詳細資訊 CloudFront，請參閱產生自訂錯誤回應。

管理內容保持在快取中達多久時間 (過期)

您可以控制檔案保留在 CloudFront 快取中的時間長度，然後再將另一個要求 CloudFront 轉寄至您的來源。降低持續時間允許您提供動態內容。增加持續時間表示您的使用者取得更好的效能，因為檔案更有可能是直接透過節點快取提供。較長的持續時間也能減少的原始伺服器的負載。

一般而言，會從邊緣位置 CloudFront 提供檔案，直到您指定傳遞的快取持續時間為止 — 也就是說，直到檔案過期為止。到期後，下次邊緣位置取得檔案的要求時，會將要求 CloudFront 轉送至原始位置，以確認快取是否包含檔案的最新版本。來自原始伺服器的回應取決於檔案是否已變更：

- 如果 CloudFront 快取已經有最新版本，來源會傳回狀態碼`304 Not Modified`。
- 如果 CloudFront 快取沒有最新版本，來源會傳回狀態碼`200 OK`和檔案的最新版本。

如果不經常要求邊緣位置中的檔案， CloudFront 可能會收回檔案 (在檔案到期日之前移除檔案)，以便為最近要求的檔案騰出空間。

根據預設，每個檔案會自動在 24 小時候過期，但您可以透過兩種方式變更預設行為：

- 若要變更符合相同路徑模式的所有檔案的快取持續時間，您可以針對快取行為變更「最小 TTL」、「最大 TTL」和「預 CloudFront 設 TTL」的設定。如需個別設定的相關資訊，請參閱the section called “分佈設定”中的最小 TTL、最大 TTL，以及預設 TTL。
- 若要變更個別檔案的快取持續期間，您可以設定原始伺服器來使用 `max-age` 或 `s-maxage` 指示詞將 `Cache-Control` 標頭，或是 `Expires` 標頭欄位新增到檔案。如需詳細資訊，請參閱 使用標頭來控制個別物件的快取持續時間。

如需有關 Minimum TTL (最短 TTL)、Default TTL (預設 TTL) 及 Maximum TTL (最長 TTL) 與 `max-age` 和 `s-maxage` 命令及 `Expires` 標頭欄位互動方式的詳細資訊，請參閱 the section called “指定 CloudFront 快取物件的時間長度”。

您還可以通過將另一個請求轉發到您的來源來控制錯誤（例如，`404 Not Found`）在再次 CloudFront 嘗試獲取請求的對象之前停留在CloudFront 緩存中的時間長度。如需詳細資訊，請參閱 the section called “如何從您的來源 CloudFront 處理和緩存 HTTP 4xx 和 5xx 狀態碼”。

主題

- 使用標頭來控制個別物件的快取持續時間
- 提供過時 (過期) 內容
- 指定 CloudFront 快取物件的時間長度

- 使用 Amazon S3 主控台新增標頭到物件

使用標頭來控制個別物件的快取持續時間

您可以使用 Cache-Control 和 Expires 標頭來控制物件在快取中保持多久的時間。Minimum TTL (最短 TTL)、Default TTL (預設 TTL) 及 Maximum TTL (最長 TTL) 的設定也會影響快取持續時間，但以下是標頭會如何影響快取持續時間的概觀：

- 此指Cache-Control max-age令可讓您指定物件在從原始伺服器再次取 CloudFront 得物件之前，要將物件保留在快取中的時間長度 (以秒為單位)。 CloudFront 支持的最短到期時間為 0 秒。最長值為 100 年。請在下列格式中指定值：

  ```
  Cache-Control: max-age=#
  ```

  例如，下面的指令告訴 CloudFront 保持在緩存中的關聯對象 3600 秒（一小時）：

  ```
  Cache-Control: max-age=3600
  ```

  如果您希望物件停留在 CloudFront Edge 快取中的持續時間與停留在瀏覽器快取中的持續時間不同，可以一起使用Cache-Control max-age和Cache-Control s-maxage指令。如需詳細資訊，請參閱 指定 CloudFront 快取物件的時間長度。
- Expires 標頭欄位可讓您使用 RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1 Section 3.3.1, Full Date 中指定的格式來指定過期日期和時間，例如：

  ```
  Sat, 27 Jun 2015 23:59:59 GMT
  ```

我們建議您使用 Cache-Control max-age 指令，而不是 Expires 標頭欄位來控制物件快取。如果您同時為Cache-Control max-age和指定值Expires，則僅CloudFront 使用的值Cache-Control max-age。

如需詳細資訊，請參閱 指定 CloudFront 快取物件的時間長度。

您無法在檢視器的GET要求中使用 HTTP Cache-Control 或Pragma標頭欄位 CloudFront 來強制返回物件的原始伺服器。 CloudFront 會忽略檢視器要求中的這些標頭欄位。

如需有關 Cache-Control 與 Expires 標頭欄位的詳細資訊，請參閱 RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1 中的以下各節：

- Section 14.9 Cache Control (14.9 節快取控制)

- Section 14.21 Expires (14.21 節過期)

提供過時 (過期) 內容

CloudFront 支持`Stale-While-Revalidate`和`Stale-If-Error`緩存控制指令。

- 該`stale-while-revalidate`指令 CloudFront 允許從緩存中提供過時的內容，同時從原始異步獲取新版本。這可改善延遲時間，因為使用者會立即從 CloudFront邊緣位置接收回應，而無需等待背景擷取，而且會在背景中載入新內容以供 future 要求使用。

  在下列範例中， CloudFront 快取回應一小時 (`max-age=3600`)。如果在此期間之後發出請求，則會在同時向來源發送請求以重新驗證和重新整理緩存內容的同時CloudFront 提供過時內容的內容。過時內容最多可提供 10 分鐘 (`stale-while-revalidate=600`)，同時內容會被重新驗證。

  ```
  Cache-Control: max-age=3600, stale-while-revalidate=600
  ```

- 如果來源無法訪問，或返回介於 500 和 600 之間的錯誤代碼，該`stale-if-error`指令允許 CloudFront 從緩存中提供過時的內容。這可確保檢視者即使在原始伺服器中斷期間也能存取內容。

  在下列範例中， CloudFront 快取回應一小時 (`max-age=3600`)。如果來源已關閉或在此期間之後傳回錯誤，則會 CloudFront 持續為過時內容提供長達 24 小時 (`stale-if-error=86400`)。

  ```
  Cache-Control: max-age=3600, stale-if-error=86400
  ```

  > **Note**
  >
  > 同時設定`stale-if-error`和自訂錯誤回應時，如果在指定的`stale-if-error`持續時間內發生錯誤， CloudFront 首先會嘗試提供過時內容。如果無法使用過時內容，或內容超出`stale-if-error`持續時間，則會 CloudFront 提供針對對應錯誤狀態碼設定的自訂錯誤回應。

兩者一起使用

`stale-while-revalidate` 與 `stale-if-error` 是獨立的快取控制指令，可以一起使用以減少延遲，並為原始伺服器新增緩衝區以便回應或復原。

在下列範例中， CloudFront 快取回應一小時 (`max-age=3600`)。如果在此期間之後提出要求，則在重新驗證內容時，最多可CloudFront 提供 10 分鐘的過時內容 (`stale-while-`

`revalidate=600`)。如果原始伺服器在 CloudFront 嘗試重新驗證內容時傳回錯誤，則會 CloudFront 持續提供過時內容長達 24 小時 () `stale-if-error=86400`。

```
Cache-Control: max-age=3600, stale-while-revalidate=600, stale-if-error=86400
```

> **Tip**
>
> 快取是效能和更新狀態之間的平衡。使用類似 `stale-while-revalidate` 與 `stale-if-error` 指令可增強效能和使用者體驗，但請確保組態與您希望內容的更新狀態保持一致。過時內容指令最適合需要重新整理內容但不需要最新版本的使用案例。此外，如果您的內容沒有變更或很少變更，`stale-while-revalidate` 可能會新增不必要的網路要求。反之，請考慮設定較長的快取持續時間。

指定 CloudFront 快取物件的時間長度

若要控制在快取中 CloudFront 保留物件的時間長度，然後再傳送另一個要求給原始伺服器，您可以：

- 設定 CloudFront 發行版快取行為中的最小值、最大值和預設 TTL 值。您可以在連接至快取行為的快取政策 (建議使用) 或舊版快取設定中設定這些值。
- 在來自原始伺服器的回應中包含 `Cache-Control` 或 `Expires` 標頭。這些標頭還有助於確定瀏覽器在向瀏覽器發送另一個請求之前在瀏覽器緩存中保留對象的時間長度 CloudFront。

下表解釋了從原始伺服器傳送的 `Cache-Control` 和 `Expires` 標頭如何與快取行為中的 TTL 設定一起運作，以影響快取。

原始標頭	最短 TTL = 0	最短 TTL > 0
原始伺服器將 **`Cache-Control: max-age`** 指示詞新增至物件	CloudFront 快取 CloudFront 快取`Cache-Control: max-age` 指令值中較小值的物件，或 CloudFront 最大 TTL 值的值。 瀏覽器快取	CloudFront 快取 CloudFront 快取取決於最 CloudFront 小 TTL 和最大 TTL 和指令的值：`Cache-Control max-age` •

原始標頭	最短 TTL = 0	最短 TTL > 0
	瀏覽器會快取 `Cache-Control: max-age` 指令值的物件。	如果最小 TTL < `max-age` < 最大 TTL，則會 CloudFront 快取指示詞值的物件。`Cache-Control: max-age` • 如果 `max-age` < 下限 TTL，則會 CloudFront快取物件的最 CloudFront 小 TTL 值。 • 如果 `max-age` > 最大 TTL，則會 CloudFront快取 CloudFront 最大 TTL 值的物件。 瀏覽器快取 瀏覽器會快取 `Cache-Control: max-age` 指令值的物件。
原始伺服器不會將 **`Cache-Control: max-age`** 指示詞新增至物件	CloudFront 快取 CloudFront 快取 CloudFront 預設 TTL 值的物件。 瀏覽器快取 取決於瀏覽器。	CloudFront 快取 CloudFront 快取物件時，CloudFront 最小 TTL 或預設 TTL 值中較大的值。 瀏覽器快取 取決於瀏覽器。

原始標頭	最短 TTL = 0	最短 TTL > 0
原始伺服器將 **Cache-Control: max-age** 與 **Cache-Control: s-maxage** 指示詞新增至物件	CloudFront 快取 CloudFront 快取Cache-Control: s-maxage 指令值中較小值的物件，或 CloudFront 最大 TTL 值的值。 瀏覽器快取 瀏覽器會快取 Cache-Control max-age 指令值的物件。	CloudFront 快取 CloudFront 快取取決於最 CloudFront 小 TTL 和最大 TTL 和指令的值：Cache-Control: s-maxage • 如果最小 TTL < s-maxage < 最大 TTL，則會 CloudFront 快取指示詞值的物件。Cache-Control: s-maxage • 如果 s-maxage < 下限 TTL，則會 CloudFront快取物件的最 CloudFront 小 TTL 值。 • 如果 s-maxage > 最大 TTL，則會 CloudFront快取 CloudFront 最大 TTL 值的物件。 瀏覽器快取 瀏覽器會快取 Cache-Control: max-age 指令值的物件。

原始標頭	最短 TTL = 0	最短 TTL > 0
原始伺服器將 **Expires** 標頭新增至物件	CloudFront 快取 CloudFront 快取物件，直到Expires標頭中的日期或 TTL 上 CloudFront 限值 (以較早者為準)。 瀏覽器快取 瀏覽器會快取物件直到 Expires 標頭中的日期。	CloudFront 快取 CloudFront 快取取決於最 CloudFront 小 TTL 和最大 TTL 和標Expires頭的值： • 如果最小 TTL < Expires < 最大 TTL，則會 CloudFront 快取物件，直到標頭中的日期和時間為止。Expires • 如果 Expires < 下限 TTL，則會 CloudFront快取物件的最 CloudFront 小 TTL 值。 • 如果 Expires > 最大 TTL，則會 CloudFront快取 CloudFront 最大 TTL 值的物件。 瀏覽器快取 瀏覽器會快取物件直到 Expires 標頭中的日期與時間。

原始標頭	最短 TTL = 0	最短 TTL > 0
原始伺服器將 **Cache-Control: no-cache** 、**no-store** 和/或 **private** 指示詞新增至物件	CloudFront 和瀏覽器尊重標題。	CloudFront 快取 CloudFront 快取物件的 CloudFront 最小 TTL 值。請參閱此資料表下方的警告。 瀏覽器快取 瀏覽器遵守標頭。

> **Warning**
>
> 如果您的最小 TTL 大於 0，即 CloudFront 使來源標頭中有、和/或`private`指令，也會`Cache-Control: no-cache`使用快取原則的最小 TTL。`no-store`
> 如果原點可訪問，則從原點 CloudFront 獲取對象並將其返回給查看器。
> 如果原點無法存取，且最小或最大 TTL 值大於 0， CloudFront 將會提供原點之前從原點取得的物件。
> 為了避免這種行為，請將 `Cache-Control: stale-if-error=0` 指示詞與從原始伺服器傳回的物件一同包含。如果來源無法訪問，這會導致 CloudFront 返回錯誤以響應 future 的請求，而不是返回之前從原點獲得的對象。

有關如何使用 CloudFront 控制台更改發行版設置的信息，請參閱更新分佈。如需如何使用 CloudFront API 變更發行版設定的相關資訊，請參閱UpdateDistribution。

使用 Amazon S3 主控台新增標頭到物件

使用 Amazon S3 主控台將 **Cache-Control** 或 **Expires** 標頭欄位新增至 Amazon S3 物件

1. 登入 AWS Management Console 並開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/。
2. 在儲存貯體清單中，選擇包含要新增標頭之檔案的儲存貯體名稱。
3. 選取要新增標頭的檔案或資料夾名稱旁的核取方塊。當您將標頭新增至資料夾時，它會影響該資料夾內的所有檔案。

4. 選擇 Actions (動作)，然後選擇 Edit metadata (編輯中繼資料)。
5. 在 Add metadata (新增中繼資料) 面板中，執行下列動作：
   a. 選擇 Add metadata (新增中繼資料)。
   b. 對於 Type (類型)，選擇 System defined (已定義系統)。
   c. 對於 Key (金鑰)，選擇您要新增的標頭名稱 (Cache-Control (快取控制) 或 Expires (過期))。
   d. 對於 Value (值)，輸入標頭值。例如，對於 `Cache-Control` 標題，您可以輸入 `max-age=86400`。對於 `Expires`，您可以輸入過期日期和時間，例如 `Wed, 30 Jun 2021 09:28:00 GMT`。
6. 在頁面底部，請選擇 Edit metadata (編輯中繼資料)。

根據查詢字串參數快取內容

有些 Web 應用程式使用查詢字串來傳送資訊到原始伺服器。查詢字串是 Web 請求的一部分，其顯示在 ? 字元之後；此字串可包含以 & 字元分隔的一或多個參數。在下列範例中，查詢字串包含兩個參數，*color=red* 及 *size=large*：

`https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red&size=large`

對於發行版，您可以選擇是否 CloudFront 要將查詢字串轉寄至來源，以及是否要根據所有參數或選取的參數快取內容。為什麼說這可能很有用？ 請考量下列範例。

假設您的網站提供五種語言。網站所有五種版本的目錄結構和檔案名稱都相同。當使用者檢視您的網站時，會根據使用者選擇的語言來轉寄要 CloudFront 包含語言查詢字串參數的要求。您可以設定 CloudFront 將查詢字串轉寄至來源，並根據語言參數進行快取。如果您將 Web 伺服器設定為傳回與所選語言對應的指定頁面版本，則會根據語言查詢字串參數的值個別 CloudFront快取每個語言版本。

在此範例中，如果您網站的主要頁面是`main.html`，下列五個請求會導 CloudFront 致快取`main.html`五次，每個語言查詢字串參數值都會有一次：

- `https://d111111abcdef8.cloudfront.net/main.html?language=de`
- `https://d111111abcdef8.cloudfront.net/main.html?language=en`
- `https://d111111abcdef8.cloudfront.net/main.html?language=es`
- `https://d111111abcdef8.cloudfront.net/main.html?language=fr`
- `https://d111111abcdef8.cloudfront.net/main.html?language=jp`

注意下列事項：

- 有些 HTTP 伺服器不處理查詢字串參數，因此不會傳回根據參數值的物件不同版本。對於這些來源，如果您設定 CloudFront 為將查詢字串參數轉寄至原始位置，即使 origin CloudFront 針對每個參數值傳回物件的相同版本， CloudFront 仍會根據參數值進行快取。
- 為使查詢字串參數能如以上範例所述搭配多種語言運作，各查詢字串參數間必須使用 & 字元做為分隔符號。如果您使用不同的分隔符號，可能會得到非預期的結果，這取決於您指定 CloudFront 要用作快取基礎的參數，以及參數在查詢字串中出現的順序而定。

  下列範例顯示如果您使用不同的分隔符號，且您設定 CloudFront 為僅根據`color`參數進行快取時會發生什麼情況：

  - 在下列要求中，會根據`color`參數的值 CloudFront 快取您的內容，但會將值 CloudFront 解譯為*###size* =large：

    ```
    https://d111111abcdef8.cloudfront.net/images/
    image.jpg?color=red;size=large
    ```

  - 在下列要求中，會 CloudFront 快取您的內容，但不會根據查詢字串參數進行快取。這是因為您設定 CloudFront 為根據`color`參數進行快取，但 CloudFront 會將下列字串解譯為僅包含*###size###color* =red：

    ```
    https://d111111abcdef8.cloudfront.net/images/
    image.jpg?size=large;color=red
    ```

您可以設定 CloudFront 為執行下列其中一項作業：

- 完全不要轉送查詢字串到原始伺服器。如果您不轉寄查詢字串，則不CloudFront 會根據查詢字串參數進行快取。
- 轉送查詢字串到原始伺服器，以及在查詢字串中根據所有參數快取。
- 轉送查詢字串到原始伺服器，以及在查詢字串中根據指定的參數快取。

如需詳細資訊，請參閱 the section called “最佳化快取”。

主題

- 查詢字串轉送和快取的主控台和 API 設定
- 最佳化快取
- 查詢字串參數和 CloudFront 標準記錄檔 (存取記錄)

查詢字串轉送和快取的主控台和 API 設定

若要在 CloudFront 主控台中設定查詢字串轉寄和快取，請參閱中的下列設定the section called “分佈設定”：

- the section called “查詢字串轉送和快取”
- the section called “查詢字串允許清單”

若要使用 API 設定查詢字串轉寄和快取，請參閱 Amazon CloudFront CloudFront API 參考中DistributionConfig和DistributionConfigWithTags中的以下設定：

- `QueryString`
- `QueryStringCacheKeys`

最佳化快取

當您設定 CloudFront 為根據查詢字串參數進行快取時，您可以採取下列步驟來減少 CloudFront 轉寄至來源的要求數目。當節 CloudFront 點為物件提供服務時，您可以減少原始伺服器上的負載並減少延遲，因為物件是從較靠近使用者的位置提供服務。

快取僅根據原始伺服器傳回物件不同版本的參數

對於 Web 應用程式轉寄至的每個查詢字串參數 CloudFront，會將每個參數值的要求 CloudFront轉送至您的來源，並為每個參數值快取物件的個別版本。即使原始伺服器一律傳回相同物件，無論參數值如何，情況都是如此。對於多個參數，請求的數量和物件的數量相乘。

我們建議您 CloudFront 將設定為僅根據原始碼傳回不同版本的查詢字串參數進行快取，並根據每個參數仔細考量快取的優點。例如，假設您有一個零售網站。您有六種不同顏色的夾克圖片，以及夾克有 10 種不同的大小。您已有的圖片顯示夾克的不同顏色，但不顯示不同大小。要優化緩存，您應該配置 CloudFront 為僅基於 color 參數緩存，而不是根據 size 參數。這會增加 CloudFront 可以提供快取要求的可能性，進而改善效能並減少原始伺服器的負載。

一律以相同順序列出參數

在查詢字串中參數的順序很重要。在下列範例中，查詢字串完全相同，除了參數的順序不同。這會導 CloudFront 致將兩個單獨的 image.jpg 請求轉發到您的來源，並緩存兩個單獨的對象版本：

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red&size=large`

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large&color=red`

我們建議您一律以相同順序列出參數名稱，例如字母順序。

參數名稱和值請一律使用相同的大小寫

CloudFront 根據查詢字串參數進行快取時，會考量參數名稱和值的大小寫。在下列範例中，查詢字串完全相同，除了參數名稱和值的大小寫以外。這會導 CloudFront 致將四個單獨的 image.jpg 請求轉發到您的來源，並緩存四個不同版本的對象：

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=Red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?Color=red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?Color=Red`

我們建議您為參數名稱和值使用一致的大小寫，例如全部小寫。

不使用與簽署 URL 衝突的參數名稱

如果您使用已簽署的 URL 來限制對內容的存取 (如果您已將受信任的簽署者新增至您的發佈)，請先 CloudFront 移除下列查詢字串參數，然後再將 URL 的其餘部分轉寄至您的來源：

- `Expires`
- `Key-Pair-Id`
- `Policy`
- `Signature`

如果您使用已簽署的 URL，而且想 CloudFront 要設定將查詢字串轉寄至您的 origin，則您自己的查詢字串參數無法命名為`Expires`、`Key-Pair-Id`、`Policy`、或`Signature`。

查詢字串參數和 CloudFront 標準記錄檔 (存取記錄)

如果啟用記錄，則會 CloudFront 記錄完整 URL，包括查詢字串參數。無論您是否已設定 CloudFront 為將查詢字串轉寄至原點，都是如此。如需 CloudFront 記錄的詳細資訊，請參閱the section called “使用標準日誌 (存取日誌)”。

根據 Cookie 快取內容

根據預設，在處理要求和回應時，或在邊緣位置快取物件時，不 CloudFront 會考慮 Cookie。如果 CloudFront 收到兩個除了`Cookie`標頭中的內容之外相同的請求，則默認情況下， CloudFront 將請求視為相同並為兩個請求返回相同的對象。

您可以設定 CloudFront 為在檢視器要求中將部分或全部 Cookie 轉寄至您的來源，並根據物件轉寄的 Cookie 值快取物件的不同版本。當您執行這項操作時， CloudFront 會使用檢視器要求中的部分或全部 Cookie (設定為轉寄的任何 Cookie) 來唯一識別快取中的物件。

例如，假設 `locations.html` 的請求包含具有 `country` 或 uk 值的 `fr` Cookie。當您根據 `country` Cookie CloudFront 的值設定快取物件時，會將要求 CloudFront 轉寄`locations.html`至來源，並包含 `country` Cookie 及其值。您的來源會傳回`locations.html`，並針對 `country` Cookie 值所在的要求 CloudFront 快取物件一次，uk並針對值所在的要求快取一次`fr`。

> **Important**
>
> Amazon S3 和一些 HTTP 伺服器不處理 Cookie。不要配置 CloudFront 為將 cookie 轉發到不處理 cookie 或不會根據 cookie 改變其響應的來源。這可能會導 CloudFront 致將更多請求轉發給同一個對象的來源，這會降低性能並增加原始對象的負載。如果考慮到上一個例子，您的來源不會處理 `country` Cookie，或始終返回相同版本的`locations.html`到，而不 CloudFront 管 `country` cookie 的值為何，請不 CloudFront 要配置為轉發該 cookie。
> 相反地，如果您的自訂來源取決於特定 Cookie，或根據 Cookie 傳送不同的回應，請務必設定為將該 Cookie 轉寄 CloudFront 至來源。否則，請先 CloudFront 移除 Cookie，然後再將請求轉寄至您的來源。

若要設定 Cookie 轉送，請更新分佈的快取行為。如需有關快取行為的詳細資訊，請參閱 快取行為設定，尤其是 轉送 Cookie 和 允許清單 Cookie 小節。

您可以設定每個快取行為，執行下列其中一項動作：

- 將所有 cookie 轉發到您的來源 — CloudFront 包括查看者在將請求轉發到來源時發送的所有 cookie。當您的來源傳回回應時，會使用檢視器要求中的 Cookie 名稱和值來 CloudFront 快取回應。如果原始響應包含`Set-Cookie`標題，則將其與請求對象一起 CloudFront 返回給查看器。CloudFront 還可以使用從來源返回的對象緩存`Set-Cookie`標頭，並將這些`Set-Cookie`標頭發送給所有緩存命中的查看器。

- 轉寄您指定的一組 Cookie — CloudFront 移除檢視者傳送的任何不在允許清單上的 Cookie，然後再將要求轉寄至原始位置。 CloudFront 使用檢視器要求中列出的 Cookie 名稱和值來快取回應。如果原始響應包含`Set-Cookie`標題，則將其與請求對象一起 CloudFront 返回給查看器。 CloudFront 還可以使用從來源返回的對象緩存`Set-Cookie`標頭，並將這些`Set-Cookie`標頭發送給所有緩存命中的查看器。

  如需有關在 Cookie 名稱中指定萬用字元的詳細資訊，請參閱 允許清單 Cookie。

  如需有關針對每個快取行為轉送的 Cookie 名稱數量的目前配額，或是有關請求更高配額的詳細資訊，請參閱 查詢字串的配額 (舊版快取設定)。
- 不要將 cookie 轉發到您的來源- CloudFront 不會根據查看器發送的 cookie 緩存您的對象。此外，在將請求轉寄到來源之前，請先移 CloudFront除 Cookie，並在將回覆傳回應給檢視者之前移除回應中的`Set-Cookie`標頭。由於這不是使用來源資源的最佳方式，因此當您選取此快取行為時，應確保您的來源預設不會在原始回應中包含 Cookie。

請注意以下有關指定您要轉送的 Cookie：

存取日誌

如果您配置 CloudFront 為記錄請求並記錄 cookie，則會 CloudFront 記錄所有 Cookie 和所有 Cookie 屬性，即使您配置為 CloudFront 不將 Cookie 轉發到您的來源，或者您配置 CloudFront 為僅轉發特定 cookie。如需 CloudFront 記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

區分大小寫

Cookie 名稱和值都是區分大小寫。例如，如果設定 CloudFront 為轉寄所有 Cookie，而同一物件的兩個檢視器要求具有相同的 Cookie (大小寫除外)，則會 CloudFront 快取物件兩次。

CloudFront 排序餅乾

如果配置 CloudFront 為轉發 cookie（全部或子集），請在將請求轉發到您的來源之前按 Cookie 名稱自然順 CloudFront 序對 Cookie 進行排序。

`If-Modified-Since` 和 `If-None-Match`

`If-Modified-Since`當設定為轉寄 Cookie (全部或子集) 時 CloudFront ，則不支援`If-None-Match`條件式要求。

需要標準名稱值組格式

CloudFront 只有在值符合標準名稱 — 值配對格式時，才會轉寄 Cookie 標頭，例如："`Cookie: cookie1=value1; cookie2=value2`"

停用 **Set-Cookie** 標頭的快取功能

如果配置 CloudFront 為將 cookie 轉發到來源（無論是全部還是特定 cookie），它還會緩存在源響應中收到的`Set-Cookie`標頭。 CloudFront 在對原始檢視器的回應中包含這些`Set-Cookie`標頭，並且還將它們包含在從 CloudFront 快取提供的後續回應中。

如果您想在來源接收 cookie，但不想緩存 CloudFront 來源響應中的`Set-Cookie`標題，請配置您的來源以添加帶有指定`Set-Cookie`為字段名稱的指`no-cache`令的`Cache-Control`標題。例如：`Cache-Control: no-cache="Set-Cookie"`。如需詳細資訊，請參閱 Hypertext Transfer Protocol (HTTP/1.1)：快取標準中的回應快取控制指令。

Cookie 名稱的長度上限

如果您設定為 CloudFront 將特定 Cookie 轉寄至您的來源，則您設定 CloudFront 要轉寄的所有 Cookie 名稱中的位元組總數不得超過 512 減去您要轉寄的 Cookie 數目。例如，如果您設定 CloudFront 將 10 個 Cookie 轉寄至您的來源，則 10 個 Cookie 名稱的總長度不能超過 502 個位元組 (512 — 10)。

如果您配置 CloudFront 為將所有 cookie 轉發到您的來源，則 cookie 名稱的長度無關緊要。

有關使用 CloudFront 控制台更新發行版以便將 cookie CloudFront 轉發到來源的信息，請參閱更新分佈。如需使用 CloudFront API 更新分發的相關資訊，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

根據請求標頭快取內容

CloudFront 可讓您選擇是否要 CloudFront 將標頭轉寄至來源，以及是否要根據檢視器要求中的標頭值快取指定物件的個別版本。這可讓您根據使用者使用的裝置、檢視器的位置、檢視器使用的語言，以及各種其他條件，提供內容的不同版本。

主題

- 標頭和分佈 - 概觀
- 選取快取時所依據的標頭
- CloudFront 進行配置以遵守 CORS 設置
- 根據裝置類型設定快取
- 根據檢視器語言設定快取
- 根據檢視器位置設定快取
- 根據請求的通訊協定設定快取

- 設定壓縮檔案的快取
- 快取如何根據標頭影響效能
- 標頭大小寫和標頭值如何影響快取
- CloudFront 返回檢視器的標頭

標頭和分佈 - 概觀

根據預設，在邊緣位置快取物件時， CloudFront 不會考慮標頭。如果您的 origin 傳回兩個物件，而且它們只與要求標頭中的值不同，則只會 CloudFront 快取物件的一個版本。

您可以設定 CloudFront 將標頭轉寄至來源，這會 CloudFront 根據一或多個要求標頭中的值，快取物件的多個版本。若要根據特定標頭的值設定快取物件，您可CloudFront 以指定散發的快取行為設定。如需詳細資訊，請參閱根據選取的請求標頭執行快取。

例如，假設 `logo.jpg` 的檢視器請求包含具有 `Product` 或 `Acme` 值的客戶 `Apex` 標頭。當您設定 CloudFront 為根據`Product`標頭的值快取物件時，會將要求 CloudFront 轉寄`logo.jpg`至 origin，並包含標`Product`頭和標頭值。 CloudFront 針對`Product`標頭值所在的要求快取`logo.jpg`一次，`Acme`並針對值所在的要求快取一次`Apex`。

您可以在分佈中設定每個快取行為，以執行以下其中一項：

- 轉送所有標頭到原始伺服器

> **Note**
>
> 對於舊版快取設定 — 如果您設定 CloudFront 為將所有標頭轉寄至原始位置，則 CloudFront 不會快取與此快取行為相關聯的物件。反之，它會傳送每個請求到原始伺服器。

- 轉寄您指定的標頭清單。 CloudFront 根據所有指定標頭中的值來快取物件。 CloudFront 依預設，也會轉寄它轉寄的標頭，但它只會根據您指定的標頭快取物件。
- 只轉送預設標頭。在此配置中， CloudFront 不會根據請求標頭中的值緩存對象。

如需有關針對每個快取行為轉送單的標頭數量的目前配額，或是有關請求更高配額的詳細資訊，請參閱標頭的配額。

有關使用 CloudFront 控制台更新發行版以便將標頭 CloudFront 轉發到原點的信息，請參閱更新分佈。如需使用 CloudFront API 更新現有分發的相關資訊，請參閱 Amazon CloudFront API 參考中的更新分發。

選取快取時所依據的標頭

您可以轉寄到原始伺服器以及快取的 CloudFront 標頭取決於您的來源是 Amazon S3 儲存貯體還是自訂來源。

- Amazon S3 — 您可以設定 CloudFront 為根據多個特定標頭轉寄和快取物件 (請參閱以下例外清單)。不過，我們建議您避免轉送具有 Amazon S3 原始伺服器的標頭，除非您需要實作跨來源資源共享 (CORS)，或是想要在原始伺服器面向事件中使用 Lambda@Edge 將內容個人化。
  - 若要設定 CORS，您必須轉寄允許 CloudFront 為啟用跨來源資源共用 (CORS) 的網站散發內容的標頭。如需詳細資訊，請參閱 CloudFront 進行配置以遵守 CORS 設置。
  - 若要使用轉寄至 Amazon S3 來源的標頭來個人化內容，您可以撰寫並新增 Lambda @Edge 函數，並將它們與您的 CloudFront 分發產生關聯，以便由面向原點的事件觸發。如需使用標頭來個人化內容的詳細資訊，請參閱根據國家/地區或裝置類型標頭個人化 - 範例。

    我們建議您避免轉送沒有要用來個人化內容的標頭，因為轉送額外的標頭可能會降低您的快取命中率。也就是說，不 CloudFront 能作為所有請求的比例來提供盡可能多的請求來自邊緣緩存的請求。
- 自訂來源 — 您可以設定 CloudFront 為根據任何要求標頭的值進行快取，但下列項目除外：
  - `Connection`
  - `Cookie` - 如果要根據 Cookie 來轉送及快取，您可以在分佈中使用個別的設定。如需詳細資訊，請參閱 根據 Cookie 快取內容。
  - `Host (for Amazon S3 origins)`
  - `Proxy-Authorization`
  - `TE`
  - `Upgrade`

  您可以設定 CloudFront 為根據`Date`和`User-Agent`標頭中的值快取物件，但我們不建議這麼做。這些標頭具有許多可能的值，並且基於其值的緩存可能會導CloudFront 致將更多請求轉發到您的來源。

如需 HTTP 要求標頭及其 CloudFront 處理方式的完整清單，請參閱HTTP 請求標頭和 CloudFront 行為 (自訂和 Amazon S3 來源)。

CloudFront 進行配置以遵守 CORS 設置

如果您已在 Amazon S3 儲存貯體或自訂原始伺服器上啟用跨來源資源共享 (CORS)，您必須選擇特定的標頭進行轉發，以遵守 CORS 設定。您必須轉發的標頭會因原始伺服器 (Amazon S3 或自訂)，以及您是否要快取 OPTIONS 回應而異。

Amazon Simple Storage Service (Amazon S3)

- 如果您希望快取 OPTIONS 回應，請執行下列動作：
  - 選擇預設快取行為設定的選項，來啟用 OPTIONS 回應的快取。
  - 設定 CloudFront 為轉寄下列標頭：`Origin`、`Access-Control-Request-Headers`、和`Access-Control-Request-Method`。
- 如果您不想快取`OPTIONS`回應，請設定CloudFront 轉寄標`Origin`頭，以及來源所需的任何其他標頭 (例如`Access-Control-Request-Headers`、`Access-Control-Request-Method`或其他標頭)。

自訂原始伺服器 - 轉送 `Origin` 標頭與原始伺服器需要的其他任何標頭。

若 CloudFront 要設定以根據 CORS 快取回應，您必須使用快取原則設定 CloudFront 為轉寄標頭。如需詳細資訊，請參閱 使用政策。

如需 CORS 和 Amazon S3 的詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的使用跨來源資源分享 (CORS)。

根據裝置類型設定快取

如果您想 CloudFront 要根據使用者用來檢視內容的裝置快取物件的不同版本，請設定為將適用的標頭轉寄 CloudFront 至您的自訂來源：

- `CloudFront-Is-Desktop-Viewer`
- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-SmartTV-Viewer`
- `CloudFront-Is-Tablet-Viewer`

根據標`User-Agent`頭的值，將這些標頭的值 CloudFront 設置為`true`或轉發請求到您的來源`false`之前。如果裝置屬於多個類別，一個以上的值可能是 `true`。例如，對於某些平板電腦裝置， CloudFront

可能會同時CloudFront-Is-Tablet-Viewer將CloudFront-Is-Mobile-Viewer和設定為true。

根據檢視器語言設定快取

如果您想 CloudFront 要根據要求中指定的語言快取物件的不同版本，請設定將Accept-Language標頭轉寄 CloudFront 至您的來源。

根據檢視器位置設定快取

如果您想 CloudFront 要根據要求來自的國家/地區快取物件的不同版本，請設定 CloudFront 將CloudFront-Viewer-Country標頭轉寄至您的來源。 CloudFront 自動將要求來自的 IP 位址轉換為兩個字母的國碼。有關可按代碼和國家/地區名稱排序的國家/地區代碼 easy-to-use 列表，請參閱維基百科條目 ISO 316 6-1 α-2。

根據請求的通訊協定設定快取

如果您想 CloudFront 要根據要求的通訊協定 (HTTP 或 HTTPS) 快取不同版本的物件，請設定 CloudFront 為將CloudFront-Forwarded-Proto標頭轉寄至您的來源。

設定壓縮檔案的快取

如果您的原始伺服器支援 Brotli 壓縮，您可以根據 Accept-Encoding 標頭快取。只有在您的原始伺服器根據標頭提供不同內容時，才應根據 Accept-Encoding 設定快取。

快取如何根據標頭影響效能

當您設定 CloudFront 為根據一或多個標頭進行快取，且標頭具有多個可能值時，會針對相同物件將多個要求 CloudFront 轉送至原始伺服器。這會降低效能和增加原始伺服器的負載。如果您的原始服務器返回相同的對象，而不管給定標頭的值如何，我們建議您不 CloudFront 要配置為基於該標頭緩存。

如果您設定 CloudFront 轉寄多個標頭，只要值相同，檢視器要求中的標頭順序就不會影響快取。例如，如果一個要求包含標頭 A: 1、B: 2，而另一個要求包含 B: 2、A: 1，則只會 CloudFront 快取物件的一個副本。

標頭大小寫和標頭值如何影響快取

當基於標題值進行 CloudFront 緩存時，它不會考慮標題名稱的大小寫，但它確實考慮了標題值的情況：

- 如果檢視者要求同時包含Product:Acme和product:Acme，則只會 CloudFront 快取一個物件一次。它們之間的唯一差別是不會影響快取的標頭大小寫。
- 如果檢視器要求同時包含Product:Acme和Product:acme，則會 CloudFront 快取物件兩次，因為該值Acme位於某些要求和其他要求acme中。

CloudFront 返回檢視器的標頭

設定 CloudFront 轉寄和快取標頭不會影響傳 CloudFront回檢視器的標頭。 CloudFront 返回它從原點獲得的所有標題，但有一些例外。如需詳細資訊，請參閱適用的主題：

- Amazon S3 原始伺服器 - 請參閱可 CloudFront移除或更新的 HTTP 回應標頭。
- 自訂原始伺服器 - 請參閱 CloudFront 移除或取代的 HTTP 回應標頭。

故障診斷

疑難排解您在設定 Amazon CloudFront 以分發內容或使用 Lambda @Edge 時可能遇到的常見問題，並找出可能的解決方案。

主題

故障診斷分佈問題

使用此處的資訊可協助您診斷和修正憑證錯誤、存取被拒的問題，或是在使用 Amazon CloudFront 分發設定網站或應用程式時可能遇到的其他常見問題。

主題

CloudFront 返回一個Access Denied錯誤

如果您使用 Amazon S3 儲存貯體做為 CloudFront 分發的來源，您可能會在下列範例中看到拒絕存取 (403) 錯誤訊息。

內容

您從 Amazon S3 來源指定了遺失的物件

確認值區中要求的物件是否存在。物件名稱區分大小寫。輸入無效的物件名稱可能會傳回拒絕存取錯誤碼。

例如，如果您按照CloudFront 教學建立基本分發，則建立 Amazon S3 儲存貯體做為來源並上傳範例`index.html`檔案。

在網頁瀏覽器中，如果您輸入`https://d111111abcdef8.cloudfront.net/INDEX.HTML`而不是`https://d111111abcdef8.cloudfront.net/index.html`，您可能會看到類似的訊息，因為 URL 路徑中的`index.html`檔案區分大小寫。

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>22Q367AHT7Y1ABCD</RequestId>
<HostId>
ABCDE/Vg+7PSNa/d/IfFQ8Fb92TGQ0KH0ZwG5iEKbc6+e06DdMS1ZW+ryB9GFRIVtS66rSSy6So=
</HostId>
</Error>
```

您的 Amazon S3 來源缺少 IAM 許可

確認您已選取正確的 Amazon S3 儲存貯體做為原始網域和名稱。來源 (Amazon S3) 必須具有正確的許可。

如果您未指定正確的權限，則檢視者可能會看到下列「拒絕存取」訊息。

```
<Code>AccessDenied</Code>
<Message>User: arn:aws:sts::856369053181:assumed-role/OriginAccessControlRole/
EdgeCredentialsProxy+EdgeHostAuthenticationClient is not authorized to perform:
 kms:Decrypt on the resource associated with this ciphertext because the resource does
 not exist in this Region, no resource-based policies allow access, or a resource-based
 policy explicitly denies access</Message>
<RequestId>22Q367AHT7Y1ABCD/RequestId>
<HostId>
ABCDE/Vg+7PSNa/d/IfFQ8Fb92TGQ0KH0ZwG5iEKbc6+e06DdMS1ZW+ryB9GFRIVtS66rSSy6So=
</HostId>
</Error>
```

> **Note**
>
> 在這個錯誤訊息中，帳戶識別碼 856369053181 是受管理的帳戶。 AWS

當您從 Amazon S3 分發內容時，同時也使用 AWS Key Management Service (AWS KMS) 服務端加密 (SSE-KMS)，您需要為 KMS 金鑰和 Amazon S3 儲存貯體指定其他 IAM 許可。您的 CloudFront 分發需要這些許可才能使用 KMS 金鑰，該金鑰用於加密原始 Amazon S3 儲存貯體。

Amazon S3 儲存貯體政策的組態可讓 CloudFront 分發擷取加密物件以進行內容交付。

驗證您的 Amazon S3 儲存貯體和 KMS 金鑰許可

1. 確認您使用的 KMS 金鑰與 Amazon S3 儲存貯體用於預設加密的金鑰相同。如需詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的使用 AWS KMS (SSE-KMS) 指定伺服器端加密。
2. 確認儲存貯體中的物件已使用相同的 KMS 金鑰加密。您可以從 Amazon S3 儲存貯體選取任何物件，然後檢查伺服器端加密設定以驗證 KMS 金鑰 ARN。
3. 編輯 Amazon S3 儲存貯體政策，以 CloudFront 授與從 Amazon S3 儲存貯體呼叫 `GetObject` API 操作的權限。如需使用來源存取控制的 Amazon S3 儲存貯體政策範例，請參閱授予原始存取控制許可，以存取 S3 儲存貯體。
4. 編輯 KMS 金鑰原則，以 CloudFront 授與`Encrypt`、和執行動作的權`Decrypt`限`GenerateDataKey*`。若要以最低權限權限對齊，請指定`Condition`元素，以便只有指定的 CloudFront 分佈才能執行列出的動作。您可以自訂現有原則的 AWS KMS 原則。如需 KMS 金鑰原則範例，請參閱SSE-KMS.

如果您使用來源存取身分識別 (OAI) 而非 OAC，Amazon S3 儲存貯體的許可會略有不同，因為您授與身分而非. AWS 服務如需詳細資訊，請參閱 授予原始存取身分的許可，以讀取 Amazon S3 儲存貯體中的檔案。

如果仍然無法檢視發行版中的檔案，請參閱我無法在我的分佈中檢視檔案。

您使用的憑證無效或沒有足夠的權限

如果您使用不正確或過期的 AWS SCT 認證 (存取金鑰和秘密金鑰)，或者您的 IAM 角色或使用者缺少對 CloudFront 資源執行動作所需的權限，則會出現「拒絕存取」錯誤訊息。如需有關存取遭拒錯誤訊息的詳細資訊，請參閱 IAM 使用者指南中的疑難排解拒絕存取錯誤訊息。

如需 IAM 如何使用的相關資訊 CloudFront，請參閱Amazon Identity and Access Management CloudFront。

CloudFront 當我嘗試添加替代域名時返回InvalidViewerCertificate錯誤

如 CloudFront 果在嘗試將替代網域名稱 (CNAME) 新增至發行版時傳回`InvalidViewerCertificate`錯誤，請檢閱下列資訊以協助疑難排解問題。此錯誤可指出必須解決以下其中一個問題，才能成功新增替代網域名稱。

下列錯誤會依照 CloudFront 檢查新增替代網域名稱的授權順序列出。這可協助您疑難排解問題，因為根據 CloudFront 傳回的錯誤，您可以判斷哪些驗證檢查已成功完成。

沒有將憑證連接到您的分佈。

若要新增替代網域名稱 (CNAME)，您必須將信任、有效的憑證連接到您的分佈。請檢閱需求、取得足以滿足需求的有效憑證、將它連接到您的分佈，然後再試一次。如需詳細資訊，請參閱 使用備用網域名稱的需求。

您已連接憑證的憑證鏈結中有太多憑證。

您在憑證鏈結中最多只能有五個憑證。請減少鏈結中的憑證數，然後再試一次。

憑證鏈結中包含一或多個在目前日期無效的憑證。

您已新增憑證的憑證鏈結中有一或多個無效憑證，可能是因為憑證尚未有效或已過期。請檢查您憑證鏈結中憑證內的 Not Valid Before (生效日期) 和 Not Valid After (失效日期) 欄位，確保根據您所列出的日期，所有憑證都是有效的。

您已連接的憑證沒有經過信任的憑證授權單位 (CA) 簽章。

您附加用來 CloudFront 驗證替代網域名稱的憑證不能是自我簽署憑證。它必須經過信任 CA 的簽章。如需詳細資訊，請參閱 使用備用網域名稱的需求。

您連接的憑證格式不正確

包含在憑證中的網域名稱和 IP 地址格式，以及憑證本身的格式必須遵循憑證的標準。

發生內 CloudFront 部錯誤。

CloudFront 被內部問題阻止，無法對證書進行驗證檢查。在這個案例中，會 CloudFront 傳回 HTTP 500 狀態碼，並指出附加憑證時發生內部 CloudFront 問題。請等待幾分鐘，然後再次嘗試新增替代網域名稱和憑證。

您已連接的憑證並未涵蓋您嘗試新增的替代網域名稱。

對於您新增的每個替代網域名稱，都 CloudFront 需要附加來自受信任憑證授權單位 (CA) 的有效 SSL/TLS 憑證，該憑證涵蓋網域名稱，以驗證您使用該憑證的授權。請更新您的憑證，以包含能涵蓋您嘗試新增之 CNAME 的網域名稱。如需搭配萬用字元使用網域名稱的詳細資訊和範例，請參閱使用備用網域名稱的需求。

我無法在我的分佈中檢視檔案

如果您無法檢視 CloudFront 發行版中的檔案，請參閱下列主題以取得一些常見解決方案。

您是否註冊了兩者 CloudFront 和 Amazon S3？

要將 Amazon CloudFront 與 Amazon S3 起源一起使用，您必須分別註冊兩者 CloudFront 和 Amazon S3。如需註冊 CloudFront 和 Amazon S3 的詳細資訊，請參閱設定。

Amazon S3 儲存貯體與物件許可是否設定正確？

如果您 CloudFront 搭配 Amazon S3 來源使用，則內容的原始版本會存放在 S3 儲存貯體中。CloudFront 與 Amazon S3 搭配使用最簡單的方法是讓所有物件在 Amazon S3 中公開讀取。若要執行此作業，必須確實啟用上傳至 Amazon S3 的每一個物件的公有讀取權限。

如果您的內容無法公開讀取，您必須建立 CloudFront 原始存取控制 (OAC)，才 CloudFront 能存取內容。如需 CloudFront 原始存取控制的詳細資訊，請參閱the section called “限制對 Amazon 簡單儲存服務來源的存取”。

物件屬性與儲存貯體屬性無關。您必須明確授予 Amazon S3 儲存貯體中每個物件的權限。物件不會從儲存貯體繼承屬性，必須獨立於儲存貯體設定物件屬性。

替代網域名稱 (CNAME) 是否正確設定？

如果網域名稱已有現有的 CNAME 記錄，請更新該記錄或將其替換為指向分佈網域名稱的新記錄。

此外，請確保 CNAME 記錄指向分佈的網域名稱，而不是 Amazon S3 儲存貯體。可確認 DNS 系統中的 CNAME 記錄指向分佈的網域名稱。若要執行此作業，請使用像 dig 這樣的 DNS 工具。

以下範例顯示了一個名為`images.example.com` 的網域名稱的 dig 請求和回應的相關部分。在 `ANSWER SECTION`，查看包含 `CNAME` 的列。如果 CNAME 右邊的值是您 CloudFront 分發的網域名稱，則您網域名稱的 CNAME 記錄設定正確。如果是 Amazon S3 原始伺服器儲存貯體或某些其他網域名稱，則 CNAME 記錄設定便不正確。

```
[prompt]> dig images.example.com

; <<> DiG 9.3.3rc2 <<> images.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;images.example.com.    IN  A
;; ANSWER SECTION:
images.example.com. 10800 IN CNAME d111111abcdef8.cloudfront.net.
...
...
```

如需 CNAME 的詳細資訊，請參閱新增替代網域名稱 (CNAME) 以使用自訂 URL。

您是否為您的 CloudFront 分發引用了正確的 URL？

確保您引用的 URL 使用 CloudFront 分發的網域名稱 (或 CNAME)，而不是 Amazon S3 儲存貯體或自訂來源。

是否需要協助故障診斷自訂原始伺服器？

如果您需 AWS 要協助您對自訂來源進行疑難排解，我們可能需要檢查您要求中的`X-Amz-Cf-Id`標頭項目。如果尚未記錄這些項目，可能要考慮未來的記錄。如需詳細資訊，請參閱 the section called “使用 Amazon EC2（或其他自定義源地）”。如需進一步協助，請參閱 AWS 支援中心。

錯誤訊息：憑證：<certificate-id>正在使用 CloudFront

問題：您嘗試從 IAM 憑證存放區刪除 SSL/TLS 憑證，而且收到「憑證：<certificate-id>正在使用中」訊息。 CloudFront

解決方案：每個 CloudFront 發行版都必須與預設 CloudFront 憑證相關聯，或與自訂 SSL/TLS 憑證相關聯。刪除 SSL/TLS 憑證之前，您必須輪替憑證 (以另一個自訂 SSL/TLS 憑證取代目前的自訂 SSL/TLS 憑證)，或從使用自訂 SSL/TLS 憑證還原為使用預設憑證，才能刪除 SSL/TLS 憑證。 CloudFront 若要進行修復，請完成以下其中一個程序中的步驟：

- 輪換 SSL/TLS 憑證
- 從自訂 SSL/TLS 憑證還原為預設憑證 CloudFront

從原始伺服器故障診斷錯誤回應

如果從您的來源 CloudFront 請求一個對象，並且來源返回 HTTP 4xx 或 5xx 狀態碼，則與您的來源之間 CloudFront 的通信存在問題。以下幾個主題說明其中一些 HTTP 狀態碼的常見原因，並提供一些可能的解決方案。

主題

HTTP 400 狀態碼 (錯誤的請求)

您的 CloudFront 發行版可能會使用 HTTP 狀態碼 400 錯誤請求發送錯誤響應，以及類似以下內容的消息：

```
########### '< ##AWS >' ##### '< ## >'AWS
```

例如：

授權標頭格式不正確；區域 'us-east-1' 是錯誤的；應該是 'us-west-2'

在下列情況下可能會發生這個問題：

1. 您的 CloudFront 分發來源是一個 Amazon S3 存儲桶。
2. 您將 S3 儲存貯體從一個 AWS 區域移到另一個區域。也就是說，您刪除了 S3 儲存貯體，之後建立了一個具有相同儲存貯體名稱的新儲存貯體，但位於與原始 S3 儲存貯體所在的 AWS 區域不同。

若要修正此錯誤，請更新您的 CloudFront 分發，以便在儲存貯體的目前 AWS 區域中找到 S3 儲存貯體。

若要更新您的 CloudFront 發行版

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇產生此錯誤的分佈。
3. 選擇 Origins and Origin Groups (原始伺服器和原始伺服器群組)。
4. 尋找您所移動 S3 儲存貯體的原始來源。選取此原始來源旁邊的核取方塊，然後選擇 Edit (編輯)。
5. 請選擇 Yes, Edit (是，編輯)。在選擇 Yes, Edit (是、編輯) 之前，不需要變更任何設定。

當您完成這些步驟時，請 CloudFront 重新部署您的發行版本。在部署發行版時，您會在 [上次修改] 欄下看到 [部署] 狀態。部署完成一段時間後，您應該停止接收`AuthorizationHeaderMalformed`錯誤回應。

HTTP 502 狀態碼 (無效的閘道)

HTTP 502 狀態碼（錯誤的網關）表示 CloudFront 無法提供請求的對象，因為它無法連接到原始服務器。

主題

- CloudFront 和自訂原始伺服器之間的 SSL/TLS 交涉失敗
- 原始伺服器無法回應受支援的加密/通訊協定
- 原始伺服器的 SSL/TLS 憑證已過期、無效、已自我簽署，或憑證鏈順序錯誤
- 在原始伺服器設定中原始伺服器無法回應指定的連接埠

CloudFront 和自訂原始伺服器之間的 SSL/TLS 交涉失敗

如果您使用自訂來源，且設定 CloudFront 為在 CloudFront 與原始伺服器之間需要 HTTPS，則問題可能是網域名稱不相符。在原始伺服器上安裝的 SSL/TLS 憑證在 Common Name (通用名稱) 欄位包括網域名稱，且在 Subject Alternative Names (主體別名) 欄位中可能還有更多。(在憑證網域名稱中 CloudFront 支援萬用字元。) 憑證中的其中一個網域名稱必須符合一或兩個下列的值：

- 您為發行版中適用來源的原始網域指定的值。
- `Host`標頭的值，如果您配置 CloudFront 為將`Host`標頭轉發到您的來源。如需有關轉發 `Host` 標頭到原始伺服器的詳細資訊，請參閱根據請求標頭快取內容。

如果網域名稱不相符，SSL/TLS 交涉就會失敗，並 CloudFront 傳回 HTTP 狀態碼 502 (錯誤的閘道)，並將`X-Cache`標頭設定為。`Error from cloudfront`

若要判斷憑證中的網域名稱是否與發行版或`Host`標頭中的原始網域相符，您可以使用線上 SSL 檢查程式或 OpenSSL。如果網域名稱不相符，您有兩個選項：

- 您為 Origin Domain Name (原始網域名稱) 指定的值，適用於您分佈中的可用來源。
- `Host`標頭的值，如果您配置 CloudFront 為將`Host`標頭轉發到您的來源。如需有關轉發 `Host` 標頭到原始伺服器的詳細資訊，請參閱根據請求標頭快取內容。

如果網域名稱不相符，SSL/TLS 交涉就會失敗，並 CloudFront 傳回 HTTP 狀態碼 502 (錯誤的閘道)，並將`X-Cache`標頭設定為。`Error from cloudfront`

若要判斷憑證中的網域名稱是否與分佈或 `Host` 標頭中的 Origin Domain Name (原始伺服器網域名稱) 相符，您可以使用線上 SSL 檢查或 OpenSSL。如果網域名稱不相符，您有兩個選項：

- 取得包括適用的網域名稱的新 SSL/TLS 憑證。

  如果您使用 AWS Certificate Manager (ACM)，請參閱使用AWS Certificate Manager 者指南中的要求公用憑證以要求新憑證。
- 變更發佈組態，以便 CloudFront 不再嘗試使用 SSL 與原始伺服器連線。

線上 SSL 檢查

若要尋找 SSL 測試工具，請搜尋網際網路「線上 ssl 檢查。」 通常需要指定網域名稱，且此工具會傳回有關 SSL/TLS 憑證的各種資訊。確認憑證在 Common Name (通用名稱) 或 Subject Alternative Names (主體別名) 欄位中包含您的網域名稱。

OpenSSL

若要協助疑難排解來源的 HTTP 502 錯誤 CloudFront，您可以使用 OpenSSL 嘗試建立與原始伺服器的 SSL/TLS 連線。如果 OpenSSL 無法建立連線，可能表示原始伺服器的 SSL/TLS 組態發生問題。如果 OpenSSL 能夠建立連線，它會傳回原始伺服器憑證的相關資訊，包括憑證的通用名稱 (`Subject CN` 欄位) 和主體別名 (`Subject Alternative Name` 欄位)。

使用下列 OpenSSL 指令來測試與原始伺服器的連線 (以原*#######*名稱取代原始網域，例如 example.com)：

```
openssl s_client -connect origin domain name:443
```

如果下列為真：

- 原始伺服器支援有多個 SSL/TLS 憑證的多個網域名稱
- 您的分佈設定為將 `Host` 標頭轉送至原始伺服器

然後將 `-servername` 選項新增至 OpenSSL 命令，如下列範例所示 (將 *CNAME* 取代為您的分佈中設定的 CNAME)：

```
openssl s_client -connect origin domain name:443 -servername CNAME
```

原始伺服器無法回應受支援的加密/通訊協定

CloudFront 使用密碼和通訊協定連線到原始伺服器。如需 CloudFront 支援的密碼和通訊協定清單，請參閱。the section called “與來源之間支援的通訊協定 CloudFront 和密碼”如果您的來源沒有回應 SSL/TLS 交換中的這些密碼或通訊協定之一， CloudFront 則無法連線。您可以使用如 SSL Labs (SSL 實驗室) 等線上工具來驗證原始伺服器是否支援密碼加密和通訊協定。在 Hostname (主機名稱) 欄位中輸入原始伺服器的網域名稱，然後選擇 Submit (提交)。從測試來檢閱 Common names (通用名稱) 和 Alternative names (替代名稱) 欄位，查看它們是否符合原始伺服器的網域名稱。測試完成後，在測試結果中找到 Protocols (通訊協定) 與 Cipher Suites (密碼套件) 區段，查看原始伺服器支援哪些密碼加密或通訊協定。將其與 the section called “與來源之間支援的通訊協定 CloudFront 和密碼” 的清單進行比較。

原始伺服器的 SSL/TLS 憑證已過期、無效、已自我簽署，或憑證鏈順序錯誤

如果原始伺服器傳回下列內容， CloudFront 請中斷 TCP 連線、傳回 HTTP 狀態碼 502 (錯誤閘道)，並將標`X-Cache`頭設定為`Error from cloudfront`：

- 過期的憑證
- 無效的憑證
- 已自我簽署的憑證
- 憑證鏈順序錯誤

> **Note**
>
> 如果完整的憑證鏈結 (包括中繼憑證) 不存在， CloudFront 就會中斷 TCP 連線。

如需有關在自訂原始伺服器上安裝 SSL/TLS 憑證的詳細資訊，請參閱 the section called “要求使用 HTTPS 連接到自訂原始伺服器”。

在原始伺服器設定中原始伺服器無法回應指定的連接埠

當您在 CloudFront 發行版上建立來源時，您可以針對 HTTP 和 HTTPS 流量設定 CloudFront 連接至原始伺服器的連接埠。根據預設，這些都是 TCP 80/443。您可以選擇修改這些連接埠。如果您的來源因為任何原因拒絕這些連接埠上的流量，或者您的後端伺服器在連接埠上沒有回應， CloudFront 將無法連線。

若要故障診斷這些問題，請檢查基礎設施中執行的任何防火牆和驗證它們不會封鎖受支援的 IP 範圍。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的 AWS IP 地址範圍。此外，驗證 Web 伺服器是否可在原始伺服器上執行。

HTTP 502 狀態碼 (Lambda 驗證錯誤)

如果您使用 Lambda@Edge，HTTP 502 狀態碼可能表示您的 Lambda 函數回應的格式不正確或包含無效的內容。如需針對 Lambda@Edge 錯誤進行故障診斷的詳細資訊，請參閱測試和偵 Lambda 函數 @Edge。

HTTP 502 狀態碼 (DNS 錯誤)

含有錯誤碼的 HTTP 502 `NonS3OriginDnsError` 錯誤表示發生 DNS 組態問題，無法連線 CloudFront 至原始伺服器。如果您從中收到此錯誤 CloudFront，請確定來源的 DNS 組態正確且正常運作。

當 CloudFront 收到對象的請求已過期或不在其緩存中時，它會向來源發出請求以獲取該對象。若要成功向來源發出要求，請在原始網域上 CloudFront 執行 DNS 解析。如果您網域的 DNS 服務發生問題，則 CloudFront 無法解析網域名稱以取得 IP 位址，這會導致 HTTP 502 錯誤 (`NonS3OriginDnsError`)。若要修正此問題，請聯絡您的 DNS 供應商，或者如果您使用 Amazon Route 53，請參閱為什麼我無法存取使用 Route 53 DNS 服務的網站？

若要進一步疑難排解此問題，請確保原始伺服器根網域的 authoritative name servers (授權名稱伺服器) 或 Zone Apex (例如 `example.com`) 正確運作。您可以使用以下命令來尋找適用於 apex 原始伺服器的名稱，使用 dig 或 nslookup 等工具：

```
dig OriginAPEXDomainName NS +short
```

```
nslookup -query=NS OriginAPEXDomainName
```

當您有名稱伺服器的名稱時，請使用下列命令針對他們的原始伺服器的網域名稱做查詢，以確保每個回應都有一個答案：

```
dig OriginDomainName @NameServer
```

```
nslookup OriginDomainName NameServer
```

> **Important**
>
> 請務必使用連線至公用網際網路的電腦執行此 DNS 疑難排解。 CloudFront 使用網際網路上的公用 DNS 來解析原始網域，因此在類似內容中進行疑難排解很重要。

如果您的原始網域是一個子網域，其 DNS 授權委派給與根網域不同的名稱伺服器，請確定該子網域的名稱伺服器 (NS) 和授權開始 (SOA) 記錄已正確設定。您可以使用類似上述範例的命令來檢查這些記錄。

如需 DNS 的詳細資訊，請參閱 Amazon Route 53 說明文件中的網域名稱系統 (DNS) 概念。

HTTP 503 狀態碼（函數執行錯誤）

如果您使用的是 Lambda @Edge 或 CloudFront 函數，HTTP 503 狀態碼可以指出您的函數傳回了執行錯誤。

如需針對 Lambda@Edge 錯誤進行故障診斷的詳細資訊，請參閱測試和偵 Lambda 函數 @Edge。

如需疑難排解 CloudFront 函數的詳細資訊，請參閱測試函數。

HTTP 503 狀態碼 (超過 Lambda 限制)

如果您使用 Lambda@Edge，HTTP 503 狀態碼可能表示 Lambda 服務傳回了錯誤。此錯誤可能由以下其中一項原因造成：

- 函數執行次數超過 Lambda 為限制 AWS 區域中的執行而設定的其中一個配額 (以前稱為限制) (並行執行或叫用頻率)。
- 此函數已超過 Lambda 函數逾時配額。

如需有關 AWS Lambda 配額的詳細資訊，請參閱AWS Lambda 開發人員指南中的 Lambda 配額。如需針對 Lambda@Edge 錯誤進行故障診斷的詳細資訊，請參閱the section called “測試和除錯”。

HTTP 503 狀態碼 (服務無法使用)

HTTP 503 狀態碼 (服務無法使用) 通常表示原始伺服器上的效能問題。在極少數情況下，表示由於節點的資源限制， CloudFront 暫時無法滿足請求。

主題

原始伺服器沒有足夠的容量來支援請求率

CloudFront 當原始伺服器不堪重負傳入的要求時，會產生此錯誤。 CloudFront 然後將錯誤轉發回給用戶。若要解決這個問題，請嘗試下列解決方案：

- 如果您使用 Amazon S3 做為原始伺服器，則根據下列適用於金鑰命名的最佳實務來最佳化 Amazon S3 的效能。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的最佳化 Amazon S3 效能。
- 如果您使用 Elastic Load Balancing 做為原始伺服器，請參閱如何疑難排解使用 Classic Load Balancer 時傳回的 503 錯誤？
- 如果您使用自訂原始伺服器，請檢查應用程式日誌，以確保原始伺服器有足夠的資源，例如記憶體、CPU 和磁碟大小。如果使用 Amazon EC2 做為後端，請確保執行個體類型都有適當的資源，以符合傳入的請求。如需詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的執行個體類型。

CloudFront 由於邊緣位置的資源限制導致錯誤

在無法將要求路由到下一個最佳 CloudFront 可用節點的罕見情況下，您會收到此錯誤訊息，因此無法滿足要求。當您對 CloudFront發行版執行負載測試時，此錯誤很常見。為協助防止此種情況，請遵循 the section called “負載測試 CloudFront” 準則，以避免 503 (容量超過) 錯誤。

如果您的生產環境中發生這種情況，請聯絡 AWS Support。

HTTP 504 狀態碼 (閘道逾時)

HTTP 504 狀態碼（網關超時）表示將請求 CloudFront 轉發到來源時（因為請求的對象不在邊緣緩存中）時，會發生以下情況之一：

- 原點將一個 HTTP 504 狀態碼返回到 CloudFront。

- 原始伺服器未在請求逾期之前回應。

CloudFront 如果流量被防火牆或安全組阻止到來源，或者在互聯網上無法訪問來源，將返回 HTTP 504 狀態碼。請先查看這些問題。如果可以正常存取，請探索應用程式延遲和伺服器逾時，以協助您找出問題並進行修正。

主題

- 在原始伺服器上設定防火牆以允許 CloudFront 流量
- 設定原始伺服器上的安全群組以允許 CloudFront 流量
- 設定您的自訂原始伺服器可從網際網路存取
- 尋找和修正原始伺服器上應用程式的延遲回應

在原始伺服器上設定防火牆以允許 CloudFront 流量

如果原始伺服器上的防火牆封鎖了 CloudFront 流量，請 CloudFront 傳回 HTTP 504 狀態碼，因此在檢查其他問題之前，最好先確認問題不是問題所在。

用於判斷防火牆是否有問題的方法，將依原始伺服器使用的系統類型而定：

- 如果是在 Linux 伺服器使用 IPTable 防火牆，則您可以搜尋工具和資訊來協助您處理 IPTable。
- 如果您在 Windows 伺服器上使用 Windows 防火牆，請參閱 Microsoft 文件中的新增或編輯防火牆規則 (英文)。

當您在原始伺服器上評估防火牆組態時，請根據發佈的 IP 位址範圍尋找封鎖來自 CloudFront 邊緣位置之流量的任何防火牆或安全規則。

如果允許 CloudFront IP 位址範圍連線到原始伺服器，請務必更新伺服器的安全規則以納入變更。您可以訂閱 Amazon SNS 主題，即可在 IP 地址範圍檔案更新時收到通知。在收到通知後，您可以使用程式碼來擷取檔案及進行剖析，並為您的本機環境進行調整。如需詳細資訊，請參閱 AWS 新聞部落格上的透過 Amazon SNS 訂閱 AWS 公用 IP 位址變更。

設定原始伺服器上的安全群組以允許 CloudFront 流量

如果您的來源使用 Elastic Load Balancing，請檢閱 ELB 安全群組，並確定安全群組允許來自 CloudFront的輸入流量。

您也可以使用自動 AWS Lambda 更新安全性群組，以允許來自的輸入流量 CloudFront。

設定您的自訂原始伺服器可從網際網路存取

如果 CloudFront 無法存取您的自訂原始伺服器，因為它在網際網路上沒有公開，則會 CloudFront 傳回 HTTP 504 錯誤。

CloudFront 邊緣位置透過網際網路連線到原始伺服器。如果您的自訂來源位於私人網路上，則 CloudFront 無法連線。因此，您無法使用私有 CloudFront伺服器 (包括內部傳統負載平衡器) 做為.

若要檢查網際網路流量是否可以連線到您的原始伺服器，請執行下列指令 (其中 OriginDomainName 是伺服器的網域名稱)：

檢查 HTTPS 流量時：

- 北卡羅來納-ZV 443 OriginDomainName
- 遠程電話 OriginDomainName

檢查 HTTP 流量時：

- 北卡羅來納-ZV 80 OriginDomainName
- 遠程電話 OriginDomainName

尋找和修正原始伺服器上應用程式的延遲回應

導致伺服器逾時發生的原因通常是等候應用程式回應的時間過長，或是設定的逾時值過短。

幫助避免 HTTP 504 錯誤的快速修復方法是簡單地為您的發行版設置更高的 CloudFront超時值。但是，我們建議您先排除任何與應用程式和原始伺服器有關的效能和延遲問題。接著您可以設定合理的逾時值，協助避免 HTTP 504 錯誤的發生，並且正確回應使用者。

以下是找出效能問題並加以更正之步驟的概觀：

1. 測量 Web 應用程式的一般負載和高負載延遲 (回應能力)。
2. 視需要新增額外的資源，例如 CPU 記憶體。採取其他步驟來解決問題，例如調校資料庫查詢以配合高負載情況。
3. 如果需要，請調整 CloudFront 發行版的逾時值。

以下是每個步驟的詳細資訊。

測量一般負載和高負載延遲

若要判斷一台或多台後端 Web 應用程式伺服器是否發生高度延遲，請在每台伺服器上執行下列 Linux curl 命令：

```
curl -w "Connect time: %{time_connect} Time to first byte:
%{time_starttransfer} Total time: %{time_total} \n" -o /dev/null https://
www.example.com/yourobject
```

> **Note**
>
> 如果伺服器執行 Windows，您可以搜尋和下載適用於 Windows 的 Curl，執行類似的命令。

在測量和評估伺服器中執行應用程式的延遲情況時，請注意以下資訊：

- 延遲值會因應每個應用程式而有不同。不過，相對於幾秒鐘或更久時間，幾毫秒的第一個位元組的時間才算正常。
- 如果在一般負載時測出的應用程式延遲時間正常，這時應注意檢視器在高負載情況下仍然可能遇到逾時問題。當出現高需求量時，伺服器可能會延遲回應或毫無回應。為了協助防止高負載延遲問題，請檢查您的伺服器的資源，例如 CPU、記憶體和磁碟讀取和寫入，確保您的伺服器有足夠容量可因應高負載進行擴展。

  您可以執行以下 Linux 命令，檢查 Apache 程序所使用的記憶體：

  ```
  watch -n 1 "echo -n 'Apache Processes: ' && ps -C apache2 --no-headers |
  wc -l && free -m"
  ```
- 伺服器的高 CPU 使用率可能大幅降低應用程式的效能。如果您將 Amazon EC2 執行個體用於後端伺服器，請檢閱伺服器的 CloudWatch 指標以檢查 CPU 使用率。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。若是使用您自己的伺服器，請參閱伺服器說明文件，以取得如何檢查 CPU 使用率的指示。
- 請檢查高負載時的其他潛在問題，例如，若有大量請求時，資料庫查詢的執行速度就會變慢。

新增資源，並調校伺服器和資料庫

在評估過應用程式和伺服器的回應能力後，確保您有足夠的資源可供一般流量和高負載情況使用：

- 如果您有自己的伺服器，請根據您的評估，確定伺服器具備足夠的 CPU、記憶體和磁碟空間來處理檢視器的請求。

- 如果使用 Amazon EC2 執行個體做為後端伺服器，請確保該執行個體類型具備執行傳入請求的適當資源。如需細資訊，請參閱 Amazon EC2 使用者指南中的執行個體類型。

此外，請考慮以下調校步驟，以避免發生逾時：

- 如果 Curl 命令傳回的 Time to First Byte 值看似很高，請採取適當步驟以提升應用程式的效能。提升應用程式回應能力將有助於減少逾時錯誤。
- 調校資料庫查詢以確保其可處理大量請求，且不會降低效能。
- 在您的後端伺服器上設定 keep-alive (persistent) (保持活動 (持續)) 連線。當伺服器必須為後續請求或使用者重新建立連線時，此選項可避免此時發生延遲。
- 如果您使用 ELB 做為原始伺服器，請參閱以下知識中心文章：如何在我的 ELB Classic Load Balancer 上疑難排解高延遲？，了解如何減少延遲問題。

如有需要，請調整 CloudFront 逾時值

如果已評估並解決應用程式效能緩慢問題、原始伺服器容量和其他問題，但檢視器仍然遇到 HTTP 504 錯誤，這時您應該考慮變更在分佈中的原始伺服器回應逾時指定時間。如需進一步了解，請參閱the section called “回應逾時 (僅限自訂原始伺服器)”。

負載測試 CloudFront

傳統的負載測試方法不能很好地運作， CloudFront 因為 CloudFront 使用 DNS 來平衡分散各地的邊緣位置和每個節點內的負載。當用戶端向其要求內容時 CloudFront，用戶端會收到包含一組 IP 位址的 DNS 回應。如果您只將要求傳送至 DNS 傳回的其中一個 IP 位址進行測試，則只測試一個 CloudFront 節點中的一小部分資源，而這些資源並不能準確地呈現實際的流量模式。視所要求的資料量而定，以這種方式進行測試可能會超載並降低該小部分 CloudFront 伺服器的效能。

CloudFront 旨在為跨多個地理區域具有不同客戶端 IP 地址和不同 DNS 解析器的觀眾進行擴展。若要執行精確評估 CloudFront 效能的負載測試，建議您執行下列所有動作：

- 傳送來自多個地理區域的用戶端請求。
- 設定您的測試，所以每個用戶端能執行獨立的 DNS 請求；每個用戶端就會從 DNS 收到不同組的 IP 地址。
- 對於發出要求的每個用戶端，請將您的用戶端要求分散到 DNS 傳回的 IP 位址集，以確保負載分散到 CloudFront 邊緣位置的多個伺服器上。

請注意以下負載測試的限制 CloudFront：

- 不允許在具有 Lambda@Edge 檢視者請求或檢視者回應觸發程式的快取行為上執行負載測試。
- 不允許在啟用了 Origin Shield 的原始伺服器上執行負載測試。

請求和回應行為

以下各節說明如何 CloudFront 處理檢視者請求並將請求轉送到 Amazon S3 或自訂來源，以及如何 CloudFront 處理來自您來源的回應，包括CloudFront 程序和快取 4xx 和 5xx HTTP 狀態碼的方式。

主題

Amazon S3 原始伺服器之請求和回應行為

主題

如何 CloudFront 處理 HTTP 和 HTTPS 請求

對於 Amazon S3 來源，預設會針對CloudFront 分發中的物件 CloudFront 接受 HTTP 和 HTTPS 通訊協定中的請求。 CloudFront 然後使用提出請求的相同通訊協定，將請求轉送至 Amazon S3 儲存貯體。

對於自訂來源，當您建立發行版時，您可以指定如何 CloudFront 存取原點：僅限 HTTP，或符合檢視器使用的通訊協定。如需有關如何 CloudFront 處理自訂來源的 HTTP 和 HTTPS 要求的詳細資訊，請參閱通訊協定。

如需有關如何限制分佈，讓最終使用者使用 HTTPS 只能存取物件的詳細資訊，請參閱搭配使用 HTTPS CloudFront。

> **Note**
>
> HTTPS 請求的費用高過於 HTTP 請求的費用。如需有關計費費率的詳細資訊，請前往定CloudFront 價方案。

如何 CloudFront 處理和轉送請求到您的 Amazon S3 來源

本主題包含有關如何 CloudFront 處理檢視器請求以及將請求轉寄至 Amazon S3 來源的相關資訊。

主題

快取持續時間和最短 TTL

要控制對象在將另一個請求 CloudFront 轉發到您的來源之前保留在 CloudFront 緩存中的時間，您可以：

- 設定原始伺服器在每個物件中新增 `Cache-Control` 或 `Expires` 標頭欄位。

- 指定 CloudFront 快取行為中的最小 TTL 值。
- 使用預設值為 24 小時。

如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

用戶端 IP 地址

如果檢視器傳送要求至 CloudFront 且不包含`X-Forwarded-For`要求標頭，請從 TCP 連線 CloudFront 取得檢視器的 IP 位址、新增包含 IP 位址的`X-Forwarded-For`標頭，並將要求轉寄至原始位址。例如，如果`192.0.2.2`從 TCP 連接 CloudFront 獲取 IP 地址，它會將以下標頭轉發到原點：

```
X-Forwarded-For: 192.0.2.2
```

如果檢視器將要求傳送至 CloudFront 並包含`X-Forwarded-For`要求標頭，則會從 TCP 連線 CloudFront 取得檢視器的 IP 位址，將其附加至`X-Forwarded-For`標頭的結尾，並將要求轉送至原始位置。例如，如果檢視器要求包含`X-Forwarded-For: 192.0.2.4,192.0.2.3`並`192.0.2.2`從 TCP 連線 CloudFront 取得 IP 位址，則會將下列標頭轉送至原點：

```
X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2
```

> **Note**
>
> `X-Forwarded-For` 標頭包含 IPv4 地址 (如 192.0.2.44) 和 IPv6 地址 (如 2001:0db8:85a3::8a2e:0370:7334)。

有條件的 GET

當 CloudFront 收到從邊緣快取到期的物件請求時，會將請求轉送至 Amazon S3 原始伺服器，以取得物件的最新版本，或從 Amazon S3 取得 CloudFront 邊緣快取已經具有最新版本的確認。Amazon S3 最初將物件傳送到時 CloudFront，它會在回應中包含一個`ETag``LastModified`值和一個值。在 CloudFront 轉寄至 Amazon S3 的新請求中，新 CloudFront 增下列其中一項或兩項：

- 含有已過期版本物件 `If-Match` 值的 `If-None-Match` 或 `ETag` 標頭。
- 含有已過期版本物件 `If-Modified-Since` 值的 `LastModified` 標頭。

Amazon S3 會使用此資訊來判斷物件是否已更新，因此要將整個物件傳回至 CloudFront 或僅傳回 HTTP 304 狀態碼 (未修改)。

Cookie

Amazon S3 不處理 cookie。如果您設定快取行為以將 Cookie 轉寄至 Amazon S3 來源，請 CloudFront 轉送 Cookie，但 Amazon S3 會忽略它們。所有相同物件的未來請求，無論是否變更 Cookie，透過快取中的現有物件提供。

跨來源資源共享 (CORS)

如果您想 CloudFront 要遵守 Amazon S3 跨來源資源共用設定，請進行設定 CloudFront 以將選取的標頭轉寄到 Amazon S3。如需詳細資訊，請參閱 根據請求標頭快取內容。

包括本文的 GET 請求

如果檢視器GET要求包含主體，則會將 HTTP 狀態碼 403 (禁止) CloudFront 傳回給檢視器。

HTTP 方法

如果您設定 CloudFront 為處理其支援的所有 HTTP 方法，請 CloudFront接受來自檢視者的以下請求，並將它們轉送到您的 Amazon S3 來源：

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

CloudFront 始終緩存響應GET和HEAD請求。您也可以設定 CloudFront 為快取要OPTIONS求的回應。CloudFront 不會快取回應至使用其他方法的要求。

如果您想要使用多部分上傳將物件新增至 Amazon S3 儲存貯體，則必須將CloudFront 來源存取控制 (OAC) 新增至您的分發，並授予 OAC 所需的許可。如需詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。

> **Important**
>
> 如果您設定為 CloudFront 接受並轉寄給 Amazon S3 所有 CloudFront支援的 HTTP 方法，則必須建立 CloudFront 來源存取控制 (OAC) 以限制對 Amazon S3 內容的存取，並將必要的許

可授予 OAC。例如，如果您因為想要使用而設定 CloudFront 為接受和轉寄這些方法PUT，則必須設定 Amazon S3 儲存貯體政策以適當處理DELETE請求，以便檢視者無法刪除您不希望使用的資源。如需詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。

如需關於 Amazon S3 支援操作的詳細資訊，請參閱 Amazon S3 文件。

CloudFront 移除或更新的 HTTP 要求標頭

CloudFront 在將請求轉寄到 Amazon S3 來源之前，移除或更新某些標頭。對於大多數標頭，這種行為與自訂原始伺服器相同。如需 HTTP 要求標頭及其 CloudFront 處理方式的完整清單，請參閱HTTP 請求標頭和 CloudFront 行為 (自訂和 Amazon S3 來源)。

最大請求長度和最大 URL 長度

最大請求長度，包括路徑、查詢字串 (如果有) 和標頭是 20,480 位元組。

CloudFront 從請求構造一個 URL。此 URL 的最大長度為 8192 位元組。

如果要求或 URL 超過這些上限，則會 CloudFront 傳回 HTTP 狀態碼 413「要求實體太大」給檢視器，然後終止與檢視器的 TCP 連線。

OCSP 裝訂

當檢視者提交物件的 HTTPS 要求時， CloudFront 或檢視者必須向憑證授權單位 (CA) 確認該網域的 SSL 憑證尚未撤銷。OCSP 裝訂透過允許驗證憑證並快取 CloudFront 來自 CA 的回應來加速憑證驗證，因此用戶端不需要直接向 CA 驗證憑證。

當 CloudFront 收到相同網域中物件的許多 HTTPS 要求時，OCSP 裝訂的效能改善會更明顯。CloudFront 節點位置中的每個伺服器都必須提交個別的驗證要求。當 CloudFront 收到相同網域的大量 HTTPS 要求時，邊緣位置中的每部伺服器很快就會有來自 CA 的回應，它可以在 SSL 交握中「裝訂」封包；檢視者滿意憑證有效時，就 CloudFront 可以提供要求的物件。如果您的分發在 CloudFront 節點中沒有獲得太多流量，則新請求更有可能被導向到尚未通過 CA 驗證證書的服務器。在這種情況下，檢視器會分別執行驗證步驟，而 CloudFront 伺服器會為物件提供服務。該 CloudFront伺服器也會向 CA 提交驗證要求，因此當下次收到包含相同網域名稱的要求時，就會有來自 CA 的驗證回應。

通訊協定

CloudFront 根據檢視器要求的通訊協定 (HTTP 或 HTTPS)，將 HTTP 或 HTTPS 要求轉寄至原始伺服器。

> **Important**
>
> 如果您的 Amazon S3 儲存貯體設定為網站端點，則無法設定 CloudFront為使用 HTTPS 與原始伺服器通訊，因為 Amazon S3 不支援該組態中的 HTTPS 連線。

查詢字串

您可以設定是否 CloudFront 將查詢字串參數轉寄至 Amazon S3 來源。如需詳細資訊，請參閱 根據查詢字串參數快取內容。

原始伺服器連線逾時和嘗試次數

Origin 連線逾時是嘗試建立與原點的連線時 CloudFront 等待的秒數。

原始連線嘗試次數是 CloudFront 嘗試連線至原點的次數。

這些設定共同決定在容錯移轉至次要原點 (在原始群組的情況下) 或傳回錯誤回應給檢視器之前，CloudFront 嘗試連線到原點的時間長度。根據預設， CloudFront 在嘗試連線至次要原點或傳回錯誤回應之前，會等待 30 秒 (每次嘗試 10 秒)。您可以指定較短的連線逾時、較少的嘗試次數或兩者，以縮短此時間。

如需詳細資訊，請參閱 控制原始伺服器逾時和嘗試次數。

原始伺服器回應逾時

「原始伺服器回應逾時」，也稱為「原始伺服器讀取逾時」或「原始伺服器請求逾時」，適用於以下兩個數值：

- 將要求轉送至來源之後 CloudFront 等待回應的時間量 (以秒為單位)。
- 接收來自來源的回應封包 CloudFront 之後，以及在接收下一個封包之前等待的時間 (秒)。

CloudFront 行為取決於查看器請求的 HTTP 方法：

- `GET`和`HEAD`請求 — 如果來源在 30 秒內沒有回應或停止回應 30 秒，請中 CloudFront 斷連線。如果指定的來源連線嘗試次數超過 1 次，請再次 CloudFront 嘗試取得完整的回應。 CloudFront 嘗試最多 3 次，由原始連線嘗試設定的值決定。如果來源在最後一次嘗試期間 CloudFront 沒有回應，在收到另一個相同來源的內容要求之前，不要再試一次。
- `DELETE`、`OPTIONS`、`PATCH`和`PUT`、和`POST`請求 — 如果來源在 30 秒內沒有回應，請中 CloudFront 斷連線，而不會再次嘗試聯絡來源。用戶端可以視需要重新提交請求。

您無法變更 Amazon S3 原始伺服器 (「非」使用靜態網站託管所設定的 S3 儲存貯體) 的回應逾時。

相同物件之同步請求 (請求折疊)

當 CloudFront 節點位置收到對象的請求，並且該對象不在緩存中或緩存的對象已過期時，請 CloudFront 立即將請求發送到源。但是，如果同一物件有同時要求 (也就是說，如果相同物件 (具有相同快取金鑰) 的其他要求在 CloudFront 收到第一個要求的回應之前抵達邊緣位置，則會在將額外要求轉送至原始位置之前CloudFront 暫停。這個短暫的暫停有助於減少原點的負載。 CloudFront 將原始請求的響應發送到暫停時收到的所有請求。這就是所謂的請求摺疊。在 CloudFront 記錄檔中，第一個要求會在`x-edge-result-type`欄位`Miss`中識別為，而收合的要求會識別為`Hit`. 如需 CloudFront 記錄檔的詳細資訊，請參閱the section called “CloudFront 和邊緣功能記錄”。

CloudFront 只會收合共用快取金鑰的要求。如果其他要求不共用相同的快取金鑰，例如，您設定為根據要求標頭、Cookie 或查詢字串進行快取，則會 CloudFront 將具有唯一快取金鑰的所有要求 CloudFront 轉寄至您的來源。

如果您想要防止所有要求崩潰，您可以使用 Managed 緩存策略`CachingDisabled`，這也可以防止緩存。如需詳細資訊，請參閱 使用受管快取政策。

若您想防止特定物件的請求折疊，您可以將快取行為的最短 TTL 設為 0 並設定原始伺服器傳送 `Cache-Control: private`、`Cache-Control: no-store`、`Cache-Control: no-cache`、`Cache-Control: max-age=0` 或 `Cache-Control: s-maxage=0`。這些設定會增加原始伺服器的負載，並為暫停的同時要求產生額外的延遲，同時 CloudFront 等待第一個要求的回應。

如何 CloudFront 處理來自 Amazon S3 來源的回應

本主題包含如何 CloudFront 處理 Amazon S3 來源回應的相關資訊。

主題

- 已取消請求
- 可 CloudFront移除或更新的 HTTP 回應標頭
- 可快取檔案大小上限
- 重新引導

已取消請求

如果物件不在邊緣快取中，並且檢視器在從原始物件取 CloudFront 得物件之後終止工作階段 (例如，關閉瀏覽器)，則 CloudFront 不會在邊緣位置快取該物件。

可 CloudFront移除或更新的 HTTP 回應標頭

CloudFront 在將 Amazon S3 來源的回應轉送給檢視器之前，先移除或更新下列標頭欄位：

- `X-Amz-Id-2`
- `X-Amz-Request-Id`
- `Set-Cookie`— 如果您配置 CloudFront 轉發 cookie，它會轉發標`Set-Cookie`題字段到客戶端。如需詳細資訊，請參閱 根據 Cookie 快取內容。
- `Trailer`
- `Transfer-Encoding`— 如果您的 Amazon S3 來源傳回此標頭欄位，請在將回應傳回給檢視器`chunked`之前將值 CloudFront 設定為。
- `Upgrade`
- `Via`— 在對檢視器的回應中將值 CloudFront 設定為下列項目：

  `Via: http-## ##-##.cloudfront.net (CloudFront)`

  例如，該值如下所示：

  `Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)`

可快取檔案大小上限

CloudFront 儲存在快取記憶體中的回應主體大小上限為 50 GB。此包含未指定 `Content-Length` 標頭值的區塊傳輸回應。

您可以使用範圍要求 CloudFront 來要求每個 50 GB 或更小的部分來要求物件，藉此快取大於此大小的物件。 CloudFront緩存這些部分，因為它們每個都是 50 GB 或更小。檢視器擷取物件的所有部分之後，就可以重建原始、較大的物件。如需詳細資訊，請參閱 使用範圍請求快取大物件。

重新引導

您可以設定 Amazon S3 儲存貯體，將所有的請求重新引導到另一個主機名稱，這可以是另一個 Amazon S3 儲存貯體或 HTTP 伺服器。如果您將值區設定為重新導向所有要求，而儲存貯體是 CloudFront發佈的來源，建議您將值區設定為使用 CloudFront 分發的網域名稱 (例如 d111111abcdef8.cloudfront.net) 或與分發相關聯的備用網域名稱 (CNAME)，將所有要求重新導向至分發。否則，瀏覽器請求略過 CloudFront，並直接從新來源提供對象。

> **Note**
>
> 如果您重新引導請求到備用網域名稱，您必須藉由新增 CNAME 記錄，為您的網域更新 DNS 服務。如需詳細資訊，請參閱 新增替代網域名稱 (CNAME) 以使用自訂 URL。

以下是當您設定儲存貯體重新引導所有請求時，發生的情況：

1. 檢視者 (例如，瀏覽器) 要求物件 CloudFront。
2. CloudFront 將請求轉送至作為您分發的來源 Amazon S3 儲存貯體。
3. Amazon S3 傳回 HTTP 狀態碼 301 (永久移動) 以及新的位置。
4. CloudFront 快取重新導向狀態碼和新位置，並將值傳回給檢視器。 CloudFront 不遵循重定向以從新位置獲取對象。
5. 檢視器會針對物件傳送另一個要求，但是這次檢視器會指定它從中取得的新位置 CloudFront：
   - 如果 Amazon S3 儲存貯體使用 CloudFront分發的網域名稱或替代網域名稱將所有 CloudFront 請求重新導向至分發，請從 Amazon S3 儲存貯體或新位置的 HTTP 伺服器請求物件。當新位置傳回物件時，會將它 CloudFront 傳回給檢視器，並將其快取至節點位置。
   - 如果 Amazon S3 儲存貯體將請求重新導向至其他位置，則第二個請求會略過 CloudFront。位於新位置的 Amazon S3 儲存貯體或 HTTP 伺服器會直接將物件傳回給檢視器，因此永遠不會在 CloudFront 邊緣快取中快取物件。

為自訂原始伺服器之請求和回應行為

主題

- 如何 CloudFront 處理和轉發請求到您的自訂來源
- 如何 CloudFront 處理自訂來源的回應

如何 CloudFront 處理和轉發請求到您的自訂來源

本主題包含有關如何 CloudFront 處理檢視者要求，以及如何將請求轉寄至您的自訂來源的資訊。

主題

- 身分驗證
- 快取持續時間和最短 TTL

- 用戶端 IP 地址
- 用戶端 SSL 身分驗證
- 壓縮
- 條件式請求
- Cookie
- 跨來源資源共享 (CORS)
- 加密
- 包括內文的 GET 請求
- HTTP 方法
- HTTP 請求標頭和 CloudFront 行為 (自訂和 Amazon S3 來源)
- HTTP 版本
- 最大請求長度和最大 URL 長度
- OCSP 裝訂
- 持久性連線
- 通訊協定
- 查詢字串
- 原始伺服器連線逾時和嘗試次數
- 原始伺服器回應逾時
- 相同物件之同步請求 (請求折疊)
- User-Agent 標頭

身分驗證

如果您將標`Authorization`頭轉寄到原始伺服器，則可以將原始伺服器設定為要求下列類型的要求的用戶端驗證：

- `DELETE`
- `GET`
- `HEAD`
- `PATCH`
- `PUT`

- POST

針對OPTIONS要求，只有在您使用下列設定時，才能 CloudFront 設定用戶端驗證：

- CloudFront 被配置為將Authorization標題轉發到您的來源
- CloudFront 配置為不緩存對OPTIONS請求的響應

如需詳細資訊，請參閱 配置 CloudFront 轉發標Authorization頭。

您可以使用 HTTP 或 HTTPS 將要求轉寄至原始伺服器。如需詳細資訊，請參閱 搭配使用 HTTPS CloudFront。

快取持續時間和最短 TTL

要控制對象在將另一個請求 CloudFront 轉發到您的來源之前保留在 CloudFront 緩存中的時間，您可以：

- 設定原始伺服器在每個物件中新增 Cache-Control 或 Expires 標頭欄位。
- 指定 CloudFront 快取行為中的最小 TTL 值。
- 使用預設值為 24 小時。

如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

用戶端 IP 地址

如果檢視器傳送要求至 CloudFront 且不包含X-Forwarded-For要求標頭，請從 TCP 連線 CloudFront 取得檢視器的 IP 位址、新增包含 IP 位址的X-Forwarded-For標頭，並將要求轉寄至原始位址。例如，如果192.0.2.2從 TCP 連接 CloudFront 獲取 IP 地址，它會將以下標頭轉發到原點：

```
X-Forwarded-For: 192.0.2.2
```

如果檢視器將要求傳送至 CloudFront 並包含X-Forwarded-For要求標頭，則會從 TCP 連線 CloudFront 取得檢視器的 IP 位址，將其附加至X-Forwarded-For標頭的結尾，並將要求轉送至原始位置。例如，如果檢視器要求包含X-Forwarded-For: 192.0.2.4,192.0.2.3並192.0.2.2從 TCP 連線 CloudFront 取得 IP 位址，則會將下列標頭轉送至原點：

```
X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2
```

某些應用程式 (例如負載平衡器 (包括 Elastic Load Balancing)、Web 應用程式防火牆、反向 Proxy、入侵預防系統和 API Gateway，會將轉寄要求的 CloudFront 邊緣伺服器 IP 位址附加到標頭的`X-Forwarded-For`結尾。例如，如果 CloudFront 包含`X-Forwarded-For: 192.0.2.2`在其轉發給 ELB 的請求中，並且 CloudFront 邊緣伺服器的 IP 地址是 192.0.2.199，則 EC2 執行個體收到的請求會包含以下標頭：

```
X-Forwarded-For: 192.0.2.2,192.0.2.199
```

> **Note**
>
> `X-Forwarded-For` 標頭包含 IPv4 地址 (如 192.0.2.44) 和 IPv6 地址 (如 2001:0db8:85a3::8a2e:0370:7334)。
> 另請注意，當前服務器（CloudFront）路徑上的每個節點都可以修改`X-Forwarded-For`頭文件。如需詳細資訊，請參閱 RFC 7239 的第 8.1 節。您也可以使用 CloudFront 邊緣計算函數修改標頭。

用戶端 SSL 身分驗證

CloudFront 不支援使用用戶端 SSL 憑證進行用戶端驗證。如果來源要求用戶端憑證，請 CloudFront 捨棄要求。

壓縮

如需詳細資訊，請參閱 提供壓縮檔案。

條件式請求

當 CloudFront 收到已從 Edge 快取到期之物件的要求時，會將要求轉送至原始位置，以取得物件的最新版本，或從來源取得 CloudFront 邊緣快取已經具有最新版本的確認。通常，當 origin 最後一次將對象發送到時 CloudFront，它在響應中包含一個`ETag``LastModified`值，一個值或兩個值。在 CloudFront 轉寄至來源的新要求中，新 CloudFront 增下列其中一項或兩項：

- 含有已過期版本物件 `If-Match` 值的 `If-None-Match` 或 `ETag` 標頭。
- 含有已過期版本物件 `If-Modified-Since` 值的 `LastModified` 標頭。

來源會使用此資訊來判斷物件是否已更新，因此要將整個物件傳回至 CloudFront 或只傳回 HTTP 304 狀態碼 (未修改)。

> **Note**
>
> `If-Modified-Since`當設定為轉寄 Cookie (全部或子集) 時 CloudFront ，則不支援`If-None-Match`條件式要求。
> 如需詳細資訊，請參閱 根據 Cookie 快取內容。

Cookie

您可以設定 CloudFront 將 Cookie 轉寄至您的來源。如需詳細資訊，請參閱 根據 Cookie 快取內容。

跨來源資源共享 (CORS)

如果您想 CloudFront 要遵守跨來源資源共用設定，請進行設定，將`Origin`標頭轉寄 CloudFront 至您的來源。如需詳細資訊，請參閱 根據請求標頭快取內容。

加密

您可以要求檢視者使用 HTTPS 將請求傳送至您的自訂原始伺服器， CloudFront 並 CloudFront 要求使用檢視者使用的通訊協定將要求轉寄至您的自訂原始伺服器。如需詳細資訊，請參閱下列分佈設定：

- 檢視器通訊協定政策
- 通訊協定 (僅限自訂原始伺服器)

CloudFront 使用 SSLv3、TLSv1.0、TLSv1.1 和 TLSv1.2 通訊協定，將 HTTPS 要求轉寄至原始伺服器。對於自訂來源，您可以選擇與來源通訊時CloudFront 要使用的 SSL 通訊協定：

- 如果您使用的是 CloudFront 主控台，請使用「Origin SSL 通訊協定」核取方塊來選擇通訊協定。如需詳細資訊，請參閱 建立分發。
- 如果您使用的是 CloudFront API，請使用`OriginSslProtocols`元素指定通訊協定。如需詳細資訊 OriginSslProtocols，請參閱 Amazon CloudFront API 參考DistributionConfig中的和。

如果來源是 Amazon S3 存儲桶，請 CloudFront 始終使用 TLSv1.2。

> **Important**
>
> 其他 SSL 和 TLS 的版本不支援。

如需搭配使用 HTTPS 的詳細資訊 CloudFront，請參閱搭配使用 HTTPS CloudFront。如需在檢視者之間以及您的來源之間以及 CloudFront您的來源之間 CloudFront 支援 HTTPS 通訊的密碼清單，請參閱。 CloudFront 檢視器與之間支援的通訊協定和密碼 CloudFront

包括內文的 GET 請求

如果檢視器GET要求包含主體，則會將 HTTP 狀態碼 403 (禁止) CloudFront 傳回給檢視器。

HTTP 方法

如果您設定 CloudFront 為處理其支援的所有 HTTP 方法，請 CloudFront接受來自檢視者的下列要求，並將它們轉寄至您的自訂來源：

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

CloudFront 始終緩存響應GET和HEAD請求。您也可以設定 CloudFront 為快取要OPTIONS求的回應。CloudFront 不會快取回應至使用其他方法的要求。

如需有關設定您的自訂原始伺服器是否處理這些方法的詳細資訊，請參閱您的原始伺服器的文件。

> **Important**
>
> 如果您設定 CloudFront 為接受並轉送至原始伺服器，請將原始伺服器設定為處理所有方法。CloudFront 例如，如果您因為想要使用而設定 CloudFront 為接受和轉寄這些方法POST，則必須將原始伺服器設定為適當處理要DELETE求，以便檢視者無法刪除您不希望使用的資源。如需詳細資訊，請參閱您的 HTTP 伺服器文件。

HTTP 請求標頭和 CloudFront 行為 (自訂和 Amazon S3 來源)

下表列出可以轉送到自訂和 Amazon S3 原始伺服器的 HTTP 請求標頭 (已指明例外狀況)。對於每個標頭，該表包含下列資訊：

- CloudFront 如果您沒有配置 CloudFront 將標題轉發到您的來源，這會導 CloudFront 致根據標題值緩存對象的行為。
- 您是否可以配置 CloudFront 為基於該頭的標題值緩存對象。

  您可以設定 CloudFront 為根據`Date`和`User-Agent`標頭中的值快取物件，但我們不建議這麼做。這些標頭具有許多可能的值，並且基於其值的緩存將導致 CloudFront 向您的來源轉發更多請求。

如需有關根據標頭值快取的詳細資訊，請參閱根據請求標頭快取內容。

標頭	如果您沒有配置 CloudFront 為基於標題值緩存的行為	支援根據標頭值進行快取
其他定義的標頭	舊版快取設定 — CloudFront 將標頭轉寄至您的來源。	是
`Accept`	CloudFront 刪除標題。	是
`Accept-Charset`	CloudFront 刪除標題。	是
`Accept-Encoding`	如果該值包含`gzip`或`br`，則將標準化`Accept-Encoding` 標題 CloudFront 轉發到您的原點。 如需詳細資訊，請參閱 壓縮支援 及 提供壓縮檔案。	是
`Accept-Language`	CloudFront 刪除標題。	是
`Authorization`	• `GET`和 request — 在將`HEAD`請求轉發到您的來源之前 CloudFront 刪除`Authorization` 標頭字段。 • `OPTIONS`request — 如果您配置為緩存請求的響應，則在將`OPTIONS`請求轉發 CloudFront 到來	是

標頭	如果您沒有配置 CloudFront 為基於標題值緩存的行為	支援根據標頭值進行快取
	源之前 CloudFront刪除`Authorization` 標頭字段。 CloudFront 如果您未配置緩存 OPTIONS 請求的響應，則CloudFront 將`Authorization` 頭字段轉發到您的來源。 • `DELETE`、`PATCH`、`POST`、和 request — 在將`PUT`請求轉寄至您的來源之前， CloudFront 不會移除標頭欄位。	
`Cache-Control`	CloudFront 將標題轉發到您的來源。	否
`CloudFront-Forwarded-Proto`	CloudFront 在將請求轉寄至您的來源之前，不會新增標頭。 如需詳細資訊，請參閱「根據請求的通訊協定設定快取」。	是
`CloudFront-Is-Desktop-Viewer`	CloudFront 在將請求轉寄至您的來源之前，不會新增標頭。 如需詳細資訊，請參閱「根據裝置類型設定快取」。	是
`CloudFront-Is-Mobile-Viewer`	CloudFront 在將請求轉寄至您的來源之前，不會新增標頭。 如需詳細資訊，請參閱「根據裝置類型設定快取」。	是

標頭	如果您沒有配置 CloudFront 為基於標題值緩存的行為	支援根據標頭值進行快取
`CloudFront-Is-Tablet-Viewer`	CloudFront 在將請求轉寄至您的來源之前，不會新增標頭。 如需詳細資訊，請參閱「根據裝置類型設定快取」。	是
`CloudFront-Viewer-Country`	CloudFront 在將請求轉寄至您的來源之前，不會新增標頭。	是
`Connection`	CloudFront 在將請求轉發到您的來源`Connection: Keep-Alive` 之前替換此標頭。	否
`Content-Length`	CloudFront 將標題轉發到您的來源。	否
`Content-MD5`	CloudFront 將標題轉發到您的來源。	是
`Content-Type`	CloudFront 將標題轉發到您的來源。	是
`Cookie`	如果您配置 CloudFront 轉發 cookie，它會將`Cookie`標題字段轉發到您的來源。如果你不這樣做， CloudFront 刪除標`Cookie`題字段。如需詳細資訊，請參閱「根據 Cookie 快取內容」。	否
`Date`	CloudFront 將標題轉發到您的來源。	可以，但不建議
`Expect`	CloudFront 刪除標題。	是

標頭	如果您沒有配置 CloudFront 為基於標題值緩存的行為	支援根據標頭值進行快取
`From`	CloudFront 將標題轉發到您的來源。	是
`Host`	CloudFront 將該值設置為與請求對象相關聯的來源的域名。 您無法根據 Amazon S3 或 MediaStore來源的主機標頭進行快取。	是 (自訂) 否 (S3 和 MediaStore)
`If-Match`	CloudFront 將標題轉發到您的來源。	是
`If-Modified-Since`	CloudFront 將標題轉發到您的來源。	是
`If-None-Match`	CloudFront 將標題轉發到您的來源。	是
`If-Range`	CloudFront 將標題轉發到您的來源。	是
`If-Unmodified-Since`	CloudFront 將標題轉發到您的來源。	是
`Max-Forwards`	CloudFront 將標題轉發到您的來源。	否
`Origin`	CloudFront 將標題轉發到您的來源。	是
`Pragma`	CloudFront 將標題轉發到您的來源。	否
`Proxy-Authenticate`	CloudFront 刪除標題。	否

標頭	如果您沒有配置 CloudFront 為基於標題值緩存的行為	支援根據標頭值進行快取
`Proxy-Authorization`	CloudFront 刪除標題。	否
`Proxy-Connection`	CloudFront 刪除標題。	否
`Range`	CloudFront 將標題轉發到您的來源。如需詳細資訊，請參閱 如何 CloudFront 處理對象的部分請求（範圍 GET）。	是，根據預設。
`Referer`	CloudFront 刪除標題。	是
`Request-Range`	CloudFront 將標題轉發到您的來源。	否
`TE`	CloudFront 刪除標題。	否
`Trailer`	CloudFront 刪除標題。	否
`Transfer-Encoding`	CloudFront 將標題轉發到您的來源。	否
`Upgrade`	CloudFront 刪除標題，除非你已經建立了一個Web Socket 連接。	否 (除了 WebSocket 連線)
`User-Agent`	CloudFront 替換此標題字段的值`Amazon CloudFront` 。如果您 CloudFront 要根據使用者使用的裝置快取內容，請參閱根據裝置類型設定快取。	可以，但不建議

標頭	如果您沒有配置 CloudFront 為基於標題值緩存的行為	支援根據標頭值進行快取
`Via`	CloudFront 將標題轉發到您的來源。	是
`Warning`	CloudFront 將標題轉發到您的來源。	是
`X-Amz-Cf-Id`	CloudFront 在將請求轉發到您的來源之前，將標頭添加到查看者請求中。此標頭值包含可唯一識別請求的加密字串。	否
`X-Edge-*`	CloudFront 刪除所有`X-Edge-*`標題。	否
`X-Forwarded-For`	CloudFront 將標題轉發到您的來源。如需詳細資訊，請參閱「用戶端 IP 地址」。	是
`X-Forwarded-Proto`	CloudFront 刪除標題。	否
`X-HTTP-Method-Override`	CloudFront 刪除標題。	是
`X-Real-IP`	CloudFront 刪除標題。	否

HTTP 版本

CloudFront 使用 HTTP/1.1 將請求轉發到您的自定義源。

最大請求長度和最大 URL 長度

最大請求長度，包括路徑、查詢字串 (如果有) 和標頭是 20,480 位元組。

CloudFront 從請求構造一個 URL。此 URL 的最大長度為 8192 位元組。

如果要求或 URL 超過這些上限，則會 CloudFront 傳回 HTTP 狀態碼 413「要求實體太大」給檢視器，然後終止與檢視器的 TCP 連線。

OCSP 裝訂

當檢視者提交物件的 HTTPS 要求時， CloudFront 或檢視者必須向憑證授權單位 (CA) 確認該網域的 SSL 憑證尚未撤銷。OCSP 裝訂透過允許驗證憑證並快取 CloudFront 來自 CA 的回應來加速憑證驗證，因此用戶端不需要直接向 CA 驗證憑證。

當收到相同網域中物件的大量 HTTPS 要求時，OCSP 裝訂的效能改善 CloudFront 會更明顯。CloudFront節點位置中的每個伺服器都必須提交個別的驗證要求。當 CloudFront 收到相同網域的大量 HTTPS 要求時，邊緣位置中的每部伺服器很快就會有來自 CA 的回應，它可以在 SSL 交握中「裝訂」封包；檢視者滿意憑證有效時，就 CloudFront 可以提供要求的物件。如果您的分發在CloudFront 節點中沒有獲得太多流量，則新請求更有可能被導向到尚未通過 CA 驗證證書的服務器。在這種情況下，檢視器會分別執行驗證步驟，而 CloudFront 伺服器會為物件提供服務。該 CloudFront 伺服器也會向 CA 提交驗證要求，因此當下次收到包含相同網域名稱的要求時，就會有來自 CA 的驗證回應。

持久性連線

當從您的來源 CloudFront 獲取響應時，它會嘗試保持連接幾秒鐘，以防另一個請求在該期間到達。維護持久性連線可節省重新建立 TCP 連線所需的時間，並為後續請求執行另一個 TLS 交握。

如需包括如何設定持續連線時間的詳細資訊，請參閱保持連線逾時 (僅限自訂原始伺服器)一節的發佈設定參考。

通訊協定

CloudFront 根據下列項目，將 HTTP 或 HTTPS 要求轉寄至原始伺服器：

- 檢視器傳送至 CloudFront的要求通訊協定 (HTTP 或 HTTPS)。
- CloudFront主控台中「原始通訊協定原則」欄位的值，如果您使用的是 CloudFront API，則為`DistributionConfig`複雜類型中的`OriginProtocolPolicy`元素。在 CloudFront 主控台中，選項包括「僅限 HTTP」、「僅限 HTTPS」和「比對檢視器」。

如果您指定「僅限 HTTP」或「僅限 HTTPS」，則無 CloudFront論檢視器要求中的通訊協定為何，都會使用指定的通訊協定將要求轉送至原始伺服器。

如果您指定「比對檢視器」，請使用檢視器要求中的通訊協定將要求 CloudFront 轉寄至原始伺服器。請注意，即使檢視者同時使用 HTTP 和 HTTPS 通訊協定發出要求，也只會 CloudFront 快取物件一次。

> **Important**
>
> 如果使用 HTTPS 通訊協定將要求 CloudFront 轉寄至原始伺服器，而且原始伺服器傳回無效的憑證或自我簽署憑證，則會中 CloudFront 斷 TCP 連線。

如需如何使用 CloudFront 主控台更新發行版的詳細資訊，請參閱更新分佈。如需有關如何使用 CloudFront API 更新分發的詳細資訊，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

查詢字串

您可以配置是否將查詢字符串參數 CloudFront 轉發到您的來源。如需詳細資訊，請參閱 根據查詢字串參數快取內容。

原始伺服器連線逾時和嘗試次數

Origin 連線逾時是嘗試建立與原點的連線時 CloudFront 等待的秒數。

原始連線嘗試次數是 CloudFront 嘗試連線至原點的次數。

這些設定共同決定在容錯移轉至次要原點 (在原始群組的情況下) 或傳回錯誤回應給檢視器之前，CloudFront 嘗試連線到原點的時間長度。根據預設，CloudFront 在嘗試連線至次要原點或傳回錯誤回應之前，會等待 30 秒 (每次嘗試 10 秒)。您可以指定較短的連線逾時、較少的嘗試次數或兩者，以縮短此時間。

如需詳細資訊，請參閱 控制原始伺服器逾時和嘗試次數。

原始伺服器回應逾時

「原始伺服器回應逾時」，也稱為「原始伺服器讀取逾時」或「原始伺服器請求逾時」，適用於以下兩個數值：

- 將要求轉送至來源之後 CloudFront 等待回應的時間量 (以秒為單位)。
- 接收來自來源的回應封包 CloudFront 之後，以及在接收下一個封包之前等待的時間 (秒)。

CloudFront 行為取決於查看器請求的 HTTP 方法：

- GET和HEAD請求-如果來源沒有響應或在響應超時期間內停止響應，則中 CloudFront 斷連接。如果指定的來源連線嘗試次數超過 1 次，請再次 CloudFront 嘗試取得完整的回應。 CloudFront 嘗試最

多 3 次，由原始連線嘗試設定的值決定。如果來源在最後一次嘗試期間 CloudFront 沒有回應，在收到另一個相同來源的內容要求之前，不要再試一次。

- DELETE、OPTIONS、PATCHPUT、和POST請求 — 如果來源在 30 秒內沒有回應，請中 CloudFront 斷連線，而不會再次嘗試聯絡來源。用戶端可以視需要重新提交請求。

如需詳細資訊，包括如何設定原始伺服器回應逾時，請參閱回應逾時 (僅限自訂原始伺服器)。

相同物件之同步請求 (請求折疊)

當 CloudFront 節點位置收到對象的請求，並且該對象不在緩存中或緩存的對象已過期時，請 CloudFront 立即將請求發送到源。但是，如果同一物件有同時要求 (也就是說，如果相同物件 (具有相同快取金鑰) 的其他要求在 CloudFront 收到第一個要求的回應之前抵達邊緣位置，則會在將額外要求轉送至原始位置之前CloudFront 暫停。這個短暫的暫停有助於減少原點的負載。 CloudFront 將原始請求的響應發送到暫停時收到的所有請求。這就是所謂的請求摺疊。在 CloudFront 記錄檔中，第一個要求會在x-edge-result-type欄位Miss中識別為，而收合的要求會識別為Hit. 如需 CloudFront 記錄檔的詳細資訊，請參閱the section called “CloudFront 和邊緣功能記錄”。

CloudFront 只會收合共用快取金鑰的要求。如果其他要求不共用相同的快取金鑰，例如，您設定為根據要求標頭、Cookie 或查詢字串進行快取，則會 CloudFront 將具有唯一快取金鑰的所有要求 CloudFront 轉寄至您的來源。

如果您想要防止所有要求崩潰，您可以使用 Managed 緩存策略CachingDisabled，這也可以防止緩存。如需詳細資訊，請參閱 使用受管快取政策。

若您想防止特定物件的請求折疊，您可以將快取行為的最短 TTL 設為 0 並設定原始伺服器傳送 Cache-Control: private、Cache-Control: no-store、Cache-Control: no-cache、Cache-Control: max-age=0 或 Cache-Control: s-maxage=0。這些設定會增加原始伺服器的負載，並為暫停的同時要求產生額外的延遲，同時 CloudFront 等待第一個要求的回應。

User-Agent 標頭

如果您想 CloudFront 要根據使用者用來檢視內容的裝置快取不同版本的物件，建議您設定為將下列一或多個標頭轉寄 CloudFront 至您的自訂來源：

- CloudFront-Is-Desktop-Viewer
- CloudFront-Is-Mobile-Viewer
- CloudFront-Is-SmartTV-Viewer
- CloudFront-Is-Tablet-Viewer

根據標User-Agent頭的值，將這些標頭的值 CloudFront 設置為true或轉發請求到您的來源false之前。如果裝置屬於多個類別，一個以上的值可能是 true。例如，對於某些平板電腦裝置， CloudFront 可能會同時CloudFront-Is-Tablet-Viewer將CloudFront-Is-Mobile-Viewer和設定為true。如需有關根據要求標頭設定 CloudFront 快取的詳細資訊，請參閱根據請求標頭快取內容。

您可以配置 CloudFront 為基於User-Agent標頭中的值緩存對象，但我們不建議這樣做。標User-Agent頭有許多可能的值，並且基於這些值的緩存將導 CloudFront 致將更多請求轉發到您的來源。

如果您沒有設定 CloudFront 為根據User-Agent標頭中的值快取物件，請在 CloudFront 將要求轉送至來源之前，新增具有下列值的User-Agent標頭：

```
User-Agent = Amazon CloudFront
```

CloudFront 無論來自檢視器的要求是否包含標頭，都會新增此User-Agent標頭。如果來自檢視器的要求包含標User-Agent頭，請將其 CloudFront 移除。

如何 CloudFront 處理自訂來源的回應

本主題包含有關如何 CloudFront 處理自訂來源回應的資訊。

主題

- 100 Continue 回應
- 快取
- 已取消請求
- 內容議價
- Cookie
- 捨棄 TCP 連線
- CloudFront 移除或取代的 HTTP 回應標頭
- 可快取檔案大小上限
- 原始伺服器無法使用
- 重新引導
- Transfer-Encoding 標頭

`100 Continue` 回應

您的來源無法傳送超過一個 100 繼續回覆給 CloudFront。在第一個 100-繼續回應之後， CloudFront 需要一個 HTTP 200 確定回應。如果您的來源在第一個響應之後發送另一個 100-Continue 響應， CloudFront 將返回錯誤。

快取

- 確保原始伺服器集為有效且為 `Date` 和 `Last-Modified` 標頭欄位準確的值。
- CloudFront 通常在來自原點的響應中尊重`Cache-Control: no-cache`標題。如需例外，請參閱相同物件之同步請求 (請求折疊)。

已取消請求

如果物件不在邊緣快取中，並且檢視器在從原始物件取 CloudFront 得物件之後終止工作階段 (例如，關閉瀏覽器)，則 CloudFront 不會在邊緣位置快取該物件。

內容議價

如果您的來源`Vary:*`在回應中傳回，且對應快取行為的「最小 TTL」值為 0，則會快取物件， CloudFront 但仍會將物件的每個後續要求轉送至原始位置，以確認快取包含物件的最新版本。CloudFront 不包含任何條件標頭，例如`If-None-Match`或`If-Modified-Since`。因此，您的 origin 會將物件傳回至以 CloudFront 回應每個要求。

如果您的來源`Vary:*`在回應中傳回，且對應快取行為的「最小 TTL」值是任何其他值，則會依照中 CloudFront 移除或取代的 HTTP 回應標頭所述 CloudFront 處理`Vary`標頭。

Cookie

如果您為快取行為啟用 Cookie，並且來源傳回包含物件的 Cookie，則會 CloudFront 快取物件和 Cookie。請注意，這可減少物件的快取能力。如需詳細資訊，請參閱 根據 Cookie 快取內容。

捨棄 TCP 連線

如果您的來源將對象返回到時， CloudFront 和您的來源之間的 TCP 連接中斷 CloudFront，則 CloudFront 行為取決於您的來源是否在響應中包含`Content-Length`頭文件：

- 內容長度標題-將對象 CloudFront 返回給查看器，因為它從您的原點獲取對象。但是，如果標`Content-Length`頭的值與對象的大小不匹配，則 CloudFront不會緩存該對象。

- 傳輸編碼：分塊 — 從您的來源取得物件時，將物件 CloudFront 傳回給檢視器。但是，如果區塊回應不完整，則 CloudFront 不會快取物件。
- 沒有 Content-Length 標頭-將對象 CloudFront 返回給查看器並緩存它，但對象可能不完整。如果沒有`Content-Length`標頭， CloudFront 則無法確定 TCP 連接是意外還是故意中斷。

我們建議您將 HTTP 伺服器設定為新增`Content-Length`標頭，以防 CloudFront 止快取部分物件。

CloudFront 移除或取代的 HTTP 回應標頭

CloudFront 在將來源的響應轉發給查看器之前，刪除或更新以下標題字段：

- `Set-Cookie`— 如果您配置 CloudFront 轉發 cookie，它會轉發標`Set-Cookie`題字段到客戶端。如需詳細資訊，請參閱 根據 Cookie 快取內容。
- `Trailer`
- `Transfer-Encoding`— 如果您的 origin 傳回此標頭欄位，請在將回應傳回給檢視器`chunked`之前將值CloudFront 設定為。
- `Upgrade`
- `Vary` – 請注意以下各項：
  - 如果您設定 CloudFront 將任何裝置特定的標頭轉寄至您的來源 (`CloudFront-Is-Desktop-Viewer`、、`CloudFront-Is-Mobile-Viewer`、`CloudFront-Is-Tablet-Viewer`)`CloudFront-Is-SmartTV-Viewer`，並將原點設定為 CloudFront 返回 CloudFront，則會返回`Vary:User-Agent``Vary:User-Agent`給檢視器。如需詳細資訊，請參閱 根據裝置類型設定快取。
  - 如果您將來源配置為包含`Accept-Encoding`或`Cookie`在`Vary`標題中，請在對查看器的響應中 CloudFront 包含這些值。
  - 如果您配置為將標題轉發 CloudFront 到您的來源，並且如果將 origin 配置為在標頭 CloudFront 中返回標`Vary`頭名稱（例如，`Vary:Accept-Charset,Accept-Language`），則將帶有這些值的標`Vary`頭 CloudFront 返回給查看器。
  - 如需如何 CloudFront 處理`Vary`標頭*中的值的資訊，請參閱內容議價。
  - 如果您將 origin 設定為在`Vary`標頭中包含任何其他值，請先 CloudFront 移除這些值，然後再將回應傳回給檢視器。
- `Via`— 在對檢視器的回應中將值 CloudFront 設定為下列項目：

  `Via: http-## ##-##.cloudfront.net (CloudFront)`

  例如，該值如下所示：

```
Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)
```

可快取檔案大小上限

CloudFront 儲存在快取記憶體中的回應主體大小上限為 50 GB。此包含未指定 `Content-Length` 標頭值的區塊傳輸回應。

您可以使用範圍要求 CloudFront 來要求每個 50 GB 或更小的部分來要求物件，藉此快取大於此大小的物件。 CloudFront緩存這些部分，因為它們每個都是 50 GB 或更小。檢視器擷取物件的所有部分之後，就可以重建原始、較大的物件。如需詳細資訊，請參閱 使用範圍請求快取大物件。

原始伺服器無法使用

如果您的原始伺服器無法使用，而且取 CloudFront 得要求位於 Edge 快取中但已過期的物件 (例如，因為指`Cache-Control max-age`令中指定的期間已過)， CloudFront 則可以提供物件的過期版本或提供自訂錯誤頁面。如需設定自訂錯誤頁面時CloudFront 行為的詳細資訊，請參閱設定自訂錯誤頁面時如何 CloudFront 處理錯誤。

在某些情況下，很少要求的物件會被逐出，而且邊緣快取中不再可用。 CloudFront 不能為已被驅逐的對象提供服務。

重新引導

如果您在原始伺服器中變更物件的位置，您可以設定您的 Web 伺服器重新引導請求至新的位置。設定重新導向之後，檢視者第一次提交物件的要求時， CloudFront Front 會將要求傳送至原點，而來源會以重新導向回應 (例如，`302 Moved Temporarily`)。 CloudFront 緩存重定向並將其返回給查看器。 CloudFront 不遵循重定向。

您可以設定您的 Web 伺服器重新引導請求以下其中一個位置：

- 在原始伺服器中物件的新 URL。當檢視器遵循重新導向至新 URL 時，檢視者會略過 CloudFront 並直接前往原點。因此，我們建議您不要重新引導請求到原始伺服器中物件的新 URL。
- 物件的新 CloudFront URL。當檢視者提交包含新 CloudFront URL 的要求時，從原始位置的新位置取 CloudFront 得物件，在節點位置快取該物件，然後將物件傳回給檢視器。物件的後續請求會被節點提供。這可避免與從原始伺服器檢視器請求的物件有關的延遲和負載。但是，對象的每個新請求都會產生兩個請求的費用 CloudFront。

`Transfer-Encoding` 標頭

CloudFront 僅支援`Transfer-Encoding`標頭的`chunked`值。如果原點傳 CloudFront 回`Transfer-Encoding: chunked`，則會在邊緣位置接收物件時將物件傳回給用戶端，並以區塊格式快取物件以供後續要求使用。

如果檢視器提出要`Range GET`求且 origin 傳回`Transfer-Encoding: chunked`，則會將整個物件 CloudFront 傳回給檢視器，而不是要求的範圍。

如果您無法預定回應內容的長度，我們建議您使用區塊編碼。如需詳細資訊，請參閱 捨棄 TCP 連線。

原始伺服器群組的請求和回應行為

請求原始群組的運作方式與請求未設為原始群組的原始伺服器的運作方式相同，但存在原始伺服器容錯移轉時例外。與任何其他來源一樣，當CloudFront 收到請求且內容已緩存在節點時，內容會從緩存中提供給查看者。當發生命中遺漏時，會將檢視器請求轉送到原始伺服器群組中的主要原始伺服器。

主要原始伺服器的請求和回應行為是相同的，因為它是針對不在原始伺服器群組中的原始伺服器。如需詳細資訊，請參閱 Amazon S3 原始伺服器之請求和回應行為 及 為自訂原始伺服器之請求和回應行為。

以下描述當主要原始伺服器傳回特定 HTTP 狀態碼時，原始伺服器容錯移轉的行為：

- HTTP 2xx 狀態碼（成功）： CloudFront 緩存文件並將其返回給查看器。
- HTTP 3xx 狀態碼（重定向）：將狀態碼 CloudFront 返回給查看器。
- HTTP 4xx 或 5xx 狀態碼 (用戶端/伺服器錯誤)：如果傳回的狀態碼已設定為容錯移轉，則會將相同的要求 CloudFront 傳送至原始群組中的次要原始伺服器。
- HTTP 4xx 或 5xx 狀態碼 (用戶端/伺服器錯誤)：如果傳回的狀態碼尚未設定進行容錯移轉，則會將錯誤傳 CloudFront 回給檢視器。

CloudFront 只有當檢視器要求的 HTTP 方法為`GET`、`HEAD`或時，才容錯移轉至次要原點`OPTIONS`。CloudFront 當檢視器傳送不同的 HTTP 方法 (例如`POST`，等等) 時`PUT`，不會容錯移轉。

當 CloudFront 將請求發送到次要來源時，響應行為與不在CloudFront 原始組中的來源的響應行為相同。

如需原始伺服器群組的詳細資訊，請參閱透過 CloudFront 原始容錯移轉將高可用性。

將自訂標頭新增到原始伺服器請求

您可以設定 CloudFront 為將自訂標頭新增至傳送至原始伺服器的請求。這些自訂標頭可讓您傳送並收集來自原始伺服器的資訊，該資訊無法透過一般檢視器請求取得。甚至可以針對每個原點自訂這些標頭。 CloudFront 支援自訂和 Amazon S3 起源的自訂標頭。

主題

原始伺服器自訂標頭的使用案例

您可以將自訂標頭用於各種用途，例如：

識別要求來源 CloudFront

您可以識別來源接收的請求 CloudFront。如果您想知道用戶是否繞過 CloudFront，或者您正在使用多個 CDN 並且需要有關來自每個 CDN 的請求的信息，此功能非常有用。

> **Note**
>
> 如果您使用的是 Amazon S3 原始伺服器，並且啟用 Amazon S3 伺服器存取記錄，則日誌不包含標頭資訊。

判斷哪些請求是來自特定分發

如果您將多個 CloudFront 發行版設定為使用相同來源，則可以在每個發行版中新增不同的自訂標頭。然後，您可以使用來自原始伺服器的日誌，判斷哪些請求來自哪個 CloudFront 分發。

啟用跨來源資源分享 (CORS)

如果某些檢視器不支援跨來源資源共用 (CORS)，您可以設定為永遠將`Origin`標頭新增 CloudFront 至傳送至原始伺服器的請求。接著，您可以將原始伺服器設定為針對每個請求傳回 `Access-Control-Allow-Origin` 標頭。您還必須 CloudFront 進行配置以遵守 CORS 設置。

控制內容的存取

您可以使用自訂標頭來控制內容的存取。透過將您的來源設定為僅在包含新增的自訂標頭時才回應要求 CloudFront，您可以防止使用者直接在來源略過 CloudFront 和存取您的內容。如需詳細資訊，請參閱 在自訂原始伺服器上限制存取檔案。

設定 CloudFront 為將自訂標頭新增至原始請求

若要將分佈設定為將自訂標頭新增到傳送給您原始伺服器的請求，請使用下列其中一種方法來更新原始伺服器設定：

- CloudFront 控制台 — 當您建立或更新發行版時，請在 Origin 自訂標頭設定中指定標頭名稱和值。如需詳細資訊，請參閱 建立分發 或 更新分佈 。
- CloudFront API — 對於您要新增自訂標頭的每個來源，請在其中的`CustomHeaders`欄位中指定標題名稱和值`Origin`。如需詳細資訊，請參閱 CreateDistribution 或 UpdateDistribution 。

如果您指定的標頭名稱和值尚未出現在檢視器要求中，請 CloudFront 將它們新增至原始請求。如果標頭存在，則在將請求轉發到來源之前 CloudFront 覆蓋標頭值。

如需套用至原始伺服器自訂標頭的配額 (先前稱為限制)，請參閱標頭的配額。

無法新增至原始請求的 CloudFront 自訂標頭

您無法設定 CloudFront 為將下列任何標頭新增至傳送至原始伺服器的請求：

- `Cache-Control`
- `Connection`
- `Content-Length`
- `Cookie`
- `Host`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Range`
- `If-Unmodified-Since`

- Max-Forwards
- Pragma
- Proxy-Authorization
- Proxy-Connection
- Range
- Request-Range
- TE
- Trailer
- Transfer-Encoding
- Upgrade
- Via
- 以 X-Amz- 做為開頭的標頭
- 以 X-Edge- 做為開頭的標頭
- X-Real-Ip

配置 CloudFront 轉發標**Authorization**頭

將查看者請求 CloudFront 轉發到您的來源時，默認情況下 CloudFront 會刪除一些查看器標題，包括標Authorization題。為了確保您的原始伺服器始終收到原始伺服器請求中的 Authorization 標頭，您有以下選項：

- 使用快取政策將 Authorization 標頭新增至快取金鑰。快取金鑰中的所有標頭都會自動包含在原始伺服器請求中。如需詳細資訊，請參閱 控制快取金鑰。
- 使用將所有檢視器標頭轉寄至原始伺服器的原始伺服器請求政策。您無法在原始要求原則中個別轉寄標Authorization頭，但是當您轉寄所有檢視器標頭時，都會在檢視器要求中 CloudFront 包含標Authorization頭。 CloudFront 為此使用案例提供受管理的原始要求原則，稱為 Managed-AllViewer。如需更多詳細資訊，請參閱 使用受管原始伺服器請求政策。

如何 CloudFront 處理對象的部分請求（範圍 GET）

針對大型物件，檢視器 (Web 瀏覽器或用戶端) 可能提出多個 GET 請求，並使用 Range 請求標頭下載分為多個較小部分的物件。這些位元範圍的請求，有時稱為 Range GET 請求，改善部分下載的效率與從部分失敗的傳輸復原。

當 CloudFront 收到Range GET請求時，它會檢查接收請求的邊緣位置中的緩存。如果該邊緣位置中的快取已包含整個物件或物件的要求部分，請 CloudFront 立即從快取提供要求的範圍。

如果緩存不包含請求的範圍，請將請求 CloudFront 轉發到來源。(若要最佳化效能， CloudFront 可能會要求比Range GET. 中要求的用戶端更大的範圍) 接下來，會發生什麼狀況取決於原始伺服器是否支援 Range GET 請求：

- 如果來源支持**Range GET**請求：它返回請求的範圍。 CloudFront 為請求的範圍提供服務，並為 future 的請求緩存它。(如同許多 HTTP 伺服器一樣，Amazon S3 支援 Range GET 請求。)
- 如果來源不支持**Range GET**請求：它返回整個對象。 CloudFront 通過發送整個對象來提供當前請求，同時還將其緩存以供 future 請求使用。在邊緣緩存中緩存整個對象之 CloudFront 後，它通過服務Range GET請求的範圍響應新的請求。

在任何一種情況下，只要第一個字節從原點到達，就會 CloudFront 開始向最終用戶提供請求的範圍或對象。

> **Note**
>
> 如果檢視器提出要Range GET求且 origin 傳回Transfer-Encoding: chunked，則會將整個物件 CloudFront 傳回給檢視器，而不是要求的範圍。

CloudFront 通常遵循Range標頭的 RFC 規範。但是，如果您的Range標頭不符合下列要求，則會 CloudFront 傳回200含有完整物件的 HTTP 狀態碼，而不是206具有指定範圍的狀態碼：

- 必須以遞增順序列出的範圍。例如，100-200,300-400 是有效的，300-400,100-200 是無效的。
- 範圍不得重疊。例如，100-200,150-250 是無效的。
- 所有範圍規格必須有效。例如，您無法指定負值為範圍的一部分。

如需 Range 請求標頭的更多資訊，請參閱 RFC 7233 中的範圍請求，或者參閱 MDN Web 文件中的範圍。

使用範圍請求快取大物件

啟用快取時， CloudFront 不會擷取或快取大於 50 GB 的物件。當來源指示物件大於此大小 (在回Content-Length應標頭中) 時，請 CloudFront 關閉與原點的連線，並將錯誤傳回給檢視器。

(停用快取功能後， CloudFront 可以從原點擷取大於此大小的物件，並將其傳遞給檢視器。 但是， CloudFront 不會緩存對象。)

不過，對於範圍要求，您可以使用 CloudFront 來快取大於可快取檔案大小上限的物件。例如，考慮具有 100 GB 物件的原始伺服器。啟用快取後， CloudFront 不會擷取或快取這麼大的物件。不過，檢視器可以傳送多個範圍請求，以擷取多個部分的物件，其中每個部分都小於 50 GB。例如，檢視器可以請求包含 20 GB 部分的物件，方法是傳送帶有標頭 `Range: bytes=0-21474836480` 的請求來擷取第一個部分，然後傳送帶有標頭 `Range: bytes=21474836481-42949672960` 的另一個請求來擷取下一個部分，依此類推。檢視器收到所有的部分之後，它可以將這些部分組合起來建構原始的 100 GB 物件。在此情況下，會 CloudFront 快取物件的每個 20 GB 部分，並且可以回應快取中相同零件的後續要求。

如何從您的來源 CloudFront 處理 HTTP 3xx 狀態碼

從 Amazon S3 儲存貯體或自訂原始伺服器 CloudFront 請求物件時，您的原始伺服器有時會傳回 HTTP 3xx 狀態碼。這通常代表下列其中一項：

- 物件的 URL 已變更 (例如，狀態碼 301、302、307 或 308)
- 自上次 CloudFront 請求以來，該對象沒有改變 (狀態碼 304)

CloudFront 根據 CloudFront 發行版中的設定和回應中的標頭，快取 3xx 回應。 CloudFront 只有當您在來源的響應中包含`Cache-Control`標題時，才會緩存 307 和 308 響應。如需詳細資訊，請參閱 管理內容保持在快取中達多久時間 (過期)。

如果您的來源傳回重新導向狀態碼 (例如 301 或 307)，則 CloudFront 不會遵循重新導向。 CloudFront 沿著 301 或 307 響應傳遞給觀眾，誰可以通過發送一個新的請求跟隨重定向。

如何從您的來源 CloudFront 處理和緩存 HTTP 4xx 和 5xx 狀態碼

主題

- 設定自訂錯誤頁面時如何 CloudFront 處理錯誤
- 尚未設定自訂錯誤頁面時如何 CloudFront 處理錯誤
- 可快取的狀態碼 CloudFront

從 Amazon S3 儲存貯體或自訂原始伺服器 CloudFront 請求物件時，您的原始伺服器有時會傳回 HTTP 4xx 或 5xx 狀態碼，表示發生錯誤。 CloudFront 行為取決於：

- 無論您是否已設定自訂錯誤頁面。
- 您是否已設定要 CloudFront 從來源快取錯誤回應的時間長度 (錯誤快取最低 TTL)。
- 狀態碼。
- 對於 5xx 狀態碼，請求的對象當前是否在 CloudFront 邊緣緩存中。
- 對於某些 4xx 狀態代碼，無論原始伺服器傳回 `Cache-Control max-age` 或 `Cache-Control s-maxage` 標頭。

CloudFront 始終緩存響應`GET`和`HEAD`請求。您也可以設定 CloudFront 為快取要`OPTIONS`求的回應。CloudFront 不會快取回應至使用其他方法的要求。

如果來源沒有回應，則對來源的 CloudFront 請求會超時，即使來源沒有回應該錯誤，也會被視為來源的 HTTP 5xx 錯誤。在這種情況下，會 CloudFront 繼續提供快取的內容。如需詳細資訊，請參閱 原始伺服器無法使用。

如果您已啟用記錄，則無論 HTTP 狀態碼為何，都會將結果 CloudFront 寫入記錄檔。

如需有關與從傳回之錯誤訊息相關的功能和選項的詳細資訊 CloudFront，請參閱下列內容：

- 如需有關主控台中自訂錯誤頁面設定的資 CloudFront 訊，請參閱自訂錯誤頁面和錯誤快取。
- 如需有關在主控台中快取最小 TTL 錯誤的資 CloudFront 訊，請參閱錯誤快取最短 TTL (秒)。
- 如需 CloudFront 快取的 HTTP 狀態碼清單，請參閱可快取的狀態碼 CloudFront。

設定自訂錯誤頁面時如何 CloudFront 處理錯誤

如果您已設定自訂錯誤頁面， CloudFront 行為取決於要求的物件是否位於 Edge 快取中。

請求的物件不在邊緣快取中

CloudFront 當以下所有條件都成立時，繼續嘗試從您的來源獲取請求的對象：

- 一個檢視器請求了一個物件。
- 該物件未在節點快取中。
- 您的原始伺服器會傳回 HTTP 4xx 或 5xx 狀態碼，且下列其中一項為真：
  - 您的原始伺服器會傳回 HTTP 5xx 狀態碼，而不傳回 304 狀態碼 (未修改) 或物件的更新版本。
  - 您的原始伺服器會傳回 HTTP 4xx 狀態碼，且不限於快取控制標頭，並包含在以下狀態碼清單中：總是 CloudFront緩存的 HTTP 4xx 和 5xx 狀態碼。

- 您的原始伺服器會傳回沒有 `Cache-Control max-age` 標頭或 `Cache-Control s-maxage` 標頭的 HTTP 4xx 狀態碼，且該狀態碼包含在以下狀態碼清單中：控制 [根據標頭 CloudFront快取的 HTTP 4xx 狀態碼 `Cache-Control`](#)。

CloudFront 執行以下操作：

1. 在收到檢視器要求的 CloudFront Edge 快取中， CloudFront 檢查您的散發組態，並取得與原始伺服器傳回的狀態碼相對應的自訂錯誤頁面路徑。
2. CloudFront 尋找發行版中第一個具有與自訂錯誤頁面路徑相符的路徑模式的快取行為。
3. CloudFront Edge 位置會將自訂錯誤頁面的要求傳送至快取行為中指定的來源。
4. 原始伺服器傳回自訂錯誤頁面至節點。
5. CloudFront 會將自訂錯誤頁面傳回給提出要求的檢視器，並快取自訂錯誤頁面，達到下列最大值：
   - 由錯誤快取最短 TTL (預設為 10 秒) 指定的時間數
   - 當第一個請求產生錯誤時，由原始伺服器傳回，並由 `Cache-Control max-age` 標頭或 `Cache-Control s-maxage` 標頭指定的時間量
6. 過了快取時間 (在步驟 5 中決定) 之後，再次 CloudFront 嘗試將另一個要求轉送至您的來源來取得要求的物件。 CloudFront 繼續以錯誤快取最小 TTL 指定的間隔重試。

請求的物件在邊緣快取中

CloudFront 當下列所有條件都成立時，會繼續提供目前在邊緣快取中的物件：

- 一個檢視器請求了一個物件。
- 物件在節點快取但已過期
- 您的原始伺服器會傳回 HTTP 5xx 狀態碼，而不傳回 304 狀態碼 (未修改) 或物件的更新版本。

CloudFront 執行以下操作：

1. 如果您的來源傳回 5xx 狀態碼，即使物件已過期，仍會 CloudFront 提供該物件。在錯誤快取最小 TTL 的持 CloudFront 續時間內，透過從 Edge 快取提供物件，繼續回應檢視器要求。

   如果您的來源傳回 4xx 狀態碼，則會將狀態碼 (而非要求的物件) CloudFront 傳回給檢視器。
2. 錯誤快取最小 TTL 經過之後，請再次 CloudFront 嘗試將另一個要求轉送至您的來源來取得要求的物件。請注意，如果不經常請求該對象，則 CloudFront 可能會在原始服務器仍在返回 5xx 響應時將

其從邊緣緩存中驅逐出。如需有關物件在 CloudFront Edge 快取中保留多久的資訊，請參閱管理內容保持在快取中達多久時間 (過期)。

尚未設定自訂錯誤頁面時如何 CloudFront 處理錯誤

如果您尚未設定自訂錯誤頁面， CloudFront 行為取決於要求的物件是否位於 Edge 快取中。

請求的物件不在邊緣快取中

CloudFront 當以下所有條件都成立時，繼續嘗試從您的來源獲取請求的對象：

- 一個檢視器請求了一個物件。
- 該物件未在節點快取中。
- 您的原始伺服器會傳回 HTTP 4xx 或 5xx 狀態碼，且下列其中一項為真：
  - 您的原始伺服器會傳回 HTTP 5xx 狀態碼，而不傳回 304 狀態碼 (未修改) 或物件的更新版本。
  - 您的原始伺服器會傳回 HTTP 4xx 狀態碼，且不限於快取控制標頭，並包含在以下狀態碼清單中：總是 CloudFront緩存的 HTTP 4xx 和 5xx 狀態碼
  - 您的原始伺服器會傳回沒有 `Cache-Control max-age` 標頭或 `Cache-Control s-maxage` 標頭的 HTTP 4xx 狀態碼，且該狀態碼包含在以下狀態碼清單中：控制 根據標頭 CloudFront快取的 HTTP 4xx 狀態碼 `Cache-Control`。

CloudFront 執行以下操作：

1. CloudFront 將 4xx 或 5xx 狀態碼傳回給檢視器，並且還會在收到以下最大請求的邊緣快取中快取中快取狀態碼：
   - 由錯誤快取最短 TTL (預設為 10 秒) 指定的時間數
   - 當第一個請求產生錯誤時，由原始伺服器傳回，並由 `Cache-Control max-age` 標頭或 `Cache-Control s-maxage` 標頭指定的時間量
2. 對於快取時間的持續時間 (在步驟 1 中決定)，使用快取的 4xx 或 5xx 狀態碼 CloudFront 回應相同物件的後續檢視器要求。
3. 過了快取時間 (在步驟 1 中決定) 之後，再次 CloudFront 嘗試將另一個要求轉送至您的來源來取得要求的物件。 CloudFront 繼續以錯誤快取最小 TTL 指定的間隔重試。

請求的物件在邊緣快取中

CloudFront 當下列所有條件都成立時，會繼續提供目前在邊緣快取中的物件：

- 一個檢視器請求了一個物件。
- 物件在節點快取但已過期
- 您的原始伺服器會傳回 HTTP 5xx 狀態碼，而不傳回 304 狀態碼 (未修改) 或物件的更新版本。

CloudFront 執行以下操作：

1. 如果您的來源傳回 5xx 錯誤碼，即使物件已過期，仍會 CloudFront 提供該物件。在錯誤快取的持續時間下限 TTL (預設情況下為 10 秒)，透過從 Edge 快取提供物件， CloudFront 繼續回應檢視器要求。

   如果您的來源傳回 4xx 狀態碼，則會將狀態碼 (而非要求的物件) CloudFront 傳回給檢視器。
2. 錯誤快取最小 TTL 經過之後，請再次 CloudFront 嘗試將另一個要求轉送至您的來源來取得要求的物件。請注意，如果不經常請求該對象，則 CloudFront 可能會在原始服務器仍在返回 5xx 響應時將其從邊緣緩存中驅逐出。如需有關物件在 CloudFront Edge 快取中保留多久的資訊，請參閱管理內容保持在快取中達多久時間 (過期)。

可快取的狀態碼 CloudFront

CloudFront 根據返回的特定狀態碼以及您的來源是否在響應中返回特定標頭，緩存您的來源返回的 HTTP 4xx 和 5xx 狀態碼。

總是 CloudFront緩存的 HTTP 4xx 和 5xx 狀態碼

CloudFront 始終緩存您的來源返回的以下 HTTP 4xx 和 5xx 狀態碼。如果您已設定 HTTP 狀態碼的自訂錯誤頁面，請 CloudFront 快取自訂錯誤頁面。

404	找不到
414	URI 請求過大。
500	內部伺服器錯誤
501	未導入
502	無效的閘道
503	服務無法使用

504	閘道逾時

根據標頭 CloudFront快取的 HTTP 4xx 狀態碼 **Cache-Control**

CloudFront 只有在您的來源傳回`Cache-Control max-age`或`Cache-Control s-maxage`標頭時，才會快取來源傳回的下列 HTTP 4xx 狀態碼。如果您已為其中一個 HTTP 狀態碼設定了自訂錯誤頁面，而您的來源傳回其中一個快取控制標頭，則會CloudFront 快取自訂錯誤頁面。

400	錯誤的請求。
403	禁止
405	方法不允許
412[1]	先決條件失敗
415[1]	不支援的媒體類型

[1] CloudFront 不支援為這些 HTTP 狀態碼建立自訂錯誤頁面。

視頻點播和實時流視頻 CloudFront

您可以使用 CloudFront 任何 HTTP 來源傳遞隨選視訊 (VOD) 或即時串流視訊。您可以在雲端中設定視訊工作流程的其中一種方法是 CloudFront 搭配AWS媒體服務使用。

主題

關於串流視訊：隨需視訊和即時串流

您必須先使用編碼器來封裝視訊內容，才 CloudFront 能發佈內容。封裝程序會建立區段，其包含您的音訊、視訊和字幕內容。它也會產生資訊清單檔案，其以特定順序描述播放哪些區段以及何時播放。常見的封裝格式包括 MPEG DASH、Apple HLS、Microsoft Smooth Streaming 和 CMAF。

隨需視訊 (VOD) 串流

如果是隨需視訊 (VOD) 串流，您的視訊內容會存放在伺服器上，檢視器隨時都可以觀看。若要製作瀏覽者可進行串流的資產，請使用編碼器，例如 AWS Elemental MediaConvert，以格式化和封裝您的媒體檔案。

將影片封裝成正確的格式後，您可以將影片存放在伺服器或 Amazon S3 儲存貯體中，然後在檢視者 CloudFront 要求時與其交付。

即時視訊串流

對於即時視訊串流，會在發生即時事件時，即時串流您的視訊內容，或設為全天候的即時頻道。若要建立廣播和串流傳遞的即時輸出，請使用編碼器 (例如 AWS Elemental MediaLive) 以壓縮視訊，並將其格式化以供檢視裝置使用。

在視訊編碼之後，您可以將視訊存放在 AWS Elemental MediaStore 中，或是利用 AWS Elemental MediaPackage 來將其轉換為不同的遞送格式。使用這些來源中的任何一個來源來設置 CloudFront 分發以傳遞內容。如需建立分佈來搭配這些服務一起使用的特定步驟和指導方針，請參閱使用 AWS Elemental MediaStore 做為原始伺服器來提供視訊和提供以 AWS Elemental MediaPackage 格式化的即時視訊。

Wowza 和統一流媒體還提供了可用於流式視頻的工具。 CloudFront如需搭配使用 Wowza 的詳細資訊 CloudFront，請參閱 Wowza 文件網站上將您的 Wowza 串流引擎授權帶到 CloudFront 即時 HTTP 串流。如需搭配使用統一串流CloudFront 進行 VOD 串流的詳細資訊，請參閱整合串流文件網站CloudFront上的。

提供隨選視訊 (VOD) 搭配 CloudFront

若要提供隨選視訊 (VOD) 串流 CloudFront，請使用下列服務：

- Amazon S3，將內容以其原始格式存放，並存放已轉碼的視訊。
- 編碼器 (例如 AWS Elemental MediaConvert)，可將視訊轉碼至串流格式。
- CloudFront 將轉碼後的影片提供給觀眾。對於 Microsoft Smooth Streaming，請參閱為 Microsoft Smooth Streaming 設定隨需視訊。

若要使用建立 VOD 解決方案 CloudFront

1. 將內容上傳至 Amazon S3 儲存貯體。若要進一步了解如何使用 Amazon S3，請參閱 Amazon Simple Storage Service 使用者指南。
2. 使用工作對您的內容進 MediaConvert 行轉碼。此任務會將視訊轉換為您的檢視器使用之播放器所需的格式。您也可以使用任務來建立具有不同解析度和位元速率的資產。這些資產用於調整式位元速率 (ABR) 串流，可根據檢視者的可用頻寬調整檢視品質。 MediaConvert 將轉碼後的視訊儲存在 S3 儲存貯體中。
3. 使用 CloudFront 分發傳遞轉換後的內容。檢視器可以隨時在任何裝置上觀看內容。

> **Tip**
>
> 您可以了解如何使用 AWS CloudFormation 範本來部署 VOD AWS 解決方案，以及所有相關聯的元件。若要查看使用範本的步驟，請參閱 AWS 隨需視訊指南中的自動化部署。

為 Microsoft Smooth Streaming 設定隨需視訊

您可以使用下列選項 CloudFront 來發佈您已轉碼為 Microsoft 流暢串流格式的隨選視訊 (VOD) 內容：

- 指定執行 Microsoft IIS 並支援 Smooth Streaming 的 Web 伺服器做為分佈的來源。

- 在 CloudFront 發行版的快取行為中啟用「流暢串流」。由於您可以在分佈中使用多個快取行為，因此您可以將一個分佈用於 Smooth Streaming 媒體檔案及其他內容。

> **Important**
>
> 如果您指定執行 Microsoft IIS 的網頁伺服器做為原始伺服器，請勿在 CloudFront 發行版的快取行為中啟用「平滑串流」。 CloudFront 如果您啟用「流暢串流」做為快取行為，就無法使用 Microsoft IIS 伺服器做為原始伺服器。

如果您在快取行為中啟用 Smooth Streaming (也就是說，您沒有執行 Microsoft IIS 的伺服器)，請注意以下事項：

- 如果內容與該快取行為 Path Pattern (路徑模式) 的值相符，您仍然可以使用相同的快取行為來分配其他內容。
- CloudFront 可以使用 Amazon S3 儲存貯體或自訂來源進行流暢的串流媒體檔案。 CloudFront 如果您針對快取行為啟用「平滑串流」，則無法使用 Microsoft IIS 伺服器做為原始伺服器。
- 您不能使 Smooth Streaming 格式的媒體檔案無效。如果想要在過期之前更新檔案，您必須重新命名。如需詳細資訊，請參閱 新增、移除或取代 CloudFront散佈的內容。

如需有關「流暢串流」用戶端的資訊，請參閱 Microsoft 文件網站上的「流暢串流」

當 Microsoft IIS 網頁伺服器不是原始伺服器時，用 CloudFront 來散發流暢的串流檔案

1. 將您的媒體檔案轉碼成 Smooth Streaming 片段的 MP4 格式。
2. 執行以下任意一項：
   - 如果您使用 CloudFront 主控台：當您建立或更新發行版時，請在發行版的一或多個快取行為中啟用「流暢串流」。
   - 如果您使用的是 CloudFront API：將`SmoothStreaming`元素添加到`DistributionConfig`複雜類型中，以獲取一個或多個發行版本的緩存行為。
3. 將 Smooth Streaming 檔案上傳到原始伺服器。
4. 建立一個 `clientaccesspolicy.xml` 或 `crossdomainpolicy.xml` 檔案，並將它加入到可在分佈的根目錄中存取的位置，例如，`https://d111111abcdef8.cloudfront.net/clientaccesspolicy.xml`。以下為政策的範例：

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
<cross-domain-access>
<policy>
<allow-from http-request-headers="*">
<domain uri="*"/>
</allow-from>
<grant-to>
<resource path="/" include-subpaths="true"/>
</grant-to>
</policy>
</cross-domain-access>
</access-policy>
```

如需詳細資訊，請參閱 Microsoft Developer Network 網站上的 Making a Service Available Across Domain Boundaries (讓服務可跨網域邊界使用)。

5. 對於您的應用程式 (例如，媒體播放器) 中的連結，請以下列格式為媒體檔案指定 URL：

   ```
   https://d111111abcdef8.cloudfront.net/video/presentation.ism/Manifest
   ```

使用 CloudFront 和AWS媒體服務提供即時串流視訊

若要搭配 CloudFront 使用AWS媒體服務向全球使用者提供即時內容，請遵循本節中包含的指引。

使用 AWS Elemental MediaLive 即時編碼即時視訊串流。若要對大型視訊串流進行編碼，請將其 MediaLive 壓縮為較小的版本 (編碼)，以便發佈給觀眾。

壓縮即時視訊串流之後，您可以使用以下兩個主要選項的其中之一來準備和提供內容：

- 將您的內容轉換成所需格式，然後提供它：如果您需要多種格式的內容，請使用 AWS Elemental MediaPackage 來封裝不同裝置類型的內容。封裝內容時，您也可以實作其他功能，並新增數位版權管理 (DRM)，以防止未經授權使用您的內容。如需 step-by-step 使用 CloudFront 來提供 MediaPackage 格式化內容的指示，請參閱提供以 AWS Elemental MediaPackage 格式化的即時視訊。
- 使用可擴展的來源存儲和提供內容：如果 MediaLive 編碼的內容是您的所有設備所需的格式，請使用可高度擴展的來源，例AWS Elemental MediaStore如提供內容。如需 step-by-step 使用 CloudFront 來提供儲存在容器中之內 MediaStore 容的指示，請參閱使用 AWS Elemental MediaStore 做為原始伺服器來提供視訊。

當您使用其中一個選項設定您的來源後，就可以使用 CloudFront 將即時串流視訊分發給瀏覽者。

> **Tip**
>
> 您可以了解自動部署服務來打造高可用性即時觀賞體驗的 AWS 解決方案。若要檢視自動部署此解決方案的步驟，請參閱即時串流自動化部署。

主題

- 使用 AWS Elemental MediaStore 做為原始伺服器來提供視訊
- 提供以 AWS Elemental MediaPackage 格式化的即時視訊

使用 AWS Elemental MediaStore 做為原始伺服器來提供視訊

如果您有儲存在容AWS Elemental MediaStore器中的視訊，您可以建立 CloudFront 分發來提供內容。

若要開始使用，您可以授與 MediaStore 容器的 CloudFront 存取權。然後，您創建一個CloudFront 發行版並將其配置為使用 MediaStore。

提供來自 AWS Elemental MediaStore 容器的內容

1. 請遵循允許 Amazon 存取您 CloudFront 的AWS Elemental MediaStore容器中的程序，然後返回這些步驟以建立您的分發。
2. 透過下列設定建立分佈：

   原始網域

   指派給 MediaStore 容器的資料端點。從下拉列表中，選擇直播視頻的 MediaStore 容器。

   原始伺服器路徑

   儲存物件的 MediaStore 容器中的資料夾結構。如需詳細資訊，請參閱 the section called “原始伺服器路徑”。

   新增自訂標頭

   如果您想在將請求轉寄 CloudFront 至您的來源時新增自訂標頭，請新增標頭名稱和值。

檢視器通訊協定政策

選擇 Redirect HTTP to HTTPS (將 HTTP 重新引導至 HTTPS)。如需詳細資訊，請參閱 the section called “檢視器通訊協定政策”。

快取政策和原始伺服器請求政策

針對 Cache policy (快取政策)，選擇 Create policy (建立政策)，然後建立適合您快取需求和區段持續時間的快取政策。建立政策後，重新整理快取政策的清單，然後選擇您剛建立的政策。

針對 Origin 要求政策，請從下拉式清單CustomOrigin中選擇 CORS-。

對於其他設定，您可以根據您的其他技術需求或業務需求來設定特定的值。如需適用於分佈的所有選項清單以及設定相關資訊，請參閱the section called “分佈設定”。

3. 對於應用程式中的連結 (例如媒體播放程式)，請使用與您正在發佈的其他物件所使用的格式相同的格式來指定媒體檔案名稱 CloudFront。

提供以 AWS Elemental MediaPackage 格式化的即時視訊

如果您使用 AWS Elemental MediaPackage 格式化即時串流，您可以建立 CloudFront 分佈及設定快取行為來提供即時串流。下列程序假設您已使用建立頻道，並為即時影片新增端點 MediaPackage。

若要 MediaPackage 手動建立 CloudFront 發行版，請依照下列步驟執行：

步驟

- 步驟 1：建立並設定 CloudFront 散發
- 步驟 2：為 MediaPackage 端點的網域新增來源
- 步驟 3：為所有端點設定快取行為
- 步驟 4：啟用基於標題的 CDN MediaPackage 授權
- 步驟 5：用CloudFront 於提供直播頻道

步驟 1：建立並設定 CloudFront 散發

請完成下列程序，為您建立的即時視訊頻道設定 CloudFront 分發 MediaPackage。

為您的即時視訊頻道建立分佈

1. 登入AWS Management Console並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇 Create Distribution (建立分佈)。
3. 選擇分佈的設定，包括下列項目：

   原始網域

   MediaPackage 即時視訊頻道和端點所在的來源。選擇文字欄位，然後從下拉式清單中選擇即時影片的 MediaPackage 來源網域。您可以將一個網域對應到多個原始端點。

   如果您已使用另一個 AWS 帳戶建立原始網域，請在此欄位中輸入原始 URL 值。原始伺服器必須為 HTTPS URL。

   例如，對於像是 `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8` 的 HLS 端點，原始網域會是 `3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com`。

   如需詳細資訊，請參閱 the section called “原始網域”。

   原始伺服器路徑

   提供內容所在 MediaPackage 端點的路徑。

   原始路徑欄位不會自動填入。您必須手動輸入正確的原始路徑。

   如需有關原始伺服器路徑如何運作的詳細資訊，請參閱 the section called “原始伺服器路徑”。

> **Important**
>
> 通配符路徑需要*在發行版中的某個位置進 CloudFront 行路由。為了避免不符合明確路徑的請求路由至真實來源的情況，請為該萬用字元路徑建立一個「虛設」來源。

Example ：建立「虛設」來源

在下列範例中，端點 `abc123` 和 `def456` 路由至「真實」來源，但對任何其他端點的視訊內容的請求會路由至沒有適當子網域的 `mediapackage.us-west-2.amazonaws.com`，這樣會導致 HTTP 404 錯誤。

MediaPackage 端點：

```
https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/
index.m3u8
https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/def456/
index.m3u8
```

CloudFront 原產地 A:

```
Domain: 3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com
Path: None
```

CloudFront 原產地 B:

```
Domain: mediapackage.us-west-2.amazonaws.com
Path: None
```

CloudFront 緩存行為：

```
1. Path: /out/v1/abc123/* forward to Origin A
2. Path: /out/v1/def456/* forward to Origin A
3. Path: * forward to Origin B
```

對於其他分佈設定，請根據您的其他技術需求或業務需求來設定特定的值。如需適用於分佈的所有選項清單以及設定相關資訊，請參閱the section called “分佈設定”。

當您完成選擇其他分佈設定時，請選擇 Create Distribution (建立分佈)。

4. 選擇您剛才建立的分佈，然後選擇 Behaviors (行為)。
5. 選擇預設的快取行為，然後選擇 Edit (編輯)。針對您為原始伺服器選擇的頻道指定正確的快取行為設定。您之後將會新增額外的一或多部原始伺服器，並編輯其快取行為設定。
6. 轉到CloudFront 分發頁面。
7. 等到發行版的 [上次修改] 欄的值已從 [部署] 變更為日期和時間，表示 CloudFront 已建立您的發行版本。

步驟 2：為 MediaPackage 端點的網域新增來源

重複此處的步驟，將每個 MediaPackage 通道端點新增到您的發佈中，請記住建立「虛擬」來源的必要性。

新增其他端點當做原始伺服器

1. 在主 CloudFront 控台上，選擇您為頻道建立的發行版。
2. 選擇 Origins (原始伺服器)，然後選擇 Create origin (建立原始伺服器)。
3. 對於 Origin 網域，請在下拉式清單中選擇頻道的MediaPackage 端點。
4. 對於其他設定，請根據您的其他技術需求或業務需求來設定值。如需詳細資訊，請參閱 the section called “原始設定”。
5. 選擇 Create Origin (建立原始伺服器)。

步驟 3：為所有端點設定快取行為

對於每個端點，您都必須設定快取行為，以新增路徑模式來正確地路由傳遞請求。您指定的路徑模式取決於您提供的視訊格式。以下程序包含用於 Apple HLS、CMAF、DASH 和 Microsoft Smooth Streaming 格式的路徑模式資訊。

您通常會針對每個端點設定兩個快取行為：

- 父系資訊清單，也就是檔案的索引。
- 區段，也就是影片內容的檔案。

為端點建立快取行為

1. 在主 CloudFront 控台上，選擇您為頻道建立的發行版。
2. 選擇 Behaviors (行為)，然後選擇 Create behavior (建立行為)。
3. 對於路徑模式，請使用特定的 MediaPackage `OriginEndpoint` GUID 作為路徑前綴。

   路徑模式

   如果是 HLS 端點 (如 `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8`)，請建立以下兩個快取行為：
   - 對於父系和子系資訊清單，請使用 `/out/v1/abc123/*.m3u8`。
   - 對於內容區段，請使用 `/out/v1/abc123/*.ts`。

如果是 CMAF 端點 (如 https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8)，請建立以下兩個快取行為：

- 對於父系和子系資訊清單，請使用 /out/v1/abc123/*.m3u8。
- 對於內容區段，請使用 /out/v1/abc123/*.mp4。

如果是 DASH 端點 (如 https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.mpd)，請建立以下兩個快取行為：

- 對於父系資訊清單，請使用 /out/v1/abc123/*.mpd。
- 對於內容區段，請使用 /out/v1/abc123/*.mp4。

如果是 Microsoft Smooth Streaming 端點 (如 https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.ism)，則只會提供清單檔案，所以您只會建立一個快取行為：out/v1/abc123/index.ism/*。

4. 對於每個快取行為，請指定下列設定的值：

檢視器通訊協定政策

選擇 Redirect HTTP to HTTPS (將 HTTP 重新引導至 HTTPS)。

快取政策和原始伺服器請求政策

針對 Cache policy (快取政策)，選擇 Create policy (建立政策)。針對新的快取政策，請指定下列設定：

最短 TTL

設定為 5 秒或更少，以協助防止提供過時內容。

查詢字串

針對 (Cache key settings (快取金鑰設定) 中的) Query strings (查詢字串)，選擇 Include specified query strings (包含指定查詢字串)。針對 Allow (允許)，輸入下列值來進行新增，並選擇 Add item (新增項目)：

- 新增m為您要用 CloudFront 來做為快取基礎的查詢字串參數。 MediaPackage 回應永遠包含用?m=###來擷取端點修改時間的標籤。如果內容已使用此標籤的不同值快取，CloudFront 請求新的資訊清單，而不是提供快取版本。
- 如果您在中使用時移檢視功能 MediaPackage，請在資訊清單要求 (*.m3u8、start和index.ism/*) 的快取行為上指定和end做為其他查詢字串參

數。*.mpd如此一來，就會提供資訊清單請求內所要求之時段的專屬內容。如需有關時間轉移檢視和格式化內容 start 與 end 請求參數的詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的時間轉移檢視。

- 如果您在中使用資訊清單篩選功能 MediaPackage，請指定aws.manifestfilter為快取原則的其他查詢字串參數，該參數與資訊清單要求 (*.m3u8*.mpd、和index.ism/*) 的快取行為搭配使用。這將配置您的發行版以將aws.manifestfilter查詢字符串轉發到您的 MediaPackage 來源，這是清單過濾功能工作所必需的。如需詳細資訊，請參閱《AWS Elemental MediaPackage 使用者指南》中的資訊清單檔案篩選。
- 如果您使用低延遲 HLS (LL-HLS)，請指定 _HLS_msn 和 _HLS_part 作為您搭配清單檔案請求的快取行為所使用快取政策 (*.m3u8) 的其他查詢字串參數。這將配置您的發行版以將_HLS_msn和_HLS_part查詢字符串轉發到您的MediaPackage 來源，這是 LL-HLS 阻止播放列表請求功能工作所必需的。

5. 選擇建立。
6. 建立快取政策後，返回快取行為建立工作流程。重新整理快取政策的清單，然後選擇您剛建立的政策。
7. 選擇 Create behavior (建立行為)。
8. 如果您的端點並非 Microsoft Smooth Streaming 端點，請重複前述的步驟來建立第二個快取行為。

步驟 4：啟用基於標題的 CDN MediaPackage 授權

我們建議您在 MediaPackage CloudFront 端點和散佈之間啟用標頭型 MediaPackage CDN 授權。如需詳細資訊，請參閱AWS Elemental MediaPackage使用指南 MediaPackage中的啟用 CDN 授權。

步驟 5：用CloudFront 於提供直播頻道

建立散發、新增來源、建立快取行為，以及啟用標頭型 CDN 授權之後，您可以使用. CloudFront CloudFront 根據您針對快取行為所設定的設定，將檢視器的要求路由至正確的MediaPackage 端點。

對於應用程式中的連結 (例如媒體播放器)，請以 URL 的標準格式指定媒體檔案的 CloudFront URL。如需更多詳細資訊，請參閱 the section called “自訂檔案 URL”。

使用函數在邊緣自訂

使用 Amazon CloudFront，您可以撰寫自己的程式碼來自訂 CloudFront 分發處理 HTTP 請求和回應的方式。程式碼會靠近檢視器 (使用者) 執行，以將延遲降至最低，而且您不必管理伺服器或其他基礎設施。您可以撰寫程式碼來操控流經的要求和回應 CloudFront、執行基本驗證和授權、在邊緣產生 HTTP 回應等等。

您編寫並附加到 CloudFront 發行版的代碼稱為邊緣函數。 CloudFront 提供兩種編寫和管理邊緣函數的方法：

- CloudFront 函數 — 使用 F CloudFront unctions，您可以在中編寫輕量級函數，以進行高規模，JavaScript 對延遲敏感的 CDN 自定義。 CloudFront Functions 執行階段環境提供低於一毫秒的啟動時間，可立即擴展以每秒處理數百萬個要求，而且非常安全。 CloudFront 函數是的原生功能 CloudFront，這表示您可以在其中完全建置、測試和部署程式碼 CloudFront。
- Lambda@Edge – Lambda@Edge 是 AWS Lambda 的延伸，為複雜函數提供強大且靈活的運算，同時提供更靠近檢視器的完整應用應程式邏輯，並且具有高度安全性。Lambda@Edge 函數在 Node.js 或 Python 執行階段環境中執行。您可以將它們發佈到單一 AWS 區域，但是當您將函數與 CloudFront 分發產生關聯時，Lambda @Edge 會自動在全球範圍內複寫您的程式碼。

如果您 AWS WAF 在上執行 CloudFront，則可以為 CloudFront 函數和 Lambda @Edge 使用 AWS WAF 插入的標頭。這適用於查看者和源請求和響應。

主題

在 CloudFront 函數和 Lambda @Edge 之間選擇

CloudFront 函數和 Lambda @Edge 都提供了一種執行程式碼以回應 CloudFront事件的方式。但是，兩者之間存在重要差異。這些差異可以幫助您選擇適合使用案例的方式。下表列出 CloudFront 函數和 Lambda @Edge 之間的一些重要差異。

	CloudFront 函数	Lambda@Edge
程式設計語言	JavaScript (符合電子印刷稿 5.1 規範)	Node.js 和 Python
事件來源	• 檢視器請求 • 檢視器回應	• 檢視器請求 • 檢視器回應 • 原始伺服器請求 • 原始伺服器回應
支持 Amazon CloudFront KeyValueStore	是 CloudFront KeyValueStore 僅支援JavaScript 執行階段 2.0	否
擴展	每秒 10,000,000 個請求或更多	每秒每個區域最多 10,000 個請求
函數持續時間	低於一毫秒	最多 5 秒 (檢視器請求和檢視器回應) 最多 30 秒 (原始伺服器請求和原始伺服器回應)
最大記憶體容量 如需詳細資訊，請參閱 Lambda 配額。	2 MB	一千二百八十二千兆
函數程式碼和包含程式庫的最大規模	10 KB	1 MB (檢視器請求和檢視器回應) 50 MB (原始伺服器請求和原始伺服器回應)
網路存取	否	是
檔案系統存取	否	是

	CloudFront 函數	Lambda@Edge
請求內文存取	否	是
存取地理位置和裝置資料	是	否 (檢視者要求和檢視者回應) 是 (原始請求和來源回應)
可以完全在內部構建和測試 CloudFront	是	否
函數日誌記錄和指標	是	是
定價	免費方案可用；按請求收費	無免費方案；按請求和函數持續時間收費

CloudFront 函數非常適合用於以下使用案例的輕量級短時間運行功能：

- 快取金鑰標準化 – 您可以轉換 HTTP 請求屬性 (標頭、查詢字串、Cookie，甚至 URL 路徑)，以建立最佳快取金鑰，這可提升您的快取命中率。
- 標頭操作 – 您可以在請求或回應中插入、修改或刪除 HTTP 標頭。例如，您可以為每個請求新增一個 `True-Client-IP` 標頭。
- URL 重新導向或重寫 – 您可以根據請求中的資訊將檢視者重新導向至其他頁面，或將全部請求從一個路徑重寫至另一個路徑。
- 請求授權 – 您可以透過檢查授權標頭或其他請求中繼資料驗證雜湊的授權權杖，例如 JSON Web 權杖 (JWT)。

若要開始使用 CloudFront 函數，請參閱使用 CloudFront 功能在邊緣自定義。

Lambda@Edge 非常適合以下場景：

- 需要數毫秒或更長時間才能完成的函數。
- 需要可調整 CPU 或記憶體的函數。
- 依賴第三方程式庫 (包括 AWS SDK，以便與其他 AWS 服務整合) 的函數。
- 需要網路存取才能使用外部服務進行處理的函數。
- 需要檔案系統存取或存取 HTTP 請求內文的函數。

若要開始使用 Lambda@Edge，請參閱使用 Lambda @Edge 在邊緣進行自訂。

使用 CloudFront 功能在邊緣自定義

使用 F CloudFront unctions，您可以在中編寫輕量級函數，以進 JavaScript 行高規模，延遲敏感的 CDN 自定義。您的函數可以操作流經的請求和響應 CloudFront，執行基本身份驗證和授權，在邊緣生成 HTTP 響應等。 CloudFront Functions 執行階段環境提供低於一毫秒的啟動時間，可立即擴展以每秒處理數百萬個要求，而且非常安全。 CloudFront 函數是的原生功能 CloudFront，這表示您可以在其中完全建置、測試和部署程式碼 CloudFront。

CloudFront 函數非常適合用於以下使用案例的輕量級短時間運行功能：

- 快取金鑰標準化 – 您可以轉換 HTTP 請求屬性 (標頭、查詢字串、Cookie，甚至 URL 路徑)，以建立最佳快取金鑰，這可提升您的快取命中率。
- 標頭操作 – 您可以在請求或回應中插入、修改或刪除 HTTP 標頭。例如，您可以為每個請求新增一個 `True-Client-IP` 標頭。
- 狀態碼修改和本文產生 – 您可以評估標頭並使用自訂內容回應給瀏覽者。
- URL 重新導向或重寫 – 您可以根據請求中的資訊將檢視者重新導向至其他頁面，或將全部請求從一個路徑重寫至另一個路徑。
- 請求授權 – 您可以透過檢查授權標頭或其他請求中繼資料驗證雜湊的授權權杖，例如 JSON Web 權杖 (JWT)。

當您將 CloudFront 函數與 CloudFront 分發相關聯時， CloudFront 攔截 CloudFront 邊緣位置的請求和響應，並將它們傳遞給您的函數。當發生以下事件時，您可以調用 CloudFront 函數：

- CloudFront 收到來自檢視者的要求時 (檢視者要求)
- CloudFront 返回給查看者的響應之前（查看器響應）

如需快速簡介，請參閱教學課程：使用函數建立簡單 CloudFront 函數。

您可以將 CloudFront 函數設定為使用儲存在索引鍵值存放區中的索引鍵值配對，以將變數包含在函數中。如需在 CloudFront 函數中包含鍵值對的快速簡介，請參閱the section called “教學課程：具有鍵值的函數”。

若要開始撰寫函數程式碼並讀取範例程式碼，請參閱撰寫函數程式碼和範例程式碼。

教學課程：使用函數建立簡單 CloudFront 函數

本教程向您展示如何開始使用 CloudFront 函數。您可以創建一個簡單的函數，將查看器重定向到不同的 URL，並返回自定義響應標題。

必要條件

要使用 CloudFront 函數，您需要一個 CloudFront 分佈。如果沒有，請依照 開始使用基本 CloudFront 發行版 中的步驟操作。

建立函數

此程序會示範如何使用 CloudFront 主控台建立簡單函式，將檢視器重新導向至不同的 URL，並傳回自訂回應標頭。

若要在 CloudFront 主控台中建立函數

1. 登入AWS Management Console並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [函數]，然後選擇 [建立函數]。
3. 在 [建立函數] 頁面上，對於 [名稱]，輸入函數名稱，例如*MyFunctionName*。
4. (選擇性) 在說明中，輸入函數的說明，例如**Simple test function**。
5. 對於「執行階段」，保留預設選取的 JavaScript版本。
6. 選擇 建立函式。
7. 複製下列函數程式碼。此函數程式碼會將檢視者重新導向至不同的 URL，並傳回自訂回應標頭。

```
function handler(event) {
    // NOTE: This example function is for a viewer request event trigger.
    // Choose viewer request for event trigger when you associate this function
 with a distribution.
    var response = {
        statusCode: 302,
        statusDescription: 'Found',
        headers: {
            'cloudfront-functions': { value: 'generated-by-CloudFront-Functions' },
            'location': { value: 'https://aws.amazon.com/cloudfront/' }
        }
    };
    return response;
```

```
}
```

8. 對於函數程式碼，請將程式碼貼到程式碼編輯器中，以取代預設程式碼。
9. 選擇儲存變更。
10. （可選）您可以在發布之前對其進行測試。本教程不描述如何測試函數。如需詳細資訊，請參閱測試函數。
11. 選擇「發布」選項卡，然後選擇「發布」功能。您必須先發 CloudFront 佈函數，才能將其與發行版產生關聯。
12. 接下來，您可以將函數與散佈或快取行為相關聯。在*MyFunctionName*頁面上，選擇 [發佈] 索引標籤。

> **Warning**
>
> 在下列步驟中，選擇用於測試的發行版或快取行為。請勿將此測試函數與生產環境中使用的散佈或快取行為相關聯。

13. 選擇 Add association (建立關聯)。
14. 在「關聯」對話方塊中，選擇發佈和/或快取行為。對於事件類型，請保留預設值。
15. 選擇 Add association (建立關聯)。

    關聯的分佈資料表中會顯示關聯的分佈。
16. 等待幾分鐘，讓關聯的分佈完成部署。若要檢查分配的狀態，請在「相關分配」表格中選取分配，然後選擇「檢視分配」。

    當分佈的狀態為已部署時，您就可以確認該函數正常運作。

驗證函數

若要查看執行中的函數並確認其正常運作，請在 Web 瀏覽器中移至分佈的網域名稱 (例如 `https://d111111abcdef8.cloudfront.net`)。該函數返回一個重定向到瀏覽器，因此瀏覽器會自動轉到 `https://aws.amazon.com/cloudfront/`。

如果您使用類似 curl 的工具將請求傳送至分佈的網域名稱，則會看到該函數新增的重新導向回應 (302 Found) 和自訂回應標頭，如下列範例所強調。

```
curl -v https://d111111abcdef8.cloudfront.net/
> GET / HTTP/1.1
```

```
> Host: d111111abcdef8.cloudfront.net
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 302 Found
< Server: CloudFront
< Date: Tue, 16 Mar 2021 18:50:48 GMT
< Content-Length: 0
< Connection: keep-alive
< Location: https://aws.amazon.com/cloudfront/
< Cloudfront-Functions: generated-by-CloudFront-Functions
< X-Cache: FunctionGeneratedResponse from cloudfront
< Via: 1.1 3035b31bddaf14eded329f8d22cf188c.cloudfront.net (CloudFront)
< X-Amz-Cf-Pop: PHX50-C2
< X-Amz-Cf-Id: ULZdIz6j43uGBlXyob_JctF9x7CCbwpNniiMlmNbmwzH1YWP9FsEHg==
```

教學課程：建立包含鍵值的函數

本教程將向您展示如何在 CloudFront 功能中包含鍵值。鍵值是鍵值對的一部分。您可以在函數程式碼中包含名稱 (來自鍵值對)。當函數運行時， CloudFront 將名稱替換為值。

鍵值對是存放在鍵值存放區中的變量。當您在函數中使用鍵 (而不是硬式編碼值) 時，您的函數會更靈活。您可以變更鍵的值，而不需要部署程式碼變更。鍵值對也可以減少函數的大小。如需鍵值對和鍵值存放區的詳細資訊，請參閱 ???。

必要條件

我們假設您熟悉 CloudFront 功能。如果您對函數和鍵值存放區都不熟悉，您應該先遵循 the section called “教學課程：簡單的函數” 中的教學課程。

設定鍵值存放區

步驟 1：建立鍵值存放區

1. 規劃您要包含在函數中的鍵值對。請記下這些鍵的名稱。

   請記住，您要在函數中使用的所有鍵值對都必須位於單一鍵值存放區中。
2. 決定工作的順序。有兩種方式可以繼續：
   - 建立一個鍵值存放區，並將鍵值對新增到存放區中。然後建立 (或修改) 函數並加入鍵的名稱。
   - 或者，建立 (或修改) 函數，並加入您要使用的鍵的名稱。然後建立一個鍵值存放區，並新增鍵值對。

本教學假設您正在擴充函數教學課程中的函數。它還假定您首先會建立鍵值存放區。

3. 登入AWS Management Console並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
4. 在導覽列中，選擇函數。在「函數」頁面上，選擇頁KeyValueStores籤。
5. 選擇 [建立] KeyValueStore 並依下列方式完成欄位：
   - 輸入存放區的名稱，並輸入選擇性的描述。
   - 將 S3 URI 保留空白，因為本教學課程示範如何手動輸入鍵值對。
6. 選擇建立按鈕。此時會顯示新鍵值存放區的詳細資訊頁面。此頁面包含目前空白的鍵值對區段。

步驟 2：將鍵值對新增到存放區

1. 在鍵值對區段中，選擇新增鍵值對按鈕。選擇新增配對，然後輸入名稱和值。
2. 選擇新增配對按鈕，以新增另一對。
3. 完成後，選擇儲存變更以儲存存放區中的所有鍵值對。在出現的確認對話方塊中，選擇完成。

您現在有一個包含一組鍵值對的存放區。

在函數中設定

步驟 3：建立函數與鍵值存放區的關聯

您現在已建立鍵值存放區。並且您已經建立或修改了一個函數，其中包含鍵值存放區中的鍵名稱。您現在可以建立鍵值存放區與函數的關聯。您從函數內建立該關聯。

1. 在導覽列中，選擇函數。依據預設，函數索引標籤會顯示在頂端。
2. 在「關聯 KeyValueStore」區段中，選擇「關聯現有的」 KeyValueStore。選取索引鍵值存放區，然後選擇「關聯」 KeyValueStore 按鈕。請注意，每個函數只能與一個鍵值存放區相關聯。

步驟 4：測試並發佈函數程式碼

1. 每次修改函數程式碼時，應一律對其進行測試，包括執行以下操作時：
   - 將鍵值存放區與函數建立關聯。
   - 修改函數及其鍵值存放區，以包含新的鍵值對。
   - 變更鍵值對的值。

如需有關如何測試函數的詳細資訊，請參閱 the section called “測試函數”。確定您選擇在 DEVELOPMENT 階段測試函數。

2. 當您準備好在 LIVE 環境中使用函數 (搭配新的或修訂的鍵值對) 時，請發佈該函數。

   當您發佈時，會 CloudFront 將該函數的版本從DEVELOPMENT舞台複製到即時階段。該函數具有新程式碼，並與鍵值存放區相關聯。(在即時階段無需再次執行關聯。)

   如需有關如何發佈函數的詳細資訊，請參閱 the section called “發佈函數”。

撰寫函數程式碼

使用 Amazon 中的 CloudFront 函數 CloudFront，您可以在中編寫輕量級函數，以進行高規模、JavaScript 對延遲敏感的 CDN 自訂。您的函數代碼可以操縱流過的請求和響應 CloudFront，執行基本身份驗證和授權，在邊緣生成 HTTP 響應等。

下列主題可協助您撰寫函數的函數程 CloudFront 式碼。

主題

- 確定函數的用途
- CloudFront 函數事件結構
- JavaScript 函數的執行階段 CloudFront 功能
- 鍵值存放區的協助程式方法
- CloudFront 函數的範例程式碼

確定函數的用途

在撰寫函數程式碼之前，請先確定函數的用途。函數中的大多 CloudFront 數函數具有以下目的之一。如需詳細資訊，請參閱與函數用途對應的主題。

無論函數的用途如何，handler 都是任何函數的入口點。它需要一個稱為的參數event，該參數通過傳遞給函數 CloudFront。event 是一個 JSON 對象，其中包含 HTTP 請求的表示 (以及回應，前提是您的函數修改了 HTTP 回應)。如需 event 物件結構的詳細資訊，請參閱 CloudFront 函數事件結構。

如需適用於 CloudFront 函數和 Lambda @Edge 之限制的詳細資訊，請參閱對邊緣函數的限制。

主題

- 修改檢視器請求事件類型中的 HTTP 請求

- 在檢視器請求事件類型中產生 HTTP 回應
- 在檢視者回應事件類型中修改 HTTP 回應

修改檢視器請求事件類型中的 HTTP 請求

您的函數可以修改從查看器（客戶端）CloudFront 接收的 HTTP 請求，並將 CloudFront 修改後的請求返回到以繼續處理。例如，您的函數程式碼可能會標準化快取金鑰或修改請求標頭。

建立修改 HTTP 請求的函數時，請務必選擇檢視者請求事件類型。這意味著該函數在每次 CloudFront 接收來自檢視器的要求時都會執行，然後再檢查要求的物件是否在 CloudFront 快取中。

下面的虛擬程式碼顯示了修改 HTTP 請求的函數結構。

```
function handler(event) {
    var request = event.request;

    // Modify the request object here.

    return request;
}
```

該函數返回修改後的`request`對象 CloudFront。 CloudFront透過檢查快取是否有 CloudFront 快取命中，繼續處理傳回的要求，並在必要時將要求傳送至來源。

如需 `event` 和 `request` 物件結構的詳細資訊，請參閱 事件結構。

在檢視器請求事件類型中產生 HTTP 回應

您的函數可以在邊緣生成 HTTP 響應，並將其直接返回給查看器（客戶端），而無需檢查緩存響應或任何進一步的處理 CloudFront。例如，您的函數程式碼可能會將請求重新導向至新的 URL，或檢查授權並將 `401` 或 `403` 回應返回給未經授權的請求。

建立產生 HTTP 回應的函數時，請務必選擇檢視者請求事件類型。這意味著該函數在每次 CloudFront 接收來自查看器的請求時都會運行，然後再 CloudFront 對請求進行任何進一步處理。

下面的虛擬程式碼顯示了產生 HTTP 回應的函數結構。

```
function handler(event) {
    var request = event.request;

    var response = ...; // Create the response object here,
```

```
                        // using the request properties if needed.

    return response;
}
```

該函數返回一個response對象 CloudFront，該對象 CloudFront立即返回給查看器，而不檢查 CloudFront 緩存或向原點發送請求。

如需 event、request 和 response 物件結構的詳細資訊，請參閱 事件結構。

在檢視者回應事件類型中修改 HTTP 回應

您的函數可以在將 HTTP 響應 CloudFront 發送到查看器（客戶端）之前修改 HTTP 響應，無論響應是來自 CloudFront 緩存還是來自源。例如，您的函數代碼可能會新增或修改回應標頭、狀態碼，與本文內容。

建立可修改 HTTP 回應的函數時，請務必選擇檢視者回應事件類型。這意味著無論響應是來自 CloudFront 緩存還是來自源，該函數都在向查看器 CloudFront 返回響應之前運行。

下面的虛擬程式碼顯示了修改 HTTP 回應的函數結構。

```
function handler(event) {
    var request = event.request;
    var response = event.response;

    // Modify the response object here,
    // using the request properties if needed.

    return response;
}
```

此函數會將修改後的response物件傳回至 CloudFront，該物件會 CloudFront立即傳回給檢視器。

如需 event 和 response 物件結構的詳細資訊，請參閱 事件結構。

如需有關為 CloudFront Functions 撰寫函數程式碼的詳細資訊事件結構，請參閱JavaScript運行時功能、和範例程式碼。

CloudFront 函數事件結構

CloudFront 函數會在執行函數時，將event物件傳遞至函數程式碼做為輸入。測試函數時，建立 event 物件並將其傳遞給函數。建立用於測試函數的 event 物件時，您可以省略

distributionDomainName 物件中的 distributionId、requestId 和 context 欄位。確保標題的名稱是小寫的，在 CloudFront Functions 在生產中傳遞給函數的event對象中始終如此。

以下是此事件物件結構的概觀。如需詳細資訊，請參閱隨後的主題。

```
{
    "version": "1.0",
    "context": {
        <context object>
    },
    "viewer": {
        <viewer object>
    },
    "request": {
        <request object>
    },
    "response": {
        <response object>
    }
}
```

主題

版本欄位

此version欄位包含字串，指定 CloudFront Functions 事件物件的版本。目前版本是 1.0。

內容物件

context 物件包含有關事件的關聯式資訊。它包括以下欄位：

distributionDomainName

與事件相關聯的發行版的網 CloudFront 域名稱 (例如，d111111abcdef8.cloudfront.net)。

distributionId

與事件相關聯的發佈的 ID (例如 EDFDVBD6EXAMPLE)。

eventType

事件類型，`viewer-request` 或 `viewer-response`。

requestId

唯一識別 CloudFront 要求 (及其相關回應) 的字串。

檢視者物件

`viewer` 物件包含一個 `ip` 欄位，其值是傳送請求的檢視者 (用戶端) 的 IP 地址。如果檢視者的請求來自 HTTP 代理或負載平衡器，此值為代理或負載平衡器的 IP 地址。

請求物件

該`request`對象包含查看器對 HTTP 請求的表示。CloudFront 在傳遞給函數的`event`對象中，該`request`對象表示從查看器 CloudFront 收到的實際請求。

如果您的函數代碼返回一個`request`對象 CloudFront，它必須使用相同的結構。

`request` 物件包含下列欄位：

method

請求的 HTTP 方法。如果您的函數程式碼返回 `request`，則無法修改此欄位。這是 `request` 物件中唯一的唯讀欄位。

uri

請求物件的相對路徑。如果您的函數修改 `uri` 值，請注意下列事項：

- 全新的 `uri` 值必須以正斜線 (`/`) 作為開頭。
- 當函數變更 `uri` 值時，它會變更檢視者請求的物件。
- 當函數變更 `uri` 值時，它不會變更請求的快取行為或原始伺服器請求傳送的來源。

querystring

代表請求中的查詢字串的物件。如果請求不包含查詢字串，request 物件仍會包含空白的 querystring 物件。

querystring 物件針對請求中的每個查詢字串參數包含一個欄位。

headers

代表請求中 HTTP 標頭的物件。如果請求包含任何 Cookie 標頭，則這些標頭不是 headers 物件的一部分。Cookies 在 cookies 物件中單獨表示。

headers 物件針對請求中的每個標頭包含一個欄位。在事件物件中，標頭名稱會轉換為小寫，而在函數程式碼新增標頭名稱時，標頭名稱必須是小寫。當 CloudFront Functions 將事件物件轉換回 HTTP 要求時，標頭名稱中每個單字的第一個字母都會大寫。單字以連字號分隔 (-)。例如，如果您的函數代碼添加了一個名為的標頭example-header-name，請在 HTTP 請求Example-Header-Name中將其 CloudFront 轉換為。

cookies

代表請求 (Cookie 標頭) 中 Cookie 的物件。

cookies 物件針對請求中的每個 Cookie 包含一個欄位。

如需有關查詢字串、標頭和 Cookie 結構的詳細資訊，請參閱 查詢字串、標頭和 Cookie 的結構。

如需範例 event 物件，請參閱 範例事件物件。

回應物件

該對response象包含 CloudFront對查看者 HTTP 響應的表示。在傳遞給函數的event對象中，該response對象代表 CloudFront對查看器請求的實際響應。

如果您的函數程式碼返回 response 物件，其必須使用相同的結構。

response 物件包含下列欄位：

statusCode

回應的 HTTP 狀態碼。該值是一個整數，而不是字串。

您的函數可以產生或修改 statusCode。

statusDescription

回應的 HTTP 狀態說明。如果您的函數程式碼產生回應，則此欄位為選用。

headers

代表回應中 HTTP 標頭的物件。如果回應包含任何 Set-Cookie 標頭，則這些標頭不是 headers 物件的一部分。Cookies 在 cookies 物件中單獨表示。

headers 物件針對回應中的每個標頭包含一個欄位。在事件物件中，標頭名稱會轉換為小寫，而在函數程式碼新增標頭名稱時，標頭名稱必須是小寫。當 CloudFront Functions 將事件物件轉換回 HTTP 回應時，標頭名稱中每個單字的第一個字母都會大寫。單字以連字號分隔 (-)。例如，如果您的函數代碼添加了一個名為的標頭example-header-name，請在 HTTP 響應Example-Header-Name中將其 CloudFront 轉換為。

cookies

代表回應 (Set-Cookie 標頭) 中 Cookie 的物件。

cookies 物件針對回應中的每個 Cookie 包含一個欄位。

body

新增 body 欄位是選擇性的，除非您在函數中進行指定，否則它不會出現在 response 物件中。您的函數無法訪問 CloudFront緩存或 origin 返回的原始主體。如果您沒有在檢視器回應函數中指定body欄位，則 CloudFront 快取或 origin 傳回的原始主體會傳回給檢視者。

如果您想 CloudFront 要將自訂內文傳回給檢視器，請在欄位中指定內文內容，並在data欄位中指定encoding內文編碼。您可以將編碼指定為純文字 ("encoding": "text") 或 Base64 編碼的內容 ("encoding": "base64")。

作為捷徑，您也可以直接在 body 欄位 ("body": "<specify the body content here>") 中指定本文內容。執行此操作時，請省略data和encoding欄位。 CloudFront 在這種情況下，將正文視為純文本。

encoding

body 內容 (data 欄位) 的編碼。唯一的有效編碼是 text 和 base64。

如果您指定encoding為，base64但主體不是有效的 base64，則 CloudFront 返回一個錯誤。

data

body 內容。

如需有關已修改狀態碼和本文內容的更多資訊，請參閱 狀態碼和本文。

如需有關標頭和 Cookie 結構的詳細資訊，請參閱 查詢字串、標頭和 Cookie 的結構。

如需範例 `response` 物件，請參閱 範例回應物件。

狀態碼和本文

透過 CloudFront Functions，您可以更新檢視器回應狀態碼、以新的回應內文取代整個回應本文，或移除回應內文。在評估 CloudFront快取或來源回應的各個層面之後，更新檢視器回應的一些常見案例包括：

- 變更狀態以設定 HTTP 200 狀態碼並建立靜態本文內容，以傳回給檢視器。
- 變更狀態以設定 HTTP 301 或 302 狀態碼來重新導向使用者到另一個網站。
- 決定是否要提供或捨棄檢視者回應的本文。

> **Note**
>
> 如果來源傳回 400 及以上的 HTTP 錯誤， CloudFront 函式將無法執行。如需更多資訊，請參閱對所有邊緣函數的限制。

當您使用 HTTP 響應時， CloudFront 函數無法訪問響應主體。您可以透過將本文內容設定為所需的值來進行替換，或設定為空值來移除本文。如果您不更新函數中的 body 字段，則 CloudFront 緩存或 origin 返回的原始主體將返回給查看者。

> **Tip**
>
> 使用 CloudFront Functions 來取代主體時，請務必將對應的標題 (例如`content-encoding`、`content-typecontent-length`、或) 對齊新的本文內容。
> 例如，如果 CloudFront 原點或緩存返回，`content-encoding: gzip`但查看器響應函數設置了純文本的主體，該函數也需要相應地更改`content-encoding`和`content-type`標題。

如果您的 CloudFront 函數配置為返回 400 或更高的 HTTP 錯誤，您的查看器將不會看到您為相同狀態碼指定的自定義錯誤頁面。

查詢字串、標頭和 Cookie 的結構

查詢字串、標頭和 Cookie 共用相同的結構。查詢字串可能會出現在請求中。標頭會出現在請求和回應中。Cookie 會出現在請求和回應中。

每個查詢字串、標頭或 Cookie 都是父系 `querystring`、`headers` 或 `cookies` 物件中的唯一欄位。欄位名稱是查詢字串、標頭或 Cookie 的名稱。每個欄位都包含具有查詢字串、標頭或 Cookie 值的 `value` 屬性。

主題

查詢字串值或查詢字串物件

除了查詢字串物件之外，函數還可以傳回查詢字串值。查詢字串值可用來依任意自訂順序排列查詢字串參數。例如，若要在函數程式碼中修改查詢字串，請使用如下所示的程式碼：

```
var request = event.request;
request.querystring =
 'ID=42&Exp=1619740800&TTL=1440&NoValue=&querymv=val1&querymv=val2,val3';
```

標頭的特殊考量

僅針對標頭，標頭名稱會在事件物件中轉換為小寫，而在函數程式碼新增標頭名稱時，標頭名稱則必須是小寫。當 CloudFront Functions 將事件物件轉換回 HTTP 要求或回應時，標頭名稱中每個單字的第一個字母都會大寫。單字以連字號分隔 (-)。例如，如果您的函數代碼添加了一個名為的標頭`example-header-name`，請`Example-Header-Name`在 HTTP 請求或響應中將其 CloudFront 轉換為。

例如，請考慮 HTTP 請求中的下列 `Host` 標頭：

```
Host: video.example.com
```

此標頭在 `request` 物件中的表示方式如下：

```
"headers": {
```

```
    "host": {
        "value": "video.example.com"
    }
}
```

若要在函數程式碼中存取 Host 標頭，請使用如下所示的程式碼：

```
var request = event.request;
var host = request.headers.host.value;
```

若要在函數程式碼中新增或修改標頭，請使用如下所示的程式碼 (此程式碼會新增名為 X-Custom-Header 且包含值 example value 的標頭)：

```
var request = event.request;
request.headers['x-custom-header'] = {value: 'example value'};
```

重複的查詢字串、標頭和 Cookie (**multiValue** 陣列)

HTTP 請求或回應可以包含多個具有相同名稱的查詢字串、標頭或 Cookie。在此情況下，重複的查詢字串、標頭或 Cookie 會折疊成 request 或 response 物件中的一個欄位，但此欄位包含一個名為 multiValue 的額外屬性。multiValue 屬性包含一個陣列，其中帶有每個重複查詢字串、標頭或 Cookie 的值。

例如，請考慮帶有下列 Accept 標頭的 HTTP 請求：

```
Accept: application/json
Accept: application/xml
Accept: text/html
```

這些標頭在 request 物件中的表示方式如下：

```
"headers": {
    "accept": {
        "value": "application/json",
        "multiValue": [
            {
                "value": "application/json"
            },
            {
                "value": "application/xml"
```

```
            },
            {
                "value": "text/html"
            }
        ]
    }
}
```

請注意，第一個標頭值 (在此情況為 application/json) 會在 value 和 multiValue 屬性中重複。這可讓您透過在 multiValue 陣列中執行迴圈來存取所有值。

如果您的函數程式碼修改了具有multiValue陣列的查詢字串、標頭或 Cookie， CloudFront Functions 會使用下列規則來套用變更：

1. 如果 multiValue 陣列存在且有任何修改，則套用此修改。value 屬性中的第一個元素被忽略。
2. 否則，會套用 value 屬性的任何修改，並且後續的值 (如果存在) 保持不變。

只有當 HTTP 請求或回應包含具有相同名稱的重複查詢字串、標頭或 Cookie 時，才會使用 multiValue 屬性，如上述範例所示。但是，如果單一查詢字串、標頭或 Cookie 中有多個值，則不會使用該 multiValue 屬性。

例如，請考慮帶有一個 Accept 標頭的請求，而該標頭包含三個值，如下列範例所示：

```
Accept: application/json, application/xml, text/html
```

此標頭在 request 物件中的表示方式如下：

```
"headers": {
    "accept": {
        "value": "application/json, application/xml, text/html"
    }
}
```

Cookie 屬性

在 HTTP 回應的 Set-Cookie 標頭中，標頭包含 Cookie 的名稱/值對，以及選用的一組屬性 (以分號分隔)。例如：

```
Set-Cookie: cookie1=val1; Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT
```

在 response 物件中，這些屬性會在 Cookie 欄位的 attributes 屬性中表示。例如，前面的 Set-Cookie 標頭表示如下：

```
"cookie1": {
    "value": "val1",
    "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT"
}
```

範例回應物件

以下範例顯示一個本文已被檢視器回應函數替換的 response 物件 (檢視器回應函數的輸出)。

```
{
  "response": {
    "statusCode": 200,
    "statusDescription": "OK",
    "headers": {
      "date": {
        "value": "Mon, 04 Apr 2021 18:57:56 GMT"
      },
      "server": {
        "value": "gunicorn/19.9.0"
      },
      "access-control-allow-origin": {
        "value": "*"
      },
      "access-control-allow-credentials": {
        "value": "true"
      },
      "content-type": {
        "value": "text/html"
      },
      "content-length": {
        "value": "86"
      }
    },
    "cookies": {
      "ID": {
```

```
        "value": "id1234",
        "attributes": "Expires=Wed, 05 Apr 2021 07:28:00 GMT"
      },
      "Cookie1": {
        "value": "val1",
        "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT",
        "multiValue": [
          {
            "value": "val1",
            "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
 07:28:00 GMT"
          },
          {
            "value": "val2",
            "attributes": "Path=/cat; Domain=example.com; Expires=Wed, 10 Jan 2021
 07:28:00 GMT"
          }
        ]
      }
    },

    // Adding the body field is optional and it will not be present in the response
 object
    // unless you specify it in your function.
    // Your function does not have access to the original body returned by the
 CloudFront
    // cache or origin.
    // If you don't specify the body field in your viewer response function, the
 original
    // body returned by the CloudFront cache or origin is returned to viewer.

     "body": {
      "encoding": "text",
      "data": "<!DOCTYPE html><html><body><p>Here is your custom content.</p></body></
html>"
    }
  }
}
```

範例事件物件

以下範例顯示完整的 event 物件。

> **Note**
>
> event 物件是函數的輸入。您的函數僅返回 request 或 response 物件，而不是完整的 event 物件。

```
{
    "version": "1.0",
    "context": {
        "distributionDomainName": "d111111abcdef8.cloudfront.net",
        "distributionId": "EDFDVBD6EXAMPLE",
        "eventType": "viewer-response",
        "requestId": "EXAMPLEntjQpEXAMPLE_SG5Z-EXAMPLEPmPfEXAMPLEu3EqEXAMPLE=="
    },
    "viewer": {"ip": "198.51.100.11"},
    "request": {
        "method": "GET",
        "uri": "/media/index.mpd",
        "querystring": {
            "ID": {"value": "42"},
            "Exp": {"value": "1619740800"},
            "TTL": {"value": "1440"},
            "NoValue": {"value": ""},
            "querymv": {
                "value": "val1",
                "multiValue": [
                    {"value": "val1"},
                    {"value": "val2,val3"}
                ]
            }
        },
        "headers": {
            "host": {"value": "video.example.com"},
            "user-agent": {"value": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0)
 Gecko/20100101 Firefox/83.0"},
            "accept": {
                "value": "application/json",
                "multiValue": [
                    {"value": "application/json"},
                    {"value": "application/xml"},
                    {"value": "text/html"}
                ]
```

```
            },
            "accept-language": {"value": "en-GB,en;q=0.5"},
            "accept-encoding": {"value": "gzip, deflate, br"},
            "origin": {"value": "https://website.example.com"},
            "referer": {"value": "https://website.example.com/videos/12345678?
action=play"},
            "cloudfront-viewer-country": {"value": "GB"}
        },
        "cookies": {
            "Cookie1": {"value": "value1"},
            "Cookie2": {"value": "value2"},
            "cookie_consent": {"value": "true"},
            "cookiemv": {
                "value": "value3",
                "multiValue": [
                    {"value": "value3"},
                    {"value": "value4"}
                ]
            }
        }
    },
    "response": {
        "statusCode": 200,
        "statusDescription": "OK",
        "headers": {
            "date": {"value": "Mon, 04 Apr 2021 18:57:56 GMT"},
            "server": {"value": "gunicorn/19.9.0"},
            "access-control-allow-origin": {"value": "*"},
            "access-control-allow-credentials": {"value": "true"},
            "content-type": {"value": "application/json"},
            "content-length": {"value": "701"}
        },
        "cookies": {
            "ID": {
                "value": "id1234",
                "attributes": "Expires=Wed, 05 Apr 2021 07:28:00 GMT"
            },
            "Cookie1": {
                "value": "val1",
                "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr
 2021 07:28:00 GMT",
                "multiValue": [
                    {
                        "value": "val1",
```

```
                        "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed,
 05 Apr 2021 07:28:00 GMT"
                    },
                    {
                        "value": "val2",
                        "attributes": "Path=/cat; Domain=example.com; Expires=Wed, 10
 Jan 2021 07:28:00 GMT"
                    }
                ]
            }
        }
    }
}
```

JavaScript 函數的執行階段 CloudFront 功能

Amazon CloudFront 函數 JavaScript 運行時環境符合 ECMAScript（ES）5.1 版，並且還支持 ES 版本 6 到 12 的某些功能。

我們建議您使用執行時間 2.0，以獲得最新功能。請注意，與 1.0 相比，執行時間 2.0 有下列變更：

- 可使用緩衝區模組方法。
- 以下非標準字串原型方法無法使用：
  - `String.prototype.bytesFrom()`
  - `String.prototype.fromBytes()`
  - `String.prototype.fromUTF8()`
  - `String.prototype.toBytes()`
  - `String.prototype.toUTF8()`
- 密碼編譯模組有以下變更：
  - `hash.digest()` - 如果未提供編碼，則傳回類型變更為 `Buffer`
  - `hmac.digest()` - 如果未提供編碼，則傳回類型變更為 `Buffer`
- 其他新功能在 JavaScript 函數的執行階段 2.0 CloudFront 功能 中說明。

主題

- JavaScript 函數的執行階段 1.0 CloudFront 功能
- JavaScript 函數的執行階段 2.0 CloudFront 功能

JavaScript 函數的執行階段 1.0 CloudFront 功能

CloudFront 函數 JavaScript 運行時環境與 ECMAScript（ES）5.1 版本兼容，並且還支持 ES 版本 6 到 9 的某些功能。它還提供了一些不屬於 ES 規格的非標準方法。下列主題列出所有支援的語言功能。

主題

核心功能

支援 ES 的以下核心功能。

類型

支援所有 ES 5.1 類型。這包括布林值、數字、字串、物件、陣列、函數、函數建構子和常規表達式。

運算子

支援所有 ES 5.1 運算子。

支援 ES 7 指數運算子 (**)。

聲明

> **Note**
>
> 不支援 `const` 和 `let` 陳述式。

支援下列 ES 5.1 陳述式：

- `break`
- `catch`

- continue
- do-while
- else
- finally
- for
- for-in
- if
- return
- switch
- throw
- try
- var
- while
- 標記的陳述式

文字

支援 ES 6 範本文字：多行字串、表達式插補和巢狀範本。

函數

支援所有 ES 5.1 函數功能。

支援 ES 6 箭頭函數，支援 ES 6 剩餘參數語法。

Unicode

來源文字和字串常值可以包含 Unicode 編碼的字元。也支援由六個字元 (例如 \uXXXX) 組成的 Unicode 字碼指標逸出序列。

嚴格模式

函數按預設會在嚴格模式下運作，因此您不需要在函數程式碼中新增 `use strict` 陳述式。無法對此進行變更。

基本物件

支援 ES 的以下基本物件。

物件

支援物件上的以下 ES 5.1 方法：

- create(不含屬性清單)
- defineProperties
- defineProperty
- freeze
- getOwnPropertyDescriptor
- getOwnPropertyNames
- getPrototypeOf
- hasOwnProperty
- isExtensible
- isFrozen
- prototype.isPrototypeOf
- isSealed
- keys
- preventExtensions
- prototype.propertyIsEnumerable
- seal
- prototype.toString
- prototype.valueOf

支援物件上的以下 ES 6 方法：

- assign
- is
- prototype.setPrototypeOf

支援物件上的以下 ES 8 方法：

- entries
- values

字串

支援以下針對字串的 ES 5.1 方法：

- `fromCharCode`
- `prototype.charAt`
- `prototype.concat`
- `prototype.indexOf`
- `prototype.lastIndexOf`
- `prototype.match`
- `prototype.replace`
- `prototype.search`
- `prototype.slice`
- `prototype.split`
- `prototype.substr`
- `prototype.substring`
- `prototype.toLowerCase`
- `prototype.trim`
- `prototype.toUpperCase`

支援字串上的以下 ES 6 方法：

- `fromCodePoint`
- `prototype.codePointAt`
- `prototype.endsWith`
- `prototype.includes`
- `prototype.repeat`
- `prototype.startsWith`

支援字串上的以下 ES 8 方法：

- `prototype.padStart`
- `prototype.padEnd`

支援字串上的以下 ES 9 方法：

- prototype.trimStart
- prototype.trimEnd

支援字串上的以下非標準方法：

- prototype.bytesFrom(*array* | *string*, *encoding*)

  從八位元陣列或編碼字串建立一個位元組字串。字串編碼選項為 hex、base64 和 base64url。

- prototype.fromBytes(*start*[, *end*])

  從位元組字串建立 Unicode 字串，其中每個位元組都會以對應的 Unicode 字碼指標取代。

- prototype.fromUTF8(*start*[, *end*])

  從 UTF-8 編碼的位元組字串建立一個 Unicode 字串。如果編碼不正確，則返回 null。

- prototype.toBytes(start[, end])

  從 Unicode 字串建立一個位元組字串。所有字元均必須在 [0,255] 範圍內。如果不在此範圍內，則返回 null。

- prototype.toUTF8(start[, *end*])

  從一個 Unicode 字串建立一個 UTF-8 編碼的位元組字串。

數字

支援數字上的所有 ES 5.1 方法。

支援數字上的以下 ES 6 方法：

- isFinite
- isInteger
- isNaN
- isSafeInteger
- parseFloat
- parseInt
- prototype.toExponential
- prototype.toFixed
- prototype.toPrecision

- EPSILON
- MAX_SAFE_INTEGER
- MAX_VALUE
- MIN_SAFE_INTEGER
- MIN_VALUE
- NEGATIVE_INFINITY
- NaN
- POSITIVE_INFINITY

內建物件

支援 ES 的以下內建物件。

數學

支援所有 ES 5.1 數學方法。

> **Note**
>
> 在「 CloudFront 函數」執行階段環境中，`Math.random()`實作會使用`arc4random`內建的 OpenBSD 以及函數執行時間戳記。

支援以下 ES 6 數學方法：

- acosh
- asinh
- atanh
- cbrt
- clz32
- cosh
- expm1
- fround
- hypot

- imul
- log10
- log1p
- log2
- sign
- sinh
- tanh
- trunc
- E
- LN10
- LN2
- LOG10E
- LOG2E
- PI
- SQRT1_2
- SQRT2

日期

支援所有 ES 5.1 Date 功能。

> **Note**
>
> 基於安全考量，在單一函數執行的生命週期內，Date 始終返回相同的值 (函數的開始時間)。如需詳細資訊，請參閱 限制功能。

函數

支援 apply、bind 和 call 方法。

不支援函數建構子。

常規表達式

支援所有 ES 5.1 常規表達式功能。常規表達式語言與 Perl 相容。支援 ES 9 命名的擷取群組。

JSON

支援所有 ES 5.1 JSON 功能，包括 `parse` 和 `stringify`。

Array

支援陣列上的以下 ES 5.1 方法：

- `isArray`
- `prototype.concat`
- `prototype.every`
- `prototype.filter`
- `prototype.forEach`
- `prototype.indexOf`
- `prototype.join`
- `prototype.lastIndexOf`
- `prototype.map`
- `prototype.pop`
- `prototype.push`
- `prototype.reduce`
- `prototype.reduceRight`
- `prototype.reverse`
- `prototype.shift`
- `prototype.slice`
- `prototype.some`
- `prototype.sort`
- `prototype.splice`
- `prototype.unshift`

支援陣列上的以下 ES 6 方法：

- `of`
- `prototype.copyWithin`
- `prototype.fill`

- prototype.find
- prototype.findIndex

支援陣列上的以下 ES 7 方法：

- prototype.includes

類型陣列

支援以下 ES 6 類型陣列：

- Int8Array
- Uint8Array
- Uint8ClampedArray
- Int16Array
- Uint16Array
- Int32Array
- Uint32Array
- Float32Array
- Float64Array
- prototype.copyWithin
- prototype.fill
- prototype.join
- prototype.set
- prototype.slice
- prototype.subarray
- prototype.toString

ArrayBuffer

支援 ArrayBuffer 上的以下方法：

- prototype.isView
- prototype.slice

Promise

支援 Promise 上的以下方法：

- reject
- resolve
- prototype.catch
- prototype.finally
- prototype.then

加密

密碼編譯模組提供標準雜湊和雜湊型訊息身分驗證碼 (HMAC) 協助程式。您可以使用 require('crypto') 加載模組。此模組會公開下列方法，其行為與 Node.js 對應方法完全相同：

- createHash(*algorithm*)
- hash.update(*data*)
- hash.digest([*encoding*])
- createHmac(*algorithm*, *secret key*)
- hmac.update(*data*)
- hmac.digest([*encoding*])

如需詳細資訊，請參閱內建模組一節中的 加密 (雜湊和 HMAC)。

主控台

這是一個用於偵錯的協助程式物件。它僅支援 log() 方法以記錄日誌訊息。

> **Note**
>
> CloudFront 函數不支援逗號語法，例如console.log('a', 'b'). 請改用格console.log('a' + ' ' + 'b')式。

錯誤類型

支援以下錯誤物件：

- Error
- EvalError
- InternalError
- MemoryError

- RangeError
- ReferenceError
- SyntaxError
- TypeError
- URIError

全域變數

支援 globalThis 物件。

支援以下 ES 5.1 全局函數：

- decodeURI
- decodeURIComponent
- encodeURI
- encodeURIComponent
- isFinite
- isNaN
- parseFloat
- parseInt

支援以下全局常數：

- NaN
- Infinity
- undefined

內建模組

支援以下內建模組：

模組

- 加密 (雜湊和 HMAC)
- 查詢字串

加密 (雜湊和 HMAC)

密碼編譯模組 (crypto) 提供標準雜湊和雜湊型訊息身分驗證碼 (HMAC) 協助程式。您可以使用 `require('crypto')` 加載模組。此模組提供下列方法，其行為與 Node.js 對應方法完全相同：

雜湊方法

`crypto.createHash(`*`algorithm`*`)`

建立並傳回雜湊物件，藉助此物件，您可以使用給定的演算法產生雜湊摘要：md5、sha1 或 sha256。

`hash.update(`*`data`*`)`

使用給定的 data 更新雜湊內容。

`hash.digest([`*`encoding`*`])`

計算使用 hash.update() 傳遞的所有資料的摘要。編碼可以是 hex、base64 或 base64url。

HMAC 方法

`crypto.createHmac(`*`algorithm, secret key`*`)`

建立並返回使用給定 algorithm 和 secret key 的 HMAC 物件。演算法可以是 md5、sha1 或 sha256。

`hmac.update(`*`data`*`)`

使用給定的 data 更新 HMAC 內容。

`hmac.digest([`*`encoding`*`])`

計算使用 hmac.update() 傳遞的所有資料的摘要。編碼可以是 hex、base64 或 base64url。

查詢字串

> **Note**
>
> CloudFront 函數事件對象會自動為您解析 URL 查詢字符串。這意味著在大多數情況下，您不需要使用此模組。

查詢字串模組 (querystring) 提供剖析和格式化 URL 查詢字串的方法。您可以使用 `require('querystring')` 加載模組。此模組提供下列方法。

`querystring.escape(`*`string`*`)`

URL 編碼給定的 `string`，傳回逸出的查詢字串。該方法由 `querystring.stringify()` 使用，並且不應直接使用。

`querystring.parse(`*`string`*`[, `*`separator`*`[, `*`equal`*`[, `*`options`*`]]])`

剖析查詢字串 (string) 並傳回物件。

`separator` 參數是用來分隔查詢字串中的鍵/值對的子字串。其在預設情況下為 &。

`equal` 參數是用來分隔查詢字串中的鍵和值的子字串。其在預設情況下為 =。

`options` 參數是具有下列鍵的物件：

`decodeURIComponent` *`function`*

解碼查詢字串中百分比編碼字元的函數。其在預設情況下為 `querystring.unescape()`。

`maxKeys` *`number`*

要剖析的鍵的最大數量。其在預設情況下為 `1000`。使用 `0` 的值移除計數鍵的限制。

根據預設，查詢字串中的百分比編碼字元會假設為使用 UTF-8 編碼。無效的 UTF-8 序列會被取代為 U+FFFD 取代字元。

例如，對於下列查詢字串：

```
'name=value&abc=xyz&abc=123'
```

`querystring.parse()` 的返回值是：

```
{
name: 'value',
abc: ['xyz', '123']
}
```

`querystring.decode()` 是 的別名。`querystring.parse()`

`querystring.stringify(`*`object`*`[, `*`separator`*`[, `*`equal`*`[, `*`options`*`]]])`

序列化 `object` 並傳回查詢字串。

separator 參數是用來分隔查詢字串中的鍵/值對的子字串。其在預設情況下為 &。

equal 參數是用來分隔查詢字串中的鍵和值的子字串。其在預設情況下為 =。

options 參數是具有下列鍵的物件：

encodeURIComponent *function*

用於將 URL 不安全字元轉換為查詢字串中的百分比編碼的函數。其在預設情況下為 querystring.escape()。

根據預設，在查詢字串中需要百分比編碼的字元會編碼為 UTF-8。若要使用不同的編碼，請指定 encodeURIComponent 選項。

例如，對於以下程式碼：

```
querystring.stringify({ name: 'value', abc: ['xyz', '123'], anotherName: '' });
```

返回值是：

```
'name=value&abc=xyz&abc=123&anotherName='
```

querystring.encode() 是 querystring.stringify() 的別名。

querystring.unescape(*string*)

解碼給定 string 中的 URL 百分比編碼字元，傳回未逸出的查詢字串。此方法由 querystring.parse() 使用，並且不應直接使用。

限制功能

基於安全考量，下列 JavaScript 語言功能不受支援或受限制。

動態程式碼評估

不支援動態程式碼評估。如果嘗試此評估，eval() 和 Function 建構子都會丟出錯誤。例如，const sum = new Function('a', 'b', 'return a + b') 丟出錯誤。

計時器

不支援 setTimeout()、setImmediate() 和 clearTimeout() 函數。在函數執行期間未推遲或產生任何佈建。您的函數必須同步執行方可完成。

日期和時間戳記

基於安全考量，無法存取高解析度計時器。查詢當前時間的所有 Date 方法始終在單個函數執行的生命週期內返回相同的值。返回的時間戳記是函數開始執行的時間。因此，您無法測量函數中的經過時間。

檔案系統存取

沒有檔案系統存取權。例如，沒有類似 Node.js 中的檔案系統存取 fs 模組。

網路存取

不支援網路呼叫。例如，不支援 XHR、HTTP(S) 和通訊端。

JavaScript 函數的執行階段 2.0 CloudFront 功能

CloudFront 函數 JavaScript 運行時環境與 ECMAScript（ES）5.1 版本兼容，並且還支持 ES 版本 6 到 12 的某些功能。它還提供了一些不屬於 ES 規格的非標準方法。以下主題列出此執行期支援的所有功能。

主題

- 核心功能
- 基本物件
- 內建物件
- 錯誤類型
- 全域變數
- 內建模組
- 限制功能

核心功能

支援 ES 的以下核心功能。

類型

支援所有 ES 5.1 類型。這包括布林值、數字、字串、物件、陣列、函數和常規表達式。

運算子

支援所有 ES 5.1 運算子。

支援 ES 7 指數運算子 (**)。

陳述式

支援下列 ES 5.1 陳述式：

- break
- catch
- continue
- do-while
- else
- finally
- for
- for-in
- if
- label
- return
- switch
- throw
- try
- var
- while

支援以下 ES 6 陳述式：

- async
- await
- const
- let

> **Note**
>
> async、awaitconst、和let是 JavaScript 執行階段 2.0 中的新功能。

文字

支援 ES 6 範本文字：多行字串、表達式插補和巢狀範本。

函數

支援所有 ES 5.1 函數功能。

支援 ES 6 箭頭函數，支援 ES 6 剩餘參數語法。

Unicode

來源文字和字串常值可以包含 Unicode 編碼的字元。也支援由六個字元 (例如 \uXXXX) 組成的 Unicode 字碼指標逸出序列。

嚴格模式

函數按預設會在嚴格模式下運作，因此您不需要在函數程式碼中新增 `use strict` 陳述式。無法對此進行變更。

基本物件

支援 ES 的以下基本物件。

物件

支援物件上的以下 ES 5.1 方法：

- `Object.create()`(不含屬性清單)
- `Object.defineProperties()`
- `Object.defineProperty()`
- `Object.freeze()`
- `Object.getOwnPropertyDescriptor()`
- `Object.getOwnPropertyDescriptors()`
- `Object.getOwnPropertyNames()`
- `Object.getPrototypeOf()`
- `Object.isExtensible()`
- `Object.isFrozen()`
- `Object.isSealed()`

- Object.keys()
- Object.preventExtensions()
- Object.seal()

支援物件上的以下 ES 6 方法：

- Object.assign()

支援物件上的以下 ES 8 方法：

- Object.entries()
- Object.values()

支援以下針對物件的 ES 5.1 原型方法：

- Object.prototype.hasOwnProperty()
- Object.prototype.isPrototypeOf()
- Object.prototype.propertyIsEnumerable()
- Object.prototype.toString()
- Object.prototype.valueOf()

支援以下針對物件的 ES 6 原型方法：

- Object.prototype.is()
- Object.prototype.setPrototypeOf()

字串

支援以下針對字串的 ES 5.1 方法：

- String.fromCharCode()

支援以下針對字串的 ES 6 方法：

- String.fromCodePoint()

支援以下針對字串的 ES 5.1 原型方法：

- String.prototype.charAt()
- String.prototype.concat()
- String.prototype.indexOf()
- String.prototype.lastIndexOf()

- String.prototype.match()
- String.prototype.replace()
- String.prototype.search()
- String.prototype.slice()
- String.prototype.split()
- String.prototype.substr()
- String.prototype.substring()
- String.prototype.toLowerCase()
- String.prototype.trim()
- String.prototype.toUpperCase()

支援以下針對字串的 ES 6 原型方法：

- String.prototype.codePointAt()
- String.prototype.endsWith()
- String.prototype.includes()
- String.prototype.repeat()
- String.prototype.startsWith()

支援以下針對字串的 ES 8 原型方法：

- String.prototype.padStart()
- String.prototype.padEnd()

支援以下針對字串的 ES 9 原型方法：

- String.prototype.trimStart()
- String.prototype.trimEnd()

支援以下針對字串的 ES 12 原型方法：

- String.prototype.replaceAll()

> **Note**
>
> String.prototype.replaceAll()在 JavaScript 執行階段 2.0 中是新的。

Number

支援所有 ES 5 數字。

支援以下 ES 6 數字屬性：

- `Number.EPSILON`
- `Number.MAX_SAFE_INTEGER`
- `Number.MIN_SAFE_INTEGER`
- `Number.MAX_VALUE`
- `Number.MIN_VALUE`
- `Number.NaN`
- `Number.NEGATIVE_INFINITY`
- `Number.POSITIVE_INFINITY`

支援數字上的以下 ES 6 方法：

- `Number.isFinite()`
- `Number.isInteger()`
- `Number.isNaN()`
- `Number.isSafeInteger()`
- `Number.parseInt()`
- `Number.parseFloat()`

支援以下針對數字的 ES 5.1 原型方法：

- `Number.prototype.toExponential()`
- `Number.prototype.toFixed()`
- `Number.prototype.toPrecision()`

支援 ES 12 數字分隔符號。

> **Note**
>
> ES 12 數字分隔符是 JavaScript 運行時 2.0 中的新功能。

內建物件

支援 ES 的以下內建物件。

數學

支援所有 ES 5.1 數學方法。

> **Note**
>
> 在「 CloudFront 函數」執行階段環境中，`Math.random()`實作會使用`arc4random`內建的 OpenBSD 以及函數執行時間戳記。

支援以下 ES 6 數學屬性：

- `Math.E`
- `Math.LN10`
- `Math.LN2`
- `Math.LOG10E`
- `Math.LOG2E`
- `Math.PI`
- `Math.SQRT1_2`
- `Math.SQRT2`

支援以下 ES 6 數學方法：

- `Math.abs()`
- `Math.acos()`
- `Math.acosh()`
- `Math.asin()`
- `Math.asinh()`
- `Math.atan()`
- `Math.atan2()`
- `Math.atanh()`
- `Math.cbrt()`

- Math.ceil()
- Math.clz32()
- Math.cos()
- Math.cosh()
- Math.exp()
- Math.expm1()
- Math.floor()
- Math.fround()
- Math.hypot()
- Math.imul()
- Math.log()
- Math.log1p()
- Math.log2()
- Math.log10()
- Math.max()
- Math.min()
- Math.pow()
- Math.random()
- Math.round()
- Math.sign()
- Math.sinh()
- Math.sin()
- Math.sqrt()
- Math.tan()
- Math.tanh()
- Math.trunc()

日期

支援所有 ES 5.1 Date 功能。

> **Note**
>
> 基於安全考量，在單一函數執行的生命週期內，Date 始終返回相同的值 (函數的開始時間)。如需詳細資訊，請參閱 限制功能。

函數

支援以下 ES 5.1 原型方法：

- `Function.prototype.apply()`
- `Function.prototype.bind()`
- `Function.prototype.call()`

不支援函數建構子。

常規表達式

支援所有 ES 5.1 常規表達式功能。常規表達式語言與 Perl 相容。

支援以下 ES 5.1 原型存取子屬性：

- `RegExp.prototype.global`
- `RegExp.prototype.ignoreCase`
- `RegExp.protoype.multiline`
- `RegExp.protoype.source`
- `RegExp.prototype.sticky`
- `RegExp.prototype.flags`

> **Note**
>
> `RegExp.prototype.sticky`並且`RegExp.prototype.flags`是 JavaScript運行時 2.0 中的新功能。

支援以下 ES 5.1 原型方法：

- `RegExp.prototype.exec()`
- `RegExp.prototype.test()`
- `RegExp.prototype.toString()`

- `RegExp.prototype[@@replace]()`
- `RegExp.prototype[@@split]()`

> **Note**
>
> `RegExp.prototype[@@split]()`在 JavaScript執行階段 2.0 中是新的。

支援以下 ES 5.1 執行個體屬性：

- `lastIndex`

支援 ES 9 命名的擷取群組。

JSON

支援以下 ES 5.1 方法：

- `JSON.parse()`
- `JSON.stringify()`

陣列

支援陣列上的以下 ES 5.1 方法：

- `Array.isArray()`

支援陣列上的以下 ES 6 方法：

- `Array.of()`

支援以下 ES 5.1 原型方法：

- `Array.prototype.concat()`
- `Array.prototype.every()`
- `Array.prototype.filter()`
- `Array.prototype.forEach()`
- `Array.prototype.indexOf()`
- `Array.prototype.join()`
- `Array.prototype.lastIndexOf()`
- `Array.prototype.map()`
- `Array.prototype.pop()`

- Array.prototype.push()
- Array.prototype.reduce()
- Array.prototype.reduceRight()
- Array.prototype.reverse()
- Array.prototype.shift()
- Array.prototype.slice()
- Array.prototype.some()
- Array.prototype.sort()
- Array.prototype.splice()
- Array.prototype.unshift()

支援以下 ES 6 原型方法

- Array.prototype.copyWithin()
- Array.prototype.fill()
- Array.prototype.find()
- Array.prototype.findIndex()

支援以下 ES 7 原型方法：

- Array.prototype.includes()

類型陣列

支援以下 ES 6 類型陣列建構子：

- Float32Array
- Float64Array
- Int8Array
- Int16Array
- Int32Array
- Uint8Array
- Uint8ClampedArray
- Uint16Array
- Uint32Array

支援以下 ES 6 方法：

- `TypedArray.from()`
- `TypedArray.of()`

> **Note**
>
> `TypedArray.from()`並且`TypedArray.of()`是 JavaScript 運行時 2.0 中的新功能。

支援以下 ES 6 原型方法：

- `TypedArray.prototype.copyWithin()`
- `TypedArray.prototype.every()`
- `TypedArray.prototype.fill()`
- `TypedArray.prototype.filter()`
- `TypedArray.prototype.find()`
- `TypedArray.prototype.findIndex()`
- `TypedArray.prototype.forEach()`
- `TypedArray.prototype.includes()`
- `TypedArray.prototype.indexOf()`
- `TypedArray.prototype.join()`
- `TypedArray.prototype.lastIndexOf()`
- `TypedArray.prototype.map()`
- `TypedArray.prototype.reduce()`
- `TypedArray.prototype.reduceRight()`
- `TypedArray.prototype.reverse()`
- `TypedArray.prototype.some()`
- `TypedArray.prototype.set()`
- `TypedArray.prototype.slice()`
- `TypedArray.prototype.sort()`
- `TypedArray.prototype.subarray()`
- `TypedArray.prototype.toString()`

> **Note**
>
> `TypedArray.prototype.every()`、、`TypedArray.prototype.fill()`、`TypedArray.p`
> 和`TypedArray.prototype.some()`是 JavaScript 執行階段 2.0 中的新功能。

ArrayBuffer

支援下列 ES 6 方法： ArrayBuffer

- `isView()`

支持以下 ES 6 原型方法： ArrayBuffer

- `ArrayBuffer.prototype.slice()`

Promise

支援以下針對 Promise 的 ES 6 方法：

- `Promise.all()`
- `Promise.allSettled()`
- `Promise.any()`
- `Promise.reject()`
- `Promise.resolve()`
- `Promise.race()`

> **Note**
>
> `Promise.all()`、`Promise.allSettled()`Promise.any()、
> 和`Promise.race()`是 JavaScript 執行階段 2.0 中的新功能。

支援以下針對 Promise 的 ES 6 原型方法：

- `Promise.prototype.catch()`
- `Promise.prototype.finally()`
- `Promise.prototype.then()`

DataView

支援以下 ES 6 原型方法：

- DataView.prototype.getFloat32()
- DataView.prototype.getFloat64()
- DataView.prototype.getInt16()
- DataView.prototype.getInt32()
- DataView.prototype.getInt8()
- DataView.prototype.getUint16()
- DataView.prototype.getUint32()
- DataView.prototype.getUint8()
- DataView.prototype.setFloat32()
- DataView.prototype.setFloat64()
- DataView.prototype.setInt16()
- DataView.prototype.setInt32()
- DataView.prototype.setInt8()
- DataView.prototype.setUint16()
- DataView.prototype.setUint32()
- DataView.prototype.setUint8()

> **Note**
>
> 所有數據視圖 ES 6 原型方法在 JavaScript運行時 2.0 中都是新的。

符號

支援以下 ES 6 方法：

- Symbol.for()
- Symbol.keyfor()

> **Note**
>
> 所有符號 ES 6 方法在 JavaScript 運行時 2.0 中都是新的。

文字解碼器

支援以下原型方法：

- TextDecoder.prototype.decode()

支援以下原型存取子屬性：

- TextDecoder.prototype.encoding
- TextDecoder.prototype.fatal
- TextDecoder.prototype.ignoreBOM

文字編碼器

支援以下原型方法：

- TextEncoder.prototype.encode()
- TextEncoder.prototype.encodeInto()

錯誤類型

支援以下錯誤物件：

- Error
- EvalError
- InternalError
- RangeError
- ReferenceError
- SyntaxError
- TypeError
- URIError

全域變數

支援 globalThis 物件。

支援以下 ES 5.1 全局函數：

- decodeURI()
- decodeURIComponent()
- encodeURI()
- encodeURIComponent()

- isFinite()
- isNaN()
- parseFloat()
- parseInt()

支援以下 ES 6 全域函數：

- atob()
- btoa()

> **Note**
>
> atob()並且btoa()是 JavaScript 運行時 2.0 中的新功能。

支援以下全局常數：

- NaN
- Infinity
- undefined
- arguments

內建模組

支援以下內建模組：

模組

- 緩衝區
- 查詢字串
- 加密

緩衝區

此模組提供以下方法：

- Buffer.alloc(size[, fill[, encoding]])

配置 Buffer。

- size：緩衝區大小。輸入整數。
- fill：選用。輸入字串、Buffer、Uint8Array 或整數。預設值為 0。
- encoding：選用。當 fill 為字串，請輸入以下其中一項：utf8、hex、base64、base64url。預設值為 utf8。

- Buffer.allocUnsafe(size)

  配置一個未初始化的 Buffer。

  - size：輸入整數。

- Buffer.byteLength(value[, encoding])

  返回值的長度，以位元組為單位。

  - value：字串、、Buffer TypedArray、資料檢視或陣列緩衝區。
  - encoding：選用。當 value 為字串，請輸入以下其中一項：utf8、hex、base64、base64url。預設值為 utf8。

- Buffer.compare(buffer1, buffer2)

  比較兩個 Buffer 以協助對陣列進行排序。如果兩者相同則傳回 0，如果 buffer1 在前面則傳回 -1，或如果 buffer2 在前面則傳回 1。

  - buffer1：輸入 Buffer。
  - buffer2：輸入不同的 Buffer。

- Buffer.concat(list[, totalLength])

  連接多個 Buffer。如果沒有則傳回 0。最多傳回 totalLength。

  - list：輸入 Buffer 的清單。請注意，這將被截斷為 totalLength。
  - totalLength：選用。輸入不帶正負號的整數。如果清單為空白，則使用清單中 Buffer 執行個體的總和。

- Buffer.from(array)

  從陣列建立 Buffer。

  - array: 輸入從 0 到 255 的位元組陣列。

- Buffer.from(arrayBuffer, byteOffset[, length]))

  從 arrayBuffer 建立檢視，從偏移值 byteOffset 開始，長度為 length。

  - arrayBuffer：輸入 Buffer 陣列。
  - byteOffset：輸入整數。
  - length：選用。輸入整數。
- Buffer.from(buffer)

  建立 Buffer 的複本。

  - buffer：輸入 Buffer。
- Buffer.from(object[, offsetOrEncoding[, length]])

  從物件建立 Buffer。如果 valueOf() 不等於物件，則傳回 Buffer.from(object.valueOf(), offsetOrEncoding, length)。

  - object：輸入物件。
  - offsetOrEncoding：選用。輸入整數或編碼字串。
  - length：選用。輸入整數。
- Buffer.from(string[, encoding])

  從字串建立一個 Buffer。

  - string：輸入字串。
  - encoding：選用。輸入以下其中之一：utf8、hex、base64、base64url。預設值為 utf8。
- Buffer.isBuffer(object)

  檢查 object 是否為緩衝區。傳回 true 或 false。

  - object：輸入物件。
- Buffer.isEncoding(encoding)

  檢查是否支援 encoding。傳回 true 或 false。

  - encoding：選用。輸入以下其中之一：utf8、hex、base64、base64url。預設值為 utf8。

此模組提供以下緩衝區原型方法：

- Buffer.prototype.compare(target[, targetStart[, targetEnd[, sourceStart[, sourceEnd]]]])

  將 Buffer 與目標比較。如果兩者相同則傳回 0，如果 buffer 在前面則傳回 1，或如果 target 在前面則傳回 -1。

  - target：輸入 Buffer。
  - targetStart：選用。輸入整數。預設值為 0。
  - targetEnd：選用。輸入整數。預設值為 target 長度。
  - sourceStart：選用。輸入整數。預設值為 0。
  - sourceEnd：選用。輸入整數。預設值為 Buffer 長度。
- Buffer.prototype.copy(target[, targetStart[, sourceStart[, sourceEnd]]])

  將緩衝區複製到 target。
  - target：輸入 Buffer 或 Uint8Array。
  - targetStart：選用。輸入整數。預設值為 0。
  - sourceStart：選用。輸入整數。預設值為 0。
  - sourceEnd：選用。輸入整數。預設值為 Buffer 長度。
- Buffer.prototype.equals(otherBuffer)

  將 Buffer 與 otherBuffer 比較。傳回 true 或 false。
  - otherBuffer：輸入字串。
- Buffer.prototype.fill(value[, offset[, end][, encoding])

  以 value 填入 Buffer。
  - value：輸入字串、Buffer 或整數。
  - offset：選用。輸入整數。
  - end：選用。輸入整數。
  - encoding：選用。輸入以下其中之一：utf8、hex、base64、base64url。預設值為 utf8。
- Buffer.prototype.includes(value[, byteOffset][, encoding])

  搜尋 Buffer 中的 value。傳回 true 或 false。
  - value：輸入字串、Buffer、Uint8Array、或整數。
  - byteOffset：選用。輸入整數。
  - encoding：選用。輸入以下其中之一：utf8、hex、base64、base64url。預設值為 utf8。
- Buffer.prototype.indexOf(value[, byteOffset][, encoding])

  首先搜尋 Buffer 中的第一個 value。如果找到了，則傳回 index；如果找不到，則傳回 -1。
  - value: 輸入字串、Buffer、Unit8Array 或 0 到 255 之間的整數。

  - byteOffset：選用。輸入整數。
  - encoding：選用。如果 value 是字串，請輸入以下其中一項：utf8、hex、base64、base64url。預設值為 utf8。
- Buffer.prototype.lastIndexOf(value[, byteOffset][, encoding])

  搜尋 Buffer 中的最後一個 value。如果找到了，則傳回 index；如果找不到，則傳回 -1。
  - value: 輸入字串、Buffer、Unit8Array 或 0 到 255 之間的整數。
  - byteOffset：選用。輸入整數。
  - encoding：選用。如果 value 是字串，請輸入以下其中一項：utf8、hex、base64、base64url。預設值為 utf8。
- Buffer.prototype.readInt8(offset)

  從 Buffer 的偏移值 offset 讀取 Int8。
  - offset：輸入整數。
- Buffer.prototype.readIntBE(offset, byteLength)

  從 Buffer 的偏移值 offset 讀取大端序 Int。
  - offset：輸入整數。
  - byteLength：選用。輸入從 1 到 6 的整數。
- Buffer.prototype.readInt16BE(offset)

  從 Buffer 的偏移值 offset 讀取大端序 Int16。
  - offset：輸入整數。
- Buffer.prototype.readInt32BE(offset)

  從 Buffer 的偏移值 offset 讀取大端序 Int32。
  - offset：輸入整數。
- Buffer.prototype.readIntLE(offset, byteLength)

  從 Buffer 的偏移值 offset 讀取小端序 Int。
  - offset：輸入整數。
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.readInt16LE(offset)

  從 Buffer 的偏移值 offset 讀取小端序 Int16。

  - offset：輸入整數。
- Buffer.prototype.readInt32LE(offset)

  從 Buffer 的偏移值 offset 讀取小端序 Int32。

  - offset：輸入整數。
- Buffer.prototype.readUInt8(offset)

  從 Buffer 的偏移值 offset 讀取 UInt8。

  - offset：輸入整數。
- Buffer.prototype.readUIntBE(offset, byteLength)

  從 Buffer 的偏移值 offset 讀取大端序 UInt。

  - offset：輸入整數。
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.readUInt16BE(offset)

  從 Buffer 的偏移值 offset 讀取大端序 UInt16。
- • offset：輸入整數。
- Buffer.prototype.readUInt32BE(offset)

  從 Buffer 的偏移值 offset 讀取大端序 UInt32。

  - offset：輸入整數。
- Buffer.prototype.readUIntLE(offset, byteLength)

  從 Buffer 的偏移值 offset 讀取小端序 UInt。

  - offset：輸入整數。
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.readUInt16LE(offset)

  從 Buffer 的偏移值 offset 讀取小端序 UInt16。

  - offset：輸入整數。
- Buffer.prototype.readUInt32LE(offset)

  從 Buffer 的偏移值 offset 讀取小端序 UInt32。

  - offset：輸入整數。

- Buffer.prototype.readDoubleBE([offset])

  從 Buffer 的偏移值 offset 讀取 64 位元大端序雙精度浮點數。

  - offset：選用。輸入整數。
- Buffer.prototype.readDoubleLE([offset])

  從 Buffer 的偏移值 offset 讀取 64 位元小端序雙精度浮點數。

  - offset：選用。輸入整數。
- Buffer.prototype.readFloatBE([offset])

  從 Buffer 的偏移值 offset 讀取 32 位元大端序浮點數。

  - offset：選用。輸入整數。
- Buffer.prototype.readFloatLE([offset])

  從 Buffer 的偏移值 offset 讀取 32 位元小端序浮點數。

  - offset：選用。輸入整數。
- Buffer.prototype.subarray([start[, end]])

  傳回 Buffer 的副本，並使用新的 start 和 end 偏移和裁剪。

  - start：選用。輸入整數。預設值為 0。
  - end：選用。輸入整數。預設值為緩衝區長度。
- Buffer.prototype.swap16()

  交換 Buffer 陣列的位元組順序，將其視為 16 位元數字的陣列。Buffer 的長度必須是 2 的倍數，否則您將收到錯誤訊息。
- Buffer.prototype.swap32()

  交換 Buffer 陣列的位元組順序，將其視為 32 位元數字的陣列。Buffer 的長度必須是 4 的倍數，否則您將收到錯誤訊息。
- Buffer.prototype.swap64()

  交換 Buffer 陣列的位元組順序，將其視為 64 位元數字的陣列。Buffer 的長度必須是 8 的倍數，否則您將收到錯誤訊息。
- Buffer.prototype.toJSON()

  以 JSON 格式傳回 Buffer。

- Buffer.prototype.toString([encoding[, start[, end]]])

  將 Buffer 從 start 到 end 轉換為編碼字串。

  - encoding：選用。輸入以下其中之一：utf8、hex、base64 或 base64url。預設值為 utf8。
  - start：選用。輸入整數。預設值為 0。
  - end：選用。輸入整數。預設值為緩衝區長度。
- Buffer.prototype.write(string[, offset[, length]][, encoding])

  如果空間足夠，則將編碼 string 寫入 Buffer，如果空間不足，則寫入被截斷的 string。

  - string：輸入字串。
  - offset：選用。輸入整數。預設值為 0。
  - length：選用。輸入整數。預設值是字串的長度。
  - encoding：選用。選擇性地輸入以下其中一項：utf8、hex、base64 或 base64url。預設值為 utf8。
- Buffer.prototype.writeInt8(value, offset, byteLength)

  將 Int8 value (長度為 byteLength) 寫入 Buffer 的偏移值 offset。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeIntBE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用大端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeInt16BE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用大端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeInt32BE(value, offset, byteLength)

將 value 寫入 Buffer 的偏移值 offset，使用大端序。

- value：輸入整數。
- offset：輸入整數
- byteLength：輸入從 1 到 6 的整數。

- Buffer.prototype.writeIntLE(offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - offset：輸入整數。
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeInt16LE(offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - offset：輸入整數。
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeInt32LE(offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - offset：輸入整數。
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeUInt8(value, offset, byteLength)

  將 UInt8 value (長度為 byteLength) 寫入 Buffer 的偏移值 offset。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeUIntBE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用大端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeUInt16BE(value, offset, byteLength)

將 value 寫入 Buffer 的偏移值 offset，使用大端序。

- value：輸入整數。
- offset：輸入整數
- byteLength：輸入從 1 到 6 的整數。

- Buffer.prototype.writeUInt32BE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用大端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeUIntLE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeUInt16LE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeUInt32LE(value, offset, byteLength)

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - value：輸入整數。
  - offset：輸入整數
  - byteLength：輸入從 1 到 6 的整數。
- Buffer.prototype.writeDoubleBE(value, [offset])

  將 value 寫入 Buffer 的偏移值 offset，使用大端序。

  - value：輸入整數。
  - offset：選用。輸入整數。預設值為 0。

- Buffer.prototype.writeDoubleLE(value, [offset])

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - value：輸入整數。
  - offset：選用。輸入整數。預設值為 0。
- Buffer.prototype.writeFloatBE(value, [offset])

  將 value 寫入 Buffer 的偏移值 offset，使用大端序。

  - value：輸入整數。
  - offset：選用。輸入整數。預設值為 0。
- Buffer.prototype.writeFloatLE(value, [offset])

  將 value 寫入 Buffer 的偏移值 offset，使用小端序。

  - value：輸入整數。
  - offset：選用。輸入整數。預設值為 0。

支援以下執行個體方法：

- buffer[index]

  取得和設定 Buffer 中偏移值 index 的八位元組 (位元組)。

  - 取得從 0 到 255 的數字。或設定一個從 0 到 255 的數字。

支援以下執行個體屬性：

- buffer

  取得緩衝區的 ArrayBuffer 物件。
- byteOffset

  取得緩衝區 Arraybuffer 物件的 byteOffset。
- length

  取得緩衝區位元組計數。

> **Note**
>
> 所有緩衝區模塊方法在 JavaScript 運行時 2.0 中都是新的。

查詢字串

> **Note**
>
> CloudFront 函數事件對象會自動為您解析 URL 查詢字符串。這意味著在大多數情況下，您不需要使用此模組。

查詢字串模組 (querystring) 提供剖析和格式化 URL 查詢字串的方法。您可以使用 `require('querystring')` 加載模組。此模組提供下列方法。

`querystring.escape(`*`string`*`)`

URL 編碼給定的 `string`，傳回逸出的查詢字串。該方法由 `querystring.stringify()` 使用，並且不應直接使用。

`querystring.parse(`*`string`*`[, `*`separator`*`[, `*`equal`*`[, `*`options`*`]]])`

剖析查詢字串 (string) 並傳回物件。

`separator` 參數是用來分隔查詢字串中的鍵/值對的子字串。其在預設情況下為 &。

`equal` 參數是用來分隔查詢字串中的鍵和值的子字串。其在預設情況下為 =。

`options` 參數是具有下列鍵的物件：

`decodeURIComponent` *`function`*

解碼查詢字串中百分比編碼字元的函數。其在預設情況下為 `querystring.unescape()`。

`maxKeys` *`number`*

要剖析的鍵的最大數量。其在預設情況下為 `1000`。使用 `0` 的值移除計數鍵的限制。

根據預設，查詢字串中的百分比編碼字元會假設為使用 UTF-8 編碼。無效的 UTF-8 序列會被取代為 U+FFFD 取代字元。

例如，對於下列查詢字串：

```
'name=value&abc=xyz&abc=123'
```

querystring.parse() 的返回值是：

```
{
name: 'value',
abc: ['xyz', '123']
}
```

querystring.decode() 是 的別名。querystring.parse()

querystring.stringify(*object*[, *separator*[, *equal*[, *options*]]])

序列化 object 並傳回查詢字串。

separator 參數是用來分隔查詢字串中的鍵/值對的子字串。其在預設情況下為 &。

equal 參數是用來分隔查詢字串中的鍵和值的子字串。其在預設情況下為 =。

options 參數是具有下列鍵的物件：

encodeURIComponent *function*

用於將 URL 不安全字元轉換為查詢字串中的百分比編碼的函數。其在預設情況下為 querystring.escape()。

根據預設，在查詢字串中需要百分比編碼的字元會編碼為 UTF-8。若要使用不同的編碼，請指定 encodeURIComponent 選項。

例如，對於以下程式碼：

```
querystring.stringify({ name: 'value', abc: ['xyz', '123'], anotherName: '' });
```

返回值是：

```
'name=value&abc=xyz&abc=123&anotherName='
```

querystring.encode() 是 querystring.stringify() 的別名。

querystring.unescape(*string*)

解碼給定 string 中的 URL 百分比編碼字元，傳回未逸出的查詢字串。此方法由 querystring.parse() 使用，並且不應直接使用。

加密

密碼編譯模組 (crypto) 提供標準雜湊和雜湊型訊息身分驗證碼 (HMAC) 協助程式。您可以使用 require('crypto') 加載模組。

雜湊方法

crypto.createHash(*algorithm*)

建立並傳回雜湊物件，藉助此物件，您可以使用給定的演算法產生雜湊摘要：md5、sha1 或 sha256。

hash.update(*data*)

使用給定的 data 更新雜湊內容。

hash.digest([*encoding*])

計算使用 hash.update() 傳遞的所有資料的摘要。編碼可以是 hex、base64 或 base64url。

HMAC 方法

crypto.createHmac(*algorithm, secret key*)

建立並返回使用給定 algorithm 和 secret key 的 HMAC 物件。演算法可以是 md5、sha1 或 sha256。

hmac.update(*data*)

使用給定的 data 更新 HMAC 內容。

hmac.digest([*encoding*])

計算使用 hmac.update() 傳遞的所有資料的摘要。編碼可以是 hex、base64 或 base64url。

限制功能

基於安全考量，下列 JavaScript 語言功能不受支援或受限制。

動態程式碼評估

不支援動態程式碼評估。如果嘗試此評估，eval() 和 Function 建構子都會丟出錯誤。例如，const sum = new Function('a', 'b', 'return a + b') 丟出錯誤。

計時器

不支援 setTimeout()、setImmediate() 和 clearTimeout() 函數。在函數執行期間未推遲或產生任何佈建。您的函數必須同步執行方可完成。

日期和時間戳記

基於安全考量，無法存取高解析度計時器。查詢當前時間的所有 Date 方法始終在單個函數執行的生命週期內返回相同的值。返回的時間戳記是函數開始執行的時間。因此，您無法測量函數中的經過時間。

檔案系統存取

沒有檔案系統存取權。

網路存取

不支援網路呼叫。例如，不支援 XHR、HTTP(S) 和通訊端。

鍵值存放區的協助程式方法

如果您使用「索引鍵值存放區」將CloudFront 索引鍵值包含在您建立的函數中，則適用本節。CloudFront 函數有一個模塊，它提供了三個輔助方法來讀取鍵值存儲中的值。

若要在函數程式碼中使用此模組，請確定您已將索引鍵值存放區與函數相關聯。

接下來，在函數程式碼的第一行中包含下列陳述式：

```
import cf from 'cloudfront';
const kvsId = "key value store ID";
const kvsHandle = cf.kvs(kvsId);
```

您的###### *ID* 可能如下所示：a1b2c3d4-5678-90ab-cdef-EXAMPLE1

get() 方法

使用此方法可傳回您指定之金鑰名稱的索引鍵值。

請求

```
get("key", options);
```

- key：需要擷取其值的鍵名稱
- options：有一個選擇，format。它可以確保函數正確解析資料。可能的值如下：
  - string：(預設值) 以 UTF8 編碼
  - json
  - bytes：原始二進位資料緩衝區

請求示例

```
const value = await kvsHandle.get("myFunctionKey", { format: "string"});
```

回應

響應是一promise個解析為使用請求的格式的值options。默認情況下，該值作為字符串返回。

exists() 方法

使用此方法可識別索引鍵是否存在於索引鍵值存放區中。

請求

```
exists("key");
```

請求示例

```
const exist = await kvsHandle.exists("myFunctionkey");
```

回應

回應是傳回布林值 (true或false) 的回應。promise此值指定索引鍵是否存在於索引鍵值存放區中。

錯誤處理

當您請求的密鑰不存在於關聯的鍵值存儲中時，該get()方法將返回錯誤。若要管理此使用案例，您可以在程式碼中新增try和catch區塊。

meta() 方法

使用此方法返回有關鍵值存儲的元數據。

請求

```
meta();
```

請求示例

```
const meta = await kvsHandle.meta();
```

回應

回應是一個 promise，可解析為具有以下屬性的物件：

- creationDateTime：鍵值存放區建立的日期和時間，以 ISO 8601 格式表示。
- lastUpdatedDateTime：上次從來源同步的鍵值存放區的日期和時間，以 ISO 8601 格式表示。該值不包括到邊緣的傳播時間。
- keyCount：上次從來源同步後 KVS 中的總鍵數。

回應範例

```
{keyCount:3,creationDateTime:2023-11-30T23:07:55.765Z,lastUpdatedDateTime:2023-12-15T03:57:52.4
```

CloudFront 函數的範例程式碼

使用下列範例函數來協助您開始撰寫 CloudFront Functions 的函數程式碼。所有這些範例都可在的amazon-cloudfront-functions存放庫中取得 GitHub。

範例

- 將 Cache-Control 標頭新增至回應
- 將跨來源資源分享 (CORS) 標頭新增至回應
- 將跨來源資源分享 (CORS) 標頭新增至請求
- 將安全標頭新增至回應
- 將 True-Client-IP 標頭新增至請求

- 將檢視者重新導向至新的 URL
- 將 index.html 新增至不包含檔案名稱的請求 URL
- 驗證請求中的簡單權杖
- 使用 async 和 await
- 標準化查詢字串參數
- 在函數中使用鍵值對

將 **Cache-Control** 標頭新增至回應

下列範例函數會將 Cache-Control HTTP 標頭新增至回應。標頭使用 max-age 指示詞來告訴 Web 瀏覽器快取最多兩年 (63,072,000 秒) 的回應。如需詳細資訊，請參閱 MDN Web Docs 網站上的 Cache-Control。

這是一個檢視者回應函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event) {
    const response = event.response;
    const headers = response.headers;

    // Set the cache-control header
     headers['cache-control'] = {value: 'public, max-age=63072000'};

    // Return response to viewers
    return response;
}
```

JavaScript runtime 1.0

```
function handler(event) {
    var response = event.response;
    var headers = response.headers;

    // Set the cache-control header
    headers['cache-control'] = {value: 'public, max-age=63072000'};
```

```
    // Return response to viewers
    return response;
}
```

將跨來源資源分享 (CORS) 標頭新增至回應

如果回應尚未包含此標頭，下列範例函數會將 `Access-Control-Allow-Origin` HTTP 標頭新增至回應。此標頭是跨來源資源分享 (CORS) 的一部分。標頭的值 (*) 告訴 Web 瀏覽器允許來自任何來源的程式碼存取此資源。如需詳細資訊，請參閱 MDN Web Docs 網站上的 Access-Control-Allow-Origin。

這是一個檢視者回應函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event)  {
    const request = event.request;
    const response  = event.response;

    // If Access-Control-Allow-Origin CORS header is missing, add it.
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
 dict["key"] notation.
    if (!response.headers['access-control-allow-origin'] &&
 request.headers['origin']) {
        response.headers['access-control-allow-origin'] = {value:
 request.headers['origin'].value};
        console.log("Access-Control-Allow-Origin was missing, adding it now.");
    }

    return response;
}
```

JavaScript runtime 1.0

```
function handler(event)  {
    var response  = event.response;
    var headers  = response.headers;

    // If Access-Control-Allow-Origin CORS header is missing, add it.
```

```
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
 dict["key"] notation.
    if (!headers['access-control-allow-origin']) {
        headers['access-control-allow-origin'] = {value: "*"};
        console.log("Access-Control-Allow-Origin was missing, adding it now.");
    }

    return response;
}
```

將跨來源資源分享 (CORS) 標頭新增至請求

如果請求尚未包含此標頭，下列範例函數會將 `Origin` HTTP 標頭新增至請求。此標頭是跨來源資源分享 (CORS) 的一部分。此範例會將標頭的值設定為請求的 `Host` 標頭中的值。如需詳細資訊，請參閱 MDN Web Docs 網站上的 Origin。

這是一個檢視者請求函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event) {
    const request = event.request;
    const headers = request.headers;
    const host = request.headers.host.value;

   // If origin header is missing, set it equal to the host header.
   if (!headers.origin)
       headers.origin = {value:`https://${host}`};

   return request;
}
```

JavaScript runtime 1.0

```
function handler(event) {
    var request = event.request;
    var headers = request.headers;
    var host = request.headers.host.value;

   // If origin header is missing, set it equal to the host header.
```

```
    if (!headers.origin)
        headers.origin = {value:`https://${host}`};

    return request;
}
```

將安全標頭新增至回應

下列範例函數會將多個常見的安全相關 HTTP 標頭新增至回應。如需詳細資訊，請參閱 MDN Web Docs 網站中的下列頁面：

- Strict-Transport-Security
- Content-Security-Policy
- X-Content-Type-Options
- X-Frame-Options
- X-XSS-Protection

這是一個檢視者回應函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event) {
    const response = event.response;
    const headers = response.headers;

    // Set HTTP security headers
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
 dict["key"] notation
    headers['strict-transport-security'] = { value: 'max-age=63072000;
 includeSubdomains; preload'};
    headers['content-security-policy'] = { value: "default-src 'none'; img-src
 'self'; script-src 'self'; style-src 'self'; object-src 'none'; frame-ancestors
 'none'"};
    headers['x-content-type-options'] = { value: 'nosniff'};
    headers['x-frame-options'] = {value: 'DENY'};
    headers['x-xss-protection'] = {value: '1; mode=block'};
    headers['referrer-policy'] = {value: 'same-origin'};
```

```
    // Return the response to viewers
    return response;
}
```

JavaScript runtime 1.0

```
function handler(event) {
    var response = event.response;
    var headers = response.headers;

    // Set HTTP security headers
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
 dict["key"] notation
    headers['strict-transport-security'] = { value: 'max-age=63072000;
 includeSubdomains; preload'};
    headers['content-security-policy'] = { value: "default-src 'none'; img-src
 'self'; script-src 'self'; style-src 'self'; object-src 'none'"};
    headers['x-content-type-options'] = { value: 'nosniff'};
    headers['x-frame-options'] = {value: 'DENY'};
    headers['x-xss-protection'] = {value: '1; mode=block'};

    // Return the response to viewers
    return response;
}
```

將 **True-Client-IP** 標頭新增至請求

下列範例函數會將 True-Client-IP HTTP 標頭新增至請求，並以檢視者的 IP 地址做為標頭的值。將請求發 CloudFront 送到來源時，來源可以確定發送請求的 CloudFront 主機的 IP 地址，但不能確定發送原始請求的查看器（客戶端）的 IP 地址 CloudFront。此函數會新增 True-Client-IP 標頭，以便來源看到檢視者的 IP 地址。

> **Important**
>
> 若要確定在原始要求中 CloudFront 包含此標頭，您必須將其新增至原始要求原則中允許的標頭清單。

這是一個檢視者請求函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event) {
    var request = event.request;
    var clientIP = event.viewer.ip;

    //Add the true-client-ip header to the incoming request
    request.headers['true-client-ip'] = {value: clientIP};

    return request;
}
```

JavaScript runtime 1.0

```
function handler(event) {
    var request = event.request;
    var clientIP = event.viewer.ip;

    //Add the true-client-ip header to the incoming request
    request.headers['true-client-ip'] = {value: clientIP};

    return request;
}
```

將檢視者重新導向至新的 URL

下列範例函數會產生回應，以便在請求來自特定國家/地區內時，將檢視者重新導向至特定國家/地區的 URL。此函數會依賴 `CloudFront-Viewer-Country` 標頭的值來判斷檢視者所在的國家/地區。

> **Important**
>
> 若要讓此功能運作，您必須設定為將標`CloudFront-Viewer-Country`頭新增 CloudFront 至快取原則或原始要求原則中允許的標頭，將標頭新增至連入要求。

當檢視者請求來自德國時，這個範例會將檢視者重新導向至德國特定的 URL。如果檢視者請求不是來自德國，函數會傳回未修改的原始請求。

這是一個檢視者請求函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event) {
    const request = event.request;
    const headers = request.headers;
    const host = request.headers.host.value;
    const country = Symbol.for('DE'); // Choose a country code
    const newurl = `https://${host}/de/index.html`; // Change the redirect URL to
 your choice

    if (headers['cloudfront-viewer-country']) {
        const countryCode = Symbol.for(headers['cloudfront-viewer-country'].value);
        if (countryCode === country) {
            const response = {
                statusCode: 302,
                statusDescription: 'Found',
                headers:
                    { "location": { "value": newurl } }
                }

            return response;
        }
    }
    return request;
}
```

JavaScript runtime 1.0

```
function handler(event) {
    var request = event.request;
    var headers = request.headers;
    var host = request.headers.host.value;
    var country = 'DE' // Choose a country code
    var newurl = `https://${host}/de/index.html` // Change the redirect URL to your
 choice

    if (headers['cloudfront-viewer-country']) {
        var countryCode = headers['cloudfront-viewer-country'].value;
        if (countryCode === country) {
            var response = {
                statusCode: 302,
                statusDescription: 'Found',
                headers:
```

```
                    { "location": { "value": newurl } }
                }

            return response;
        }
    }
    return request;
}
```

有關重寫和重定向的更多信息，請參閱在AWS工作室中使用邊緣函數處理重寫和重定向。

將 **index.html** 新增至不包含檔案名稱的請求 URL

下列範例函數會將 index.html 附加至 URL 中不包含檔案名稱或副檔名的請求。此函數對於在 Amazon S3 儲存貯體中託管的單頁應用程式或靜態產生的網站非常有用。

這是一個檢視者請求函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
async function handler(event) {
    const request = event.request;
    const uri = request.uri;

    // Check whether the URI is missing a file name.
    if (uri.endsWith('/')) {
        request.uri += 'index.html';
    }
    // Check whether the URI is missing a file extension.
    else if (!uri.includes('.')) {
        request.uri += '/index.html';
    }

    return request;
}
```

JavaScript runtime 1.0

```
function handler(event) {
    var request = event.request;
```

```
    var uri = request.uri;

    // Check whether the URI is missing a file name.
    if (uri.endsWith('/')) {
        request.uri += 'index.html';
    }
    // Check whether the URI is missing a file extension.
    else if (!uri.includes('.')) {
        request.uri += '/index.html';
    }

    return request;
}
```

驗證請求中的簡單權杖

如下的範例函數驗證請求的查詢字串中的 JSON Web 權杖 (JWT)。如果令牌有效，該函數返回原始的，未修改的請求。 CloudFront如果權杖無效，函數會產生錯誤回應。此函數使用 `crypto` 模組。如需詳細資訊，請參閱 內建模組。

此函數假設請求在名為 `jwt` 的查詢字串參數中包含 JWT 值。

> **Warning**
>
> 若要使用此函數，您必須將私密金鑰放在函數程式碼中。

這是一個檢視者請求函數。

請參閱（詳見）的範例 GitHub。

JavaScript runtime 2.0

```
const crypto = require('crypto');

//Response when JWT is not valid.
const response401 = {
    statusCode: 401,
    statusDescription: 'Unauthorized'
};

function jwt_decode(token, key, noVerify, algorithm) {
```

```
    // check token
    if (!token) {
        throw new Error('No token supplied');
    }
    // check segments
    const segments = token.split('.');
    if (segments.length !== 3) {
        throw new Error('Not enough or too many segments');
    }

    // All segment should be base64
    const headerSeg = segments[0];
    const payloadSeg = segments[1];
    const signatureSeg = segments[2];

    // base64 decode and parse JSON
    const header = JSON.parse(_base64urlDecode(headerSeg));
    const payload = JSON.parse(_base64urlDecode(payloadSeg));

    if (!noVerify) {
        const signingMethod = 'sha256';
        const signingType = 'hmac';

        // Verify signature. `sign` will return base64 string.
        const signingInput = [headerSeg, payloadSeg].join('.');

        if (!_verify(signingInput, key, signingMethod, signingType, signatureSeg)) {
            throw new Error('Signature verification failed');
        }

        // Support for nbf and exp claims.
        // According to the RFC, they should be in seconds.
        if (payload.nbf && Date.now() < payload.nbf*1000) {
            throw new Error('Token not yet active');
        }

        if (payload.exp && Date.now() > payload.exp*1000) {
            throw new Error('Token expired');
        }
    }

    return payload;
}
```

```
//Function to ensure a constant time comparison to prevent
//timing side channels.
function _constantTimeEquals(a, b) {
    if (a.length != b.length) {
        return false;
    }

    var xor = 0;
    for (var i = 0; i < a.length; i++) {
    xor |= (a.charCodeAt(i) ^ b.charCodeAt(i));
    }

    return 0 === xor;
}

function _verify(input, key, method, type, signature) {
    if(type === "hmac") {
        return _constantTimeEquals(signature, _sign(input, key, method));
    }
    else {
        throw new Error('Algorithm type not recognized');
    }
}

function _sign(input, key, method) {
    return crypto.createHmac(method, key).update(input).digest('base64url');
}

function _base64urlDecode(str) {
    return Buffer.from(str, 'base64url')
}

function handler(event) {
    const request = event.request;
    //Secret key used to verify JWT token.
    //Update with your own key.
    var key = "LzdWGpAToQ1DqYuzHxE6YOqi7G3X2yvNBot9mCXfx5k";

    // If no JWT token, then generate HTTP redirect 401 response.
    if(!request.querystring.jwt) {
        console.log("Error: No JWT in the querystring");
        return response401;
    }
```

```
    const jwtToken = request.querystring.jwt.value;

    try{
        jwt_decode(jwtToken, key);
    }
    catch(e) {
        console.log(e);
        return response401;
    }

    //Remove the JWT from the query string if valid and return.
    delete request.querystring.jwt;
    console.log("Valid JWT token");
    return request;
}
```

JavaScript runtime 1.0

```
var crypto = require('crypto');

//Response when JWT is not valid.
var response401 = {
    statusCode: 401,
    statusDescription: 'Unauthorized'
};

function jwt_decode(token, key, noVerify, algorithm) {
    // check token
    if (!token) {
        throw new Error('No token supplied');
    }
    // check segments
    var segments = token.split('.');
    if (segments.length !== 3) {
        throw new Error('Not enough or too many segments');
    }

    // All segment should be base64
    var headerSeg = segments[0];
    var payloadSeg = segments[1];
    var signatureSeg = segments[2];

    // base64 decode and parse JSON
```

```
    var header = JSON.parse(_base64urlDecode(headerSeg));
    var payload = JSON.parse(_base64urlDecode(payloadSeg));

    if (!noVerify) {
        var signingMethod = 'sha256';
        var signingType = 'hmac';

        // Verify signature. `sign` will return base64 string.
        var signingInput = [headerSeg, payloadSeg].join('.');

        if (!_verify(signingInput, key, signingMethod, signingType, signatureSeg)) {
            throw new Error('Signature verification failed');
        }

        // Support for nbf and exp claims.
        // According to the RFC, they should be in seconds.
        if (payload.nbf && Date.now() < payload.nbf*1000) {
            throw new Error('Token not yet active');
        }

        if (payload.exp && Date.now() > payload.exp*1000) {
            throw new Error('Token expired');
        }
    }

    return payload;
}

function _verify(input, key, method, type, signature) {
    if(type === "hmac") {
        return (signature === _sign(input, key, method));
    }
    else {
        throw new Error('Algorithm type not recognized');
    }
}

function _sign(input, key, method) {
    return crypto.createHmac(method, key).update(input).digest('base64url');
}

function _base64urlDecode(str) {
    return String.bytesFrom(str, 'base64url')
}
```

```
function handler(event) {
    var request = event.request;

    //Secret key used to verify JWT token.
    //Update with your own key.
    var key = "LzdWGpAToQ1DqYuzHxE6YOqi7G3X2yvNBot9mCXfx5k";

    // If no JWT token, then generate HTTP redirect 401 response.
    if(!request.querystring.jwt) {
        console.log("Error: No JWT in the querystring");
        return response401;
    }

    var jwtToken = request.querystring.jwt.value;

    try{
        jwt_decode(jwtToken, key);
    }
    catch(e) {
        console.log(e);
        return response401;
    }

    //Remove the JWT from the query string if valid and return.
    delete request.querystring.jwt;
    console.log("Valid JWT token");
    return request;
}
```

使用 async 和 await

Amazon CloudFront JavaScript 執行階段函數 2.0 提供處理`Promise`物件的`await`語法`async`和語法。Promises 代表可以透過函數中標記為 `async` 的關鍵字 `await` 存取延遲結果。各種新 WebCrypto 功能使用承諾。

如需 `Promise` 物件的詳細資訊，請參閱 Promise。

```
async function answer() {
    return 42;
}
```

```
// Note: async, await can be used only inside an async function.

async function handler(event) {
    // var answer_value = answer(); // returns Promise, not a 42 value
    let answer_value = await answer(); // resolves Promise, 42
    console.log("Answer"+answer_value);
    event.request.headers['answer'] = { value : ""+answer_value };
    return event.request;
}
```

下面的示例 JavaScript 代碼演示了如何查看與then鏈方法承諾。您可以使用 catch 來檢視錯誤。

```
async function answer() {
    return 42;
}

async function squared_answer() {
   // before, in NJS 0.4.3 we have to write as following
   // return answer().then(function(value) { return value * value; })

   // in NJS 0.7.11 we can simplify
   return answer().then(value => value * value)
}
// note async, await can be used only inside async function
async function handler(event) {
    // var answer_value = answer(); // returns Promise, not a 42 value
    let answer_value = await squared_answer(); // resolves Promise, 42
    console.log("Answer"+answer_value);
    event.request.headers['answer'] = { value : ""+answer_value };
    return event.request;
}
```

> **Note**
>
> async並且僅await在您使用 JavaScript 執行階段 2.0 時可用。

標準化查詢字串參數

您可以標準化查詢字串參數，以提升快取命中率。

下列範例會示範如何在將要求 CloudFront 轉寄至來源之前，先依字母順序排列查詢字串，藉此改善快取命中率。

```
function handler(event) {
     var qs=[];
     for (var key in event.request.querystring) {
         if (event.request.querystring[key].multiValue) {
             event.request.querystring[key].multiValue.forEach((mv) => {qs.push(key +
 "=" + mv.value)});
         } else {
             qs.push(key + "=" + event.request.querystring[key].value);
         }
     };

     event.request.querystring = qs.sort().join('&');


     return event.request;
}
```

在函數中使用鍵值對

您可以在函數中使用來自鍵值存放區的鍵值對。

下列範例會顯示使用 HTTP 要求中 URL 內容在索引鍵值存放區中查詢自訂路徑的函數。 CloudFront 然後使用該自定義路徑發出請求。此函數有助於管理屬於網站一部分的多個路徑。

```
import cf from 'cloudfront';

// Declare the ID of the key value store that you have associated with this function
// The import fails at runtime if the specified key value store is not associated with
 the function

const kvsId = "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111";

const kvsHandle = cf.kvs(kvsId);

async function handler(event) {
    const request = event.request;
    // Use the first segment of the pathname as key
    // For example http(s)://domain/<key>/something/else
    const pathSegments = request.uri.split('/')
    const key = pathSegments[1]
```

```
    try {
        // Replace the first path of the pathname with the value of the key
        // For example http(s)://domain/<value>/something/else
        pathSegments[1] = await kvsHandle.get(key);
        const newUri = pathSegments.join('/');
        console.log(`${request.uri} -> ${newUri}`)
        request.uri = newUri;
    } catch (err) {
        // No change to the pathname if the key is not found
        console.log(`${request.uri} | ${err}`);
    }
    return request;
}
```

管理函數中的 CloudFront 函數

使用 F CloudFront unctions，您可以在中編寫輕量級函數，以進 JavaScript 行高規模，延遲敏感的 CDN 自定義。撰寫函數程式碼之後，請參閱下列主題以在 CloudFront Functions 中建立函數。接下來，您可以測試、更新、發佈，然後將函數與發 CloudFront 行版產生關聯。

主題

- 建立函數
- 測試函數
- 更新函數
- 發佈函數
- 將函數與分佈相關聯

建立函數

您可以分兩個階段建立函數。首先 JavaScript，您將函數代碼創建為外部 CloudFront。然後，您可 CloudFront 以使用創建函數並包含代碼。程式碼存在於函數內部 (而不是引用形式)。

新函數即會新增至 DEVELOPMENT 階段。您必須發佈函數，才能將其複製到LIVE舞台 (在主控台中發佈)。

Console

若要建立 函數 (主控台)

1. 登入 AWS Management Console 並在主 CloudFront 控台中開啟「功能」頁面，位於https://console.aws.amazon.com/cloudfront/v4/home#/functions。
2. 選擇建立函數。
3. 輸入 AWS 帳戶中唯一的函數名稱，然後選擇 Java 指令碼版本，然後選擇 [繼續]。該函數現在已存在。新函數的詳細資訊頁面會隨即顯示。

   > **Note**
   >
   > 如果您想要在函數中使用鍵值對，您必須選擇 Java Script 2.0。

4. 在函數程式碼區段中，選取建置索引標籤，然後輸入您的函數程式碼。建置索引標籤中包含的範例程式碼會說明函數程式碼的基本語法。您可完成如下所示的程式碼：

   - 使用預設函數，以便您可以開始使用。
   - 使用從範例程式碼複製的程式碼取代它 GitHub。
   - 用您自己的程式碼取代它。

   如需撰寫函數程式碼的詳細資訊，請參閱下列內容：

   - 撰寫函數程式碼
   - the section called “事件結構”
5. 隨時根據需要選擇儲存變更，以儲存函數程式碼。
6. 如果函數程式碼使用鍵值對，則必須關聯鍵值存放區。

   您可以在函數初始建立期間關聯鍵值存放區。或者，您可以稍後透過更新函數來關聯它。

   若要立即關聯鍵值存放區，請依照下列步驟執行：

   - 移至「關聯」 KeyValueStore 區段，然後選擇「關聯現有」 KeyValueStore。
   - 選取函數中包含索引鍵值對的索引鍵值儲存庫，然後選擇「關聯」 KeyValueStore。

   CloudFront 立即將商店與該功能相關聯。您無需儲存函數。

CLI

如果您使用 CLI，通常會先在檔案中建立函數程式碼，然後使用 AWS CLI來建立函數。

1. 在檔案中建立函數程式碼，並將其儲存於電腦可以連線的目錄中。如需撰寫函數程式碼的詳細資訊，請參閱下列內容：
   - 撰寫函數程式碼
   - the section called “事件結構”
2. 執行命令，如範例所示。此範例會使用 fileb:// 標記法來傳入檔案。它還會包括換行符號，讓命令更易於讀取。

```
aws cloudfront create-function \
    --name MaxAge \
    --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":
[{"KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}]}}' \
    --function-code fileb://function-max-age-v1.js
```

備註：

- Runtime：Java Script 版本。如果您想要在函數中使用鍵值對，您必須指定版本 2.0。
- KeyValueStoreAssociations：如果您的函數使用鍵值對，則可以在函數初始建立期間與鍵值存放區相關聯。或者，您可以稍後使用 update-function 來關聯它。Quantity 永遠等於 1，因為每個函數只能有一個與其關聯的鍵值存放區。

如果命令成功執行，您會看到如下所示的輸出。

```
ETag: ETVABCEXAMPLE
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years
    Runtime: cloudfront-js-2.0
    KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
  FunctionMetadata:
    CreatedTime: '2021-04-18T20:38:56.915000+00:00'
```

```
    FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge
    LastModifiedTime: '2023-11-19T20:38:56.915000+00:00'
    Stage: DEVELOPMENT
  Name: MaxAge
  Status: UNPUBLISHED
Location: https://cloudfront.amazonaws.com/2020-05-31/function/
arn:aws:cloudfront:::function/MaxAge
```

大多數資訊都是從請求中複製的。其他資訊由新增 CloudFront。

> **備註**
>
> - ETag：每次修改鍵值存放區時，這個值都會變更。您可以使用此值和函數名稱來引用 future 的函數。確保始終使用目前的 ETag。
> - FunctionARN
> - Stage
> - 111122223333
> - Status

測試函數

在將CloudFront 函數部署到即時階段 (實際執行) 之前，您可以測試 Function，以確保其運作如預期般運作。若要測試函數，您需要提供一個事件物件，代表您的 CloudFront 發行版可以在生產環境中接收的 HTTP 要求或回應。 CloudFront 函數會執行以下作業：

1. 執行該函數，使用提供的事件物件作為函數的輸入。
2. 返回函數的結果 (修改後的事件物件)，同時返回任何函數日誌或錯誤訊息以及函數的運算利用率。如需運算使用率的詳細資訊，請參閱 the section called “瞭解運算利用率”。

主題

- 設定事件物件
- 測試函數
- 瞭解運算利用率

設定事件物件

在測試函數之前，您必須建立事件物件以進行測試。有幾種選項。

選項 1：設定事件物件而不儲存

您可以在 CloudFront控制台的可視化編輯器中設置事件對象，而不保存它。

您可以使用此事件物件從 CloudFront主控台測試函數，即使它沒有儲存。

選項 2：在視覺化編輯器中建立事件物件

您可以在 CloudFront控制台的可視化編輯器中設置事件對象，而不保存它。您可以針對每個函數建立 10 個事件物件，例如，可測試不同的可能輸入。

當您以這種方式建立事件物件時，您可以使用事件物件來測試主 CloudFront 控台中的函數。您不能使用它來測試使用 AWS API 或 SDK 的功能。

選項 3：使用文字編輯器建立事件物件

您可以使用文字編輯器，以 JSON 格式建立事件物件。如需有關事件物件結構的詳細資訊，請參閱事件結構。

您可以使用此事件物件來測試使用 CLI 的函數。但是您不能使用它來測試 CloudFront 控制台中的功能。

使用選項 1 或 2 建立

1. 在 CloudFront 主控台中顯示 [函數] 頁面，然後選擇您要測試的函數。
2. 在函數詳細資訊頁面上，選擇測試索引標籤。測試函數區段會隨即顯示，其中包含編輯 JSON 和測試函數按鈕。
3. 完整事件類型：
   - 如果函數會根據請求修改 HTTP 請求或產生回應，請選擇檢視者請求。已顯示的請求區段適用於此類型。
   - 或選擇檢視者回應。已顯示的請求區段適用於此類型。此外，回應區段會顯示。
4. 完成您想要包含在事件中的所有欄位。在進行時，您可選擇編輯 JSON 來檢視原始 JSON。
5. 儲存事件 (如果您想要的話)。

您也可以選擇 「編輯 JSON」 並複製原始 JSON，並將其儲存在您自己的檔案中，在之外 CloudFront。

使用選項 3 建立

使用文字編輯器建立事件物件。將檔案儲存於電腦可以連線的目錄中。

請確定您遵循這些準則：

- 省略 `distributionDomainName`、`distributionId` 和 `requestId` 欄位。
- 確認標頭、Cookie 和查詢字串的名稱為小寫。

以這種方式建立事件物件的一個選項是使用視覺化編輯器建立範例。您可以確定範例格式正確。然後您可以複製原始 JSON 並將其貼到文字編輯器中並儲存檔案。

如需有關事件結構的詳細資訊，請參閱 事件結構。

測試函數

您可以在 CloudFront 主控台或使用 AWS CLI.

Console

在 CloudFront 主控台中，您可以測試使用主控台建立的函數。

若要測試函數

1. 在 CloudFront主控台中顯示 [函數] 頁面，然後選擇您要測試的函數。
2. 在函數頁面上，選擇測試索引標籤。測試函數區段會隨即顯示，其中包含編輯 JSON 和測試函數按鈕。
3. 確定已顯示正確的事件。

   如果要從目前顯示的事件切換，請在選取測試事件欄位中選擇另一個事件。
4. 選擇測試函數按鈕。控制台顯示函數的輸出，包括函數日誌。它也會顯示運算使用率。如需詳細資訊，請參閱 the section called “瞭解運算利用率”。

CLI

您可以使用 aws cloudfront test-function 命令來測試函數。

1. 執行命令，如範例所示。從包含此檔案的相同目錄執行命令。

   此範例會使用 fileb:// 標記法來傳入事件物件檔案。它還會包括換行符號，讓命令更易於讀取。

```
aws cloudfront test-function \
    --name MaxAge \
    --if-match ETVABCEXAMPLE \
    --event-object fileb://event-maxage-test01.json \
    --stage DEVELOPMENT
```

備註：

- 您可以透過其名稱和 ETag (在 `if-match` 參數中) 引用該函數。您可以依照事件物件在檔案系統中的位置來參照事件物件。
- 此階段可以是 DEVELOPMENT 或 LIVE。

如果命令成功執行，您會看到如下所示的輸出。

```
TestResult:
  ComputeUtilization: '21'
  FunctionErrorMessage: ''
  FunctionExecutionLogs: []
  FunctionOutput: '{"response":{"headers":{"cloudfront-functions":
{"value":"generated-by-CloudFront-Functions"},"location":{"value":"https://
aws.amazon.com/cloudfront/"}},"statusDescription":"Found","cookies":
{},"statusCode":302}}'
  FunctionSummary:
    FunctionConfig:
      Comment: MaxAge function
      Runtime: cloudfront-js-2.0
      KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
    FunctionMetadata:
      CreatedTime: '2021-04-18T20:38:56.915000+00:00'
      FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge
      LastModifiedTime: '2023-17-20T10:38:57.057000+00:00'
      Stage: DEVELOPMENT
    Name: MaxAge
```

```
    Status: UNPUBLISHED
```

備註

- FunctionExecutionLogs 包含函數在 console.log() 語句中撰寫的日誌行清單 (如果有的話)。
- ComputeUtilization。請參閱 the section called “瞭解運算利用率”。
- FunctionOutput 包含該函數返回的事件物件。

瞭解運算利用率

運用利用率是指執行函數所花費的時間，以所允許時間上限的百分比表示。例如，35 的值表示該函數以所允許時間上限的 35% 完成。

如果函數連續超過允許的最大時間，請 CloudFront 節流函數。下列清單說明，根據運算利用率的值，函數限流的可能性。

運算利用率值：

- 1 — 50— 函數遠低於允許時間上限，應不會受到限流。
- 51 — 70— 函數接近允許時間上限。考慮將函數程式碼最佳化。
- 71 — 100 — 該功能非常接近或超過最大允許時間。 CloudFront 如果將此函數與分佈相關聯，則可能會限制此函數。

更新函數

您隨時都可以更新。這些變更只會對 DEVELOPMENT 階段中的函數版本進行。您必須發佈函數，才能將變更從 DEVELOPMENT 階段複製到 LIVE。

您可以在 CloudFront 主控台或使用 AWS CLI.

Console

更新函數程式碼 (主控台)

1. 在 CloudFront 主控台中開啟 [函數] 頁面https://console.aws.amazon.com/cloudfront/v4/home#/functions，然後選擇您要更新的函數。

2. 進行變更：

   - 您可以選擇編輯按鈕，然後變更詳細資訊區段中的欄位。
   - 您可以變更或移除關聯的鍵值存放區。選擇適當的按鈕。如需鍵值存放區的詳細資訊，請參閱 the section called “使用 CloudFront KeyValueStore”。
   - 您無法變更函數程式碼。選擇建置索引標籤，進行變更，然後選擇儲存變更，僅儲存對程式碼的變更。

CLI

更新函數程式碼 (CLI)

執行命令，如範例所示。

此範例會使用 fileb:// 標記法來傳入檔案。它還會包括換行符號，讓命令更易於讀取。

```
aws cloudfront update-function \
    --name MaxAge \
    --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":
[{"KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}]}}' \
    --function-code fileb://function-max-age-v1.js \
    --if-match ETVABCEXAMPLE
```

備註：

- 您可以透過名稱和 ETag (在 `if-match` 參數中) 來識別函數。請確定您使用目前的 ETag。您可以使用描述操作獲取它。
- 即使您不想要變更，您也必須包含 `function-code`。
- 要小心 `function-config`。您應該傳遞您想要在組態中保留的所有內容。具體而言，請依下列方式處理鍵值存放區：
  - 如果要保留現有的鍵值存放區關聯 (如果有的話)，請指定現有存放區的名稱。
  - 如果要變更關聯，請指定新鍵值存放區的名稱。
  - 如果您想要移除關聯，請省略 `KeyValueStoreAssociations` 參數。

如果命令成功執行，您會看到如下所示的輸出。

```
ETag: ETVXYZEXAMPLE
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years \
    Runtime: cloudfront-js-2.0 \
    KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
  FunctionMetadata: \
    CreatedTime: '2021-04-18T20:38:56.915000+00:00' \
    FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge \
    LastModifiedTime: '2023-12-19T23:41:15.389000+00:00' \
    Stage: DEVELOPMENT \
  Name: MaxAge \
  Status: UNPUBLISHED
```

大多數資訊都是從請求中複製的。其他資訊由新增 CloudFront。

備註

- ETag：每次修改鍵值存放區時，這個值都會變更。
- FunctionARN
- Stage
- Status

發佈函數

發佈函數會將其從 DEVELOPMENT 階段複製到 LIVE。

Important

發佈函數時，與該函數相關聯的所有快取行為會在分佈完成部署後立即開始使用新發佈的副本。

如果沒有與函數相關聯的快取行為，發佈該函數可讓您將其與快取行為產生關聯。您只能將快取行為與 LIVE 階段中的函數產生關聯。

您可以在 CloudFront 主控台或使用 AWS CLI.

在發佈之前，您必須測試函數。

Console

要發布您的功能，您可以使用 CloudFront 控制台。控制台還顯示與該函數相關聯的 CloudFront 發行版。

若要發佈函數 (主控台)

1. 若要發佈函數，請在 CloudFront 主控台中開啟 Functions 頁面https://console.aws.amazon.com/cloudfront/v4/home#/functions，然後選擇您要發佈的函數。
2. 在函數頁面上，選擇發佈索引標籤。然後選擇發佈按鈕 (或者，如果您的函數已附加至一或多個快取行為，請選擇發佈並更新按鈕)。
3. (選擇性) 若要查看與函數相關聯的分佈，請選擇 CloudFront「關聯的分佈」 以展開該區段。

成功後，您會在頁面頂端看到一個橫幅，指出已成功發佈**##**名稱。您也可以選擇建置索引標籤，然後選擇即時以檢視函數程式碼的即時版本。

CLI

若要發佈函數，請執行 aws cloudfront publish-function 命令，如範例所示。在此範例中，提供分行符號以使範例更具可讀性。

```
aws cloudfront publish-function \
    --name MaxAge \
    --if-match ETVXYZEXAMPLE
```

如果命令成功執行，您會看到如下所示的輸出。

```
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years
    Runtime: cloudfront-js-2.0
  FunctionMetadata:
    CreatedTime: '2021-04-18T21:24:21.314000+00:00'
```

```
  FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
  LastModifiedTime: '2023-12-19T23:41:15.389000+00:00'
  Stage: LIVE
Name: MaxAge
Status: UNASSOCIATED
```

將函數與分佈相關聯

若要使用 CloudFront Functions 中的函數搭 CloudFront 配發行版，您可以將函數與散佈中的一或多個快取行為建立關聯。您可以在多個分佈中將函數與的多個快取行為相關聯。在關聯函數之前，您必須將其發佈至 LIVE 階段。

將函數與快取行為建立關聯時，您必須選擇事件類型。事件類型決定 CloudFront 函數何時執行函數。有兩種事件類型可供選擇：

如需有關事件類型的詳細資訊，請參閱 CloudFront 可以觸發 @Edge 函數的事件。您不能將面向起點的事件類型（原始請求和來源響應）與 CloudFront Functions 一起使用。

- 檢視器要求 — 當 CloudFront 收到來自檢視器的要求時，函數會執行。
- 「查看器響應」— 該函數在 CloudFront 返回給查看器響應之前運行。

您可以將函數與 CloudFront 主控台中的發行版產生關聯，也可以將函數與 AWS CLI.

Console

您可以使用主 CloudFront 控台將函數與現有 CloudFront 發行版中的現有快取行為建立關聯。如需有關建立分佈的詳細資訊，請參閱 the section called “建立分發”。

將函數與現有的快取行為建立關聯 (主控台)

1. 在 CloudFront 主控台中開啟「函數」頁面https://console.aws.amazon.com/cloudfront/v4/home#/functions，然後選擇要關聯的函數名稱。
2. 在函數頁面上，選擇發佈索引標籤。
3. 選擇「發佈功能」。
4. 選擇 Add association (建立關聯)。在出現的對話方塊中，選取分佈、事件類型及/或快取行為。

   在事件類型中，選擇您希望此函數執行的時間：

- 若要在每次 CloudFront 收到請求時執行函數，請選擇「檢視器請求」。
- 要在每次 CloudFront 返回響應時運行該函數，請選擇「查看器響應」。

若要儲存組態，請選擇新增關聯。

CloudFront 將分配與函數相關聯。等待幾分鐘，讓關聯的分佈完成部署。您可以在函數詳細資訊頁面上選擇檢視分佈來檢查進度。

CLI

您可以將函數與下列任何行為建立關聯：

- 現有的快取行為。
- 現有分佈中新的新快取行為。
- 新分佈中的新快取行為。

下列程序顯示如何將函數與現有的快取行為建立關聯。

將函數與現有快取行為 (AWS CLI) 產生關聯

1. 使用下列命令儲存分佈的組態，該分佈的快取行為將與函數產生關聯。此命令會將分佈組態儲存到名為 `dist-config.yaml` 的檔案中。若要使用此命令，請執行下列動作：
   - 將 *`DistributionID`* 取代為分佈的 ID。
   - 在一行上執行命令。在此範例中，提供分行符號以使範例更具可讀性。

   ```
   aws cloudfront get-distribution-config \
       --id DistributionID \
       --output yaml > dist-config.yaml
   ```

   當命令成功時，不會 AWS CLI 返回任何輸出。
2. 開啟您剛才建立且命名為 `dist-config.yaml` 的檔案。編輯檔案以進行下列變更。
   a. 將 `ETag` 欄位重新命名為 `IfMatch`，但不要變更欄位的值。
   b. 在快取行為中，尋找名為 `FunctionAssociations` 的物件。更新此物件以新增函數關聯。如下的範例給出函數關聯的 YAML 語法。

- 下列範例顯示檢視者請求事件物件 (觸發條件)。若要使用檢視者回應事件類型，請將 `viewer-request` 取代為 `viewer-response`。
- 將 *`arn:aws:cloudfront::111122223333:function/ExampleFunction`* 取代為與此快取行為相關聯之函數的 Amazon Resource Name (ARN)。要獲取函數 ARN，您可以使用 aws cloudfront list-functions 命令。

```
FunctionAssociations:
  Items:
    - EventType: viewer-request
      FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
  Quantity: 1
```

進行這些變更後，請儲存檔案。

3. 使用以下命令更新分佈，同時新增函數關聯。若要使用此命令，請執行下列動作：

- 將 *`DistributionID`* 取代為分佈的 ID。
- 在一行上執行命令。在此範例中，提供分行符號以使範例更具可讀性。

```
aws cloudfront update-distribution \
    --id DistributionID \
    --cli-input-yaml file://dist-config.yaml
```

如果命令成功執行，您會看到如下所示的輸出，其中描述剛使用函數關聯更新的分佈。為便於閱讀，對如下的範例輸出進行了截斷。

```
Distribution:
  ARN: arn:aws:cloudfront::111122223333:distribution/EBEDLT3BGRBBW
  ... truncated ...
  DistributionConfig:
    ... truncated ...
    DefaultCacheBehavior:
      ... truncated ...
      FunctionAssociations:
        Items:
        - EventType: viewer-request
          FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
```

```
        Quantity: 1
      ... truncated ...
  DomainName: d111111abcdef8.cloudfront.net
  Id: EDFDVBD6EXAMPLE
  LastModifiedTime: '2021-04-19T22:39:09.158000+00:00'
  Status: InProgress
ETag: E2VJGGQEG1JT8S
```

關聯分佈的影響

重新部署分佈時，此分佈的 `Status` 變更為 `InProgress`。一旦新的發佈組態到達 CloudFront邊緣位置，該邊緣位置就會開始使用關聯的函數。完全部署發行版後，會`Status`變更回`Deployed`，表示關聯的 CloudFront 函數會在全球的所有 CloudFront 邊緣位置上線。通常這需要幾分鐘的時間。

Amazon CloudFront KeyValueStore

CloudFront KeyValueStore 是安全、全球、低延遲的金鑰值資料存放區，可從 CloudFront Functions 內部進行讀取存取，從而在 CloudFront 邊緣位置啟用進階可自訂邏輯。

使用時 CloudFront KeyValueStore，您可以更新函數程式碼，並對與函數相關聯的資料進行獨立更新。這種分離簡化了函數程式碼，並且可以輕鬆更新資料，而無需部署程式碼變更。

> **Note**
>
> 若要使用 CloudFront KeyValueStore，您的 CloudFront 函數必須使用JavaScript 執行階段 2.0。

使用鍵值對的一般程序如下：

- 創建鍵值存儲，並用一組鍵值對填充它。您可以將金鑰價值存放區新增至 Amazon S3 儲存貯體，或手動輸入。
- 將鍵值存儲與您的 CloudFront 函數相關聯。
- 在函數程式碼中，使用鍵的名稱來擷取與鍵關聯的值或評估鍵是否存在。如需有關在函數程式碼中使用鍵值配對的詳細資訊，以及有關 Helper 方法的詳細資訊，請參閱the section called “鍵值存放區的協助程式方法”。

如需開始使用的詳細資訊 CloudFront KeyValueStore，請參閱 Amazon CloudFront KeyValueStore AWS 部落格文章簡介。

您可以使用 CloudFront 主控台、 CloudFront API 或支援的 AWS SDK。若要開始使用 CloudFront KeyValueStore，請參閱下列主題。

主題

使用案例

鍵值對的典型使用案例如下：

- URL 重寫或重新導向。鍵值對可以保存重寫的 URL 或重定向 URL。
- A/B 測試和功能旗標。指派一定百分比的流量至特定版本的網站，即可建立執行實驗的函數。
- 存取授權。您可以實施訪問控制，以根據您定義的條件和存儲在密鑰值存儲中的數據來允許或拒絕請求。

支援的值格式

鍵值對中的值可以以下列任何一種格式儲存：

- 字串
- 位元組編碼字串
- JSON

安全

CloudFront 函數及其所有關鍵值存儲的數據都被安全地處理，如下所示：

- CloudFront 當您呼叫 CloudFront KeyValueStoreAPI 作業時，會加密靜態和傳輸期間 (讀取或寫入金鑰值存放區時) 的每個金鑰值。
- 執行函數時，會 CloudFront 解密 CloudFront 邊緣位置記憶體中的每個鍵值對。

使用關鍵值存儲

您必須建立索引鍵值存放區，以保存要在 CloudFront Functions 中使用的索引鍵值配對。

建立索引鍵值存放區並加入索引鍵值配對之後，您可以在 CloudFront 函數程式碼中使用索引鍵值。JavaScript 執行階段 2.0 包含一些協助程式方法，可用來處理函數程式碼中的索引鍵值。如需詳細資訊，請參閱 the section called “鍵值存放區的協助程式方法”。

主題

建立金鑰值存放區

您可以創建一個空鍵值存儲，然後在以後添加鍵值對。或者，您可以同時創建一個鍵值存儲及其鍵值對。

> **Note**
>
> 如果您從 Amazon S3 儲存貯體指定資料來源，則必須擁有該儲存貯體的`s3:GetObject`和`s3:GetBucketLocation`許可。如果您沒有這些權限，則 CloudFront 無法成功建立金鑰值存放區。

Console

若要建立金鑰值存放區 (主控台)

1. 決定是否要在建立索引鍵值存放區的同時新增索引鍵值配對。 CloudFront 主控台以及 CloudFront API 和 AWS SDK 都支援此匯入功能。但是，僅當您最初創建鍵值存儲時才支持它。

   如果您想要使用檔案，請立即建立檔案。

2. 登入 AWS Management Console 並在主 CloudFront 控台中開啟「功能」頁面，位於https://console.aws.amazon.com/cloudfront/v4/home#/functions。
3. 選擇 (KeyValueStores) 索引標籤。選擇 [建立] KeyValueStore。
4. 輸入鍵值存放區的名稱和可選描述。
5. 完成 S3 URI：
   - 如果您準備了鍵值對的檔案，請輸入存放該檔案的 Amazon S3 儲存貯體的路徑。
   - 如果您打算手動輸入鍵值對，請將此欄位保留空白。
6. 選擇建立。鍵值存儲現在存在。

   新索引鍵值存放區的詳細資訊頁面隨即顯示。頁面上的資訊包括鍵值存放區的 ID 和 ARN。
   - ID 是您 AWS 帳戶中唯一的隨機字元字串。
   - ARN 具有以下語法：

     AWS ##:key-value-store/#### ID
7. 請查看鍵值對區段。如果您匯入了檔案，此區段會顯示一些鍵值對。否則為空白。您可以執行下列作業：
   - 如果您沒有從 Amazon S3 儲存貯體匯入檔案，而且您現在想要新增機碼值配對，可以完成本節。
   - 如果您確實匯入了檔案，也可以手動新增更多值。
   - 您可以將此區段保留空白，稍後再透過編輯索引鍵值存放區來加入這些配對。

   立即新增鍵值對：
   - 選擇新增鍵值配對按鈕。
   - 選擇新增配對，然後輸入名稱和值。
   - 再次選擇新增配對按鈕，以新增更多鍵值對。

     完成後，選擇儲存變更以儲存在鍵值存放區中的所有鍵值對。在出現的確認對話方塊中，選擇完成。
8. 如果您要立即將索引鍵值存放區與函數產生關聯，請完成「關聯函數」區段。您也可以稍後從此索引鍵值儲存詳細資訊頁面或從函數詳細資訊頁面建立此關聯。

   若要立即建立關聯，請選擇移至函數按鈕。如需詳細資訊，請參閱 ??? 或 ??? 。

Programmatically

若要建立索引鍵值存放區

1. 決定是否要在建立索引鍵值存放區的同時新增索引鍵值配對。（您也可以稍後添加鍵值對。）CloudFront 主控台以及 CloudFront API 和 SDK 都支援此匯入功能。但只有當您最初創建鍵值存儲時才支持它。

   如果您想要使用檔案，請立即建立檔案。

2. 使用 CloudFront API 的創建操作或您首選的 AWS SDK。例如，對於其餘 API，請使用 CloudFront. CreateKeyValueStore。該操作需要幾個參數：
   - 名稱。
   - 包含註解的 `configuration` 參數。
   - 一種`import-source`參數，可讓您從存放在 Amazon S3 儲存貯體中的檔案匯入金鑰-值配對。請注意，您只能在初始建立索引鍵值存放區時從檔案匯入。如需檔案格式的資訊，請參閱 the section called “創建鍵值對的文件”。

操作回應包含下列資訊：

- 請求中傳遞的值，包括您指派的名稱。
- 建立時間等資料。
- 一個 ETag（例如，ETVABCEXAMPLE2），ARN，其中包含鍵值存儲的名稱（例如，`arn:aws:cloudfront::111122223333:key-value-store/MaxAge`）。

  您將使用 ETag，ARN 和名稱的某些組合來處理密鑰值存儲以編程方式。

關鍵值存儲狀態

當您建立索引鍵值存放區時，資料倉庫可以具有下列狀態值。

Value	描述
佈建	已建立索引鍵值存放區，並 CloudFront 正在處理您指定的資料來源。
備妥	已建立索引鍵值存放區，並 CloudFront 成功處理您指定的資料來源。

Value	描述
匯入失敗	CloudFront 無法處理您指定的資料來源。如果您的檔案格式無效或超過大小限制，就會顯示此狀態。如需詳細資訊，請參閱 創建鍵值對的文件。

將鍵值存儲與函數相關聯

您可以透過在函數中工作，使鍵值存放區與函數產生關聯。您必須建立此關聯，才能在該函數中使用該存儲中的鍵值對。適用的規定如下：

- 一個函數可以有一個鍵值存放區。
- 一個鍵值存放區可以與多個函數相關聯。

您可以利用下列方式來使用關聯：

- 您可以建立函數和鍵值存放區之間的關聯：
  - 在 CloudFront 主控台上，檢視索引鍵值存放區詳細資訊頁面，然後選擇移至函數按鈕。適當的頁面會隨即顯示 - 函數清單 (如果目前沒有關聯的函數) 或函數詳細資訊頁面 (如果目前有關聯的話)。如需詳細資訊，請參閱 the section called “將鍵值存儲與函數相關聯”。
  - 以程式設計方式，使用慣用 CloudFront API 或 SDK 的函數更新作業。

  建立關聯之後 (或者如果您變更關聯)，您應該測試函數，並且必須重新發佈函數。

- 如果您修改索引鍵值存放區而不變更索引鍵值配對，則不需要更新關聯 (這表示您不需要再次發佈)。但是您應該測試該函數。
- 如果您變更索引鍵值存放區中的索引鍵值配對，則不需要更新關聯 (這表示您不需要再次發佈)。但是您應該測試該函數以驗證它是否適用於對鍵值對的更改。
- 您可以查看使用特定鍵值存儲的所有功能。在 CloudFront 主控台上，查看索引鍵值存放區詳細資料頁面。

修改鍵值存儲

您可以使用鍵值對，並且可以更改鍵值存儲和函數之間的關聯。

Console

修改鍵值存放區的步驟

1. 登入 AWS Management Console 並在主 CloudFront 控台中開啟「功能」頁面，位於https://console.aws.amazon.com/cloudfront/v4/home#/functions。
2. 選擇 (KeyValueStores) 索引標籤。選擇您想要變更的鍵值存放區。詳細資訊頁面會隨即顯示。
   - 若要使用索引鍵值配對，請選擇 [金鑰值配對] 區段中的 [編輯] 按鈕。您可以新增更多索引鍵值對、刪除任何鍵值組，也可以變更現有索引鍵值組的值。完成之後，請選擇 Save changes (儲存變更)。
   - 若要使用此索引鍵值存放區的關聯，請選擇移至函數按鈕。適當的頁面會隨即顯示 - 函數清單 (如果目前沒有關聯的函數) 或函數詳細資訊頁面 (如果目前有關聯的話)。如需詳細資訊，請參閱 the section called “將鍵值存儲與函數相關聯”。

Programmatically

您可以使用以下方式使用索引鍵值存放區。

更改鍵值對

您可以新增多個索引鍵值配對、刪除一或多個索引鍵值配對，也可以變更現有鍵值配對的值。如需詳細資訊，請參閱 the section called “以編程方式使用鍵值對”。

更改鍵值存儲的函數關聯

若要使用此索引鍵值存放區的關聯，請參閱the section called “更新函數”。您將需要關鍵值存儲的 ARN。如需詳細資訊，請參閱 the section called “獲取對鍵值存儲的引用”。

刪除鍵值存放區

您可以使用 CloudFront 主控台或 API 刪除金鑰值存放區。

Console

刪除鍵值存放區的步驟

1. 登入 AWS Management Console 並在主 CloudFront 控台中開啟「功能」頁面，位於https://console.aws.amazon.com/cloudfront/v4/home#/functions。

2. 驗證鍵值存儲是否與函數相關聯。如果是，將移除關聯。如需這些步驟的詳細資訊，請參閱 ???
3. 選擇 (KeyValueStores) 索引標籤。選取您要變更的金鑰值存放區，然後選擇 [刪除]。

Programmatically

刪除鍵值存放區的步驟

1. 獲取 ETag 和密鑰值存儲的名稱。如需詳細資訊，請參閱 the section called “獲取對鍵值存儲的引用”。
2. 驗證鍵值存儲是否與函數相關聯。如果是，將移除關聯。如需這些步驟的詳細資訊，請參閱 ???。
3. 若要刪除金鑰值存放區，請使用慣用 CloudFront API 或 SDK 的刪除作業。例如，對於其餘 API，請使用 CloudFront. DeleteKeyValueStore。

獲取對鍵值存儲的引用

要以編程方式使用鍵值存儲，您需要 ETag 和密鑰值存儲的名稱。若要取得此資料，請使用 CloudFront API 或您偏好的 AWS SDK，並依照下列步驟執行：

1. 使用 CloudFront.ListKeyValueStoresAPI 操作返回鍵值存儲的列表。尋找您要變更的金鑰值存放區名稱。
2. 使用 CloudFront.DescribeKeyValueStoreAPI 作業並指定您從上一個步驟傳回的金鑰值存放區名稱。

該響應包括一個 UUID，鍵值存儲的 ARN 以及密鑰值存儲的 ETag。

- UUID 是 128 位元。例如：`a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`
- ARN 包括 AWS 帳戶 數字`key-value-store`、常數和 UUID。例如：

  `arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`
- ETag 會如下所示：`ETVABCEXAMPLE2`

如需有關`DescribeKeyValueStore`作業的更多資訊，請參閱the section called “關於 CloudFront KeyValueStore”。

創建鍵值對的文件

當您建立 UTF-8 編碼檔案時，請使用下列 JSON 格式：

```
{
  "data":[
    {
      "key":"key1",
      "value":"value"
    },
    {
      "key":"key2",
      "value":"value"
    }
  ]
}
```

您的檔案不能包含重複的索引鍵。如果您在 Amazon S3 儲存貯體中指定了無效的檔案，您可以更新檔案以移除任何重複項目，然後再次嘗試建立金鑰值存放區。

如需詳細資訊，請參閱 建立金鑰值存放區。

> **Note**
>
> 資料來源及其鍵值組的檔案具有以下限制：
>
> - 檔案大小 - 5 MB
> - 鍵大小 — 512 個字元
> - 鍵大小 — 1024 個字元

使用鍵值資料

您可以使用下列方式在現有索引鍵值儲存區中使用鍵值配對：

- 使用 Amazon CloudFront 控制台。
- 使用 CloudFront KeyValueStore API 或您偏好的 AWS SDK。

本節說明如何將索引鍵值配對新增至現有的索引鍵值存放區。若要在最初建立索引鍵值存放區時包含鍵值配對，請參閱the section called “建立金鑰值存放區”。

主題

使用控制台使用鍵值對 CloudFront

您可以使用 CloudFront 控制台來處理鍵值對。

使用鍵值配對的步驟

1. 登入 AWS Management Console 並在主 CloudFront 控台中開啟「功能」頁面，位於https://console.aws.amazon.com/cloudfront/v4/home#/functions。
2. 選擇 (KeyValueStores) 索引標籤。選擇您想要變更的鍵值存放區。詳細資訊頁面會隨即顯示。
3. 在 [金鑰值配對] 區段中，選擇 [編輯]。
4. 您可以新增鍵值配對、刪除鍵值配對，或變更現有索引鍵值配對的值。
5. 完成之後，請選擇 Save changes (儲存變更)。

以編程方式使用鍵值對

> **Note**
>
> 該 CloudFront KeyValueStoreAPI 具有與 CloudFront API 不同的命名空間。

主題

獲取對鍵值存放區的引用

當您使用輸入寫操作時 CloudFront KeyValueStore，您需要傳入 ARN 和鍵值存儲的 ETag。若要取得此資料，請依下列步驟執行：

1. 使用您偏好的 CloudFront API 或 SDK 的列表操作。例如，對於 REST API，請使用 CloudFront.ListKeyValueStores。回應包含鍵值存放區的清單。尋找您想要變更的鍵值存放區的名稱。
2. 使用您偏好的 CloudFront KeyValueStore API 或 SDK 的描述操作。例如，對於 REST API，請使用 CloudFrontKeyValueStore.DescribeKeyValueStore。傳入您在上一個步驟取得的名稱。

> **Note**
>
> 使用 API 中的操作，而不是 CloudFront KeyValueStore API 中的 CloudFront 操作。如需詳細資訊，請參閱 the section called “關於 CloudFront KeyValueStore”。

該響應包括 ARN 和鍵值存儲的 ETag。

- ARN 包括 AWS 帳戶 數字`key-value-store`、常數和 UUID。例如：

  ```
  arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
  ```

- ETag 會如下所示：`ETVABCEXAMPLE2`

更改鍵值存儲中的鍵值對

您可以使用首選 CloudFront KeyValueStore API 或 SDK 的以下操作來處理鍵值對。所有這些操作都在一個指定的鍵值存儲中工作：

- `CloudFrontKeyValueStore.DeleteKey`：刪除一個鍵。請參閱DeleteKey。
- `CloudFrontKeyValueStore.GetKey`：獲得一個鍵。請參閱GetKey。
- `CloudFrontKeyValueStore.ListKeys`：列出鍵。請參閱ListKeys。
- `CloudFrontKeyValueStore.PutKey`：您可以執行兩個動作：
  - 在一個鍵值存儲中創建一個新的鍵值對：在這種情況下，傳遞一個新的鍵名和值。
  - 在一個現有的鍵值對中設置不同的值：在這種情況下，傳遞現有的鍵名和一個新的鍵值。

  請參閱PutKey。
- `CloudFrontKeyValueStore.UpdateKeys`：您可以在一項作業中執行下列一或多個動 all-or-nothing 作：

- 刪除一或多個鍵值對。
- 創建一個或多個新的鍵-值對。
- 在一個或多個現有鍵值對中設定不同的值。

請參閱UpdateKeys。

關於 CloudFront KeyValueStore

若要以程式設計方式在現有索引鍵值存放區中使用索引鍵值配對，請使用 CloudFront KeyValueStore 服務。

若要在最初建立索引鍵值存放區時在索引鍵值存放區中包含一些索引鍵值配對，請使用 CloudFront 服務。

描述操作

CloudFront API 和 API 都有一個描述操作，該操作會返回有關鍵值存儲的數據： CloudFront KeyValueStore

- CloudFront API 會提供資料，例如商店本身上次修改的狀態和日期。
- 該 CloudFront KeyValueStore API 提供有關存儲資源的內容的數據-存儲中的鍵值對以及內容的大小。

兩個 API 中的描述操作返回略有不同的數據，這些數據標識密鑰值存儲：

- CloudFront API 中的描述操作返回 ETag，UUID 和密鑰值存儲的 ARN。

- 在 CloudFront KeyValueStore API 中的描述操作返回 ETag 和鍵值存儲的 ARN。

> **Note**
>
> 每個描述操作會傳回不同的 ETag。該 ETag 是不可互換的。
> 當您在其中一個 API 中執行操作時，您必須從適當的 API 傳入 ETag。例如，在中的刪除作業中 CloudFront KeyValueStore，傳入您從中描述作業中 CloudFront KeyValueStore取得的 ETag。

範例程式碼 CloudFront KeyValueStore

Example ：呼叫 **`DescribeKeyValueStore`** API 作業

下列範例程式碼示範如何呼叫金鑰值存放區的 `DescribeKeyValueStore` API 作業。

```
const {
  CloudFrontKeyValueStoreClient,
  DescribeKeyValueStoreCommand,
} = require("@aws-sdk/client-cloudfront-keyvaluestore");

require("@aws-sdk/signature-v4-crt");

(async () => {
  try {
    const client = new CloudFrontKeyValueStoreClient({
      region: "us-east-1"
    });
    const input = {
      KvsARN: "arn:aws:cloudfront::123456789012:key-value-store/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
    };
    const command = new DescribeKeyValueStoreCommand(input);

    const response = await client.send(command);
  } catch (e) {
    console.log(e);
  }
})();
```

使用 Lambda @Edge 在邊緣進行自訂

Lambda @Edge 是的延伸 AWS Lambda。Lambda @Edge 是一項運算服務，可讓您執行自訂 Amazon CloudFront 交付內容的函數。您可以在美國東部 (維吉尼亞北部) 的 Lambda 主控台中編寫 Node.js 或 Python 函數。 AWS 區域

然後，您可以在 Lambda 或 CloudFront 主控台中新增觸發器，使函數在較靠近檢視器的 AWS 位置執行，而無需佈建或管理伺服器。或者，您可以使用 Lambda 和 CloudFront API 操作以程式設計方式設定函數和觸發程序。

Lambda@Edge 會自動擴展，從每天幾個請求自動擴展到每秒數千個請求。在距離檢視器 (而非原始伺服器) 較近的 AWS 位置處理要求，可大幅減少延遲並改善使用者體驗。

主題

瞭解 Lambda @Edge 如何處理請求和回應

當您將 CloudFront 分發與 Lambda @Edge 函數產生關聯時，會在節 CloudFront 點 CloudFront 攔截請求和回應。發生下列CloudFront 事件時，您可以執行 Lambda 函數：

- CloudFront 收到來自檢視者的要求時 (檢視者要求)
- 在將請求 CloudFront 轉發到原始（原始請求）之前
- 當 CloudFront 收到來自來源的響應（原始響應）
- CloudFront 返回給查看者的響應之前（查看器響應）

如果您正在使用 AWS WAF，則會在套用任何 AWS WAF 規則後執行 Lambda @Edge 檢視器要求。

如需詳細資訊，請參閱 使用請求和回應 及 Lambda@Edge 事件結構說明頁面。

使用 Lambda @Edge 的方法

使用您的 Amazon CloudFront 分發進行 Lambda @Edge 處理有很多用途。例如：

- Lambda 函數可檢查 Cookie 和重新寫入 URL，讓使用者看到不同版本網頁的 A/B 測試。
- CloudFront 您可以檢查`User-Agent`標頭 (包括裝置相關資訊)，根據檢視者使用的裝置，將不同的物件傳回給檢視者。例如， CloudFront 可以根據其設備的屏幕尺寸返回不同的圖像。同樣，該函數可以考慮標`Referer`題的值，並導致圖像返回 CloudFront 到具有最低可用分辨率的機器人。

- 或者，您可以檢查 Cookie 的其他條件。例如，在銷售服裝的零售網站上，如果您使用 Cookie 來指出使用者為外套選擇了哪種顏色，Lambda 函數可以變更請求，以便 CloudFront傳回所選顏色的夾克影像。
- Lambda 函數可在 CloudFront 檢視器要求或原始要求事件發生時產生 HTTP 回應。
- 函數可以檢查標題或授權令牌，並插入標題以控制對內容的訪問，然後再將請求 CloudFront 轉發到您的來源。
- Lambda 函數也可以讓網路呼叫外部資源，確認使用者登入資料，或擷取額外的內容以自訂回應。

如需更多構想，包括範例程式碼，請參閱Lambda@Edge 範例函數。

如需說明如何在主控台中設定 Lambda @Edge 的程序，請參閱教學課程：建立基本 Lambda @Edge 函數。

開始使用 Lambda 函數 @Edge

透過 Lambda @Edge，您可以使用 CloudFront 觸發程序來叫用 Lambda 函數。當您將CloudFront 分佈與 Lambda 函數相關聯時， CloudFront 會攔截 CloudFront 邊緣位置的請求和回應，並執行函數。Lambda 函數可以改善安全性或自訂接近檢視者的資訊，以提升效能。

下列清單提供如何搭配建立和使用 Lambda 函數的基本概觀 CloudFront。如需 step-by-step 自學課程，請參閱〈〉教學課程：建立基本 Lambda @Edge 函數。

1. 在 AWS Lambda 主控台中，在美國東部 (維吉尼亞北部) 區域建立 Lambda 函數。（或者，您可以通過使用其中一個 AWS SDK 以編程方式創建函數。）
2. 儲存並發佈有編號的函數版本。

   如果您想要變更函數，您必須編輯美國東部 (維吉尼亞北部) 區域內 $LATEST 版本的函數。然後，在將其設置為使用之前 CloudFront，您要發布一個新的編號版本。
3. 將函數與 CloudFront 散佈和快取行為相關聯。然後指定一或多個會導致函數執行的 CloudFront 事件 (觸發器)。例如，您可以建立觸發程序，讓函數在 CloudFront 收到來自檢視器的要求時執行。
4. 當您建立觸發器時，Lambda 會在全球各 AWS 地建立函數的複本。

> **Tip**
>
> 進一步了解如何將 Lambda @Edge 用於您自己的自訂解決方案。深入瞭解如何建立和更新函數、事件結構以及新增 CloudFront 觸發程序。您也可以在Lambda@Edge 範例函數中找到更多想法並取得程式碼範例。

主題

- 教學課程：建立基本 Lambda @Edge 函數

教學課程：建立基本 Lambda @Edge 函數

本教學課程說明如何透過建立和設定可在中執行的範例 Node.js 函數來開始使用 Lambda @Edge CloudFront。此範例會在 CloudFront 擷取檔案時，將 HTTP 安全性標頭新增至回應。（這可以提高網站的安全性和隱私性。）

在本教程中，您不需要自己的網站。但是，當您選擇建立自己的 Lambda @Edge 解決方案時，請遵循類似的步驟，並從相同的選項中進行選取。

主題

- 步驟 1：註冊 AWS 帳戶
- 步驟 2：建立 CloudFront 分佈
- 步驟 3：建立函數
- 步驟 4：添加 CloudFront 觸發器以運行該函數
- 步驟 5：驗證函數正常執行
- 步驟 6：排除問題
- 步驟 7：清除範例資源
- 進一步了解的資源

步驟 1：註冊 AWS 帳戶

如果您尚未這樣做，請註冊 AWS 帳戶. 如需詳細資訊，請參閱 註冊一個 AWS 帳戶。

步驟 2：建立 CloudFront 分佈

在建立範例 Lambda @Edge 函數之前，您必須擁有一個可以使用的 CloudFront 環境，其中包含提供內容的來源。

在此範例中，您建立使用 Amazon S3 儲存貯體做為 CloudFront 分發來源的分發。如果您已有環境可使用，可以略過此步驟。

若要使用 Amazon S3 來源建立 CloudFront 分發

1. 使用一兩個檔案來建立 Amazon S3 儲存貯體，例如範例內容適用的映像檔案。如需說明，請遵循將您的內容上傳到 Amazon S3 中的步驟。請確定您有設定對應的許可，以授予儲存貯體中物件的公有讀取權限。
2. 按照建立 CloudFront Web CloudFront 分發中的步驟，建立分發並將 S3 儲存貯體新增為來源。如果您已有分佈，可以改為新增儲存貯體當做該分佈的來源。

> **Tip**
>
> 請記下您的分佈 ID。稍後在本教學課程中，當您為函數新增 CloudFront 觸發器時，您必須在下拉式清單中為您的發行版選擇 ID，例如。E653W22221KDDL

步驟 3：建立函數

在此步驟中，您可以從 Lambda 主控台的藍圖範本建立 Lambda 函數。該函數添加代碼以更新 CloudFront 發行版中的安全標頭。

建立 Lambda 函式

1. 登入 AWS Management Console 並開啟 AWS Lambda 主控台，位於https://console.aws.amazon.com/lambda/。

> **Important**
>
> 請確定您位於美國東部 -1 (維吉尼亞北部) (us-east-1) AWS 區域 。您必須位在此區域，才能建立 Lambda@Edge 函數。

2. 選擇 Create function (建立函數)。
3. 在 [建立函數] 頁面上，選擇 [使用藍圖]，然後在搜尋欄位**`cloudfront`**中輸入以篩選 CloudFront 藍圖。

> **Note**
>
> CloudFront 藍圖僅在美國東部 -1 (維吉尼亞北部) 區域 (us-east-1) 中提供。

4. 選擇修改 HTTP 回應標頭藍圖作為函數的範本。
5. 輸入以下有關函數的資訊：

   函數名稱

   輸入函數的名稱。

   執行角色

   選擇如何設定函數的許可。若要使用建議的基本 Lambda @Edge 許可政策範本，請選擇從 AWS 政策範本建立新角色。

   角色名稱

   輸入政策範本建立的角色名稱。

   政策範本

   Lambda 會自動新增原則範本基本 Lambda @Edge 許可，因為您選擇 CloudFront藍圖做為函數的基礎。此原則範本新增執行角色權限，可 CloudFront 讓您在世界各 CloudFront 地為您執行 Lambda 函數。如需詳細資訊，請參閱 設定身分與存取權管理權限和角色 @Edge。
6. 選擇 Create function (建立函數)。
7. 在出現的「部署至 Lambda @Edge」窗格中，選擇「取消」。在本教學課程中，您必須先修改函數程式碼，然後再將函數部署至 Lambda @Edge。)
8. 向下捲動至頁面的「程式碼來源」區段。
9. 使用修改原始伺服器傳回之安全性標頭的函數，來取代範本程式碼。例如，您可能會使用與下列類似的程式碼：

```
'use strict';
exports.handler = (event, context, callback) => {

    //Get contents of response
    const response = event.Records[0].cf.request;
    const headers = response.headers;

    //Set new headers
```

```
    headers['strict-transport-security'] = [{key: 'Strict-Transport-Security',
 value: 'max-age= 63072000; includeSubdomains; preload'}];
    headers['content-security-policy'] = [{key: 'Content-Security-Policy', value:
 "default-src 'none'; img-src 'self'; script-src 'self'; style-src 'self'; object-
src 'none'"}];
    headers['x-content-type-options'] = [{key: 'X-Content-Type-Options', value:
 'nosniff'}];
    headers['x-frame-options'] = [{key: 'X-Frame-Options', value: 'DENY'}];
    headers['x-xss-protection'] = [{key: 'X-XSS-Protection', value: '1;
 mode=block'}];
    headers['referrer-policy'] = [{key: 'Referrer-Policy', value: 'same-origin'}];

    //Return modified response
    callback(null, response);
};
```

10. 選擇 [檔案]、[儲存] 以儲存更新的程式碼。

繼續下一節以新增執行函數的 CloudFront 觸發器。

步驟 4：添加 CloudFront 觸發器以運行該函數

現在您已經有了 Lambda 函數來更新安全標頭，請將 CloudFront 觸發器設定為執行函數，以便在從發佈的來源 CloudFront 接收到的任何回應中新增標頭。

若要設定功能的 CloudFront觸發器

1. 在 Lambda 主控台中，在函數的 [函數概觀] 頁面上，選擇 [新增觸發器]。
2. 針對 [觸發器] 組態，選擇CloudFront。
3. 選擇「部署 Lambda @Edge」。
4. 在「部署至 Lambda @Edge」窗格的「設定 CloudFront觸發器」下，輸入下列資訊：

   發佈

   與函數相關聯的 CloudFront 分發 ID。在下拉式清單中，選擇分發 ID。

   快取行為

   要用於觸發條件的快取行為。在此範例中，請將值的設定保留為 *，這代表分佈的預設快取行為。如需詳細資訊，請參閱 發佈設定參考 主題中的 快取行為設定。

CloudFront 事件

指定函數何時執行的觸發條件。我們希望每當從來源 CloudFront 返回響應時運行安全頭函數。因此，在下拉列表中，選擇 Origin 響應。如需詳細資訊，請參閱 為 Lambda @Edge 函數新增觸發程序。

5. 選取 [確認部署至 Lambda @Edge] 核取方塊。
6. 選取 Deploy (部署) 新增觸發條件，並將函數複寫到全球各地的 AWS 據點。
7. 請等候函數完成複寫。這通常需要幾分鐘的時間。

   您可以通過轉到 CloudFront 控制台並查看您的發行版來檢查複寫是否已完成。等待發佈狀態從 [部署] 變更為日期和時間，這表示您的函數已複寫。若要驗證函數是否可正常運作，請依照下一節的步驟進行。

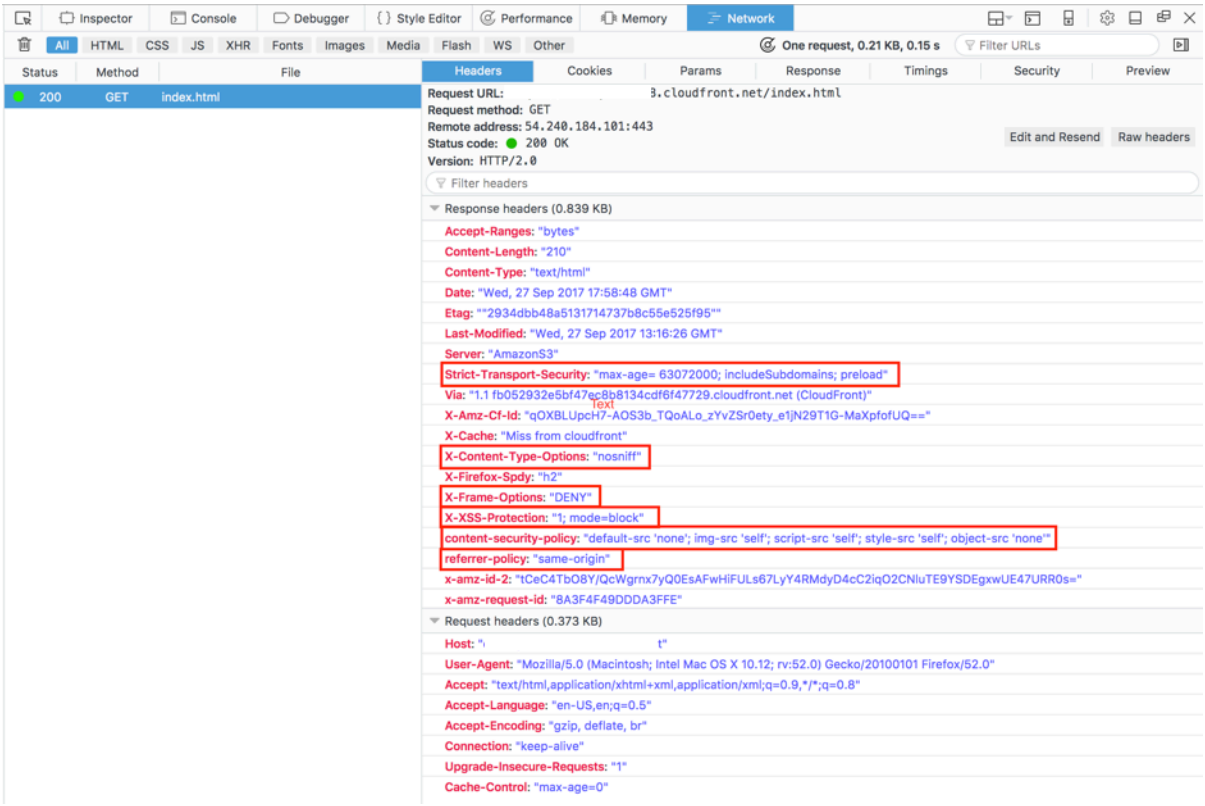
步驟 5：驗證函數正常執行

既然您已建立 Lambda 函數，並設定觸發器以針對 CloudFront分發執行，請檢查以確定函數已完成您預期的功能。在此示例中，我們檢查 CloudFront 返回的 HTTP 標頭，以確保添加了安全標頭。

驗證您的 Lambda@Edge 函數有新增安全標頭

1. 在瀏覽器中，輸入 S3 儲存貯體中某個檔案的 URL。例如，您可以使用類似以下的 URL：`https://d111111abcdef8.cloudfront.net/image.jpg`。

   如需要在檔案 URL 中使用的 CloudFront 網域名稱的詳細資訊，請參閱自訂中檔案的 URL 格式 CloudFront。
2. 開啟瀏覽器的網頁開發人員工具列。例如，在 Chrome 瀏覽器視窗中，開啟內容 (按一下滑鼠右鍵) 選單，然後選擇 Inspect (檢查)。
3. 選擇 Network (網路) 索引標籤。
4. 重新載入頁面來查看您的影像，然後選擇左窗格中的 HTTP 請求。您會看到 HTTP 標頭顯示在個別的窗格中。
5. 查看 HTTP 標頭清單，以驗證預期的安全標頭包含在清單中。例如，您可能會看到類似於以下螢幕擷取畫面所示的標頭。



如果安全標頭包含在您的標頭清單中，那就太棒了！這表示您已成功建立第一個 Lambda@Edge 函數。如果 CloudFront 傳回錯誤或有其他問題，請繼續執行下一個步驟以疑難排解問題。

步驟 6：排除問題

如果 CloudFront 返回錯誤或未按預期添加安全標頭，則可以通過查看 CloudWatch 日誌來調查函數的執行情況。請務必使用儲存在最 AWS 接近執行函數位置的記錄檔。

例如，如果您從倫敦檢視檔案，請嘗試將 CloudWatch 主機中的 [區域] 變更為 [歐洲 (倫敦)]。

若要 CloudWatch 檢查您的 Lambda @Edge 函數的記錄

1. 請登入 AWS Management Console 並開啟 CloudWatch 主控台，網址為 https://console.aws.amazon.com/cloudwatch/。
2. 將 Region (區域) 變更為您在瀏覽器中檢視檔案時所顯示的位置。這是函數執行所在的位置。
3. 在左窗格中，選擇 Logs (日誌) 來檢視分佈的日誌。

如需詳細資訊，請參閱 使用 Amazon CloudFront 監控指標 CloudWatch。

步驟 7：清除範例資源

如果您只為本教學建立 Amazon S3 儲存貯體並 CloudFront 分發，請刪除您分配的 AWS 資源，以免再產生費用。刪除 AWS 資源後，您新增的任何內容將無法再使用。

工作

- 刪除 S3 儲存貯體
- 刪除 Lambda 函數
- 刪除分 CloudFront配

刪除 S3 儲存貯體

刪除 Amazon S3 儲存貯體之前，請務必停用儲存貯體的記錄。否則，當您刪除值區時，會 AWS 繼續將記錄寫入值區。

停用儲存貯體的記錄

1. 在以下網址開啟 Amazon S3 主控台：https://console.aws.amazon.com/s3/。
2. 選取您的儲存貯體，然後選擇 Properties (屬性)。
3. 從 Properties (屬性) 選擇 Logging (記錄日誌)。
4. 清除 Enabled (已啟用) 核取方塊。
5. 選擇 Save (儲存)。

現在即可刪除儲存貯體。如需詳細資訊，請參閱 Amazon Simple Storage Service 主控台使用者指南中的我該如何刪除 S3 儲存貯體？。

刪除 Lambda 函數

如需刪除 Lambda 函數關聯以及選擇性地刪除函數本身的指示，請參閱刪 Lambda 函數和複本 @Edge。

刪除分 CloudFront配

刪除 CloudFront 發行版之前，您必須先將其停用。已停用的分佈如此即不再有用，也不會產生費用。您隨時都可以啟用之前停用的分佈。刪除停用的分佈之後，它即不再可供使用。

停用並刪除 CloudFront 分佈

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。

2. 選取您要停用的分佈，然後選擇 Disable (停用)。
3. 出現確認提示時，請選擇 Yes, Disable (是，停用)。
4. 選取已停用的分佈，然後選擇 Delete (刪除)。
5. 出現確認提示時，選擇 Yes, Delete (是，刪除)。

進一步了解的資源

現在您對於 Lambda@Edge 函數的運作方式已有了基本了解，請閱讀以下內容來進一步了解：

- Lambda@Edge 範例函數
- Lambda 計最佳實務 @Edge
- 使用 Lambda @Edge 減少延遲並將運算轉移到邊緣

設定身分與存取權管理權限和角色 @Edge

若要設定 Lambda @Edge，您必須擁有下列的身分與存取權管理權限和角色：

- IAM 許可 — 這些許可允許您創建 AWS Lambda 功能並將其與 CloudFront 分發相關聯。
- Lambda 函數執行角色 (IAM 角色) — Lambda 服務主體會擔任此角色來執行您的函數。
- Lambda @Edge 的服務連結角色 — 服務連結角色可讓特定角色將 Lambda 函數複寫 AWS 服務 至記錄檔， AWS 區域 並讓其能 CloudWatch夠使用 CloudFront 記錄檔。

將 Lambda @Edge 函數與 CloudFront 分發產生關聯所需的 IAM 許可

除了 Lambda 所需的 IAM 許可之外，您還需要下列許可，才能將 Lambda 函數與 CloudFront 分發產生關聯：

- `lambda:GetFunction`— 授予取得 Lambda 函數組態資訊的權限，以及預先簽署的 URL，以下載包含函數的`.zip`檔案。
- `lambda:EnableReplication*`— 授予資源政策的權限，以便 Lambda 複寫服務可以取得函數程式碼和組態。
- `lambda:DisableReplication*`— 授予資源政策的權限，以便 Lambda 複寫服務可以刪除函數。

> **Important**
>
> 您必須在`lambda:EnableReplication*`和`lambda:DisableReplication*`動作的結尾加入星號 (*)。

- 對於資源，請指定 CloudFront 事件發生時要執行的函數版本的 ARN，例如下列範例：

  `arn:aws:lambda:us-east-1:123456789012:function:`*`TestFunction`*`:2`

- `iam:CreateServiceLinkedRole`— 授予建立 Lambda @Edge 用來在 CloudFront中複寫 Lambda 函數的服務連結角色的權限。第一次設定 Lambda @Edge 之後，系統會自動為您建立服務連結角色。您不需要將此權限新增至使用 Lambda @Edge 的其他發行版本。

- `cloudfront:UpdateDistribution`或 `cloudfront:CreateDistribution` — 授予更新或建立發行版的權限。

如需詳細資訊，請參閱下列主題：

- Amazon Identity and Access Management CloudFront
- AWS Lambda 開發人員指南中的 Lambda 資源存取權限

服務主體的函數執行角色

您必須建立 IAM 角色，以`lambda.amazonaws.com`及`edgelambda.amazonaws.com`服務主體在執行您的函數時可以承擔這個角色。

> **Tip**
>
> 在 Lambda 主控台中建立函數時，您可以選擇使用 AWS 政策範本建立新的執行角色。此步驟會自動新增必要的 Lambda @Edge 許可以執行您的函數。請參閱教學課程中的步驟 5：建立簡單的 Lambda @Edge 函數。

如需有關手動建立 IAM 角色的詳細資訊，請參閱 IAM 使用者指南中的建立角色和附加政策 (主控台)。

Example 範例：角色信任原則

您可以在 IAM 主控台的「信任關係」索引標籤下新增此角色。請勿在 [權限] 索引標籤下新增此原則。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "Service": [
               "lambda.amazonaws.com",
               "edgelambda.amazonaws.com"
            ]
         },
         "Action": "sts:AssumeRole"
      }
   ]
}
```

如需需要授與執行角色之權限的詳細資訊，請參閱AWS Lambda 開發人員指南中的 Lambda 資源存取權限。

備註

- 根據預設，每當 CloudFront 事件觸發 Lambda 函數時，資料就會寫入 CloudWatch 日誌。如果您想要使用這些記錄檔，執行角色需要將資料寫入 CloudWatch 記錄檔的權限。您可以使用預先定義的AWSLambdaBasicExecutionRole來授與執行角色的權限。

  如需有關 CloudWatch 記錄檔的詳細資訊，請參閱the section called “邊緣函數日誌”。
- 如果您的 Lambda 函數程式碼存取其他 AWS 資源，例如從 S3 儲存貯體讀取物件，則執行角色需要執行該動作的權限。

Lambda@Edge 的服務連結角色

Lambda @Edge 使用 IAM 服務連結角色。服務連結角色是直接連結至服務的一種特殊 IAM 角色類型。服務連結角色由服務預先定義，並包含該服務在代表您呼叫其他 AWS 服務時，需要用到的所有權限。

Lambda @Edge 使用下列 IAM 服務連結角色：

- AWSServiceRoleForLambdaReplicator – Lambda@Edge 使用此角色讓 Lambda@Edge 將函數複寫至 AWS 區域。

當您第一次在中新增 Lambda @Edge 觸發程序時 CloudFront，會自動建立名 AWSServiceRoleForLambdaReplicator為的角色，以允許 Lambda @Edge 將函數複寫到 AWS 區域。需要此角色才能使用 Lambda @Edge 函數。角色的 ARN 看 AWSServiceRoleForLambdaReplicator起來像下列範例：

```
arn:aws:iam::123456789012:role/aws-service-role/
replicator.lambda.amazonaws.com/AWSServiceRoleForLambdaReplicator
```

- AWSServiceRoleForCloudFrontLogger— CloudFront 使用此角色將記錄檔推送至 CloudWatch。您可以使用記錄檔來偵錯 Lambda @Edge 驗證錯誤。

  當您新增 Lambda @Edge 函數關聯時，會自動建立AWSServiceRoleForCloudFrontLogger 角色，以 CloudFront 允許將 Lambda @Edge 錯誤記錄檔推送至 CloudWatch。AWSServiceRoleForCloudFrontLogger 角色的 ARN 看起來類似如下：

  ```
  arn:aws:iam::account_number:role/aws-service-role/
  logger.cloudfront.amazonaws.com/AWSServiceRoleForCloudFrontLogger
  ```

服務連結角色可讓設定及使用 Lambda@Edge 變得更輕鬆，因為您不必手動新增必要的許可。Lambda@Edge 會定義其服務連結角色的許可，而且只有 Lambda@Edge 能夠擔任此角色。已定義的許可包括信任政策和許可政策。許可政策無法連接到其他任何 IAM 實體。

您必須先移除任何關聯 CloudFront 或 Lambda @Edge 資源，才能刪除服務連結角色。這有助於保護您的 Lambda @Edge 資源，這樣您就不會移除仍然需要存取作用中資源的服務連結角色。

如需服務連結角色的詳細資訊，請參閱服務連結角色 CloudFront。

Lambda@Edge 的服務連結角色許可

Lambda@Edge 使用兩個服務連結角色，分別名為 AWSServiceRoleForLambdaReplicator 及 AWSServiceRoleForCloudFrontLogger。以下章節說明這些角色的許可。

內容

Lambda Replicator 的服務連結角色許可

這個服務連結的角色可讓 Lambda 將 Lambda@Edge 函式複製到 AWS 區域。

AWSServiceRoleForLambdaReplicator 服務連結角色信任 `replicator.lambda.amazonaws.com` 服務來擔任該角色。

角色許可政策允許 Lambda@Edge 在指定資源上完成下列動作：

- `arn:aws:lambda:*:*:function:*` 的 `lambda:CreateFunction`
- `arn:aws:lambda:*:*:function:*` 的 `lambda:DeleteFunction`
- `arn:aws:lambda:*:*:function:*` 的 `lambda:DisableReplication`
- `all AWS resources` 的 `iam:PassRole`
- `all AWS resources` 的 `cloudfront:ListDistributionsByLambdaFunction`

記錄器 CloudFront的服務連結角色權限

此服務連結角色可 CloudFront 將記錄檔推入，以 CloudWatch 便偵錯 Lambda @Edge 驗證錯誤。

AWSServiceRoleForCloudFrontLogger 服務連結角色信任 `logger.cloudfront.amazonaws.com` 服務來擔任該角色。

角色權限政策允許 Lambda @Edge 在指定`arn:aws:logs:*:*:log-group:/aws/cloudfront/*`資源上完成下列動作：

- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:PutLogEvents`

您必須設定許可，允許 IAM 實體 (例如使用者、群組或角色) 刪除 Lambda@Edge 服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

建立 Lambda@Edge 的服務連結角色

一般而言，您不需要手動建立 Lambda@Edge 的服務連結角色。此服務會在以下情境為您自動建立角色：

- 當您第一次建立觸發器時，服務會建立AWSServiceRoleForLambdaReplicator角色 (如果該角色尚未存在)。此角色可讓 Lambda 將 @Edge 函數複寫到 AWS 區域.

  如果您刪除服務連結角色，則當您在分佈中為 Lambda@Edge 新增觸發條件時，將會重新建立此角色。

- 當您更新或建立具有 Lambda @Edge 關聯的 CloudFront 發佈時，服務會建立 AWSServiceRoleForCloudFrontLogger角色 (如果該角色尚未存在)。此角色可 CloudFront 讓您將記錄檔推送至 CloudWatch。

  如果您刪除服務連結角色，則當您更新或建立具有 Lambda @Edge 關聯的 CloudFront 發佈時，會再次建立該角色。

若要手動建立這些服務連結角色，您可以執行下列 AWS Command Line Interface (AWS CLI) 命令：

建立 AWSServiceRoleForLambdaReplicator 角色

- 執行下列命令。

```
aws iam create-service-linked-role --aws-service-name
 replicator.lambda.amazonaws.com
```

建立 AWSServiceRoleForCloudFrontLogger 角色

- 執行下列命令。

```
aws iam create-service-linked-role --aws-service-name
 logger.cloudfront.amazonaws.com
```

編輯 Lambda@Edge 服務連結角色

Lambda @Edge 不允許您編輯AWSServiceRoleForLambdaReplicator或 AWSServiceRoleForCloudFrontLogger服務連結的角色。服務建立服務連結角色之後，您就無法變更角色的名稱，因為各種實體可能會參照該角色。不過，您可以使用 IAM 來編輯角色描述。如需詳細資訊，請參閱 IAM 使用者指南中的編輯服務連結角色。

支援 CloudFront 服 AWS 區域 務連結角色

CloudFront 在下列情況下，支援針對 Lambda @Edge 使用服務連結角色： AWS 區域

- 美國東部 (維吉尼亞北部) – `us-east-1`
- 美國東部 (俄亥俄) – `us-east-2`
- 美國西部 (加利佛尼亞北部) – `us-west-1`
- 美國西部 (奧勒岡) – `us-west-2`

- 亞太區域 (孟買) – ap-south-1
- 亞太區域 (首爾) – ap-northeast-2
- 亞太區域 (新加坡) – ap-southeast-1
- 亞太區域 (雪梨) – ap-southeast-2
- 亞太區域 (東京) – ap-northeast-1
- 歐洲 (法蘭克福) – eu-central-1
- 歐洲 (愛爾蘭) – eu-west-1
- 歐洲 (倫敦) – eu-west-2
- 南美洲 (聖保羅) – sa-east-1

撰寫並建立 Lambda 函數 @Edge

若要使用 Lambda @Edge，您可以撰寫 AWS Lambda 函數的程式碼。接下來，您將 Lambda 設定為根據特定 CloudFront事件 (稱為觸發程序) 執行函數。

您可以使用 AWS Management Console 來處理 Lambda 函數和 CloudFront 觸發程序，也可以使用 API 以程式設計方式使用 Lambda @Edge。

主題

- 撰寫您 Lambda @Edge 函數
- 建立 Lambda 函數 @Edge
- 變更您 Lambda 函數

撰寫您 Lambda @Edge 函數

若要協助您撰寫 Lambda @Edge 函數，請參閱下列資源：

- Lambda@Edge 事件結構說明頁面— 瞭解與 Lambda @Edge 搭配使用的事件結構。
- Lambda@Edge 範例函數— 範例函數，例如 A/B 測試和產生 HTTP 重新導向。

使用 Node.js 或 Python 搭配使用 Lambda @Edge 的程式設計模型與在 AWS 區域. 如需詳細資訊，請參閱AWS Lambda 開發人員指南中的使用 Node.js 建置 Lambda 函數或使用 Python 建置 Lambda 函數。

在您的 Lambda @Edge 函數中，包含callback參數，並傳回要求或回應事件的適用物件：

- 請求事件 – 在回應中包含 `cf.request` 物件。

  如果您要產生回應，請在回應中包含 `cf.response` 物件。如需詳細資訊，請參閱 在要求觸發程序中產生 HTTP 回應。
- 回應事件 – 在回應中包含 `cf.response` 物件。

建立 Lambda 函數 @Edge

若要設定執行 AWS Lambda 以 CloudFront 事件為基礎的 Lambda 函數，請遵循此程序。

若要建立 Lambda @Edge 函數 (主控台)

1. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。
2. 如果您已擁有一個或多個 Lambda 函數，請選擇 Create function (建立函數)。

   如果您未擁有任何函數，請選擇 Get Started Now (立即開始)。
3. 在頁面頂端的「區域」清單中，選擇美國東部 (維吉尼亞北部)。
4. 使用您自己的程式碼建立函數，或建立以CloudFront 藍圖開始的函數。
   - 若要使用自己的程式碼來建立函數，請選擇 Author from scratch (從頭開始編寫)。
   - 若要顯示的藍圖清單 CloudFront，請在篩選器欄位中輸入 cloudfront，然後選擇 [輸入]。

     如果找到想要使用的藍圖，請選擇該藍圖名稱。
5. 在 Basic information (基本資訊) 區段中，指定下列的值：
   a. 名稱 — 輸入函數的名稱。
   b. 角色 — 若要快速開始使用，請選擇 [從範本建立新角色]。您也可以選擇 [選擇現有角色] 或 [建立自訂角色]，然後依照提示完成此區段的資訊。
   c. 角色名稱 — 輸入角色的名稱。
   d. 政策範本 — 選擇基本邊緣 Lambda 許可。
6. 如果您在步驟 4 中選擇 Author from scratch (從頭開始編寫)，請跳到步驟 7。

   如果您在步驟 4 中選擇藍圖，cloudfront 區段可讓您建立一個觸發器，將此函數與 CloudFront 分發和 CloudFront 事件中的快取建立關聯。我們建議您在此處選擇 Remove (移除)，如此函數在建立時就不會有觸發條件。您可以在稍後新增觸發。

> **Tip**
>
> 我們建議您在新增觸發程序之前測試和偵錯函式。如果您現在新增觸發程序，則函數會在您建立函數並完成複寫至全球各 AWS 地的位置後立即執行，並且會部署對應的散發。

7. 選擇 Create function (建立函數)。

   Lambda 會建立兩個版本的函數：$LATEST 和 Version 1。您只能編輯 $LATEST 版本，但是主控台最初會顯示 Version 1。

8. 若要編輯函數，請選擇該函數 ARN 下方、靠近頁面頂端的 Version 1 (版本 1)。接著，在 Versions (版本) 索引標籤中，選擇 $LATEST ($LATEST)。(如果離開函數再返回，按鈕的標籤會是 Qualifiers (修飾詞)。)
9. 在 Configuration (組態) 索引標籤中，選擇適用的 Code entry type (程式碼項目類型)。然後遵循提示來編輯或上傳程式碼。
10. 針對 Runtime (執行時間)，根據函數的程式碼來選擇值。
11. 在 Tags (標籤) 區段中，新增任何適用的標籤。
12. 選擇 Actions (動作)，然後選擇 Publish new version (發佈新版本)。
13. 輸入函數新版本的說明。
14. 選擇 Publish (發佈)。
15. 對函數進行測試與偵錯。如需利用 Lambda 主控台進行測試的詳細資訊，請在 AWS Lambda 開發人員指南中，請參閱使用主控台建立 Lambda 函數的呼叫 Lambda 函數並驗證結果、日誌和指標一節。
16. 當您準備好為 CloudFront 事件執行函數時，請發布另一個版本並編輯該函數以添加觸發器。如需詳細資訊，請參閱 為 Lambda @Edge 函數新增觸發程序。

使用 API 或使 AWS CLI 用 Lambda @Edge

您也可以使用 Lambda 和 CloudFront API 作業來設定 Lambda @Edge 函數，並以程式設計方式 CloudFront 觸發。如需詳細資訊，請參閱下列主題：

- AWS Lambda API 參考
- Amazon CloudFront API 參考
- 您也可以使用下列 AWS Command Line Interface (AWS CLI) 指令：
  - Lambda 建函數

- CloudFront 創建分佈
- CloudFront create-distribution-with-tags
- CloudFront 更新分發

- AWS SDK（請參閱 SDK 和工具包部分。）
- AWS Tools for PowerShell 指令程式參考

變更您 Lambda 函數

建立 Lambda @Edge 函數之後，您可以使用 Lambda 主控台對其進行變更。

> **備註**
>
> - 原始版本標記為 $LATEST。
> - 您只能編輯 $LATEST 版本。
> - 每次您編輯 $LATEST 版本時，必須發佈新的編號版本。
> - 您無法為 $LATEST 建立觸發。
> - 當您發佈新函數版本時，Lambda 不會自動將觸發條件由前一個版本複製到新版本。您必須為新版本重新產生觸發。
> - 當您將 CloudFront 事件的觸發器新增至函數時，如果相同函數的舊版本已經有相同散佈、快取行為和事件的觸發程序，Lambda 會從舊版中刪除觸發器。
> - 對發行 CloudFront 版進行更新 (例如新增觸發程序) 之後，您必須等待變更傳播到邊緣位置，才能在觸發程序中指定的函數運作。

若要變更 Lambda 函數 (主控台)

1. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。
2. 在頁面頂端的「區域」清單中，選擇美國東部 (維吉尼亞北部)。
3. 在函數清單中，選擇函數的名稱。

   根據預設，主控台會顯示 $LATEST 版本。您可以檢視較早的版本 (選擇 Qualifiers (修飾詞))，但是只能編輯 $LATEST。

4. 在程式碼索引標籤上，針對程式碼項目類型，選擇在瀏覽器中編輯程式碼、上傳 .zip 檔案，或從 Amazon S3 上傳檔案。
5. 選擇 Save (儲存) 或 Save and test (儲存並測試)。
6. 選擇 Actions (動作)，然後選擇 Publish new version (發佈新版本)。
7. 在 Publish new version from $LATEST (從 $LATEST 發佈新版本) 對話方塊中，輸入新版本的說明。此說明會與自動產生的版本編號一起顯示在版本清單中。
8. 選擇 Publish (發佈)。

   新版本會自動成為最新版本。版本號碼會顯示在頁面左上角的「版本」上。
9. 選擇 Triggers (觸發條件) 索引標籤。
10. 選擇 Add trigger (新增觸發條件)。
11. 在 [新增觸發器] 對話方塊中，選擇虛線方塊，然後選擇CloudFront。

> **Note**
>
> 如果您已經為函數建立了一或多個觸發程序， CloudFront 就是預設服務。

12. 指定下列值，以指示您希望 Lambda 函數在何時執行。

    a. 分佈 ID — 選擇您要新增觸發程式的分佈 ID。
    b. 快取行為 — 選擇指定要在其上執行函數之物件的快取行為。
    c. CloudFront 事件 — 選擇會導致函數執行的 CloudFront 事件。
    d. 啟用觸發和複寫 — 選取此核取方塊，以便 Lambda 將函數複寫到 AWS 區域 全域。
13. 選擇提交。
14. 若要為此函數新增更多觸發，請重複操作步驟 10 到 13。

為 Lambda @Edge 函數新增觸發程序

Lambda @Edge 觸發程序是導致函數執行的 CloudFront 散佈、快取行為和事件的其中一個組合。您可以指定一或多個會導致函數執行的 CloudFront 觸發程序。例如，您可以建立一個觸發程序，在 CloudFront 收到檢視器針對您為發佈設定的特定快取行為的要求時，執行函數。

> **Tip**
>
> 當您建立 CloudFront 發佈時，您可以指定設定，以便在收到不同的要求時告知 CloudFront 如何回應。預設設定稱為散發的預設快取行為。您可以設定其他快取行為，以定義在特定情況下

的 CloudFront 回應方式，例如，當它收到特定檔案類型的要求時。如需詳細資訊，請參閱快取行為設定。

當您建立 Lambda 函數時，只能指定一個觸發條件。您可以稍後使用 Lambda 主控台或在主控台中編輯分發，將更多觸發程序新增至相同函數。 CloudFront

- 如果您想要為相同發CloudFront 行版的函數新增更多觸發程序，Lambda 主控台運作良好。
- 如果您想為多個發行版添加觸發器，則 CloudFront 控制台可能會更好，因為找到要更新的發行版更容易。您也可以同時更新其他 CloudFront 設定。

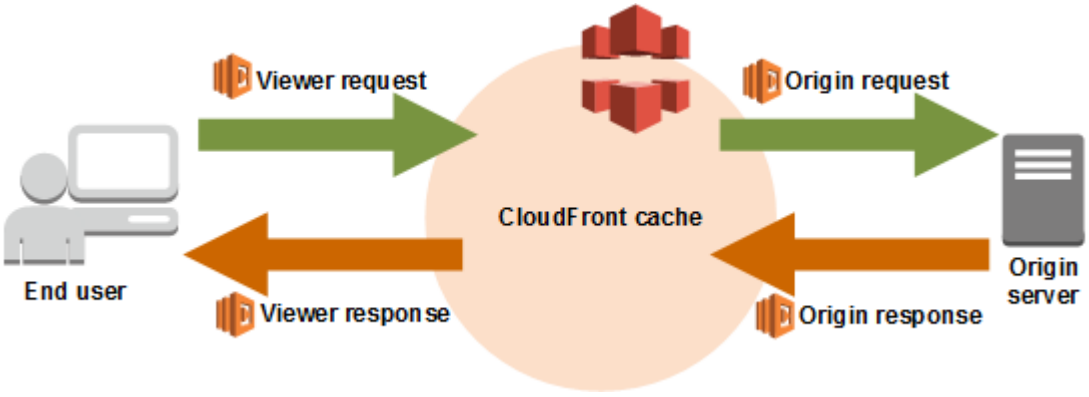
> **Note**
>
> 若要以程式設計方式使用 Lambda @Edge，請參閱 使用 API 或使 AWS CLI 用 Lambda @Edge

主題

- CloudFront 可以觸發 @Edge 函數的事件
- 決定要使用哪個 CloudFront 事件觸發 Lambda @Edge 函數
- 將觸發程序新增 Lambda @Edge 函數

CloudFront 可以觸發 @Edge 函數的事件

對於 Amazon CloudFront 分發中的每個快取行為，您最多可以新增四個觸發器 (關聯)，這些觸發器會在特定 CloudFront 事件發生時執行 Lambda 函數。 CloudFront 觸發器可以基於四個 CloudFront 事件之一，如下圖所示。



可用來觸發 Lambda @Edge 函數的 CloudFront 事件如下：

檢視者請求

該函數在 CloudFront 收到來自查看器的請求時執行，然後再檢查請求的對象是否在 CloudFront 緩存中。

原始伺服器請求

該函數僅在將請求 CloudFront 轉發到您的來源時執行。當請求的對象在 CloudFront 緩存中時，該函數不會執行。

原始伺服器回應

該函數在 CloudFront 收到來自來源的響應之後以及在響應中緩存對象之前執行。請注意，即使原始伺服器傳回錯誤，函數仍會執行。

函數不在以下情況執行：

- 當請求的文件在 CloudFront 緩存中並且未過期時。
- 當回應是從原始伺服器請求事件觸發的函數所產生。

檢視者回應

函數會在請求的檔案傳回給檢視器之前執行。請注意，無論文件是否已經在 CloudFront緩存中，該函數都會執行。

函數不在以下情況執行：

- 當原始伺服器傳回 HTTP 狀態碼 400 或更高版本。
- 當傳回自訂錯誤頁面。
- 當回應是被檢視器請求事件觸發的函數所產生。
- 當 CloudFront 自動將 HTTP 要求重新導向至 HTTPS 時 (當的值檢視器通訊協定政策是將 HTTP 重新導向至 HTTPS 時)。

當您新增多個觸發條件至相同的快取行為時，可將其用於針對每個觸發條件執行相同或不同的函數。您也可以將相同的函數與一個以上的分佈建立關聯。

> **Note**
>
> 當 CloudFront 事件觸發 Lambda 函數的執行時，函數必須先完成，才 CloudFront 能繼續。例如，如果 Lambda 函數是由CloudFront 檢視器要求事件觸發，則在 Lambda 函數完成執行之

前， CloudFront 不會將回應傳回檢視器或將請求轉送至原始位置。這表示觸發 Lambda 函數的每個請求都會增加請求的延遲，因此您會希望函數執行速度越快越好。

決定要使用哪個 CloudFront 事件觸發 Lambda @Edge 函數

當您決定要使用哪個 CloudFront 事件來觸發 Lambda 函數時，請考慮下列事項：

是否 CloudFront 要快取由 Lambda 函數變更的物件？

如果您想 CloudFront 要快取由 Lambda 函數修改的物件， CloudFront 以便在下次要求物件時從邊緣位置提供物件，請使用原始要求或來源回應事件。這樣可以降低原始伺服器的負載，減少後續請求的延遲，並降低叫用 Lambda@Edge 在後續請求的費用。

例如，如果您想要新增、移除或變更來源傳回之物件的標頭，並且想 CloudFront 要快取結果，請使用原始回應事件。

您想要函數執行每個請求嗎？

如果您希望函數針對每個 CloudFront 接收發佈的要求執行，請使用檢視器要求或檢視器回應事件。只有當請求的對象未緩存在邊緣位置並將請求 CloudFront 轉發到來源時，Origin 請求和源響應事件才會發生。

該函數是否有變更快取金鑰？

如果您想要變更函數的值做為快取基礎，請使用檢視器請求事件。例如，若函數變更其 URL 以包含語言縮寫在路徑中 (例如，因為使用者從下拉式清單中選擇其語言)，請使用檢視器請求事件：

- 檢視器要求中的網址 — https://example.com/en/index.html
- 當請求來自德國的 IP 地址時的網址 — https://example.com/de/index.html

如果您快取根據 Cookie 或請求標頭，也可以使用檢視器請求事件。

> **Note**
>
> 如果功能變更 Cookie 或標頭，請設定 CloudFront 為將要求的適用部分轉寄至來源。如需詳細資訊，請參閱下列主題：
>
> - 根據 Cookie 快取內容
> - 根據請求標頭快取內容

函數是否會影響原始伺服器的回應？

如果您希望函數依影響原始伺服器回應的方法變更，請使用原始伺服器請求事件。一般而言，大多數檢視器要求事件不會轉寄至來源；CloudFront 回應要求時會使用已存在於 Edge 快取中的物件。如果函數根據來源要求事件變更要求，則會 CloudFront 快取變更的原始要求的回應。

將觸發程序新增 Lambda @Edge 函數

您可以使用主 AWS Lambda 控台或 Amazon CloudFront 主控台將觸發器新增至您的 Lambda @Edge 函數。

> **Important**
>
> 您只能為函數的編號版本（而不是 $LATEST）創建觸發器。

Lambda console

若要將觸發程式新增至 Lambda @Edge 函數

1. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。
2. 在頁面頂端的「區域」清單中，選擇美國東部 (維吉尼亞北部)。
3. 在 Functions (函數) 頁面上，選擇您要為其新增觸發條件的函數名稱。
4. 在 [函數概觀] 頁面上，選擇 [版本] 索引標籤。
5. 選擇您要為其新增觸發的版本。

   選擇版本之後，按鈕的名稱會變更為 Version: $LATEST (版本：$LATEST) 或 Version: (版本：) 版本編號。
6. 選擇 Triggers (觸發條件) 索引標籤。
7. 選擇 Add trigger (新增觸發條件)。
8. 針對 [觸發器] 組態，選擇 [選取來源]、[輸入]**`cloudfront`**，然後選擇CloudFront。

   > **Note**
   >
   > 如果您已建立一或多個觸發程序，則 CloudFront 為預設服務。
9. 指定下列值，以指示您希望 Lambda 函數在何時執行。

a. 分佈 — 選擇您要新增觸發程式的分佈。

b. 快取行為 — 選擇指定要在其上執行函數之物件的快取行為。

> **Note**
>
> 如果您為快取行為指定 *，Lambda 函式則會部署至預設的快取行為。

c. CloudFront 事件 — 選擇會導致函數執行的CloudFront 事件。

d. 包含主體 — 如果您想要存取函數中的請求主體，請選取此核取方塊。

e. 確認部署至 Lambda @Edge — 選取此核取方塊，以便將函數 AWS Lambda 複寫到 AWS 區域 全域。

10. 選擇新增。

當部署更新的 CloudFront 發佈時，函數會開始處理指定 CloudFront事件的要求。若要判斷是否已部署分佈，請在導覽窗格中選擇 Distributions (分佈)。部署發佈時，發佈的 [狀態] 欄的值會從 [部署] 變更為部署的日期和時間。

CloudFront console

若要將 CloudFront 事件的觸發程序新增至 Lambda 函數

1. 取得您希望新增觸發的 Lambda 函數的 ARN：

   a. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。

   b. 在頁面頂端的區域清單中，選擇美國東部 (維吉尼亞北部)。

   c. 在函數清單上，選取您想要為其新增觸發的函數名稱。

   d. 在 [函數概觀] 頁面上，選擇 [版本] 索引標籤，然後選擇要新增觸發程式的編號版本。

   e. 選擇「複製 ARN」按鈕，將 ARN 複製到剪貼簿。Lambda 函數的 ARN 看起來像這樣：

   ```
   arn:aws:lambda:us-east-1:123456789012:function:TestFunction:2
   ```

   最後面的數字 (此範例中為 2 (2)) 是該函數的版本編號。

2. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。

3. 在分佈清單中，選取您想要為其新增觸發的分佈 ID。

4. 選擇 Behaviors (動作) 索引標籤。

5. 選取您要新增觸發程式的快取行為，然後選擇 [編輯]。
6. 對於函數關聯，請在 [函數類型] 清單中，選擇 Lambda @Edge 來執行函數：針對檢視器要求、檢視器回應、來源請求或來源回應。

   如需詳細資訊，請參閱 決定要使用哪個 CloudFront 事件觸發 Lambda @Edge 函數。
7. 在函數 ARN/名稱文字方塊中，貼上您要在所選事件發生時執行的 Lambda 函數的 ARN。這是您從 Lambda 主控台複製的值。
8. 如果您想要在函數中存取要求主體，請選取「包含內文」。

   如果您只想要替換請求本體，就不需要選取此選項。
9. 若要針對更多事件類型執行相同的函數，請重複步驟 6 和 7。
10. 選擇儲存變更。
11. 若要將觸發程序新增至此發行版的更多快取行為，請重複步驟 5 到 10。

    當部署更新的 CloudFront 發佈時，函數會開始處理指定 CloudFront事件的要求。若要判斷是否已部署分佈，請在導覽窗格中選擇 Distributions (分佈)。部署發佈時，發佈的 [狀態] 欄的值會從 [部署] 變更為部署的時間和日期。

測試和偵 Lambda 函數 @Edge

本主題包括的內容，說明了進行 Lambda@Edge 函數測試與除錯的策略。請務必單獨測試 Lambda @Edge 函數程式碼，以確保它完成預期的工作，並執行整合測試，以確保函數可以正常運作 CloudFront。

在整合測試期間或部署函數之後，您可能需要偵 CloudFront 錯錯誤，例如 HTTP 5xx 錯誤。錯誤可能是從 Lambda 函數傳回的無效回應、觸發函數時的執行錯誤，或是由於 Lambda 服務進行調節所產生的錯誤。本主題中的段落，說明了用來判斷是哪些故障類型造成問題的策略，以及您可採取的問題修正步驟。

> **Note**
>
> 當您在疑難排解錯誤時檢閱 CloudWatch 記錄檔或指標時，請注意這些記錄檔或指標會顯示或儲存在 AWS 區域 距離函數執行位置最近的位置。因此，舉例來說，如果您的網站或 Web 應用程式有英國的使用者，而且您的發佈有關聯的 Lambda 函數，則必須變更區域以檢視倫敦的 CloudWatch 指標或記錄檔 AWS 區域。如需詳細資訊，請參閱 the section called “ 確定 Lambda @Edge 區域”。

主題

測試您 Lambda @Edge 函數

測試您的 Lambda 函數包括兩個步驟：獨立測試和整合測試。

測試獨立的功能

在將 Lambda 函數新增至之前 CloudFront，請務必先使用 Lambda 主控台中的測試功能或使用其他方法來測試功能。如需利用 Lambda 主控台進行測試的詳細資訊，請在 AWS Lambda 開發人員指南中，請參閱使用主控台建立 Lambda 函數的呼叫 Lambda 函數並驗證結果、日誌和指標一節。

測試你的函數的操作 CloudFront

完成集成測試很重要，其中您的函數與分發相關聯，並根據 CloudFront 事件運行。請確定已針對正確的事件觸發函數，並傳回有效且正確的回應 CloudFront。例如，請確定事件結構是正確的，只包含有效的標頭，以此類推。

當您在 Lambda 主控台中使用函數進行整合測試時，請參閱 Lambda @Edge 教學課程中的步驟，同時修改程式碼或變更呼叫函數的 CloudFront 觸發程序。例如，請確定您使用函式的編號版本，如教學課程的這項步驟中所述：步驟 4：添加 CloudFront 觸發器以運行該函數。

當您進行變更並部署它們時，請注意，更新的函數和 CloudFront 觸發程序將需要幾分鐘的時間才能跨所有區域進行複寫。這通常需要幾分鐘的時間，但最多可能需要 15 分鐘。

您可以通過轉到 CloudFront控制台並查看您的發行版來檢查複寫是否已完成。

若要檢查您的複寫是否已完成部署

1. 在開啟 CloudFront 主控台https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇分配名稱。
3. 檢查分佈的狀態是否已從 In Progress (進行中) 變回 Deployed (已部署)，這表示您的函數已複寫完成。接著，請依照下一節的步驟來驗證函數是否正常運作。

請注意，在主控台中進行的測試只會驗證您的函數邏輯，並不會套用 Lambda@Edge 特定的服務配額 (先前稱為限制)。

識別 Lambda 的 @Edge 函數錯誤 CloudFront

在驗證函數邏輯正常工作之後，當您的函數運行時，您可能仍會看到 HTTP 5xx 錯誤。 CloudFront 可能會傳回 HTTP 5xx 錯誤的原因有多種，其中CloudFront可能包括中的 Lambda 函數錯誤或其他問題。

- 如果您使用 Lambda @Edge 函數，則可以使用 CloudFront 主控台中的圖形來協助追蹤造成錯誤的原因，然後進行修正。例如，您可以查看 HTTP 5xx 錯誤是由 Lambda 函數引起 CloudFront 還是由 Lambda 函數引起，然後針對特定函數，您可以檢視相關的記錄檔以調查問題。
- 若要疑難排解中的一般 HTTP 錯誤 CloudFront，請參閱下列主題中的疑難排解步驟：從原始伺服器故障診斷錯誤回應。

導致 Lambda @Edge 函數錯誤的原因 CloudFront

Lambda 函數造成 HTTP 5xx 錯誤的原因有許多種，您應該依據錯誤的類型採取相應的疑難排解步驟。錯誤分類如下：

Lambda 函數執行錯誤

由於函數中存在未處理的例外狀況或程式碼中有錯誤，因此 CloudFront 沒有從 Lambda 取得回應時，就會產生執行錯誤。例如，如果程式碼包含回呼 (錯誤)。如需詳細資訊，請參閱AWS Lambda開發人員指南中的 Lambda 函數錯誤。

傳回無效的 Lambda 函數回應 CloudFront

函數運行後， CloudFront 接收來自 Lambda 的響應。如果回應的物件結構不符合 Lambda@Edge 事件結構說明頁面，或回應中包含無效的標頭或其他無效的欄位，系統會傳回錯誤。

由於 Lambda 服務配額 (先前稱為限制)，中 CloudFront 的執行會受到限制

Lambda 服務會在各區域中調節執行作業，並在您超出配額時傳回錯誤。

如何判斷故障的類型

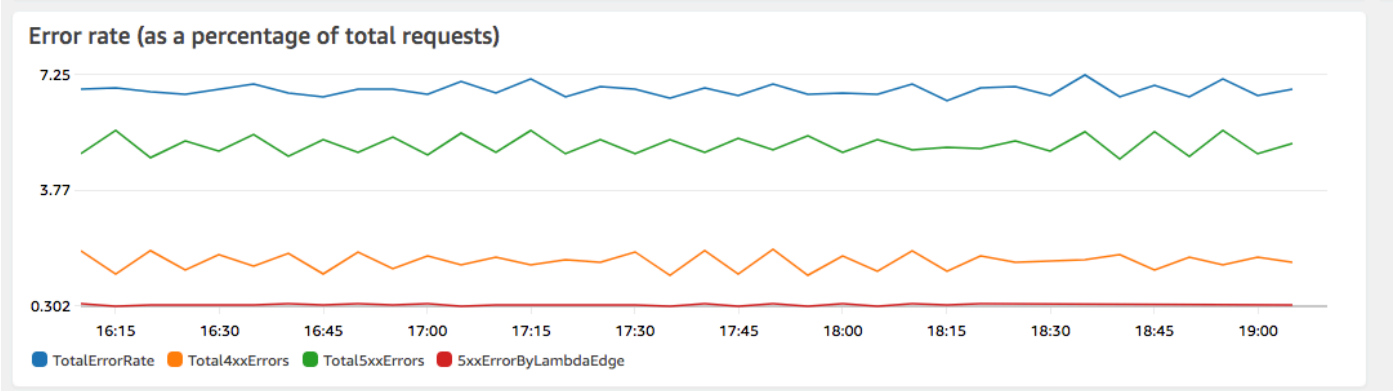
為了協助您決定在偵錯和解決傳回的錯誤時要集中在何處 CloudFront，找出傳回 HTTP 錯誤的原 CloudFront 因會很有幫助。若要開始使用，您可以使用 CloudFront 主控台的 [監視] 區段中提供的圖形

AWS Management Console。如需有關在主控台的 [監視] 區段中檢視圖形的詳細資 CloudFront 訊，請參閱使用 Amazon CloudFront 監控指標 CloudWatch。

以下圖表在您追縱原始伺服器或 Lambda 函數是否傳回錯誤時特別實用，當錯誤是由於 Lambda 函數造成時，也可縮小問題的類型。

錯誤率圖表

在每一個分佈的 Overview (概觀) 標籤上，您可以檢視的其中一個圖表就是 Error rates (錯誤率) 圖表。此圖表顯示錯誤率佔進入您分配的請求總數的百分比。此圖表顯示總錯誤率，總共 4xx 個錯誤、總共 5xx 個錯誤，以及總共 5xx 個 Lambda 函數的錯誤。根據錯誤類型和磁碟區，您可以採取步驟以針對原因進行調查和故障診斷。



- 如果您看到 Lambda 錯誤，您可以透過查看該函數傳回的特定錯誤類型，以進一步進行調查。Lambda@Edge 錯誤標籤包含了依類型分類的函數錯誤圖表，可協助您找出特定函數的問題。
- 如果您看到 CloudFront 錯誤，您可以進行疑難排解並努力修復原始錯誤或變更您的 CloudFront 組態。如需詳細資訊，請參閱 從原始伺服器故障診斷錯誤回應。

執行錯誤和無效函數回應圖表

Lambda@Edge 錯誤標籤包含針對特定分佈 (依類型) 分類 Lambda@Edge 錯誤的圖表。例如，一個圖形顯示所有執行錯誤 AWS 區域。

若要更容易疑難排解問題，您可以依照區域開啟並檢查記錄檔中的特定功能，以尋找特定問題。

若要依區域檢視特定功能的記錄檔

1. 在 Lambda @Edge 錯誤索引標籤的關聯 Lambda @Edge 函數下，選擇函數名稱，然後選擇檢視指標。
2. 接下來，在具有函數名稱的頁面上，選擇右上角的 [檢視功能記錄]，然後選擇 [區域]。

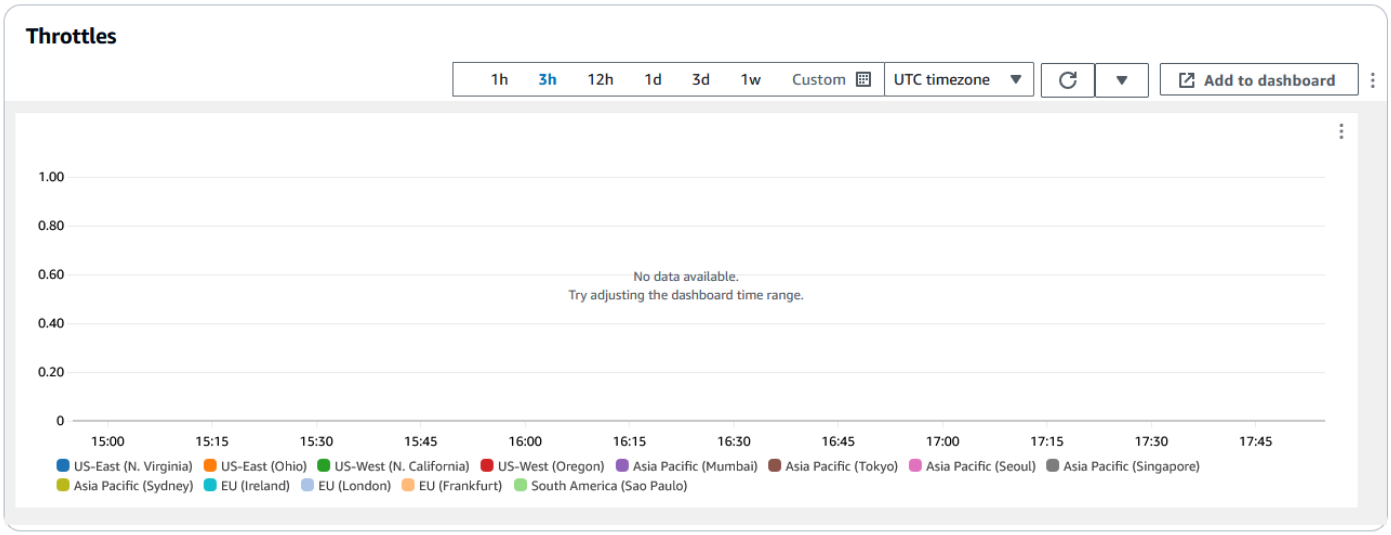
例如，如果您在美國西部 (奧勒岡) 區域的「錯誤」圖表中看到問題，請從下拉式清單中選擇該區域。這將打開 Amazon CloudWatch 控制台。

3. 在該區域的 CloudWatch 主控台的 [記錄串流] 下，選擇記錄資料流以檢視函數的事件。

此外，請閱讀此章的下列各節，以了解有關故障排除和修復錯誤的更多建議。

調節圖表

Lambda@Edge 錯誤標籤也包含調節圖表。有時，如果您到達區域並行數量配額 (先前稱為限制)，則 Lambda 服務會依每一區域為基礎調節您的函數呼叫。如果出現超過限制錯誤，表示您的函數已到達 Lambda 服務對「區域」中的執行作業所施加的配額。如需詳細資訊，包括如何請求提高配額，請參閱Lambda@Edge 的配額。



如需如何使用此資訊進行 HTTP 錯誤故障診斷的詳細資訊，請參閱在 AWS上針對您的內容交付執行偵錯的四個步驟。

疑難排解無效的 Lambda @Edge 函數回應 (驗證錯誤)

如果您發現問題是 Lambda 驗證錯誤，則表示您的 Lambda 函數傳回無效的回應 CloudFront。請遵循本節中的指引，採取步驟檢閱您的功能，並確保您的回應符合需 CloudFront 求。

CloudFront 驗證來自 Lambda 函數的回應有兩種方式：

- Lambda 回應必須符合所請求的物件結構。錯誤的物件結構範例包括：無法剖析的 JSON、遺漏必要的欄位，以及在回應中包含無效的物件。如需更多資訊，請參閱Lambda@Edge 事件結構說明頁面。

- 回應必須只包含有效的物件值。如果回應中包含有效的物件，但是具有不支援的值，將會發生錯誤。此種情況的範例包括：新增或更新被列入不允許或唯讀的標頭 (請參閱 對邊緣函數的限制)、超過內文大小的上限 (請參閱 Lambda@Edge 錯誤 主題中的對所產生回應的大小限制)，以及無效的字元或值 (請參閱 Lambda@Edge 事件結構說明頁面)。

當 Lambda 傳回無效的回應時 CloudFront，會將錯誤訊息寫入日誌檔，該檔會CloudWatch 在 Lambda 函數執行的區域中 CloudFront 推送至該檔案。這是發生無效回應 CloudWatch時將記錄檔傳送至的預設行為。但是，如果您在功能發布 CloudFront 之前將 Lambda 函數與相關聯，則可能不會為您的函數啟用該函數。如需詳細資訊，請參閱主題中的判斷您的帳戶是否將記錄檔推送至 CloudWatch稍後。

CloudFront 將日誌文件推送到與您的分發相關聯的日誌組中的功能執行位置對應的區域。日誌組具有以下格式：`/aws/cloudfront/LambdaEdge/DistributionId`，其中`DistributionId`是分發的 ID。若要判斷可以找到 CloudWatch 記錄檔的區域，請參閱本主題稍後的判斷 Lambda @Edge 區域。

如果錯誤可重現，您可以建立會導致錯誤的新要求，然後在失敗的 CloudFront 回應 (`X-Amz-Cf-Id`標頭) 中尋找要求識別碼，以便在記錄檔中尋找單一失敗。日誌檔案記錄所包含的資訊，可協助您找出傳回錯誤的原因，也可以列出對應的 Lambda 請求 ID，來讓您針對單一請求的範圍，分析錯誤的根本原因。

如果錯誤是間歇性的，您可以使用 CloudFront 存取記錄檔來尋找失敗之要求的要求識別碼，然後搜尋對應錯誤訊息的 CloudWatch 記錄檔。如需詳細資訊，請參閱先前的段落判斷故障的類型。

疑難 Lambda 解函數執行錯誤 @Edge

如果問題是 Lambda 執行錯誤，建立 Lambda 函數的記錄陳述式、將訊息寫入記 CloudWatch 錄檔，以監控函數的執行情況，以 CloudFront 及判斷其是否如預期般運作會很有幫助。然後，您可以在 CloudWatch 記錄檔中搜尋這些陳述式，以確認您的函數是否正常運作。

> **Note**
>
> 即使您未變更您的 Lambda@ Edge 函數，Lambda 函數執行環境的更新仍會對其造成影響，並因而傳回執行錯誤。如需測試及移轉至更新版本的相關資訊，請參閱 AWS Lambda 和 AWS Lambda @Edge 執行環境的即將更新。

確定 Lambda @Edge 區域

若要查看 Lambda @Edge 函數接收流量的區域，請在CloudFront 主控台上檢視該函數的指標 AWS Management Console。顯示每個 AWS 區域的量度。在同一頁面中，您可以選擇一個區域並檢視

該區域的日誌檔，以便調查問題。您必須檢閱正確 AWS 區域中的 CloudWatch 記錄檔，以查看 CloudFront 執行 Lambda 函數時建立的記錄檔。

如需有關在主控台的 [監視] 區段中檢視圖形的詳細資 CloudFront 訊，請參閱使用 Amazon CloudFront 監控指標 CloudWatch。

判斷您的帳戶是否將記錄推送至 CloudWatch

依預設，會 CloudFront 啟用記錄無效的 Lambda 函數回應，並使用其中一個將記錄檔推 CloudWatch 送至. Lambda@Edge 的服務連結角色 如果您在發行無效的 Lambda 函數回應記錄功能 CloudFront 之前已新增 Lambda @Edge 函數，則當您下次更新 Lambda @Edge 組態時 (例如，透過新增 CloudFront觸發程序)，就會啟用記錄功能。

您可以執行下列動作，確認您 CloudWatch 的帳戶已啟用將記錄檔推送至：

- 檢查記錄檔是否出現在中 CloudWatch。請務必查看 Lambda@Edge 函數執行所在的區域。如需詳細資訊，請參閱 確定 Lambda @Edge 區域。
- 判斷您在 IAM 中的帳戶，是否存在相關的服務連結角色。若要這樣做，請在 https://console.aws.amazon.com/iam/, 開啟 IAM 主控台，然後選擇角色以檢視您帳戶的服務連結角色清單。尋找下列的角色：`AWSServiceRoleForCloudFrontLogger`。

刪 Lambda 函數和複本 @Edge

只有在函數的複本已被刪除時，您才可以刪除 Lambda @Edge 函數。 CloudFrontLambda 函數的複本將在下列情況自動刪除：

- 從所有 CloudFront發行版中移除函數的最後一個關聯之後。如果一個以上的分佈使用一個函數，則只有在您從上一個分佈移除函數關聯之後，複本才會刪除。
- 在您刪除最後一個與函數相關聯的分佈後。

複本一般會在幾個小時內刪除。您無法手動刪除 Lambda@Edge 函數複本。這有助於防止刪除仍在使用中的複本，以免導致錯誤情況。

> **Warning**
>
> 請勿在以外建置使用 Lambda @Edge 函數複本的 CloudFront應用程式。當這些複本與分佈的關聯遭到移除，或分佈本身遭到刪除時，這些複本就會隨之刪除。外部應用程式所依據使用的複本可能會在無預警的情況下移除，導致應用程式作業失敗。

若要從 CloudFront 發佈 (主控台) 刪除 Lambda @Edge 函數關聯

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 選擇具有您要刪除之 Lambda @Edge 函數關聯的分佈識別碼。
3. 選擇 Behaviors (動作) 索引標籤。
4. 選取具有您要刪除之 Lambda @Edge 函數關聯的快取行為，然後選擇 [編輯]。
5. 在函數關聯的函數類型下，選擇無關聯以刪除 Lambda @Edge 函數關聯。
6. 選擇儲存變更。

從 CloudFront 分發中刪除 Lambda @Edge 函數關聯後，您可以選擇性地從中刪除 Lambda 函數或函數版本 AWS Lambda。刪除函數關聯後，請等待幾個小時，以便清除 Lambda @Edge 函數複本。之後，您就可以使用 Lambda 主控台 AWS CLI、Lambda API 或 AWS 開發套件來刪除函數。

如果該版本沒有任何與 Lambda 函數相關聯的 CloudFront發行版，您也可以刪除特定版本的 Lambda 函數。移除 Lambda 函數版本的所有關聯之後，請等待幾個小時。然後，您將能夠刪除功能版本。

Lambda@Edge 事件結構說明頁面

下列主題說明觸發時 CloudFront 傳遞至 Lambda @Edge 函數的請求和回應事件物件。

主題

- 動態原始伺服器選擇
- 請求事件
- 回應事件

動態原始伺服器選擇

您可以根據路徑和請求物件名稱 (例如 `images/*.jpg`)，使用快取行為中的路徑模式將請求路由傳送到原始伺服器。使用 Lambda@Edge，您也可以根據其他特性路由傳送請求到原始伺服器，例如請求標頭中的值。

此動態原始伺服器選擇在很多方面是非常有用的。例如，您可以將請求分佈到各個在不同地理區域的原始伺服器，以協助全域負載平衡。或者，您可以選擇性地路由傳送請求到服務特定函數之不同的原始伺服器：機器人處理、SEO 最佳化、驗證，以此類推。如需示範如何使用此功能的程式碼範例，請參閱以內容為基礎的動態原始伺服器選擇 - 範例。

在 CloudFront 原始要求事origin件中，事件結構中的物件會根據路徑模式，包含要求將路由到的來源相關資訊。您可以更新 origin 物件的值，將請求路由傳送到不同的原始伺服器。更新 origin 物件時，您不需要定義分佈中的原始伺服器。您也可以使用自訂原始伺服器物件來取代 Amazon S3 原始伺服器物件，反之亦然。不過，您只能為每個請求指定單一原始伺服器；也就是指定自訂原始伺服器或 Amazon S3 原始伺服器，但不能兩者同時指定。

請求事件

下列主題顯示 CloudFront 傳遞至 Lambda 函數以供檢視器和原始請求事件使用的物件結構。這些範例顯示不含任何主體的 GET 請求。在範例之後，會提供檢視器和原始伺服器請求事件中的所有可能欄位清單。

主題

- 檢視者請求範例
- 原始伺服器請求範例
- 請求事件欄位

檢視者請求範例

下列範例顯示檢視器請求事件物件。

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "viewer-request",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
              }
            ],
            "user-agent": [
```

```
            {
              "key": "User-Agent",
              "value": "curl/7.66.0"
            }
          ],
          "accept": [
            {
              "key": "accept",
              "value": "*/*"
            }
          ]
        },
        "method": "GET",
        "querystring": "",
        "uri": "/"
      }
    }
  }
]
}
```

原始伺服器請求範例

下列範例顯示原始伺服器請求事件物件。

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "origin-request",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "x-forwarded-for": [
              {
                "key": "X-Forwarded-For",
                "value": "203.0.113.178"
              }
            ],
```

```
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "Amazon CloudFront"
              }
            ],
            "via": [
              {
                "key": "Via",
                "value": "2.0 2afae0d44e2540f472c0635ab62c232b.cloudfront.net
(CloudFront)"
              }
            ],
            "host": [
              {
                "key": "Host",
                "value": "example.org"
              }
            ],
            "cache-control": [
              {
                "key": "Cache-Control",
                "value": "no-cache"
              }
            ]
          },
          "method": "GET",
          "origin": {
            "custom": {
              "customHeaders": {},
              "domainName": "example.org",
              "keepaliveTimeout": 5,
              "path": "",
              "port": 443,
              "protocol": "https",
              "readTimeout": 30,
              "sslProtocols": [
                "TLSv1",
                "TLSv1.1",
                "TLSv1.2"
              ]
            }
          },
          "querystring": "",
```

```
          "uri": "/"
        }
      }
    }
  ]
}
```

請求事件欄位

請求事件物件資料包含在兩個子物件中：config (Records.cf.config) 和 request (Records.cf.request)。下列清單說明每個子物件的欄位。

Config 物件中的欄位

下列清單說明 config 物件 (Records.cf.config) 中的欄位。

distributionDomainName (唯讀)

與請求相關的分佈網域名稱。

distributionID (唯讀)

與請求相關的分佈 ID。

eventType (唯讀)

與請求關聯的觸發條件類型：viewer-request 或 origin-request。

requestId (唯讀)

可唯一識別檢視者對象要求的加密字串。CloudFront該requestId值也會在 CloudFront存取記錄中顯示為x-edge-request-id。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌) 及 標準日誌檔案欄位。

請求物件中的欄位

下列清單說明 request 物件 (Records.cf.request) 中的欄位。

clientIp (唯讀)

提出請求之檢視器的 IP 地址。如果檢視器使用 HTTP Proxy 或負載平衡器傳送請求，此值為代理或負載平衡器的 IP 地址。

標頭 (讀取/寫入)

請求中的標頭。注意下列事項：

- 在 headers 物件的金鑰為標準 HTTP 標頭名稱的小寫版本。使用小寫金鑰提供您區分大小寫的標頭值存取權。
- 每個標頭物件 (例如 headers["accept"] 或 headers["host"]) 都是鍵值組的陣列。針對已知的標頭，陣列會在請求中包含每個值的一組鍵值組。
- key 包含在 HTTP 請求中出現的標頭的區分大小寫名稱，例如 Host、User-Agent、X-Forwarded-For 等。
- value 包含在 HTTP 請求中出現的標頭值。
- 當 Lambda 函數新增或修改請求標頭，而您不想要包含標頭 key 欄位時，Lambda@Edge 會使用您提供的標頭名稱來自動插入 key 標頭。無論您如何設定標頭名稱格式，各部分插入的標頭索引鍵名稱都會自動以大寫開頭，以連字號 (-) 分隔。

  例如，您可以在沒有 key 標頭的情況下，新增標頭如下：

```
"user-agent": [
  {
    "value": "ExampleCustomUserAgent/1.X.0"
  }
]
```

  在此範例中，Lambda@Edge 會自動插入 "key": "User-Agent"。

如需標頭使用限制的詳細資訊，請參閱對邊緣函數的限制。

method (唯讀)

請求的 HTTP 方法。

querystring (讀取/寫入)

請求中的查詢字串 (如果有的話)。如果請求不包含查詢字串，事件物件仍會包含具有空白值的 querystring。如需查詢字串的詳細資訊，請參閱根據查詢字串參數快取內容。

uri (讀取/寫入)

請求物件的相對路徑。如果 Lambda 函數修改了 uri 值，請注意下列事項：

- 全新的 uri 值必須以正斜線 (/) 做為開頭。
- 當函數變更 uri 值時，這會變更檢視器請求的物件。

- 當函數變更 `uri` 值時，並不會變更請求的快取行為或請求傳送的目標原始伺服器。

`body` (讀取/寫入)

HTTP 請求的主題。body 結構可包含下列欄位：

`inputTruncated` (唯讀)

布林值旗標，此旗標會指出 Lambda@Edge 是否截斷了本體。如需詳細資訊，請參閱 使用包含內文選項時的要求內文限制。

`action` (讀取/寫入)

您想要對本體採取的動作。`action` 的選項如下：

- `read-only`: 此為預設值。從 Lambda 函式將回應傳回時，如果 `action` 為唯讀狀態，則 Lambda@Edge 會忽略對 `encoding` 或 `data` 所進行的任何變更。
- `replace`: 如果想取代傳送到原始伺服器的本體，請指定此選項。

`encoding` (讀取/寫入)

本體的編碼。當 Lambda@Edge 向 Lambda 函式顯露內文時，會先將內文轉換為 base64-encoding。如果針對 `action` 選擇 `replace` 以取代本體，您可以選擇使用 `base64` (預設) 或 `text` 編碼。如果您指定`encoding`為，`base64`但主體無效base64，則會 CloudFront傳回錯誤。

`data` (讀取/寫入)

請求本體的內容。

`origin` (讀取/寫入) (僅限原始伺服器事件)

傳送請求的目標原始伺服器。`origin` 結構必須確切地包含一個原始伺服器，這可以是自訂原始伺服器或 Amazon S3 原始伺服器。原始伺服器結構可包含下列欄位：

`customHeaders` (讀取/寫入) (自訂和 Amazon S3 原始伺服器)

您可以透過指定的標頭名稱與每個自訂標頭的值對，於請求中包含自訂標頭。您無法新增已列入不允許的標頭，且具有相同名稱的標頭也無法出現在 `Records.cf.request.headers` 中。請求標頭的相關注意事項也適用於自訂標頭。如需詳細資訊，請參閱 無法新增至原始請求的 CloudFront 自訂標頭 及 對邊緣函數的限制。

`domainName` (讀取/寫入) (自訂和 Amazon S3 原始伺服器)

原始伺服器的網域名稱。網域名稱不能空白。

- 適用於自訂原始伺服器：指定 DNS 網域名稱，例如 `www.example.com`。網域名稱不能包含冒號 (:)，而且不能是 IP 地址。網域名稱長度上限為 253 個字元。

- 對於 Amazon S3 原始伺服器 — 指定 Amazon S3 儲存貯體的 DNS 網域名稱，例如 `awsexamplebucket.s3.eu-west-1.amazonaws.com`. 名稱可以高達 128 個字元，而且必須全部小寫。

`path` (讀取/寫入) (自訂和 Amazon S3 原始伺服器)

目錄路徑位於需定位內容請求的原始伺服器中。路徑的開頭應為正斜線 (/)，但結尾不應為正斜線 (例如，結尾不應為 `example-path/`)。僅適用於自訂原始伺服器：路徑應為 URL 編碼，且其長度上限為 255 個字元。

`keepaliveTimeout` (讀取/寫入) (僅限自訂原始伺服器)

在收到回應的最後一個封包之後， CloudFront 應該嘗試維持與原始伺服器的連線時間 (以秒為單位)。此值必須是介於 1–60 (含) 的數字。

`port` (讀取/寫入) (僅限自訂原始伺服器)

CloudFront 應該在自定義原點連接到的端口。此連結埠必須是 80、443，或是介於 1024–65535 之間的數字。

`protocol` (讀取/寫入) (僅限自訂原始伺服器)

連線到原始伺服器時 CloudFront 應使用的連線通訊協定。此值可以為 `http` 或 `https`。

`readTimeout` (讀取/寫入) (僅限自訂原始伺服器)

將請求發送到您的來源後， CloudFront 應等待響應的時間（以秒為單位）。這也會指定在接收下一個封包之前，接收回 CloudFront 應封包之後應等待多久。此值必須是介於 4–60 (含) 的數字。

如果您的使用案例需要 60 秒以上，您可以要求更高的配額`Response timeout per origin`。如需詳細資訊，請參閱 分佈的一般配額。

`sslProtocols` (讀取/寫入) (僅限自訂原始伺服器)

與您的來源建立 HTTPS 連線時， CloudFront 可以使用的最低 SSL/TLS 通訊協定。可為以下任何一個值：`TLSv1.2`、`TLSv1.1`、`TLSv1` 或 `SSLv3`。

`authMethod` (讀取/寫入) (僅限 Amazon S3 原始伺服器)

如果您使用原始存取身分 (OAI)，請將此欄位設定為 `origin-access-identity`。如果您未使用 OAI，請將其設定為 `none`。如果您將 `authMethod` 設定為 `origin-access-identity`，有幾項要求如下：

- 您必須指定 `region` (請參閱下列欄位)。

- 當您將請求從某個 Amazon S3 原始伺服器變更為其他原始伺服器時，您必須使用相同的 OAI。
- 當您將請求從某個自訂原始伺服器變更為 Amazon S3 原始伺服器時，您無法使用 OAI。

> **Note**
>
> 此欄位不支援原始存取控制 (OAC)。

`region` (讀取/寫入) (僅限 Amazon S3 原始伺服器)

您的 Amazon S3 儲存貯體的 AWS 區域。只有在您將 `authMethod` 設定為 `origin-access-identity` 時才需要。

回應事件

下列主題顯示針對檢視器和來源回應事件 CloudFront 傳遞至 Lambda 函數的物件結構。在範例之後，會提供檢視器和原始伺服器回應事件中的所有可能欄位清單。

主題

- 原始伺服器回應範例
- 檢視者回應範例
- 回應事件欄位

原始伺服器回應範例

下列範例顯示原始伺服器回應事件物件。

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "origin-response",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
```

```
          "clientIp": "203.0.113.178",
          "headers": {
            "x-forwarded-for": [
              {
                "key": "X-Forwarded-For",
                "value": "203.0.113.178"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "Amazon CloudFront"
              }
            ],
            "via": [
              {
                "key": "Via",
                "value": "2.0 8f22423015641505b8c857a37450d6c0.cloudfront.net
(CloudFront)"
              }
            ],
            "host": [
              {
                "key": "Host",
                "value": "example.org"
              }
            ],
            "cache-control": [
              {
                "key": "Cache-Control",
                "value": "no-cache"
              }
            ]
          },
          "method": "GET",
          "origin": {
            "custom": {
              "customHeaders": {},
              "domainName": "example.org",
              "keepaliveTimeout": 5,
              "path": "",
              "port": 443,
              "protocol": "https",
              "readTimeout": 30,
```

```
            "sslProtocols": [
              "TLSv1",
              "TLSv1.1",
              "TLSv1.2"
            ]
          }
        },
        "querystring": "",
        "uri": "/"
      },
      "response": {
        "headers": {
          "access-control-allow-credentials": [
            {
              "key": "Access-Control-Allow-Credentials",
              "value": "true"
            }
          ],
          "access-control-allow-origin": [
            {
              "key": "Access-Control-Allow-Origin",
              "value": "*"
            }
          ],
          "date": [
            {
              "key": "Date",
              "value": "Mon, 13 Jan 2020 20:12:38 GMT"
            }
          ],
          "referrer-policy": [
            {
              "key": "Referrer-Policy",
              "value": "no-referrer-when-downgrade"
            }
          ],
          "server": [
            {
              "key": "Server",
              "value": "ExampleCustomOriginServer"
            }
          ],
          "x-content-type-options": [
            {
```

```
                "key": "X-Content-Type-Options",
                "value": "nosniff"
              }
            ],
            "x-frame-options": [
              {
                "key": "X-Frame-Options",
                "value": "DENY"
              }
            ],
            "x-xss-protection": [
              {
                "key": "X-XSS-Protection",
                "value": "1; mode=block"
              }
            ],
            "content-type": [
              {
                "key": "Content-Type",
                "value": "text/html; charset=utf-8"
              }
            ],
            "content-length": [
              {
                "key": "Content-Length",
                "value": "9593"
              }
            ]
          },
          "status": "200",
          "statusDescription": "OK"
        }
      }
    }
  ]
}
```

檢視者回應範例

下列範例顯示檢視器回應事件物件。

```
{
  "Records": [
    {
```

```
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "viewer-response",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
              }
            ],
            "user-agent": [
              {
                "key": "User-Agent",
                "value": "curl/7.66.0"
              }
            ],
            "accept": [
              {
                "key": "accept",
                "value": "*/*"
              }
            ]
          },
          "method": "GET",
          "querystring": "",
          "uri": "/"
        },
        "response": {
          "headers": {
            "access-control-allow-credentials": [
              {
                "key": "Access-Control-Allow-Credentials",
                "value": "true"
              }
            ],
            "access-control-allow-origin": [
              {
                "key": "Access-Control-Allow-Origin",
```

```
            "value": "*"
          }
        ],
        "date": [
          {
            "key": "Date",
            "value": "Mon, 13 Jan 2020 20:14:56 GMT"
          }
        ],
        "referrer-policy": [
          {
            "key": "Referrer-Policy",
            "value": "no-referrer-when-downgrade"
          }
        ],
        "server": [
          {
            "key": "Server",
            "value": "ExampleCustomOriginServer"
          }
        ],
        "x-content-type-options": [
          {
            "key": "X-Content-Type-Options",
            "value": "nosniff"
          }
        ],
        "x-frame-options": [
          {
            "key": "X-Frame-Options",
            "value": "DENY"
          }
        ],
        "x-xss-protection": [
          {
            "key": "X-XSS-Protection",
            "value": "1; mode=block"
          }
        ],
        "age": [
          {
            "key": "Age",
            "value": "2402"
          }
```

```
            ],
            "content-type": [
              {
                "key": "Content-Type",
                "value": "text/html; charset=utf-8"
              }
            ],
            "content-length": [
              {
                "key": "Content-Length",
                "value": "9593"
              }
            ]
          },
          "status": "200",
          "statusDescription": "OK"
        }
      }
    }
  ]
}
```

回應事件欄位

回應事件物件資料包含在三個子物件中：config (Records.cf.config)、request (Records.cf.request) 和 response (Records.cf.response)。如需請求物件中欄位的詳細資訊，請參閱請求物件中的欄位。下列清單說明 config 和 response 子物件中的欄位。

Config 物件中的欄位

下列清單說明 config 物件 (Records.cf.config) 中的欄位。

distributionDomainName (唯讀)

與回應關聯的分佈的網域名稱。

distributionID (唯讀)

與回應關聯的分佈 ID。

eventType (唯讀)

與回應關聯的觸發類型：origin-response 或 viewer-response。

requestId (唯讀)

加密字串，可唯一識別與此回應相關聯的檢視者對CloudFront 要求。該requestId值也會在 CloudFront 存取記錄中顯示為x-edge-request-id。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌) 及 標準日誌檔案欄位。

回應物件中的欄位

下列清單說明 response 物件 (Records.cf.response) 中的欄位。如需如何使用 Lambda@Edge 函式產生 HTTP 請求的資訊，請參閱在要求觸發程序中產生 HTTP 回應。

headers (讀取/寫入)

回應中的標頭。注意下列事項：

- 在 headers 物件的金鑰為標準 HTTP 標頭名稱的小寫版本。使用小寫金鑰提供您區分大小寫的標頭值存取權。
- 每個標頭物件 (例如 headers["content-type"] 或 headers["content-length"]) 都是鍵值組的陣列。針對已知的標頭，陣列會在回應中包含每個值的一組鍵值組。
- key包含標頭出現在 HTTP 回應中時區分大小寫的名稱；例如Content-TypeContent-Length、Cookie、等等。
- value 包含在 HTTP 回應中出現的標頭值。
- 當 Lambda 函數新增或修改回應標頭，而您不想要包含標頭 key 欄位時，Lambda@Edge 會使用您提供的標頭名稱來自動插入 key 標頭。無論您如何設定標頭名稱格式，各部分插入的標頭索引鍵名稱都會自動以大寫開頭，以連字號 (-) 分隔。

  例如，您可以在沒有 key 標頭的情況下，新增標頭如下：

```
"content-type": [
  {
    "value": "text/html;charset=UTF-8"
  }
]
```

  在此範例中，Lambda@Edge 會自動插入 "key": "Content-Type"。

如需標頭使用限制的詳細資訊，請參閱對邊緣函數的限制。

status

回應的 HTTP 狀態碼。

`statusDescription`

回應的 HTTP 狀態說明。

使用請求和回應

本節主題說明 Lambda@Edge 請求和回應的數種使用方式。

主題

- 將 Lambda @Edge 函數搭配原始容錯移轉
- 在要求觸發程序中產生 HTTP 回應
- 更新原始響應觸發器中的 HTTP 響應
- 選擇包含主體選項以存取要求主體

將 Lambda @Edge 函數搭配原始容錯移轉

您可以將 Lambda @Edge 函數與原始群組設定的 CloudFront 發佈搭配使用，例如針對您設定的原始容錯移轉，以協助確保高可用性。若要使用 Lambda 函數與原始伺服器群組搭配，請在您建立快取行為時針對原始伺服器群組觸發的原始伺服器請求或原始伺服器回應中指定函數。

如需詳細資訊，請參閱下列內容：

- 建立原點群組：建立原始伺服器群組
- 原始伺服器容錯移轉如何使用 Lambda@Edge：使用原始伺服器容錯移轉與 Lambda@Edge 函數搭配

在要求觸發程序中產生 HTTP 回應

CloudFront 收到請求時，您可以使用 Lambda 函數產生 HTTP 回應，該 CloudFront 回應會直接傳回給檢視器，而不需將回應轉送至來源。產生 HTTP 回應降低原始伺服器的負載，且通常也可減少檢視器的延遲。

產生 HTTP 回應的一些常見案例包括下列項目：

- 傳回小型網頁給檢視器
- 傳回 HTTP 301 或 302 狀態碼來重新導向使用者到另一個網頁

- 當使用者尚未驗證時，傳回 HTTP 401 狀態碼至檢視器

當發生下列 CloudFront 事件時，Lambda @Edge 函數可以產生 HTTP 回應：

檢視器請求事件

當一個函數由查看器請求事件觸發時， CloudFront 返回響應給查看器，並且不緩存它。

原始伺服器請求事件

當一個函數由來源請求事件觸發時， CloudFront 檢查邊緣緩存是否有先前由該函數生成的響應。

- 如果響應在緩存中，則不執行該函數，並將緩存響應 CloudFront 返回給查看器。
- 如果響應不在緩存中，則執行該函數，將響應 CloudFront 返回給查看器，並緩存它。

若要查看產生 HTTP 回應的範本程式碼，請參閱 Lambda@Edge 範例函數。您也可以在回應觸發中取代 HTTP 回應。如需詳細資訊，請參閱 更新原始響應觸發器中的 HTTP 響應。

程式設計模型

本節說明使用 Lambda@Edge 產生 HTTP 回應的程式設計模型。

主題

- 回應物件
- 錯誤
- 必要欄位

回應物件

以 `result` 方法的 `callback` 參數傳回的回應，需具備下列結構 (請注意，僅 `status` 欄位為必要)。

```
const response = {
    body: 'content',
    bodyEncoding: 'text' | 'base64',
    headers: {
        'header name in lowercase': [{
            key: 'header name in standard case',
            value: 'header value'
         }],
         ...
    },
```

```
    status: 'HTTP status code (string)',
    statusDescription: 'status description'
};
```

回應物件可以包含以下值：

body

您要 CloudFront 在產生的回應中傳回的主體 (如果有的話)。

bodyEncoding

您在 body 中指定的值的編碼。唯一的有效編碼是 `text` 和 `base64`。如果您在`response`物件`body`中包含但省略`bodyEncoding`，則 CloudFront 會將內文視為文字。

如果您指定`bodyEncoding`為，`base64`但主體不是有效的 base64，則 CloudFront 返回一個錯誤。

headers

您想要在產生的 CloudFront 回應中傳回的標頭。注意下列事項：

- 在 `headers` 物件的金鑰為標準 HTTP 標頭名稱的小寫版本。使用小寫金鑰提供您區分大小寫的標頭值存取權。
- 每個標頭 (例如 `headers["accept"]` 或 `headers["host"]`) 是一系列的鍵值組。於已知的標頭，陣列在產生的回應中包含每個值的一組鍵值組。
- `key` (選用) 為在 HTTP 請求中出現的標頭區分大小寫的名稱，例如 `accept` 或 `host`。
- 指定 `value` 為標頭值。
- 如果您未加入索引鍵/值組的標頭索引鍵部分，Lambda@Edge 會自動使用您提供的標頭名稱，來插入標頭索引鍵。無論您如何安排標頭格式，各部分插入的標頭索引鍵名稱都會自動以大寫開頭，以連字號 (-) 分隔。

  例如，您可以不用標頭索引鍵，新增標頭如下：`'content-type': [{ value: 'text/html;charset=UTF-8' }]`

  在這個範例中，Lambda@Edge 建立了下列標頭索引鍵：`Content-Type`。

如需標頭使用限制的詳細資訊，請參閱[對邊緣函數的限制]。

status

HTTP 狀態碼 。以字串形式提供狀態碼。 CloudFront 使用提供的狀態碼進行以下操作：

- 在回應中傳回
- CloudFront 邊緣緩存中的緩存，當響應由源請求事件觸發的函數生成時
- 登入 CloudFront 設定和使用標準日誌 (存取日誌)

如果該`status`值不在 200 和 599 之間，則會將錯誤 CloudFront 傳回給檢視器。

`statusDescription`

要在響應中 CloudFront 返回的描述，以伴隨 HTTP 狀態碼。您不需要使用標準的描述，例如 HTTP 200 狀態碼為 0K。

錯誤

以下是產生的 HTTP 回應的可能錯誤。

回應包含本文與指定 204 (無內容) 狀態

當檢視器要求觸發函數時，當下列兩項成立時，會將 HTTP 502 狀態碼 (錯誤閘道) CloudFront 傳回給檢視器：

- `status` 的值是 204 (無內容)
- 回應包含 `body` 的值

這是因為 Lambda@Edge 強加於 RFC 2616 中選用的限制，也就是 HTTP 204 回應不需要包含訊息本文。

已產生回應的大小限制

由 Lambda 函數產生的回應大小上限取決於觸發函數的事件：

- 檢視器請求事件 – 40 KB
- 原始伺服器請求事件 – 1 MB

如果 CloudFront 回應大於允許的大小，則會將 HTTP 502 狀態碼 (錯誤閘道) 傳回給檢視器。

必要欄位

`status` 欄位是必要的。

所有其他欄位是選用的。

更新原始響應觸發器中的 HTTP 響應

當從原始伺服器 CloudFront 收到 HTTP 回應時，如果存在與快取行為相關聯的來源回應觸發程序，您可以修改 HTTP 回應，以覆寫原始伺服器傳回的內容。

更新 HTTP 回應的一些常用案例包括下列項目：

- 變更狀態以設定 HTTP 200 狀態碼並建立靜態本文內容，以在原始伺服器傳回錯誤狀態碼 (4xx 和 5xx) 時，傳回給檢視器。如需程式碼範例，請參閱 範例：使用來源回應觸發程序將錯誤狀態碼更新為 200。
- 變更狀態來設定 HTTP 301 或 HTTP 302 狀態碼，以在原始伺服器會傳回錯誤狀態碼 (4xx 和 5xx) 時，將使用者重新導向到另一個網站。如需程式碼範例，請參閱 範例：使用來源回應觸發程序將錯誤狀態碼更新為 302。

> **Note**
>
> 該函數必須返回到200和599（含）之間的狀態值，否則 CloudFront 返回一個錯誤給查看器。

您也可以在檢視器與原始伺服器請求事件中取代 HTTP 回應。如需詳細資訊，請參閱 在要求觸發程序中產生 HTTP 回應。

當您使用 HTTP 回應時，Lambda@Edge 不會公開原始伺服器傳回至原始伺服器回應觸發條件的本文。您可以藉由設定為所需的值來產生靜態內容本文，或藉由設定空值來移除函數內的本文。如果您不更新函數的本文欄位，原始伺服器回傳的原始主體會傳回給檢視器。

選擇包含主體選項以存取要求主體

您可以選擇讓 Lambda@Edge 在可寫入的 HTTP 方法 (POST、PUT 和 DELETE 等) 中公開請求的內文，如此您就能在 Lambda 函數中存取該內文。您可以選擇唯讀存取或指定您將替換內文。

若要啟用此選項，請在針對檢視器要求或來源要求事件建立函數的 CloudFront 觸發器時，選擇「包含內文」。如需詳細資訊，請參閱 為 Lambda @Edge 函數新增觸發程序；或者，若要進一步了解使用包含內文與您的函數，請參閱 Lambda@Edge 事件結構說明頁面。

您可能會想使用此功能的情境，包括下列的案例：

- 處理 Web 表單，例如「聯絡我們」表單，而不將客戶輸入的資料傳回給原始伺服器。

- 收集由檢視器的瀏覽器所傳送的網站信標資料，並且在邊緣處理這些資料。

如需程式碼範例，請參閱 Lambda@Edge 範例函數。

> **Note**
>
> 如果請求主體的資料大小過大，Lambda@Edge 會將主體截斷。如需大小上限和截斷的詳細資訊，請參閱使用包含內文選項時的要求內文限制。

Lambda@Edge 範例函數

如需將 Lambda 函數與 Amazon 搭配使用的範例，請參閱下列各節 CloudFront。

主題

- 一般範例
- 產生回應-範例
- 查詢字串-範例
- 根據國家/地區或裝置類型標頭個人化 - 範例
- 以內容為基礎的動態原始伺服器選擇 - 範例
- 更新錯誤狀態-範例
- 訪問請求主體-示例

一般範例

本節中的範例說明在中使用 Lambda @Edge 的一些常見方法CloudFront。

主題

- 範例：A/B 測試
- 範例：覆寫回應標頭

範例：A/B 測試

您可以使用下列範例測試兩種不同版本的影像，而不需要建立重新導向或變更 URL。此範例會讀取檢視器請求中的 Cookie，並據此修改請求 URL。如果檢視器未傳送具有其中一個預期值的 Cookie，此範例會將檢視器隨機指派給其中一個 URL。

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    if (request.uri !== '/experiment-pixel.jpg') {
        // do not process if this is not an A-B test request
        callback(null, request);
        return;
    }

    const cookieExperimentA = 'X-Experiment-Name=A';
    const cookieExperimentB = 'X-Experiment-Name=B';
    const pathExperimentA = '/experiment-group/control-pixel.jpg';
    const pathExperimentB = '/experiment-group/treatment-pixel.jpg';

    /*
     * Lambda at the Edge headers are array objects.
     *
     * Client may send multiple Cookie headers, i.e.:
     * > GET /viewerRes/test HTTP/1.1
     * > User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1
 OpenSSL/1.0.1u zlib/1.2.3
     * > Cookie: First=1; Second=2
     * > Cookie: ClientCode=abc
     * > Host: example.com
     *
     * You can access the first Cookie header at headers["cookie"][0].value
     * and the second at headers["cookie"][1].value.
     *
     * Header values are not parsed. In the example above,
     * headers["cookie"][0].value is equal to "First=1; Second=2"
     */
    let experimentUri;
    if (headers.cookie) {
        for (let i = 0; i < headers.cookie.length; i++) {
            if (headers.cookie[i].value.indexOf(cookieExperimentA) >= 0) {
                console.log('Experiment A cookie found');
                experimentUri = pathExperimentA;
                break;
            } else if (headers.cookie[i].value.indexOf(cookieExperimentB) >= 0) {
```

```
                console.log('Experiment B cookie found');
                experimentUri = pathExperimentB;
                break;
            }
        }
    }

    if (!experimentUri) {
        console.log('Experiment cookie has not been found. Throwing dice...');
        if (Math.random() < 0.75) {
            experimentUri = pathExperimentA;
        } else {
            experimentUri = pathExperimentB;
        }
    }

    request.uri = experimentUri;
    console.log(`Request uri set to "${request.uri}"`);
    callback(null, request);
};
```

Python

```
import json
import random

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    if request['uri'] != '/experiment-pixel.jpg':
        # Not an A/B Test
        return request

    cookieExperimentA, cookieExperimentB = 'X-Experiment-Name=A', 'X-Experiment-
Name=B'
    pathExperimentA, pathExperimentB = '/experiment-group/control-pixel.jpg', '/
experiment-group/treatment-pixel.jpg'

    '''
    Lambda at the Edge headers are array objects.

    Client may send multiple cookie headers. For example:
```

```
    > GET /viewerRes/test HTTP/1.1
    > User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1
OpenSSL/1.0.1u zlib/1.2.3
    > Cookie: First=1; Second=2
    > Cookie: ClientCode=abc
    > Host: example.com

    You can access the first Cookie header at headers["cookie"][0].value
    and the second at headers["cookie"][1].value.

    Header values are not parsed. In the example above,
    headers["cookie"][0].value is equal to "First=1; Second=2"
    '''

    experimentUri = ""

    for cookie in headers.get('cookie', []):
        if cookieExperimentA in cookie['value']:
            print("Experiment A cookie found")
            experimentUri = pathExperimentA
            break
        elif cookieExperimentB in cookie['value']:
            print("Experiment B cookie found")
            experimentUri = pathExperimentB
            break

    if not experimentUri:
        print("Experiment cookie has not been found. Throwing dice...")
        if random.random() < 0.75:
            experimentUri = pathExperimentA
        else:
            experimentUri = pathExperimentB

    request['uri'] = experimentUri
    print(f"Request uri set to {experimentUri}")
    return request
```

範例：覆寫回應標頭

以下範例說明如何根據另一個標頭的值變更回應標頭的值。

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
   const response = event.Records[0].cf.response;
   const headers = response.headers;

   const headerNameSrc = 'X-Amz-Meta-Last-Modified';
   const headerNameDst = 'Last-Modified';

   if (headers[headerNameSrc.toLowerCase()]) {
      headers[headerNameDst.toLowerCase()] = [
         headers[headerNameSrc.toLowerCase()][0],
      ];
      console.log(`Response header "${headerNameDst}" was set to ` +
               `"${headers[headerNameDst.toLowerCase()][0].value}"`);
   }

   callback(null, response);
};
```

Python

```
import json

def lambda_handler(event, context):
    response = event["Records"][0]["cf"]["response"]
    headers = response["headers"]

    headerNameSrc = "X-Amz-Meta-Last-Modified"
    headerNameDst = "Last-Modified"

    if headers.get(headerNameSrc.lower(), None):
        headers[headerNameDst.lower()] = [headers[headerNameSrc.lower()][0]]
        print(f"Response header {headerNameDst.lower()} was set to
 {headers[headerNameSrc.lower()][0]}")

    return response
```

產生回應-範例

本小節中的範例示範如何使用 Lambda@Edge 來產生回應。

主題

- 示例：提供靜態內容（生成的響應）
- 範例：產生 HTTP 重新導向 (產生的回應)

示例：提供靜態內容（生成的響應）

以下範例說明如何使用 Lambda 函數提供靜態網站內容，其可減少原始伺服器的負載，並降低整體延遲。

> **Note**
>
> 您可以針對檢視器要求及原始伺服器請求事件產生 HTTP 回應。如需詳細資訊，請參閱 the section called “在要求觸發程序中產生 HTTP 回應”。
> 您也可以在原始伺服器回應請求事件中取代或移除 HTTP 回應的主體。如需詳細資訊，請參閱 the section called “更新原始響應觸發器中的 HTTP 響應”。

Node.js

```
'use strict';

const content = `
<\!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Simple Lambda@Edge Static Content Response</title>
  </head>
  <body>
    <p>Hello from Lambda@Edge!</p>
  </body>
</html>
`;

exports.handler = (event, context, callback) => {
    /*
```

```
     * Generate HTTP OK response using 200 status code with HTML body.
     */
    const response = {
        status: '200',
        statusDescription: 'OK',
        headers: {
            'cache-control': [{
                key: 'Cache-Control',
                value: 'max-age=100'
            }],
            'content-type': [{
                key: 'Content-Type',
                value: 'text/html'
            }]
        },
        body: content,
    };
    callback(null, response);
};
```

Python

```
import json

CONTENT = """
<\!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Simple Lambda@Edge Static Content Response</title>
</head>
<body>
    <p>Hello from Lambda@Edge!</p>
</body>
</html>
"""

def lambda_handler(event, context):
    # Generate HTTP OK response using 200 status code with HTML body.
    response = {
        'status': '200',
        'statusDescription': 'OK',
        'headers': {
```

```
            'cache-control': [
                {
                    'key': 'Cache-Control',
                    'value': 'max-age=100'
                }
            ],
            "content-type": [
                {
                    'key': 'Content-Type',
                    'value': 'text/html'
                }
            ]
        },
        'body': CONTENT
    }
    return response
```

範例：產生 HTTP 重新導向 (產生的回應)

以下範例說明如何產生 HTTP 重新導向。

> **Note**
>
> 您可以針對檢視器要求及原始伺服器請求事件產生 HTTP 回應。如需詳細資訊，請參閱 在要求觸發程序中產生 HTTP 回應。

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    /*
     * Generate HTTP redirect response with 302 status code and Location header.
     */
    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
```

```
                value: 'https://docs.aws.amazon.com/lambda/latest/dg/lambda-
edge.html',
            }],
        },
    };
    callback(null, response);
};
```

Python

```
def lambda_handler(event, context):

    # Generate HTTP redirect response with 302 status code and Location header.

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': 'https://docs.aws.amazon.com/lambda/latest/dg/lambda-
edge.html'
            }]
        }
    }

    return response
```

查詢字串-範例

本節中的範例包括您可以使用 Lambda@Edge 搭配查詢字串的方式。

主題

- 範例：根據查詢字串參數新增標頭
- 範例：正規化查詢字串參數以改善快取命中率
- 範例：將未驗證的使用者重新導向至登入頁面

範例：根據查詢字串參數新增標頭

以下範例說明如何取得查詢字串參數的鍵值對，然後根據這些值新增標頭。

Node.js

```
'use strict';

const querystring = require('querystring');
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    /* When a request contains a query string key-value pair but the origin server
     * expects the value in a header, you can use this Lambda function to
     * convert the key-value pair to a header. Here's what the function does:
     * 1. Parses the query string and gets the key-value pair.
     * 2. Adds a header to the request using the key-value pair that the function
 got in step 1.
     */

    /* Parse request querystring to get javascript object */
    const params = querystring.parse(request.querystring);

    /* Move auth param from querystring to headers */
    const headerName = 'Auth-Header';
    request.headers[headerName.toLowerCase()] = [{ key: headerName, value:
 params.auth }];
    delete params.auth;

    /* Update request querystring */
    request.querystring = querystring.stringify(params);

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    '''
    When a request contains a query string key-value pair but the origin server
    expects the value in a header, you can use this Lambda function to
    convert the key-value pair to a header. Here's what the function does:
        1. Parses the query string and gets the key-value pair.
```

```
        2. Adds a header to the request using the key-value pair that the function
 got in step 1.
    '''

    # Parse request querystring to get dictionary/json
    params = {k : v[0] for k, v in parse_qs(request['querystring']).items()}

    # Move auth param from querystring to headers
    headerName = 'Auth-Header'
    request['headers'][headerName.lower()] = [{'key': headerName, 'value':
 params['auth']}]
    del params['auth']

    # Update request querystring
    request['querystring'] = urlencode(params)

    return request
```

範例：正規化查詢字串參數以改善快取命中率

下列範例會示範如何在將要求 CloudFront 轉寄至來源之前，對查詢字串進行下列變更，藉此改善快取命中率：

- 依字母排序鍵值組的參數名稱。
- 將鍵值組變更為小寫。

如需詳細資訊，請參閱 根據查詢字串參數快取內容。

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    /* When you configure a distribution to forward query strings to the origin and
     * to cache based on an allowlist of query string parameters, we recommend
     * the following to improve the cache-hit ratio:
     * - Always list parameters in the same order.
     * - Use the same case for parameter names and values.
```

```
     *
     * This function normalizes query strings so that parameter names and values
     * are lowercase and parameter names are in alphabetical order.
     *
     * For more information, see:
     * https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/
QueryStringParameters.html
     */

    console.log('Query String: ', request.querystring);

    /* Parse request query string to get javascript object */
    const params = querystring.parse(request.querystring.toLowerCase());
    const sortedParams = {};

    /* Sort param keys */
    Object.keys(params).sort().forEach(key => {
        sortedParams[key] = params[key];
    });

    /* Update request querystring with normalized  */
    request.querystring = querystring.stringify(sortedParams);

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    '''
    When you configure a distribution to forward query strings to the origin and
    to cache based on an allowlist of query string parameters, we recommend
    the following to improve the cache-hit ratio:
    Always list parameters in the same order.
    - Use the same case for parameter names and values.

    This function normalizes query strings so that parameter names and values
    are lowercase and parameter names are in alphabetical order.

    For more information, see:
```

```
    https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/
QueryStringParameters.html
    '''
    print("Query string: ", request["querystring"])

    # Parse request query string to get js object
    params = {k : v[0] for k, v in parse_qs(request['querystring'].lower()).items()}

    # Sort param keys
    sortedParams = sorted(params.items(), key=lambda x: x[0])

    # Update request querystring with normalized
    request['querystring'] = urlencode(sortedParams)

    return request
```

範例：將未驗證的使用者重新導向至登入頁面

以下範例說明，如果尚未輸入他們的登入資料，如何將使用者重新導向至登入頁面。

Node.js

```
'use strict';

function parseCookies(headers) {
    const parsedCookie = {};
    if (headers.cookie) {
        headers.cookie[0].value.split(';').forEach((cookie) => {
            if (cookie) {
                const parts = cookie.split('=');
                parsedCookie[parts[0].trim()] = parts[1].trim();
            }
        });
    }
    return parsedCookie;
}

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    /* Check for session-id in request cookie in viewer-request event,
     * if session-id is absent, redirect the user to sign in page with original
```

```
     * request sent as redirect_url in query params.
     */

    /* Check for session-id in cookie, if present then proceed with request */
    const parsedCookies = parseCookies(headers);
    if (parsedCookies && parsedCookies['session-id']) {
        callback(null, request);
        return;
    }

    /* URI encode the original request to be sent as redirect_url in query params */
    const encodedRedirectUrl = encodeURIComponent(`https://
${headers.host[0].value}${request.uri}?${request.querystring}`);
    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
                value: `https://www.example.com/signin?redirect_url=
${encodedRedirectUrl}`,
            }],
        },
    };
    callback(null, response);
};
```

Python

```
import urllib

def parseCookies(headers):
    parsedCookie = {}
    if headers.get('cookie'):
        for cookie in headers['cookie'][0]['value'].split(';'):
            if cookie:
                parts = cookie.split('=')
                parsedCookie[parts[0].strip()] = parts[1].strip()
    return parsedCookie

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']
```

```
    '''
    Check for session-id in request cookie in viewer-request event,
    if session-id is absent, redirect the user to sign in page with original
    request sent as redirect_url in query params.
    '''

    # Check for session-id in cookie, if present, then proceed with request
    parsedCookies = parseCookies(headers)

    if parsedCookies and parsedCookies['session-id']:
        return request

    # URI encode the original request to be sent as redirect_url in query params
    redirectUrl = "https://%s%s?%s" % (headers['host'][0]['value'], request['uri'],
request['querystring'])
    encodedRedirectUrl = urllib.parse.quote_plus(redirectUrl.encode('utf-8'))

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': 'https://www.example.com/signin?redirect_url=%s' %
encodedRedirectUrl
            }]
        }
    }
    return response
```

根據國家/地區或裝置類型標頭個人化 - 範例

本節中的範例說明如何使用 Lambda@Edge 來根據位置或根據檢視器使用的裝置類型自訂行為。

主題

- 範例：將檢視者要求重新導向至特定國家/地區的 URL
- 範例：根據裝置提供不同版本的物件

範例：將檢視者要求重新導向至特定國家/地區的 URL

以下範例說明如何產生含國家/地區特定 URL 的 HTTP 重新導向回應，並將回應傳回至檢視器。當您希望提供特定國家的回應時，此方法很有用。例如：

- 如果您有國家/地區特定的子網域 (例如 us.example.com 和 tw.example.com)，當檢視器請求 example.com 時，您可以產生一個重新導向回應。
- 如果您在串流視訊，但沒有在特定國家/地區串流此內容的權利，您可以在該國家/地區將使用者重新導向到說明他們為何無法檢視影片的頁面。

注意下列事項：

- 您必須根據 `CloudFront-Viewer-Country` 標頭設定您的分佈為快取。如需詳細資訊，請參閱 [根據選取請求標頭的快取]。
- CloudFront 在檢視器要求事件之後加入`CloudFront-Viewer-Country`標頭。要使用此範例，您必須建立原始伺服器請求事件的觸發。

Node.js

```
'use strict';

/* This is an origin request function */
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    /*
     * Based on the value of the CloudFront-Viewer-Country header, generate an
     * HTTP status code 302 (Redirect) response, and return a country-specific
     * URL in the Location header.
     * NOTE: 1. You must configure your distribution to cache based on the
     *          CloudFront-Viewer-Country header. For more information, see
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
     *       2. CloudFront adds the CloudFront-Viewer-Country header after the
 viewer
     *          request event. To use this example, you must create a trigger for
 the
     *          origin request event.
     */
```

```
    let url = 'https://example.com/';
    if (headers['cloudfront-viewer-country']) {
        const countryCode = headers['cloudfront-viewer-country'][0].value;
        if (countryCode === 'TW') {
            url = 'https://tw.example.com/';
        } else if (countryCode === 'US') {
            url = 'https://us.example.com/';
        }
    }

    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
                value: url,
            }],
        },
    };
    callback(null, response);
};
```

Python

```
# This is an origin request function

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    '''
    Based on the value of the CloudFront-Viewer-Country header, generate an
    HTTP status code 302 (Redirect) response, and return a country-specific
    URL in the Location header.
    NOTE: 1. You must configure your distribution to cache based on the
            CloudFront-Viewer-Country header. For more information, see
            https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
          2. CloudFront adds the CloudFront-Viewer-Country header after the viewer
            request event. To use this example, you must create a trigger for the
            origin request event.
    '''
```

```
    url = 'https://example.com/'
    viewerCountry = headers.get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        if countryCode == 'TW':
            url = 'https://tw.example.com/'
        elif countryCode == 'US':
            url = 'https://us.example.com/'

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': url
            }]
        }
    }

    return response
```

範例：根據裝置提供不同版本的物件

以下範例說明如何根據使用者使用的裝置類型，提供不同的物件版本，例如行動裝置或平板電腦。注意下列事項：

- 您必須根據 CloudFront-Is-*-Viewer 標頭設定您的分佈為快取。如需詳細資訊，請參閱 根據選取請求標頭的快取。
- CloudFront 在檢視器要求事件之後加入CloudFront-Is-*-Viewer標頭。要使用此範例，您必須建立原始伺服器請求事件的觸發。

Node.js

```
'use strict';

/* This is an origin request function */
exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;
```

```
    /*
     * Serve different versions of an object based on the device type.
     * NOTE: 1. You must configure your distribution to cache based on the
     *          CloudFront-Is-*-Viewer headers. For more information, see
     *          the following documentation:
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-device-type
     *       2. CloudFront adds the CloudFront-Is-*-Viewer headers after the viewer
     *          request event. To use this example, you must create a trigger for
 the
     *          origin request event.
     */

    const desktopPath = '/desktop';
    const mobilePath = '/mobile';
    const tabletPath = '/tablet';
    const smarttvPath = '/smarttv';

    if (headers['cloudfront-is-desktop-viewer']
        && headers['cloudfront-is-desktop-viewer'][0].value === 'true') {
        request.uri = desktopPath + request.uri;
    } else if (headers['cloudfront-is-mobile-viewer']
               && headers['cloudfront-is-mobile-viewer'][0].value === 'true') {
        request.uri = mobilePath + request.uri;
    } else if (headers['cloudfront-is-tablet-viewer']
               && headers['cloudfront-is-tablet-viewer'][0].value === 'true') {
        request.uri = tabletPath + request.uri;
    } else if (headers['cloudfront-is-smarttv-viewer']
               && headers['cloudfront-is-smarttv-viewer'][0].value === 'true') {
        request.uri = smarttvPath + request.uri;
    }
    console.log(`Request uri set to "${request.uri}"`);

    callback(null, request);
};
```

Python

```
# This is an origin request function
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
```

```
    headers = request['headers']

    '''
    Serve different versions of an object based on the device type.
    NOTE: 1. You must configure your distribution to cache based on the
            CloudFront-Is-*-Viewer headers. For more information, see
            the following documentation:
            https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
            https://docs.aws.amazon.com/console/cloudfront/cache-on-device-type
          2. CloudFront adds the CloudFront-Is-*-Viewer headers after the viewer
            request event. To use this example, you must create a trigger for the
            origin request event.
    '''

    desktopPath = '/desktop';
    mobilePath = '/mobile';
    tabletPath = '/tablet';
    smarttvPath = '/smarttv';

    if 'cloudfront-is-desktop-viewer' in headers and headers['cloudfront-is-desktop-
viewer'][0]['value'] == 'true':
        request['uri'] = desktopPath + request['uri']
    elif 'cloudfront-is-mobile-viewer' in headers and headers['cloudfront-is-mobile-
viewer'][0]['value'] == 'true':
        request['uri'] = mobilePath + request['uri']
    elif 'cloudfront-is-tablet-viewer' in headers and headers['cloudfront-is-tablet-
viewer'][0]['value'] == 'true':
        request['uri'] = tabletPath + request['uri']
    elif 'cloudfront-is-smarttv-viewer' in headers and headers['cloudfront-is-
smarttv-viewer'][0]['value'] == 'true':
        request['uri'] = smarttvPath + request['uri']

    print("Request uri set to %s" % request['uri'])

    return request
```

以內容為基礎的動態原始伺服器選擇 - 範例

本節中的範例示範如何使用 Lambda@Edge 來根據請求中的資訊路由到不同的原始伺服器。

主題

- 範例：使用原始請求觸發程序從自訂來源變更為 Amazon S3 來源

- 範例：使用來源請求觸發程序變更 Amazon S3 來源區域
- 範例：使用原始請求觸發程序從 Amazon S3 來源變更為自訂來源
- 範例：使用來源請求觸發程序將流量從一個 Amazon S3 儲存貯體逐步傳輸到另一個儲存貯體
- 範例：使用來源要求觸發程式，根據國家/地區標頭變更原始網域名稱

範例：使用原始請求觸發程序從自訂來源變更為 Amazon S3 來源

此函數示範如何使用 origin-request 觸發條件，根據請求屬性從自訂原始伺服器變更至內容被擷取的 Amazon S3 原始伺服器。

Node.js

```
'use strict';

 const querystring = require('querystring');

 exports.handler = (event, context, callback) => {
     const request = event.Records[0].cf.request;

     /**
      * Reads query string to check if S3 origin should be used, and
      * if true, sets S3 origin properties.
      */

     const params = querystring.parse(request.querystring);

     if (params['useS3Origin']) {
         if (params['useS3Origin'] === 'true') {
             const s3DomainName = 'my-bucket.s3.amazonaws.com';

             /* Set S3 origin fields */
             request.origin = {
                 s3: {
                     domainName: s3DomainName,
                     region: '',
                     authMethod: 'none',
                     path: '',
                     customHeaders: {}
                 }
             };
             request.headers['host'] = [{ key: 'host', value: s3DomainName}];
```

```
        }
    }

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    '''
    Reads query string to check if S3 origin should be used, and
    if true, sets S3 origin properties
    '''
    params = {k: v[0] for k, v in parse_qs(request['querystring']).items()}
    if params.get('useS3Origin') == 'true':
        s3DomainName = 'my-bucket.s3.amazonaws.com'

        # Set S3 origin fields
        request['origin'] = {
            's3': {
                'domainName': s3DomainName,
                'region': '',
                'authMethod': 'none',
                'path': '',
                'customHeaders': {}
            }
        }
        request['headers']['host'] = [{'key': 'host', 'value': s3DomainName}]
    return request
```

範例：使用來源請求觸發程序變更 Amazon S3 來源區域

此函數示範如何根據請求屬性，使用 origin-request 觸發條件來變更內容被擷取的 Amazon S3 原始伺服器。

在這個範例中，我們使用 CloudFront-Viewer-Country 標頭的值來將 S3 儲存貯體的網域名稱更新為較靠近檢視器的區域內的儲存貯體。這在數種方式中非常受用：

- 當指定的區域更靠近檢視器的國家/地區時，能減少延遲。

- 藉由確保該資料與請求來自位於相同國家/地區的原始伺服器，提供資料主權服務。

要使用此範例，須執行下列項目：

- 您必須根據 CloudFront-Viewer-Country 標頭設定您的分佈為快取。如需詳細資訊，請參閱 根據選取請求標頭的快取。
- 在原始請求事件中為此函數創建一個觸發器。 CloudFront在查看器請求事件之後添加CloudFront-Viewer-Country標題，因此要使用此示例，您必須確保該函數針對原始請求執行。

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    /**
     * This blueprint demonstrates how an origin-request trigger can be used to
     * change the origin from which the content is fetched, based on request
 properties.
     * In this example, we use the value of the CloudFront-Viewer-Country header
     * to update the S3 bucket domain name to a bucket in a Region that is closer to
     * the viewer.
     *
     * This can be useful in several ways:
     *      1) Reduces latencies when the Region specified is nearer to the viewer's
     *         country.
     *      2) Provides data sovereignty by making sure that data is served from an
     *         origin that's in the same country that the request came from.
     *
     * NOTE: 1. You must configure your distribution to cache based on the
     *          CloudFront-Viewer-Country header. For more information, see
     *          https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
     *       2. CloudFront adds the CloudFront-Viewer-Country header after the
 viewer
     *          request event. To use this example, you must create a trigger for
 the
     *          origin request event.
     */
```

```
    const countryToRegion = {
        'DE': 'eu-central-1',
        'IE': 'eu-west-1',
        'GB': 'eu-west-2',
        'FR': 'eu-west-3',
        'JP': 'ap-northeast-1',
        'IN': 'ap-south-1'
    };

    if (request.headers['cloudfront-viewer-country']) {
        const countryCode = request.headers['cloudfront-viewer-country'][0].value;
        const region = countryToRegion[countryCode];

        /**
         * If the viewer's country is not in the list you specify, the request
         * goes to the default S3 bucket you've configured.
         */
        if (region) {
            /**
             * If you've set up OAI, the bucket policy in the destination bucket
             * should allow the OAI GetObject operation, as configured by default
             * for an S3 origin with OAI. Another requirement with OAI is to provide
             * the Region so it can be used for the SIGV4 signature. Otherwise, the
             * Region is not required.
             */
            request.origin.s3.region = region;
            const domainName = `my-bucket-in-${region}.s3.amazonaws.com`;
            request.origin.s3.domainName = domainName;
            request.headers['host'] = [{ key: 'host', value: domainName }];
        }
    }

    callback(null, request);
};
```

Python

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    '''
    This blueprint demonstrates how an origin-request trigger can be used to
```

```
    change the origin from which the content is fetched, based on request
properties.
    In this example, we use the value of the CloudFront-Viewer-Country header
    to update the S3 bucket domain name to a bucket in a Region that is closer to
    the viewer.

    This can be useful in several ways:
        1) Reduces latencies when the Region specified is nearer to the viewer's
            country.
        2) Provides data sovereignty by making sure that data is served from an
            origin that's in the same country that the request came from.

    NOTE: 1. You must configure your distribution to cache based on the
            CloudFront-Viewer-Country header. For more information, see
            https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
          2. CloudFront adds the CloudFront-Viewer-Country header after the viewer
            request event. To use this example, you must create a trigger for the
            origin request event.
    '''

    countryToRegion = {
        'DE': 'eu-central-1',
        'IE': 'eu-west-1',
        'GB': 'eu-west-2',
        'FR': 'eu-west-3',
        'JP': 'ap-northeast-1',
        'IN': 'ap-south-1'
    }

    viewerCountry = request['headers'].get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        region = countryToRegion.get(countryCode)

        # If the viewer's country in not in the list you specify, the request
        # goes to the default S3 bucket you've configured
        if region:
            '''
            If you've set up OAI, the bucket policy in the destination bucket
            should allow the OAI GetObject operation, as configured by default
            for an S3 origin with OAI. Another requirement with OAI is to provide
            the Region so it can be used for the SIGV4 signature. Otherwise, the
            Region is not required.
            '''
```

```
            request['origin']['s3']['region'] = region
            domainName = 'my-bucket-in-%s.s3.amazonaws.com' % region
            request['origin']['s3']['domainName'] = domainName
            request['headers']['host'] = [{'key': 'host', 'value': domainName}]

    return request
```

範例：使用原始請求觸發程序從 Amazon S3 來源變更為自訂來源

此函數示範如何根據請求屬性，使用原始伺服器請求觸發來變更內容被擷取的自訂原始伺服器。

Node.js

```
'use strict';

const querystring = require('querystring');

 exports.handler = (event, context, callback) => {
     const request = event.Records[0].cf.request;

     /**
      * Reads query string to check if custom origin should be used, and
      * if true, sets custom origin properties.
      */

     const params = querystring.parse(request.querystring);

     if (params['useCustomOrigin']) {
         if (params['useCustomOrigin'] === 'true') {

             /* Set custom origin fields*/
             request.origin = {
                 custom: {
                     domainName: 'www.example.com',
                     port: 443,
                     protocol: 'https',
                     path: '',
                     sslProtocols: ['TLSv1', 'TLSv1.1'],
                     readTimeout: 5,
                     keepaliveTimeout: 5,
                     customHeaders: {}
                 }
             };
```

```
            request.headers['host'] = [{ key: 'host', value: 'www.example.com'}];
        }
     }
    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    # Reads query string to check if custom origin should be used, and
    # if true, sets custom origin properties

    params = {k: v[0] for k, v in parse_qs(request['querystring']).items()}

    if params.get('useCustomOrigin') == 'true':
            # Set custom origin fields
            request['origin'] = {
                'custom': {
                    'domainName': 'www.example.com',
                    'port': 443,
                    'protocol': 'https',
                    'path': '',
                    'sslProtocols': ['TLSv1', 'TLSv1.1'],
                    'readTimeout': 5,
                    'keepaliveTimeout': 5,
                    'customHeaders': {}
                }
            }
            request['headers']['host'] = [{'key': 'host', 'value':
 'www.example.com'}]

    return request
```

範例：使用來源請求觸發程序將流量從一個 Amazon S3 儲存貯體逐步傳輸到另一個儲存貯體

此功能示範如何以受控方式逐步將流量從一個 Amazon S3 儲存貯體傳輸到另一個儲存貯體。

Node.js

```
'use strict';

    function getRandomInt(min, max) {
        /* Random number is inclusive of min and max*/
        return Math.floor(Math.random() * (max - min + 1)) + min;
 }

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const BLUE_TRAFFIC_PERCENTAGE = 80;

    /**
      * This Lambda function demonstrates how to gradually transfer traffic from
      * one S3 bucket to another in a controlled way.
      * We define a variable BLUE_TRAFFIC_PERCENTAGE which can take values from
      * 1 to 100. If the generated randomNumber less than or equal to
 BLUE_TRAFFIC_PERCENTAGE, traffic
      * is re-directed to blue-bucket. If not, the default bucket that we've
 configured
      * is used.
      */

    const randomNumber = getRandomInt(1, 100);

if (randomNumber <= BLUE_TRAFFIC_PERCENTAGE) {
         const domainName = 'blue-bucket.s3.amazonaws.com';
         request.origin.s3.domainName = domainName;
         request.headers['host'] = [{ key: 'host', value: domainName}];
     }
    callback(null, request);
};
```

Python

```
import math
import random

def getRandomInt(min, max):
    # Random number is inclusive of min and max
    return math.floor(random.random() * (max - min + 1)) + min
```

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    BLUE_TRAFFIC_PERCENTAGE = 80

    '''
    This Lambda function demonstrates how to gradually transfer traffic from
    one S3 bucket to another in a controlled way.
    We define a variable BLUE_TRAFFIC_PERCENTAGE which can take values from
    1 to 100. If the generated randomNumber less than or equal to
 BLUE_TRAFFIC_PERCENTAGE, traffic
    is re-directed to blue-bucket. If not, the default bucket that we've configured
    is used.
    '''

    randomNumber = getRandomInt(1, 100)

    if randomNumber <= BLUE_TRAFFIC_PERCENTAGE:
        domainName = 'blue-bucket.s3.amazonaws.com'
        request['origin']['s3']['domainName'] = domainName
        request['headers']['host'] = [{'key': 'host', 'value': domainName}]

    return request
```

範例：使用來源要求觸發程式，根據國家/地區標頭變更原始網域名稱

此函數示範如何根據 CloudFront-Viewer-Country 標頭來變更原始伺服器的網域名稱，使內容能從靠近檢視者國家/地區的原始伺服器提供。

為您的分佈實施此功能可以擁有如下所述的優勢：

- 當指定的區域更靠近檢視器的國家/地區時，可減少延遲
- 藉由確保該資料與請求來自位於相同國家/地區的原始伺服器來提供資料主權服務

請注意，要啟用此功能，您必須根據 CloudFront-Viewer-Country 標頭設定您的分佈為快取。如需詳細資訊，請參閱 the section called “根據選取請求標頭的快取”。

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
```

```
    const request = event.Records[0].cf.request;

  if (request.headers['cloudfront-viewer-country']) {
        const countryCode = request.headers['cloudfront-viewer-country'][0].value;
        if (countryCode === 'GB' || countryCode === 'DE' || countryCode === 'IE' )
 {
            const domainName = 'eu.example.com';
            request.origin.custom.domainName = domainName;
            request.headers['host'] = [{key: 'host', value: domainName}];
        }
    }

    callback(null, request);
};
```

Python

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    viewerCountry = request['headers'].get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        if countryCode == 'GB' or countryCode == 'DE' or countryCode == 'IE':
            domainName = 'eu.example.com'
            request['origin']['custom']['domainName'] = domainName
            request['headers']['host'] = [{'key': 'host', 'value': domainName}]
    return request
```

更新錯誤狀態-範例

本節中的範例提供如何使用 Lambda@Edge 來變更傳回給使用者的錯誤狀態的指導。

主題

- 範例：使用來源回應觸發程序將錯誤狀態碼更新為 200
- 範例：使用來源回應觸發程序將錯誤狀態碼更新為 302

範例：使用來源回應觸發程序將錯誤狀態碼更新為 200

此函數示範如何更新回應狀態為 200 並產生靜態本文內容，以在以下案例傳回給檢視器：

- 函數在原始伺服器回應中觸發。
- 原始伺服器的回應狀態為錯誤狀態碼 (4xx 和 5xx)

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const response = event.Records[0].cf.response;

    /**
     * This function updates the response status to 200 and generates static
     * body content to return to the viewer in the following scenario:
     * 1. The function is triggered in an origin response
     * 2. The response status from the origin server is an error status code (4xx or
 5xx)
     */

    if (response.status >= 400 && response.status <= 599) {
        response.status = 200;
        response.statusDescription = 'OK';
        response.body = 'Body generation example';
    }

    callback(null, response);
};
```

Python

```
def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']

    '''
    This function updates the response status to 200 and generates static
    body content to return to the viewer in the following scenario:
    1. The function is triggered in an origin response
    2. The response status from the origin server is an error status code (4xx or
 5xx)
    '''

    if int(response['status']) >= 400 and int(response['status']) <= 599:
        response['status'] = 200
```

```
        response['statusDescription'] = 'OK'
        response['body'] = 'Body generation example'
    return response
```

範例：使用來源回應觸發程序將錯誤狀態碼更新為 302

此函數示範如何更新 HTTP 狀態碼為 302，以重新導向到另一個由不同原始伺服器設定的路徑 (快取行為)。注意下列事項：

- 函數在原始伺服器回應中觸發。
- 原始伺服器的回應狀態為錯誤狀態碼 (4xx 和 5xx)

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const response = event.Records[0].cf.response;
    const request = event.Records[0].cf.request;

    /**
     * This function updates the HTTP status code in the response to 302, to
 redirect to another
     * path (cache behavior) that has a different origin configured. Note the
 following:
     * 1. The function is triggered in an origin response
     * 2. The response status from the origin server is an error status code (4xx or
 5xx)
     */

    if (response.status >= 400 && response.status <= 599) {
        const redirect_path = `/plan-b/path?${request.querystring}`;

        response.status = 302;
        response.statusDescription = 'Found';

        /* Drop the body, as it is not required for redirects */
        response.body = '';
        response.headers['location'] = [{ key: 'Location', value: redirect_path }];
    }
```

```
    callback(null, response);
};
```

Python

```
def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']
    request = event['Records'][0]['cf']['request']

    '''
    This function updates the HTTP status code in the response to 302, to redirect
 to another
    path (cache behavior) that has a different origin configured. Note the
 following:
    1. The function is triggered in an origin response
    2. The response status from the origin server is an error status code (4xx or
 5xx)
    '''

    if int(response['status']) >= 400 and int(response['status']) <= 599:
        redirect_path = '/plan-b/path?%s' % request['querystring']

        response['status'] = 302
        response['statusDescription'] = 'Found'

        # Drop the body as it is not required for redirects
        response['body'] = ''
        response['headers']['location'] = [{'key': 'Location', 'value':
 redirect_path}]

    return response
```

訪問請求主體-示例

本節中的範例說明如何使用 Lambda@Edge 來搭配 POST 請求。

> **Note**
>
> 若要使用這些範例，您必須在分佈的 Lambda 函數關聯中啟用包含內文選項。依預設不會啟用此功能。

- 若要在 CloudFront 主控台中啟用此設定，請選取「在 Lambda 函數關聯中包含主體」的核取方塊。
- 若要在 CloudFront API 或使用中啟用此設定 AWS CloudFormation，請將`IncludeBody`欄位設定為`true`中`LambdaFunctionAssociation`。

主題

範例：使用要求觸發程式來讀取 HTML 表單

此函數示範如何處理 HTML 表單 (Web 表單) 所產生 POST 請求的主體，例如「聯絡我們」表單。例如，您可能會有如下的 HTML 表單：

```
<html>
  <form action="https://example.com" method="post">
    Param 1: <input type="text" name="name1"><br>
    Param 2: <input type="text" name="name2"><br>
    input type="submit" value="Submit">
  </form>
</html>
```

對於下面的示例函數，該函數必須在CloudFront 查看器請求或源請求中觸發。

Node.js

```
'use strict';

const querystring = require('querystring');

/**
 * This function demonstrates how you can read the body of a POST request
 * generated by an HTML form (web form). The function is triggered in a
 * CloudFront viewer request or origin request event type.
 */

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
```

```
    if (request.method === 'POST') {
        /* HTTP body is always passed as base64-encoded string. Decode it. */
        const body = Buffer.from(request.body.data, 'base64').toString();

        /* HTML forms send the data in query string format. Parse it. */
        const params = querystring.parse(body);

        /* For demonstration purposes, we only log the form fields here.
         * You can put your custom logic here. For example, you can store the
         * fields in a database, such as Amazon DynamoDB, and generate a response
         * right from your Lambda@Edge function.
         */
        for (let param in params) {
            console.log(`For "${param}" user submitted "${params[param]}".\n`);
        }
    }
    return callback(null, request);
};
```

Python

```
import base64
from urllib.parse import parse_qs

'''
Say there is a POST request body generated by an HTML such as:

<html>
<form action="https://example.com" method="post">
    Param 1: <input type="text" name="name1"><br>
    Param 2: <input type="text" name="name2"><br>
    input type="submit" value="Submit">
</form>
</html>

'''

'''
This function demonstrates how you can read the body of a POST request
generated by an HTML form (web form). The function is triggered in a
CloudFront viewer request or origin request event type.
'''
```

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    if request['method'] == 'POST':
        # HTTP body is always passed as base64-encoded string. Decode it
        body = base64.b64decode(request['body']['data'])

        # HTML forms send the data in query string format. Parse it
        params = {k: v[0] for k, v in parse_qs(body).items()}

        '''
        For demonstration purposes, we only log the form fields here.
        You can put your custom logic here. For example, you can store the
        fields in a database, such as Amazon DynamoDB, and generate a response
        right from your Lambda@Edge function.
        '''
        for key, value in params.items():
            print("For %s use submitted %s" % (key, value))

    return request
```

範例：使用要求觸發程式來修改 HTML 表單

此函數示範如何修改 HTML 表單 (Web 表單) 所產生 POST 請求的主體。該功能在 CloudFront 查看器請求或源請求中觸發。

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
    var request = event.Records[0].cf.request;
    if (request.method === 'POST') {
        /* Request body is being replaced. To do this, update the following
        /* three fields:
         *    1) body.action to 'replace'
         *    2) body.encoding to the encoding of the new data.
         *
         *       Set to one of the following values:
```

```
         *
         *           text - denotes that the generated body is in text format.
         *               Lambda@Edge will propagate this as is.
         *           base64 - denotes that the generated body is base64 encoded.
         *               Lambda@Edge will base64 decode the data before sending
         *               it to the origin.
         *    3) body.data to the new body.
         */
        request.body.action = 'replace';
        request.body.encoding = 'text';
        request.body.data = getUpdatedBody(request);
    }
    callback(null, request);
};

function getUpdatedBody(request) {
    /* HTTP body is always passed as base64-encoded string. Decode it. */
    const body = Buffer.from(request.body.data, 'base64').toString();

    /* HTML forms send data in query string format. Parse it. */
    const params = querystring.parse(body);

    /* For demonstration purposes, we're adding one more param.
     *
     * You can put your custom logic here. For example, you can truncate long
     * bodies from malicious requests.
     */
    params['new-param-name'] = 'new-param-value';
    return querystring.stringify(params);
}
```

Python

```
import base64
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    if request['method'] == 'POST':
        '''
        Request body is being replaced. To do this, update the following
        three fields:
            1) body.action to 'replace'
```

```
            2) body.encoding to the encoding of the new data.

            Set to one of the following values:

                text - denotes that the generated body is in text format.
                    Lambda@Edge will propagate this as is.
                base64 - denotes that the generated body is base64 encoded.
                    Lambda@Edge will base64 decode the data before sending
                    it to the origin.
            3) body.data to the new body.
        '''
        request['body']['action'] = 'replace'
        request['body']['encoding'] = 'text'
        request['body']['data'] = getUpdatedBody(request)
    return request

def getUpdatedBody(request):
    # HTTP body is always passed as base64-encoded string. Decode it
    body = base64.b64decode(request['body']['data'])

    # HTML forms send data in query string format. Parse it
    params = {k: v[0] for k, v in parse_qs(body).items()}

    # For demonstration purposes, we're adding one more param

    # You can put your custom logic here. For example, you can truncate long
    # bodies from malicious requests
    params['new-param-name'] = 'new-param-value'
    return urlencode(params)
```

對邊緣函數的限制

下列主題說明套用至 CloudFront 函數和 Lambda @Edge 的限制。某些限制適用於所有邊緣函數，而其他限制則僅適用於 CloudFront 函數或 Lambda @Edge。

如需有關配額 (先前稱為限制) 的詳細資訊，請參閱 CloudFront 功能配額 和 Lambda@Edge 的配額。

主題

- 對所有邊緣函數的限制
- CloudFront 功能限制
- 對 Lambda@Edge 的限制

對所有邊緣函數的限制

下列限制適用於所有邊緣函數，包括 CloudFront 函數和 Lambda @Edge。

主題

AWS 帳戶 擁有權

要將邊函數與 CloudFront 分佈相關聯，該函數和分佈必須由相同的擁有 AWS 帳戶。

將 CloudFront 函數與 @Edge Lambda 結合

下列限制適用於指定的快取行為：

- 每個事件類型 (檢視器請求、原始伺服器請求、原始伺服器回應和檢視器回應) 只能有一個邊緣函數關聯。
- 您無法在檢視器事件 (檢視器要求和檢視器回應) 中合併 CloudFront 函數和 Lambda @Edge。

允許邊緣函數的所有其他組合。下表說明了允許的組合。

		CloudFront 函数	
		檢視者請求	檢視者回應
Lambda@Edge	檢視者請求	不允許	不允許

	原始伺服器請求	已允許	已允許
	原始伺服器回應	已允許	已允許
	檢視者回應	不允許	不允許

HTTP 狀態碼

CloudFront 當來源傳回 HTTP 狀態碼 400 或更高版本時，不會針對檢視器回應事件叫用邊緣函數。

CloudFront 會針對所有原始伺服器回應，呼叫原始伺服器回應事件的 Lambda@Edge 函數，包括在原始伺服器傳回 HTTP 狀態碼 400 (或更高值) 時。如需詳細資訊，請參閱 更新原始響應觸發器中的 HTTP 響應。

HTTP 標頭

某些 HTTP 標頭不允許使用，這意味著這些標頭不會公開給邊緣函數，且函數無法新增這些標頭。其他標頭則設為唯讀，代表函數可以讀取這些標頭，但無法新增或修改。

主題

- 不允許的標頭
- 唯讀標頭

不允許的標頭

下列 HTTP 標頭不會公開給邊緣函數，且函數無法新增這些標頭。如果您的函數添加了其中一個標頭，則 CloudFront驗證失敗，並將 HTTP 狀態碼 502（錯誤網關） CloudFront 返回給查看器。

- `Connection`
- `Expect`
- `Keep-Alive`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `Proxy-Connection`
- `Trailer`
- `Upgrade`

- X-Accel-Buffering
- X-Accel-Charset
- X-Accel-Limit-Rate
- X-Accel-Redirect
- X-Amz-Cf-*
- X-Amzn-Auth
- X-Amzn-Cf-Billing
- X-Amzn-Cf-Id
- X-Amzn-Cf-Xff
- X-Amzn-Errortype
- X-Amzn-Fle-Profile
- X-Amzn-Header-Count
- X-Amzn-Header-Order
- X-Amzn-Lambda-Integration-Tag
- X-Amzn-RequestId
- X-Cache
- X-Edge-*
- X-Forwarded-Proto
- X-Real-IP

唯讀標頭

下列標頭為唯讀的狀態。您的函數可以讀取這些標頭並將其作為函數邏輯的輸入，但無法變更其值。如果您的函數新增或編輯唯讀標頭，則要求 CloudFront 驗證失敗，並將 HTTP 狀態碼 502 (錯誤閘道) CloudFront 傳回給檢視器。

檢視器請求事件中的唯讀標頭

下列標頭在檢視器請求事件中為唯讀的狀態。

- Content-Length
- Host

- Transfer-Encoding
- Via

原始伺服器請求事件中的唯讀標頭 (僅限 Lambda@Edge)

下列標頭在原始伺服器請求事件中為唯讀的狀態，僅存在於 Lambda@Edge 中。

- Accept-Encoding
- Content-Length
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Transfer-Encoding
- Via

原始伺服器回應事件中的唯讀標頭 (僅限 Lambda@Edge)

下列標頭在原始伺服器回應事件中為唯讀的狀態，僅存在於 Lambda@Edge 中。

- Transfer-Encoding
- Via

檢視器回應事件中的唯讀標頭

下列標頭在 CloudFront 函數和 Lambda @Edge 的檢視器回應事件中都是唯讀的。

- Warning
- Via

下列標頭在 Lambda@Edge 檢視器回應事件中為唯讀狀態。

- Content-Length
- Content-Encoding

- `Transfer-Encoding`

查詢字串

下列限制適用於讀取、更新或在請求 URI 中建立查詢字串的函數。

- (僅限 Lambda@Edge) 若要存取原始伺服器請求或原始伺服器回應函數中的查詢字串，您的快取政策或原始伺服器請求政策必須針對查詢字串設定為 All (全部)。
- 函數可以為檢視器請求和原始伺服器請求事件建立或更新查詢字串 (原始伺服器請求事件僅存在於 Lambda@Edge 中)。
- 函數可以讀取查詢字串，但無法為原始伺服器回應和檢視器回應事件建立或更新查詢字串 (原始伺服器回應事件僅存在於 Lambda@Edge 中)。
- 如果函數建立或更新查詢字串，將適用下列限制：
  - 查詢字串不可包含空格、控制字元或片段識別碼 (#)。
  - URI 的總大小 (包含查詢字串) 必須小於 8,192 個字元。
  - 我們建議您於 URI 和查詢字串使用 % 編碼。如需詳細資訊，請參閱 URI 和查詢字串編碼。

URI

如果函數為請求變更了 URI，這不會改變針對請求進行的快取動作，也不會改變請求轉傳目的地的原始伺服器。

URI 的總大小 (包含查詢字串) 必須小於 8,192 個字元。

URI 和查詢字串編碼

傳遞給邊緣函數的 URI 和查詢字串值是使用 UTF-8 編碼。您的函數應針對其傳回的 URI 和查詢字符串值使用 UTF-8 編碼。百分比編碼與 UTF-8 編碼相容。

下列清單說明如何 CloudFront 處理 URI 和查詢字串值編碼：

- 當請求中的值是 UTF-8 編碼時，將值 CloudFront 轉發到您的函數而不進行更改。
- 當請求中的值是 ISO-8859-1 編碼時， CloudFront 將值轉換為 UTF-8 編碼，然後再將它們轉發到函數。
- 當要求中的值使用其他字元編碼進行編碼時， CloudFront 假設它們是 ISO-8859-1 編碼，並嘗試從 ISO-8859-1 轉換為 UTF-8。

> **Important**
>
> 轉換後的字元中的值可能是原始伺服器請求的不正確轉譯。這可能會導致函數或原始伺服器產生意外結果。

CloudFront 轉發到原點的 URI 和查詢字符串值取決於函數是否更改值：

- 如果函數未變更 URI 或查詢字串，請將要求中接收到的值 CloudFront 轉寄至您的來源。
- 如果函數變更 URI 或查詢字串，則 CloudFront 會轉寄 UTF-8 編碼值。

Microsoft Smooth Streaming

您無法將邊緣函數與您用於串流媒體檔案的 CloudFront 分發搭配使用，而這些檔案已轉換為 Microsoft 流暢串流格式。

標記

您無法將標籤新增至邊緣函數。若要進一步瞭解中標記 CloudFront，請參閱測試分佈。

CloudFront 功能限制

下列限制僅適用於 CloudFront 函數。

如需有關配額 (先前稱為限制) 的資訊，請參閱 CloudFront 功能配額。

日誌

函數中的 CloudFront 函數日誌被截斷為 10 KB。

請求內文

CloudFront 函數無法訪問 HTTP 請求的主體。

使用 AWS Security Token Service CloudFront KeyValueStore API 時的區域端點

當您使用具有臨時安全登入資料的簽名版本 4A (SigV4a) 呼叫 CloudFront KeyValueStore API 時 (例如，使用 AWS Identity and Access Management (IAM) 角色時，請務必從中的區域端點要求臨時登入資料。 AWS STS如果您使用 AWS STS (`sts.amazonaws.com`) 的全域端點， AWS STS 將會從

SigV4a 不支援的全域端點產生臨時認證。結果，您將收到一個身份驗證錯誤。若要解決此問題，請使用 IAM 使用者指南 AWS STS中列出的任何區域端點。如果您將 SAML 設定為使用 AWS STS 區域端點，請參閱如何使用地區性 SAML 端點進行容錯移轉部落格文章。

執行期

CloudFront Functions 執行階段環境不支援動態程式碼評估，而且會限制對網路、檔案系統和計時器的存取。如需詳細資訊，請參閱 限制功能。

> **Note**
>
> 若要使用 CloudFront KeyValueStore，您的 CloudFront 函數必須使用JavaScript 執行階段 2.0。

運算利用率

CloudFront 函數對執行所需的時間有限制，以計算使用率來衡量。運算利用率是介於 0 到 100 之間的數字，表示函數執行所花費的時間，以所允許時間上限的百分比表示。例如，35 的運算利用率表示函數以所允許時間上限的 35% 完成。

當您測試函數時，您可以在測試事件的輸出中看到運算利用率值。對於實際執行功能，您可以在 CloudFront 主控台的 [監視] 頁面或中檢視運算使用率測量結果 CloudWatch。

對 Lambda@Edge 的限制

下列限制僅適用於 Lambda@Edge。

如需配額的詳細資訊，請參閱 Lambda@Edge 的配額。

DNS 解析

CloudFront 在執行原始請求 Lambda @Edge 函數之前，對原始網域名稱執行 DNS 解析。如果您網域的 DNS 服務遇到問題，而且 CloudFront 無法解析網域名稱以取得 IP 位址，則不會叫用 Lambda @Edge 函數。 CloudFront會將 HTTP 502 狀態碼（錯誤的網關）返回給客戶端。如需詳細資訊，請參閱 HTTP 502 狀態碼 (DNS 錯誤)。

如需管理 DNS 容錯移轉的詳細資訊，請參閱 Amazon Route 53 開發人員指南中的設定 DNS 容錯移轉。

HTTP 狀態碼

檢視器回應事件的 Lambda @Edge 函數無法修改回應的 HTTP 狀態碼，無論回應來自來源還是 CloudFront 快取。

Lambda 函數版本

您必須使用 Lambda 函數的已編號版本，而不是 $LATEST 或別名。

Lambda 區

Lambda 函數必須位於美國東部 (維吉尼亞北部)區域。

Lambda 角色許可

與 Lambda 函數關聯的 IAM 執行角色必須可由服務主體 `lambda.amazonaws.com` 和 `edgelambda.amazonaws.com` 擔任。如需詳細資訊，請參閱 設定身分與存取權管理權限和角色 @Edge。

Lambda 功能

Lambda@Edge 不支援下列 Lambda 函數：

- 自動以外的 Lambda 執行階段管理組態 (預設值)
- 設定 Lambda 函數以存取虛擬私人 VPC 內的資源
- Lambda 函數無效字母隊列
- Lambda 環境變數 (自動支援的保留環境變數除外)
- 含AWS Lambda 圖層的 Lambda 函數
- 使用 AWS X-Ray
- Lambda 佈建並行

> **Note**
>
> Lambda @Edge 函數具有與 Lambda 函數相同的區域並行功能。但是，當並行 Lambda @Edge 執行的配額增加時，所有複寫 Lambda @Edge 函數的 AWS 區域 配額都會增加。如需詳細資訊，請參閱 Lambda@Edge 的配額。

- 定義為容器映像的 Lambda 函數
- 使用 arm64 架構的 Lambda 函數
- 具有超過 512 MB 暫時儲存體的 Lambda 函數
- 擷取 JSON 結構化格式的 Lambda 函數記錄
- 控制 Lambda 函數日誌的日誌層級精細度
- 設定 Lambda 將日誌傳送到哪個 Amazon CloudWatch 日誌群

支援的執行期

Lambda@Edge 支援具有下列執行時間的 Lambda 函數：

Node.js	Python
• Node.js 20 • Node.js 18 • Node.js 16[1] • Node.js • Node.js 12[2] • Node.js 10[2] • Node.js 8[2] • Node.js 6[2]	• Python 3.12 • Python 3.11 • Python 3.10 • Python 3.9 • Python 3.8 • Python 3.7

[1] 此版本的 Node.js 已經到了生命週期結束，很快就會被 AWS Lambda棄用。

[2] 此版本的 Node.js 已經到了生命週期結束，並已完全取代 AWS Lambda。

您無法使用已取代版本的 Node.js 來建立或更新函式。您只能將現有函數與這些版本與 CloudFront 發行版產生關聯。與發行版相關聯的這些版本的函數將繼續執行。但是，我們建議您將函數移至較新版本的 Node.js。如需詳細資訊，請參閱AWS Lambda 開發人員指南中的執行階段淘汰原則和上 GitHub的 Node.js 發行排程。

> **Tip**
>
> 最佳作法是使用所提供執行階段的最新版本，以改善效能和新功能。

CloudFront標頭

Lambda @Edge 函數可以讀取、編輯、移除或新增中列出的任何 CloudFront 標頭新增 CloudFront 要求標頭。

> **備註**
>
> - 如果您想 CloudFront 要新增這些標頭，您必須設定 CloudFront 使用快取原則或原始要求原則來新增這些標頭。
> - CloudFront 在檢視器要求事件之後新增標頭，這表示檢視器要求中的 Lambda @Edge 函數無法使用標頭。標頭僅適用於來源請求和來源回應中的 Lambda @Edge 函數。
> - 如果檢視者要求包含具有這些名稱的標頭，而且您設定 CloudFront 為使用快取原則或原始要求原則新增這些標頭，則 CloudFront 會覆寫檢視器要求中的標頭值。面向檢視者的函數會從檢視器要求中查看標頭值，而面向原點的函數則會看到新增的標頭值。 CloudFront
> - 如果查看器請求函數添加了`CloudFront-Viewer-Country`標頭，則驗證失敗，並將 HTTP 狀態碼 502（錯誤網關） CloudFront 返回給查看器。

使用包含內文選項時的要求內文限制

如果您選擇 Include Body (包含內文) 選項，來向您的 Lambda@Edge 函數顯露要求內文，則請您注意，內文公開或替換的部分須遵循下列的資訊和大小配額。

- CloudFront 在將請求主體公開給 Lambda @Edge 之前，始終是 base64 對請求主體進行編碼。
- 如果請求主體很大，請在將其公開給 Lambda @Edge 之前將其 CloudFront 截斷，如下所示：
  - 針對檢視器請求事件，會在 40 KB 處截斷內文。
  - 針對原始伺服器請求事件，會在 1 MB 處截斷內文。
- 如果您以唯讀方式存取要求主體，請 CloudFront 將完整的原始要求主體傳送至來源。
- 如果 Lambda@Edge 函數替換了請求內文，則下列大小配額適用於函數所傳回的內文：
  - 如果 Lambda@Edge 函數以純文字格式傳回內文：
    - 針對檢視器請求事件，會在 40 KB 處截斷內文。
    - 針對原始伺服器請求事件，會在 1 MB 處截斷內文。
  - 如果 Lambda@Edge 函數以 base64 編碼的文字傳回內文：
    - 針對檢視器請求事件，會在 53.2 KB 處截斷內文。
    - 針對原始伺服器請求事件，會在 1.33 MB 處截斷內文。

回應逾時和保持連線逾時 (僅限自訂來源)

如果您使用 Lambda @Edge 函數來設定發佈來源的回應逾時或保持使用中逾時，請確認您指定的是來源可支援的值。如需更多詳細資訊，請參閱 回應和保持作用逾時配額。

報告、指標和日誌

CloudFront 提供數個報告、監視和記錄 CloudFront資源的選項：

- 您可以查看和下載報告以查看 CloudFront分佈的使用情況和活動，包括帳單報告、快取統計資料、熱門內容和熱門反向連結。
- 您可以直接在 CloudFront 主控台或使用 Amazon 監控和追蹤 CloudFront，包括邊緣運算功能 CloudWatch。 CloudFront 會針 CloudWatch 對分佈和邊緣函數 (Lambda @Edge 和CloudFront 函數) 傳送各種指標。
- 您可以查看您的 CloudFront 分發通過標準日誌或實時日誌收到的查看者請求的日誌。除了檢視器請求記錄之外，您還可以使用 CloudWatch 日誌來取得邊緣函數 (Lambda @Edge 和 F CloudFront unctions) 的日誌。您也可 AWS CloudTrail 以使用 CloudFront 在您的 AWS 帳戶.
- 您可以使用追蹤 CloudFront 資源的組態變更 AWS Config。

如需上述個別選項的詳細資訊，請參閱以下主題。

主題

AWS 的帳單和使用情況報告 CloudFront

AWS 提供以下兩種使用情況報告 CloudFront：

- 帳單報告是您正在使用之 AWS 服務的所有活動的高階檢視，其中包括 CloudFront。如需詳細資訊，請參閱 the section called “AWS 帳單報表 CloudFront”。
- 用量報告是特定服務的活動摘要 (依小時、日或月彙整)。它也包含使用情況圖表，提供您CloudFront使用情況的圖形表示。如需詳細資訊，請參閱 the section called “AWS 使用報告 CloudFront”。

若要詳細了解這些報告，請參閱the section called “解譯您的 AWS 帳單和使 AWS 用報告 CloudFront”的詳細資訊。

> **Note**
>
> 與其他 AWS 服務一樣，只會按使用量 CloudFront 向您收取費用。如需詳細資訊，請參閱CloudFront 定價。

主題

AWS 帳單報表 CloudFront

您可以在的帳單頁面上檢視 AWS 使用量和費用的摘要 (依服務列出) AWS Management Console。

您還可以 CSV 格式下載報告的更詳細版本。詳細帳單報表包含下列適用於的值 CloudFront：

- ProductCode — `AmazonCloudFront`
- UsageType— 下列其中一個值：
  - 識別資料傳輸類型的代碼
  - `Invalidations`
  - `SSL-Cert-Custom`

  如需詳細資訊，請參閱 the section called “解譯您的 AWS 帳單和使 AWS 用報告 CloudFront”。
- ItemDescription— 的計費費率說明UsageType。
- 使用開始日期/使用結束日期 — 開始計算使用的日期，採用世界協調時間 (UTC)。
- 使用量 — 下列其中一個值：
  - 指定時段期間的請求數目
  - 資料傳輸量 (以 GB 為單位)
  - 失效的物件的數量
  - 您擁有與已啟用 CloudFront 分發相關聯的 SSL 憑證的按比例分配月份總和。例如，如果您有一個整個月期已啟用分佈的相關聯憑證和另一個半月期已啟用分佈相關聯憑證，則此值將為 1.5。

顯示摘要帳單資訊並下載詳細帳單報告

1. 登錄到 AWS Management Console 在https://console.aws.amazon.com/console/home。
2. 在標題列中，選擇您的使用者名稱，然後選擇 Billing Dashboard (帳單儀表板)。
3. 在導覽窗格中，選擇 Bills (帳單)。
4. 若要檢視的摘要資訊 CloudFront，請在「詳細資訊」下選擇CloudFront。
5. 要以 CSV 格式下載詳細的帳單報告，請選擇 Download CSV (下載 CSV)，然後依照畫面上的提示來儲存報告。

AWS 使用報告 CloudFront

AWS 提供的 CloudFront 使用情況報告比計費報告更詳細，但比 CloudFront 存取記錄更詳細。用量報告按小時、日或月提供彙總用量資料；並按地區和使用類型列出操作，例如從澳洲區域轉出的資料。

CloudFront 使用情況報告包含下列值：

- 服務 — `AmazonCloudFront`
- 操作 — HTTP 方法。數值包含 `DELETE`、`GET`、`HEAD`、`OPTIONS`、`PATCH`、`POST` 和 `PUT`。
- UsageType— 下列其中一個值：
  - 識別資料傳輸類型的代碼
  - `Invalidations`
  - `SSL-Cert-Custom`

  如需詳細資訊，請參閱 the section called “解譯您的 AWS 帳單和使 AWS 用報告 CloudFront”。
- 資源 — 與使用關聯的 CloudFront 發行版 ID，或與發 CloudFront 佈相關聯的 SSL 憑證的憑證 ID。
- StartTime/EndTime— 使用套用到的日期，以國際標準時間 (UTC) 表示。
- UsageValue— (1) 指定時段內的要求數目，或 (2) 以位元組為單位傳輸的資料量。

如果您使用 Amazon S3 做為的來源 CloudFront，也請考慮執行 Amazon S3 的用量報告。但是，如果您將 Amazon S3 用於作為分 CloudFront發的來源以外的目的，則可能不清楚哪些部分適用於您的 CloudFront 用量。

> **Tip**
>
> 如需針對您的物件 CloudFront 收到的每個要求的詳細資訊，請開啟散發的 CloudFront存取記錄。如需詳細資訊，請參閱 the section called “使用標準日誌 (存取日誌)”。

解譯您的 AWS 帳單和使 AWS 用報告 CloudFront

您的 AWS 帳單 CloudFront 包括代碼和縮寫，這些代碼和縮寫可能不會立即顯而易見。下表中的第一欄列出了帳單中所顯示的項目，並說明了每個項目的含義。

此外，您還可以取得包含比帳單更詳細資訊的 AWS 使用情況 AWS 報告 CloudFront。 CloudFront 表格中的第二個欄位列出用量報告中所顯示的項目，並顯示帳單項目與用量報告項目之間的關聯。

兩欄中的大多數代碼都包含兩個字母的縮寫，指出活動的位置。在下表中，代碼中的##會以下列其中一個兩個字母的縮寫來取代您的 AWS 帳單與使用量報表中：

- AP：香港特別行政區、菲律賓、南韓、台灣和新加坡 (亞太區域)
- AU：澳洲
- CA：加拿大
- EU：歐洲和以色列
- IN：印度
- JP：日本
- ME：中東
- SA：南美洲
- US：美國
- ZA：南非

如需按區域定價的詳細資訊，請參閱 Amazon CloudFront 定價。

> **Note**
>
> 此表不包括將物件從 Amazon S3 儲存貯體傳輸到 CloudFront節點的費用。這些費用（如果有的話）會顯示在 AWS 帳單的「AWS 資料傳輸」部分中。

CloudFront 帳單中的項目	使用情況報告中使用情況類型欄中的 CloudFront 值
## -輸出DataTransfer字節 從*##*中的節 CloudFront 點回應使用者GET和HEAD要求所提供的位元組總數。	*region*-Out-Bytes-HTTP-Static: 透過 HTTP 為 TTL ≥ 3,600 秒的物件提供的位元組數。 *region*-Out-Bytes-HTTPS-Static: 透過 HTTPS 為 TTL ≥ 3,600 秒的物件提供的位元組數。 *region*-Out-Bytes-HTTP-Dynamic: 透過 HTTP 為 TTL < 3,600 秒的物件提供的位元組數。 *region*-Out-Bytes-HTTPS-Dynamic: 透過 HTTPS 為 TTL < 3,600 秒的物件提供的位元組數。 *region*-Out-Bytes-HTTP-Proxy: 透過 HTTP 傳 CloudFront 回檢視器以回應DELETE、OPTIONSPATCHPOST、和PUT要求的位元組。 *region*-Out-Bytes-HTTPS-Proxy: 透過 HTTPS 回應、、和PUT要求DELETEOPTIONS，PATCH從 CloudFront 檢視者傳回的位元組。POST
## -輸出 DataTransfer OB 從節點傳輸到原始或 CloudFront 邊緣函數以回應DELETE、OPTIONSPATCHPOST、和PUT請求	*region*-Out-OBytes-HTTP-Proxy 透過 HTTP 從 CloudFront 邊緣位置傳輸到原始或邊緣函式以回

CloudFront 帳單中的項目	使用情況報告中使用情況類型欄中的 CloudFront 值
的位元組總數。費用包括從客戶端到服務器的 WebSocket 數據傳輸。	應DELETE、OPTIONSPATCHPOST、和PUT要求的位元組總計。 *region*-Out-OBytes-HTTPS-Proxy 透過 HTTPS 從 CloudFront 邊緣位置傳輸到原始或邊緣函式以回應DELETE、OPTIONSPATCHPOST、和PUT要求的位元組總數。
region-Requests-Tier1 HTTP GET 和 HEAD 請求數。	*region*-Requests-HTTP-Static 為物件提供的 HTTP GET 和 HEAD 請求數 (TTL ≥ 3,600 秒)。 *region*-Requests-HTTP-Dynamic 為物件提供的 HTTP GET 與 HEAD 請求數 (TTL < 3,600 秒)。
region-Requests-Tier2-HTTPS HTTPS GET 和 HEAD 請求數。	*region*-Requests-HTTPS-Static 為物件提供的 HTTPS GET 和 HEAD 請求數 (TTL ≥ 3,600 秒)。 *region*-Requests-HTTPS-Dynamic 為物件提供的 HTTPS GET 與 HEAD 請求數 (TTL < 3,600 秒)。

CloudFront 帳單中的項目	使用情況報告中使用情況類型欄中的 CloudFront 值
region-Requests-HTTP-Proxy CloudFront轉寄至原始或邊緣函式的 HTTP DELETE POST、、、和PUT要求數目。OPTIONS PATCH 還包括 CloudFront 轉發到源或邊緣函數的 HTTP GET 請WebSocket求（帶有Upgrade: websocket 標頭的請求）的數量。	*region*-Requests-HTTP-Proxy 與 CloudFront 帳單中的相應項目相同。
region-Requests-HTTPS-Proxy CloudFront轉寄至原始或邊緣函式的 HTTPS DELETE POST、、、和PUT要求數目。OPTIONS PATCH 還包括 CloudFront 轉發到源或邊緣函數的 HTTPS GET 請WebSocket求（帶有Upgrade: websocket 標頭的請求）的數量。	*region*-Requests-HTTPS-Proxy 與 CloudFront 帳單中的相應項目相同。
region-Requests-HTTPS-Proxy-FLE 使用欄位層級加密處理的 HTTPS DELETE OPTIONS PATCH、、和POST要求數目，並 CloudFront 轉送至您的來源或邊緣功能。	*region*-Requests-HTTPS-Proxy-FLE 與 CloudFront 帳單中的相應項目相同。
## -字節-OriginShield 從原始伺服器傳輸至任何區域邊緣快取的總位元組數，包括做為 Origin Shield 啟用的區域邊緣快取。	*##* -字節-OriginShield 從原始伺服器傳輸至任何區域邊緣快取的總位元組數，包括做為 Origin Shield 啟用的區域邊緣快取。
## -卵節-OriginShield 從任何區域邊緣快取傳輸至原始伺服器的總位元組數，包括做為 Origin Shield 啟用的區域邊緣快取。	*##* -卵節-OriginShield 從任何區域邊緣快取傳輸至原始伺服器的總位元組數，包括做為 Origin Shield 啟用的區域邊緣快取。

CloudFront 帳單中的項目	使用情況報告中使用情況類型欄中的 CloudFront 值
## -請求-OriginShield 轉到 Origin Shield 的請求數做為增量改進層。對於代理至原始伺服器的動態 (非可快取) 請求，Origin Shield 一律為增量改進層。對於可快取的請求，Origin Shield 有時是增量改進層。 如需詳細資訊，請參閱 the section called “預估 Origin Shield 成本”。	## -請求-OriginShield 轉到 Origin Shield 的請求數做為增量改進層。對於代理至原始伺服器的動態 (非可快取) 請求，Origin Shield 一律為增量改進層。對於可快取的請求，Origin Shield 有時是增量改進層。 如需詳細資訊，請參閱 the section called “預估 Origin Shield 成本”。
無效化 使物件無效 (從 CloudFront 邊緣位置移除物件) 的費用；如需詳細資訊，請參閱。支付檔案失效	無效化 與 CloudFront 帳單中的相應項目相同。
SSL-Cert-Custom 使用具 CloudFront 備備用網域名稱 (例如 example.com) 的 SSL 憑證，而不是使用預設 CloudFront SSL 憑證和 CloudFront 指派給您分發的網域名稱的費用。	SSL-Cert-Custom 與 CloudFront 帳單中的相應項目相同。

在主控台中檢視 CloudFront 報告

您可以在主控台中檢視下列 CloudFront 活動報告：

主題

- 檢視 CloudFront 快取統計報告
- 檢視 CloudFront 熱門物件報告
- 檢視 CloudFront 熱門反向連結報告
- 檢視 CloudFront 使用量報告
- 檢視 CloudFront 觀眾報表

這些報告大多是基於 CloudFront 訪問日誌中的數據，其中包含有關 CloudFront 接收的每個用戶請求的詳細信息。您不需要啟用存取日誌來查看報告。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌)。

檢視 CloudFront 快取統計報告

Amazon CloudFront 快取統計資料報告包含下列資訊：

- 請求總數 – 顯示所有 HTTP 狀態碼 (例如，200 或 404) 和所有方法 (例如，GET、HEAD 或 POST) 的請求總數。
- 依結果類型排列的檢視器要求百分比 — 以所選CloudFront 發佈的檢視器要求總數百分比顯示點擊、遺漏和錯誤。
- 已傳輸給檢視器的位元組數 – 顯示總位元組數和未命中的位元組數。
- HTTP 狀態碼 – 根據 HTTP 狀態碼來顯示檢視器的請求。
- 未完成下載的 GET 請求百分比 – 針對未完成下載所請求物件的檢視器 GET 請求數，顯示該請求數佔總請求數的百分比。

這些統計資料的資料來自與 CloudFront 存取記錄相同的來源，但是您不需要啟用存取記錄即可檢視快取統計資料。

您可以使用每小時或每天的資料點來顯示過去 60 天內指定日期範圍的圖表。您通常可以檢視一小時前最近 CloudFront 收到的要求的相關資料，但資料偶爾可能會延遲多達 24 小時。

主題

- 在主控台中檢視 CloudFront 快取統計資料報告
- 以 CSV 格式下載資料
- 快取統計資料圖表如何與 CloudFront標準記錄檔中的資料相關 (存取日誌)

在主控台中檢視 CloudFront 快取統計資料報告

您可以在主控台中檢視 CloudFront 快取統計資料報告。

檢視 CloudFront 快取統計資料

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。

2. 在功能窗格中，選擇 [快取統計資料]。
3. 在「CloudFront 快取統計資料報表」窗格中，針對「開始日期」和「結束日期」，選取您要顯示快取統計資料圖表的日期範圍。可用的範圍取決於您針對 Granularity (精細度) 選擇的值：

   - 每日 – 若要顯示每天一個資料點的圖表，請選擇前 60 天內的任何日期範圍。
   - 每小時 – 若要顯示每小時一個資料點的圖表，請選擇前 60 天內的任意日期範圍 (最多 14 天)。

   日期和時間都使用國際標準時間 (UTC)。
4. 對於 Granularity (精細度)，指定要在圖表中顯示每天一個資料點，或每小時一個資料點。如果您指定的日期範圍大於 14 天，則每小時指定一個資料點的選項不可用。
5. 針對 Viewer Location (檢視器位置)，選擇發出檢視器請求的洲別，或是選擇 All Locations (所有位置)。快取統計資料圖表包括從指定位置 CloudFront 接收之要求的資料。
6. 在 Distribution (分佈) 清單中，選擇要在用量圖表中顯示其資料的分佈：

   - 個別分佈 — 圖表會顯示所選 CloudFront 分佈的資料。Distribution (分佈) 清單會顯示分佈的分佈 ID 和替代網域名稱 (CNAME) (如果有的話)。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。
   - 所有分配 — 圖表會顯示與目前 AWS 帳戶相關聯的所有分配的總和資料，但不包括您已刪除的分配。
7. 選擇更新。

若要檢視圖表中每日或每小時資料點的資料，請將游標暫留在資料點上。

對於顯示傳輸資料的圖表，請注意，可以將每個圖表的垂直擴展變更為 GB、MB 或 KB。

以 CSV 格式下載資料

您可使用 CSV 格式下載快取統計資料報告。本節說明如何下載報告和描述報告中的值。

以 CSV 格式下載快取統計資料報告。

1. 檢視快取統計資料報告時，選擇 [CSV]。
2. 在開啟檔案名稱對話方塊中，選擇要開啟或儲存檔案。

有關報告的資訊

報告的前幾行包含以下資訊：

版本

此 CSV 檔案的格式版本。

報告

報告名稱。

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

StartDateUTC

您所執行報告日期範圍的開始時間，是以國際標準時間 (UTC) 為準。

EndDateUTC

您執行報告的日期範圍的結束時間，以國際標準時間 (UTC) 為準。

GeneratedTimeUTC

您執行報告的日期和時間，以國際標準時間 (UTC) 為準。

精細程度

報告中的每一行表示一小時還是一天。

ViewerLocation

該檢視器所請求來源的洲別，或是如果您選擇用於所有位置而下載報告的 ALL。

快取統計資料報告中的資料

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

ViewerLocation

該檢視器所請求來源的洲別，或是如果您選擇用於所有位置而下載報告的 ALL。

TimeBucket

資料所應用的小時或天，以國際標準時間 (UTC) 為準。

RequestCount

所有 HTTP 狀態碼 (例如，200 或 404) 和所有方法 (例如，GET、HEAD 或 POST) 的請求總數

HitCount

從 CloudFront 邊緣快取提供物件的檢視器要求數目。

MissCount

物件目前不在邊緣快取中的檢視器要求數目，因此 CloudFront 必須從原始位置取得物件。

ErrorCount

導致錯誤的檢視者要求數目，因此 CloudFront 未提供物件服務。

IncompleteDownloadCount

檢視器啟動下載物件但未完成的檢視器請求數量。

HTTP2xx

用於 HTTP 狀態碼為 2xx 的值 (成功) 的檢視器請求數量。

HTTP3xx

用於 HTTP 狀態碼 3xx 的值 (需要額外動作) 的檢視器請求數量。

HTTP4xx

用於 HTTP 狀態碼 4xx 的值 (用戶端錯誤) 的檢視器請求數量。

HTTP5xx

用於 HTTP 狀態碼 5xx 的值 (伺服器錯誤) 的檢視器請求數量。

TotalBytes

針對所有 HTTP 方法的所有要求，提供給檢視者的位元組總數。 CloudFront

BytesFromMisses

用於在請求時不在節點快取中的物件，所提供給檢視器的位元組數。這個值是從原始碼傳輸到 CloudFront 邊緣快取的位元組的一個很好的近似值。但是，它會排除已經在節點快取但已過期的物件的請求。

快取統計資料圖表如何與 CloudFront標準記錄檔中的資料相關 (存取日誌)

下表顯示 CloudFront 主控台中的快取統計資料圖表如何與 CloudFront 存取日誌中的值對應。如需 CloudFront 存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

請求總數

此圖表顯示所有 HTTP 狀態碼 (例如，200 或 404) 和所有方法 (例如，GET、 HEAD 或 POST) 的請求總數 此圖表中顯示的請求總數等於相同時間段中存取日誌檔中的請求總數。

依結果類型的檢視器請求百分比

此圖表會以所選分 CloudFront 佈的檢視者要求總數百分比顯示點擊、遺漏和錯誤：

- 點擊 — 從CloudFront 邊緣快取提供物件的檢視器要求。在存取日誌中，這些都是 `x-edge-response-result-type` 值為 `Hit` 的請求。
- 錯過 — 物件目前不在邊緣快取中的檢視器要求，因此 CloudFront 必須從原始位置取得物件。在存取日誌中，這些都是 `x-edge-response-result-type` 值為 `Miss` 的請求。
- Error — 導致錯誤的檢視者要求，因此 CloudFront 未提供物件。在存取日誌中，這些都是 `x-edge-response-result-type` 值為 `Error`、`LimitExceeded` 或 `CapacityExceeded` 的請求。

圖表並不包括用於在邊緣快取中但已過期的物件所做的重新整理命中請求。在存取日誌中，重新整理命中是 `x-edge-response-result-type`值為 `RefreshHit` 的請求。

傳輸到檢視器的位元組數

此圖表顯示兩個值：

- 位元組總數 — 為了回應所有 HTTP 方法的所有要求而 CloudFront 提供給檢視者的位元組總數。在 CloudFront 存取記錄檔中，「總位元組」是相同時段內所有要求之`sc-bytes`資料行中值的總和。
- 未命中的位元組數 – 在請求發出時，針對不在邊緣快取中的物件，提供給檢視器的位元組數。在 CloudFront 存取記錄中，未命中的位元組是`sc-bytes`資料行中要求值的總和，其值`x-edge-result-type`為`Miss`。這個值是從原始碼傳輸到CloudFront 邊緣快取的位元組的一個很好的近似值。但是，它會排除已經在節點快取但已過期的物件的請求。

HTTP 狀態碼

此圖表透過 HTTP 狀態碼顯示檢視器請求。在 CloudFront 存取記錄中，狀態碼會顯示在`sc-status`欄中：

- 2xx – 請求已成功。
- 3xx – 需要執行其他動作。例如，301 (永久移除) 表示請求的物件已經移到不同的位置。
- 4xx – 用戶端明顯出錯。例如，404 (未找到) 表示未找到用戶端請求的物件。
- 5xx – 原始伺服器並未完成請求的請求。例如，503 (服務無法使用) 表示原始伺服器目前無法使用。

未完成下載的 GET 請求的百分比

此圖表顯示未完成下載請求物件的檢視器 GET 請求，其表示為請求總數的百分比。一般而言，因為檢視器取消下載，致使下載物件不完全；例如，按下了不同的連結或關閉瀏覽器。在 CloudFront 存取記錄中，這些要求`200`在`sc-status`欄中的值為，且`x-edge-result-type`欄`Error`中的值為。

檢視 CloudFront 熱門物件報告

檢視 Amazon CloudFront 熱門物件報告，查看過去 60 天內指定日期範圍內分發的 50 個最受歡迎的物件。您也可以檢視這些物件的統計資料，包括下列項目：

- 物件的要求數目
- 命中和未命中數
- 命中率
- 遺漏服務的位元組數
- 服務的總位元組
- 未完成的下載次數
- 依照 HTTP 狀態碼的要求數目 (2xx、3 XX、4 XX 及 5xx)

這些統計資料的資料來自與 CloudFront 存取記錄相同的來源，但您不需要啟用存取記錄功能即可檢視常用物件。

主題

- 在主控台中檢視 CloudFront 熱門物件報告
- 如何 CloudFront 計算流行對象統計
- 以 CSV 格式下載資料
- 常用物件報表中的資料與 CloudFront 標準記錄檔中的資料 (存取記錄) 之間的關聯性

在主控台中檢視 CloudFront 熱門物件報告

您可以在主控 CloudFront 台中檢視常用物件報告。

若要檢視 CloudFront 散佈的常用物件

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [常用物件]。
3. 在「CloudFront 常用物件報表」窗格中，針對「開始日期」和「結束日期」，選取您要顯示常用物件清單的日期範圍。您可以選擇過去 60 天內的任何日期範圍。

   日期和時間都使用國際標準時間 (UTC)。
4. 在 Distribution (分佈) 清單中，選擇要顯示其熱門物件清單的分佈。
5. 選擇更新。

如何 CloudFront 計算流行對象統計

若要準確計算發行版中前 50 個物件的 CloudFront 數量，請從午夜開始以 10 分鐘的間隔計算所有物件的要求，並在接下來的 24 小時內保留前 150 個物件的總計。(CloudFront 也會保留前 150 個物件的每日總數，持續 60 天。)

在清單底部附近，物件不斷上升或從清單中移除，因此這些物件的總計是近似值。150 個對象列表頂部的 50 個對象可能會在列表中上升和落入列表中，但它們很少完全退出列表，因此這些對象的總計更可靠。

當物件從前 150 個物件的清單中移除，然後在一天的過程中再次上升到清單時，就會 CloudFront 增加清單中遺失該物件期間的估計要求數目。此預估是根據在該時段內，由任一物件在清單底部所接收到的請求數量。

如果物件在當天晚些時候升到前 50 個物件，當物件超出前 150 個物件時 CloudFront 收到的要求數目預估值，通常會導致常用物件報告中的要求數目超過該物件存取記錄中出現的要求數目。

以 CSV 格式下載資料

您可使用 CSV 格式下載熱門物件報告。本節說明如何下載報告和描述報告中的值。

以 CSV 格式下載熱門物件報告。

1. 檢視熱門物件報表時，請選擇 [CSV]。

2. 在開啟檔案名稱對話方塊中，選擇要開啟或儲存檔案。

有關報告的資訊

報告的前幾行包含以下資訊：

版本

此 CSV 檔案的格式版本。

報告

報告名稱。

DistributionID

您為執行報告所分佈的的 ID。

StartDateUTC

您所執行報告日期範圍的開始時間，是以國際標準時間 (UTC) 為準。

EndDateUTC

您執行報告的日期範圍的結束時間，以國際標準時間 (UTC) 為準。

GeneratedTimeUTC

您執行報告的日期和時間，以國際標準時間 (UTC) 為準。

熱門物件報告中的資料

該報告包含以下值：

DistributionID

您為執行報告所分佈的的 ID。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

物件

該物件 URL 的最後 500 個字元。

RequestCount

此物件的請求總數。

HitCount

從 CloudFront 邊緣快取提供物件的檢視器要求數目。

MissCount

物件目前不在邊緣快取中的檢視器要求數目，因此 CloudFront 必須從原始位置取得物件。

HitCountPct

`HitCount` 值做為 `RequestCount` 值的百分比。

BytesFromMisses

當請求時，物件不在節點快取中，為了該物件所提供的檢視器位元組數。

TotalBytes

CloudFront 為了回應所有 HTTP 方法的所有要求，此物件所提供給檢視者的位元組總數。

IncompleteDownloadCount

檢視器啟動下載但未完成的檢視器請求數量。

HTTP2xx

用於 HTTP 狀態碼為 2xx 的值 (成功) 的檢視器請求數量。

HTTP3xx

用於 HTTP 狀態碼 3xx 的值 (需要額外動作) 的檢視器請求數量。

HTTP4xx

用於 HTTP 狀態碼 4xx 的值 (用戶端錯誤) 的檢視器請求數量。

HTTP5xx

用於 HTTP 狀態碼 5xx 的值 (伺服器錯誤) 的檢視器請求數量。

常用物件報表中的資料與 CloudFront 標準記錄檔中的資料 (存取記錄) 之間的關聯性

下列清單顯示 CloudFront 主控台中常用物件報告中的值如何與 CloudFront 存取記錄中的值對應。如需 CloudFront 存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

URL

檢視器用於存取該物件的 URL 的最後 500 個字元。

請求

物件的請求總數。這個值通常與 CloudFront 存取記錄檔中物件的GET要求數目密切相對應。

命中

從 CloudFront Edge 快取提供物件的檢視器要求數目。在存取日誌中，這些都是 x-edge-response-result-type 值為 Hit 的請求。

未命中數

物件不在邊緣快取中的檢視器要求數目，因此會從您的來源 CloudFront 擷取物件。在存取日誌中，這些都是 x-edge-response-result-type 值為 Miss 的請求。

命中率

Hits (命中) 欄位的值，佔 Requests (請求) 欄位值的百分比。

未命中的位元組

用於在請求時不在節點快取中的物件，所提供給檢視器的位元組數。在 CloudFront 存取記錄中，未命中的位元組是sc-bytes資料行中要求值的總和，其值x-edge-result-type為Miss。

總位元組數

為了回應所有 HTTP 方法之物件的所有要求而 CloudFront 提供給檢視者的位元組總數。在 CloudFront 存取記錄中，位元組總數是相同時段內所有要求之sc-bytes資料行中值的總和。

不完整的下載

未完成下載請求物件的檢視器請求數量。一般而言，未下載完全的原因是檢視器取消下載，例如，按下了不同的連結或關閉瀏覽器。在 CloudFront 存取記錄中，這些要求200在sc-status欄中的值為，且x-edge-result-type欄Error中的值為。

2xx

HTTP 狀態碼為 2xx、Successful 的請求數。在 CloudFront 存取記錄中，狀態碼會顯示在sc-status欄中。

3xx

其 HTTP 狀態碼為 3xx、Redirection 的請求數量。3xx 狀態碼代表需要執行其他的動作。例如，301 (永久移除) 表示請求的物件已經移到不同的位置。

4xx

其 HTTP 狀態碼為 4xx、Client Error 的請求數量。4xx 狀態碼代表用戶端明顯出錯。例如，404 (未找到) 表示未找到用戶端請求的物件。

5xx

其 HTTP 狀態碼為 5xx、Server Error 的請求數量。5xx 狀態碼代表原始伺服器並未完成請求的要求。例如，503 (服務無法使用) 表示原始伺服器目前無法使用。

檢視 CloudFront 熱門反向連結報告

CloudFront 排名最高的反向連結報表包括過去 60 天內任何日期範圍的下列項目：

- 前 25 名反向連結 (針對分發的物件產生最多 HTTP 和 HTTPS 要求的 CloudFront 網站網域)
- 來自反向連結的要求數目
- 來自反向連結的要求數目佔指定期間內要求總數的百分比

排名靠前的反向連結報表的資料是從與存取記錄相同的來源擷 CloudFront 取，但您不需要啟用存取記錄即可檢視頂端反向連結。

頂級引薦網址可以是搜索引擎，直接鏈接到您的對象的其他網站或您自己的網站。例如，如果`https://example.com/index.html`連結至 10 個圖形，`example.com`就是所有 10 個圖形的反向連結。

> **Note**
>
> 如果使用者直接將 URL 輸入到瀏覽器的地址行中，則不會有所請求物件的推薦網站。

主題

- 在主控台中檢視 CloudFront 熱門反向連結報告
- 如何 CloudFront 計算頂級反向連結統計資
- 以 CSV 格式下載資料
- 頂端反向連結報表中的資料與CloudFront 標準記錄檔 (存取記錄) 中的資料之間的關聯性

在主控台中檢視 CloudFront 熱門反向連結報告

您可以在主控台中檢視 CloudFront 熱門反向連結報告。

若要檢視分佈的頂端反向連結 CloudFront

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在功能窗格中，選擇頂端反向連結。
3. 在「CloudFront 頂端反向連結報表」窗格中，針對「開始日期」和「結束日期」，選取您要顯示頂端反向連結清單的日期範圍。

   日期和時間都使用國際標準時間 (UTC)。
4. 在 Distribution (分佈) 清單中，選擇要顯示其最佳推薦網站清單的分佈。
5. 選擇更新。

如何 CloudFront 計算頂級反向連結統計資

若要取得前 25 名反向連結的準確計 CloudFront 數，請以 10 分鐘的間隔計算所有物件的要求，並保留前 75 個反向連結的總計。在清單底部附近，反向連結不斷上升或刪除清單，因此這些反向連結的總計是近似值。

75 個反向連結清單頂端的 25 個反向連結可能會在清單中上升和下降，但它們很少完全退出清單，因此這些反向連結的總計通常更可靠。

以 CSV 格式下載資料

您可使用 CSV 格式下載最佳推薦網站報告。本節說明如何下載報告和描述報告中的值。

以 CSV 格式下載最佳推薦網站報告。

1. 檢視「熱門反向連結」報表時，選擇「CSV」。
2. 在開啟檔案名稱對話方塊中，選擇要開啟或儲存檔案。

有關報告的資訊

報告的前幾行包含以下資訊：

版本

此 CSV 檔案的格式版本。

報告

報告名稱。

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

StartDateUTC

您所執行報告日期範圍的開始時間，是以國際標準時間 (UTC) 為準。

EndDateUTC

您執行報告的日期範圍的結束時間，以國際標準時間 (UTC) 為準。

GeneratedTimeUTC

您執行報告的日期和時間，以國際標準時間 (UTC) 為準。

最佳推薦網站報告中的資料

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

推薦網站

推薦網站的網域名稱。

RequestCount

在 Referrer (推薦網站) 欄位中網域名稱的請求總數。

RequestsPct

推薦網站所提交的請求數量佔指定期間請求總數的百分比。

頂端反向連結報表中的資料與CloudFront 標準記錄檔 (存取記錄) 中的資料之間的關聯性

下列清單顯示 CloudFront 主控台中「熱門反向連結」報表中的值如何與 CloudFront 存取記錄中的值對應。如需 CloudFront 存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

推薦網站

推薦網站的網域名稱。在存取日誌中，推薦網站列於 `cs(Referer)` 欄位中。

請求計數

從 Referrer (推薦網站) 欄位中的網域名稱，所發出請求的總數。此值通常與 CloudFront 存取記錄中來自反向連結的`GET`要求數目密切相對應。

要求 %

推薦網站所提交的請求數量佔指定期間請求總數的百分比。如果您擁有超過 25 個推薦網站，則無法根據此表格中的資料來計算 Request % (請求百分比)，因為 Request Count (請求計數) 欄位並未包含在指定期間內的所有請求。

檢視 CloudFront 使用量報告

CloudFront 使用情況報告包括下列資訊：

- 要求數目 — 顯示在指定 CloudFront 分配的每個時間間隔內，從所選區域中節點 CloudFront 回應的要求總數。
- 通訊協定傳輸的資料和目的地傳輸的資料 — 兩者都會顯示在指定 CloudFront 分佈的每個時間間隔內，從所選區域中 CloudFront 節點傳輸的資料總量。它們以不同的方式分隔資料，如下所示：
  - 依據通訊協定 — 以通訊協定分隔資料：HTTP 或 HTTPS。
  - 依目的地 — 依目的地分隔資料：傳送給您的使用者或您的來源。

CloudFront 使用情況報告以的 AWS 使用情況報告為基礎 CloudFront，不需要任何特殊設定。如需詳細資訊，請參閱 AWS 使用報告 CloudFront。

您可以檢視過去 60 天內指定日期範圍的報告，每小時或每天都有資料點。您通常可以檢視 4 小時前最近 CloudFront 收到的要求的相關資料，但資料偶爾會延遲多達 24 小時。

如需詳細資訊，請參閱 使用情況圖表與使用情 CloudFront 況報告中的資料之間的關聯。

主題

- 在主控台中檢視 CloudFront 使用情況報告
- 以 CSV 格式下載資料
- 使用情況圖表與使用情 CloudFront 況報告中的資料之間的關聯

在主控台中檢視 CloudFront 使用情況報告

您可以在主控台中檢視 CloudFront 使用情況報告。

若要檢視 CloudFront 使用量報告

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [使用報告]。
3. 在「CloudFront 使用情況報告」窗格中，針對「開始日期」與「結束日期」，選取您要顯示其使用情況圖表的日期範圍。可用的範圍取決於您針對 Granularity (精細度) 選擇的值：

   - 每日 — 若要顯示每天一個資料點的圖表，請選擇前 60 天內的任何日期範圍。
   - 每小時 — 若要顯示每小時一個資料點的圖表，請選擇前 60 天內的任意日期範圍 (最多 14 天)。

   日期和時間都使用國際標準時間 (UTC)。
4. 對於 Granularity (精細度)，指定要在圖表中顯示每天一個資料點，或每小時一個資料點。如果您指定的日期範圍大於 14 天，則每小時指定一個資料點的選項不可用。
5. 針對「帳單區域」，選擇具有您要檢視之資料的 CloudFront 帳單區域，或選擇「所有區域」。使用狀況圖表包含在指定區域中節點 CloudFront 處理之要求的資料。 CloudFront 處理請求的區域可能會或可能不符合您使用者的位置。

   僅選取包含在分配價格類別中的「區域」。否則，使用情況圖表可能不會包含任何資料。例如，如果您為分配選擇「價格類別 200」，則不會包含南美洲和澳洲的帳單區域，因此 CloudFront 通常不會處理這些區域的請求。有關價格類別的更多信息，請參閱CloudFront 定價。
6. 在 Distribution (分佈) 清單中，選擇要在用量圖表中顯示其資料的分佈：

   - 個別分佈 — 圖表會顯示所選 CloudFront 分佈的資料。Distribution (分佈) 清單會顯示分佈的分佈 ID 和替代網域名稱 (CNAME) (如果有的話)。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。
   - 所有分佈 (不包括已刪除的分佈) – 圖表會針對與目前 AWS 帳戶具有關聯的所有分佈 (不包括已刪除的分佈)，顯示摘要的資料。

- 所有已刪除的分佈 — 圖表會顯示與目前 AWS 帳戶相關聯且在過去 60 天內刪除的所有分佈的總和資料。

7. 選擇「更新圖形」。

若要檢視圖表中每日或每小時資料點的資料，請將游標暫留在資料點上。

對於顯示傳輸資料的圖表，請注意，可以將每個圖表的垂直擴展變更為 GB、MB 或 KB。

以 CSV 格式下載資料

您可以下載 CSV 格式的使用情況報告。本節說明如何下載報告和描述報告中的值。

以 CSV 格式下載用量報告。

1. 檢視「使用情況」報告時，請選擇 [CSV]。
2. 在開啟檔案名稱對話方塊中，選擇要開啟或儲存檔案。

有關報告的資訊

報告的前幾行包含以下資訊：

版本

此 CSV 檔案的格式版本。

報告

報告名稱。

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告為 ALL，或者如果您為所有已刪除的分佈執行報告，則為 ALL_DELETED。

StartDateUTC

您所執行報告日期範圍的開始時間，是以國際標準時間 (UTC) 為準。

EndDateUTC

您執行報告的日期範圍的結束時間，以國際標準時間 (UTC) 為準。

GeneratedTimeUTC

您執行報告的日期和時間，以國際標準時間 (UTC) 為準。

精細程度

報告中的每一行表示一小時還是一天。

BillingRegion

如果您選擇下載所有計費區域的報告，則該檢視器請求源發自洲別或 ALL。

用量報告中的資料

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告為 ALL，或者如果您為所有已刪除的分佈執行報告，則為 ALL_DELETED。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

BillingRegion

您執行或報表的 CloudFront 帳單區域ALL。

TimeBucket

資料所應用的小時或天，以國際標準時間 (UTC) 為準。

HTTP

在指定 CloudFront 分配的每個時間間隔內，從所選區域中的節點 CloudFront 回應的 HTTP 要求數目。數值包含：

- 數量GET和HEAD請求，這會導致數據傳輸 CloudFront 給您的用戶
- 導致資料傳輸 CloudFront 到來源的、、、和PUT請求的DELETE數量 OPTIONS PATCH POST

HTTPS

在指定 CloudFront 分配的每個時間間隔內，從所選區域中的節點 CloudFront 回應的 HTTPS 要求數目。數值包含：

- 數量GET和HEAD請求，這會導致數據傳輸 CloudFront 給您的用戶
- 導致資料傳輸 CloudFront 到來源的、、、和PUT請求的DELETE數量 OPTIONS PATCH POST

HTTPBytes

在指定 CloudFront 分配的期間內，從所選帳單區域的 CloudFront 節點透過 HTTP 傳輸的資料總量。數值包含：

- 從 CloudFront 您的用戶轉移的數據以響應GET和HEAD請求
- 針對、、和PUT請求 DELETEOPTIONS，PATCH從 CloudFront 您的來源傳輸資料 POST
- 回應刪除、選項、修補程式、POST 和 PUT 要求而從 CloudFront 您的使用者傳輸的資料

HTTPSBytes

在指定 CloudFront 分配的期間內，從所選帳單區域的 CloudFront 節點透過 HTTPS 傳輸的資料總量。數值包含：

- 從 CloudFront 您的用戶轉移的數據以響應GET和HEAD請求
- 針對、、和PUT請求 DELETEOPTIONS，PATCH從 CloudFront 您的來源傳輸資料 POST
- 回應刪除、選項、修補程式、POST 和 PUT 要求而從 CloudFront 您的使用者傳輸的資料

BytesIn

在指定 CloudFront 分配的每個時間間隔內 DELETEOPTIONS，在所選區域內 PATCHPOST，從、、、和PUT請求傳輸 CloudFront 到來源的資料總量。

BytesOut

在指定 CloudFront 分佈的每個時間間隔內，透過 HTTP 和 HTTPS 傳輸給所選區域中使用者的資料總量。 CloudFront 數值包含：

- 從 CloudFront 您的用戶轉移的數據以響應GET和HEAD請求
- 回應DELETE、OPTIONS、和PUT請求而從 CloudFront 您的使用者傳輸的資料 PATCH POST

使用情況圖表與使用情 CloudFront 況報告中的資料之間的關聯

下列清單顯示 CloudFront 主控台中的使用情況圖表如何與使用情況報告中「使用類型」欄中的 CloudFront 值相對應。

主題

- 請求數

- 依據通訊協定傳輸的資料
- 依目的地的資料傳輸

請求數

此圖表顯示在指定分 CloudFront佈的每個時間間隔內，從所選區域中節點 CloudFront 回應的要求總數，並以通訊協定 (HTTP 或 HTTPS) 和類型 (靜態、動態或 Proxy) 分隔。

HTTP 請求數

- *region*-Requests-HTTP-Static：在 TTL ≥ 3600 秒下，為了物件而提供的 HTTP GET 和 HEAD 請求數
- *region*-Requests-HTTP-Dynamic：已為其提供物件的 HTTP GET 與 HEAD 請求數量 (TTL < 3600 秒)
- *##*要求 HTTP 代理伺服器：轉寄至原始伺服器的 HTTP DELETE OPTIONS、、PATCHPOST、和PUT要求數目 CloudFront

HTTPS 請求數

- *region*-Requests-HTTPS-Static：在 TTL ≥ 3600 秒下，為了物件而提供的 HTTPS GET 和 HEAD 請求數
- *region*-Requests-HTTPS-Dynamic：已為其提供物件的 HTTPS GET 與 HEAD 請求數量 (TTL < 3600 秒)
- *##* -請求-代理伺服器：轉發到原始伺服器的 HTTPS DELETE OPTIONS、PATCH、POST、和請PUT求數量 CloudFront

依據通訊協定傳輸的資料

此圖表顯示在指定分 CloudFront 佈的每個時間間隔內，從所選區域中 CloudFront 節點傳輸的資料總量，並以通訊協定 (HTTP 或 HTTPS)、類型 (靜態、動態或 Proxy) 和目的地 (使用者或來源) 區隔。

透過 HTTP 傳輸的資料

- *region*-Out-Bytes-HTTP-Static：在 TTL ≥ 3600 秒下，為了物件而透過 HTTP 所提供的位元組
- *region*-Out-Bytes-HTTP-Dynamic：透過 HTTP 提供給物件的位元組數 (TTL < 3600 秒)
- *##*輸出字節 HTTP 代理：通過 HTTP 返回給查看者的字節 CloudFront 以響應，，和請求 DELETE OPTIONS PATCH POST PUT
- *##*輸出對象 HTTP 代理：通過 HTTP 從 CloudFront 邊緣位置傳輸到您的來源以響應、、和請求的總字節數 DELETE OPTIONS PATCH POST PUT

透過 HTTPS 傳輸的資料

- *region*-Out-Bytes-HTTPS-Static： 在 TTL ≥ 3600 秒下，為了物件而透過 HTTPS 所提供的位元組
- *region*-Out-Bytes-HTTPS-Dynamic：透過 HTTPS 提供給物件的位元組數 (TTL < 3600 秒)
- *##*輸出字節 HTTP 代理：通過 HTTPS 返回給查看者的字節 CloudFront 以響應，，和請求 DELETE OPTIONS PATCH POST PUT
- *##*輸出對象-HTTP 代理：通過 HTTPS 從 CloudFront 邊緣位置傳輸到您的來源以響應，，和請求的總字節數 DELETE OPTIONS PATCH POST PUT

依目的地的資料傳輸

此圖表顯示在指定分 CloudFront 佈的每個時間間隔內，從所選區域中 CloudFront 節點傳輸的資料總量，並以目的地 (使用者或來源)、通訊協定 (HTTP 或 HTTPS) 和類型 (靜態、動態或 Proxy) 區隔開來。

從 CloudFront 您的使用者傳輸的資料

- *region*-Out-Bytes-HTTP-Static：在 TTL ≥ 3600 秒下，為了物件而透過 HTTP 所提供的位元組
- *region*-Out-Bytes-HTTPS-Static： 在 TTL ≥ 3600 秒下，為了物件而透過 HTTPS 所提供的位元組
- *region*-Out-Bytes-HTTP-Dynamic：透過 HTTP 提供給物件的位元組數 (TTL < 3600 秒)
- *region*-Out-Bytes-HTTPS-Dynamic：透過 HTTPS 提供給物件的位元組數 (TTL < 3600 秒)
- *##*輸出字節 HTTP 代理：通過 HTTP 返回給查看者的字節 CloudFront 以響應，，和請求 DELETE OPTIONS PATCH POST PUT
- *##*輸出字節 HTTP 代理：通過 HTTPS 返回給查看者的字節 CloudFront 以響應，，和請求 DELETE OPTIONS PATCH POST PUT

從 CloudFront 您的來源傳輸的資料

- *##*輸出對象 HTTP 代理：通過 HTTP 從 CloudFront 邊緣位置傳輸到您的來源以響應、、和請求的總字節數 DELETE OPTIONS PATCH POST PUT
- *##*輸出對象-HTTP 代理：通過 HTTPS 從 CloudFront 邊緣位置傳輸到您的來源以響應，，和請求的總字節數 DELETE OPTIONS PATCH POST PUT

檢視 CloudFront 觀眾報表

檢 CloudFront 視者報表包含過去 60 天內任何日期範圍的下列資訊：

- 裝置 — 最常用來存取您內容的裝置類型 (例如桌上型電腦或行動裝置)
- 瀏覽器 — 最常用於訪問您的內容的 10 大瀏覽器 (例如 Chrome 或火狐瀏覽器)
- 作業系統 — 存取您的內容 (例如 Linux、macOS 或 Windows) 時最常使用的前 10 種作業系統
- 位置 — 最常存取您內容的觀眾排名前 50 位 (國家/地區或美國/地區)
  - 還可以查看過去 60 天內任何日期範圍內最多 14 天的每小時數據點的位置

您不需要啟用存取記錄，即可查看檢視者圖表和報表。

主題

在主控台中檢視檢視者圖表和報表

您可以在主控台中 CloudFront 檢視檢視者圖表和報表。

檢視檢視 CloudFront 者圖表和報表

1. 登入 AWS Management Console 並開啟 CloudFront 主控台，位於https://console.aws.amazon.com/cloudfront/v4/home。
2. 在導覽窗格中，選擇 [檢視者]。
3. 在「CloudFront 檢視器」窗格中，針對「開始日期」和「結束日期」，選取您要顯示檢視器圖表和報表的日期範圍。

   如果是位置圖表，可用的範圍取決於您針對 Granularity (精細度) 所選擇的值：

   - 每日 – 若要顯示每天一個資料點的圖表，請選擇前 60 天內的任何日期範圍。
   - 每小時 – 若要顯示每小時一個資料點的圖表，請選擇前 60 天內的任意日期範圍 (最多 14 天)。

   日期和時間都使用國際標準時間 (UTC)。
4. (僅適用於瀏覽器與作業系統圖表) 針對 Grouping (分組)，指定要依據名稱 (Chrome、Firefox) 或名稱和版本 (Chrome 40.0、Firefox 35.0)，來將瀏覽器與作業系統分組。

5. (僅適用於位置圖表) 針對 Granularity (精細度)，指定要在圖表中顯示每天一個資料點，或每個小時一個資料點。如果您指定的日期範圍大於 14 天，則每小時指定一個資料點的選項不可用。
6. (僅適用於位置圖表) 針對 Details (詳細資訊)，指定要依據國家/地區或美國的州別，來顯示排名在前面的位置。
7. 在 Distribution (分佈) 清單中，選擇要在用量圖表中顯示其資料的分佈：
   - 個別分佈 — 圖表會顯示所選 CloudFront 分佈的資料。Distribution (分佈) 清單會顯示分佈的分佈 ID 和替代網域名稱 (CNAME) (如果有的話)。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。
   - 所有分配 (不包括已刪除) — 圖表會顯示與目前 AWS 帳戶相關聯的所有分配的總和資料，但不包括您已刪除的分配。
8. 選擇更新。

若要檢視圖表中每日或每小時資料點的資料，請將游標暫留在資料點上。

以 CSV 格式下載資料

您可使用 CSV 格式下載每個檢視器報告。本節說明如何下載報告和描述報告中的值。

以 CSV 格式下載檢視器報告。

1. 檢視「檢視器」報表時，請選擇 [CSV]。
2. 選擇您想要下載的資料，例如 Devices (裝置) 或 Devices Trends (裝置趨勢)。
3. 在開啟檔案名稱對話方塊中，選擇要開啟或儲存檔案。

檢視者報表中包含的資料

每份報告的前幾列包含下列資訊：

版本

此 CSV 檔案的格式版本。

報告

報告名稱。

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

StartDateUTC

您所執行報告日期範圍的開始時間，是以國際標準時間 (UTC) 為準。

EndDateUTC

您執行報告的日期範圍的結束時間，以國際標準時間 (UTC) 為準。

GeneratedTimeUTC

您執行報告的日期和時間，以國際標準時間 (UTC) 為準。

群組 (僅限瀏覽器和作業系統報告)

該資料是否依照名稱，還是瀏覽器或作業系統的名稱和版本進行分組。

精細程度

報告中的每一行表示一小時還是一天。

詳細資訊 (僅限位置報告)

無論請求是按國家/地區或美國州/區域列出。

下列主題說明不同檢視器報表中的資訊。

主題

- 裝置報告
- 裝置趨勢報告
- 瀏覽器報告
- 瀏覽器趨勢報告
- 作業系統報告
- 作業系統趨勢報告
- 位置報告
- 位置趨勢報告

裝置報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

請求

從每種裝置類型 CloudFront 接收的要求數目。

RequestsPct

從每種裝置類型 CloudFront 接收的要求數目佔從所有裝置 CloudFront 接收的要求總數的百分比。

裝置趨勢報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

TimeBucket

適用於小時或天的資料，以國際標準時間 (UTC) 為準。

桌面

在此期間從桌上型電腦 CloudFront 接收的要求數目。

行動應用程式

在此期間從行動裝置 CloudFront 收到的要求數目。行動裝置可以包括平板電腦和行動電話。如果 CloudFront 無法判斷要求是來自行動裝置還是平板電腦，就會計入Mobile欄中。

智慧型電視

此期間從智慧型電視 CloudFront 收到的要求數目。

平板電腦

期間從平板電腦 CloudFront 收到的要求數目。如果 CloudFront 無法判斷要求是來自行動裝置還是平板電腦，就會計入Mobile欄中。

不明

User-Agent HTTP 標頭的值與其中一個標準裝置類型 (例如，Desktop 或 Mobile) 沒有關聯的請求。

空白

期間內 CloudFront 收到的 HTTP User-Agent 標頭中未包含值的要求數目。

瀏覽器報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

群組

從中 CloudFront 接收要求的瀏覽器或瀏覽器和版本，視的值而定Grouping。除了瀏覽器名稱外，可能的值包括下列項目：

- 機器人/爬蟲程式 – 對此值的請求，主要來自於將您的內容建立索引的搜尋引擎。
- 空白 – 其 User-Agent HTTP 標頭值為空白的請求。
- 其他 — CloudFront 識別但不是最受歡迎的瀏覽器。如果 Bot/Crawler、Empty 和/或 Unknown 不出現在前 9 個值中，那麼它們也會包括在 Other 中。
- 未知 – 其 User-Agent HTTP 標頭的值與標準瀏覽器不具有關聯的請求。在這個類別的大多數請求都來自自訂應用程式或指令碼。

請求

從每種瀏覽器類型 CloudFront 接收的要求數目。

RequestsPct

從每種瀏覽器類型 CloudFront 接收的要求數目佔該期間 CloudFront 收到的要求總數的百分比。

瀏覽器趨勢報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

TimeBucket

適用於小時或天的資料，以國際標準時間 (UTC) 為準。

(瀏覽器)

報告中的其餘欄位列出了瀏覽器或瀏覽器及其版本，具體取決於 Grouping 的值。除了瀏覽器名稱外，可能的值包括下列項目：

- 機器人/爬蟲程式 – 對此值的請求，主要來自於將您的內容建立索引的搜尋引擎。
- 空白 – 其 User-Agent HTTP 標頭值為空白的請求。
- 其他 — CloudFront 識別但不是最受歡迎的瀏覽器。如果 Bot/Crawler、Empty 和/或 Unknown 不出現在前 9 個值中，那麼它們也會包括在 Other 中。
- 未知 – 其 User-Agent HTTP 標頭的值與標準瀏覽器不具有關聯的請求。在這個類別的大多數請求都來自自訂應用程式或指令碼。

作業系統報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

群組

從中 CloudFront 接收要求的作業系統或作業系統和版本 (視的值而定) `Grouping`。除了作業系統名稱外，可能的值包括下列項目：

- 機器人/爬蟲程式 – 對此值的請求，主要來自於將您的內容建立索引的搜尋引擎。
- 空白 – 其 `User-Agent` HTTP 標頭值為空白的請求。
- 其他 — CloudFront 識別但不是最受歡迎的操作系統。如果 `Bot/Crawler`、`Empty` 和/或 `Unknown` 不出現在前 9 個值中，那麼它們也會包括在 `Other` 中。
- 未知 – 其 `User-Agent` HTTP 標頭的值與標準瀏覽器不具有關聯的請求。在這個類別的大多數請求都來自自訂應用程式或指令碼。

請求

從每種作業系統類型 CloudFront 接收的要求數目。

RequestsPct

從每個作業系統類型 CloudFront 接收的要求數目，佔該期間 CloudFront 收到的要求總數的百分比。

作業系統趨勢報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 `ALL`。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

TimeBucket

適用於小時或天的資料，以國際標準時間 (UTC) 為準。

(作業系統)

報告中的其餘欄位列出了作業系統或作業系統及其版本，具體取決於 Grouping 的值。除了作業系統名稱外，可能的值包括下列項目：

- 機器人/爬蟲程式 – 對此值的請求，主要來自於將您的內容建立索引的搜尋引擎。
- 空白 – 其 User-Agent HTTP 標頭值為空白的請求。
- 其他 — CloudFront 識別但不是最受歡迎的操作系統。如果 Bot/Crawler、Empty 和/或 Unknown 不出現在前 9 個值中，那麼它們也會包括在 Other 中。
- 未知 – 其 User-Agent HTTP 標頭中未註明作業系統的請求。

位置報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

LocationCode

CloudFront 收到要求的地點縮寫。如需有關可能的值的詳細資訊，請參閱位置報表中的資料與 CloudFront 標準記錄中的資料 (存取記錄) 之間的關聯的位置說明。

LocationName

CloudFront 接收要求的位置名稱。

請求

從每個位置 CloudFront 接收的要求數目。

RequestsPct

從每個位置 CloudFront 接收的要求數目，佔該期間從所有位置 CloudFront 接收到的要求總數的百分比。

TotalBytes

在指定的分佈和期間內，為此國家或州/省的檢視者 CloudFront 提供服務的位元組數目。

位置趨勢報告

該報告包含以下值：

DistributionID

為執行報告所分佈的 ID，如果您為所有分佈執行報告，則為 ALL。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

TimeBucket

適用於小時或天的資料，以國際標準時間 (UTC) 為準。

(位置)

報告中的其餘欄會列出從中 CloudFront 接收要求的位置。如需有關可能的值的詳細資訊，請參閱位置報表中的資料與CloudFront 標準記錄中的資料 (存取記錄) 之間的關聯的位置說明。

位置報表中的資料與CloudFront 標準記錄中的資料 (存取記錄) 之間的關聯

下列清單顯示 CloudFront 主控台中「位置」報表中的資料如何與 CloudFront 存取記錄中的值對應。如需 CloudFront 存取記錄的詳細資訊，請參閱設定和使用標準日誌 (存取日誌)。

位置

檢視器所在國家/地區或美國州。在存取日誌中，c-ip 欄位包含檢視器所正在執行之裝置的 IP 地址。我們使用地理定位資料來識別根據 IP 地址的裝置的地理位置。

如果您是依照國家/地區來顯示 Locations (位置) 報告，請注意該國家/地區的清單是根據 ISO 3166-2，國家及其地區名稱的表示代碼 – 第 2 部分：國家細分代碼。國家/地區清單包括以下附加值：

- 匿名代理 – 來自匿名代理的請求。
- 衛星服務供應商 – 來自於衛星服務供應商的請求，此等供應商提供網際網路服務給多個國家/區域。使用者可能處於詐騙風險很高的國家/地區。
- 歐洲 (未知) – 請求的原始伺服器 IP，位於多個歐洲國家/區域所使用的區塊中。無法決定要求來源的國家/地區。 CloudFront 使用「歐洲 (未知)」做為預設值。
- 亞太區域 (未知) – 請求的原始伺服器 IP，位於多個亞太區域國家/區域所使用的區塊中。無法決定要求來源的國家/地區。 CloudFront 使用「亞洲/太平洋 (未知)」做為預設值。

如果您顯示美國區域的 Locations (位置) 報告，請注意，該報告可能包括美國領土和美國兵力部署的地區。

> **Note**
>
> 如果 CloudFront 無法判斷使用者的位置，該位置會在檢視者報告中顯示為「未知」。

請求計數

在指定的分佈和期間，檢視器所在的國家/地區或美國州的請求總數。這個值通常與CloudFront 存取日誌中該國家/地區或州/地區的 IP 位址的GET要求數目密切相對應。

要求 %

下列其中一項，取決於您針對 Details (詳細資訊) 所選擇的值：

- 國家/區域 – 來自這個國家/區域的請求數量，佔請求總數的百分比。
- 美國 – 來自該州的請求占美國請求總數的百分比。

如果請求來自超過 50 個國家/地區，則無法根據此表格中的資料計算 Request % (請求百分比)，因為 Request Count (請求計數) 欄位並未包含在指定期間內的所有請求。

位元組

在指定的分佈和期間內，為此國家或州/省的檢視者 CloudFront 提供服務的位元組數目。若要將這個欄位中的資料顯示變更為 KB、MB 或 GB，請按一下在欄位標頭中的連結。

使用 Amazon CloudFront 監控指標 CloudWatch

Amazon CloudFront 與 Amazon 集成， CloudWatch 並自動發布分發和邊緣函數的操作指標（包括 Lambda @Edge 和 CloudFront 函數）。這些指標中有許多會顯示在CloudFront 主控台中的一組圖形中，也可以使用 CloudFront API 或 CLI 存取。所有這些指標都可在主CloudWatch 控台或透過 CloudWatch API 或 CLI 取得。 CloudFront 指標不會計入CloudWatch 配額 (先前稱為限制)，也不會產生任何額外費用。

除了 CloudFront 分佈的預設量度外，您還可以支付額外費用開啟其他量度。其他量度適用於 CloudFront 分佈，且必須分別為每個分佈開啟。如需成本的詳細資訊，請參閱the section called “估計其他 CloudFront 指標的成本”。

檢視這些指標可協助您疑難排解、追蹤和偵錯問題。若要在 CloudFront 主控台中檢視這些測量結果，請參閱監督頁面。若要檢視特定 CloudFront 分佈或邊緣函數之活動的相關圖形，請選擇其中一個，然後選擇 [檢視分佈量度] 或 [檢視量度]。

您也可以在主控台或 CloudFront 主控台、API 或 CLI 中根據這些指標設定警示 (適用標準 CloudWatch 定價)。 CloudWatch 例如，您可以根據 `5xxErrorRate` 指標設定警示，該指標代表回應的 HTTP 狀態碼在 `500` 到 `599` 範圍內之所有瀏覽者請求的百分比。當錯誤率在特定時間內達到特定值 (例如連續 5 分鐘達到 5% 的請求) 時，就會觸發警示。您可以在建立警示時指定警示的值及其時間單位。如需詳細資訊，請參閱 建立警示。

> **Note**
>
> 當您在 CloudFront 主控台中建立 CloudWatch 警示時，它會在美國東部 (維吉尼亞北部) 區域 (`us-east-1`) 為您建立警示。如果您從 CloudWatch 控制台創建警報，則必須使用相同的區域。由於 CloudFront 是全域服務，因此服務的指標會傳送至美國東部 (維吉尼亞北部)。

主題

- 檢視 CloudFront 和邊緣函數度量
- 建立 指標的 警示
- 以 CSV 格式下載指標資料
- 使用 CloudWatch API 取得指標

檢視 CloudFront 和邊緣函數度量

您可以在控制台中查看有關 CloudFront 分佈和邊緣功能的CloudFront 操作指標。若要檢視這些測量結果，請參閱 CloudFront 主控台中的「監督」頁面。若要檢視特定 CloudFront 分佈或邊緣函數之活動的相關圖形，請選擇其中一個，然後選擇 [檢視分佈量度] 或 [檢視量度]。

主題

- 檢視預設 CloudFront 分佈量度
- 開啟其他 CloudFront分佈量度
- 檢視預設 Lambda@Edge 函數指標
- 檢視預設的CloudFront 函數測量結果

檢視預設 CloudFront 分佈量度

下列預設量度適用於所有 CloudFront 發行版本，無須額外付費：

請求

所有 HTTP 方法以及 HTTP 和 HTTPS 要求所接收的檢視器要求總數。 CloudFront

下載的位元組數

瀏覽者執行 GET、HEAD 及 OPTIONS 請求時下載的位元組總數。

上傳的位元組數

檢視者使用CloudFront、使用POST和PUT要求上傳至您來源的位元組總數。

4xx 錯誤率

回應的 HTTP 狀態碼為 4xx 之所有瀏覽者請求的百分比。

5xx 錯誤率

回應的 HTTP 狀態碼為 5xx 之所有瀏覽者請求的百分比。

總錯誤率

回應的 HTTP 狀態碼為 4xx 或 5xx 之所有瀏覽者請求的百分比。

這些測量結果會以圖表形式顯示在 CloudFront主控台的 [監視] 頁面上的每個 CloudFront 分佈。在每個圖表上，總計會以 1 分鐘為間隔來顯示。除了檢視圖形之外，您也可以以 CSV 檔案格式下載指標報告。

您可以執行以下動作以自訂圖表：

- 若要變更顯示在圖表中資訊的時間範圍，請選擇 1h (1 小時)、3h (3 小時) 或其他範圍或指定自訂範圍。
- 若要變更更 CloudFront 新圖形中資訊的頻率，請選擇重新整理圖示旁邊的向下箭頭，然後選擇重新整理速率。預設重新整理速率為 1 分鐘，但您可以選擇 10 秒、2 分鐘或其他選項。

若要在 CloudWatch 主控台中檢視 CloudFront 圖形，請選擇 [新增至儀表板]。

開啟其他 CloudFront分佈量度

除了預設指標之外，您還可以開啟其他指標，但需要支付額外費用。如需成本的詳細資訊，請參閱the section called “估計其他 CloudFront 指標的成本”。

必須分別為每個分佈開啟這些額外指標：

快取命中率

為其 CloudFront 提供快取內容的所有可快取要求的百分比。HTTP POST 和 PUT 請求及錯誤不視為可快取請求。

來源延遲

從 CloudFront 接收請求到開始對網路 (非檢視者) 提供回應 (而非檢視者) 從來源提供要求 (而非 CloudFront 快取) 所花費的總時間。這也稱為第一個位元組延遲，或time-to-first-byte。

依狀態碼分類的錯誤率

其回應的 HTTP 狀態碼是 4xx 或 5xx 範圍中之特定程式碼的所有瀏覽者請求的百分比。此指標適用於下列所有錯誤碼：401、403、404、502、503 和 504。

開啟其他指標

您可以使用、使用 AWS Command Line Interface (AWS CLI) 或使用 AWS CloudFormation CloudFront API 在 CloudFront 主控台中開啟其他指標。

Console

開啟其他指標 (主控台)

1. 登入 AWS Management Console 並在主控台中開啟 [監視] 頁 CloudFront 面。
2. 選擇要開啟其他指標的分佈，然後選擇 View distribution metrics (檢視分佈指標)。
3. 選擇 Manage additional metrics (管理其他指標)。
4. 在 Manage additional metrics (管理其他指標) 視窗中，開啟 Enabled (已啟用)。開啟其他指標後，您可以關閉 Manage additional metrics (管理其他指標) 視窗。

開啟其他指標後，它們會顯示在圖表中。在每個圖表上，總計會以 1 分鐘為間隔來顯示。除了檢視圖形之外，您也可以以 CSV 檔案格式下載指標報告。

您可以執行以下動作以自訂圖表：

- 若要變更顯示在圖表中資訊的時間範圍，請選擇 1h (1 小時)、3h (3 小時) 或其他範圍或指定自訂範圍。
- 若要變更更 CloudFront 新圖形中資訊的頻率，請選擇重新整理圖示旁邊的向下箭頭，然後選擇重新整理速率。預設重新整理速率為 1 分鐘，但您可以選擇 10 秒、2 分鐘或其他選項。

若要在 CloudWatch 主控台中檢視 CloudFront 圖形，請選擇 [新增至儀表板]。

AWS CloudFormation

若要使用開啟其他量度 AWS CloudFormation，請使用`AWS::CloudFront::MonitoringSubscription`資源類型。下列範例顯示用於啟用其他量度的 YAML 格式的範 AWS CloudFormation 本語法。

```
Type: AWS::CloudFront::MonitoringSubscription
Properties:
  DistributionId: EDFDVBD6EXAMPLE
  MonitoringSubscription:
    RealtimeMetricsSubscriptionConfig:
      RealtimeMetricsSubscriptionStatus: Enabled
```

CLI

若要使用 AWS Command Line Interface (AWS CLI) 管理其他量度，請使用下列其中一個指令：

開啟分佈的其他指標 (CLI)

- 使用 create-monitoring-subscription 命令，如下列範例所示。將 *EDFDVBD6EXAMPLE* 取代為您要啟用其他指標的分佈 ID。

```
aws cloudfront create-monitoring-subscription --
distribution-id EDFDVBD6EXAMPLE --monitoring-subscription
 RealtimeMetricsSubscriptionConfig={RealtimeMetricsSubscriptionStatus=Enabled}
```

查看是否已針對分佈開啟其他指標 (CLI)

- 使用 get-monitoring-subscription 命令，如下列範例所示。將 *EDFDVBD6EXAMPLE* 取代為您正在檢查的分佈 ID。

```
aws cloudfront get-monitoring-subscription --distribution-id EDFDVBD6EXAMPLE
```

關閉分佈的其他指標 (CLI)

- 使用 delete-monitoring-subscription 命令，如下列範例所示。將 *EDFDVBD6EXAMPLE* 取代為您要關閉其他指標的分佈 ID。

```
aws cloudfront delete-monitoring-subscription --distribution-id EDFDVBD6EXAMPLE
```

API

若要使用 CloudFront API 管理其他指標，請使用下列其中一個 API 作業。

- 若要開啟發佈的其他量度，請使用CreateMonitoringSubscription。
- 若要查看是否針對發佈開啟其他量度，請使用GetMonitoringSubscription。
- 若要關閉發佈的其他量度，請使用DeleteMonitoringSubscription。

如需這些 API 呼叫的詳細資訊，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

估計其他 CloudFront 指標的成本

當您開啟分發的其他量度時，最多 CloudFront 會傳送 8 個指標至 CloudWatch 美國東部 (維吉尼亞北部) 區域。 CloudWatch 針對每個量度收取較低的固定費率。此費率每個指標每月僅收取一次 (每個分佈最多 8 個指標)。這是固定費率，因此無論 CloudFront 分配接收或發送的請求或響應數量如何，您的費用都保持不變。有關每個指標費率，請參閱 Amazon 定 CloudWatch 價頁面和定CloudWatch 價計算器。使用 API 擷取指標時，需支付額外的 CloudWatch API 費用。

檢視預設 Lambda@Edge 函數指標

您可以使用 CloudWatch 指標來即時監控 Lambda @Edge 函數的問題。使用這些指標無需負擔額外費用。

當您將 Lambda @Edge 函數附加至 CloudFront 分發中的快取行為時，Lambda 會開始 CloudWatch 自動將指標傳送至。指標適用於所有 Lambda 區域，但若要在 CloudWatch 主控台中檢視指標或從 CloudWatch API 取得指標資料，您必須使用美國東部 (維吉尼亞北部) 區域 (`us-east-1`)。指標群組名稱的格式為:`AWS/CloudFront/`*`distribution-ID`*，其中`#####`是與 Lambda @Edge 函數相關聯之 CloudFront 分佈的識別碼。如需有關指 CloudWatch標的詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

下列預設指標會顯示在主控台的 [監 CloudFront 控] 頁面上的每個 Lambda @Edge 函數的圖形中：

- 5xxLambda@Edge 的 錯誤率
- Lambda 執行錯誤
- Lambda 無效回應
- Lambda 節流器

每個圖表都包含了呼叫、錯誤、調節等資訊。在每個圖表上，總計會以 1 分鐘的粒度顯示，並依「 AWS 區域」分組。

如果您看到想要調查的錯誤激增，您可以選擇一個函數，然後按 AWS 地區檢視記錄檔，直到您判斷哪個函數造成問題，以及在哪個 AWS 區域。如需針對 Lambda@Edge 錯誤進行故障診斷的詳細資訊，請參閱：

- the section called “如何判斷故障的類型”
- 調試內容交付的四個步驟 AWS

您可以執行以下動作以自訂圖表：

- 若要變更顯示在圖表中資訊的時間範圍，請選擇 1h (1 小時)、3h (3 小時) 或其他範圍或指定自訂範圍。
- 若要變更更 CloudFront 新圖形中資訊的頻率，請選擇重新整理圖示旁邊的向下箭頭，然後選擇重新整理速率。預設重新整理速率為 1 分鐘，但您可以選擇 10 秒、2 分鐘或其他選項。

若要在 CloudWatch 主控台中檢視圖形，請選擇 [新增至儀表板]。您必須使用美國東部 (維吉尼亞北部) 區域 (us-east-1) 在主控台中檢視圖形。 CloudWatch

檢視預設的CloudFront 函數測量結果

CloudFront 函數會將操作指標傳送至 Amazon，以 CloudWatch 便您監控您的功能。檢視這些指標可協助您疑難排解、追蹤和偵錯問題。CloudFront 函數會將下列量度發佈至 CloudWatch：

- 叫用 (FunctionInvocations) – 在給定時間段內啟動 (叫用) 函數的次數。
- 驗證錯誤 (FunctionValidationErrors) – 函數在給定時間段內產生的驗證錯誤數。函數成功執行但返回無效的資料 (無效的 事件物件) 時，就會發生驗證錯誤。
- 執行錯誤 (FunctionExecutionErrors) – 在給定時間段內發生的執行錯誤次數。當函數無法成功完成時，就會發生執行錯誤。

- 運用利用率(FunctionComputeUtilization) – 執行函數所花費的時間，以所允許時間上限的百分比表示。例如，35 的值表示該函數以所允許時間上限的 35% 完成。此指標是介於 0 到 100 之間的數字。

  如果此值達到或接近 100，則函數已使用或接近使用允許的執行時間，並且可能會限制後續要求。如果您的函數以 80% 以上的使用率執行，建議您檢閱函數以縮短執行時間並提高使用率。例如，您可能只想記錄錯誤、簡化任何複雜的 regex 運算式，或移除不必要的複雜 JSON 物件剖析。
- 調節 (FunctionThrottles) – 在特定期間內調節函數的次數。可能會因下列原因而對函數進行調節：
  - 函數連續超過允許執行的最長時間
  - 函數導致編譯錯誤
  - 每秒請求數異常高

CloudFront KeyValueStore 還會將以下操作指標發送給 Amazon CloudWatch：

- 讀取要求 (KvsReadRequests) — 函數在指定時間段內成功從索引鍵值存放區讀取的次數。
- Read errors (KvsReadErrors) — 函數在指定時間段內從索引鍵值存放區讀取失敗的次數。

若要在 CloudFront 主控台中檢視這些測量結果，請移至「監督」頁面。若要檢視特定函數的圖表，請選擇函數，選取該函數，然後選擇檢視函數指標。

所有這些量度都會發佈至 CloudFront 命名空間 CloudWatch 中的美國東部 (維吉尼亞北部us-east-1) 區域 ()。您也可以在 CloudWatch 主控台中檢視這些指標。在 CloudWatch 控制台中，您可以查看每個函數或每個分佈每個函數的指標。

您也可以使用 CloudWatch 根據這些指標設定警示。例如，您可以根據執行時間 (FunctionComputeUilization) 指標來設定警示，這代表函數執行所花費的可用時間百分比。當執行時間在特定時間內達到特定值 (例如，連續 15 分鐘的可用時間超過 70%) 時，就會觸發警示。您可以在建立警示時指定警示的值及其時間單位。

> **Note**
>
> CloudFront 函數 CloudWatch 只會針對LIVE階段中為回應生產要求和回應而執行的函數傳送量度。當您測試函數時， CloudFront不會將任何指標傳送至 CloudWatch。測試輸出包含有關錯誤、計算使用率和函數記錄檔 (console.log()陳述式) 的資訊，但這項資訊不會傳送至 CloudWatch。

如需如何使用 CloudWatch API 取得這些指標的相關資訊，請參閱the section called “使用 API 取得指標”。

建立 指標的 警示

在 CloudFront 主控台中，您可以設定警示，以根據特定 CloudFront 指標透過 Amazon 簡單通知服務 (Amazon SNS) 通知您。您可以在 CloudFront 控制台的 [警報] 頁面上設定警報。

若要在主控台中建立警示，請指定下列值：

指標

您要建立警示的指標。

發佈

您要為其建立警示的 CloudFront 分配。

Name of alarm (警示的名稱)

警示的名稱。

傳送通知到

當此指標觸發警示時，要傳送通知的目標 Amazon SNS 主題。

每當 ***<metric> <operator> <value>***

指定何時 CloudWatch 應觸發警示，並傳送通知給 Amazon SNS 主題。例如，要在 5xx 錯誤率超過 1% 時收到通知，請指定以下內容：

每當平均 5 xxErrorRate **>** **1**

請注意以下有關指定值的下列事項：

- 僅輸入沒有標點符號的整數。例如，若要指定一千，請輸入 **1000**。
- 對於 4xx、5xx 和總錯誤率，您指定的值為百分比。
- 對於請求、下載的位元組和上傳的位元組，您指定的值是單位。例如，1073742000 個位元組。

至少 ***<time period>*** 的 ***<number>*** 個連續期間

指定指標在指定持續時間內必須符合準則的連續時間段數，才會 CloudWatch 觸發警示。當您選擇值時，請盡量在值之間取得適當平衡，該值不會針對暫時或短暫的問題發出警示，但會針對持續或實際問題發出警示。

以 CSV 格式下載指標資料

您可以下載 CSV 格式的分 CloudFront 佈 CloudWatch 量度資料。您可以在CloudFront主控台中檢視特定發佈的分佈指標時下載資料。

有關報告的資訊

報告的前幾行包含以下資訊：

版本

CloudFront 報告版本。

報告

報告名稱。

DistributionID

您執行報表的分佈 ID。

StartDateUTC

您所執行報告日期範圍的開始時間，是以國際標準時間 (UTC) 為準。

EndDateUTC

您執行報告的日期範圍的結束時間，以國際標準時間 (UTC) 為準。

GeneratedTimeUTC

您執行報告的日期和時間，以國際標準時間 (UTC) 為準。

精細程度

在報告中每行的時間段，例如 `ONE_MINUTE`。

指標報告中的資料

該報告包含以下值：

DistributionID

您執行報表的分佈 ID。

FriendlyName

分佈的備用網域名稱 (CNAME)，如果有的話。如果分佈沒有備用網域名稱，則該清單包括分佈的原始網域名稱。

TimeBucket

適用於小時或天的資料，以國際標準時間 (UTC) 為準。

請求

在這段期間，所有 HTTP 狀態碼 (例如 200、404 等) 和所有方法 (例如 GET、HEAD、POST 等) 的請求總數。

BytesDownloaded

瀏覽者在這段期間為指定分佈下載的位元組數。

BytesUploaded

瀏覽者在這段期間為指定分佈上傳的位元組數。

TotalErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 4xx 或 5xx 錯誤的請求百分比。。

xxErrorRate百分之四

在這段期間，對於指定分佈，HTTP 狀態碼為 4xx 錯誤的請求百分比。

xxErrorRate百分之五

在這段期間，對於指定分佈，HTTP 狀態碼為 5xx 錯誤的請求百分比。

如果您已為分佈開啟其他指標，則報表也會包含下列額外值：

401 ErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 401 錯誤的請求百分比。

403 ErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 403 錯誤的請求百分比。

404 ErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 404 錯誤的請求百分比。

502 ErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 502 錯誤的請求百分比。

503 ErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 503 錯誤的請求百分比。

504 ErrorRatePct

在這段期間，對於指定分佈，HTTP 狀態碼為 504 錯誤的請求百分比。

OriginLatency

從 CloudFront 收到要求到開始對網路 (非檢視者) 提供回應的時間總計 (以毫秒為單位)，以毫秒為單 CloudFront 位。這也稱為第一個位元組延遲，或time-to-first-byte。

CacheHitRate

為其 CloudFront 提供快取內容的所有可快取要求的百分比。HTTP POST 和 PUT 請求及錯誤不視為可快取請求。

使用 CloudWatch API 取得指標

您可以使用 Amazon CloudWatch API 或 CLI 在您建立的程式或應用程式中取得 CloudFront 指標。您可以使用原始資料來建置自己的自訂儀表板、您自己的警示工具等等。

若要從 CloudWatch API 取得 CloudFront 指標，您必須使用美國東部 (維吉尼亞北部) 區域 (us-east-1)。您還必須知道每個指標的特定值和類型。

主題

所有 CloudFront 量度的值

下列值適用於所有 CloudFront 測量結果：

Namespace

Namespace 的值永遠為 AWS/CloudFront。

維度

每個 CloudFront 量度都有以下兩個維度：

`DistributionId`

您要取得量度的 CloudFront 分佈識別碼。

`FunctionName`

您要取得量度之函數的名稱 (在 CloudFront 函數中)。

此維度僅適用於函數。

`Region`

的價值始終`Region`是`Global`，因為 CloudFront 是全球性的服務。

> **Note**
> 若要從 CloudWatch API 取得 CloudFront 指標，您必須使用美國東部 (維吉尼亞北部) 區域 (us-east-1)。

CloudFront分佈量度的值

使用下列清單中的資訊，從 CloudWatch API 取得有關特定 CloudFront 發佈指標的詳細資料。其中一些指標只有在您已為分佈開啟其他指標時才能使用。

> **Note**
> 每個指標僅適用一個統計資料 (`Average` 或 `Sum`)。下列清單指定適用於該指標的統計資料。

4xx 錯誤率

回應的 HTTP 狀態碼為 `4xx` 之所有瀏覽者請求的百分比。

- 指標名稱：`4xxErrorRate`
- 有效統計資訊：`Average`
- 單位：`Percent`

401 錯誤率

回應的 HTTP 狀態碼為 401 之所有瀏覽者請求的百分比。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：401ErrorRate
- 有效統計資訊：Average
- 單位：Percent

403 錯誤率

回應的 HTTP 狀態碼為 403 之所有瀏覽者請求的百分比。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：403ErrorRate
- 有效統計資訊：Average
- 單位：Percent

404 錯誤率

回應的 HTTP 狀態碼為 404 之所有瀏覽者請求的百分比。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：404ErrorRate
- 有效統計資訊：Average
- 單位：Percent

5xx 錯誤率

回應的 HTTP 狀態碼為 5xx 之所有瀏覽者請求的百分比。

- 指標名稱：5xxErrorRate
- 有效統計資訊：Average
- 單位：Percent

502 錯誤率

回應的 HTTP 狀態碼為 502 之所有瀏覽者請求的百分比。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：502ErrorRate
- 有效統計資訊：Average
- 單位：Percent

503 錯誤率

回應的 HTTP 狀態碼為 503 之所有瀏覽者請求的百分比。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：5O3ErrorRate
- 有效統計資訊：Average
- 單位：Percent

504 錯誤率

回應的 HTTP 狀態碼為 504 之所有瀏覽者請求的百分比。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：5O4ErrorRate
- 有效統計資訊：Average
- 單位：Percent

下載的位元組數

瀏覽者執行 GET、HEAD 及 OPTIONS 請求時下載的位元組總數。

- 指標名稱：BytesDownloaded
- 有效統計資訊：Sum
- 單位：None

上傳的位元組數

檢視者使用CloudFront、使用POST和PUT要求上傳至您來源的位元組總數。

- 指標名稱：BytesUploaded
- 有效統計資訊：Sum
- 單位：None

快取命中率

為其 CloudFront 提供快取內容的所有可快取要求的百分比。HTTP POST 和 PUT 請求及錯誤不視為可快取請求。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：CacheHitRate
- 有效統計資訊：Average
- 單位：Percent

來源延遲

從 CloudFront 接收要求到開始對網路 (非檢視者) 提供回應 (而非檢視者) 的要求 (而非 CloudFront 快取) 所花費的總時間 (以毫秒為單位)。這也稱為第一個位元組延遲，或time-to-first-byte。若要取得此指標，您必須先開啟其他指標。

- 指標名稱：OriginLatency
- 有效統計資訊：Percentile
- 單位：Milliseconds

> **Note**
>
> 要從 CloudWatch API 獲取Percentile統計信息，請使用ExtendedStatistics參數，而不是Statistics。如需詳細資訊，請參閱 GetMetricStatisticsAmazon CloudWatch API 參考或AWS 開發套件的參考文件。

請求

所有 HTTP 方法以及 HTTP 和 HTTPS 要求所接收的檢視器要求總數。 CloudFront

- 指標名稱：Requests
- 有效統計資訊：Sum
- 單位：None

總錯誤率

回應的 HTTP 狀態碼為 4xx 或 5xx 之所有瀏覽者請求的百分比。

- 指標名稱：TotalErrorRate
- 有效統計資訊：Average
- 單位：Percent

CloudFront 函數量度量的值

使用下列清單中的資訊，從 CloudWatch API 取得有關特定 CloudFront函數指標的詳細資訊。

> **Note**
>
> 每個指標僅適用一個統計資料 (Average 或 Sum)。下列清單指定適用於該指標的統計資料。

呼叫

在給定時間段內啟動 (叫用) 函數的次數。

- 指標名稱：FunctionInvocations
- 有效統計資訊：Sum
- 單位：None

驗證錯誤

函數在給定時間段內產生的驗證錯誤數。函數成功執行但返回無效的資料 (無效的事件物件) 時，就會發生驗證錯誤。

- 指標名稱：FunctionValidationErrors
- 有效統計資訊：Sum
- 單位：None

執行錯誤

在給定時間段內發生的執行錯誤次數。當函數無法成功完成時，就會發生執行錯誤。

- 指標名稱：FunctionExecutionErrors
- 有效統計資訊：Sum
- 單位：None

運算利用率

執行函數所花費的時間 (0-100)，以所允許時間上限的百分比表示。例如，35 的值表示該函數以所允許時間上限的 35% 完成。

- 指標名稱：FunctionComputeUtilization
- 有效統計資訊：Average
- 單位：Percent

限流

在特定期間內調節函數的次數。

- 指標名稱：FunctionThrottles
- 有效統計資訊：Sum
- 單位：None

CloudFront 和邊緣功能記錄

Amazon CloudFront 提供不同類型的日誌記錄。您可以記錄來自您的 CloudFront 發行版的檢視者請求，也可以在您的 AWS 帳戶中記錄 CloudFront 服務活動 (API 活動)。您也可以從邊緣運算函數取得日誌。

記錄請求

CloudFront 提供了以下方法來記錄發送到您的發行版的請求。

標準日誌 (存取日誌)

CloudFront 標準記錄檔會提供對散發的每個要求的詳細記錄。這些日誌適用於許多案例，包括安全性和存取稽核。

CloudFront 標準日誌會傳送到您選擇的 Amazon S3 儲存貯體。 CloudFront不過存放和存取日誌檔時會產生 Amazon S3 費用，但不會收取標準日誌費用。

如需詳細資訊，請參閱 使用標準日誌 (存取日誌)。

即時日誌

CloudFront 即時記錄會即時提供對分發要求的相關資訊 (記錄記錄會在收到要求後的幾秒內傳送)。您可以為即時日誌選擇抽樣率，即要接收即時日誌記錄的請求百分比。此外，您還可以選擇要在日誌記錄中接收的特定欄位。

CloudFront 即時日誌會在 Amazon Kinesis 資料串流中傳送至您選擇的資料串流。 CloudFront 除了使用 Kinesis Data Streams 所產生的費用外，即時記錄的費用也會產生費用。

如需詳細資訊，請參閱 即時日誌。

記錄邊緣函數

您可以使用 Amazon CloudWatch 日誌來獲取邊緣函數的日誌，包括 Lambda @Edge 和 CloudFront 函數。您可以使用 CloudWatch 控制台或日 CloudWatch 誌 API 訪問日誌。如需詳細資訊，請參閱 the section called “邊緣函數日誌”。

記錄服務活動

您可以使 AWS CloudTrail 用在 AWS 帳戶中記錄 CloudFront 服務活動（API 活動）。 CloudTrail 提供使用者、角色或 AWS 服務所採取之 API 動作的記錄 CloudFront。使用收集的資訊 CloudTrail，您可

以判斷向其發出的 API 要求 CloudFront、提出要求的 IP 位址、提出要求的人員、提出要求的時間以及其他詳細資訊。

如需更多詳細資訊，請參閱 使用記錄 Amazon CloudFront API 呼叫 AWS CloudTrail。

主題

- 設定和使用標準日誌 (存取日誌)
- 即時日誌
- 邊緣函數日誌
- 使用記錄 Amazon CloudFront API 呼叫 AWS CloudTrail

設定和使用標準日誌 (存取日誌)

您可以設定 CloudFront 為建立記錄檔，其中包含有關 CloudFront 接收之每個使用者要求的詳細資訊。這些稱為 標準日誌，也稱為存取日誌。如果啟用標準日誌，也可以指定要 CloudFront 在其中儲存檔案的 Amazon S3 儲存貯體。

您可以在建立或更新分佈時啟用標準日誌。如需詳細資訊，請參閱 發佈設定參考。

CloudFront 也提供即時記錄，提供即時對分發的要求的相關資訊 (記錄會在收到要求後的幾秒內傳送)。您可以使用即時日誌來監控、分析並根據內容交付效能採取動作。如需詳細資訊，請參閱 即時日誌。

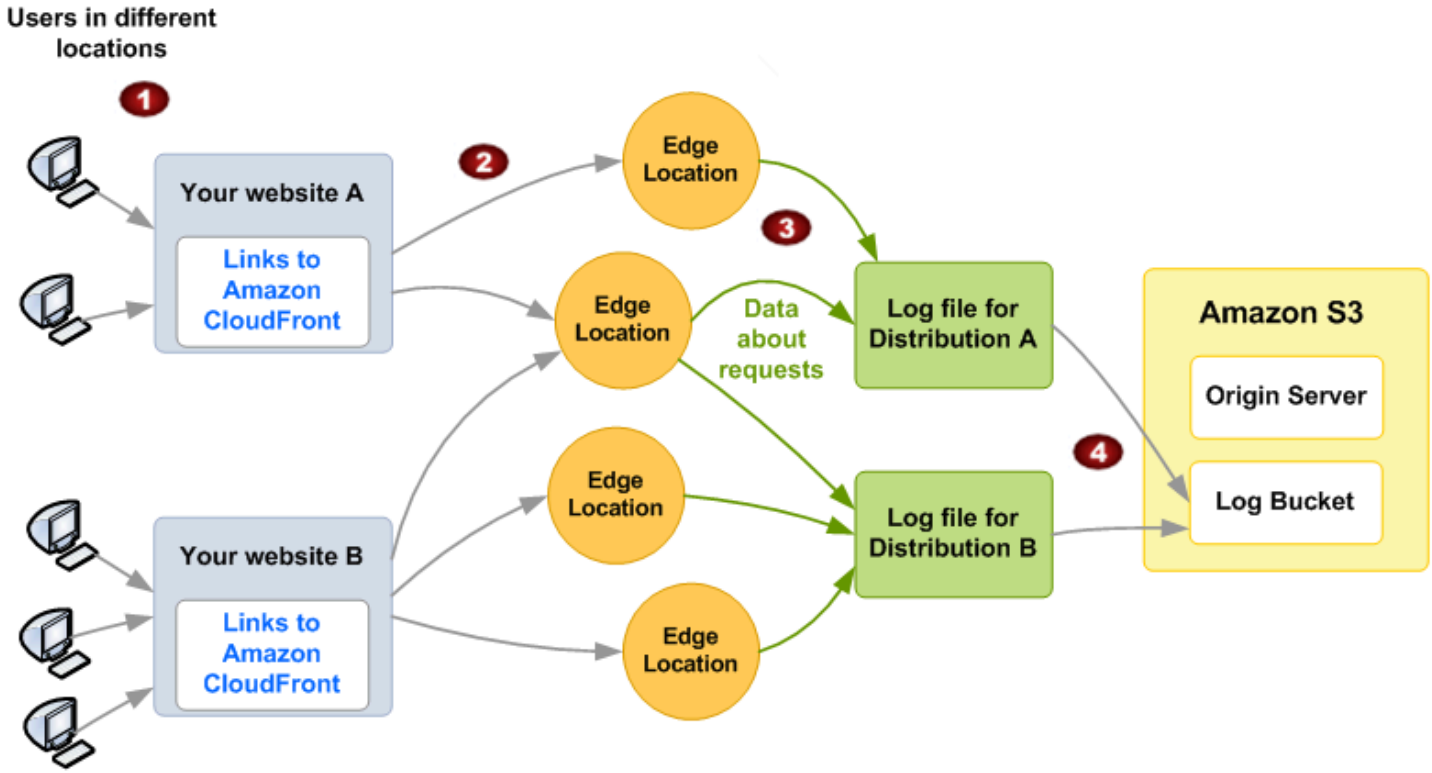
主題

- 標準記錄的運作方式
- 為您的標準日誌選擇 Amazon S3 儲存貯體
- 設定標準記錄和存取日誌檔案所需的許可
- 適用於 SSE-KMS 儲存貯體的必要金鑰政策
- 檔案名稱格式
- 標準日誌檔案交付的時間
- 請求 URL 或標頭超過大小上限時，記錄請求的方式
- 分析標準日誌
- 編輯標準記錄設定
- 從 Amazon S3 儲存貯體中刪除標準日誌檔案

- 標準日誌檔案格式
- 標準日誌的費用

標準記錄的運作方式

下圖顯示有關物件要求的 CloudFront 記錄資訊的方式。



以下說明如何 CloudFront 記錄物件請求的相關資訊，如上圖所示。

1. 在此圖中，您有兩個網站 A 和 B，以及兩個對應的 CloudFront 發行版。使用者使用與分佈相關聯的 URL 請求物件。
2. CloudFront 將每個請求路由到適當的邊緣位置。
3. CloudFront 將有關每個請求的數據寫入特定於該分發的日誌文件中。在這個範例中，有關與分佈 A 相關的請求資訊只會進入分佈 A 的日誌檔案，且有關與分佈 B 相關的請求資訊只會進入分佈 B 的日誌檔案。
4. CloudFront 定期將分發的日誌檔儲存在您啟用記錄時指定的 Amazon S3 儲存貯體中。CloudFront 然後開始將有關後續請求的信息保存在新的日誌文件中以進行分發。

如果在指定的某小時內沒有使用者存取您的內容，您不會收到該小時的任何日誌檔案。

每個日誌檔案中的項目提供單一請求的詳細資訊。如需關於日誌檔案格式的詳細資訊，請參閱標準日誌檔案格式。

> **Note**
>
> 我們建議您使用這些記錄來瞭解內容要求的性質，而不是對所有要求進行完整記錄。CloudFront 盡最大努力提供存取記錄。在實際處理請求之後，才可能長時間交付特定請求的日誌項目，在極少數的情況下，有可能完全不會交付日誌項目。當存取記錄中省略記錄項目時，存取記錄檔中的項目數量將與 AWS 帳單和使用情況報告中顯示的使用量不符。

為您的標準日誌選擇 Amazon S3 儲存貯體

啟用分發的記錄功能時，您可 CloudFront以指定要在其中存放日誌檔的 Amazon S3 儲存貯體。如果您是使用 Amazon S3 做為原始伺服器，我們建議不要為您的日誌檔案使用相同的儲存貯體；使用個別的儲存貯體簡化維護。

> **Important**
>
> 不要選擇 S3 物件擁有權設定為強制執行儲存貯體擁有者的 Amazon S3 儲存貯體。該設定會停用值區及其中物件的 ACL，以防 CloudFront止將記錄檔傳送至值區。

> **Important**
>
> 請勿在下列任何區域中選擇 Amazon S3 儲存貯體，因為 CloudFront 不會將標準日誌傳遞到這些區域的儲存貯體：
>
> - 非洲 (開普敦)
> - 亞太區域 (香港)
> - 亞太區域 (海德拉巴)
> - 亞太區域 (雅加達)
> - 亞太區域 (墨爾本)
> - 加拿大西部 (卡加利)
> - 歐洲 (米蘭)
> - 歐洲 (西班牙)
> - 歐洲 (蘇黎世)

- 以色列 (特拉維夫)
- Middle East (Bahrain)
- 中東 (阿拉伯聯合大公國)

您可以在同一個儲存貯體存放多個分佈的日誌檔案。啟用記錄時，您可以指定檔案名稱的選用字首，以便您可以繼續追蹤哪個日誌檔案與那個分佈關聯。

設定標準記錄和存取日誌檔案所需的許可

> **Important**
>
> 從 2023 年 4 月開始，您將需要為用於 CloudFront 標準日誌的新 S3 儲存貯體啟用 S3 存取控制清單 (ACL)。ACL 可以在儲存貯體建立步驟期間或建立儲存貯體之後啟用。
> 有關變更的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的新 S3 儲存貯體的預設設定常見問答集和 AWS 新聞部落格中的 Heads-Up: Amazon S3 Security Changes Are Coming in April of 2023。

對於您為記錄檔指定的值區，您的 AWS 帳戶必須具備下列權限：

- 儲存貯體的 S3 存取控制清單 (ACL) 必須授予您 `FULL_CONTROL`。如果您是儲存貯體擁有者，您的帳戶會有預設的該許可。如果您不是，儲存貯體擁有者則必須更新儲存貯體的 ACL。
- `s3:GetBucketAcl`
- `s3:PutBucketAcl`

注意下列事項：

儲存貯體的 ACL

當您建立或更新發行版並啟用記錄時， CloudFront 會使用這些權限來更新值區的 ACL，以授與`awslogsdelivery`帳戶`FULL_CONTROL`權限。`awslogsdelivery` 帳戶將日誌檔案寫入儲存貯體。如果您的帳戶沒有更新 ACL 所需的許可，則建立或更新分佈會失敗。

在某些情況下，如果您以程式設計的方式提交請求建立儲存貯體，但有指定名稱的儲存貯體已存在，則 S3 在儲存貯體上重設許可為預設值。如果您設定將存 CloudFront 取日誌儲存在 S3 儲存貯體中，而您停止在該儲存貯體中取得日誌，請檢查儲存貯體的許可，以確保 CloudFront 具有必要的權限。

還原儲存貯體的 ACL

如果您移除awslogsdelivery帳戶的許可，將CloudFront 無法將日誌儲存到 S3 儲存貯體。要啟用 CloudFront 以再次開始保存分發的日誌，請執行以下操作之一來恢復 ACL 權限：

- 禁用分發的日誌記錄 CloudFront，然後再次啟用它。如需詳細資訊，請參閱 發佈設定參考。
- 在 Amazon S3 主控台中瀏覽至 S3 儲存貯體，接著新增許可，即可為 awslogsdelivery 手動新增 ACL 許可。若要為 awslogsdelivery 新增 ACL，您必須提供該帳戶的正式 ID，如下所示：

  c4c1ede66af53448b93c283ce9448c4ba468c9432aa01d700d3878632f77d2d0

  如需將 ACL 新增至 S3 儲存貯體的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的如何設定 ACL 儲存貯體許可？。

每個日誌檔案的 ACL

除了儲存貯體上的 ACL，每個日誌檔案上都有一個 ACL。每個日誌檔案上儲存貯體擁有者擁有 FULL_CONTROL 許可，分佈擁有者 (如果與儲存貯體擁有者不同) 沒有許可，且 awslogsdelivery 帳戶擁有讀取和寫入許可。

停用記錄

如果停用記錄功 CloudFront 能，請勿刪除值區或記錄檔的 ACL。如有需要，您也可以自己執行該操作。

適用於 SSE-KMS 儲存貯體的必要金鑰政策

如果標準日誌的 S3 儲存貯體透過客戶受管金鑰來使用 AWS KMS keys (SSE-KMS) 實現伺服器端加密，則必須將下列陳述式新增至客戶受管金鑰的金鑰政策。這允許 CloudFront 將日誌文件寫入存儲桶。您無法搭配使用 SSE-KMS， AWS 受管金鑰 因為 CloudFront 無法將記錄檔寫入值區。)

```
{
    "Sid": "Allow CloudFront to use the key to deliver logs",
    "Effect": "Allow",
    "Principal": {
        "Service": "delivery.logs.amazonaws.com"
    },
    "Action": "kms:GenerateDataKey*",
    "Resource": "*"
}
```

如果標準日誌的 S3 儲存貯體使用 SSE-KMS 搭配 S3 儲存貯體金鑰，您還需要將 kms:Decrypt 權限新增至政策陳述式。在此情況下，完整的政策陳述式如下所示。

```
{
    "Sid": "Allow CloudFront to use the key to deliver logs",
    "Effect": "Allow",
    "Principal": {
        "Service": "delivery.logs.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

檔案名稱格式

儲存在 Amazon S3 CloudFront 儲存貯體中的每個日誌檔案名稱使用下列檔案名稱格式：

<optional prefix>/*<distribution ID>*.*YYYY*-*MM*-*DD*-*HH*.*unique-ID*.gz

使用國際標準時間 (UTC) 的日期與時間。

例如，如果您使用 example-prefix 做為前綴，且您的分佈 ID 為 EMLARXS9EXAMPLE，則檔案名稱看起來像這樣：

example-prefix/EMLARXS9EXAMPLE.2019-11-14-20.RT4KCN4SGK9.gz

啟用分佈的記錄時，您可以指定檔案名稱的選用字首，以便您可以繼續追蹤哪個日誌檔案與那個分佈關聯。如果您包含記錄檔前置詞的值，且前置詞的結尾不是正斜線 (/)，則會自動 CloudFront 附加一個值。如果您的前綴確實以正斜杠結尾，則 CloudFront 不要添加另一個。

檔案名稱末尾的表示 CloudFront 已使用 gzip 壓縮記錄檔。.gz

標準日誌檔案交付的時間

CloudFront 每小時最多可提供數次發佈的標準記錄。一般而言，記錄檔包含指定期間內 CloudFront 收到之要求的相關資訊。 CloudFront 通常會在日誌中出現的事件發生後一小時內，將該期間的日誌檔交付到 Amazon S3 儲存貯體。不過，請注意，一個時段的部分或全部日誌檔案項目有時會延遲高達 24 小時。延遲記錄項目時，會 CloudFront 將它們儲存在記錄檔中，檔案名稱包含要求發生的期間的日期和時間，而不是傳送檔案的日期和時間。

建立記錄檔時，會在記錄檔涵蓋的期間內，從收到物件要求的所有邊緣位置 CloudFront 合併發佈的資訊。

CloudFront 可以在一段時間內保存多個文件，具體取決於 CloudFront 收到與發布相關聯的對象的請求數量。

CloudFront 啟用記錄大約四小時後，便會開始可靠地傳遞存取記錄。該時段之前，您可能會收到幾個存取日誌。

> **Note**
>
> 如果在時段內沒有使用者請求您的物件，您便不會收到該時段的任何日誌檔案。

CloudFront 也提供即時記錄，提供即時對分發的要求的相關資訊 (記錄會在收到要求後的幾秒內傳送)。您可以使用即時日誌來監控、分析並根據內容交付效能採取動作。如需詳細資訊，請參閱 即時日誌。

請求 URL 或標頭超過大小上限時，記錄請求的方式

如果所有要求標頭 (包括 Cookie) 的總大小超過 20 KB，或者 URL 超過 8192 個位元組，則 CloudFront 無法完全剖析要求，也無法記錄要求。因為沒有記錄請求，您將無法在日誌檔案中看到傳回的 HTTP 錯誤狀態代碼。

如果要求主體超過大小上限，則會記錄請求，包含 HTTP 錯誤狀態代碼。

分析標準日誌

由於您每個小時可以收到多個存取日誌，我們建議您將指定期間內收到的所有日誌檔案整合成一個檔案。然後，您可以更準確且完整地分析該期間的資料。

使用 Amazon Athena 是分析存取日誌的其中一種方法。Athena 是一項互動式查詢服務，可協助您分析 AWS 服務的資料，包括 CloudFront. 若要進一步了解，請參閱 Amazon Athena 使用者指南中的查詢亞馬遜 CloudFront 日誌。

此外，下面的 AWS 博客文章討論了一些分析訪問日誌的方法。

- Amazon CloudFront 請求記錄 (適用於透過 HTTP 傳送的內容)
- 增強型 CloudFront 日誌，現在使用查詢字串

> **Important**
>
> 我們建議您使用這些記錄來瞭解內容要求的性質，而不是對所有要求進行完整記錄。CloudFront 盡最大努力提供存取記錄。在實際處理請求之後，才可能長時間交付特定請求的日誌項目，在極少數的情況下，有可能完全不會交付日誌項目。當存取記錄中省略記錄項目時，存取記錄檔中的項目數量將與使用量和帳單報告中顯示的 AWS 使用量不符。

編輯標準記錄設定

您可以使用CloudFront 主控台或 CloudFront API 啟用或停用記錄、變更存放日誌的 Amazon S3 儲存貯體，以及變更日誌檔的前置詞。記錄設定的變更在 12 小時內生效。

如需詳細資訊，請參閱下列主題：

- 若要使用 CloudFront 主控台更新發行版本，請參閱更新分佈。
- 若要使用 CloudFront API 更新分發，請參閱 Amazon CloudFront API 參考UpdateDistribution中的。

從 Amazon S3 儲存貯體中刪除標準日誌檔案

CloudFront 不會自動從您的 Amazon S3 儲存貯體刪除日誌檔。如需從 Amazon S3 儲存貯體將日誌檔案刪除的詳細資訊，請參閱下列主題：

- 使用 Amazon S3 主控台：在 Amazon Simple Storage Service 主控台使用者指南中的刪除物件。
- 使用 REST API：DeleteObject在 Amazon 簡單存儲服務 API 參考中。

標準日誌檔案格式

日誌檔案中的每個項目會提供有關單一檢視器請求的詳細資訊。日誌檔案具有下列特性：

- 使用 W3C 延伸日誌檔案格式。
- 包含索引標籤分隔值。
- 包含不一定按時間順序排列的記錄。
- 包含兩個標頭行：一個有檔案格式版本，而另一個則列出包含在每個記錄的 W3C 欄位。
- 在欄位值中包含空格和特定其他字元的 URL 編碼對等字元。

  以下字元會使用 URL 編碼的對等字元：

- ASCII 字元碼 0 到 32 (含)
- ASCII 字元碼 127 及更高
- 下表中的所有字元

URL 編碼標準定義於 RFC 1738。

URL 編碼的值	字元
%3C	<
%3E	>
%22	"
%23	#
%25	%
%7B	{
%7D	}
%7C	|
%5C	\
%5E	^
%7E	~
%5B	[
%5D	]
%60	`
%27	'
%20	空格

標準日誌檔案欄位

分佈的日誌檔案包含 33 個欄位。下列清單依序包含每個欄位名稱，以及該欄位中資訊的描述。

1. **date**

   事件發生的日期格式 YYYY-MM-DD。例如，2019-06-30。使用國際標準時間 (UTC) 的日期與時間。對於 WebSocket連線，這是連線關閉的日期。

2. **time**

   CloudFront 伺服器完成回應要求的時間 (以 UTC 為單位)，例如01:42:39。對於 WebSocket 連接，這是關閉連接的時間。

3. **x-edge-location**

   提供請求的節點。一個三字母代碼和任意指派的數字會辨識每個節點，例如 DFW3。三字母代碼通常會對應節點所在地理位置附近機場的國際航空運輸協會 (IATA) 機場代碼。(未來這些縮寫可能會改變。)

4. **sc-bytes**

   回應請求時伺服器提供給檢視器的總位元數，包括標頭。對於 WebSocket 連線，這是透過連線從伺服器傳送至用戶端的位元組總數。

5. **c-ip**

   檢視器的 IP 地址，該檢視器已執行請求，例如 192.0.2.183 或 2001:0db8:85a3::8a2e:0370:7334。如果檢視器使用 HTTP 代理或負載平衡器傳送請求，此欄位值則為代理或負載平衡器的 IP 地址。另請參閱 x-forwarded-for 欄位。

6. **cs-method**

   從檢視器接收到的 HTTP 請求方法。

7. **cs(Host)**

   CloudFront 分發的網域名稱 (例如，網域名稱)。

8. **cs-uri-stem**

   識別路徑和物件的請求 URL 部分 (例如 /images/cat.jpg)。URL 中包含問號 (?)，而且查詢字串不包含在日誌中。

9. **sc-status**

   包含以下其中一個值：

- 伺服器回應的 HTTP 狀態碼 (例如 200)。
- 000，這表示檢視器在伺服器回應請求之前已關閉連線。如果檢視器在伺服器開始傳送回應之後關閉連線，此欄位會包含伺服器開始傳送回應的 HTTP 狀態碼。

10 **cs(Referer)**

請求中的 Referer 標頭值。發出請求的網域名稱。常見推薦網站，包含搜尋引擎、其他直接連結到您物件的網站，以及您自己的網站。

11 **cs(User-Agent)**

請求中的 User-Agent 標頭值。識別請求來源的 User-Agent 標頭，例如提交請求的裝置與瀏覽器類型，或如果請求來自搜尋引擎，則識別是哪一個搜尋引擎。

12 **cs-uri-query**

請求 URI 的查詢字串部分 (如果有)。

當 URL 不包含查詢字串時，此欄位的值是一個連字號 (-)。如需詳細資訊，請參閱 根據查詢字串參數快取內容。

13 **cs(Cookie)**

請求中的 Cookie 標頭，包括名稱值對和關聯的屬性。

如果您啟用 Cookie 記錄功能， CloudFront 則無論您選擇轉發到原始伺服器的 Cookie 為何，都會在所有請求中記錄 Cookie。當請求不包含 Cookie 標頭時，此欄位的值是一個連字號 (-)。如需 Cookie 的詳細資訊，請參閱根據 Cookie 快取內容。

14 **x-edge-result-type**

在最後一個位元組離開伺服器之後，伺服器如何將回應分類。在某些情況下，結果類型會在伺服器準備好傳送回應的時間，以及完成傳送回應的時間中發生改變。另請參閱 x-edge-response-result-type 欄位。

例如，在 HTTP 串流中，假設伺服器在快取中找到串流的區段。在這種情況下，這個欄位的值通常是 Hit。不過，如果在伺服器已交付整個區段之前，檢視器關閉檢視器，則最終結果類型 (此欄位的值) 為 Error。

WebSocket 此欄位的Miss值為，因為內容無法快取，而且會直接代理至原點。

可能的值包括：

- Hit – 該伺服器從快取提供物件給檢視器。

- RefreshHit – 該伺服器在邊緣快取中找到物件，但物件已過期，因此伺服器會聯絡原始伺服器，以確認快取具有該物件的最新版本。
- Miss – 快取中的物件無法滿足請求，因此會將請求轉送至原始伺服器，並將結果傳回至檢視器。
- LimitExceeded— 因為超過 CloudFront 配額 (先前稱為限制)，因此要求遭到拒絕。
- CapacityExceeded – 伺服器會傳回 HTTP 503 狀態碼，因為在請求提供物件時沒有足夠的容量。
- Error – 通常，這表示請求導致客戶端錯誤 (sc-status 欄位的值在 4xx 範圍內) 或伺服器錯誤 (sc-status 欄位的值在 5xx 範圍內)。如果 sc-status 欄位的值是 200，或者如果該欄位的值是 Error 並且 x-edge-response-result-type 欄位的值不是 Error，這代表 HTTP 請求成功，但用戶端在接收所有位元組之前中斷連線。
- Redirect – 伺服器會根據分佈設定，將檢視器從 HTTP 重新引導至 HTTPS。

15**x-edge-request-id**

唯一識別要求的不透明字串。 CloudFront 還會在x-amz-cf-id響應頭中發送此字符串。

16**x-host-header**

包含在該請求 Host 標頭的檢視器值。如果您在物件網址中使用 CloudFront 網域名稱 (例如 d111111abcdef8.cloudfront.net)，則此欄位會包含該網域名稱。如果您使用物件 URL 中的備用網域名稱 (CNAME)，例如 www.example.com，則此欄位包含此備用網域名稱。

如果您使用備用網域名稱，請參閱網域名稱的欄位 7 中的 cs(Host)，此網域名稱與分佈相關聯。

17**cs-protocol**

檢視器請求的通訊協定 (http、https、ws 或 wss)。

18**cs-bytes**

檢視器包含在請求中的資料位元組總數，包括標頭。對於 WebSocket 連線，這是連線上從用戶端傳送至伺服器的位元組總數。

19**time-taken**

從伺服器收到檢視者請求，到伺服器將回應的最後一個位元組寫入輸出佇列的秒數 (以千分之一秒為單位，例如 0.082)，這會在伺服器上測量。從檢視器來看，取得完整回應的總時間會比該值來的長，因為網路延遲和 TCP 緩衝。

20**x-forwarded-for**

如果檢視器使用 HTTP 代理或負載平衡器傳送請求，c-ip 欄位值則為代理或負載平衡器的 IP 地址。在這種情況下，此欄位是產生請求的檢視器 IP 地址。此欄位可包含多個以逗號分隔的 IP 位址。每個 IP 位址可以是 IPv4 位址 (例如，192.0.2.183) 或 IPv6 位址 (例如2001:0db8:85a3::8a2e:0370:7334)。

如果檢視器無法使用 HTTP 代理或負載平衡器，則此欄位值是一個連字號 (-)。

21**ssl-protocol**

當請求使用 HTTPS 時，此欄位會包含檢視器和伺服器為傳輸請求和回應而交涉的 SSL/TLS 通訊協定。如需可能值的清單，請參閱 檢視器與之間支援的通訊協定和密碼 CloudFront 中支援的 SSL/TLS 通訊協定。

當欄位 17 中的 cs-protocol 為 http 時，此欄位的值是一個連字號 (-)。

22**ssl-cipher**

當請求使用 HTTPS 時，此欄位會包含檢視器和伺服器為加密請求和回應而交涉的 SSL/TLS 密碼。如需可能值的清單，請參閱 檢視器與之間支援的通訊協定和密碼 CloudFront 中支援的 SSL/TLS 密碼。

當欄位 17 中的 cs-protocol 為 http 時，此欄位的值是一個連字號 (-)。

23**x-edge-response-result-type**

在將回應傳回至檢視器之前，伺服器如何將回應分類。另請參閱 x-edge-result-type 欄位。可能的值包括：

- Hit – 該伺服器從快取提供物件給檢視器。
- RefreshHit – 該伺服器在邊緣快取中找到物件，但物件已過期，因此伺服器會聯絡原始伺服器，以確認快取具有該物件的最新版本。
- Miss – 快取中的物件無法滿足請求，因此伺服器會將請求轉送至原始伺服器，並將結果傳回至檢視器。
- LimitExceeded— 因為超過 CloudFront 配額 (先前稱為限制)，因此要求遭到拒絕。
- CapacityExceeded – 該伺服器會傳回 503 錯誤，因為在請求提供物件時沒有足夠的容量。
- Error – 通常，這表示請求導致客戶端錯誤 (sc-status 欄位的值在 4xx 範圍內) 或伺服器錯誤 (sc-status 欄位的值在 5xx 範圍內)。

  如果 x-edge-result-type 欄位的值為 Error 且此欄位的值不為 Error，則用戶端在完成下載之前中斷連線。

- Redirect – 伺服器會根據分佈設定，將檢視器從 HTTP 重新引導至 HTTPS。

24. **cs-protocol-version**

檢視器在請求中指定的 HTTP 版本。可能的值包括 HTTP/0.9、HTTP/1.0、HTTP/1.1、HTTP/2.0 及 HTTP/3.0。

25. **fle-status**

為分佈配置欄位層級加密時，此欄位包含可指出要求主體是否已成功處理的代碼。當伺服器成功處理要求主體時，會加密指定欄位中的值，並將請求轉送至原始伺服器，此欄位的值為 Processed。在這種情況下，x-edge-result-type 的值仍然可以表示用戶端或伺服器端的錯誤。

此欄位可能的值包含：

- ForwardedByContentType – 伺服器無須剖析與加密便將請求轉送到原始伺服器，因為沒有配置任何內容類型。
- ForwardedByQueryArgs – 伺服器無需剖析或加密便將請求轉送到原始伺服器，因為請求包含查詢參數，此參數不在欄位層級加密的組態裡。
- ForwardedDueToNoProfile – 伺服器無需剖析或加密便將請求轉送到原始伺服器，因為在欄位層級加密的組態裡沒有指定設定檔。
- MalformedContentTypeClientError – 因為 Content-Type 標頭的值不是有效格式，因此伺服器拒絕請求並將 HTTP 400 狀態碼傳回至檢視器。
- MalformedInputClientError – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為要求主體不是有效格式。
- MalformedQueryArgsClientError – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為查詢參數空白或不是有效格式。
- RejectedByContentType – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為在欄位層級加密的組態中沒有指定內容類型。
- RejectedByQueryArgs – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為在欄位層級加密的組態中沒有指定查詢參數。
- ServerError – 原始伺服器傳回錯誤。

如果請求超過欄位層級的加密配額 (先前稱為限制)，此欄位會包含下列其中一個錯誤碼，而伺服器會將 HTTP 狀態碼傳回給檢視器 400。如需目前欄位層級加密的配額的詳細資訊，請參閱欄位層級加密的配額。

- FieldLengthLimitClientError – 配置為加密的欄位已超過允許的長度上限。

- FieldNumberLimitClientError – 將分佈配置為加密的請求所包含的欄位數超過了允許的欄位數。
- RequestLengthLimitClientError – 當配置了欄位層級加密時，要求主體的長度超過允許的長度上限。

如果不為分佈設定欄位層級加密，則此欄位的值是一個連字號 (-)。

26. **fle-encrypted-fields**

伺服器加密並轉寄至來源的欄位層級加密欄位數目。 CloudFront server 會在加密資料時將已處理的要求串流至來源，因此即使的值為錯誤，此欄位fle-status也可以具有值。

如果不為分佈設定欄位層級加密，則此欄位的值是一個連字號 (-)。

27. **c-port**

來自檢視器之請求的連接埠號碼。

28. **time-to-first-byte**

接收請求與寫入回應的第一個位元組之間的秒數 (如伺服器上所測量)。

29. **x-edge-detailed-result-type**

此欄位包含與 x-edge-result-type 欄位相同的值，但下列情況除外：

- 該物件已從 Origin Shield 層提供給檢視器時，此欄位包含 OriginShieldHit。
- 當物件不在 CloudFront 快取中，且回應是由原始請求 Lambda @Edge 函數產生時，此欄位會包含MissGeneratedResponse。
- 當 x-edge-result-type 欄位的值為 Error 時，此欄位會包含下列其中一個值，以及有關該錯誤的詳細資訊：
  - AbortedOrigin – 該伺服器發生原始伺服器問題。
  - ClientCommError – 檢視器的回應因伺服器與檢視器之間發生通訊問題而遭到中斷。
  - ClientGeoBlocked – 分佈設定為拒絕來自檢視者地理位置的請求。
  - ClientHungUpRequest – 檢視器在傳送請求的同時提早停止。
  - Error – 發生錯誤，其錯誤類型不符合任何其他類別。當此伺服器提供來自快取的錯誤回應時，可能會發生此錯誤類型。
  - InvalidRequest – 該伺服器收到來自檢視器的無效請求。
  - InvalidRequestBlocked – 對請求的資源的存取遭到封鎖。
  - InvalidRequestCertificate – 分佈不符合建立 HTTPS 連線的 SSL/TLS 憑證。

- InvalidRequestHeader — 請求包含無效的標頭。
- InvalidRequestMethod – 分佈未設定為處理使用的 HTTP 請求方法。當分佈只支援可快取的請求時會發生此情況。
- OriginCommError – 在連接到原始伺服器或從原始伺服器讀取資料時，請求逾時。
- OriginConnectError – 該伺服器無法連線到原始伺服器。
- OriginContentRangeLengthError – 原始伺服器回應中的 Content-Length 標頭不符合 Content-Range 標頭中的長度。
- OriginDnsError – 該伺服器無法解析原始伺服器的網域名稱。
- OriginError – 原始伺服器傳回不正確的回應。
- OriginHeaderTooBigError – 原始伺服器傳回的標頭對邊緣伺服器太大，因而無法處理。
- OriginInvalidResponseError – 原始伺服器傳回無效的回應。
- OriginReadError – 該伺服器無法從原始伺服器讀取。
- OriginWriteError – 該伺服器無法寫入原始伺服器。
- OriginZeroSizeObjectError – 從原始伺服器傳送大小為零的物件，因而導致錯誤。
- SlowReaderOriginError – 檢視器讀取訊息過慢，因而導致原始伺服器錯誤。

30**sc-content-type**

回應的 HTTP Content-Type 標頭值。

31**sc-content-len**

回應的 HTTP Content-Length 標頭值。

32**sc-range-start**

當回應包含 HTTP Content-Range 標頭時，此欄位包含範圍起始值。

33**sc-range-end**

當回應包含 HTTP Content-Range 標頭時，此欄位包含範圍結束值。

以下是分佈的範例日誌檔案：

```
#Version: 1.0
#Fields: date time x-edge-location sc-bytes c-ip cs-method cs(Host) cs-uri-stem sc-
status cs(Referer) cs(User-Agent) cs-uri-query cs(Cookie) x-edge-result-type x-edge-
request-id x-host-header cs-protocol cs-bytes time-taken x-forwarded-for ssl-protocol
 ssl-cipher x-edge-response-result-type cs-protocol-version fle-status fle-encrypted-
```

```
fields c-port time-to-first-byte x-edge-detailed-result-type sc-content-type sc-
content-len sc-range-start sc-range-end
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
 SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ== d111111abcdef8.cloudfront.net
 https 23 0.001 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.001 Hit
 text/html 78 - -
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
 k6WGMNkEzR5BEM_SaF47gjtX9zBDO2m349OY2an0QPEaUum1ZOLrow== d111111abcdef8.cloudfront.net
 https 23 0.000 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.000 Hit
 text/html 78 - -
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
 f37nTMVvnKvV2ZSvEsivup_c2kZ7VXzYdjC-GUQZ5qNs-89BlWazbw== d111111abcdef8.cloudfront.net
 https 23 0.001 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.001 Hit
 text/html 78 - -
2019-12-13 22:36:27 SEA19-C1 900 192.0.2.200 GET d111111abcdef8.cloudfront.net /
favicon.ico 502 http://www.example.com/ Mozilla/5.0%20(Windows
%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Error
 1pkpNfBQ39sYMnjjUQjmH2w1wdJnbHYTbag21o_3OfcQgPzdL2RSSQ== www.example.com http 675
 0.102 - - - Error HTTP/1.1 - - 25260 0.102 OriginDnsError text/html 507 - -
2019-12-13 22:36:26 SEA19-C1 900 192.0.2.200 GET d111111abcdef8.cloudfront.net / 502
 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Error
 3AqrZGCnF_g0-5KOvfA7c9XLcf4YGvMFSeFdIetR1N_2y8jSis8Zxg== www.example.com http 735
 0.107 - - - Error HTTP/1.1 - - 3802 0.107 OriginDnsError text/html 507 - -
2019-12-13 22:37:02 SEA19-C2 900 192.0.2.200 GET d111111abcdef8.cloudfront.net / 502
 - curl/7.55.1 - - Error kBkDzGnceVtWHqSCqBUqtA_cEs2T3tFUBbnBNkB9El_uVRhHgcZfcw==
 www.example.com http 387 0.103 - - - Error HTTP/1.1 - - 12644 0.103 OriginDnsError
 text/html 507 - -
```

標準日誌的費用

標準記錄是的選擇性功能 CloudFront。啟用標準記錄沒有額外的費用。不過，在 Amazon S3 上的存放和存取檔案會依一般 Amazon S3 費用計費 (您可以隨時將它們刪除)。

如需 Amazon S3 定價的詳細資訊，請參閱 Amazon S3 定價。

如需有關 CloudFront 定價的詳細資訊，請參閱CloudFront 定價。

即時日誌

透過 CloudFront 即時記錄，您可以即時取得對分發的要求的相關資訊 (記錄會在收到要求後的幾秒內傳送)。您可以使用即時日誌來監控、分析並根據內容交付效能採取動作。

CloudFront 即時記錄是可設定的。您可以選擇：

- 您可以為即時日誌選擇抽樣率，即要接收即時日誌的請求百分比。
- 您想要在日誌中接收的特定欄位。
- 您想要接收即時日誌的特定快取行為 (路徑模式)。

CloudFront 即時日誌會在 Amazon Kinesis 資料串流中傳送至您選擇的資料串流。您可以建立自己的 Kinesis 資料串流取用者，或使用 Amazon 資料 Firehose 將日誌資料傳送到 Amazon 簡單儲存服務 (Amazon S3)、Amazon Redshift、亞馬遜 OpenSearch 服務 (服OpenSearch 務) 或第三方日誌處理服務。

CloudFront 即時記錄的費用，以及您使用 Kinesis Data Streams 所產生的費用。如需有關定價的詳細資訊，請參閱 Amazon CloudFront 定價和 Amazon Kinesis Data Streams 定價。

> **Important**
>
> 我們建議您使用這些記錄來瞭解內容要求的性質，而不是對所有要求進行完整記錄。CloudFront 盡最大努力提供即時記錄。在實際處理請求之後，才可能長時間交付特定請求的日誌項目，在極少數的情況下，有可能完全不會交付日誌項目。當即時記錄中省略記錄項目時，即時記錄檔中的項目數量將與 AWS 帳單和使用情況報告中顯示的使用量不符。

瞭解即時日誌組態

若要使用 CloudFront 即時記錄，請先建立即時記錄組態。即時日誌組態包含您要接收哪些日誌欄位、日誌的抽樣頻率，以及您要傳送日誌的 Kinesis 資料串流的相關資訊。

具體來說，即時日誌組態包含下列組態：

- 名稱

- 抽樣頻率
- 欄位
- 端點 (Kinesis 資料串流)
- IAM 角色

名稱

用來識別即時日誌組態的名稱。

抽樣頻率

抽樣率是介於 1 到 100 之間的整數 (含)，可決定以即時日誌形式傳送至 Kinesis Data Streams 的檢視器請求百分比。若要在即時日誌中包含每個檢視器請求，請指定 100 的抽樣頻率。您可以選擇較低的抽樣頻率以降低成本，同時仍然在即時日誌中收到代表性的請求資料樣本。

欄位

包含在每個即時日誌中的欄位清單。每個日誌記錄最多包含 40 個欄位，而且您可以選擇接收所有可用欄位，或只接收監視和分析效能所需的欄位。

下列清單包含每個欄位名稱以及該欄位中資訊的描述。這些欄位會依它們交付給 Kinesis Data Streams 日誌中顯示的順序列出。

字段 46-63 是媒體播放器客戶端可以隨每個請求發送到 CDN 的常見媒體客戶端數據（CMCD）。您可以使用此資料來瞭解每個要求，例如媒體類型 (音訊、視訊)、播放速率和串流長度。這些欄位只會在傳送至您的即時記錄時顯示 CloudFront。

1. **`timestamp`**

   邊緣伺服器完成回應請求的日期和時間。

2. **`c-ip`**

   檢視器的 IP 地址，該檢視器已執行請求，例如 `192.0.2.183` 或 `2001:0db8:85a3::8a2e:0370:7334`。如果檢視器使用 HTTP 代理或負載平衡器傳送請求，此欄位值則為代理或負載平衡器的 IP 地址。另請參閱 `x-forwarded-for` 欄位。

3. **`time-to-first-byte`**

   接收請求與寫入回應的第一個位元組之間的秒數 (如伺服器上所測量)。

4. **`sc-status`**

伺服器回應的 HTTP 狀態碼 (例如 200)。

5. **sc-bytes**

   回應請求時伺服器提供給檢視器的總位元數，包括標頭。對於 WebSocket 連線，這是透過連線從伺服器傳送至用戶端的位元組總數。

6. **cs-method**

   從檢視器接收到的 HTTP 請求方法。

7. **cs-protocol**

   檢視器請求的通訊協定 (http、https、ws 或 wss)。

8. **cs-host**

   包含在該請求 Host 標頭的檢視器值。如果您在物件網址中使用 CloudFront 網域名稱 (例如 d111111abcdef8.cloudfront.net)，則此欄位會包含該網域名稱。如果您使用物件 URL 中的備用網域名稱 (CNAME)，例如 www.example.com，則此欄位包含此備用網域名稱。

9. **cs-uri-stem**

   整個請求 URL，包括查詢字串 (如果存在)，但不包括網域名稱。例如，/images/cat.jpg?mobile=true。

   > **Note**
   >
   > 在標準日誌中，該 cs-uri-stem 值不包括查詢字串。

10. **cs-bytes**

    檢視器包含在請求中的資料位元組總數，包括標頭。對於 WebSocket 連線，這是連線上從用戶端傳送至伺服器的位元組總數。

11. **x-edge-location**

    提供請求的節點。一個三字母代碼和任意指派的數字會辨識每個節點，例如 DFW3。三字母代碼通常會對應節點所在地理位置附近機場的國際航空運輸協會 (IATA) 機場代碼。(未來這些縮寫可能會改變。)

12. **x-edge-request-id**

    唯一識別要求的不透明字串。 CloudFront 還會在x-amz-cf-id響應頭中發送此字符串。

13. **x-host-header**

    CloudFront 分發的網域名稱 (例如，網域名稱)。

14. **time-taken**

    從伺服器收到檢視者請求，到伺服器將回應的最後一個位元組寫入輸出佇列的秒數 (以千分之一秒為單位，例如 0.082)，這會在伺服器上測量。從檢視器來看，取得完整回應的總時間會比該值來的長，因為網路延遲和 TCP 緩衝。

15. **cs-protocol-version**

    檢視器在請求中指定的 HTTP 版本。可能的值包括 `HTTP/0.9`、`HTTP/1.0`、`HTTP/1.1`、`HTTP/2.0` 及 `HTTP/3.0`。

16. **c-ip-version**

    請求的 IP 版本 (IPv4 或 IPv6)。

17. **cs-user-agent**

    請求中的 `User-Agent` 標頭值。識別請求來源的 `User-Agent` 標頭，例如提交請求的裝置與瀏覽器類型，或如果請求來自搜尋引擎，則識別是哪一個搜尋引擎。

18. **cs-referer**

    請求中的 `Referer` 標頭值。發出請求的網域名稱。常見推薦網站，包含搜尋引擎、其他直接連結到您物件的網站，以及您自己的網站。

19. **cs-cookie**

    請求中的 `Cookie` 標頭，包括名稱值對和關聯的屬性。

    > **Note**
    >
    > 此欄位會截斷為 800 位元組。

20. **cs-uri-query**

    請求 URI 的查詢字串部分 (如果有)。

21. **x-edge-response-result-type**

    在將回應傳回至檢視器之前，伺服器如何將回應分類。另請參閱 `x-edge-result-type` 欄位。可能的值包括：

- Hit – 該伺服器從快取提供物件給檢視器。
- RefreshHit – 該伺服器在邊緣快取中找到物件，但物件已過期，因此伺服器會聯絡原始伺服器，以確認快取具有該物件的最新版本。
- Miss – 快取中的物件無法滿足請求，因此伺服器會將請求轉送至原始伺服器，並將結果傳回至檢視器。
- LimitExceeded— 因為超過 CloudFront 配額 (先前稱為限制)，因此要求遭拒。
- CapacityExceeded – 該伺服器會傳回 503 錯誤，因為在請求提供物件時沒有足夠的容量。
- Error – 通常，這表示請求導致客戶端錯誤 (sc-status 欄位的值在 4xx 範圍內) 或伺服器錯誤 (sc-status 欄位的值在 5xx 範圍內)。

  如果 x-edge-result-type 欄位的值為 Error 且此欄位的值不為 Error，則用戶端在完成下載之前中斷連線。
- Redirect – 伺服器會根據分發設定，將檢視器從 HTTP 重新導向至 HTTPS。

22**x-forwarded-for**

如果檢視器使用 HTTP 代理或負載平衡器傳送請求，c-ip 欄位值則為代理或負載平衡器的 IP 地址。在這種情況下，此欄位是產生請求的檢視器 IP 地址。此欄位可包含多個以逗號分隔的 IP 位址。每個 IP 位址都可以是 IPv4 位址 (例如，192.0.2.183) 或 IPv6 位址 (例如2001:0db8:85a3::8a2e:0370:7334)。

23**ssl-protocol**

當請求使用 HTTPS 時，此欄位會包含檢視器和伺服器為傳輸請求和回應而交涉的 SSL/TLS 通訊協定。如需可能值的清單，請參閱 檢視器與之間支援的通訊協定和密碼 CloudFront 中支援的 SSL/TLS 通訊協定。

24**ssl-cipher**

當請求使用 HTTPS 時，此欄位會包含檢視器和伺服器為加密請求和回應而交涉的 SSL/TLS 密碼。如需可能值的清單，請參閱 檢視器與之間支援的通訊協定和密碼 CloudFront 中支援的 SSL/TLS 密碼。

25**x-edge-result-type**

在最後一個位元組離開伺服器之後，伺服器如何將回應分類。在某些情況下，結果類型會在伺服器準備好傳送回應的時間，以及完成傳送回應的時間中發生改變。另請參閱 x-edge-response-result-type 欄位。

例如，在 HTTP 串流中，假設伺服器在快取中找到串流的區段。在這種情況下，這個欄位的值通常是 Hit。不過，如果在伺服器已交付整個區段之前，檢視器關閉檢視器，則最終結果類型 (此欄位的值) 為 Error。

WebSocket 連線的Miss值會為此欄位，因為內容無法快取，而且會直接代理至原點。

可能的值包括：

- Hit – 該伺服器從快取提供物件給檢視器。
- RefreshHit – 該伺服器在邊緣快取中找到物件，但物件已過期，因此伺服器會聯絡原始伺服器，以確認快取具有該物件的最新版本。
- Miss – 快取中的物件無法滿足請求，因此會將請求轉送至原始伺服器，並將結果傳回至檢視器。
- LimitExceeded— 因為超過 CloudFront 配額 (先前稱為限制)，因此要求遭拒。
- CapacityExceeded – 伺服器會傳回 HTTP 503 狀態碼，因為在請求提供物件時沒有足夠的容量。
- Error – 通常，這表示請求導致客戶端錯誤 (sc-status 欄位的值在 4xx 範圍內) 或伺服器錯誤 (sc-status 欄位的值在 5xx 範圍內)。如果 sc-status 欄位的值是 200，或者如果該欄位的值是 Error 並且 x-edge-response-result-type 欄位的值不是 Error，這代表 HTTP 請求成功，但用戶端在接收所有位元組之前中斷連線。
- Redirect – 伺服器會根據分佈設定，將檢視器從 HTTP 重新引導至 HTTPS。

26. **fle-encrypted-fields**

伺服器加密並轉寄至來源的欄位層級加密欄位數目。 CloudFront server 會在加密資料時將已處理的要求串流至來源，因此即使的值為錯誤，此欄位fle-status也可以具有值。

27. **fle-status**

為分佈配置欄位層級加密時，此欄位包含可指出要求主體是否已成功處理的代碼。當伺服器成功處理要求主體時，會加密指定欄位中的值，並將請求轉送至原始伺服器，此欄位的值為 Processed。在這種情況下，x-edge-result-type 的值仍然可以表示用戶端或伺服器端的錯誤。

此欄位可能的值包含：

- ForwardedByContentType – 伺服器無須剖析與加密便將請求轉送到原始伺服器，因為沒有配置任何內容類型。
- ForwardedByQueryArgs – 伺服器無需剖析或加密便將請求轉送到原始伺服器，因為請求包含查詢參數，此參數不在欄位層級加密的組態裡。

- ForwardedDueToNoProfile – 伺服器無需剖析或加密便將請求轉送到原始伺服器，因為在欄位層級加密的組態裡沒有指定設定檔。
- MalformedContentTypeClientError – 因為 Content-Type 標頭的值不是有效格式，因此伺服器拒絕請求並將 HTTP 400 狀態碼傳回至檢視器。
- MalformedInputClientError – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為要求主體不是有效格式。
- MalformedQueryArgsClientError – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為查詢參數空白或不是有效格式。
- RejectedByContentType – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為在欄位層級加密的組態中沒有指定內容類型。
- RejectedByQueryArgs – 伺服器拒絕請求且將 HTTP 400 狀態碼傳回給檢視器，因為在欄位層級加密的組態中沒有指定查詢參數。
- ServerError – 原始伺服器傳回錯誤。

如果請求超過欄位層級的加密配額 (先前稱為限制)，此欄位會包含下列其中一個錯誤碼，而伺服器會將 HTTP 狀態碼傳回給檢視器 400。如需目前欄位層級加密的配額的詳細資訊，請參閱欄位層級加密的配額。

- FieldLengthLimitClientError – 配置為加密的欄位已超過允許的長度上限。
- FieldNumberLimitClientError – 將分佈配置為加密的請求所包含的欄位數超過了允許的欄位數。
- RequestLengthLimitClientError – 當配置了欄位層級加密時，請求本文的長度超過允許的長度上限。

28 **sc-content-type**

回應的 HTTP Content-Type 標頭值。

29 **sc-content-len**

回應的 HTTP Content-Length 標頭值。

30 **sc-range-start**

當回應包含 HTTP Content-Range 標頭時，此欄位包含範圍起始值。

31 **sc-range-end**

當回應包含 HTTP Content-Range 標頭時，此欄位包含範圍結束值。

32 **c-port**

來自檢視器之請求的連接埠號碼。

33**x-edge-detailed-result-type**

此欄位包含與 x-edge-result-type 欄位相同的值，但下列情況除外：

- 該物件已從 Origin Shield 層提供給檢視器時，此欄位包含 OriginShieldHit。
- 當物件不在 CloudFront 快取中，且回應是由原始請求 Lambda @Edge 函數產生時，此欄位會包含MissGeneratedResponse。
- 當 x-edge-result-type 欄位的值為 Error 時，此欄位會包含下列其中一個值，以及有關該錯誤的詳細資訊：
  - AbortedOrigin – 該伺服器發生原始伺服器問題。
  - ClientCommError – 檢視器的回應因伺服器與檢視器之間發生通訊問題而遭到中斷。
  - ClientGeoBlocked – 分佈設定為拒絕來自檢視者地理位置的請求。
  - ClientHungUpRequest – 檢視器在傳送請求的同時提早停止。
  - Error – 發生錯誤，其錯誤類型不符合任何其他類別。當此伺服器提供來自快取的錯誤回應時，可能會發生此錯誤類型。
  - InvalidRequest – 該伺服器收到來自檢視器的無效請求。
  - InvalidRequestBlocked – 對請求的資源的存取遭到封鎖。
  - InvalidRequestCertificate – 分佈不符合建立 HTTPS 連線的 SSL/TLS 憑證。
  - InvalidRequestHeader — 請求包含無效的標頭。
  - InvalidRequestMethod – 分佈未設定為處理使用的 HTTP 請求方法。當分佈只支援可快取的請求時會發生此情況。
  - OriginCommError – 在連接到原始伺服器或從原始伺服器讀取資料時，請求逾時。
  - OriginConnectError – 該伺服器無法連線到原始伺服器。
  - OriginContentRangeLengthError – 原始伺服器回應中的 Content-Length 標頭不符合 Content-Range 標頭中的長度。
  - OriginDnsError – 該伺服器無法解析原始伺服器的網域名稱。
  - OriginError – 原始伺服器傳回不正確的回應。
  - OriginHeaderTooBigError – 原始伺服器傳回的標頭對邊緣伺服器太大，因而無法處理。
  - OriginInvalidResponseError – 原始伺服器傳回無效的回應。
  - OriginReadError – 該伺服器無法從原始伺服器讀取。
  - OriginWriteError – 該伺服器無法寫入原始伺服器。

- `OriginZeroSizeObjectError` – 從原始伺服器傳送大小為零的物件，因而導致錯誤。
- `SlowReaderOriginError` – 檢視器讀取訊息過慢，因而導致原始伺服器錯誤。

34 **`c-country`**

國家/地區代碼代表檢視者的地理位置，由檢視器的 IP 地址決定。如需國家/地區代碼的清單，請參閱 ISO 3166-1 alpha-2。

35 **`cs-accept-encoding`**

檢視器請求中的 `Accept-Encoding` 標頭值。

36 **`cs-accept`**

檢視器請求中的 `Accept` 標頭值。

37 **`cache-behavior-path-pattern`**

識別符合檢視器請求之快取行為的路徑模式。

38 **`cs-headers`**

檢視器請求中的 HTTP 標頭 (名稱和值)。

> **Note**
> 此欄位會截斷為 800 位元組。

39 **`cs-header-names`**

檢視器請求中 HTTP 標頭 (非值) 的名稱。

> **Note**
> 此欄位會截斷為 800 位元組。

40 **`cs-headers-count`**

檢視器請求中的 HTTP 標頭數目。

41 **`origin-fbl`**

CloudFront 與您的來源之間的第一個位元組延遲的秒數。

42 **`origin-lbl`**

CloudFront 與來源之間最後位元組延遲的秒數。

43 **`asn`**

檢視器的自治系統編號 (ASN)。

44 **`primary-distribution-id`**

啟用持續部署時，此 ID 會識別哪個發行版是目前發行版中的主要發行版。

45 **`primary-distribution-dns-name`**

啟用持續部署時，此值會顯示與目前 CloudFront 分發相關的主要網域名稱 (例如，d111111abcdef8.cloudfront.net)。

> **即時記錄檔中的 CMCD 欄位**
>
> 如需這些欄位的詳細資訊，請參閱 CTA 規格 Web 應用程式影片生態系統-通用媒體用戶端資料 CTA-5004 文件。

46 **`cmcd-encoded-bitrate`**

請求的音頻或視頻對象的編碼比特率。

47 **`cmcd-buffer-length`**

請求的媒體對象的緩衝區長度。

48 **`cmcd-buffer-starvation`**

無論是緩衝區在前面的請求和對象請求之間的某個點餓了。這可能會導致播放器處於重新緩衝狀態，這可能會阻止視頻或音頻播放。

49 **`cmcd-content-id`**

識別目前內容的唯一字串。

50 **`cmcd-object-duration`**

要求物件的播放持續時間 (以毫秒為單位)。

51 **`cmcd-deadline`**

從請求時間開始，此物件的第一個樣本必須可用的截止日期，以避免緩衝區不足狀態或其他播放問題。

52. **cmcd-measured-throughput**

用戶端與伺服器之間的輸送量 (由用戶端測量)。

53. **cmcd-next-object-request**

下一個要求物件的相對路徑。

54. **cmcd-next-range-request**

如果下一個請求是部分對象請求，則此字符串表示要請求的字節範圍。

55. **cmcd-object-type**

被請求的當前對象的媒體類型。

56. **cmcd-playback-rate**

1 如果是實時，2 如果雙速，0 如果不播放。

57. **cmcd-requested-maximum-throughput**

客戶認為足以交付資產的要求最大輸送量。

58. **cmcd-streaming-format**

定義目前要求的串流格式。

59. **cmcd-session-id**

識別目前播放工作階段的 GUID。

60. **cmcd-stream-type**

記號識別區段可用性。 v= 所有段都可用。 l= 區段隨著時間的推移變為可用。

61. **cmcd-startup**

如果在緩衝區空事件之後啟動，搜索或恢復期間緊急需要對象，則密鑰將包含在沒有值的情況下。

62. **cmcd-top-bitrate**

用戶端可播放的最高位元速率轉譯。

63. **cmcd-version**

此規格的版本，用於解譯已定義的索引鍵名稱和值。如果省略此機碼，用戶端和伺服器必須將值解譯為版本 1 所定義的值。

端點 (Kinesis 資料串流)

端點包含您要傳送即時日誌的 Kinesis 資料串流相關資訊。您提供資料串流的 Amazon Resource Name (ARN)。

如需建立 Kinesis 資料流的詳細資訊，請參閱 Amazon Kinesis Data Streams 開發人員指南中的下列主題。

- 使用主控台管理串流
- 使用執行基本 Kinesis 資料串流作業 AWS CLI
- 建立串流 (使用 AWS SDK for Java)

當您建立資料串流時，您需要指定分片的數目。使用下列資訊來協助您估計需要的碎片數量。

估計 Kinesis 資料串流的碎片數量

1. 計算（或估計） CloudFront 分佈每秒收到的請求數量。

   您可以使用使用CloudFront 情況報告 (在 CloudFront 主控台中) 和CloudFront 指標 (在 CloudFront 和 Amazon 主控 CloudWatch 台中) 來協助您計算每秒請求數。

2. 確定單一即時日誌記錄的典型大小。

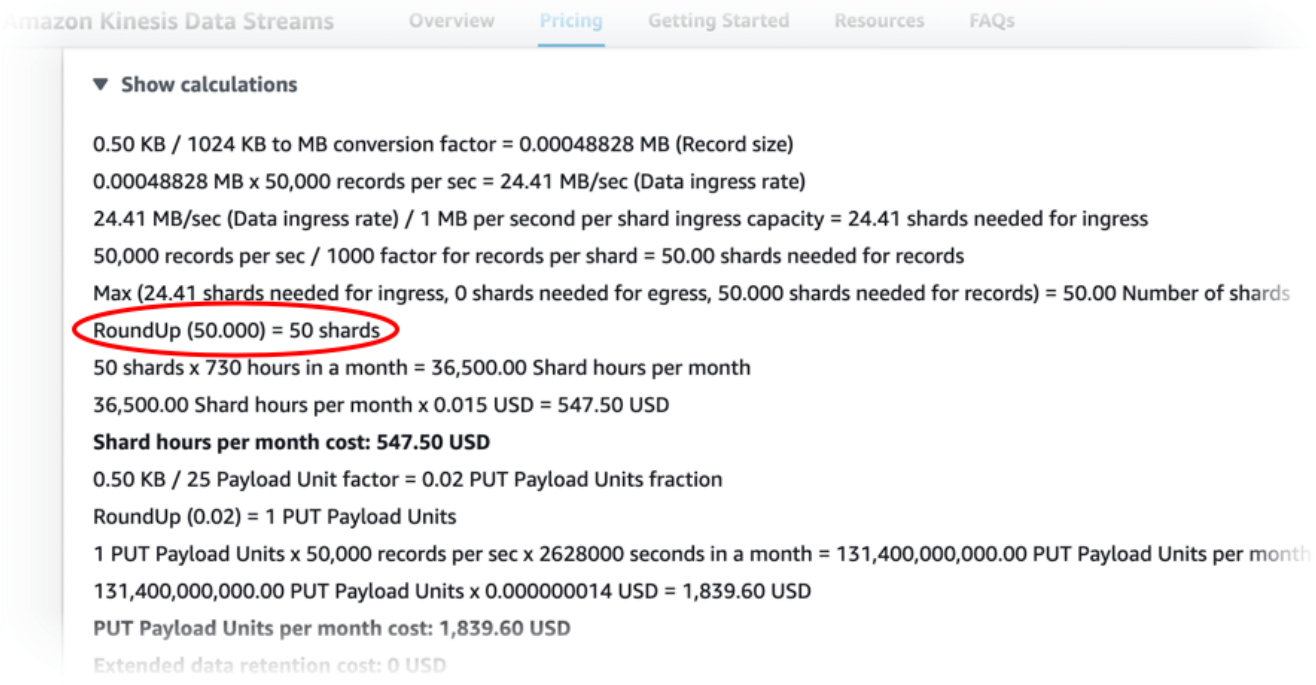
   一般而言，單一日誌記錄大約是 500 個位元組。包含所有可用欄位的大型記錄一般約為 1 KB。

   如果不確定日誌記錄的大小，您可以啟用取樣率較低 (例如 1%) 的即時日誌，然後使用 Kinesis Data Streams 中的監控資料，來計算平均記錄大小 (總傳入位元組數除以記錄總數)。

3. 在 Amazon Kinesis Data Streams 定價頁面的 Pricing calculator (定價計算工具) 中，輸入每秒請求數 (記錄) 和單一日誌記錄的平均記錄大小。然後選擇 Show calculations (顯示計算結果)。

   定價計算工具會顯示您需要的分區數量。(它也會向您顯示預估成本。)

   下列範例顯示平均記錄大小為 0.5 KB，以及每秒 50,000 個請求，您需要 50 個分區。



IAM 角色

授予將即時日誌傳遞至 Kinesis 資料串流之 CloudFront 權限的 AWS Identity and Access Management (IAM) 角色。

使用 CloudFront 主控台建立即時記錄組態時，可以選擇 [建立新服務角色]，讓主控台為您建立 IAM 角色。

使用或 CloudFront API (AWS CloudFormation AWS CLI 或 SDK) 建立即時記錄設定時，您必須自行建立 IAM 角色並提供角色 ARN。若要自行建立 IAM 角色，請使用下列政策。

IAM 角色信任原則

若要使用以下 IAM 角色信任政策，請將 *111122223333* 取代為您的 AWS 帳戶 號碼。此原則中的`Condition`元素有助於防止混淆的副問題，因為只 CloudFront 能代表您的 AWS 帳戶.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudfront.amazonaws.com"
```

```
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

未加密資料串流的 IAM 角色許可原則

若要使用下列原則，請將 *arn: aw: ##:####-2:123456789012: ##/##### Kinesis* 資料串流的 ARN。StreamName

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:DescribeStream",
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-2:123456789012:stream/StreamName"
            ]
        }
    ]
}
```

已加密資料串流的 IAM 角色許可原則

############ ARN#AWS##### 2#123456789012###/### Kinesis ###### ARN ##### ##### ARN#AW######## 2#123456789012 StreamName##/ AWS KMS key

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:DescribeStream",
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-2:123456789012:stream/StreamName"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/e58a3d0b-fe4f-4047-a495-
ae03cc73d486"
            ]
        }
    ]
}
```

建立和使用即時日誌組態

您可以使用即時日誌組態，即時取得對發佈發佈的請求相關資訊 (日誌會在收到請求後幾秒內傳送)。您可以使用 AWS Command Line Interface (AWS CLI) 或使用 CloudFront API 在 CloudFront 主控台中建立即時記錄設定。

若要使用即時記錄組態，請將其附加至 CloudFront 散發中的一或多個快取行為。

建立即時日誌組態 (主控台)

建立即時記錄組態

1. 登入 AWS Management Console 並在主 CloudFront 控台中開啟「記錄」頁面，位於https://console.aws.amazon.com/cloudfront/v4/home?#/logs。
2. 選擇即時組態索引標籤。
3. 選擇建立組態。

4. 在名稱中，輸入模型組態的名稱。
5. 在「取樣率」中，輸入要接收日誌記錄的請求百分比。
6. 在「欄位」 中，選擇要在即時記錄中接收的欄位。
   - 若要包含記錄檔的所有 CMCD 欄位，請選擇 [CMCD 所有金鑰]。
7. 在端點中，選擇一或多個 Kinesis 資料串流以接收即時記錄。

> **Note**
>
> CloudFront 即時記錄會傳送至您在 Kinesis 資料串流中指定的資料串流。若要讀取和分析即時記錄，您可以建立自己的 Kinesis 資料串流取用者。您也可以使用 Firehose 將日誌資料傳送到 Amazon S3、Amazon Redshift、Amazon OpenSearch 服務或第三方日誌處理服務。

8. 對於 IAM 角色，請選擇建立新服務角色或選擇現有角色。您必須具有建立 IAM 角色的許可。
9. (選擇性) 對於發佈，請選擇要附加至即時記錄組態的 CloudFront散佈和快取行為。
10. 選擇建立組態。

如果成功，主控台會顯示您剛才建立的即時日誌組態詳細資料。

如需詳細資訊，請參閱 瞭解即時日誌組態。

建立即時日誌組態 (AWS CLI)

若要使用 AWS Command Line Interface (AWS CLI) 建立即時記錄組態，請使用aws cloudfront create-realtime-log-config指令。您可以使用輸入檔案來提供命令的輸入參數，而不是將每個個別參數指定為命令列輸入。

建立即時日誌組態 (含輸入檔案的 CLI)

1. 使用下列命令建立一個命名為 `rtl-config.yaml` 的檔案，其中包含 create-realtime-log-config 命令的所有輸入參數。

```
aws cloudfront create-realtime-log-config --generate-cli-skeleton yaml-input > rtl-config.yaml
```

2. 開啟您剛才建立且命名為 `rtl-config.yaml` 的檔案。編輯檔案以指定所需的即時日誌組態設定，然後儲存檔案。注意下列事項：

- 對於 StreamType，唯一有效的值為 Kinesis。

如需即時長組態設定的詳細資訊，請參閱瞭解即時日誌組態。

3. 使用下列命令，使用 rtl-config.yaml 檔案中的輸入參數建立即時日誌組態。

```
aws cloudfront create-realtime-log-config --cli-input-yaml file://rtl-config.yaml
```

如果成功，命令的輸出會顯示您剛才建立的即時日誌組態的詳細資料。

若要將即時日誌組態附加至現有發行版 (包含輸入檔案的 CLI)

1. 使用下列命令來儲存您要更新之 CloudFront 發行版的發佈組態。將 *distribution_ID* 取代為分佈的 ID。

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml > dist-
config.yaml
```

2. 開啟您剛才建立且命名為 dist-config.yaml 的檔案。編輯檔案，對您要更新的每個快取行為進行下列變更，以使用即時日誌組態。

- 在快取行為中，新增名為 RealtimeLogConfigArn 的欄位。對於欄位的值，請使用您想要附加到此快取行為的即時日誌組態的 ARN。
- 將 ETag 欄位重新命名為 IfMatch，但不要變更欄位的值。

完成後儲存檔案。

3. 使用下列命令來更新分佈，以使用即時日誌組態。將 *distribution_ID* 取代為分佈的 ID。

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://
dist-config.yaml
```

如果成功，命令的輸出會顯示您剛才更新的分佈詳細資料。

建立即時日誌組態 (API)

若要使用 CloudFront API 建立即時記錄設定，請使用 CreateRealtimeLogConfig. 如需有關您在此 API 呼叫中指定之參數的詳細資訊瞭解即時日誌組態，請參閱 AWS SDK 或其他 API 用戶端的 API 參考文件。

建立即時日誌組態之後，您可以使用下列其中一個 API 呼叫，將其附加至快取行為：

- 若要將其附加至現有發行版中的快取行為，請使用UpdateDistribution。
- 若要將其附加至新發行版中的快取行為，請使用CreateDistribution。

對於這兩個 API 呼叫，請在快取行為內的 `RealtimeLogConfigArn` 欄位中提供即時日誌組態的 ARN。有關您在這些 API 調用中指定的其他字段的詳細信息，請參閱發佈設定參考和 AWS SDK 或其他 API 客戶端的 API 參考文檔。

建立 Kinesis Data Streams 取用程式

若要讀取和分析您的即時記錄，您可以建置或使用 Kinesis Data Streams 取用程式。當您建立 CloudFront 即時記錄檔的取用者時，請務必瞭解每個即時記錄檔記錄中的欄位一律以相同的順序傳遞，如欄位區段所列。請確定建立您的取用程式來適應此固定訂單。

例如，假設只包含下列三個欄位的即時日誌組態： `time-to-first-byte`、`sc-status` 和 `c-country`。在這個案例中，最後一個欄位 `c-country`，永遠是每個日誌中的欄位編號 3。不過，如果您稍後將欄位新增至即時日誌組態，則日誌中每個欄位的位置可能會變更。

例如，如果您新增欄位 `sc-bytes` 和 `time-taken` 即時日誌組態，這些欄位會根據 欄位 區段中顯示的順序插入到每個日誌中。所有五個欄位的產生順序為 `time-to-first-byte`、`sc-status`、`sc-bytes`、`time-taken` 和 `c-country`。該 `c-country` 欄位原本是欄位編號 3，但現在是欄位編號 5。請確定您的取用者應用程式可以處理變更日誌中位置的欄位，以防您將欄位新增至即時日誌組態。

故障排除即時日誌

建立即時日誌組態之後，您可能會發現沒有任何日誌 (或不是所有日誌) 會交付至 Kinesis Data Streams。在這種情況下，您應該首先驗證您的 CloudFront 分發是否正在接收查看者請求。如果是，您可以檢查下列設定來繼續故障排除。

IAM 角色許可

若要將即時日誌記錄傳遞至 Kinesis 資料串流，請在即時記錄設定中 CloudFront 使用 IAM 角色。請確定角色信任原則和角色許可原則符合 IAM 角色 中顯示的原則。

Kinesis Data Streams 調節

如果 CloudFront 將即時日誌記錄寫入 Kinesis 資料串流的速度超過串流所能處理的速度，Kinesis Data Streams 可能會限制要求。 CloudFront在此情況下，您可以增加 Kinesis 資料串流中的碎片數量。每個碎片可支援最高每秒 1,000 筆記錄的寫入數目，最高每秒 1 MB 的資料寫入上限。

邊緣函數日誌

您可以使用 Amazon CloudWatch 日誌來獲取邊緣函數的日誌，包括 Lambda @Edge 和 CloudFront 函數。使用 CloudWatch控制台或日 CloudWatch 誌 API 訪問日誌。

> **Important**
>
> 我們建議您使用這些記錄來瞭解內容要求的性質，而不是對所有要求進行完整記錄。CloudFront 以最大的努力提供邊緣功能日誌。在實際處理請求之後，才可能長時間交付特定請求的日誌項目，在極少數的情況下，有可能完全不會交付日誌項目。從邊緣函數日誌省略日誌項目時，邊緣函數日誌中的項目數量與顯示於 AWS 帳單和使用量報告中的用量會不相符。

Lambda @Edge 日誌

Lambda @Edge 會自動將函數記錄傳送至 CloudWatch 記錄檔，並在執行函數的 AWS 區域 位置建立日誌串流。日誌群組名稱的格式為`/aws/lambda/us-east-1.`*`function-name`*，其中*`function-name`*是您在建立函數時提供給函數的名稱，也`us-east-1`是建立函數的 AWS 區域 地區碼。記錄群組名稱一律包含`us-east-1`，即使是您的函數執行的其他區域的記錄群組也一樣。

> **Note**
>
> Lambda@Edge 根據請求數量和日誌的大小節流日誌。

您必須檢閱正確的 CloudWatch 記錄檔， AWS 區域 才能查看 Lambda @Edge 函數記錄檔。若要查看 Lambda @Edge 函數執行的區域，請在 CloudFront 主控台中檢視函數的指標圖形。指標將依每個 AWS 區域進行顯示。在同一頁面中，您可以選擇一個區域，然後檢視該區域的日誌檔以調查問題。

若要進一步了解如何搭配 Lambda @Edge 函數使用 CloudWatch 日誌，請參閱下列內容：

- 如需有關在主控台的 [監視] 區段中檢視圖形的詳細資 CloudFront 訊，請參閱the section called “使用 Amazon CloudFront 監控指標 CloudWatch”。

- 如需將資料傳送至 CloudWatch 記錄檔所需權限的相關資訊，請參閱the section called “設定 IAM 許可和角色”。
- 如需將記錄新增至 Lambda@Edge 函數的資訊，請參閱 AWS Lambda 開發人員指南中的 Node.js 中的AWS Lambda 函數日誌記錄或 Python 中的AWS Lambda 函數日誌記錄。
- 如需 CloudWatch 日誌配額 (先前稱為限制) 的相關資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的 CloudWatch 日誌配額。

CloudFront 函數日誌

如果 CloudFront 函數的代碼包含`console.log()`語句， CloudFront Functions 會自動將這些日誌行發送到 CloudWatch 日誌。如果沒有`console.log()`陳述式，則不會將任何資料傳送至 CloudWatch 記錄檔。

CloudFront 函數一律會在美國東部 (維吉尼亞北部) 區域 (`us-east-1`) 建立日誌串流，無論執行函式的節點為何。日誌群組名稱的格式為 `/aws/cloudfront/function/`*`FunctionName`*，其中 *`FunctionName`* 是您在建立函數時提供給函數的名稱。日誌串流名稱的格式為 `YYYY/M/D/`*`UUID`*。

以下顯示傳送至 CloudWatch 記錄檔的記錄檔訊息範例。每一行都以唯一識別 CloudFront 請求的 ID 開頭。訊息以包含 CloudFront 發佈 ID 的`START`行開頭，並以一`END`行結尾。`START` 和 `END` 行之間是函數中 `console.log()` 陳述式產生的日誌行。

```
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== START DistributionID:
 E3E5D42GADAXZZ
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== Example function log output
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== END
```

> **Note**
>
> CloudFront Functions CloudWatch 只會針對`LIVE`階段中為回應生產要求和回應而執行的函數傳送記錄檔。當您測試函數時， CloudFront 不會將任何日誌發送到 CloudWatch. 測試輸出包含有關錯誤、計算使用率和函數記錄檔 (`console.log()`陳述式) 的資訊，但這項資訊不會傳送至 CloudWatch。

CloudFront 函數使用 AWS Identity and Access Management (IAM) 服務連結角色，將記錄傳送到您帳戶中的 CloudWatch Logs。服務連結角色是直接連結至 AWS 服務的 IAM 角色。服務連結角色由服務預先定義，並包含服務代表您呼叫其他 AWS 服務所需的所有權限。 CloudFront 函數使用稱

為AWSServiceRoleForCloudFrontLogger的服務連結角色。如需有關此角色的詳細資訊，請參閱 the section called “Lambda@Edge 的服務連結角色” (Lambda@Edge 會使用相同的服務連結角色)。

當函數因驗證錯誤或執行錯誤而失敗時，資訊會記錄在 CloudFront標準記錄和即時記錄中。錯誤的相關資訊會記錄在 `x-edge-result-type`、`x-edge-response-result-type` 和 `x-edge-detailed-result-type` 欄位中。

使用記錄 Amazon CloudFront API 呼叫 AWS CloudTrail

CloudFront 與AWS CloudTrail提供使用者、角色或 AWS 服務. CloudTrail 擷取 CloudFront 作為事件的所有 API 呼叫。擷取的呼叫包括來自 CloudFront 主控台的呼叫和 CloudFront API 作業的程式碼呼叫。使用收集的資訊 CloudTrail，您可以判斷提出的要求 CloudFront、提出要求的 IP 位址、提出要求的時間，以及其他詳細資訊。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是使用根使用者還是使用者憑證提出。
- 請求是否代表 IAM 身分中心使用者提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務服務提出。

CloudTrail 在您創建帳戶 AWS 帳戶 時處於活動狀態，並且您自動可以訪問 CloudTrail 事件歷史記錄。 CloudTrail 事件歷史記錄提供了過去 90 天的記錄管理事件的可查看，可搜索，可下載和不可變的記錄。 AWS 區域若要取得更多資訊，請參閱《使用指南》中的〈AWS CloudTrail 使用 CloudTrail 事件歷程〉。查看活動歷史記錄不 CloudTrail 收取任何費用。

如需過 AWS 帳戶 去 90 天內持續的事件記錄，請建立追蹤或 CloudTrailLake 事件資料存放區。

CloudTrail 小徑

追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。使用建立的所有系統線 AWS Management Console 都是多區域。您可以使用建立單一區域或多區域系統線。 AWS CLI建議您建立多區域追蹤，因為您會擷取帳戶 AWS 區域 中的所有活動。如果您建立單一區域追蹤，則只能檢視追蹤記錄中的 AWS 區域事件。如需有關追蹤的詳細資訊，請參閱《AWS CloudTrail 使用指南》中的「為您的建立追蹤」 AWS 帳戶和「為組織建立追蹤」。

您可以透 CloudTrail 過建立追蹤，免費將一份正在進行的管理事件副本傳遞到 Amazon S3 儲存貯體，但是需要支付 Amazon S3 儲存費用。如需有關 CloudTrail 定價的詳細資訊，請參閱AWS CloudTrail 定價。如需 Amazon S3 定價的相關資訊，請參閱 Amazon S3 定價。

CloudTrail 湖泊事件資料存放區

CloudTrail L@@ ake 可讓您針對事件執行 SQL 型查詢。 CloudTrail 湖將基於行的 JSON 格式的現有事件轉換為 Apache ORC 格式。ORC 是一種單欄式儲存格式，針對快速擷取資料進行了最佳化。系統會將事件彙總到事件資料存放區中，事件資料存放區是事件的不可變集合，其依據為您透過套用進階事件選取器選取的條件。套用於事件資料存放區的選取器控制哪些事件持續存在並可供您查詢。若要取得有關 CloudTrail Lake 的更多資訊，請參閱使用指南中的〈AWS CloudTrail 使用 AWS CloudTrail Lake〉。

CloudTrail Lake 事件資料存放區和查詢會產生費用。建立事件資料存放區時，您可以選擇要用於事件資料存放區的定價選項。此定價選項將決定擷取和儲存事件的成本，以及事件資料存放區的預設和最長保留期。如需有關 CloudTrail 定價的詳細資訊，請參閱AWS CloudTrail 定價。

> **Note**
>
> CloudFront 是一項全球性的服務。 CloudTrail 記錄美國東部 (維吉尼亞北部) 區域的事件。
> CloudFront 如需詳細資訊，請參閱AWS CloudTrail 使用指南中的全域服務事件。
> 如果您使用臨時安全登入資料 AWS Security Token Service，則對區域端點的呼叫 (例如`us-west-2`) 會登 CloudTrail 入其適當的區域。
> 如需有關 CloudFront 端點的詳細資訊，請CloudFront 參閱 AWS 一般參考.

CloudFront 資料事件 CloudTrail

資料事件提供在資源上或在資源中執行之資源作業的相關資訊 (例如，讀取或寫入 CloudFront發佈)。這些也稱為資料平面操作。資料事件通常是大量資料的活動。依預設， CloudTrail不會記錄資料事件。CloudTrail 事件歷史記錄不會記錄數據事件。

資料事件需支付額外的費用。如需有關 CloudTrail 定價的詳細資訊，請參閱AWS CloudTrail 定價。

您可以使用 CloudTrail 主控台或 CloudTrail API 作業記錄 CloudFront 資源類型的資料事件。 AWS CLI有關如何記錄資料事件的詳細資訊，請參閱AWS CloudTrail 使用《使用指南》 AWS Command Line Interface中的記錄資料事件 AWS Management Console和記錄資料事件。

下表列出您可以記錄 CloudFront 資料事件的資源類型。[資料事件類型 (主控台)] 欄顯示可從主控台的 [資料事件類型 CloudTrail] 清單中選擇的值。resource .type 值欄會顯示**`resources.type`**值，您可以在使用或 API 設定進階事件選取器時指定這個值。 AWS CLI CloudTrail 記錄到資料 CloudTrail欄中的資料 API 會顯示 CloudTrail 針對資源類型記錄的 API 呼叫。

資料事件類型 (主控台)	resources.type 值	記錄到的資料 API CloudTrail
CloudFront KeyValueStore	`AWS::CloudFront::KeyValueStore`	• DeleteKeys • DescribeKeyValueStore • GetKey • ListKeys • PutKeys • UpdateKeys

您可以設定進階事件選取器來篩選`eventName`、`readOnly`和`resources.ARN`欄位，以僅記錄對您很重要的事件。如需這些欄位的詳細資訊，請參閱 AWS CloudTrail API 參考AdvancedFieldSelector中的。

CloudFront 管理事件 CloudTrail

管理事件提供有關在您的資源上執行的管理作業的資訊 AWS 帳戶。這些也稱為控制平面操作。依預設，會 CloudTrail 記錄管理事件。

Amazon 將所有 CloudFront 控制平面操作 CloudFront 記錄為管理事件。如需記 CloudFront 錄到的 Amazon CloudFront 控制平面操作清單 CloudTrail，請參閱 Amazon CloudFront API 參考資料。

CloudFront 事件範例

事件代表來自任何來源的單一請求，並包括有關請求的 API 操作，操作的日期和時間，請求參數等信息。 CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此事件不會以任何特定順序顯示。

內容

- 範例：UpdateDistribution
- 範例：UpdateKeys

範例：UpdateDistribution

下列範例顯示示範UpdateDistribution作業的 CloudTrail 事件。

對於對 CloudFront API 的呼叫，`eventSource`則為`cloudfront.amazonaws.com`.

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:role-session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/role-session-name",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-02-02T19:23:50Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-02-02T19:26:01Z",
    "eventSource": "cloudfront.amazonaws.com",
    "eventName": "UpdateDistribution",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "52.94.133.137",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/121.0.0.0 Safari/537.36",
    "requestParameters": {
        "distributionConfig": {
            "defaultRootObject": "",
            "aliases": {
                "quantity": 3,
                "items": [
                    "alejandro_rosalez.awsps.myinstance.com",
                    "cross-testing.alejandro_rosalez.awsps.myinstance.com",
                    "*.alejandro_rosalez.awsps.myinstance.com"
                ]
            },
            "cacheBehaviors": {
                "quantity": 0,
                "items": []
            },
```

```
            "httpVersion": "http2and3",
            "originGroups": {
                "quantity": 0,
                "items": []
            },
            "viewerCertificate": {
                "minimumProtocolVersion": "TLSv1.2_2021",
                "cloudFrontDefaultCertificate": false,
                "aCMCertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "sSLSupportMethod": "sni-only"
            },
            "webACLId": "arn:aws:wafv2:us-east-1:111122223333:global/webacl/testing-
acl/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "customErrorResponses": {
                "quantity": 0,
                "items": []
            },
            "logging": {
                "includeCookies": false,
                "prefix": "",
                "enabled": false,
                "bucket": ""
            },
            "priceClass": "PriceClass_All",
            "restrictions": {
                "geoRestriction": {
                    "restrictionType": "none",
                    "quantity": 0,
                    "items": []
                }
            },
            "isIPV6Enabled": true,
            "callerReference": "1578329170895",
            "continuousDeploymentPolicyId": "",
            "enabled": true,
            "defaultCacheBehavior": {
                "targetOriginId": "d111111abcdef8",
                "minTTL": 0,
                "compress": false,
                "maxTTL": 31536000,
                "functionAssociations": {
                    "quantity": 0,
                    "items": []
```

```
                },
                "trustedKeyGroups": {
                    "quantity": 0,
                    "items": [],
                    "enabled": false
                },
                "smoothStreaming": false,
                "fieldLevelEncryptionId": "",
                "defaultTTL": 86400,
                "lambdaFunctionAssociations": {
                    "quantity": 0,
                    "items": []
                },
                "viewerProtocolPolicy": "redirect-to-https",
                "forwardedValues": {
                    "cookies": {"forward": "none"},
                    "queryStringCacheKeys": {
                        "quantity": 0,
                        "items": []
                    },
                    "queryString": false,
                    "headers": {
                        "quantity": 1,
                        "items": ["*"]
                    }
                },
                "trustedSigners": {
                    "items": [],
                    "enabled": false,
                    "quantity": 0
                },
                "allowedMethods": {
                    "quantity": 2,
                    "items": [
                        "HEAD",
                        "GET"
                    ],
                    "cachedMethods": {
                        "quantity": 2,
                        "items": [
                            "HEAD",
                            "GET"
                        ]
                    }
```

```
                }
            },
            "staging": false,
            "origins": {
                "quantity": 1,
                "items": [
                    {
                        "originPath": "",
                        "connectionTimeout": 10,
                        "customOriginConfig": {
                            "originReadTimeout": 30,
                            "hTTPSPort": 443,
                            "originProtocolPolicy": "https-only",
                            "originKeepaliveTimeout": 5,
                            "hTTPPort": 80,
                            "originSslProtocols": {
                                "quantity": 3,
                                "items": [
                                    "TLSv1",
                                    "TLSv1.1",
                                    "TLSv1.2"
                                ]
                            }
                        },
                        "id": "d111111abcdef8",
                        "domainName": "d111111abcdef8.cloudfront.net",
                        "connectionAttempts": 3,
                        "customHeaders": {
                            "quantity": 0,
                            "items": []
                        },
                        "originShield": {"enabled": false},
                        "originAccessControlId": ""
                    }
                ]
            },
            "comment": "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "id": "EDFDVBD6EXAMPLE",
        "ifMatch": "E1RTLUR9YES76O"
    },
    "responseElements": {
        "distribution": {
            "activeTrustedSigners": {
```

```
                "quantity": 0,
                "enabled": false
            },
            "id": "EDFDVBD6EXAMPLE",
            "domainName": "d111111abcdef8.cloudfront.net",
            "distributionConfig": {
                "defaultRootObject": "",
                "aliases": {
                    "quantity": 3,
                    "items": [
                        "alejandro_rosalez.awsps.myinstance.com",
                        "cross-testing.alejandro_rosalez.awsps.myinstance.com",
                        "*.alejandro_rosalez.awsps.myinstance.com"
                    ]
                },
                "cacheBehaviors": {"quantity": 0},
                "httpVersion": "http2and3",
                "originGroups": {"quantity": 0},
                "viewerCertificate": {
                    "minimumProtocolVersion": "TLSv1.2_2021",
                    "cloudFrontDefaultCertificate": false,
                    "aCMCertificateArn": "arn:aws:acm:us-
east-1:111122223333:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                    "sSLSupportMethod": "sni-only",
                    "certificateSource": "acm",
                    "certificate": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
                "webACLId": "arn:aws:wafv2:us-east-1:111122223333:global/webacl/
testing-acl/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
                "customErrorResponses": {"quantity": 0},
                "logging": {
                    "includeCookies": false,
                    "prefix": "",
                    "enabled": false,
                    "bucket": ""
                },
                "priceClass": "PriceClass_All",
                "restrictions": {
                    "geoRestriction": {
                        "restrictionType": "none",
                        "quantity": 0
                    }
                },
```

```
                "isIPV6Enabled": true,
                "callerReference": "1578329170895",
                "continuousDeploymentPolicyId": "",
                "enabled": true,
                "defaultCacheBehavior": {
                    "targetOriginId": "d111111abcdef8",
                    "minTTL": 0,
                    "compress": false,
                    "maxTTL": 31536000,
                    "functionAssociations": {"quantity": 0},
                    "trustedKeyGroups": {
                        "quantity": 0,
                        "enabled": false
                    },
                    "smoothStreaming": false,
                    "fieldLevelEncryptionId": "",
                    "defaultTTL": 86400,
                    "lambdaFunctionAssociations": {"quantity": 0},
                    "viewerProtocolPolicy": "redirect-to-https",
                    "forwardedValues": {
                        "cookies": {"forward": "none"},
                        "queryStringCacheKeys": {"quantity": 0},
                        "queryString": false,
                        "headers": {
                            "quantity": 1,
                            "items": ["*"]
                        }
                    },
                    "trustedSigners": {
                        "enabled": false,
                        "quantity": 0
                    },
                    "allowedMethods": {
                        "quantity": 2,
                        "items": [
                            "HEAD",
                            "GET"
                        ],
                        "cachedMethods": {
                            "quantity": 2,
                            "items": [
                                "HEAD",
                                "GET"
                            ]
```

```
                    }
                }
            },
            "staging": false,
            "origins": {
                "quantity": 1,
                "items": [
                    {
                        "originPath": "",
                        "connectionTimeout": 10,
                        "customOriginConfig": {
                            "originReadTimeout": 30,
                            "hTTPSPort": 443,
                            "originProtocolPolicy": "https-only",
                            "originKeepaliveTimeout": 5,
                            "hTTPPort": 80,
                            "originSslProtocols": {
                                "quantity": 3,
                                "items": [
                                    "TLSv1",
                                    "TLSv1.1",
                                    "TLSv1.2"
                                ]
                            }
                        },
                        "id": "d111111abcdef8",
                        "domainName": "d111111abcdef8.cloudfront.net",
                        "connectionAttempts": 3,
                        "customHeaders": {"quantity": 0},
                        "originShield": {"enabled": false},
                        "originAccessControlId": ""
                    }
                ]
            },
            "comment": "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "aliasICPRecordals": [
            {
                "cNAME": "alejandro_rosalez.awsps.myinstance.com",
                "iCPRecordalStatus": "APPROVED"
            },
            {
                "cNAME": "cross-testing.alejandro_rosalez.awsps.myinstance.com",
                "iCPRecordalStatus": "APPROVED"
```

```
                },
                {
                    "cNAME": "*.alejandro_rosalez.awsps.myinstance.com",
                    "iCPRecordalStatus": "APPROVED"
                }
            ],
            "aRN": "arn:aws:cloudfront::111122223333:distribution/EDFDVBD6EXAMPLE",
            "status": "InProgress",
            "lastModifiedTime": "Feb 2, 2024 7:26:01 PM",
            "activeTrustedKeyGroups": {
                "enabled": false,
                "quantity": 0
            },
            "inProgressInvalidationBatches": 0
        },
        "eTag": "E1YHBLAB2BJY1G"
    },
    "requestID": "4e6b66f9-d548-11e3-a8a9-73e33example",
    "eventID": "5ab02562-0fc5-43d0-b7b6-90293example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "apiVersion": "2020_05_31",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "cloudfront.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

範例：UpdateKeys

下列範例顯示示範UpdateKeys作業的 CloudTrail 事件。

對於對 CloudFront KeyValueStore API 的調用，`edgekeyvaluestore.amazonaws.com`而不`eventSource`是`cloudfront.amazonaws.com`.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
```

```
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:role-session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/role-session-name",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2023-11-01T23:41:14Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-11-01T23:41:28Z",
    "eventSource": "edgekeyvaluestore.amazonaws.com",
    "eventName": "UpdateKeys",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "3.235.183.252",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/121.0.0.0 Safari/537.36,
    "requestParameters": {
        "kvsARN": "arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "ifMatch": "KV3O6B1CX531EBP",
        "deletes": [
            {"key": "key1"}
        ]
    },
    "responseElements": {
        "itemCount": 0,
        "totalSizeInBytes": 0,
        "eTag": "KVDC9VEVZ71ZGO"
    },
    "requestID": "5ccf104c-acce-4ea1-b7fc-73e33example",
    "eventID": "a0b1b5c7-906c-439d-9925-90293example",
    "readOnly": false,
    "resources": [
        {
```

```
            "accountId": "111122223333",
            "type": "AWS::CloudFront::KeyValueStore",
            "ARN": "arn:aws:cloudfront::111122223333:key-value-store/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "111122223333.cloudfront-kvs.global.api.aws"
    }
}
```

若要取得有關 CloudTrail 記錄內容的資訊，請參閱AWS CloudTrail 使用指南中的CloudTrail 記錄內容。

追蹤組態變更 AWS Config

用 AWS Config 於記錄 CloudFront 發佈設定的組態變更。您可以擷取發佈狀態、價格類別、起源地、地理限制設定和 Lambda @Edge 組態的變更。

> **Note**
>
> AWS Config 不會記錄 CloudFront 串流分發的索引鍵值標籤。

設定 AWS Config 方式 CloudFront

設定時 AWS Config，您可以選擇記錄所有支援的 AWS 資源，或僅記錄某些指定的資源，例如 CloudFront 僅記錄變更。如需支援CloudFront 資源的清單，請參閱AWS Config 開發人員指南中受支援的資源類型主題的 Amazon CloudFront 章節。

若要追蹤 CloudFront 發行版的組態變更，您必須登入美國東部 (維吉尼亞北部) 的 CloudFront 主控台 AWS 區域。

> **Note**
>
> 使用記錄資源可能會有延遲 AWS Config。 AWS Config 只有在發現資源後才記錄資源。

Console

AWS Config 使用 CloudFront （控制台）進行設置

1. 請登入 AWS Management Console 並開啟 AWS Config 主控台，網址為 https://console.aws.amazon.com/config/。
2. 選擇 Get Started Now (立即開始)。
3. 在 [設定] 頁面上，對於要記錄的資 AWS 源類型，指定您 AWS Config 要記錄的資源類型。如果您只想記錄 CloudFront 變更，請選擇 [特定類型]，然後在下CloudFront方選擇要追蹤變更的散佈或串流分發。

   若要新增或變更想追蹤的分佈，請在完成初始設定後，選擇左側的 Settings (設定)。
4. 指定下列項目的其他必要選項 AWS Config：設定通知、指定組態資訊的位置，以及新增評估資源類型的規則。

如需詳細資訊，請參閱AWS Config 開發人員指南中的 AWS Config 使用主控台進行設定。

AWS CLI

若要 CloudFront 使 AWS Config 用進行設定 AWS CLI，請參閱AWS Config 開發人員指南中 AWS Config 的使用 AWS CLI 進行設定。

AWS Config API

若要 CloudFront 使 AWS Config 用 API 進行設定，請參閱 AWS Config AWS Config API 參考中的StartConfigurationRecorder動作和其他資訊。

檢視 CloudFront 組態歷史記

AWS Config 開始記錄發行版的配置更改後，您可以獲取已配置的任何發行版的配置歷史記錄 CloudFront。

您可以使用下列方式檢視模型組態歷史記錄。

Console

對於每個記錄的資源，您可以檢視提供組態詳細資訊歷史記錄的時間表頁面。若要查看此頁面，請選擇專用執行個體頁面中的 Config Timeline (組態時間軸) 欄內的灰色圖示。

如需詳細資訊，請參閱AWS Config 開發人員指南中的檢視 AWS Config 主控台中的組態詳細資料。

AWS CLI

若要取得所有發行版的清單，請執行list-discovered-resources命令，如下列範例所示。

```
aws configservice list-discovered-resources --resource-type
 AWS::CloudFront::Distribution
```

若要取得特定時間間隔內發行版的組態詳細資料，請執行get-resource-config-history命令。

如需詳細資訊，請參閱AWS Config 開發人員指南中的使用 CLI 檢視組態詳細資訊。

AWS Config API

若要取得所有發行版的清單，請使用ListDiscoveredResources動作。

若要取得特定時間間隔內發佈的組態詳細資料，請使用GetResourceConfigHistory動作。如需詳細資訊，請參閱 AWS Config API 參考。

Amazon 中的安全 CloudFront

雲端安全是 AWS 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。

安全是 AWS 與您共同的責任。共同的責任模型 將此描述為雲端 本身 的安全和雲端內部的安全：

- 雲端本身的安全 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。在 AWS 合規計畫中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon 的合規計劃 CloudFront，請參閱合規計劃適用範圍的AWS服務。
- 雲端內部的安全：您的責任取決於所使用的 AWS 服務。您也必須對資料敏感度、組織要求，以及適用法律和法規等其他因素負責。

本文件可協助您瞭解如何在使用時套用共同責任模型 CloudFront。下列主題說明如何設定 CloudFront 以符合安全性與合規性目標。您還將學習如何使用其他AWS服務來幫助您監控和保護您的 CloudFront 資源。

主題

- Amazon 的數據保護 CloudFront
- Amazon Identity and Access Management CloudFront
- Amazon 中的記錄和監控 CloudFront
- Amazon 的合規驗證 CloudFront
- Amazon 的韌性 CloudFront
- Amazon 基礎設施安全 CloudFront

Amazon 的數據保護 CloudFront

AWS共同責任模型適用於 Amazon 中的資料保護 CloudFront。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您負責維護在此基礎設施上託管內容的控制權。您也必須負責您所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的相關資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 憑證，並設定個人使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 AWS CloudTrail 設定 API 和使用者活動日誌記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務 內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需 FIPS 和 FIPS 端點的相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API CloudFront 或 AWS SDK 時AWS 服務使用或其他使用時。AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

Amazon CloudFront 提供數個選項，您可以使用這些選項來協助保護其交付的內容：

- 設定 HTTPS 連線。
- 設定欄位層級加密，以在傳輸期間為特定資料提供額外的安全性。
- 限制存取內容，以使特定人員或特定區域人員才能檢視內容。

下列主題說明了關於選項的詳細資訊。

主題

傳輸中加密

若要在傳輸期間加密資料，您可以 CloudFront 將 Amazon 設定為要求檢視者使用 HTTPS 請求您的檔案，以便在與檢視者 CloudFront通訊時加密連線。您也可以設定 CloudFront 使用 HTTPS 從您的來源取得檔案，以便在與原始伺服器 CloudFront 通訊時加密連線。

如需詳細資訊，請參閱 搭配使用 HTTPS CloudFront。

欄位層級加密與 HTTPS 一起新增額外的安全層，可讓您在整個系統處理過程中保護特定的資料，以便只有特定應用程式才能看到它。透過在中設定欄位層級加密 CloudFront，您可以安全地將使用者提交的敏感資訊上傳到 Web 伺服器。您用戶端提供的敏感資訊會在更靠近使用者節點時進行加密。該敏感資訊會在整個應用程式堆疊中保持加密，以確保只有需要資料 (而且具有將資料解密的登入資料) 的應用程式才能夠這樣做。

如需詳細資訊，請參閱 使用欄位層級加密來協助保護敏感資料。

CloudFront API 端點`cloudfront-fips.amazonaws.com`，`cloudfront.amazonaws.com`而且只接受 HTTPS 流量。這表示當您使用 CloudFront API 傳送和接收資訊時，您的資料 (包括散發設定、快取原則和來源要求原則、金鑰群組和公開金鑰，以及 CloudFront 功能中的函數程式碼) 一律會在傳輸過程中加密。此外，所有傳送至 CloudFront API 端點的要求都會使用AWS認證簽署並登入AWS CloudTrail。

函數中的函數代碼和配置在 CloudFront 傳輸過程中，當複製到邊緣位置存在點（PoP）以及其他使用的存儲位置之間時，始終會加密 CloudFront。

靜態加密

函數中 CloudFront 的函數代碼和配置始終以加密格式存儲在邊緣位置 PoP 上，以及使用的其他存儲位置 CloudFront。

限制存取內容

許多透過網際網路分佈內容的公司想要限制使用者子集的文件、業務資料、媒體串流，或內容的存取許可。若要使用 Amazon 安全地提供此內容 CloudFront，您可以執行下列一或多項操作：

使用簽章的 URL 或 Cookie

> 您可以透過使用已簽署的 URL 或簽署的 Cookie 來提供此私人內容，以限制針對特定使用者 (例如付費的 CloudFront 使用者) 的存取權。如需詳細資訊，請參閱 使用已簽署的 URL 和已簽署的 Cookie 提供私有內容。

限制對 Amazon S3 儲存貯體中內容的存取

> 如果您使用簽署的 URL 或 CloudFront 已簽署的 Cookie 來限制對內容的存取，您也不會希望使用者使用檔案的直接 URL 來檢視檔案。反之，您想要他們僅使用 CloudFront URL 來存取檔案，以達保護之目的。

如果您使用 Amazon S3 儲存貯體做為 CloudFront 分發的來源，您可以設定來源存取控制 (OAC)，以便限制對 S3 儲存貯體的存取。如需詳細資訊，請參閱 the section called “限制對 Amazon 簡單儲存服務來源的存取”。

限制存取 Application Load Balancer 所提供的內容

當您將 Elastic Load Balancing 中的應用 CloudFront 程式負載平衡器作為來源使用時，您可以設定 CloudFront 為防止使用者直接存取 Application Load Balancer。這讓使用者只能透過以下方式存取 Application Load Balancer CloudFront，確保您獲得使用的好處 CloudFront。如需詳細資訊，請參閱 限制對 Application Load Balancers 的存取。

使用 AWS WAF Web ACL

您可以使用 Web 應用程式防火牆服務 AWS WAF，以建立 Web 存取控制清單 (Web ACL) 來限制存取您的內容。根據您指定的條件 (例如要求來源的 IP 位址或查詢字串的值) 會以要求的內容或 HTTP 403 狀態碼 (禁止) CloudFront 回應要求。如需詳細資訊，請參閱 使用 AWS WAF 保護。

使用地理限制

您可以使用地理限制功能 (也稱為地理封鎖)，來防止特定地理位置的使用者，存取您透過 CloudFront 分佈所提供的內容。在設定地理限制時，您有幾個選項可選擇。如需詳細資訊，請參閱 限制您內容的地理分佈。

Amazon Identity and Access Management CloudFront

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員控制哪些人可以通過身份驗證 (登入) 和授權 (具有權限) 來使用 CloudFront 資源。您可以使用 IAM AWS 服務 ，無需額外付費。

主題

- 物件
- 使用身分驗證
- 使用政策管理存取權
- Amazon 如何與 IAM 合 CloudFront 作
- Amazon 的基於身份的政策示例 CloudFront
- AWSAmazon 的受管政策 CloudFront
- 疑難排解 Amazon CloudFront 身分和存取

物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 CloudFront。

服務使用者 — 如果您使用 CloudFront 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 CloudFront 功能來完成工作時，您可能需要其他權限。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果無法存取中的圖徵 CloudFront，請參閱疑難排解 Amazon CloudFront 身分和存取。

服務管理員 — 如果您負責公司的 CloudFront 資源，您可能擁有完整的存取權 CloudFront。決定您的服務使用者應該存取哪些 CloudFront 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步瞭解貴公司如何搭配使用 IAM CloudFront，請參閱Amazon 如何與 IAM 合 CloudFront 作。

IAM 管理員 — 如果您是 IAM 管理員，您可能想要瞭解如何撰寫政策來管理存取權限的詳細資訊 CloudFront。若要檢視可在 IAM 中使用的 CloudFront 基於身分的政策範例，請參閱。Amazon 的基於身份的政策示例 CloudFront

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》 AWS 帳戶中的如何登入您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時登入資料進行存取 AWS 服務 。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identi ty Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。 AWS Directory Service同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，在中暫時擔任 IAM 角色。 AWS Management Console 您可以透過呼叫 AWS CLI 或

AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源類型政策的差異。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。
  - 服務連結角色 — 服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。 AWS WAF若要進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 實體許可範圍。
- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

Amazon 如何與 IAM 合 CloudFront 作

在您使用 IAM 管理存取權限之前 CloudFront，請先了解哪些 IAM 功能可搭配使用 CloudFront。

您可以與 Amazon 搭配使用的 IAM 功能 CloudFront

IAM 功能	CloudFront 支持
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC(政策中的標籤)	部分
臨時憑證	是
轉送存取工作階段 (FAS)	否
服務角色	否
服務連結角色	是

若要深入瞭解如何以 CloudFront 及其他 AWS 服務如何使用大多數 IAM 功能，請參閱 IAM 使用者指南中的搭配 IAM 使用的AWS 服務。

以身分識別為基礎的原則 CloudFront

支援身分型政策	是

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

以身分識別為基礎的原則範例 CloudFront

若要檢視以 CloudFront 身分為基礎的原則範例，請參閱。Amazon 的基於身份的政策示例 CloudFront

以資源為基礎的政策 CloudFront

支援以資源基礎的政策	否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

若要啟用跨帳戶存取，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 角色與資源型政策有何差異。

的政策動作 CloudFront

支援政策動作	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 CloudFront 動作清單，請參閱服務授權參考 CloudFront中 Amazon 定義的動作。

中的策略動作在動作之前 CloudFront 使用下列前置詞：

```
cloudfront
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
      "cloudfront:action1",
      "cloudfront:action2"
         ]
```

若要檢視以 CloudFront 身分為基礎的原則範例，請參閱。Amazon 的基於身份的政策示例 CloudFront

的政策資源 CloudFront

支援政策資源	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

要查看 CloudFront 資源類型及其 ARN 的列表，請參閱服務授權參考 CloudFront中由 Amazon 定義的資源。若要了解可以使用哪些動作指定每個資源的 ARN，請參閱 Amazon CloudFront 定義的動作。

若要檢視以 CloudFront 身分為基礎的原則範例，請參閱。Amazon 的基於身份的政策示例 CloudFront

的政策條件索引鍵 CloudFront

支援服務特定政策條件金鑰	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯OR運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的AWS 全域條件內容金鑰。

若要查看 CloudFront 條件金鑰清單，請參閱服務授權參考 CloudFront中的 Amazon 條件金鑰。若要了解您可以使用條件金鑰的動作和資源，請參閱 Amazon 定義的動作 CloudFront。

若要檢視以 CloudFront 身分為基礎的原則範例，請參閱。Amazon 的基於身份的政策示例 CloudFront

ACL 在 CloudFront

支援 ACL	否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

阿巴克與 CloudFront

支援 ABAC (政策中的標籤)	部分

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件金鑰，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的什麼是 ABAC?。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的使用屬性型存取控制 (ABAC)。

CloudFront 僅支持 ABAC 的發行版。

使用臨時登入資料 CloudFront

支援臨時憑證	是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料搭配AWS 服務 使用，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的切換至角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而非使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

轉寄存取工作階段 CloudFront

支援轉寄存取工作階段 (FAS)	否

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求

AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。

CloudFront 的服務角色

支援服務角色	否

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。

> **Warning**
>
> 變更服務角色的權限可能會中斷 CloudFront 功能。只有在 CloudFront 提供指引時才編輯服務角色。

服務連結角色 CloudFront

支援服務連結角色	是

服務連結角色是一種連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

Lambda @Edge 會使用服務連結角色為您執行動作。如需建立或管理 CloudFront 服務連結角色的詳細資訊，請參閱Lambda@Edge 的服務連結角色。

如需建立或管理服務連結角色的詳細資訊，請參閱可搭配 IAM 運作的AWS 服務。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

Amazon 的基於身份的政策示例 CloudFront

依預設，使用者和角色沒有建立或修改 CloudFront資源的權限。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

有關由定義的動作和資源類型的詳細資訊 CloudFront，包括每種資源類型的 ARN 格式，請參閱服務授權參考 CloudFront中適用於 Amazon 的動作、資源和條件金鑰。

主題

政策最佳實務

以身分識別為基礎的政策會決定某人是否可以建立、存取或刪除您帳戶中的 CloudFront 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始將權限授與使用者和工作負載，請使用可授與許多常見使用案例權限的AWS 受管理原則。它們在您的 AWS 帳戶. 建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需更多資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。

- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫 API 作業時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用 CloudFront 主控台

若要存取 Amazon CloudFront 主控台，您必須擁有最少一組許可。這些權限必須允許您列出和檢視有關 AWS 帳戶. CloudFront 如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

若要確保使用者和角色仍可使用 CloudFront 主控台，請同時將 CloudFront *ConsoleAccess*或受*ReadOnly* AWS 管理的原則附加至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudFront 以編程方式訪問權限

以下說明許可政策。Sid (陳述式 ID) 為選用。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowAllCloudFrontPermissions",
         "Effect": "Allow",
         "Action": ["cloudfront:*"],
         "Resource": "*"
      }
   ]
}
```

此原則會授與執行所有 CloudFront 作業的權限，這足 CloudFront 以透過程式設計方式存取。如果您使用主控台存取 CloudFront，請參閱使用 CloudFront 主控台所需的權限。

如需您指定授與或拒絕使用每個動作之權限的動作清單和 ARN，請參閱服務授權參考 CloudFront中適用於 Amazon 的動作、資源和條件金鑰。

使用 CloudFront 主控台所需的權限

若要授與 CloudFront 主控台的完整存取權，請在下列權限原則中授與權限：

```
{
```

```
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "acm:ListCertificates",
            "cloudfront:*",
            "cloudwatch:DescribeAlarms",
            "cloudwatch:PutMetricAlarm",
            "cloudwatch:GetMetricStatistics",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets",
            "s3:PutBucketPolicy"
         ],
         "Resource":"arn:aws:s3:::*"
      }
   ]
}
```

需要許可的原因如下：

`acm:ListCertificates`

當您使用 CloudFront 主控台建立和更新發行版，並且想 CloudFront 要設定為在檢視器和原始位置之間 CloudFront 或之間 CloudFront 需要 HTTPS 時，可讓您檢視 ACM 憑證的清單。

如果您不使用 CloudFront 主機，則不需要此權限。

`cloudfront:*`

可讓您執行所有 CloudFront 動作。

`cloudwatch:DescribeAlarms` 和 `cloudwatch:PutMetricAlarm`

讓您在 CloudFront 控制台中創建和查看 CloudWatch 警報。另請參閱 `sns:ListSubscriptionsByTopic` 和 `sns:ListTopics`。

如果您不使用 CloudFront主機，則不需要這些權限。

`cloudwatch:GetMetricStatistics`

讓我們在 CloudFront 控制台中 CloudFront 渲染 CloudWatch 指標。

如果您不使用 CloudFront 主機，則不需要此權限。

`elasticloadbalancing:DescribeLoadBalancers`

在建立和更新分佈時，許可會讓您可在可用的原始伺服器清單中查看 Elastic Load Balancing 負載平衡器清單。

如果您不使用 CloudFront 主機，則不需要此權限。

`iam:ListServerCertificates`

當您使用 CloudFront 主控台建立和更新發佈，並且想 CloudFront 要設定為在檢視器和來源之間 CloudFront 或之間 CloudFront 需要 HTTPS 時，可讓您檢視 IAM 憑證存放區中的憑證清單。

如果您不使用 CloudFront 主機，則不需要此權限。

`s3:ListAllMyBuckets`

當您建立和更新分佈，可以執行以下操作：

- 檢視在可用的原始伺服器清單中的 S3 儲存貯體清單
- 檢視可以保存存取記錄的 S3 儲存貯體清單

如果您不使用 CloudFront 主機，則不需要此權限。

`S3:PutBucketPolicy`

當您建立或更新限制 S3 儲存貯體存取權的發佈時，可讓使用者更新儲存貯體政策以授與 CloudFront原始存取身分的存取權。如需詳細資訊，請參閱 the section called “使用原始存取身分 (舊版，不建議使用)”。

如果您不使用 CloudFront 主機，則不需要此權限。

`sns:ListSubscriptionsByTopic` 和 `sns:ListTopics`

當您在主 CloudFront 控台中建立 CloudWatch 警示時，可讓您選擇通知的 SNS 主題。

如果您不使用 CloudFront主控台，則不需要這些權限。

`waf:GetWebACL` 和 `waf:ListWebACLs`

可讓您在 CloudFront 主控台中檢視 AWS WAF Web ACL 清單。

如果您不使用 CloudFront主控台，則不需要這些權限。

AWS 的管理 (預先定義) 策略 CloudFront

AWS 透過提供由建立和管理的獨立 IAM 政策來解決許多常見使用案例 AWS。這些 AWS 受管理的政策會為常見使用案例授與必要的權限，因此您可以避免調查需要哪些權限。如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。針對 CloudFront IAM 提供兩種受管政策：

- CloudFrontFullAccess— 授予對 CloudFront 資源的完全訪問權限。

> **Important**
>
> 如果您 CloudFront 要建立和儲存存取記錄，則需要授予其他權限。如需詳細資訊，請參閱設定標準記錄和存取日誌檔案所需的許可。

- CloudFrontReadOnlyAccess— 授與 CloudFront 資源的唯讀存取權。

客戶受管政策範例

您可以建立自己的自訂 IAM 政策，以允許 CloudFront API 動作的許可。您可以將這些自訂政策連接至需要指定許可的 IAM 使用者或群組。當您使用 CloudFront API、 AWS SDK 或. AWS CLI以下範例示範幾個常用案例的許可。如需授與使用者完整存取權的策略 CloudFront，請參閱使用 CloudFront 主控台所需的權限。

範例

- 範例 1：允許讀取存取所有分佈
- 範例 2：允許建立、更新和刪除分佈
- 範例 3：允許建立和列出失效
- 範例 4：允許建立分佈

範例 1：允許讀取存取所有分佈

以下權限策略授予用戶在 CloudFront 控制台中查看所有發行版的權限：

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "acm:ListCertificates",
            "cloudfront:GetDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:ListCloudFrontOriginAccessIdentities",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      }
   ]
}
```

範例 2：允許建立、更新和刪除分佈

下列權限原則允許使用者使用 CloudFront 主控台建立、更新和刪除散佈：

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
         "Action":[
            "acm:ListCertificates",
            "cloudfront:CreateDistribution",
            "cloudfront:DeleteDistribution",
            "cloudfront:GetDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:UpdateDistribution",
            "cloudfront:ListCloudFrontOriginAccessIdentities",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets",
            "s3:PutBucketPolicy"
         ],
         "Resource":"arn:aws:s3:::*"
      }
   ]
}
```

`cloudfront:ListCloudFrontOriginAccessIdentities` 許可讓使用者可將許可自動授予現有原始存取身分，以存取 Amazon S3 儲存貯體中的物件。如果您還希望使用者能夠建立原始存取身分，則還需要允許 `cloudfront:CreateCloudFrontOriginAccessIdentity` 許可。

範例 3：允許建立和列出失效

以下許可政策可讓使用者建立和失效清單。它包含對 CloudFront 分佈的讀取存取權，因為您先顯示分配的設定來建立並檢視無效：

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
         "Action":[
            "acm:ListCertificates",
            "cloudfront:GetDistribution",
            "cloudfront:GetStreamingDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:ListCloudFrontOriginAccessIdentities",
            "cloudfront:CreateInvalidation",
            "cloudfront:GetInvalidation",
            "cloudfront:ListInvalidations",
            "elasticloadbalancing:DescribeLoadBalancers",
            "iam:ListServerCertificates",
            "sns:ListSubscriptionsByTopic",
            "sns:ListTopics",
            "waf:GetWebACL",
            "waf:ListWebACLs"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      }
   ]
}
```

範例 4：允許建立分佈

以下權限策略授予用戶在 CloudFront 控制台中創建和列出發行版的權限。對於`CreateDistribution`動作，請為分配 ARN () 指定萬用字元 (`*arn:aws:cloudfront::123456789012:distribution/*`)，`Resource`而不是萬用字元。如需有關`Resource`元素的詳細資訊，請參閱 IAM JSON 政策元素：IAM 使用者指南中的資源。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "cloudfront:CreateDistribution",
```

```
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "cloudfront:ListDistributions",
            "Resource": "*"
        }
    ]
}
```

AWSAmazon 的受管政策 CloudFront

若要新增許可給使用者、群組和角色，使用 AWS 受管政策比自己撰寫政策更容易。建立 IAM 客戶受管政策需要時間和專業知識，而受管政策可為您的使用者提供其所需的許可。若要快速開始使用，您可以使用 AWS 受管政策。這些政策涵蓋常見的使用案例，並可在您的 AWS 帳戶 中使用。如需有關 AWS 受管政策的詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 服務會維護和更新 AWS 受管政策。您無法更改 AWS 受管政策中的許可。服務偶爾會在 AWS 受管政策中新增其他許可以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (使用者、群組和角色)。當新功能啟動或新許可可用時，服務很可能會更新 AWS 受管政策。服務不會從 AWS 受管政策中移除許可，因此政策更新不會破壞您現有的許可。

此外，AWS 支援跨越多項服務之任務職能的受管政策。例如，ReadOnlyAccess AWS 受管政策提供針對所有 AWS 服務和資源的唯讀存取權限。當服務啟動新功能時，AWS 會為新的操作和資源新增唯讀許可。如需任務職能政策的清單和說明，請參閱 IAM 使用者指南中有關任務職能的 AWS 受管政策。

AWS 受管政策： CloudFrontReadOnlyAccess

您可將 CloudFrontReadOnlyAccess 政策連接到 IAM 身分。此原則允許 CloudFront 資源的唯讀權限。它還允許對與 CloudFront 主控台中相關 CloudFront 且可見的其他AWS服務資源提供唯讀權限。

許可詳細資訊

此政策包含以下許可。

- `cloudfront:Describe*`— 允許主參與者取得有關資源中繼資料的 CloudFront 資訊。
- `cloudfront:Get*`— 允許主參與者取得 CloudFront 資源的詳細資訊和組態。
- `cloudfront:List*`— 允許主參與者取得 CloudFront資源清單。
- `cloudfront-keyvaluestore:Describe*`-允許主參與者取得有關索引鍵值存放區的資訊。
- `cloudfront-keyvaluestore:Get*`-允許主參與者取得索引鍵值存放區的詳細資訊和組態。
- `cloudfront-keyvaluestore:List*`-允許主參與者取得索引鍵值存放區的清單。
- `acm:ListCertificates` – 允許主參與者取得 ACM 憑證清單。
- `iam:ListServerCertificates` – 允許主參與者取得存放在 IAM 的伺服器憑證清單。
- `route53:List*` – 允許主參與者取得 Route 53 資源的清單。
- `waf:ListWebACLs` – 允許主參與者取得 AWS WAF 的 Web ACL 清單。
- `waf:GetWebACL` – 允許主參與者取得 AWS WAF Web ACL 的詳細資訊。
- `wafv2:ListWebACLs` – 允許主參與者取得 AWS WAF 的 Web ACL 清單。
- `wafv2:GetWebACL` – 允許主參與者取得 AWS WAF Web ACL 的詳細資訊。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "cfReadOnly",
      "Effect": "Allow",
      "Action": [
        "acm:ListCertificates",
        "cloudfront:Describe*",
        "cloudfront:Get*",
        "cloudfront:List*",
        "cloudfront-keyvaluestore:Describe*",
        "cloudfront-keyvaluestore:Get*",
        "cloudfront-keyvaluestore:List*",
        "iam:ListServerCertificates",
        "route53:List*",
        "waf:ListWebACLs",
        "waf:GetWebACL",
        "wafv2:ListWebACLs",
        "wafv2:GetWebACL"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

AWS 受管政策： CloudFrontFullAccess

您可將 CloudFrontFullAccess 政策連接到 IAM 身分。此原則允許 CloudFront 資源的管理權限。它還允許對與 CloudFront 主控台中相關 CloudFront 且可見的其他AWS服務資源提供唯讀權限。

許可詳細資訊

此政策包含以下許可。

- `s3:ListAllMyBuckets` – 允許主參與者取得所有 Amazon S3 儲存貯體的清單。
- `acm:ListCertificates` – 允許主參與者取得 ACM 憑證清單。
- `cloudfront:*`— 允許主參與者對所有 CloudFront資源執行所有動作。
- `cloudfront-keyvaluestore:*`-允許主參與者對索引鍵值存放區執行所有動作。
- `iam:ListServerCertificates` – 允許主參與者取得存放在 IAM 的伺服器憑證清單。
- `waf:ListWebACLs` – 允許主參與者取得 AWS WAF 的 Web ACL 清單。
- `waf:GetWebACL` – 允許主參與者取得 AWS WAF Web ACL 的詳細資訊。
- `wafv2:ListWebACLs` – 允許主參與者取得 AWS WAF 的 Web ACL 清單。
- `wafv2:GetWebACL` – 允許主參與者取得 AWS WAF Web ACL 的詳細資訊。
- `kinesis:ListStreams` – 允許主參與者取得 Amazon Kinesis 串流的清單。
- `kinesis:DescribeStream` – 允許主參與者取得 Kinesis 串流的詳細資訊。
- `iam:ListRoles` – 允許主參與者取得 IAM 中角色的清單。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "cfflistbuckets",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
```

```
    },
    {
      "Sid": "cffullaccess",
      "Action": [
        "acm:ListCertificates",
        "cloudfront:*",
        "cloudfront-keyvaluestore:*",
        "iam:ListServerCertificates",
        "waf:ListWebACLs",
        "waf:GetWebACL",
        "wafv2:ListWebACLs",
        "wafv2:GetWebACL",
        "kinesis:ListStreams"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "cffdescribestream",
      "Action": [
        "kinesis:DescribeStream"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:kinesis:*:*:*"
    },
    {
      "Sid": "cfflistroles",
      "Action": [
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:*"
    }
  ]
}
```

AWS 受管政策： AWSCloudFrontLogger

您無法將該AWSCloudFrontLogger政策附加到 IAM 身分。此原則附加至服務連結角色，可 CloudFront 代表您執行動作。如需詳細資訊，請參閱 the section called “Lambda@Edge 的服務連結角色”。

此政策允許 CloudFront 將日誌文件推送到 Amazon CloudWatch。如需此政策中包含之許可的詳細資訊，請參閱 the section called “記錄器 CloudFront的服務連結角色權限”。

AWS 受管政策： AWSLambdaReplicator

您無法將該AWSLambdaReplicator政策附加到 IAM 身分。此原則附加至服務連結角色，可 CloudFront 代表您執行動作。如需詳細資訊，請參閱 the section called “Lambda@Edge 的服務連結角色”。

此原則 CloudFront 允許在中建立、刪除和停用函數，AWS Lambda以將 Lambda @Edge 函數複寫到中AWS 區域。如需此政策中包含之許可的詳細資訊，請參閱 the section called “Lambda Replicator 的服務連結角色許可”。

CloudFront AWS受管理策略的更新

檢視 CloudFront 自此服務開始追蹤這些變更以來的AWS受管理策略更新詳細資料。如需有關此頁面變更的自動警示，請訂閱「 CloudFront 文件歷史記錄」頁面上的 RSS 摘要。

變更	描述	日期
CloudFrontReadOnlyAccess 和 CloudFrontFullAccess – 對兩個現有政策的更新。	CloudFront 為鍵值存儲添加了新的權限。 新權限可讓使用者取得關於索引鍵值存放區的相關資訊，並對其採取動作。	2023 年 12 月 19 日
CloudFrontReadOnlyAccess – 更新現有政策	CloudFront 添加了描述 CloudFront 函數的新權限。 此權限允許使用者、群組或角色讀取有關函數的資訊和中繼資料，但不能讀取函數的程式碼。	2021 年 9 月 8 日
CloudFront 開始追蹤變更	CloudFront 開始追蹤其AWS受管理策略的變更。	2021 年 9 月 8 日

疑難排解 Amazon CloudFront 身分和存取

使用下列資訊可協助您診斷和修正使用和 IAM 時可能會遇到的 CloudFront 常見問題。

主題

我沒有執行操作的授權 CloudFront

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 cloudfront:*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudfront:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 cloudfront:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我沒有授權執行 iam：PassRole

如果您收到未獲授權執行iam:PassRole動作的錯誤訊息，則必須更新您的原則以允許您將角色傳遞給 CloudFront。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者marymajor嘗試使用主控台執行中的動作時，會發生下列範例錯誤 CloudFront。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人訪 AWS 帳戶 問我的 CloudFront 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解是否 CloudFront 支援這些功能，請參閱Amazon 如何與 IAM 合 CloudFront 作。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱《IAM 使用者指南》中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 角色與資源型政策的差異。

Amazon 中的記錄和監控 CloudFront

監控是維護 CloudFront 及您 AWS 解決方案之可用性和效能的重要部分。您應該從AWS解決方案的所有部分收集監視資料，以便在發生多點失敗時更輕鬆地偵錯。 AWS提供數種工具來監控您的 CloudFront 資源和活動，以及回應潛在事件：

Amazon CloudWatch 警報

> 您可以使用 CloudWatch 警示來監視指定期間內的單一量度。如果指標超過指定的閾值，會傳送一則通知至 Amazon SNS 主題或 AWS Auto Scaling 政策。CloudWatch 當測量結果處於特定狀態時，警示不會叫用動作。必須是狀態已變更並維持了所指定的時間長度，才會呼叫動作。如需更多詳細資訊，請參閱 使用 Amazon CloudFront 監控指標 CloudWatch。

AWS CloudTrail 日誌

> CloudTrail 提供使用者、角色或AWS服務所採取之 API 動作的記錄 CloudFront。使用收集的資訊 CloudTrail，您可以判斷向其發出的 API 要求 CloudFront、提出要求的 IP 位址、提出要求的人員、提出要求的時間以及其他詳細資訊。如需詳細資訊，請參閱 使用記錄 Amazon CloudFront API 呼叫 AWS CloudTrail。

CloudFront 標準記錄和即時記錄

CloudFront 記錄檔會提供有關發佈要求的詳細記錄。這些日誌對許多應用程式來說都是很有用的資料。舉例來說，日誌資訊在安全與存取稽核中相當實用。如需詳細資訊，請參閱 CloudFront 和邊緣功能記錄。

邊緣函數日誌

由邊緣函數 (函數和 Lambda @Edge) 產生的日誌會直接傳送至 Amazon CloudWatch 日誌，而且不會由任何地方存放 CloudFront。 CloudFront CloudFront 函數使用 AWS Identity and Access Management (IAM) 服務連結角色，將客戶產生的記錄檔直接傳送到您帳戶中的 CloudWatch Logs。

CloudFront 主控台報告

主 CloudFront 控台包含各種報告，包括快取統計資料報表、熱門物件報表，以及排名靠前的反向連結報表。大多數 CloudFront 主控台報告都是以CloudFront 存取記錄中的資料為基礎，其中包含有關 CloudFront 接收之每個使用者要求的詳細資訊。不過，您不需要啟用存取日誌來查看報告。如需更多詳細資訊，請參閱 在主控台中檢視 CloudFront 報告。

Amazon 的合規驗證 CloudFront

第三方稽核員會在多個合規計劃中評估 Amazon CloudFront 的安全性和合 AWS 規性。這些包括 SOC、PCI、HIPAA 等。

如需特定合規方案範圍內的 AWS 服務清單，請參閱合規方案範圍內的AWS 服務。如需一般資訊，請參閱 AWS 合規計劃。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告 AWS Artifact。

您在使用時的合規責任取決 CloudFront 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供在上部署以安全性和法規遵循為重點的基準環境的步驟。 AWS
- 建構 HIPAA 安全性與合規性 AWS— 本白皮書說明公司如何使用建立符合 HIPAA 標準的應 AWS 用程式。

  AWS HIPAA 合規計劃包括 CloudFront (不包括透過 CloudFront嵌入式 PoP 傳遞內容) 作為 HIPAA 合格服務。如果您與已執行的「商業夥伴增補合約」(BAA) AWS，您可以使用 CloudFront (不包括

透過 CloudFront 嵌入式 POP 傳遞內容) 來傳遞含有受保護健康資訊 (PHI) 的內容。如需詳細資訊，請參閱 HIPAA 合規。

- AWS 法規遵循資源 — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS Config— 此 AWS 服務評估您的資源配置是否符合內部實踐，行業準則和法規。
- AWS Security Hub— 此 AWS 服務使用安全控制來評估資源配置和安全標準，以幫助您遵守各種合規性框架。如需有關使用 Security Hub 評估CloudFront 資源的詳細資訊，請參閱使用AWS Security Hub 者指南中的 Amazon CloudFront 控制項。

CloudFront 合規性最佳做法

當您使用 Amazon CloudFront 提供內容時，本節提供合規的最佳實務和建議。

如果您執行以AWS 共用責任模型為基礎的 PCI 相容或符合 HIPAA 規範的工作負載，建議您記錄過去 365 天的 CloudFront 使用資料，以供 future 稽核之用。若要記錄用量資料，您可以執行以下操作：

- 啟用 CloudFront 存取記錄。如需詳細資訊，請參閱 設定和使用標準日誌 (存取日誌)。
- 擷取傳送至 CloudFront API 的要求。如需詳細資訊，請參閱 使用記錄 Amazon CloudFront API 呼叫 AWS CloudTrail。

此外，有關如何符合 PCI DSS 和 SOC 標準 CloudFront 的詳細資訊，請參閱以下內容。

支付卡產業資料安全標準 (PCI DSS)

CloudFront (不包括透過 CloudFront Embedded PoP 傳遞的內容) 支援商家或服務供應商處理、儲存和傳輸信用卡資料，並已通過驗證符合支付卡產業 (PCI) 資料安全標準 (DSS)。如需 PCI DSS 的詳細資訊，包括如何要求 AWS PCI 符合性 Package 的副本，請參閱 PCI DSS 等級 1。

我們建議您不要在 CloudFront Edge 快取中快取信用卡資訊，做為安全性最佳作法。例如，您可以將原始伺服器設定為在包含信用卡資訊 (例如信用卡號碼的最後四碼及持卡人聯絡資訊) 的回應中包含 `Cache-Control:no-cache="####"` 標題。

系統和組織控制 (SOC)

CloudFront (不包括透過 CloudFront 嵌入式 PoP 傳遞的內容) 符合系統與組織控制 (SOC) 措施，包括 SOC 1、SOC 2 和 SOC 3。SOC 報告是獨立的第三方檢查報告，展示如何 AWS 實現關鍵合規性控制和目標。這些稽核可確保執行恰當得宜的安全防禦措施與程序，以針對可能影響到客戶與公司資料安全性、機密性和可用性的風險，提供安全防護。這些第三方稽核的結果可在 AWS SOC 合規性網站上取得，您可以在其中檢視已發佈的報告，以取得有關支援 AWS 作業和法規遵循之控制項的詳細資訊。

Amazon 的韌性 CloudFront

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。AWS區域提供多個分開且隔離的實際可用區域，它們以低延遲、高輸送量和高度備援聯網功能相互連結。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需有關 AWS 區域與可用區域的更多相關資訊，請參閱 AWS 全球基礎設施。

CloudFront 原始容錯移

除了支援AWS全球基礎設施之外，Amazon 還 CloudFront 提供來源容錯移轉功能，以協助支援您的資料彈性需求。CloudFront 這是一項全球性服務，可透過稱為節點或存在點 (PoP) 的全球資料中心網路傳遞您的內容。如果您的內容尚未在節點中進行快取，CloudFront 會從您指定為最終版本內容的原始伺服器擷取該內容。

您可以將 CloudFront 設定原始伺服器容錯移轉，以為特定案例改善彈性和提高可用性。要開始使用，請建立一個原點群組，在其中指定主要原點 CloudFront 加上第二個原點。 CloudFront 當主要來源傳回特定的 HTTP 狀態碼失敗回應時，會自動切換至第二個原點。如需更多詳細資訊，請參閱 透過 CloudFront 原始容錯移轉將高可用性。

Amazon 基礎設施安全 CloudFront

作為一項受管服務，Amazon CloudFront 受到AWS全球網路安全的保護。如需 AWS 安全服務以及 AWS 如何保護基礎設施的相關資訊，請參閱 AWS 雲端安全。若要使用基礎設施安全性的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS 架構良好的框架中的基礎設施保護。

您可以使用AWS已發佈的 API 呼叫透 CloudFront 過網路進行存取。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

CloudFront 功能在AWS帳戶之間使用高度安全的隔離屏障，確保客戶環境可以安全地抵禦 Spectre 和崩潰等側通道攻擊。Functions 無法存取或修改屬於其他客戶的資料。Functions 在專用的 CPU 上運

行於專用的單一執行緒程序中，而不需要超執行緒。在任何給定的 CloudFront邊緣位置存在點 (POP) 中， CloudFront Functions 一次只能為一位客戶提供服務，並且在函數執行之間清除所有客戶特定資料。

配額

CloudFront 須遵守以下配額。

主題

一般配額

實體	預設配額
每個分佈的資料傳輸率	150 Gbps 申請提高配額
每個分佈的每秒請求數	250,000 申請提高配額
可新增至分佈的標籤	50

實體	預設配額
	申請提高配額
您可以為每個分佈提供服務的檔案	沒有配額
要求或原始回應的最大長度，包括標頭和查詢字串，但不包括內文內容	20,480 位元組
URL 的最大長度	8,192 位元組

分佈的一般配額

實體	預設配額
每個分佈的備用網域名稱 (CNAME) 如需更多詳細資訊，請參閱 新增替代網域名稱 (CNAME) 以使用自訂 URL。	100 申請提高配額
每個分佈的快取行為	25 申請提高配額
每個原始伺服器的連線嘗試次數 如需更多詳細資訊，請參閱 連線嘗試。	1-3
每個原始伺服器的連線逾時 如需更多詳細資訊，請參閱 連線逾時。。	1-10 秒
每分佈 AWS 帳戶 如需詳細資訊，請參閱 建立分發。	200 申請提高配額
每個來源存取控制的發佈	100 申請提高配額

實體	預設配額
檔案壓縮：壓縮的檔案大小範圍 CloudFront 如需詳細資訊，請參閱 提供壓縮檔案。	1,000 到 10,000,000 位元組
每個來源的保持活動超時 如需詳細資訊，請參閱 保持連線逾時 (僅限自訂原始伺服器)。	1-60 秒 申請提高配額
每個 HTTP GET 回應的可快取檔案大小上限。 只會快取 HTTP GET 的回應。不會快取 POST 或 PUT 的回應。	50 GB
每個原點存取控制 AWS 帳戶	100
每個原始存取身分 AWS 帳戶	100 申請提高配額
每個分佈的來源數	25 申請提高配額
每個分佈的原始伺服器群組數	10 申請提高配額
每個來源的回應逾時 如需更多詳細資訊，請參閱 回應逾時 (僅限自訂原始伺服器)。	1-60 秒 申請提高配額
分期每個分佈 AWS 帳戶 如需詳細資訊，請參閱 the section called “使用持續部署，安全地測試變更”。	20 申請提高配額

政策的一般配額

實體	預設配額
每個快取政策 AWS 帳戶	20
與相同快取政策相關聯的分佈	100
每個快取政策的查詢字串	10 申請提高配額
每個快取政策的標頭	10 申請提高配額
每個快取政策的 Cookie	10 申請提高配額
快取政策中所有查詢字串、標頭和 Cookie 名稱的總合長度	1024
每個原始請求政策 AWS 帳戶	20
與相同原始伺服器請求政策相關聯的分佈	100
每個原始伺服器請求政策的查詢字串	10 申請提高配額
每個原始伺服器請求政策的標頭	10 申請提高配額
根據原始伺服器請求政策	10 申請提高配額
原始伺服器請求政策中所有查詢字串、標頭和 Cookie 名稱的總合長度	1024
回應標頭政策 (每) AWS 帳戶	20

實體	預設配額
	申請提高配額
與相同回應標頭政策相關聯的分佈	100 申請提高配額
每個回應標頭政策的自訂標頭	10 申請提高配額
每個持續部署原則 AWS 帳戶	20 申請提高配額

CloudFront 功能配額

實體	預設配額
每個功能 AWS 帳戶	100
最大函數大小	10 KB 申請提高配額
最大函數記憶體	2 MB
與相同函數相關聯的分佈	100

除了這些配額之外，使用 CloudFront 函數時還有其他一些限制。如需詳細資訊，請參閱 CloudFront 功能限制。

鍵值存放區的配額

實體	預設配額
鍵值對中鍵的最大大小	512 個位元組

實體	預設配額
鍵值對中值的最大大小	1 KB
您可以在單一 API 請求中更新的鍵值對上限	50 個按鍵或 3 MB 承載資料，以先達到者為準
個別鍵值存放區的大小上限	5 MB
單一鍵值存放區可以關聯的函數數量上限	10
每個函數的鍵值存放區數量上限	1
每個帳戶的鍵值存放區數量上限	50 申請提高配額

Lambda@Edge 的配額

在這個部分的配額適用於 Lambda@Edge。除了預設 AWS Lambda 配額之外，這些配額也適用。如需 Lambda 配額，請參閱 AWS Lambda 開發人員指南中的配額。

> **Note**
>
> 依您 AWS 帳戶的配額，Lambda 將會因應流量增加而動態擴展容量。如需詳細資訊，請參閱 AWS Lambda 開發人員指南中的函數擴展。

一般配額

實體	預設配額
每 AWS 帳戶 個可以具有 Lambda @Edge 函數的分佈	500 申請提高配額
每個分佈的 Lambda@Edge 函數	100

實體	預設配額
	申請提高配額
每秒請求數	1 萬人（每人 AWS 區域） 申請提高配額
並行執行數 如需詳細資訊，請參閱 AWS Lambda 開發人員指南中的函數擴展。	一千人 (每人 AWS 區域) 申請提高配額
與相同函數相關聯的分佈	500

因事件類型而異的配額

實體	檢視器請求和檢視器回應事件	原始伺服器請求和原始伺服器回應事件
函數記憶體大小	128 MB	與 Lambda 配額相同。
函數逾時。此函數可以對 AWS 區域中的 Amazon S3 儲存貯體、DynamoDB 表格或 Amazon EC2 執行個體等資源進行網路呼叫。	5 秒	30 秒
由 Lambda 函式產生之回應的大小，包括標頭和內文	40 KB	1 MB
Lambda 函式和任何包含程式庫的最大壓縮大小	1 MB	50 MB

除了這些配額，使用 Lambda@Edge 函數時還有一些其他限制。如需更多詳細資訊，請參閱 對 Lambda@Edge 的限制。

SSL 憑證的配額

實體	預設配額
使用專用 IP 位址提供 HTTPS 要求 AWS 帳戶 時的每個 SSL 憑證 (使用 SNI 提供 HTTPS 要求時無配額) 如需詳細資訊，請參閱 搭配使用 HTTPS CloudFront。	2 申請提高配額
可與 CloudFront發行版相關聯的 SSL 憑證	1

失效的配額

實體	預設配額
檔案失效：作用中失效請求所允許的最大檔案數量，不包括萬用字元失效 如需更多詳細資訊，請參閱 使檔案失效。	3,000
檔案失效：允許作用中萬用字元失效的最大數量	15
檔案失效：一個萬用字元失效可以處理的檔案數量上限	沒有配額

金鑰群組的配額

實體	預設配額
單一金鑰群組中的公有金鑰	5 申請提高配額
與單一快取行為相關聯的金鑰群組	4 申請提高配額
每個主要群組 AWS 帳戶	10 申請提高配額

實體	預設配額
與單一金鑰群組相關聯的分佈	100 申請提高配額

WebSocket 連線配額

實體	預設配額
原始伺服器回應逾時 (閒置逾時)	10 分鐘 如果在過去 10 分鐘內 CloudFront 未偵測到從來源傳送到用戶端的任何位元組，則會假設連線處於閒置狀態且已關閉。

欄位層級加密的配額

實體	預設配額
要加密的欄位的最大長度 如需更多詳細資訊，請參閱 使用欄位層級加密來協助保護敏感資料。	16 KB
設定欄位層級加密時，請求內文中的欄位數目上限	10
設定欄位層級加密時，請求內文中的最大長度	1 MB
可與一個 AWS 帳戶關聯的欄位層級加密組態的最大數量	10
可與一個 AWS 帳戶關聯的欄位層級加密設定檔的最大數量	10
可以添加到一個公鑰的最大數量 AWS 帳戶	10
可在一個設定檔中指定的最大加密欄位數	10

實體	預設配額
可與欄位層級加密 CloudFront 配置相關聯的最大發行版數	20
可包含在欄位層級加密組態中的查詢參數設定檔對應之最大數量	5

Cookie 的配額 (舊版快取設定)

這些配額會套用至 CloudFront舊版快取設定。建議您使用快取政策或原始伺服器請求政策，而不是舊版設定。

實體	預設配額
每個快取行為的 Cookie 如需更多詳細資訊，請參閱 根據 Cookie 快取內容。	10 申請提高配額
Cookie 名稱中的位元組總數 (如果您設定 CloudFront 將所有 Cookie 轉寄至來源，則不適用)	512 減去 Cookies 的數量

查詢字串的配額 (舊版快取設定)

這些配額會套用至 CloudFront舊版快取設定。建議您使用快取政策或原始伺服器請求政策，而不是舊版設定。

實體	預設配額
查詢字串中的字元數目上限	128 個字元
相同參數中所有查詢字串的最大總字元數	512 個字元
每個快取行為的查詢字串 如需更多詳細資訊，請參閱 根據查詢字串參數快取內容。	10 申請提高配額

標頭的配額

實體	預設配額
每個快取行為的標頭 (舊版快取設定) 如需詳細資訊，請參閱 the section called “根據請求標頭快取內容”。	10 申請提高配額
自訂標頭：您可以設定 CloudFront 為新增至原始要求的自訂標頭數目上限 如需詳細資訊，請參閱 the section called “將自訂標頭新增到原始伺服器請求”。	10 申請提高配額
自訂標頭：您可以新增至回應標頭政策的最大自訂標頭數量	10 申請提高配額
自訂標頭：標頭名稱的最大長度	256 個字元
自訂標頭：標頭值的最大長度	1,783 個字元
自訂標頭：所有標頭值和名稱組合的最大長度	10,240 個字元
`Content-Security-Policy` 標題值的最大長度	1,783 個字元 申請提高配額

CloudFront 使用 AWS SDK 的程式碼範例

下列程式碼範例顯示如何搭 CloudFront 配 AWS 軟體開發套件 (SDK) 使用。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

Scenarios (案例) 是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

程式碼範例

- CloudFront 使用 AWS SDK 的動作
  - 搭CreateDistribution配 AWS SDK 或命令列工具使用
  - 搭CreateFunction配 AWS SDK 或命令列工具使用
  - 搭CreateInvalidation配 AWS SDK 或命令列工具使用
  - 搭CreateKeyGroup配 AWS SDK 或命令列工具使用
  - 搭CreatePublicKey配 AWS SDK 或命令列工具使用
  - 搭DeleteDistribution配 AWS SDK 或命令列工具使用
  - 搭GetCloudFrontOriginAccessIdentity配 AWS SDK 或命令列工具使用
  - 搭GetCloudFrontOriginAccessIdentityConfig配 AWS SDK 或命令列工具使用
  - 搭GetDistribution配 AWS SDK 或命令列工具使用
  - 搭GetDistributionConfig配 AWS SDK 或命令列工具使用
  - 搭ListCloudFrontOriginAccessIdentities配 AWS SDK 或命令列工具使用
  - 搭ListDistributions配 AWS SDK 或命令列工具使用
  - 搭UpdateDistribution配 AWS SDK 或命令列工具使用
- CloudFront 使用 AWS SDK 的案例
  - 使用 AWS SDK 刪除 CloudFront 簽署資源
  - 使用 AWS SDK 建立已簽署的網址和 Cookie

CloudFront 使用 AWS SDK 的動作

下列程式碼範例示範如何使用 AWS SDK 執 CloudFront行個別動作。這些摘錄會呼叫 CloudFront API，是來自必須在內容中執行的大型程式碼摘錄。每個範例都包含一個連結 GitHub，您可以在其中找到設定和執行程式碼的指示。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 Amazon CloudFront API 參考資料。

範例

- 搭CreateDistribution配 AWS SDK 或命令列工具使用
- 搭CreateFunction配 AWS SDK 或命令列工具使用
- 搭CreateInvalidation配 AWS SDK 或命令列工具使用
- 搭CreateKeyGroup配 AWS SDK 或命令列工具使用
- 搭CreatePublicKey配 AWS SDK 或命令列工具使用
- 搭DeleteDistribution配 AWS SDK 或命令列工具使用
- 搭GetCloudFrontOriginAccessIdentity配 AWS SDK 或命令列工具使用
- 搭GetCloudFrontOriginAccessIdentityConfig配 AWS SDK 或命令列工具使用
- 搭GetDistribution配 AWS SDK 或命令列工具使用
- 搭GetDistributionConfig配 AWS SDK 或命令列工具使用
- 搭ListCloudFrontOriginAccessIdentities配 AWS SDK 或命令列工具使用
- 搭ListDistributions配 AWS SDK 或命令列工具使用
- 搭UpdateDistribution配 AWS SDK 或命令列工具使用

搭`CreateDistribution`配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用`CreateDistribution`。

CLI

AWS CLI

若要建立 CloudFront 分佈

下列範例會為名為的 S3 儲存貯體建立分發`awsexamplebucket`，並使用命令列引數指定`index.html`為預設根物件：

```
aws cloudfront create-distribution \
    --origin-domain-name awsexamplebucket.s3.amazonaws.com \
    --default-root-object index.html
```

您可以在 JSON 檔案中提供發佈組態，而不是使用命令列引數，如下列範例所示：

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

該文件dist-config.json是當前文件夾中的 JSON 文檔，其中包含以下內容：

```
{
    "CallerReference": "cli-example",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
                "DomainName": "awsexamplebucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
```

```
            },
            "Headers": {
                "Quantity": 0
            },
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
```

```
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

無論您是使用命令行參數還是 JSON 文件提供分發信息，輸出都是相同的：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EMLARXS9EXAMPLE",
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-11-22T00:55:15.705Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
```

```
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
                        "DomainName": "awsexamplebucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
```

```
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
```

```
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考CreateDistribution中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

下列範例使用 Amazon Simple Storage Service (Amazon S3) 儲存貯體做為內容來源。

創建分發後，代碼創建一個CloudFrontWaiter等待，直到發行版部署後返回發行版。

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.CreateDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.ItemSelection;
import software.amazon.awssdk.services.cloudfront.model.Method;
import software.amazon.awssdk.services.cloudfront.model.ViewerProtocolPolicy;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;
import software.amazon.awssdk.services.s3.S3Client;
```

```
import java.time.Instant;

public class CreateDistribution {

        private static final Logger logger =
 LoggerFactory.getLogger(CreateDistribution.class);

        public static Distribution createDistribution(CloudFrontClient
 cloudFrontClient, S3Client s3Client,
                        final String bucketName, final String keyGroupId, final
 String originAccessControlId) {

                final String region = s3Client.headBucket(b ->
 b.bucket(bucketName)).sdkHttpResponse().headers()
                                .get("x-amz-bucket-region").get(0);
                final String originDomain = bucketName + ".s3." + region +
 ".amazonaws.com";
                String originId = originDomain; // Use the originDomain value for
 the originId.

                // The service API requires some deprecated methods, such as
                // DefaultCacheBehavior.Builder#minTTL and #forwardedValue.
                CreateDistributionResponse createDistResponse =
 cloudFrontClient.createDistribution(builder -> builder
                                .distributionConfig(b1 -> b1
                                                .origins(b2 -> b2
                                                                .quantity(1)
                                                                .items(b3 -> b3

 .domainName(originDomain)

 .id(originId)

 .s3OriginConfig(builder4 -> builder4

               .originAccessIdentity(

                               ""))

 .originAccessControlId(

               originAccessControlId)))
                                                .defaultCacheBehavior(b2 -> b2
```

```
.viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)

.targetOriginId(originId)
                                                                .minTTL(200L)

.forwardedValues(b5 -> b5

.cookies(cp -> cp

              .forward(ItemSelection.NONE))

.queryString(true))

.trustedKeyGroups(b3 -> b3

.quantity(1)

.items(keyGroupId)

.enabled(true))

.allowedMethods(b4 -> b4

.quantity(2)

.items(Method.HEAD, Method.GET)

.cachedMethods(b5 -> b5

              .quantity(2)

              .items(Method.HEAD,

                            Method.GET))))
                                          .cacheBehaviors(b -> b
                                                          .quantity(1)
                                                          .items(b2 -> b2

.pathPattern("/index.html")

.viewerProtocolPolicy(

              ViewerProtocolPolicy.ALLOW_ALL)
```

```
.targetOriginId(originId)

.trustedKeyGroups(b3 -> b3

              .quantity(1)

              .items(keyGroupId)

              .enabled(true))

.minTTL(200L)

.forwardedValues(b4 -> b4

              .cookies(cp -> cp

                            .forward(ItemSelection.NONE))

              .queryString(true))

.allowedMethods(b5 -> b5.quantity(2)

              .items(Method.HEAD,

                            Method.GET)

              .cachedMethods(b6 -> b6

                            .quantity(2)

                            .items(Method.HEAD,

                                          Method.GET)))))
                                           .enabled(true)
                                           .comment("Distribution built with
java")

.callerReference(Instant.now().toString())));

               final Distribution distribution =
createDistResponse.distribution();
               logger.info("Distribution created. DomainName: [{}]  Id: [{}]",
distribution.domainName(),
```

```
                                distribution.id());
                logger.info("Waiting for distribution to be deployed ...");
                try (CloudFrontWaiter cfWaiter =
 CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
                        ResponseOrException<GetDistributionResponse>
 responseOrException = cfWaiter
                                        .waitUntilDistributionDeployed(builder ->
 builder.id(distribution.id()))
                                        .matched();
                        responseOrException.response()
                                        .orElseThrow(() -> new
 RuntimeException("Distribution not created"));
                        logger.info("Distribution deployed. DomainName: [{}]  Id:
 [{}]", distribution.domainName(),
                                        distribution.id());
                }
                return distribution;
        }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考CreateDistribution中的。

PowerShell

適用的工具 PowerShell

範例 1：建立使用記錄和快取設定的基本 CloudFront 散發。

```
$origin = New-Object Amazon.CloudFront.Model.Origin
$origin.DomainName = "ps-cmdlet-sample.s3.amazonaws.com"
$origin.Id = "UniqueOrigin1"
$origin.S3OriginConfig = New-Object Amazon.CloudFront.Model.S3OriginConfig
$origin.S3OriginConfig.OriginAccessIdentity = ""
New-CFDistribution `
      -DistributionConfig_Enabled $true `
      -DistributionConfig_Comment "Test distribution" `
      -Origins_Item $origin `
      -Origins_Quantity 1 `
      -Logging_Enabled $true `
      -Logging_IncludeCookie $true `
      -Logging_Bucket ps-cmdlet-sample-logging.s3.amazonaws.com `
      -Logging_Prefix "help/" `
```

```
    -DistributionConfig_CallerReference Client1 `
    -DistributionConfig_DefaultRootObject index.html `
    -DefaultCacheBehavior_TargetOriginId $origin.Id `
    -ForwardedValues_QueryString $true `
    -Cookies_Forward all `
    -WhitelistedNames_Quantity 0 `
    -TrustedSigners_Enabled $false `
    -TrustedSigners_Quantity 0 `
    -DefaultCacheBehavior_ViewerProtocolPolicy allow-all `
    -DefaultCacheBehavior_MinTTL 1000 `
    -DistributionConfig_PriceClass "PriceClass_All" `
    -CacheBehaviors_Quantity 0 `
    -Aliases_Quantity 0
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程CreateDistribution式參考中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateFunction配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用CreateFunction。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.CloudFrontException;
import software.amazon.awssdk.services.cloudfront.model.CreateFunctionRequest;
import software.amazon.awssdk.services.cloudfront.model.CreateFunctionResponse;
import software.amazon.awssdk.services.cloudfront.model.FunctionConfig;
```

```
import software.amazon.awssdk.services.cloudfront.model.FunctionRuntime;
import java.io.InputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateFunction {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <functionName> <filePath>

                Where:
                    functionName - The name of the function to create.\s
                    filePath - The path to a file that contains the application
 logic for the function.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String functionName = args[0];
        String filePath = args[1];
        CloudFrontClient cloudFrontClient = CloudFrontClient.builder()
                .region(Region.AWS_GLOBAL)
                .build();

        String funArn = createNewFunction(cloudFrontClient, functionName,
 filePath);
        System.out.println("The function ARN is " + funArn);
        cloudFrontClient.close();
    }
```

```
    public static String createNewFunction(CloudFrontClient cloudFrontClient,
String functionName, String filePath) {
        try {
            InputStream fileIs =
CreateFunction.class.getClassLoader().getResourceAsStream(filePath);
            SdkBytes functionCode = SdkBytes.fromInputStream(fileIs);

            FunctionConfig config = FunctionConfig.builder()
                    .comment("Created by using the CloudFront Java API")
                    .runtime(FunctionRuntime.CLOUDFRONT_JS_1_0)
                    .build();

            CreateFunctionRequest functionRequest =
CreateFunctionRequest.builder()
                    .name(functionName)
                    .functionCode(functionCode)
                    .functionConfig(config)
                    .build();

            CreateFunctionResponse response =
cloudFrontClient.createFunction(functionRequest);
            return response.functionSummary().functionMetadata().functionARN();

        } catch (CloudFrontException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考CreateFunction中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateInvalidation配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用CreateInvalidation。

CLI

AWS CLI

若要建立分配 CloudFront 的無效驗證，請執行下列

下列create-invalidation範例會針對指定 CloudFront 發行版中的指定檔案建立無效驗證：

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",
    "Invalidation": {
        "Id": "I1JLWSDAP8FU89",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:24:51.407Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file2.png",
                    "/example-path/example-file.jpg"
                ]
            },
            "CallerReference": "cli-1575570291-670203"
        }
    }
}
```

在上一個範例中， AWS CLI 會自動產生隨機的CallerReference。若要指定您自己的參數CallerReference，或避免將失效參數作為命令列引數傳遞，您可以使用 JSON 檔案。下列範例會在名為的 JSON 檔案中提供無效驗證參數，為兩個檔案建立無效驗證：inv-batch.json

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
```

```
    --invalidation-batch file://inv-batch.json
```

inv-batch.json 的內容：

```
{
    "Paths": {
        "Quantity": 2,
        "Items": [
            "/example-path/example-file.jpg",
            "/example-path/example-file2.png"
        ]
    },
    "CallerReference": "cli-example"
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUYOIK",
    "Invalidation": {
        "Id": "I2J0I21PCUYOIK",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:40:49.413Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file2.png"
                ]
            },
            "CallerReference": "cli-example"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考CreateInvalidation中的。

PowerShell

適用的工具 PowerShell

範例 1：此範例會在識別碼為 EXAMPLENSTXAXE 的發佈上建立新的無效驗證。這 CallerReference 是使用者選擇的唯一識別碼；在此情況下，會使用代表 2019 年 5 月 15 日上午 9 點的時間戳記。$Path 變數會儲存使用者不希望做為散發快取一部分的影像和媒體檔案的三個路徑。路徑 _ 數量參數值是在-Paths_Item 參數中指定的路徑總數。

```
$Paths = "/images/*.gif", "/images/image1.jpg", "/videos/*.mp4"
New-CFInvalidation -DistributionId "EXAMPLENSTXAXE" -
InvalidationBatch_CallerReference 20190515090000 -Paths_Item $Paths -
Paths_Quantity 3
```

輸出：

```
Invalidation                          Location

------------                          --------

Amazon.CloudFront.Model.Invalidation https://cloudfront.amazonaws.com/2018-11-05/
distribution/EXAMPLENSTXAXE/invalidation/EXAMPLE8NOK9H
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程CreateInvalidation式參考中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateKeyGroup配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用CreateKeyGroup。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

金鑰群組至少需要一個用來驗證已簽署的 URL 或 Cookie 的公開金鑰。

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;

import java.util.UUID;

public class CreateKeyGroup {
    private static final Logger logger =
 LoggerFactory.getLogger(CreateKeyGroup.class);

    public static String createKeyGroup(CloudFrontClient cloudFrontClient, String
 publicKeyId) {
        String keyGroupId = cloudFrontClient.createKeyGroup(b ->
 b.keyGroupConfig(c -> c
                .items(publicKeyId)
                .name("JavaKeyGroup" + UUID.randomUUID()))
                .keyGroup().id();
        logger.info("KeyGroup created with ID: [{}]", keyGroupId);
        return keyGroupId;
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考CreateKeyGroup中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreatePublicKey配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用CreatePublicKey。

CLI

AWS CLI

若要建立 CloudFront 公開金鑰

下列範例會在名為的 JSON 檔案中提供參數，以建立 CloudFront 公開金鑰pub-key-config.json。您必須先擁有 PEM 編碼的公開金鑰，才能使用此命令。如需詳細資訊，請參閱 Amazon CloudFront 開發人員指南中的建立 RSA 金鑰配對。

```
aws cloudfront create-public-key \
    --public-key-config file://pub-key-config.json
```

該文件pub-key-config.json是包含以下內容的當前文件夾中的 JSON 文檔。請注意，公開金鑰會以 PEM 格式編碼。

```
{
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----
END PUBLIC KEY-----\n",
    "Comment": "example public key"
}
```

輸出：

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
```

```
        "Id": "KDFB19YGCR002",
        "CreatedTime": "2019-12-05T18:51:43.781Z",
        "PublicKeyConfig": {
            "CallerReference": "cli-example",
            "Name": "ExampleKey",
            "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----
END PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考CreatePublicKey中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

下列程式碼範例會讀取公開金鑰並將其上傳至 Amazon CloudFront。

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.CreatePublicKeyResponse;
import software.amazon.awssdk.utils.IoUtils;

import java.io.IOException;
import java.io.InputStream;
import java.util.UUID;
```

```
public class CreatePublicKey {
    private static final Logger logger =
 LoggerFactory.getLogger(CreatePublicKey.class);

    public static String createPublicKey(CloudFrontClient cloudFrontClient,
 String publicKeyFileName) {
        try (InputStream is =
 CreatePublicKey.class.getClassLoader().getResourceAsStream(publicKeyFileName)) {
            String publicKeyString = IoUtils.toUtf8String(is);
            CreatePublicKeyResponse createPublicKeyResponse = cloudFrontClient
                    .createPublicKey(b -> b.publicKeyConfig(c -> c
                            .name("JavaCreatedPublicKey" + UUID.randomUUID())
                            .encodedKey(publicKeyString)
                            .callerReference(UUID.randomUUID().toString())));
            String createdPublicKeyId = createPublicKeyResponse.publicKey().id();
            logger.info("Public key created with id: [{}]", createdPublicKeyId);
            return createdPublicKeyId;

        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考CreatePublicKey中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DeleteDistribution配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用DeleteDistribution。

CLI

AWS CLI

若要刪除分 CloudFront 配

下列範例會刪除具有 ID 的 CloudFront 分佈EDFDVBD6EXAMPLE。刪除發行版之前，您必須先停用它。若要停用散發，請使用更新分發命令。如需詳細資訊，請參閱更新發佈範例。

當分發被禁用時，您可以將其刪除。若要刪除發行版，您必須使用該--if-match選項來提供發行版ETag。若要取得ETag，請使用取得分發或get-distribution-config 命令。

```
aws cloudfront delete-distribution \
    --id EDFDVBD6EXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

成功時，此命令沒有輸出。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考DeleteDistribution中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

下列程式碼範例會將發行版更新為 disabled，使用服務員等待變更部署，然後刪除該發行版。

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
 software.amazon.awssdk.services.cloudfront.model.DeleteDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionConfig;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;

public class DeleteDistribution {
        private static final Logger logger =
 LoggerFactory.getLogger(DeleteDistribution.class);

        public static void deleteDistribution(final CloudFrontClient
 cloudFrontClient, final String distributionId) {
                // First, disable the distribution by updating it.
                GetDistributionResponse response =
 cloudFrontClient.getDistribution(b -> b
```

```
                                .id(distributionId));
                String etag = response.eTag();
                DistributionConfig distConfig =
response.distribution().distributionConfig();

                cloudFrontClient.updateDistribution(builder -> builder
                                .id(distributionId)
                                .distributionConfig(builder1 -> builder1

.cacheBehaviors(distConfig.cacheBehaviors())

.defaultCacheBehavior(distConfig.defaultCacheBehavior())
                                                        .enabled(false)
                                                        .origins(distConfig.origins())
                                                        .comment(distConfig.comment())

.callerReference(distConfig.callerReference())

.defaultCacheBehavior(distConfig.defaultCacheBehavior())

.priceClass(distConfig.priceClass())
                                                        .aliases(distConfig.aliases())
                                                        .logging(distConfig.logging())

.defaultRootObject(distConfig.defaultRootObject())

.customErrorResponses(distConfig.customErrorResponses())

.httpVersion(distConfig.httpVersion())

.isIPV6Enabled(distConfig.isIPV6Enabled())

.restrictions(distConfig.restrictions())

.viewerCertificate(distConfig.viewerCertificate())
                                                        .webACLId(distConfig.webACLId())

.originGroups(distConfig.originGroups()))
                                .ifMatch(etag));

                logger.info("Distribution [{}] is DISABLED, waiting for
deployment before deleting ...",
                                distributionId);
                GetDistributionResponse distributionResponse;
```

```
                try (CloudFrontWaiter cfWaiter =
 CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
                        ResponseOrException<GetDistributionResponse>
 responseOrException = cfWaiter
                                        .waitUntilDistributionDeployed(builder ->
 builder.id(distributionId)).matched();
                        distributionResponse = responseOrException.response()
                                        .orElseThrow(() -> new
 RuntimeException("Could not disable distribution"));
                }

                DeleteDistributionResponse deleteDistributionResponse =
 cloudFrontClient
                                .deleteDistribution(builder -> builder
                                                .id(distributionId)

 .ifMatch(distributionResponse.eTag()));
                if (deleteDistributionResponse.sdkHttpResponse().isSuccessful())
 {
                        logger.info("Distribution [{}] DELETED", distributionId);
                }
        }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考DeleteDistribution中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭GetCloudFrontOriginAccessIdentity配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用GetCloudFrontOriginAccessIdentity。

CLI

AWS CLI

若要取得 CloudFront 原始存取身分

下列範例會取得具有識別碼的 CloudFront 來源存取身分 (OAI)E74FTE3AEXAMPLE，包括其ETag和相關聯的 S3 標準 ID。OAI ID 會在存取識別和create-cloud-front-origin存取身分命令的輸出中傳回。list-cloud-front-origin

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[GetCloudFrontOriginAccessIdentity](#)中的。

PowerShell

適用的工具 PowerShell

範例 1：此範例會傳回由-Id 參數指定的特定 Amazon CloudFront 來源存取身分。雖然不需要-Id 參數，但如果未指定，則不會傳回任何結果。

```
Get-CFCloudFrontOriginAccessIdentity -Id E3XXXXXXXXXXRT
```

輸出：

```
     CloudFrontOriginAccessIdentityConfig    Id
    S3CanonicalUserId
     ------------------------------------    --
    -----------------
     Amazon.CloudFront.Model.CloudFrontOr... E3XXXXXXXXXXRT
    4b6e...
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程[GetCloudFrontOriginAccessIdentity](#)式參考中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭 CloudFront 配 AWS SDK 使用](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭`GetCloudFrontOriginAccessIdentityConfig`配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用`GetCloudFrontOriginAccessIdentityConfig`。

CLI

AWS CLI

取得 CloudFront 原始存取身分識別組態

下列範例會取得有關 CloudFront 原始存取身分識別 (OAI) 的中繼資料及IDE74FTE3AEXAMPLE，包括其ETag。OAI ID 會在存取識別和create-cloud-front-origin存取身分命令的輸出中傳回。list-cloud-front-origin

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[GetCloudFrontOriginAccessIdentityConfig](#)中的。

PowerShell

適用的工具 PowerShell

範例 1：此範例會傳回由-Id 參數指定之單一 Amazon CloudFront 原始存取身分的組態資訊。如果未指定-Id 參數，就會發生錯誤。

```
Get-CFCloudFrontOriginAccessIdentityConfig -Id E3XXXXXXXXXXRT
```

輸出：

```
     CallerReference                                                Comment
     ---------------                                                -------
     mycallerreference: 2/1/2011 1:16:32 PM                         Caller
reference: 2/1/2011 1:16:32 PM
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程GetCloudFrontOriginAccessIdentityConfig式參考中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭GetDistribution配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用GetDistribution。

CLI

AWS CLI

若要取得 CloudFront 分配

下列範例會取得包含 ID 的 CloudFront 分佈EDFDVBD6EXAMPLE，包括其ETag。分佈 ID 會在「建立-分佈」和「清單分佈」指令中傳回。

```
aws cloudfront get-distribution --id EDFDVBD6EXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
```

```
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
        "Status": "Deployed",
        "LastModifiedTime": "2019-12-04T23:35:41.433Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
                        "DomainName": "awsexamplebucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
```

```
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
```

```
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[GetDistribution](#)中的。

PowerShell

適用的工具 PowerShell

範例 1：擷取特定發佈的資訊。

```
Get-CFDistribution -Id EXAMPLE0000ID
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程[GetDistribution](#)式參考中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭 CloudFront 配 AWS SDK 使用](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭GetDistributionConfig配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用GetDistributionConfig。

CLI

AWS CLI

若要取得散 CloudFront 發組態

下列範例會取得有關具有 ID 之 CloudFront 分佈的中繼資料EDFDVBD6EXAMPLE，包括其ETag。分佈 ID 會在「建立-分佈」和「清單分佈」指令中傳回。

```
aws cloudfront get-distribution-config --id EDFDVBD6EXAMPLE
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
                    "DomainName": "awsexamplebucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
```

```
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
```

```
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[GetDistributionConfig](#)中的。

PowerShell

適用的工具 PowerShell

範例 1：擷取特定發行版的組態。

```
Get-CFDistributionConfig -Id EXAMPLE0000ID
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程GetDistributionConfig式參考中的。

Python

適用於 Python (Boto3) 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client


    def update_distribution(self):
        distribution_id = input(
            "This script updates the comment for a CloudFront distribution.\n"
            "Enter a CloudFront distribution ID: "
        )

        distribution_config_response =
 self.cloudfront_client.get_distribution_config(
            Id=distribution_id
        )
        distribution_config = distribution_config_response["DistributionConfig"]
        distribution_etag = distribution_config_response["ETag"]

        distribution_config["Comment"] = input(
            f"\nThe current comment for distribution {distribution_id} is "
            f"'{distribution_config['Comment']}'.\n"
            f"Enter a new comment: "
        )
```

```
        self.cloudfront_client.update_distribution(
            DistributionConfig=distribution_config,
            Id=distribution_id,
            IfMatch=distribution_etag,
        )
        print("Done!")
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件GetDistributionConfig中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭ListCloudFrontOriginAccessIdentities配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用ListCloudFrontOriginAccessIdentities。

CLI

AWS CLI

列出 CloudFront 原始存取身分

下列範例會取得您 AWS 帳戶中的 CloudFront 來源存取身分 (OAI) 清單：

```
aws cloudfront list-cloud-front-origin-access-identities
```

輸出：

```
{
    "CloudFrontOriginAccessIdentityList": {
        "Items": [
            {
                "Id": "E74FTE3AEXAMPLE",
                "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
```

```
                "Comment": "Example OAI"
            },
            {
                "Id": "EH1HDMBEXAMPLE",
                "S3CanonicalUserId":
 "1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
                "Comment": "Test OAI"
            },
            {
                "Id": "E2X2C9TEXAMPLE",
                "S3CanonicalUserId":
 "cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
                "Comment": "Example OAI #2"
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[ListCloudFrontOriginAccessIdentities](#)中的。

PowerShell

適用的工具 PowerShell

範例 1：此範例會傳回 Amazon CloudFront 來源存取身分的清單。因為-MaxItem 參數指定的值為 2，所以結果包含兩個識別。

```
Get-CFCloudFrontOriginAccessIdentityList -MaxItem 2
```

輸出：

```
IsTruncated : True
Items       : {E326XXXXXXXXXT, E1YWXXXXXXX9B}
Marker      :
MaxItems    : 2
NextMarker  : E1YXXXXXXXXX9B
Quantity    : 2
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程[ListCloudFrontOriginAccessIdentities](#)式參考中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭ListDistributions配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用ListDistributions。

CLI

AWS CLI

若要列出 CloudFront 分配

下列範例會取得您 AWS 帳戶中的 CloudFront 分配清單：

```
aws cloudfront list-distributions
```

輸出：

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "EMLARXS9EXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
EMLARXS9EXAMPLE",
                "Status": "InProgress",
                "LastModifiedTime": "2019-11-22T00:55:15.705Z",
                "InProgressInvalidationBatches": 0,
                "DomainName": "d111111abcdef8.cloudfront.net",
                "ActiveTrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "DistributionConfig": {
                    "CallerReference": "cli-example",
                    "Aliases": {
                        "Quantity": 0
                    },
                    "DefaultRootObject": "index.html",
                    "Origins": {
                        "Quantity": 1,
                        "Items": [
```

```
                            {
                                "Id": "awsexamplebucket.s3.amazonaws.com-cli-
example",
                                "DomainName":
 "awsexamplebucket.s3.amazonaws.com",
                                "OriginPath": "",
                                "CustomHeaders": {
                                    "Quantity": 0
                                },
                                "S3OriginConfig": {
                                    "OriginAccessIdentity": ""
                                }
                            }
                        ]
                    },
                    "OriginGroups": {
                        "Quantity": 0
                    },
                    "DefaultCacheBehavior": {
                        "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-
example",
                        "ForwardedValues": {
                            "QueryString": false,
                            "Cookies": {
                                "Forward": "none"
                            },
                            "Headers": {
                                "Quantity": 0
                            },
                            "QueryStringCacheKeys": {
                                "Quantity": 0
                            }
                        },
                        "TrustedSigners": {
                            "Enabled": false,
                            "Quantity": 0
                        },
                        "ViewerProtocolPolicy": "allow-all",
                        "MinTTL": 0,
                        "AllowedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
```

```
                        ],
                        "CachedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "SmoothStreaming": false,
                    "DefaultTTL": 86400,
                    "MaxTTL": 31536000,
                    "Compress": false,
                    "LambdaFunctionAssociations": {
                        "Quantity": 0
                    },
                    "FieldLevelEncryptionId": ""
                },
                "CacheBehaviors": {
                    "Quantity": 0
                },
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "Logging": {
                    "Enabled": false,
                    "IncludeCookies": false,
                    "Bucket": "",
                    "Prefix": ""
                },
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": true,
                    "MinimumProtocolVersion": "TLSv1",
                    "CertificateSource": "cloudfront"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
                        "Quantity": 0
                    }
                },
```

```
                    "WebACLId": "",
                    "HttpVersion": "http2",
                    "IsIPV6Enabled": true
                }
            },
            {
                "Id": "EDFDVBD6EXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
EDFDVBD6EXAMPLE",
                "Status": "InProgress",
                "LastModifiedTime": "2019-12-04T23:35:41.433Z",
                "InProgressInvalidationBatches": 0,
                "DomainName": "d930174dauwrn8.cloudfront.net",
                "ActiveTrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "DistributionConfig": {
                    "CallerReference": "cli-example",
                    "Aliases": {
                        "Quantity": 0
                    },
                    "DefaultRootObject": "index.html",
                    "Origins": {
                        "Quantity": 1,
                        "Items": [
                            {
                                "Id": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
                                "DomainName":
 "awsexamplebucket1.s3.amazonaws.com",
                                "OriginPath": "",
                                "CustomHeaders": {
                                    "Quantity": 0
                                },
                                "S3OriginConfig": {
                                    "OriginAccessIdentity": ""
                                }
                            }
                        ]
                    },
                    "OriginGroups": {
                        "Quantity": 0
                    },
```

```
                    "DefaultCacheBehavior": {
                        "TargetOriginId": "awsexamplebucket1.s3.amazonaws.com-
cli-example",
                        "ForwardedValues": {
                            "QueryString": false,
                            "Cookies": {
                                "Forward": "none"
                            },
                            "Headers": {
                                "Quantity": 0
                            },
                            "QueryStringCacheKeys": {
                                "Quantity": 0
                            }
                        },
                        "TrustedSigners": {
                            "Enabled": false,
                            "Quantity": 0
                        },
                        "ViewerProtocolPolicy": "allow-all",
                        "MinTTL": 0,
                        "AllowedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ],
                            "CachedMethods": {
                                "Quantity": 2,
                                "Items": [
                                    "HEAD",
                                    "GET"
                                ]
                            }
                        },
                        "SmoothStreaming": false,
                        "DefaultTTL": 86400,
                        "MaxTTL": 31536000,
                        "Compress": false,
                        "LambdaFunctionAssociations": {
                            "Quantity": 0
                        },
                        "FieldLevelEncryptionId": ""
                    },
```

```
                    "CacheBehaviors": {
                        "Quantity": 0
                    },
                    "CustomErrorResponses": {
                        "Quantity": 0
                    },
                    "Comment": "",
                    "Logging": {
                        "Enabled": false,
                        "IncludeCookies": false,
                        "Bucket": "",
                        "Prefix": ""
                    },
                    "PriceClass": "PriceClass_All",
                    "Enabled": true,
                    "ViewerCertificate": {
                        "CloudFrontDefaultCertificate": true,
                        "MinimumProtocolVersion": "TLSv1",
                        "CertificateSource": "cloudfront"
                    },
                    "Restrictions": {
                        "GeoRestriction": {
                            "RestrictionType": "none",
                            "Quantity": 0
                        }
                    },
                    "WebACLId": "",
                    "HttpVersion": "http2",
                    "IsIPV6Enabled": true
                }
            },
            {
                "Id": "E1X5IZQEXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1X5IZQEXAMPLE",
                "Status": "Deployed",
                "LastModifiedTime": "2019-11-06T21:31:48.864Z",
                "DomainName": "d2e04y12345678.cloudfront.net",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
```

```
                        {
                            "Id": "awsexamplebucket2",
                            "DomainName": "awsexamplebucket2.s3.us-
west-2.amazonaws.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            }
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "awsexamplebucket2",
                    "ForwardedValues": {
                        "QueryString": false,
                        "Cookies": {
                            "Forward": "none"
                        },
                        "Headers": {
                            "Quantity": 0
                        },
                        "QueryStringCacheKeys": {
                            "Quantity": 0
                        }
                    },
                    "TrustedSigners": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "MinTTL": 0,
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ],
                        "CachedMethods": {
```

```
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "SmoothStreaming": false,
                    "DefaultTTL": 86400,
                    "MaxTTL": 31536000,
                    "Compress": false,
                    "LambdaFunctionAssociations": {
                        "Quantity": 0
                    },
                    "FieldLevelEncryptionId": ""
                },
                "CacheBehaviors": {
                    "Quantity": 0
                },
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": true,
                    "MinimumProtocolVersion": "TLSv1",
                    "CertificateSource": "cloudfront"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
                        "Quantity": 0
                    }
                },
                "WebACLId": "",
                "HttpVersion": "HTTP1_1",
                "IsIPV6Enabled": true
            }
        ]
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考ListDistributions中的。

PowerShell

適用的工具 PowerShell

範例 1：傳回分配。

```
Get-CFDistributionList
```

- 如需 API 詳細資訊，請參閱AWS Tools for PowerShell 指令程ListDistributions式參考中的。

Python

適用於 Python (Boto3) 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client


    def list_distributions(self):
        print("CloudFront distributions:\n")
        distributions = self.cloudfront_client.list_distributions()
        if distributions["DistributionList"]["Quantity"] > 0:
            for distribution in distributions["DistributionList"]["Items"]:
                print(f"Domain: {distribution['DomainName']}")
                print(f"Distribution Id: {distribution['Id']}")
                print(
                    f"Certificate Source: "
```

```
                    f"{distribution['ViewerCertificate']['CertificateSource']}"
                )
                if distribution["ViewerCertificate"]["CertificateSource"] ==
 "acm":
                    print(
                        f"Certificate: {distribution['ViewerCertificate']
['Certificate']}"
                    )
                print("")
        else:
            print("No CloudFront distributions detected.")
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件ListDistributions中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭UpdateDistribution配 AWS SDK 或命令列工具使用

下列程式碼範例會示範如何使用UpdateDistribution。

CLI

AWS CLI

更新 CloudFront 發佈的預設根物件的步驟

下列範例會使用 ID 將CloudFront 發佈的預設根物件更新index.html為EDFDVBD6EXAMPLE：

```
aws cloudfront update-distribution --id EDFDVBD6EXAMPLE \
    --default-root-object index.html
```

輸出：

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "Distribution": {
        "Id": "EDFDVBD6EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
```

```
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:55:39.870Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "example-website",
                        "DomainName": "www.example.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "CustomOriginConfig": {
                            "HTTPPort": 80,
                            "HTTPSPort": 443,
                            "OriginProtocolPolicy": "match-viewer",
                            "OriginSslProtocols": {
                                "Quantity": 2,
                                "Items": [
                                    "SSLv3",
                                    "TLSv1"
                                ]
                            },
                            "OriginReadTimeout": 30,
                            "OriginKeepaliveTimeout": 5
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
```

```
            "DefaultCacheBehavior": {
                "TargetOriginId": "example-website",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 1,
                        "Items": [
                            "*"
                        ]
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
```

```
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http1.1",
            "IsIPV6Enabled": true
        }
    }
}
```

若要更新發 CloudFront 佈

下列範例會在 CloudFront 名為的 JSON 檔案中EMLARXS9EXAMPLE提供散發組態，以停用 ID 的散佈dist-config-disable.json。若要更新發行版，您必須使用--if-match選項來提供發行版的ETag。若要取得ETag，請使用取得分發或get-distribution-config 命令。

在您使用下列範例停用散發之後，您可以使用刪除散發命令將其刪除。

```
aws cloudfront update-distribution \
    --id EMLARXS9EXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --distribution-config file://dist-config-disable.json
```

該文件dist-config-disable.json是包含以下內容的當前文件夾中的 JSON 文檔。請注意，該Enabled字段設置為false：

```
{
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
                "DomainName": "awsexamplebucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
        "ForwardedValues": {
            "QueryString": false,
            "Cookies": {
                "Forward": "none"
            },
            "Headers": {
                "Quantity": 0
            },
```

```
            "QueryStringCacheKeys": {
                "Quantity": 0
            }
        },
        "TrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "MinTTL": 0,
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        },
        "SmoothStreaming": false,
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
```

```
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": false,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
```

輸出：

```
{
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:32:35.553Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1574382155-496510",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
```

```
                "Items": [
                    {
                        "Id":
"awsexamplebucket.s3.amazonaws.com-1574382155-273939",
                        "DomainName": "awsexamplebucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId":
"awsexamplebucket.s3.amazonaws.com-1574382155-273939",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
```

```
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": false,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
```

```
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[UpdateDistribution](#)中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionRequest;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.DistributionConfig;
import
 software.amazon.awssdk.services.cloudfront.model.UpdateDistributionRequest;
import software.amazon.awssdk.services.cloudfront.model.CloudFrontException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ModifyDistribution {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:
                    <id>\s

                Where:
                    id - the id value of the distribution.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String id = args[0];
        CloudFrontClient cloudFrontClient = CloudFrontClient.builder()
                .region(Region.AWS_GLOBAL)
                .build();

        modDistribution(cloudFrontClient, id);
        cloudFrontClient.close();
    }

    public static void modDistribution(CloudFrontClient cloudFrontClient, String
idVal) {
        try {
            // Get the Distribution to modify.
            GetDistributionRequest disRequest = GetDistributionRequest.builder()
                    .id(idVal)
                    .build();

            GetDistributionResponse response =
cloudFrontClient.getDistribution(disRequest);
            Distribution disObject = response.distribution();
            DistributionConfig config = disObject.distributionConfig();

            // Create a new DistributionConfig object and add new values to
comment and
            // aliases
            DistributionConfig config1 = DistributionConfig.builder()
                    .aliases(config.aliases()) // You can pass in new values here
                    .comment("New Comment")
                    .cacheBehaviors(config.cacheBehaviors())
                    .priceClass(config.priceClass())
                    .defaultCacheBehavior(config.defaultCacheBehavior())
                    .enabled(config.enabled())
```

```
                    .callerReference(config.callerReference())
                    .logging(config.logging())
                    .originGroups(config.originGroups())
                    .origins(config.origins())
                    .restrictions(config.restrictions())
                    .defaultRootObject(config.defaultRootObject())
                    .webACLId(config.webACLId())
                    .httpVersion(config.httpVersion())
                    .viewerCertificate(config.viewerCertificate())
                    .customErrorResponses(config.customErrorResponses())
                    .build();

            UpdateDistributionRequest updateDistributionRequest =
 UpdateDistributionRequest.builder()
                    .distributionConfig(config1)
                    .id(disObject.id())
                    .ifMatch(response.eTag())
                    .build();

            cloudFrontClient.updateDistribution(updateDistributionRequest);

        } catch (CloudFrontException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考UpdateDistribution中的。

Python

適用於 Python (Boto3) 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
class CloudFrontWrapper:
```

```
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client


    def update_distribution(self):
        distribution_id = input(
            "This script updates the comment for a CloudFront distribution.\n"
            "Enter a CloudFront distribution ID: "
        )

        distribution_config_response =
self.cloudfront_client.get_distribution_config(
            Id=distribution_id
        )
        distribution_config = distribution_config_response["DistributionConfig"]
        distribution_etag = distribution_config_response["ETag"]

        distribution_config["Comment"] = input(
            f"\nThe current comment for distribution {distribution_id} is "
            f"'{distribution_config['Comment']}'.\n"
            f"Enter a new comment: "
        )
        self.cloudfront_client.update_distribution(
            DistributionConfig=distribution_config,
            Id=distribution_id,
            IfMatch=distribution_etag,
        )
        print("Done!")
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件UpdateDistribution中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

CloudFront 使用 AWS SDK 的案例

下列程式碼範例說明如何在 AWS SDK 中 CloudFront實作常見案例。這些案例會示範如何透過在其中呼叫多個函式來完成特定工作 CloudFront。每個案例都包含一個連結 GitHub，您可以在其中找到如何設定和執行程式碼的指示。

範例

- 使用 AWS SDK 刪除 CloudFront 簽署資源
- 使用 AWS SDK 建立已簽署的網址和 Cookie

使用 AWS SDK 刪除 CloudFront 簽署資源

下列程式碼範例顯示如何刪除 Amazon Simple Storage Service (Amazon S3) 儲存貯體中用來存取受限內容的資源。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.DeleteKeyGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.DeleteOriginAccessControlResponse;
import software.amazon.awssdk.services.cloudfront.model.DeletePublicKeyResponse;
import software.amazon.awssdk.services.cloudfront.model.GetKeyGroupResponse;
import
 software.amazon.awssdk.services.cloudfront.model.GetOriginAccessControlResponse;
import software.amazon.awssdk.services.cloudfront.model.GetPublicKeyResponse;

public class DeleteSigningResources {
```

```
   private static final Logger logger =
LoggerFactory.getLogger(DeleteSigningResources.class);

   public static void deleteOriginAccessControl(final CloudFrontClient
cloudFrontClient,
           final String originAccessControlId) {
       GetOriginAccessControlResponse getResponse = cloudFrontClient
               .getOriginAccessControl(b -> b.id(originAccessControlId));
       DeleteOriginAccessControlResponse deleteResponse =
cloudFrontClient.deleteOriginAccessControl(builder -> builder
               .id(originAccessControlId)
               .ifMatch(getResponse.eTag()));
       if (deleteResponse.sdkHttpResponse().isSuccessful()) {
           logger.info("Successfully deleted Origin Access Control [{}]",
originAccessControlId);
       }
   }

   public static void deleteKeyGroup(final CloudFrontClient cloudFrontClient,
final String keyGroupId) {

       GetKeyGroupResponse getResponse = cloudFrontClient.getKeyGroup(b ->
b.id(keyGroupId));
       DeleteKeyGroupResponse deleteResponse =
cloudFrontClient.deleteKeyGroup(builder -> builder
               .id(keyGroupId)
               .ifMatch(getResponse.eTag()));
       if (deleteResponse.sdkHttpResponse().isSuccessful()) {
           logger.info("Successfully deleted Key Group [{}]", keyGroupId);
       }
   }

   public static void deletePublicKey(final CloudFrontClient cloudFrontClient,
final String publicKeyId) {
       GetPublicKeyResponse getResponse = cloudFrontClient.getPublicKey(b ->
b.id(publicKeyId));

       DeletePublicKeyResponse deleteResponse =
cloudFrontClient.deletePublicKey(builder -> builder
               .id(publicKeyId)
               .ifMatch(getResponse.eTag()));

       if (deleteResponse.sdkHttpResponse().isSuccessful()) {
           logger.info("Successfully deleted Public Key [{}]", publicKeyId);
```

```
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的下列主題。
  - DeleteKeyGroup
  - DeleteOriginAccessControl
  - DeletePublicKey

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 AWS SDK 建立已簽署的網址和 Cookie

下列程式碼範例會示範如何建立已簽署的 URL 和 Cookie，以便存取受限制的資源。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

使用CannedSignerRequest類別以固定原則簽署網址或 Cookie。

```
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;

import java.net.URL;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class CreateCannedPolicyRequest {
```

```
    public static CannedSignerRequest createRequestForCannedPolicy(String
 distributionDomainName,
            String fileNameToUpload,
            String privateKeyFullPath, String publicKeyId) throws Exception {
        String protocol = "https";
        String resourcePath = "/" + fileNameToUpload;

        String cloudFrontUrl = new URL(protocol, distributionDomainName,
 resourcePath).toString();
        Instant expirationDate = Instant.now().plus(7, ChronoUnit.DAYS);
        Path path = Paths.get(privateKeyFullPath);

        return CannedSignerRequest.builder()
                .resourceUrl(cloudFrontUrl)
                .privateKey(path)
                .keyPairId(publicKeyId)
                .expirationDate(expirationDate)
                .build();
    }
}
```

使用CustomSignerRequest類別來使用自訂政策簽署網址或Cookie。activeDate和ipRange是選擇性的方法。

```
import software.amazon.awssdk.services.cloudfront.model.CustomSignerRequest;

import java.net.URL;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class CreateCustomPolicyRequest {

    public static CustomSignerRequest createRequestForCustomPolicy(String
 distributionDomainName,
            String fileNameToUpload,
            String privateKeyFullPath, String publicKeyId) throws Exception {
        String protocol = "https";
        String resourcePath = "/" + fileNameToUpload;
```

```
        String cloudFrontUrl = new URL(protocol, distributionDomainName,
 resourcePath).toString();
        Instant expireDate = Instant.now().plus(7, ChronoUnit.DAYS);
        // URL will be accessible tomorrow using the signed URL.
        Instant activeDate = Instant.now().plus(1, ChronoUnit.DAYS);
        Path path = Paths.get(privateKeyFullPath);

        return CustomSignerRequest.builder()
                .resourceUrl(cloudFrontUrl)
                .privateKey(path)
                .keyPairId(publicKeyId)
                .expirationDate(expireDate)
                .activeDate(activeDate) // Optional.
                // .ipRange("192.168.0.1/24") // Optional.
                .build();
    }
}
```

下面的例子演示了如何使用該CloudFrontUtilities類來生成簽名的 cookie 和 URL。檢視上的此程式碼範例 GitHub。

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontUtilities;
import software.amazon.awssdk.services.cloudfront.cookie.CookiesForCannedPolicy;
import software.amazon.awssdk.services.cloudfront.cookie.CookiesForCustomPolicy;
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;
import software.amazon.awssdk.services.cloudfront.model.CustomSignerRequest;
import software.amazon.awssdk.services.cloudfront.url.SignedUrl;

public class SigningUtilities {
    private static final Logger logger =
 LoggerFactory.getLogger(SigningUtilities.class);
    private static final CloudFrontUtilities cloudFrontUtilities =
 CloudFrontUtilities.create();

    public static SignedUrl signUrlForCannedPolicy(CannedSignerRequest
 cannedSignerRequest) {
        SignedUrl signedUrl =
 cloudFrontUtilities.getSignedUrlWithCannedPolicy(cannedSignerRequest);
        logger.info("Signed URL: [{}]", signedUrl.url());
        return signedUrl;
```

```
    }

    public static SignedUrl signUrlForCustomPolicy(CustomSignerRequest
 customSignerRequest) {
        SignedUrl signedUrl =
 cloudFrontUtilities.getSignedUrlWithCustomPolicy(customSignerRequest);
        logger.info("Signed URL: [{}]", signedUrl.url());
        return signedUrl;
    }

    public static CookiesForCannedPolicy
 getCookiesForCannedPolicy(CannedSignerRequest cannedSignerRequest) {
        CookiesForCannedPolicy cookiesForCannedPolicy = cloudFrontUtilities
                .getCookiesForCannedPolicy(cannedSignerRequest);
        logger.info("Cookie EXPIRES header [{}]",
 cookiesForCannedPolicy.expiresHeaderValue());
        logger.info("Cookie KEYPAIR header [{}]",
 cookiesForCannedPolicy.keyPairIdHeaderValue());
        logger.info("Cookie SIGNATURE header [{}]",
 cookiesForCannedPolicy.signatureHeaderValue());
        return cookiesForCannedPolicy;
    }

    public static CookiesForCustomPolicy
 getCookiesForCustomPolicy(CustomSignerRequest customSignerRequest) {
        CookiesForCustomPolicy cookiesForCustomPolicy = cloudFrontUtilities
                .getCookiesForCustomPolicy(customSignerRequest);
        logger.info("Cookie POLICY header [{}]",
 cookiesForCustomPolicy.policyHeaderValue());
        logger.info("Cookie KEYPAIR header [{}]",
 cookiesForCustomPolicy.keyPairIdHeaderValue());
        logger.info("Cookie SIGNATURE header [{}]",
 cookiesForCustomPolicy.signatureHeaderValue());
        return cookiesForCustomPolicy;
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考CloudFrontUtilities中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭 CloudFront 配 AWS SDK 使用。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

文件歷史紀錄

下表說明 CloudFront 文件所做的重要變更。如需獲取更新通知，您可以訂閱 RSS 摘要。

變更	描述	日期
增加了原始訪問控制支持	您現在可以為 AWS Elemental MediaPackage V2 和 AWS Lambda 函數 URL 建立原始存取控制 (OAC)。	2024年4月11日
CMCD 的即時記錄欄位	新增 18 個常見的媒體用戶端資料 (CMCD) 欄位以進行即時記錄。	2024年4月9日
開始使用基本 CloudFront 發行版	已更新使用具有來源存取控制 (OAC) 之 Amazon S3 來源的基本散發教學課程。	2024年3月18日
CloudFront 使用 AWS SDK 的程式碼範例	已新增程式碼範例，顯示如何 CloudFront 搭配 AWS 軟體開發套件 (SDK) 使用。這些範例分為程式碼摘錄 (示範如何呼叫個別服務函數) 和範例 (示範如何透過呼叫相同服務中的多個函數來完成特定任務)。	2024年2月16日
AWS 受管理策略更新	`CloudFrontReadOnlyAccess` 和 `CloudFrontFullAccess` IAM 政策現在支援 `KeyValueStore` 操作。	2023 年 12 月 19 日
JavaScript 运行时 2.0	為函數添加了 JavaScript 運行時 2.0 CloudFront 功能。	2023 年 11 月 21 日
CloudFront KeyValueStore	Amazon CloudFront 現在支持 CloudFront KeyValueS	2023 年 11 月 21 日

	tore。此功能是安全、全域、低延遲的金鑰值資料存放區，可從 CloudFront Functions 內部進行讀取存取，從而在 CloudFront 邊緣位置啟用進階可自訂邏輯。	
Lambda@Edge 支援較新的執行時間版本	Lambda@Edge 現在支援具有 Node.js 20 執行期的 Lambda 函數。	2023 年 11 月 15 日
安全性儀表板	CloudFront 建立發佈時，會建立安全性儀表板。啟用 AWS WAF、管理地理限制，以及檢視請求、機器人和記錄的高階資料。	2023 年 11 月 8 日
排序函數中的查詢字串	CloudFront 現在支援使用 CloudFront 函數進行查詢字串排序。	2023 年 10 月 3 日
AWS WAF 安全性建議	Amazon CloudFront 現在會在 CloudFront 主控台上顯示 AWS WAF 安全建議。	2023 年 9 月 26 日
提供過時 (過期) 快取內容的支援	CloudFront 支持`Stale-While-Revalidate` 和`Stale-If-Error` 緩存控制指令。	2023 年 5 月 15 日
一鍵啟用 AWS WAF 保護	為 CloudFront 發行版新增安 AWS WAF 全性保護的簡化方法。	2023 年 5 月 10 日
為用於標準日誌的新 S3 儲存貯體啟用 ACL	已新增附註和連結，以解決新 S3 儲存貯體的預設 ACL 設定。	2023 年 4 月 11 日

使用 Amazon S3 Object Lambda 建立原始伺服器	您可以使用 Amazon S3 Object Lambda 存取點別名作為您發佈的原始伺服器。	2023 年 3 月 31 日
使用 CloudFront 函數自定義 HTTP 狀態和正文	您可以使用 CloudFront 函數來更新檢視器回應狀態碼，以及取代或移除回應內文。	2023 年 3 月 29 日
已為連接埠新增 CORS 標頭萬用字元選項	您現在可以在 CORS 存取控制標頭中包含連接埠的萬用字元組態。	2023 年 3 月 20 日
為 AWS Security Hub 用戶指南添加了新的鏈接	更新了語言，並在AWS Security Hub 用戶指南中添加了重新組織的 Amazon CloudFront 控件的鏈接。	2023 年 3 月 9 日
CloudFront 現在在原始請求策略中支持阻止列表 (「除了全部」)	使用原始要求原則中的封鎖清單，將所有查詢字串、HTTP 標頭或 Cookie (指定的字串除外) 包含在 CloudFront 傳送至原始位置的要求中。	2023 年 2 月 22 日
CloudFront 添加新的受管理的來源請求策略，以轉發除 Host 標頭以外的所有查看器標題	使用新 CloudFront的受管理來源要求原則，將檢視者要求中的所有標頭 (標`Host`頭除外) 包含在 CloudFront 傳送至原始位置的要求中。	2023 年 2 月 22 日
更新對 Lambda@Edge 的限制	Lambda @Edge 支援設為 Auto (自動) 的 Lambda 執行階段管理組態。	2023 年 2 月 16 日
更新了以下項目的 IAM 指引 CloudFront	更新了指南以符合 IAM 最佳實務。如需更多詳細資訊，請參閱 IAM 中的安全最佳實務。	2023 年 2 月 15 日

擁有增強型安全的原始伺服器存取控制	您現在可以透過僅允許存取指定的發 CloudFront 行版 MediaStore 來確保來源安全。	2023 年 2 月 9 日
用於判斷檢視者標頭結構的新標頭	您現在可以新增標頭順序和數量，協助根據檢視者傳送的標頭來識別檢視者。	2023 年 1 月 13 日
Lambda@Edge 支援較新的執行時間版本	Lambda@Edge 現在支援具有 Node.js 18 執行時間的 Lambda 函數。	2023 年 1 月 12 日
使用回應標頭政策移除回應標頭	您現在可以使用 CloudFront 回應標頭政策從來源移除回應中 CloudFront收到的標頭。傳送給檢視者的回應中不 CloudFront 會包含指定的標頭。	2023 年 1 月 3 日
新的受管原始伺服器請求政策	已新增 `AllViewerAndCloudFrontHeaders-2022-06` 原始伺服器存取政策。	2022 年 12 月 2 日
持續部署以安全測試組態變更	您現在可以測試生產流量子集，以將變更部署到 CDN 組態。	2022 年 11 月 18 日
發行 `CloudFront-Viewer-JA3-Fingerprint` 標頭	您現在可以使用 JA3 指紋，協助判斷請求是否來自已知用戶端。	2022 年 11 月 16 日
已新增 CORS 標頭萬用字元選項	您現在可以在部份 CORS 存取控制標頭中使用各種萬用字元組態。	2022 年 11 月 11 日

CloudFront 分佈的其他量度	CloudFront API 和 AWS CloudFormation. `MonitoringSubscription` 中的 Support	2022 年 10 月 3 日
擁有增強型安全的原始伺服器存取控制	您現在可以透過僅允許存取指定的 CloudFront 分發來保護 Amazon S3 來源的安全。	2022 年 8 月 24 日
對於發行版的 HTTP/3 支持 CloudFront	您現在可以為您 CloudFront 的發行版選擇 HTTP/3。	2022 年 8 月 15 日
將握手詳細資訊新增至 CloudFront檢視器-TLS 標頭	您可以檢視所用 SSL/TLS 交握的資訊。	2022 年 6 月 27 日
Server-Timing 標頭中的新指標	已將新 `cdn-downstream-fbl` 指標新增至 `Server-Timing` 標頭。	2022 年 6 月 13 日
用於獲取有關 TLS 版本和密碼資訊的新標頭	您現在可以使用`CloudFront-Viewer-TLS` 標頭來取得 TLS (或 SSL) 版本的相關資訊，以及用於檢視器與CloudFront之間連線的密碼。	2022 年 5 月 23 日
CloudFront 函數的新FunctionThrottles量度	使用 Amazon CloudWatch，您現在可以監視 CloudFront功能在給定時間段內限制的次數。	2022 年 5 月 4 日
CloudFront 支援 Lambda 函數網址	如果您使用 Lambda 函數搭配函數 URL 來建置無伺服器 Web 應用程式，您現在可以新增 CloudFront 一系列權益。	2022 年 4 月 6 日

HTTP 回應中的 Server-Timing 標頭	您現在可以在傳送的 HTTP 回應中啟用`Server-Timing` 標頭， CloudFront 以檢視指標，以協助您深入瞭解的行為和效能 CloudFront。	2022 年 3 月 30 日
使用 AWS-managed 前綴列表來限制入站流量	現在，您可以將輸入 HTTP 和 HTTPS 流量限制為您的來源，只能從屬於對向來源的伺服器 CloudFront的 IP 位址。	2022 年 2 月 7 日
新功能	CloudFront 新增對回應標頭原則的支援，可讓您指定 HTTP 標頭，以 CloudFront 新增至傳送給檢視者 (網頁瀏覽器或其他用戶端) 的 HTTP 回應。您可以指定所需的標頭 (及其值)，而不需對原始伺服器進行任何變更或撰寫任何程式碼。如需詳細資訊，請參閱在回應中新增或移除 CloudFront HTTP 標頭。	2021 年 11 月 2 日
新的`CloudFront-Viewer-Address` 請求標頭	CloudFront 添加對新標頭的支持`CloudFront-Viewer-Address` ，該標頭包含向其發送 HTTP 請求的查看器的 IP 地址 CloudFront。如需詳細資訊，請參閱新增 CloudFront要求標頭。	2021 年 10 月 25 日
Lambda @Edge 支援新執行階段版本	Lambda@Edge 現在支援具有 Python 3.9 執行時間的 Lambda 函數。如需詳細資訊，請參閱支援的執行階段。	2021 年 9 月 22 日

AWS 受管理策略更新	CloudFront 更新了CloudFrontReadOnlyAccess策略。如需詳細資訊，請參閱AWS 受管理策略的更CloudFront 新。	2021 年 9 月 8 日
新功能	CloudFront 現在支援面向檢視者的 HTTPS 連線的 ECDSA 憑證。如需詳細資訊，請參閱檢視器之間支援的通訊協定和加密 CloudFront和搭配使用 SSL/TLS 憑證的需求。CloudFront	2021 年 7 月 14 日
新功能	CloudFront 現在支持更多方式將替代域名從一個分發移到另一個分發，而無需聯繫 AWS Support。如需詳細資訊，請參閱將替代網域名稱移至其他發行版本。	2021 年 7 月 7 日
新的安全性原則	CloudFront 現在支援新的安全性原則 TLSv1.2_2021，其中包含一組支援的較少密碼。如需詳細資訊，請參閱檢視器與. CloudFront	2021 年 6 月 23 日
新功能	Amazon CloudFront 現在支援F CloudFront unctions，這是一項原生功能，可 CloudFront 讓您在中撰寫輕量型函式，以進 JavaScript 行高規模、延遲敏感的 CDN 自訂。如需詳細資訊，請參閱使用 CloudFront 函數在邊緣自訂。	2021 年 5 月 3 日

Lambda @Edge 支援較新的執行階段	Lambda@Edge 現在支援具有 Node.js 14 執行時間的 Lambda 函數。如需詳細資訊，請參閱支援的執行階段。	2021 年 4 月 29 日
移除 RTMP 發行版的文件	Amazon CloudFront 已於 2020 年 12 月 31 日棄用即時通訊協定 (RTMP) 分發。RTMP 發行版的文件現在已從 Amazon CloudFront 開發人員指南中移除。	2021 年 2 月 10 日
新的定價選項	Amazon CloudFront 推出了CloudFront 安全節省捆綁包，這是一種簡單的方法，可以節省高達 30% 的 CloudFront AWS 費用。有關更多信息，請參閱節省捆綁常見問題解答。	2021 年 2 月 5 日
新增教學	Amazon 開 CloudFront 發人員指南現在包含使用 Amazon CloudFront 限制 Elastic Load Balancing 中對應用程式負載平衡器存取的教學課程。如需詳細資訊，請參閱限制對應用程式負載平衡器的存取。	2020 年 12 月 18 日
公開金鑰管理的新選項	CloudFront 現在支援透過 CloudFront 主控台和 API 管理已簽署 URL 和已簽署 Cookie 的公開金鑰，而不需要 AWS 帳戶 root 使用者存取權。如需詳細資訊，請參閱指定可以建立簽署 URL 和已簽署 Cookie 的簽署者。	2020 年 10 月 22 日

新功能 — 起源護 Shield	CloudFront 現在支援 CloudFront Origin Shield，這是 CloudFront 快取基礎架構中的一個額外層，有助於將原始伺服器的負載降到最低、改善其可用性並降低其營運成本。如需詳細資訊，請參閱使用 Amazon CloudFront 起源 Shield。	2020 年 10 月 20 日
新的壓縮格式	CloudFront 現在，當您配置 CloudFront 為在 CloudFront 邊緣位置壓縮對象時，支持 Brotli 壓縮形成。您也可以配置使 CloudFront 用標準化標`Accept-Encoding` 頭緩存 Brotli 對象。如需詳細資訊，請參閱提供壓縮檔案和壓縮支援。	2020 年 9 月 14 日
全新 TLS 通訊協定	CloudFront 現在支援 TLS 1.3 通訊協定，用於檢視器和 CloudFront發行版之間的 HTTPS 連線。根據預設，所有 CloudFront 安全性原則中都會啟用 TLS 1.3。如需詳細資訊，請參閱檢視器與. CloudFront	2020 年 9 月 3 日
新的即時記錄	CloudFront 現在支援可設定的即時記錄。透過即時日誌，您可以即時取得分佈請求的相關資訊。您可以使用即時日誌來監控、分析並根據內容交付效能採取動作。如需詳細資訊，請參閱即時記錄。	2020 年 8 月 31 日

其他指標的 API 支援	CloudFront 現在支援使用 CloudFront API 啟用八個額外的即時指標。如需詳細資訊，請參閱開啟其他量度。	2020 年 8 月 28 日
新的 CloudFront HTTP 標頭	CloudFront 添加了額外的 HTTP 標頭，用於確定有關查看器的信息，例如設備類型，地理位置等。如需詳細資訊，請參閱新增 CloudFront 要求標頭。	2020 年 7 月 23 日
新功能	CloudFront 現在支援快取原則和原始要求政策，讓您更精細地控制 CloudFront 散發的快取金鑰和原始要求。如需詳細資訊，請參閱使用原則。	2020 年 7 月 22 日
新的安全性原則	CloudFront 現在支援新的安全性原則 TLSv1.2_2019，其中包含一組支援的較少密碼。如需詳細資訊，請參閱檢視器與. CloudFront	2020 年 7 月 8 日
控制原始逾時和嘗試次數的新設定	CloudFront 添加了控制來源超時和嘗試的新設置。如需詳細資訊，請參閱控制來源逾時和嘗試次數。	2020 年 6 月 5 日
建立安全靜態網站入門的新文件 CloudFront	開始 CloudFront 使用使用 Amazon S3、 CloudFront Lambda @Edge 等建立安全的靜態網站，這些網站都使用 AWS CloudFormation. 如需詳細資訊，請參閱安全靜態網站入門。	2020 年 6 月 2 日

Lambda @Edge 支援較新的執行階段	Lambda@Edge 現在支援具有 Node.js 12 和 Python 3.8 執行時間的 Lambda 函數。如需詳細資訊，請參閱支援的執行階段。	2020 年 2 月 27 日
新的即時指標 CloudWatch	Amazon 在 Amazon CloudFrontnow 提供八個額外的即時指標 CloudWatch。如需詳細資訊，請參閱開啟其他 CloudFront 分佈量度。	2019 年 12 月 19 日
存取記錄中的新欄位	CloudFront 新增七個新欄位以存取記錄檔。如需詳細資訊，請參閱標準記錄檔欄位。	2019 年 12 月 12 日
AWS WordPress 插件	您可以使用該 AWS WordPress 插件為您的 WordPress 網站的訪問者提供使用的加速觀看體驗 CloudFront。（更新：自 2022 年 9 月 30 日起， AWS 對於 WordPress 插件已被棄用。）	2019 年 10 月 30 日
以標籤為基礎和資源層級 IAM 許可政策	CloudFront 現在支援另外兩種指定 IAM 權限政策的方式：標籤型和資源層級政策許可。如需詳細資訊，請參閱管理資源存取。	2019 年 8 月 8 日
Support Python 程式設計語言	除了 Node.js 之外，您現在可以使用 Python 程式設計語言來開發 Lambda@Edge 中的函數。如需涵蓋各種案例的範例函數，請參閱 Lambda@Edge 範例函數。	2019 年 8 月 1 日

更新的監測圖	內容更新說明可讓您直接從 CloudFront 主控台監視與 CloudFront 分發相關聯的 Lambda 函數的新方法，以便更輕鬆地追蹤和偵錯錯誤。如需詳細資訊，請參閱監視 CloudFront。	2019 年 6 月 20 日
整合式安全內容	新的安全章節整合了有關資料保護、IAM、記錄、合規等 CloudFront 功能和實作的相關資訊。如需詳細資訊，請參閱安全性。	2019 年 5 月 24 日
現在需要域驗證	CloudFront 現在需要您使用 SSL 證書來驗證您是否有權使用替代域名與分發。如需詳細資訊，請參閱使用備用網域名稱與 HTTPS。	2019 年 4 月 9 日
更新的 PDF 文件名	Amazon CloudFront 開發人員指南的新文件名是： AmazonCloudFront_DevGuide。先前檔名是：cf-dg。	2019 年 1 月 7 日
新功能	CloudFront 現在支持基於 TCP 的協議 WebSocket，當您需要客戶端和服務器之間的長壽命連接時非常有用。您現在也可以針對需要高可用性的案例設定來源容錯移轉。 CloudFront 如需詳細資訊，請參閱 WebSocket 搭配 CloudFront 發佈使用和使用 CloudFront 來源容錯移轉最佳化高可用性。	2018 年 11 月 20 日

新功能	CloudFront 現在支援執行 Lambda 函數的 HTTP 要求的詳細錯誤記錄功能。您可以將日誌存儲在中 CloudWatch 並使用它們來幫助解決當函數返回無效響應時的 HTTP 5xx 錯誤。如需詳細資訊，請參閱 Lambda 函數的 CloudWatch 指標和 CloudWatch 記錄。	2018 年 10 月 8 日
新功能	您現在可以選擇讓 Lambda@Edge 在可寫入的 HTTP 方法 (POST、PUT 和 DELETE 等) 中公開請求的內文，如此您就能在 Lambda 函數中存取該內文。您可以選擇唯讀存取或指定您將替換內文。如需詳細資訊，請參閱選擇 Include Body (包含內文) 選項以存取請求內文。	2018 年 8 月 14 日
新功能	CloudFront 現在支援提供使用 brotli 或其他壓縮演算法壓縮的內容，以及或取代 gzip。如需詳細資訊，請參閱提供壓縮檔案。	2018 年 7 月 25 日
重組	Amazon 開 CloudFront 發人員指南經過重新整理，以簡化尋找相關內容，並改善可掃描性和瀏覽。	2018 年 6 月 28 日

新功能	Lambda@Edge 現允許您在面對原始伺服器的事件內存取其他標頭 (包括自訂標頭)，以讓您進一步自訂 Amazon S3 儲存貯體所存放內容的交付方式。如需詳細資訊，請參閱下列根據瀏覽者位置和瀏覽者裝置類型顯示個人化內容的範例。	2018 年 3 月 20 日
新功能	您現在可以使用 Amazon 使 CloudFront 用橢圓曲線數位簽章演算法 (ECDSA) 協商 HTTPS 連線到原點。ECDSA 使用的金鑰更小更快，但與舊版 RSA 演算法一樣安全。如需詳細資訊，請參閱支援的 SSL/TLS 通訊協定與您的來源之間通訊的密碼以 CloudFront 及關於 R SA 和 ECDSA 密碼。	2018 年 3 月 15 日
新功能	Lambda @Edge 允許您執行 Lambda 函數來回應 Amazon CloudFrontreceives 來自原始伺服器的 HTTP 錯誤，讓您能夠自訂來源的錯誤回應。如需詳細資訊，請參閱下列顯示重新導向至其他地點以及用狀態代碼 200 (OK) 產生回應的範例。	2017 年 12 月 21 日
新功能	新 CloudFront 功能，即欄位層級加密功能，可協助您進一步加強敏感資料的安全性，例如信用卡號碼或個人識別資訊 (PII)，例如社會安全號碼。如需詳細資訊，請參閱使用欄位層級加密協助保護機密資料。	2017 年 12 月 14 日

文檔歷史存檔	舊文件歷史記錄已存檔。	2017 年 12 月 1 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。