



AWS 白皮書

# WordPress 上的最佳實務 AWS



# WordPress 上的最佳實務 AWS: AWS 白皮書

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

# Table of Contents

摘要 .....	1
您是 Well-Architected 嗎？ .....	1
簡介 .....	2
簡單部署 .....	3
考量事項 .....	3
可用方法 .....	3
Amazon Lightsail .....	4
選取 Amazon Lightsail 定價計畫 .....	4
安裝 WordPress .....	4
從失敗中復原 .....	5
提高效能和成本效率 .....	6
加速內容交付 .....	6
靜態內容卸載 .....	7
動態內容 .....	7
資料庫快取 .....	8
位元碼快取 .....	8
彈性部署 .....	10
參考架構 .....	10
擴展 Web 層 .....	11
無狀態 Web 層 .....	12
共用儲存 ( Amazon S3 和 AmazonEFS ) .....	13
資料層 ( Amazon Aurora 和 Amazon ElastiCache ) .....	14
結論 .....	15
作者群 .....	16
文件修訂 .....	17
附錄 A：CloudFront 組態 .....	18
原始伺服器 and 行為 .....	18
CloudFront 分佈建立 .....	18
附錄 B：靜態內容組態 .....	21
建立使用者 .....	21
Amazon S3 儲存貯體建立 .....	21
靜態原始伺服器建立 .....	22
附錄 C：備份和復原 .....	24
附錄 D：部署新外掛程式和佈景主題 .....	26

---

注意 .....	27
AWS 詞彙表 .....	28
.....	xxix

# WordPress 上的最佳實務 AWS

發佈日期：2021 年 10 月 19 日 ( [文件修訂](#) )

本白皮書為系統管理員提供特定指引，說明如何在 Amazon Web Services ( AWS ) WordPress 上開始使用，以及如何改善部署的成本效率以及最終使用者體驗。它還概述了解決常見可擴展性和高可用性要求的參考架構。

## 您是 Well-Architected 嗎？

[AWS Well-Architected 架構](#)可協助您了解在雲端建置系統時所做出決策的優缺點。架構的六個支柱可讓您學習架構最佳實務，以設計和操作可靠、安全、高效、符合成本效益且永續的系統。使用 [AWS Well-Architected Tool](#)免費提供的 [AWS 管理主控台](#)，您可以針對這些最佳實務檢閱工作負載，方法是回答每個支柱的一組問題。

如需雲端架構的更多專家指引和最佳實務，請參閱[AWS 架構中心](#)：參考架構部署、圖表和白皮書。

# 簡介

WordPress 是以 PHP 和 MySQL 為基礎的開放原始碼部落格工具及內容管理系統 (CMS)，不管是個人部落格或高流量網站，都可提供支援。

WordPress 的第一個版本於 2003 年推出，建置時並未考慮到現在的彈性及可擴展以雲端為基礎的基礎設施。透過 WordPress 社群的努力以及不斷推出各種 WordPress 模組，此 CMS 解決方案的功能也不斷地擴充。現在建立的 WordPress 架構已能夠充分利用 AWS 雲端的各種優點。

## 簡單部署

對於沒有嚴格高可用性需求的低流量部落格或網站，可能適合簡單部署單一伺服器。這種部署不是最有彈性或可擴展的架構，但它是讓網站啟動和執行的最快速且最經濟的方式。

主題

- [考量事項](#)
- [可用方法](#)
- [Amazon Lightsail](#)

## 考量事項

此討論從單一 Web 伺服器部署開始。有時候，您可能會超出範圍，例如：

- WordPress 網站部署的虛擬機器是單一故障點。此執行個體的問題會導致網站的服務損失。
- 擴展資源以提高效能只能透過「垂直擴展」來實現，也就是說，透過增加執行 WordPress 網站的虛擬機器的大小來實現。

## 可用方法

AWS 有許多不同的選項可用於佈建虛擬機器。在上託管您自己的 WordPress 網站有三種主要方式 AWS：

- Amazon Lightsail
- Amazon Elastic Compute Cloud ( Amazon EC2 )
- AWS Marketplace

[Amazon Lightsail](#) 是一項服務，可讓您快速啟動虛擬私有伺服器（ Lightsail 執行個體 ）來託管 WordPress 網站。如果您不需要高度可設定的執行個體類型或存取進階聯網功能，Lightsail 是開始工作的最簡單方法。

[Amazon EC2](#) 是一種 Web 服務，可提供可調整大小的運算容量，因此您可以在幾分鐘內啟動虛擬伺服器。Amazon EC2 提供比 Lightsail 更多的組態和管理選項，這是更進階的架構中理想的選項。您可以管理執行個體，EC2 並可安裝您選擇的任何軟體套件，包括 WordPress。

[AWS Marketplace](#) 是線上商店，您可以在其中尋找、購買和快速部署在 AWS 上執行的軟體。您可以使用一-Click式部署，在幾分鐘內直接將預先設定 WordPress 的影像啟動至您自己的AWS帳戶中EC2的 Amazon。有許多提供 ready-to-run WordPress 執行個體的 Marketplace 供應商。

此白皮書涵蓋 Lightsail 選項，作為單一伺服器 WordPress 網站的建議實作。

## Amazon Lightsail

對於需要簡單虛擬私有伺服器（VPS）解決方案AWS的開發人員、小型企業、學生和其他使用者而言，Lightsail 是最簡單的入門方式。

該服務會摘要許多較複雜的基礎設施管理元素，遠離使用者。因此，如果您的基礎設施體驗較少，或需要專注於執行網站和簡化產品時，這是理想的起點，足以滿足您的需求。

透過 Amazon Lightsail，您可以選擇 Windows 或 Linux/Unix 作業系統和熱門 Web 應用程式，包括 WordPress，並從預先設定的範本按一下即可部署這些應用程式。

隨著您的需求增加，您能夠順利走出初始界限，並連接至其他AWS資料庫、物件儲存、快取和內容分發服務。

## 選取 Amazon Lightsail 定價計畫

[Lightsail 計畫](#)會定義您用來託管 WordPress 網站的 Lightsail 資源的每月成本。有許多計畫可用於涵蓋各種使用案例，具有不同層級CPU的資源、記憶體、固態硬碟（SSD）儲存和資料傳輸。如果您的網站很複雜，您可能需要具有更多資源的大型執行個體。您可以使用 [Web 主控台](#)或 [Amazon Lightsail 文件中所述](#)，將伺服器遷移至更大的計畫來實現此目標。 [Amazon Lightsail CLI](#)

## 安裝 WordPress

Lightsail 提供常用應用程式的範本，例如 WordPress。此範本是執行您自己的 WordPress 網站的絕佳起點，因為它預先安裝了您所需的大多數軟體。您可以使用瀏覽器內終端機或您自己的SSH用戶端，或透過管理 Web 介面安裝其他軟體或 WordPress 自訂軟體組態。

Amazon Lightsail 與 GoDaddy Pro Site 產品建立合作夥伴關係，協助客戶免費 WordPress 輕鬆管理執行個體。Lightsail WordPress 虛擬伺服器已預先設定並最佳化，以實現快速效能和安全性，讓您的 WordPress 網站能夠快速啟動和執行。執行多個 WordPress 執行個體的客戶發現更新、維護和管理其所有網站具有挑戰性和耗時性。透過此整合，您只需按幾下滑鼠，就能在幾分鐘內輕鬆管理多個 WordPress 執行個體。

如需有關安裝 Lightsail 後在 Lightsail WordPress 上管理的詳細資訊，請參閱 [WordPress 從您的 Amazon Lightsail 執行個體開始使用](#)。完成 WordPress 網站自訂後，建議您拍攝執行個體的快照。

**快照**是建立 Lightsail 執行個體備份映像的一種方式。這是系統磁碟的複本，也會儲存原始機器組態（即記憶體、CPU磁碟大小和資料傳輸率）。快照可用於在部署或升級不良後還原至已知良好的組態。

此快照可讓您視需要復原伺服器，但也可以啟動具有相同自訂的新執行個體。

## 從失敗中復原

單一 Web 伺服器是單一失敗點，因此您必須確保網站資料已備份。上述快照機制也可以用於此目的。若要從失敗中復原，您可以從最近的快照還原新的執行個體。若要減少還原期間可能遺失的資料量，您的快照必須盡可能保持最新狀態。

為了將資料遺失的可能性降至最低，請確保定期拍攝快照。您可以排程 Lightsail Linux/Unix 執行個體的自動快照。如需步驟，請參閱[啟用或停用 Amazon Lightsail 中執行個體或磁碟的自動快照](#)。

AWS 建議您使用靜態 IP：專用於 Lightsail 帳戶的固定公有 IP 地址。如果您需要將執行個體取代為另一個執行個體，您可以將靜態 IP 重新指派給新的執行個體。如此一來，您不必在每次想要取代執行個體時，重新設定任何外部系統（例如 DNS 記錄）來指向新的 IP 地址。

# 提高效能和成本效率

您最終可能會超出單一伺服器部署的增長範圍。在此情況下，您可能需要考慮提高網站效能的選項。在遷移到多伺服器、可擴展的部署之前 (本白皮書稍後會討論)，您可以套用許多效能和成本效益。這些是您應該遵循的良好做法，即使您確實移至多伺服器架構也是如此。

以下小節會介紹可改善您的 WordPress 網站的效能和可擴展性面向的一些選項。有些可以套用至單一伺服器部署，而其他則可利用多伺服器的可擴展性。這些修改的許多部分需要使用一或多個 WordPress 外掛程式。雖然各種選項都可用，[W3 總快取](#) 是一個流行的選擇，可將這些修改當中的許多結合在單一外掛程式中。

## 主題

- [加速內容交付](#)
- [資料庫快取](#)
- [位元碼快取](#)

## 加速內容交付

任何 WordPress 網站需要提供靜態和動態內容的混合。靜態內容包括影像、JavaScript 檔案或樣式表。動態內容包括使用 WordPress PHP 程式碼在伺服器端產生的任何內容，例如，從資料庫產生或針對每個檢視器個人化的網站元素。

最終使用者體驗的一個重要面向是向世界各地的使用者提供先前的內容時所牽涉的網路延遲。加速先前內容的交付可改善最終使用者體驗，特別是實體上散佈在全球各地的使用者。這可以透過 Amazon CloudFront 之類的內容交付網路 (CDN) 來實現。

[Amazon CloudFront](#) 是一種 Web 服務，提供一種簡單且具成本效益的方式，透過全球多個節點以低延遲和高資料傳輸速度來分送內容。檢視器請求會自動路由到合適的 CloudFront [節點](#)，以降低延遲。如果可以快取內容 (幾秒鐘、幾分鐘甚至幾天)，並且已存放在特定的節點，則 CloudFront 會立即交付它。如果該內容不應被快取、已過期或目前不在該節點，則 CloudFront 會從 CloudFront 組態中的一或多個真實來源 (在此情況中的 Lightsail 執行個體) 擷取內容。此擷取會透過最佳化網路連線進行，這有助於加快您的網站上內容的交付速度。除了改善最終使用者體驗之外，所討論的模型還可以降低原始伺服器上的負載，並有可能創造顯著成本節省。

## 靜態內容卸載

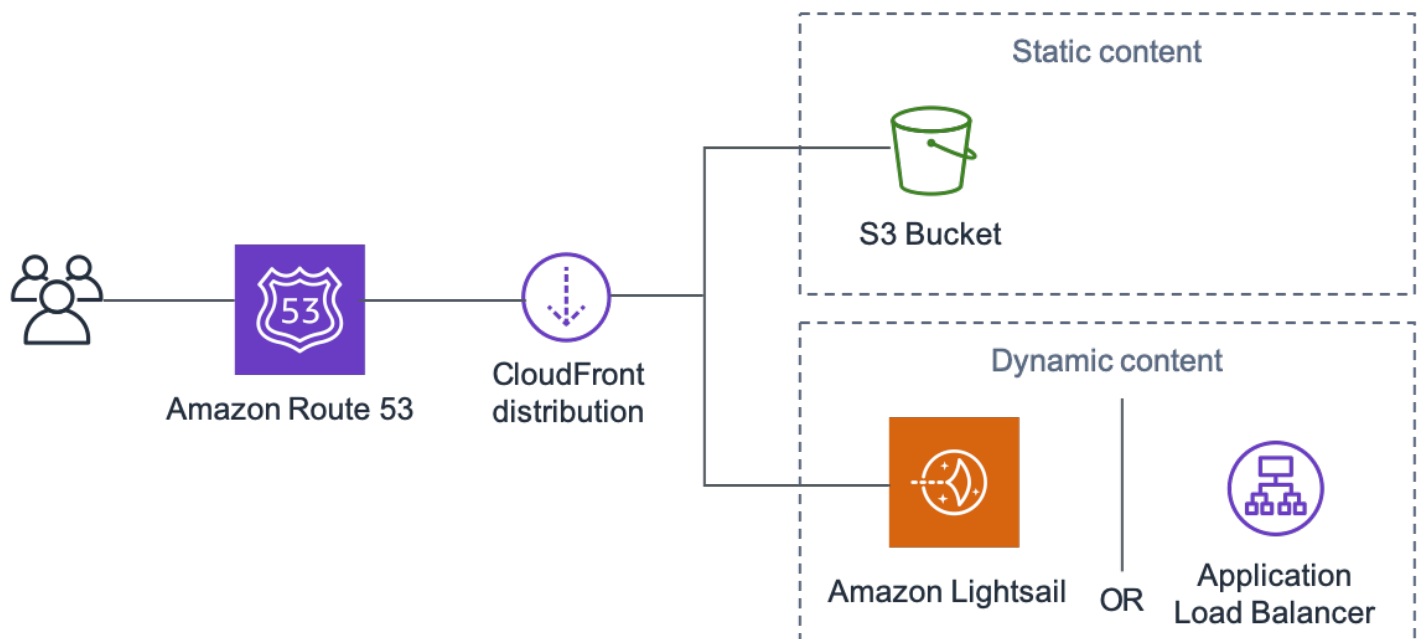
這包括 CSS、JavaScript 和影像檔案，這些是您的 WordPress 佈景主題的一部分，或內容管理員上傳的那些媒體檔案。這所有檔案都可以使用 W3 總快取之類的外掛程式存放在 Amazon Simple Storage Service (Amazon S3) 中，並以可擴展和高度可用的方式提供給使用者。[Amazon S3](#) 以低成本提供高度可擴展、可靠且低延遲的資料儲存基礎設施，可透過 REST API 存取。Amazon S3 會備援地存放您的物件，不僅會在多個裝置上，還可以跨 AWS 區域的多個設施，從而提供極高的耐久性。

這會產生正面的副作用，即將此工作負載從 Lightsail 執行個體中卸載，並讓它專注於動態內容產生。這可減少伺服器上的負載，並且是建立無狀態架構的一個重要步驟 (實作自動擴展之前的先決條件)。

您可以後續將 Amazon S3 設定為 CloudFront 的來源，以改善向全球各地的使用者交付這些靜態資產的方式。雖然 WordPress 沒有與 Amazon S3 和 CloudFront 立即可用的整合，但各種外掛程式為這些服務新增支援 (例如，W3 總快取)。

## 動態內容

動態內容包括伺服器端 WordPress PHP 指令碼的輸出。動態內容也可以透過將 WordPress 網站設定為來源，而透過 CloudFront 提供。由於動態內容包括個人化內容，因此您需要設定 CloudFront 以將某些 HTTP cookie 和 HTTP 標頭作為請求的一部分轉送到您的自訂原始伺服器。CloudFront 使用轉送的 cookie 值作為可識別其快取中唯一物件的金鑰一部分。若要確保將快取效率最大化，您應該將 CloudFront 設定為僅轉送實際上會變更內容的 HTTP cookie 和 HTTP 標頭 (而非僅在用戶端上或由第三方應用程式使用的 cookie，例如用於 Web 分析)。



透過 Amazon CloudFront 進行整個網站交付

上圖包含兩個來源：一個用於靜態內容，而另一個用於動態內容。如需實作的詳細資訊，請參閱[附錄 A：CloudFront 組態](#)和[附錄 B：外掛程式安裝和組態](#)。

CloudFront 使用標準快取控制標頭來識別是否應快取特定 HTTP 回應以及快取多長時間。Web 瀏覽器還使用相同的快取控制標頭來決定何時以及要快取內容多長時間，以獲得更佳的最終使用者體驗 (例如，已下載的 .css 檔案不會在每次回訪的訪客檢視頁面時重新下載)。您可以在 Web 伺服器層級設定快取控制標頭 (例如，透過 .htaccess 檔案或 httpd.conf 檔案的修改)，或安裝 WordPress 外掛程式 (例如，W3 總快取) 來指出如何為靜態和動態內容設定這些標頭。

## 資料庫快取

資料庫快取可以顯著減少延遲並提高包含大量讀取應用程式 (如 WordPress) 工作負載的輸送量。透過將頻繁存取的資料片段儲存在記憶體中以實現低延遲存取 (例如，I/O 密集型資料庫查詢的結果)，從而提高應用程式效能。當大部分查詢都從快取中提供時，需要存取資料庫的查詢數量會減少，因而降低與執行資料庫相關聯的成本。

雖然 WordPress 具有立即可用的有限快取功能，但各種外掛程式支援使用 [Memcached](#) 整合，這是一個廣泛採用的記憶體物件快取系統。W3 總快取外掛程式就是一個很好的例子。

在最簡單的方案中，您可以在 Web 伺服器上安裝 Memcached 並將結果擷取為新快照。在此情況下，您會負責與執行中快取相關聯的管理任務。

另一種選項是利用 [Amazon ElastiCache](#) 之類的受管服務並避免該營運負擔。ElastiCache 使您可以輕鬆在雲端部署、操作和擴展分散式記憶體內快取。您可以在 [Amazon ElastiCache 文件](#) 中找到如何連線到 ElastiCache 叢集節點的相關資訊。

如果您正在使用 Lightsail，並希望私密存取 AWS 帳戶中的 ElastiCache 叢集，則可以使用 VPC 對等互連來執行此動作。如需啟用 VPC 對等互連的指示，請參閱[設定 Amazon VPC 對等互連以在 Amazon Lightsail 外部使用 AWS 資源](#)。

## 位元碼快取

每次執行 PHP 指令碼時，它都會被剖析和編譯。透過使用 PHP 位元碼快取，PHP 編譯的輸出會存放在 RAM 中，因此不必重複地編譯相同的指令碼。這樣可以減少與執行 PHP 指令碼相關的負擔，從而提高效率並降低 CPU 需求。

位元碼快取可以安裝在託管 WordPress 的任何 Lightsail 執行個體上，並可以大幅減少其負擔。對於 PHP 5.5 及更新版本，AWS 建議使用 [OPcache](#)，這是一個與該 PHP 版本組合的延伸。

請注意，OPcache 會預設在 Bitnami WordPress Lightsail 範本中啟用，因此不需要進一步的動作。

## 彈性部署

在許多情況下，單一伺服器部署可能不足以用於您的網站。在這些情況下，您需要多伺服器、可擴展的架構。

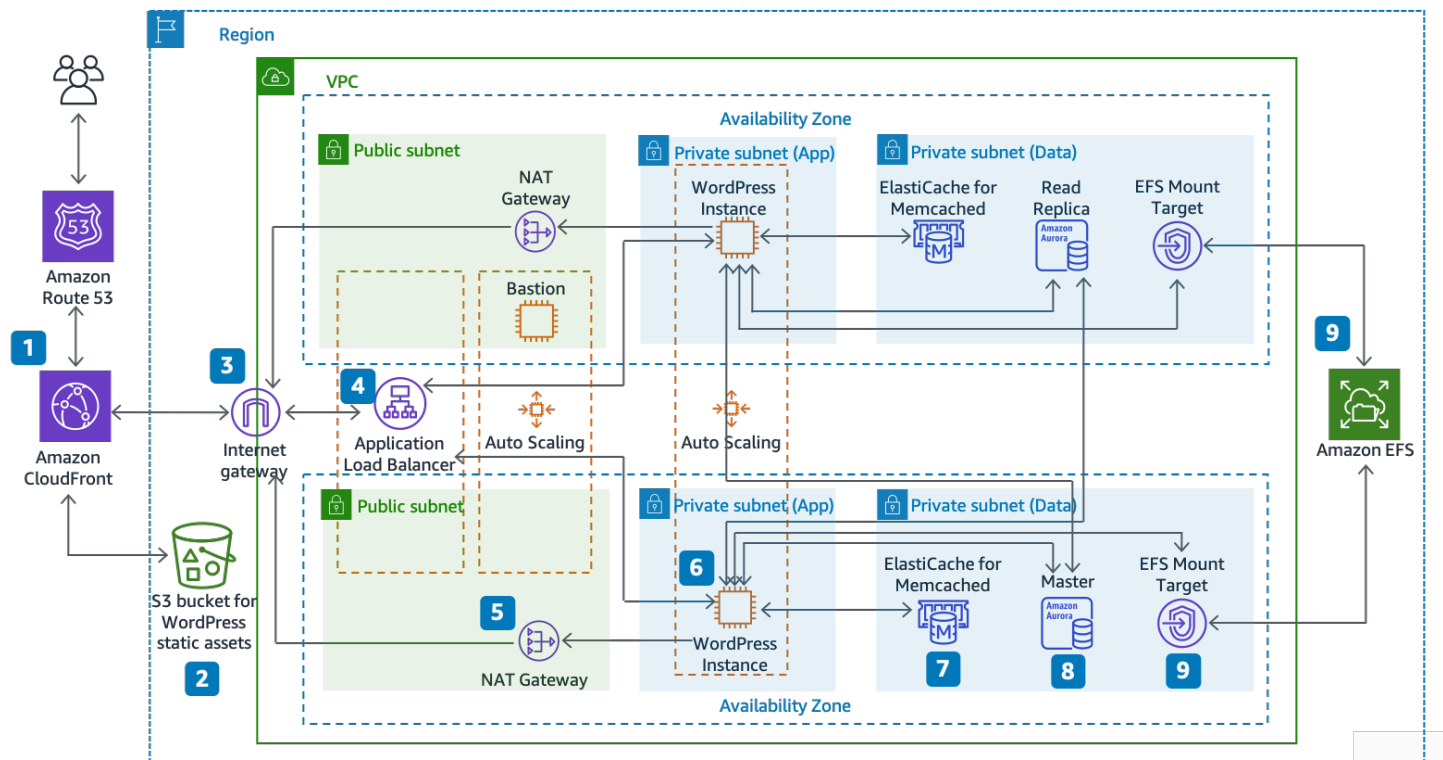
主題

- [參考架構](#)
- [擴展 Web 層](#)
- [無狀態 Web 層](#)

## 參考架構

上的 GitHub [參考 AWS 架構 WordPress 託管](#) 概述了在 WordPress 上部署的最佳實務，AWS 並包含一組 AWS CloudFormation 範本，可讓您快速啟動和執行。下列架構是以該參考架構為基礎。本節其餘部分將檢閱架構選擇背後的原因。

AMI 中的 GitHub 已於 2021 年 7 月從 Amazon Linux1 變更為 Amazon Linux2。不過，S3 的部署範本尚未變更。GitHub 如果 S3 使用範本部署參考架構時發生問題，建議您在 使用範本。



WordPress 用於託管的參考架構 AWS

## 架構元件

參考架構說明 上 WordPress 網站的完整最佳實務部署AWS。

- 它從 Amazon CloudFront ( 1 ) 中的邊緣快取開始，以快取接近最終使用者的內容，以加快交付速度。
- CloudFront 從 Web 執行個體前面的 S3 儲存貯體 ( 2 ) 提取靜態內容，並從 Application Load Balancer ( 4 ) 提取動態內容。
- Web 執行個體在 Auto Scaling 群組中的 Amazon EC2執行個體 ( 6 ) 中執行。
- ElastiCache 叢集 ( 7 ) 會快取經常查詢的資料，以加快回應速度。

Amazon Aurora MySQL 執行個體 ( 8 ) WordPress託管資料庫。

- 執行個體會透過每個可用區域中的EFS掛載目標 WordPress EC2 ( 9 ) 存取 Amazon EFS 檔案系統上的共用 WordPress 資料。
- 網際網路閘道 ( 3 ) 可啟用 VPC和網際網路中資源之間的通訊。
- 每個可用區域中的NAT閘道 ( 5 ) 可讓私有子網路 ( 應用程式和資料 ) 中的EC2執行個體存取網際網路。

在 Amazon VPC 中有兩種子網路類型：公有 ( 公有子網路 ) 和私有 ( 應用程式子網路和資料子網路 )。部署到公有子網路的資源將會收到公有 IP 地址，並將在網際網路上公開可見。Application Load Balancer ( 4 ) 和用於管理的 Bastion 主機部署在此處。部署到私有子網路的資源只會收到私有 IP 地址，因此不會在網際網路上公開顯示，從而提高這些資源的安全性。WordPress Web 伺服器執行個體 ( 6 )、ElastiCache叢集執行個體 ( 7 )、Aurora MySQL 資料庫執行個體 ( 8 ) 和EFS掛載目標 ( 9 ) 都部署在私有子網路中。

本節的其餘部分更詳細地涵蓋了這些考量。

## 擴展 Web 層

若要將單一伺服器架構演變為多伺服器、可擴展的架構，您必須使用五個關鍵元件：

- Amazon EC2執行個體
- Amazon Machine Images ( AMIs )
- 負載平衡器
- 自動調整規模

## • 運作狀態檢查

AWS 提供各種 EC2 執行個體類型，讓您可以選擇效能和成本的最佳伺服器組態。一般而言，運算最佳化（例如 C4）執行個體類型可能是 WordPress Web 伺服器的最佳選擇。您可以在 AWS 區域內的多個可用區域部署執行個體，以提高整體架構的可靠性。

由於您可以完全控制 EC2 執行個體，因此您可以使用根存取權登入，以安裝和設定執行 WordPress 網站所需的所有軟體元件。完成後，您可以將該組態儲存為 AMI，您可以使用它來啟動具有所有自訂項目的新執行個體。

若要將最終使用者請求分發至多個 Web 伺服器節點，您需要負載平衡解決方案。AWS 透過 [Elastic Load Balancing](#) 提供此功能，這是一種高可用性服務，可將流量分散至多個 EC2 執行個體。由於您的網站透過 HTTP 或為使用者提供內容 HTTPS，因此我們建議您使用 Application Load Balancer、具有內容路由的應用程式層負載平衡器，以及在必要時在不同網域上執行多個 WordPress 網站的能力。

Elastic Load Balancing 支援在 AWS 區域內多個可用區域之間分發請求。您也可以設定運作狀態檢查，讓 Application Load Balancer 自動停止將流量傳送至失敗的個別執行個體（例如，由於硬體問題或軟體當機）。AWS 建議使用 WordPress 管理員登入頁面（/wp-login.php）進行運作狀態檢查，因為此頁面會確認 Web 伺服器正在執行，以及 Web 伺服器已設定為正確提供 PHP 檔案。

您可以選擇建置自訂運作狀態檢查頁面，以檢查其他相依資源，例如資料庫和快取資源。如需詳細資訊，請參閱 Application Load Balancer 指南中的 [目標群組運作狀態檢查](#)。

彈性是 AWS Cloud 的關鍵特徵。您可以在需要時啟動更多運算容量（例如 Web 伺服器），而不需要時則執行更少。[Amazon EC2 Auto Scaling](#) 是一項 AWS 服務，可協助您自動化此佈建，根據您定義的條件來擴展或縮減 Amazon EC2 容量，而不需要手動介入。您可以設定 Amazon EC2 Auto Scaling，以便在需求激增期間無縫增加使用中的 EC2 執行個體數量，以維持效能並在流量減少時自動減少，以將成本降至最低。

Elastic Load Balancing 也支援動態新增和移除負載平衡輪換中的 Amazon EC2 主機。Elastic Load Balancing 本身也會動態增加和減少負載平衡容量，以適應流量需求，而無需手動介入。

## 無狀態 Web 層

若要在自動擴展組態中利用多個 Web 伺服器，您的 Web 層必須是無狀態的。無狀態應用程式是不需要了解先前互動且不會儲存工作階段資訊的應用程式。對於 WordPress，這表示所有最終使用者都會收到相同的回應，無論哪個 Web 伺服器處理了其請求。無狀態應用程式可以水平擴展，因為任何可用的運算資源（即 Web 伺服器執行個體）都可以為任何請求提供服務。當不再需要該容量時，任何個

別資源都可以安全地終止（在已耗盡執行中的任務之後）。這些資源不需要知道同事的存在，只需要將工作負載分發給他們。

在使用者工作階段資料儲存方面，WordPress 核心完全無狀態，因為它依賴存放在用戶端 Web 瀏覽器中的 Cookie。除非您安裝了任何自訂程式碼（例如 WordPress 外掛程式），而這些程式碼完全依賴原生 PHP 工作階段，否則工作階段儲存就不是問題。

不過，WordPress 最初的設計是要在單一伺服器上執行。因此，它會在伺服器的本機檔案系統上存放一些資料。在多伺服器組態 WordPress 中執行時，這會產生問題，因為 Web 伺服器之間存在不一致。例如，如果使用者上傳新映像，則只會儲存在其中一個伺服器上。

這說明了為什麼我們需要改進預設 WordPress 執行中組態，才能將重要資料移至共用儲存體。最佳實務架構具有資料庫作為 Web 伺服器外部的個別層，並使用共用儲存來存放使用者上傳、主題和外掛程式。

## 共用儲存（Amazon S3 和 AmazonEFS）

根據預設，會將使用者上傳 WordPress 存放在本機檔案系統上，因此不是無狀態。因此，我們需要將 WordPress 安裝和所有使用者自訂（例如組態、外掛程式、主題和使用者產生的上傳）移至共用資料平台，以協助減少 Web 伺服器的負載，並讓 Web 層無狀態。

[Amazon Elastic File System](#)（Amazon EFS）提供可擴展的網路檔案系統，可與 EC2 執行個體搭配使用。Amazon EFS 檔案系統分散在不受限制的儲存伺服器數量，讓檔案系統彈性成長，並允許從 EC2 執行個體進行大規模平行存取。Amazon 的分散式設計可避免傳統檔案伺服器固有的瓶頸和限制。

透過將整個 WordPress 安裝目錄移動到 EFS 檔案系統上，並在啟動時將其掛載到每個 EC2 執行個體，您的 WordPress 網站及其所有資料會自動存放在不依賴於任何一個 EC2 執行個體的分散式檔案系統上，讓您的 Web 層完全無狀態。此架構的優點是您不需要在每次新執行個體啟動時安裝外掛程式和主題，而且可以大幅加快 WordPress 執行個體的安裝和復原速度。在中部署外掛程式和主題的變更也比較容易 WordPress，如本文件的[部署考量](#)一節所述。

若要確保從 EFS 檔案系統執行時網站的最佳效能，請檢查適用於 Amazon EFS 和 [AWS 參考架構 WordPress OPcache](#) 的建議組態設定。

您也可以選擇將所有靜態資產，例如映像、CSS 和 JavaScript 檔案，卸載至 S3 儲存貯體，同時 CloudFront 快取在前面。在多伺服器架構中執行此操作的機制與單一伺服器架構完全相同，如本白皮書的[靜態內容](#) 章節所述。優點與單一伺服器架構相同：您可以將與為靜態資產提供服務相關聯的工作卸載至 Amazon S3 和 CloudFront，藉此讓您的 Web 伺服器僅專注於產生動態內容，並為每個 Web 伺服器提供更多使用者請求。

## 資料層 ( Amazon Aurora 和 Amazon ElastiCache )

透過存放在分散式、可擴展、共用網路檔案系統的 WordPress 安裝，以及從 Amazon S3 提供的靜態資產，您可以將注意力集中在剩餘的具狀態元件：資料庫。與儲存層一樣，資料庫不應依賴任何單一伺服器，因此無法託管在其中一個 Web 伺服器上。而是在 Amazon Aurora 上託管 WordPress 資料庫。

[Amazon Aurora](#) 是為雲端建置的 MySQL 和 PostgreSQL 相容關聯式資料庫，結合了高階商業資料庫的效能和可用性，以及開放原始碼資料庫的簡單性和成本效益。Aurora MySQL 透過緊密整合資料庫引擎與由支援的專用分散式儲存系統，來提高我的 SQL 效能和可用性 SSD。它具有容錯性和自我修復功能，可跨三個可用區域複寫六份資料，旨在實現超過 99.99% 的可用性，並在 Amazon S3 中持續備份您的資料。Amazon Aurora 旨在自動偵測資料庫當機和重新啟動，而不需要當機復原或重建資料庫快取。

Amazon Aurora 提供多種[執行個體類型](#)，以適應不同的應用程式設定檔，包括記憶體最佳化和爆量執行個體。若要改善資料庫的效能，您可以選擇大型執行個體類型，以提供更多 CPU 和記憶體資源。

Amazon Aurora 會自動處理主要執行個體和 [Aurora 複本](#) 之間的容錯移轉，讓您的應用程式可以盡快恢復資料庫操作，而無需手動管理介入。容錯移轉通常需要不到 30 秒的時間。

建立至少一個 Aurora 複本後，請使用叢集端點連線至您的主要執行個體，以允許應用程式在主要執行個體失敗時自動容錯移轉。您可以在三個可用區域中建立最多 15 個低延遲僅供讀取複本。

隨著資料庫擴展，您的資料庫快取也需要擴展。如先前在[資料庫快取](#)章節中所述，ElastiCache 具有將快取擴展到 ElastiCache 叢集中多個節點，以及區域中多個可用區域的功能，以提高可用性。當您擴展 ElastiCache 叢集時，請確定您設定快取外掛程式以使用組態端點進行連線，以便 WordPress 可以在新增叢集節點時使用新的叢集節點，並在移除舊叢集節點時停止使用它們。您也必須設定 Web 伺服器以使用 [ElastiCache 的叢集用戶端 PHP](#)，並更新 AMI 以存放此變更。

## 結論

AWS 提供許多用於執行 WordPress 的架構選項。最簡單的選項是用於低流量網站的單一伺服器安裝。對於更進階的網站，網站管理員可以新增其他幾個選項，每個選項都代表在可用性和可擴展性方面的增量改進。管理員可以選取最密切符合其需求和預算的功能。

# 作者群

此文件的作者包括：

- Amazon Web Services 資深解決方案架構師 Paul Lewis
- Amazon Web Services 解決方案架構師 Ronan Guilfoyle
- Amazon Web Services 解決方案架構師 Andreas Chatzakis
- Amazon Web Services 資深技術客戶經理 Jibril Touzi
- Amazon Web Services 遷移合作夥伴解決方案架構師 Hakmin Kim

## 文件修訂

若要收到此白皮書更新的相關通知，請訂閱RSS摘要。

變更	描述	日期
<a href="#">白皮書已更新</a>	更新以修改參考架構和 AWS WordPress 外掛程式。	2021 年 10 月 19 日
<a href="#">白皮書已更新</a>	更新以包含新的部署方法和 AWS WordPress 外掛程式。	2019 年 10 月 30 日
<a href="#">白皮書已更新</a>	更新以說明 Amazon Aurora 產品訊息。	2018 年 2 月 1 日
<a href="#">白皮書已更新</a>	更新以包含自第一次發佈以來啟動AWS的服務。	2017 年 12 月 1 日
<a href="#">初次出版</a>	首次發佈。	2014 年 12 月 1 日

## 附錄 A：CloudFront 組態

若要在 Amazon CloudFront 搭配您的 WordPress 網站使用時獲得最佳效能和效率，請務必針對提供的不同類型內容正確設定網站。

主題

- [原始伺服器 and 行為](#)
- [CloudFront 分佈建立](#)

### 原始伺服器和行為

[原始](#)伺服器是 CloudFront 傳送透過邊緣位置分發之內容請求的位置。根據您的實作，您可以擁有一或兩個原始伺服器。一個用於使用自訂原始伺服器之動態內容（[單一伺服器部署選項](#) 中的 Lightsail 執行個體，或[彈性部署選項](#) 中的 Application Load Balancer）。您可能有第二個原始伺服器來針對靜態內容 CloudFront 導向。在上述[參考架構](#)中，這是 S3 儲存貯體。當您使用 Amazon S3 作為分發的原始伺服器時，您需要使用[儲存貯體政策](#)來公開存取內容。

[行為](#)可讓您設定規則，以控制 CloudFront 快取內容的方式，進而判斷快取的有效性。行為可讓您控制網站可存取的通訊協定和HTTP方法。它們也可讓您控制是否要將HTTP標頭、Cookie 或查詢字串傳遞至後端（如果是，則傳遞哪些字串）。行為適用於特定URL路徑模式。

### CloudFront 分佈建立

按照分佈建立 CloudFront Web 分佈，自動建立的預設原始伺服器和行為將用於動態內容。建立四個額外的行為，以進一步自訂處理靜態和動態請求的方式。下表摘要說明五種行為的組態屬性。

表 1：CloudFront 行為的組態屬性摘要

屬性	靜態	動態（管理員）	動態（前端）
路徑（行為）	wp-content/* wp-includes/*	wp-admin/* wp-login.php	預設（*）
通訊協定	HTTP 和 HTTPS	重新導向至 HTTPS	HTTP 和 HTTPS
HTTP 方法	GET, HEAD	ALL	ALL

屬性	靜態	動態 ( 管理員 )	動態 ( 前端 )
HTTP 標頭	NONE	ALL	主機 CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookie	NONE	ALL	comment_* wordpress_* wp-settings-*
查詢字串	YES ( 驗證中 )	YES	YES

對於預設行為，AWS建議下列組態：

- 允許原始伺服器通訊協定政策相符檢視器，以便如果檢視者使用連線至 CloudFront HTTPS，HTTPS也請使用 CloudFront 連線至原始伺服器，從而實現 end-to-end加密。請注意，這需要您在負載平衡器上安裝受信任的SSL憑證。如需詳細資訊，請參閱[要求 CloudFront 與自訂原始伺服器 之間的HTTPS通訊](#)。
- 允許所有HTTP方法，因為網站的動態部分需要 GET和 POST請求（例如，支援POST評論提交表單）。
- 僅轉送不同 WordPress 輸出的 Cookie；例如 >wordpress\_\*、wp-settings-\*和 comment\_\*。如果您已安裝任何依賴不在清單中的其他 Cookie 的外掛程式，則必須延伸該清單。
- 僅轉送影響輸出的HTTP標頭 WordPress，例如 Host、CloudFront-Forwarded-ProtoCloudFront-is-Desktop-Viewer、CloudFront-is-Mobile-Viewer和 CloudFront-is-Tablet-Viewer：

- Host 允許將多個 WordPress 網站託管在相同的原始伺服器。
- CloudFront-Forwarded-Proto 允許快取不同版本的頁面，具體取決於是否透過 HTTP 或 存取HTTPS。
- CloudFront-is-Desktop-Viewer、CloudFront-is-Mobile-ViewerCloudFront-is-Tablet-Viewer可讓您根據最終使用者的裝置類型自訂主題的輸出。
- 將所有查詢字串根據其值轉送至快取，因為它們也 WordPress 可用於使快取的物件失效。

如果您想要以自訂網域名稱（即不是 \*.cloudfront.net）為網站提供服務，請在分發設定中的替代網域名稱URLs下輸入適當的。在這種情況下，您還需要自訂網域名稱的SSL憑證。您可以透過 AWS Certificate Manager [請求SSL憑證](#)，並根據 CloudFront 分佈進行設定。

現在，為動態內容建立另外兩種快取行為：一種用於登入頁面（路徑模式：wp-login.php），另一種用於管理員儀表板（路徑模式：wp-admin/\*）。這兩個行為具有完全相同的設定，如下所示：

- 強制執行HTTPS僅限的檢視器通訊協定政策。
- 允許所有HTTP方法。
- 根據所有HTTP標頭進行快取。
- 轉送所有 Cookie。
- 根據所有查詢字串轉送和快取。

此組態背後的原因是網站的此區段高度個人化，通常只有少數使用者，因此快取效率不是主要問題。重點是讓組態保持簡單，透過將所有 Cookie 和標頭傳遞至原始伺服器，以確保與任何已安裝外掛程式的最大相容性。

依預設，會將所有內容 WordPress 儲存在 Web 伺服器本機上，也就是用於[單一伺服器部署](#)的區塊儲存（Amazon EBS），以及用於[彈性部署](#)的檔案儲存（Amazon EFS）。除了降低儲存和資料傳輸成本之外，將靜態資產移至 Amazon S3 還提供可擴展性、資料可用性、安全性和效能。有幾個外掛程式可讓您輕鬆地將靜態內容移至 Amazon S3；其中一個是 [W3 總快取](#)，也涵蓋在[附錄 B：外掛程式安裝和組態](#)中。

## 附錄 B：靜態內容組態

依預設，會將所有內容 WordPress 儲存在 Web 伺服器本機上，也就是用於[單一伺服器部署](#)的區塊儲存（Amazon EBS），以及用於[彈性部署](#)的檔案儲存（Amazon EFS）。除了降低儲存和資料傳輸成本之外，將靜態資產移至 Amazon S3 還提供可擴展性、資料可用性、安全性和效能。

在此範例中，W3 Total Cache（W3TC）外掛程式用於在 Amazon S3 上存放靜態資產。不過，還有其他具有類似功能的外掛程式可供使用。如果您想要使用替代方案，您可以相應地調整下列步驟。這些步驟僅參考與此範例相關的功能或設定。所有設定的詳細描述超出本文件的範圍。如需詳細資訊，請參閱 [W3 總快取外掛程式頁面](#)：[wordpress.org](http://wordpress.org)。

### 建立使用者

您需要為 WordPress 外掛程式建立使用者，才能將靜態資產存放在 Amazon S3 中。如需步驟，請參閱[在AWS帳戶中建立使用者](#)。

注意：角色提供管理 AWS 資源存取權的更好方法，但在撰寫時，W3 總快取外掛程式不支援[角色](#)。

請記下使用者安全憑證，並以安全的方式存放它們 – 您稍後需要這些憑證。

### Amazon S3 儲存貯體建立

1. 首先，在您選擇的AWS區域中建立 Amazon S3 儲存貯體。如需步驟，請參閱[建立儲存貯體](#)。遵循[教學課程：在 Amazon S3 上設定靜態網站，為儲存貯體啟用靜態網站託管](#)。
2. 建立政策以提供先前建立的使用者存取指定的 S3 儲存貯體，並將政策連接至使用者。如需建立下列政策的步驟，請參閱[管理政策](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1389783689000",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:ListBucket",
```

```

        "s3:PutObject",
        "s3:PutObjectAcl"
    ],
    "Resource": [
        "arn:aws:s3:::wp-demo",
        "arn:aws:s3:::wp-demo/*"
    ]
}
]
}

```

3. 從 WordPress 管理員面板安裝和啟用 W3TC 外掛程式。
4. 瀏覽至外掛程式組態的一般設定區段，並確保CDN已啟用瀏覽器快取和。
5. 從CDN組態中的下拉式清單中，選擇 Origin Push：Amazon CloudFront（此選項以 Amazon S3 作為原始伺服器）。
6. 瀏覽至外掛程式組態的瀏覽器快取區段，並啟用過期的、快取控制和實體標籤（ETag）標頭。
7. 同時啟用設定變更後防止物件快取選項，以便在變更任何設定時產生新的查詢字串並附加到物件。
8. 瀏覽至外掛程式組態的 CDN區段，並輸入您先前建立之使用者的安全憑證，以及 S3 儲存貯體的名稱。
9. 如果您透過 服務網站 CloudFront URL，請在相關方塊中輸入分佈網域名稱。否則，請CNAMEs為自訂網域名稱輸入一或多個（一或多個）。
- 10最後，匯出媒體程式庫，並使用 W3TC 外掛程式將 wp-include、主題檔案和自訂檔案上傳至 Amazon S3。W3TC 這些上傳函數可在CDN組態頁面的一般區段中使用。

## 靜態原始伺服器建立

現在靜態檔案存放在 Amazon S3 上，請返回 CloudFront 主控台 CloudFront 組態，並將 Amazon S3 設定為靜態內容的原始伺服器。若要這麼做，請將第二個原始伺服器新增至您為該目的建立的 S3 儲存貯體。然後建立另外兩個快取行為，兩個資料夾（wp-content 和 wp-includes）中每個都應使用 S3 原始伺服器，而不是動態內容的預設原始伺服器。以相同的方式設定兩者：

- 僅提供HTTPGET請求。
- Amazon S3 不會根據 Cookie 或HTTP標頭變更其輸出，因此您可以透過不透過 將快取轉送到原始伺服器來提高快取效率 CloudFront。
- 雖然這些行為僅提供靜態內容（不接受任何參數），但您會將查詢字串轉送至原始伺服器。這樣您就可以使用查詢字串作為版本識別符，以在部署新版本時立即使舊CSS檔案失效。如需詳細資訊，請參閱 [Amazon CloudFront 開發人員指南](#)。

**Note**

將靜態原始伺服器行為新增至 CloudFront 分發之後，請檢查 順序，以確保的行為 `wp-admin/*` 和 `wp-login.php` 的優先順序高於靜態內容的行為。否則，您可能會在存取管理員面板時看到奇怪的行為。

## 附錄 C：備份和復原

相較於傳統託管環境，在 AWS 中從故障中復原速度更快且更容易。例如，您可以在幾分鐘內便啟動替代執行個體以回應硬體故障，也可以利用我們的許多受管服務中的自動容錯移轉來去除因例行維護而造成重新開機的影響。

但是，您仍然需要確保要備份的是正確的資料，才能成功復原。若要重新建立 WordPress 網站的可用性，您必須能夠復原以下元件：

- 作業系統 (OS) 和服務安裝和組態 (Apache、MySQL 等)
- WordPress 應用程式碼和組態
- WordPress 佈景主題和外掛程式
- 上傳 (例如，貼文的媒體檔案)
- 資料庫內容 (貼文、評論等)

AWS 提供多種備份和還原 Web 應用程式資料和資產的方法。

此白皮書先前討論如何使用 Lightsail 快照來保護存放在執行個體本機儲存上的所有資料。如果您的 WordPress 網站僅複製了 Lightsail 執行個體，一般的 Lightsail 快照應該足以讓您整個復原您的 WordPress 網站。但是，如果您確實從快照復原，您仍將遺失則自上次取得快照以來，對您的網站套用的所有變更。

在多伺服器部署中，您必須使用不同機制來備份稍早討論的每個元件。每個元件對備份頻率可能有不同的要求，例如，作業系統和 WordPress 安裝和組態的變更頻率將遠低於使用者產生的內容，因此，備份頻率可以較低，而不會在復原時遺失資料。

若要備份作業系統和服務的安裝和組態以及 WordPress 應用程式碼和組態，您可以建立正確設定的 EC2 執行個體的 AMI。AMI 可以有兩個用途：充當執行個體狀態的備份，以及在啟動新執行個體時充當範本。

若要備份 WordPress 應用程式碼和組態，您必須使用 AMI 和 Aurora 備份。

若要備份您的網站上安裝的 WordPress 佈景主題和外掛程式，請備份 Amazon S3 儲存貯體或存放其所在的 Amazon EFS 檔案系統。

- 對於存放在 S3 儲存貯體中的佈景主題和外掛程式，您可以啟用[跨區域複寫](#)，使得上傳到主要儲存貯體的所有物件，都會自動複寫到另一個 AWS 區域中的備份儲存貯體。跨區域複寫要求在來源和目的

地儲存貯體上同時啟用[版本控制](#)，這會為您提供額外的一層保護，並可讓您還原為儲存貯體中任何給定物件的舊版。

- 對於存放在 EFS 檔案系統上的佈景主題和外掛程式，您可以建立 AWS Data Pipeline 以將資料從生產 EFS 檔案系統複製到另一個 EFS 檔案系統，如[備份 Amazon EFS 檔案系統](#)文件頁面所述。您還可以使用您已熟悉的任何備份應用程式備份 EFS 檔案系統。
- 若要備份使用者上傳，您應該按照稍早概述的步驟來備份 WordPress 佈景主題和外掛程式。
- 若要備份資料庫內容，您需要使用 [Aurora 備份](#)。Aurora 會自動備份叢集磁碟區，並以備份保留期為還原資料的保留期限。Aurora 備份具有連續和增量性質，可讓您快速還原至備份保留期之內的任何時間點。寫入備份資料時不會影響資料庫服務的效能或中斷服務。您可以將備份保留期指定為 1 至 35 天。您還可以建立[手動資料庫快照](#)，這些快照將持續到您刪除它們為止。手動資料庫快照對於長期備份和封存非常有用。

## 附錄 D：部署新外掛程式和佈景主題

很少網站會保持靜態。在大多數情況下，您將定期新增公開可用的 WordPress 佈景主題和外掛程式，或升級至較新的 WordPress 版本。在其他情況下，您將從頭開發自己的自訂佈景主題和外掛程式。

對您的 WordPress 安裝進行結構性變更的任何時候，都有引入意外問題的一定風險。至少，在套用任何重大變更 (例如安裝新外掛程式) 之前，請先備份您的應用程式碼、組態和資料庫。對於具有業務價值或其他價值的網站，請先在單獨的臨時環境中測試這些變更。有了 AWS，您可以輕鬆複寫生產環境的組態，並以安全的方式執行整個部署程序。完成測試後，您只需卸除測試環境並停止為這些資源付費。本白皮書稍後會討論一些 WordPress 特定的考量。

某些外掛程式會將組態資訊寫入 `wp_options` 資料庫表 (或推出資料庫模式變更)，而另一些外掛程式則會在 WordPress 安裝目錄中建立組態檔案。由於我們已將資料庫和儲存移動到共用平台，不需您的進一步動作，這些變更即會立即可用於您所有執行中的執行個體。

在 WordPress 中部署新佈景主題時，可能需要稍微多一點的動作。如果您只使用 Amazon EFS 來存放所有的 WordPress 安裝檔案，那麼新佈景主題將立即提供給所有執行中的執行個體。但是，如果您要將靜態內容卸載到 Amazon S3，則必須將這些內容的副本處理到正確的儲存貯體位置。W3 總快取之類的外掛程式提供手動啟動該任務的方法。或者，您可以將此步驟自動化，作為建置程序的一部分。

由於佈景主題資產可以在 CloudFront 和瀏覽器上快取，部署變更時您需要一種方法來使舊版本失效。達成這一點的最佳方法是在您的物件中包含某種版本識別碼。此識別碼可以是具有日期-時間戳記或隨機字串的查詢字串。如果您使用 W3 總快取外掛程式，則可以更新附加到媒體檔案 URL 的媒體查詢字串。

## 注意

客戶有責任對本文件中的資訊進行自己的獨立評定。本文件：(a) 僅供參考，(b) 代表目前的AWS產品和實務，這些產品和實務可能隨時變更，恕不另行通知，以及(c) 不會從AWS及其附屬公司、供應商或授權方建立任何承諾或保證。AWS 產品或服務的提供方式是「原樣」，沒有任何類型的保證、陳述或條件，無論是明示還是暗示。AWS 對其客戶的責任和責任由AWS協議控制，本文件不屬於AWS與其客戶之間的任何協議，也未對其進行修改。

© 2023 Amazon Web Services, Inc. 或其附屬公司。保留所有權利。

# AWS 詞彙表

如需最新的 AWS 術語，請參閱AWS 詞彙表 參考 中的[AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。