



AWS 決策指南

AWS Fargate 或 AWS Lambda ?



AWS Fargate 或 AWS Lambda ? : AWS 決策指南

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，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

決策指南	1
簡介	1
差異	3
使用	8
文件歷史紀錄	11
.....	xii

AWS Fargate 或 AWS Lambda ?

了解差異並挑選適合您的差異

用途	探索 AWS Fargate 或 AWS Lambda 是否符合您對無伺服器運算服務的需求。
上次更新	2024 年 11 月 15 日
涵蓋的服務	<ul style="list-style-type: none">• AWS Fargate• AWS Lambda

簡介

在您開始探索是否選擇 AWS Lambda 或 AWS Fargate 作為無伺服器運算服務之前，您可能已考慮過更廣泛的 AWS 運算服務（在[選擇 AWS 運算服務決策指南](#)中涵蓋），並將其縮小到這兩個選擇，因為它們提供：

- 降低營運開銷：Lambda 和 Fargate 都會抽象化伺服器管理，減少修補、維護和容量規劃的需求。
- Pay-per-use定價：您只需支付實際使用的運算資源，可能降低可變工作負載的成本。
- 更快的部署：與佈建和設定 EC2 執行個體相比，通常提供更快的部署時間。
- 內建高可用性：兩種服務都會自動處理基礎設施備援。
- 簡化的合規：減少攻擊面和內建的安全功能可以簡化合規工作。
- 專注於程式碼：開發人員可以更專注於撰寫應用程式程式碼，而不是管理基礎設施。

雖然 Lambda 和 Fargate 都是無伺服器選項，但兩者之間存在顯著差異：

AWS Fargate 是容器的無伺服器運算引擎，主要與 Amazon ECS 搭配使用。它會自動管理您的基礎設施，讓您專注於部署和擴展容器化應用程式。Fargate 非常適合長時間執行的應用程式、微服務或批次處理，其中您需要精細控制資源配置（CPU、記憶體），並希望避免管理基礎伺服器。

AWS Lambda 是一種無伺服器運算服務，可自動執行程式碼以回應事件，並管理基礎運算資源。它最適合事件驅動的應用程式，例如處理上傳至 Amazon S3 的檔案、回應 HTTP 請求或執行排程任務。Lambda 也非常適合串流處理和資料處理應用程式，因為它能夠自動擴展以回應事件並即時處理大

量資料。Lambda 可以處理來自 Amazon Kinesis 或 Amazon DynamoDB 等來源的資料串流，從而實現高效、無伺服器的資料轉換、篩選和分析，而無需管理基礎設施。Lambda 專為短期任務（最多 15 分鐘）而設計，並根據請求數和執行時間計費，使其對偶發工作負載具有成本效益。

如果您的專案涉及事件驅動、短期任務或無法預測的工作負載，AWS Lambda 可能更適合。如果您需要執行具有特定資源需求的容器化應用程式（或需要持久性程序），AWS Fargate 則更為適當。

下表提供這些服務之間一些差異的更詳細介紹，以協助您開始使用。

功能	AWS Fargate	AWS Lambda
執行模型	容器型、無伺服器運算	事件驅動、無伺服器函數
支援的語言	可在容器中執行的任何語言	支援的語言：Node.js、Python、Java、C#、Go、Ruby 和 PowerShell。您也可以 建置自訂執行時間 ，以您選擇的語言實作 AWS Lambda 函數。
使用案例	長時間執行的容器化應用程式	短期、事件驅動的任務
擴展	根據所需的任務計數自動擴展	每個請求的自動擴展
冷啟動	35 秒至 2 分鐘	100 毫秒到 2 秒
執行時間限制	無硬性限制	最多 15 分鐘
記憶體配置	高達 120 GiB	高達 10 GiB
CPU 配置	最多 16 個 vCPU	與記憶體成比例，最多 6 個 vCPU
聯網	在 VPC 中執行，可以使用 ENIs	可以在 AWS 受管 VPC 中執行，或使用 Hyperplane 連接到客戶受管 VPC AWS
狀態管理	只要容器正在執行，Fargate 中的容器就可以在請求之間維持狀態，因此可以處理工作階段、快取資料或保持記憶體內	無狀態設計（狀態必須外部管理，例如 Amazon S3、Amazon DynamoDB、Amazon EFS）

	狀態，而不需要外部儲存。建議對關鍵資料使用外部儲存。	
容器支援	支援容器	有限的容器支援（透過容器映像部署）
協調	與 Amazon ECS 整合	不需要協同運作
定價方式	使用的 vCPU 和記憶體每秒計費	每次調用和持續時間 (GB-秒)
並行限制	根據叢集容量	預設 1000 個並行執行（可增加）
事件驅動調用	需要額外設定	對各種 AWS 事件來源的原生支援
冷啟動緩解措施	使用 Seekable OCI 延遲載入映像可以加速啟動 Fargate 任務	可用的佈建並行
套件大小限制	無特定限制（容器大小受限於設定的暫時性儲存，上限為 200 GiB）	250 MB 解壓縮，包括 layer，容器映像部署為 10GB

Fargate 和 Lambda 之間的差異

探索 Fargate 和 Lambda 在許多關鍵領域之間的差異。

Languages supported

Fargate：AWS Fargate 是容器協同運作服務，表示它支援任何可封裝至 Docker 容器的程式設計語言或執行時間環境。這種靈活性可讓開發人員使用幾乎符合其應用程式需求的任何語言、架構或程式庫。無論您是使用 Python、Java、Node.js、Go、.NET、Ruby、PHP，甚至是自訂語言和環境，只要它們封裝在容器中，Fargate 就可以執行它們。這種廣泛的語言支援使 Fargate 非常適合執行各種應用程式，包括傳統系統、多語言微服務和現代雲端原生應用程式。

相較於 Fargate，**Lambda**：AWS Lambda 提供較有限語言集的原生支援，專為事件驅動函數而設計。截至目前為止，Lambda 正式支援下列語言：

- Node.js
- Python
- Java
- Go
- Ruby
- C#
- PowerShell

Lambda 也支援自訂執行期，可讓您使用自己的語言或執行期環境，但相較於使用原生支援的選項，這需要更多的設定和管理。如果您選擇從容器映像部署 Lambda 函數，您可以使用僅限 AWS 作業系統的基礎映像，並在映像中包含 Rust 執行時間用戶端，在 Rust 中寫入函數。如果您使用的語言沒有 AWS 提供的執行時間界面用戶端，您必須建立自己的。

Event-driven invocation

Lambda 本質上專為事件驅動型運算而設計。Lambda 函數會觸發以回應各種事件，包括資料、使用者動作或排程任務的變更。它與許多無縫整合 AWS 服務，例如 Amazon S3（例如，在上傳檔案時叫用函數）、DynamoDB（例如，在資料更新時觸發）和 API Gateway（例如，處理 HTTP 請求）。Lambda 事件驅動架構非常適合需要立即回應事件的應用程式，而不需要持久性運算資源。

Fargate 不是原生的事件驅動型，但具有一些額外的樣板邏輯，可以與 Amazon SQS 和 Kinesis 等事件來源整合。雖然 Lambda 會為您處理大量此整合邏輯，但您必須使用這些服務 APIs 自行實作此整合。

Runtime/use cases

Fargate 旨在執行容器化應用程式，提供彈性的執行期環境，您可以在其中定義容器的 CPU、記憶體和聯網設定。由於 Fargate 在容器型模型上運作，因此支援長時間執行的程序、持久性服務和具有特定執行期需求的應用程式。Fargate 中的容器可以無限期執行，因為執行時間沒有硬性限制，因此非常適合需要持續啟動和執行的應用程式。

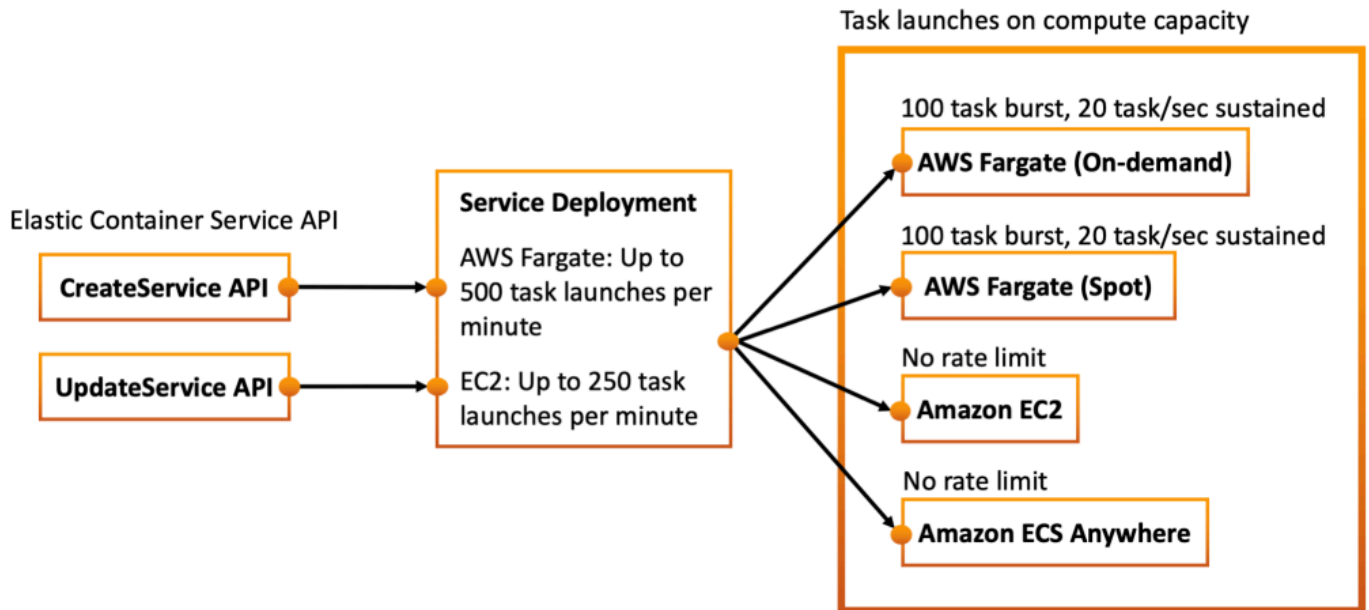
另一方面，Lambda 已針對短期的事件驅動任務進行最佳化。Lambda 函數是在無狀態環境中執行，執行時間上限為 15 分鐘。這使得 Lambda 非常適合檔案處理、即時資料串流和 HTTP 請求處理等案例，其中任務很簡短，不需要長時間執行的程序。

在 Lambda 中，執行期環境較抽象，而且對基礎基礎設施的控制較低。Lambda 的無狀態性質表示每個函數叫用都是獨立的，而且必須在叫用之間保留的任何狀態或資料都必須從外部管理（例如，在資料庫或儲存服務中）。

Scaling

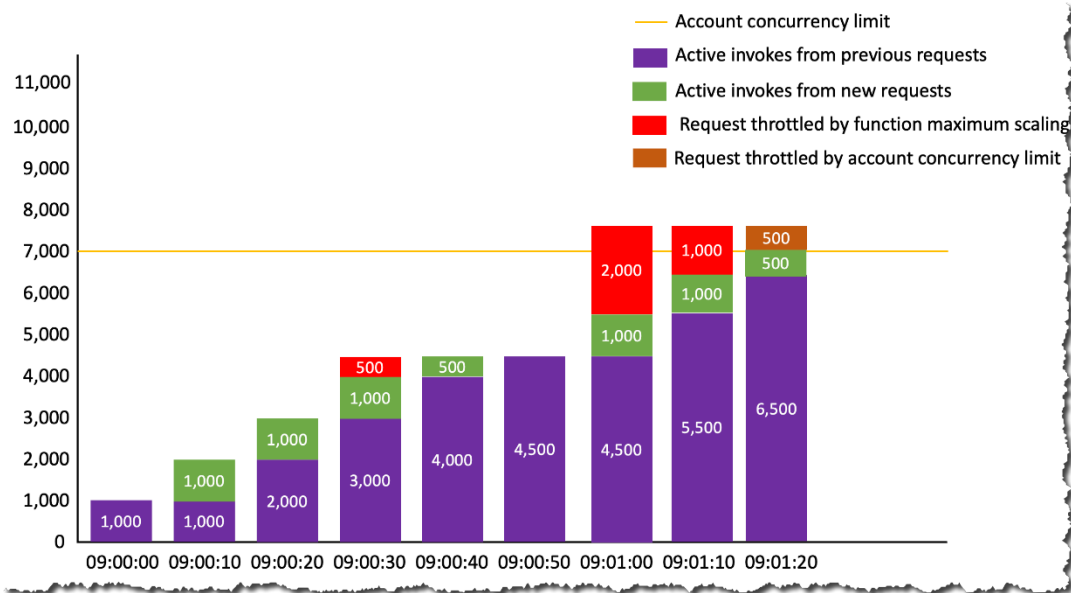
Fargate 會根據容器協同運作服務 (Amazon ECS) 中定義的所需狀態，調整執行中的容器數量來擴展。此擴展可以透過 Amazon EC2 Auto Scaling 手動或自動完成。[此部落格文章](#)提供如何做的更多詳細資訊。

在 Fargate 中，每個容器都會在其隔離的環境中執行，擴展涉及啟動其他容器或根據負載停止容器。Amazon ECS 服務排程器能夠針對 Web 和其他長時間執行的服務，在每個服務一分鐘內啟動多達 500 個任務。



對於 Lambda，並行是 AWS Lambda 函數同時處理的傳輸中請求數量。這與 Fargate 中的並行不同，只要有可用的運算和網路資源，每個 Fargate 任務都可以處理並行請求。Lambda 會針對每個並行請求佈建個別的執行環境執行個體。函數收到更多請求時，Lambda 會自動處理執行環境的擴展數量，直到達到帳戶的並行上限為止。根據預設，Lambda 會為您的帳戶提供中所有函數之間 1,000 個並行執行的總並行限制 AWS 區域，而且您可以視需要請求增加配額。

對於區域中的每個 Lambda 函數，並行擴展速率為每 10 秒 1,000 個執行執行個體，直到帳戶並行上限為止。如[本部落格所述](#)，如果 10 秒期間內的請求數量超過 1,000，額外的請求將會受到限流。下圖示範 Lambda 擴展如何運作，假設帳戶並行 7000。



Cold start and cold-start mitigation

Lambda 可能會遇到冷啟動，這發生在閒置一段時間後調用函數時。在冷啟動期間，Lambda 服務需要初始化新的執行環境，包括載入執行時間、相依性和函數程式碼。此程序可能會引入延遲，特別是初始化時間較長的語言（例如 Java 或 C#）。冷啟動可能會影響應用程式的效能，尤其是需要低延遲回應的應用程式。

若要緩解 Lambda 中的冷啟動，可以採用多種策略：

- 最小化函數大小：減少函數套件的大小及其相依性可以減少初始化所需的時間。
- 增加記憶體配置：較高的記憶體配置會增加 CPU 容量，從而可能縮短初始化時間。
- 保持函數暖機：定期叫用 Lambda 函數（例如，使用 CloudWatch Events）可以保持其作用中狀態，並減少冷啟動的可能性。
- Lambda SnapStart：使用 [Lambda SnapStart](#) for Java 函數來縮短啟動時間。
- 佈建並行：此功能可讓指定數量的函數執行個體保持暖機狀態，並準備好為請求提供服務，從而減少冷啟動延遲。不過，它在您為佈建的執行個體付費時會增加成本，即使它們並未主動處理請求。

Fargate 通常不會像 Lambda 一樣受到冷啟動的影響。啟動 Fargate 任務所需的時間與從[映像登錄檔提取任務中定義的容器](#)映像所需的時間直接相關。Fargate 也支援以 [Seekable OCI \(SOC\)](#) 編製索引的容器映像延遲載入。使用 SOC 延遲載入容器映像可減少在 Fargate 上啟動 Amazon ECS 任務所需的時間。Fargate 會視需要執行保持作用中狀態的容器，這表示它們隨時準備好處

理請求。不過，如果您需要啟動新的容器以回應擴展事件，容器初始化時可能會有一些延遲，但與 Lambda 冷啟動相比，這通常不太重要。

Memory and CPU options

Fargate 為您的容器化應用程式提供記憶體和 CPU 資源的精細控制。當您在 Fargate 中啟動任務時，您可以根據應用程式的需求指定確切的 CPU 和記憶體需求。CPU 和記憶體配置是獨立的，可讓您選擇最適合工作負載的組合。例如，您可以根據組態，選取從 0.25 個 vCPUs 到 16 vCPUs 和記憶體從 0.5 GB 到 120 GB 的 CPU 值。

此彈性非常適合執行需要特定效能特性的應用程式，例如記憶體密集型資料庫或 CPU 限制的運算任務。Fargate 可讓您最佳化資源配置，以有效平衡成本和效能。

在 Lambda 中，記憶體和 CPU 會連結，CPU 會自動根據您選取的記憶體量按比例配置。您可以選擇介於 128 MB 到 10 GB 之間的記憶體配置，以 1 MB 為單位遞增。CPU 會隨記憶體擴展，最多 6 個 vCPU，這表示較高的記憶體設定會產生更多的 CPU 功率，但您無法直接控制 CPU 配置本身。

此模型的設計簡單，可讓開發人員快速調整記憶體設定，而不需要管理 CPU 組態。不過，對於需要 CPU 和記憶體資源之間特定平衡的工作負載，其彈性可能較低。Lambda 的模型適用於您希望根據記憶體需求直接擴展的任務，但可能不適合具有複雜或高度特定資源需求的應用程式。

Networking

當您在 Fargate 中部署任務時，它們會在 Amazon VPC (Amazon Virtual Private Cloud) 中執行，讓您完全控制聯網環境。這包括設定安全群組、網路存取控制清單 ACLs) 和路由表。每個 Fargate 任務都會使用專用的私有 IP 地址取得自己的網路介面，並視需要指派公有 IP 地址。

Fargate 支援進階聯網功能，例如負載平衡 (使用 AWS Elastic Load Balancing)、VPC 對等互連，以及直接存取 VPC AWS 服務中的其他。您也可以使用 AWS PrivateLink 進行受支援的安全私有連線 AWS 服務，而無需周遊網際網路。

根據預設，Lambda 函數會在受管網路環境中執行，而無需直接控制網路介面或 IP 地址。不過，Lambda 可以使用 AWS Hyperplane 連接到客戶管理的 VPC，讓您能夠控制對 VPC 內資源的存取。

當 Lambda 函數連接到客戶受管的 VPC 時，它們會繼承 VPC 的安全群組和子網路組態，允許它們在相同的 VPC 內安全地與其他 AWS 服務 (例如 RDS 資料庫) 互動。

Lambda 服務使用 Network Function Virtualization 平台，將 NAT 功能從 Lambda VPC 提供給客戶 VPC。這會在建立或更新 Lambda 函數時設定所需的彈性網路介面 (ENI)。透過它，您帳戶中的 ENI 能在多個執行環境中共用，這可讓 Lambda 在函數擴展時更有效率地使用有限的網路資源。

由於 ENI 是會耗竭的資源，而且每個區域有 250 個 ENI 的軟性限制，因此如果您要設定 Lambda 函數以進行 VPC 存取，應該監控彈性網路介面的使用情況。相同 AZ 和相同安全群組中的 Lambda 函數可以共用 ENIs。一般而言，如果您在 Lambda 中增大並行限制，應該評估是否需要增加彈性網路介面。如果達到限制，會導致啟用 VPC 之 Lambda 函數的調用受到節制。

Pricing model

Fargate 定價是根據分配給您容器的資源，特別是您為每個任務選取的 vCPU 和記憶體。針對容器使用的 CPU 和記憶體，您每秒需支付 1 分鐘最低費用。這些成本直接與您的應用程式使用的資源相關，這表示無論您的應用程式是否正在積極處理請求，您都需為您佈建的項目付費。Fargate 非常適合您需要特定資源組態的可預測工作負載，並且可以透過調整配置的資源來最佳化成本。此外，相關服務可能會產生額外費用，例如資料傳輸、儲存和聯網（例如 VPC、Elastic Load Balancing）。

Lambda 具有不同的定價結構，其為事件驅動型和 pay-per-execution。系統會根據函數收到的請求數量和每次執行的持續時間向您收費，以毫秒為單位。Lambda 也會考量您配置給函數的記憶體量，並根據使用的記憶體和執行時間來擴展成本。定價模型包含免費方案，每月提供 100 萬個免費請求和 400,000 GB 的運算時間，這使得 Lambda 特別符合低容量、零星工作負載的成本效益。

Lambda 定價模型非常適合流量模式無法預測或爆增的應用程式，因為您只需要支付實際的函數叫用和執行時間，而不需要佈建或支付閒置容量。

使用

現在您已閱讀 AWS Fargate 和 之間的選擇條件 AWS Lambda，您可以選擇符合您需求的服務，並使用下列資訊協助您開始使用每個服務。

AWS Fargate

- 了解如何為 Fargate 啟動類型建立 Amazon ECS Linux 任務

使用 Linux 任務的 Fargate 啟動類型 AWS Fargate，在 上開始使用 Amazon ECS。

[探索指南](#)

- 了解如何為 Fargate 啟動類型建立 Amazon ECS Windows 任務

使用適用於 Windows 任務的 Fargate 啟動類型 AWS Fargate，在 上開始使用 Amazon ECS。

[探索指南](#)

- Fargate 和 Amazon EKS 入門

本指南說明如何開始使用 Amazon EKS 叢集在 上執行 AWS Fargate Pod。

[探索指南](#)

- AWS Fargate 定價

使用本指南了解 vCPU、記憶體、儲存和作業系統組態如何影響 AWS Fargate 定價。

[探索指南](#)

- AWS Fargate 常見問答集

取得有關 AWS Fargate 功能和實作最佳實務的常見問題解答。

[探索指南](#)

AWS Lambda

- 建立無伺服器檔案處理應用程式

設定和使用 Amazon SNS 的 step-by-step 演練。它涵蓋主題，例如建立主題、訂閱主題的端點、發佈訊息，以及設定存取許可。

[探索指南](#)

- 無伺服器開發人員指南

本指南可協助您更深入了解無伺服器應用程式開發，以及各種 AWS 服務 方式如何結合在一起，以建立構成雲端應用程式核心的應用程式模式。

[探索指南](#)

- 無伺服器土地

此網站匯集 AWS 了 Serverless 的最新資訊、部落格、影片、程式碼和學習資源。了解如何使用和建置可在低成本、全受管無伺服器架構上自動擴展的應用程式。

[探索網站](#)

- AWS Lambda 定價

使用本指南根據函數用量和組態來預估費用並最佳化成本。它包含定價計算器，可在單一估算中計算您的 AWS Lambda 和 架構成本。

[探索指南](#)

- AWS Lambda 常見問答集

取得有關 AWS Lambda 功能和實作最佳實務的常見問題解答。

[探索指南](#)

AWS Fargate 或 的文件歷史記錄 AWS Lambda ?

下表說明此決策指南的重要變更。如需有關本指南更新的通知，您可以訂閱 RSS 摘要。

變更	描述	日期
初始版本	決策指南的初始版本。	2024 年 11 月 15 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。