

AWS Academy Cloud Foundations

模組 10：自動擴展和監控



歡迎學習模組 10：自動擴展和監控

模組概覽



主題

- Elastic Load Balancing
- Amazon CloudWatch
- Amazon EC2 Auto Scaling

活動

- Elastic Load Balancing 活動
- Amazon CloudWatch 活動

實驗

- 擴展架構並實現負載均衡



知識測驗

本模組將介紹以下主題：

- Elastic Load Balancing
- Amazon CloudWatch
- Amazon EC2 Auto Scaling

本模組還包括兩個活動。一個活動會要求您說明 Elastic Load Balancing 使用案例。另一個活動會要求您識別 Amazon CloudWatch 示例。

本模組還包括一個動手實踐實驗，您將使用 Amazon EC2 Auto Scaling、Elastic Load Balancing 和 Amazon CloudWatch 以創建動態可擴展的架構。

最後，您需要完成一份知識測驗，以測試您對本模組中涵蓋的關鍵概念的理解程度。

模組目標



完成本模組後，您應能夠：

- 說明如何使用 Elastic Load Balancing 在 Amazon Elastic Compute Cloud (Amazon EC2) 實例之間分配流量
- 瞭解如何使用 Amazon CloudWatch 即時監控 AWS 資源和應用程式
- 解釋 Amazon EC2 Auto Scaling 如何啟動並釋放伺服器以回應工作負載變化
- 執行擴展和負載均衡任務以改善架構

完成本模組後，您應能夠：

- 說明如何使用 Elastic Load Balancing 在 Amazon Elastic Compute Cloud (Amazon EC2) 實例之間分配流量
- 瞭解如何使用 Amazon CloudWatch 即時監控 AWS 資源和應用程式
- 解釋 Amazon EC2 Auto Scaling 如何啟動並釋放伺服器以回應工作負載變化
- 執行擴展和負載均衡任務以改善架構

模組 10：自動擴展和監控

第 1 部分：Elastic Load Balancing

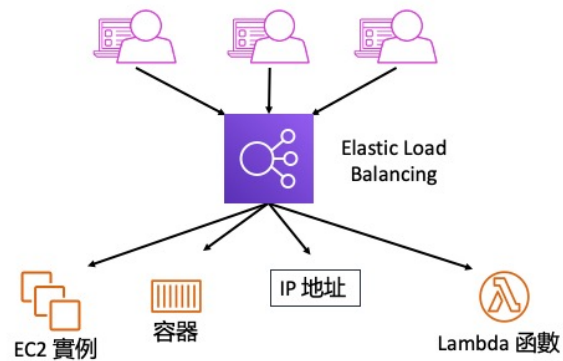


第 1 部分：Elastic Load Balancing

Elastic Load Balancing



- 跨一個或多個可用區中的多個目標分配傳入的應用程式或網路流量。
- 根據傳輸到應用程式的流量在不同時間的變化對負載等化器進行擴展。



現代高流量網站必須回應來自用戶或用戶端的數十萬甚至數百萬併發請求，然後通過快速、可靠的方式返回正確的文本、圖像、視頻或應用程式資料。為了滿足這種大資料量的需求，通常需要加設更多的伺服器。

Elastic Load Balancing 是一項 AWS 服務，可以將傳入的應用程式或網路流量分配給多個目標，例如位於單個或多個可用區中的 Amazon Elastic Compute Cloud (Amazon EC2) 實例、容器、互聯網協定 (IP) 位址和 Lambda 函數等。Elastic Load Balancing 會根據傳輸到應用程式的流量隨時間的變化對負載等化器進行擴展。它可以自動擴展到大多數工作負載。

負載等化器類型



| Application Load Balancer (應用程式負載均衡器) | Network load Balancer (網路負載均衡器) | Classic Load Balancer (上一代) |
|---|--|---|
| <ul style="list-style-type: none">• HTTP 和 HTTPS 流量的負載均衡• 根據請求的內容將流量路由到目標• 提供以交付現代化應用程式架構 (包括微服務和容器) 為目標的高級請求路由• 在應用層 (OSI 模型第 7 層) 運行 | <ul style="list-style-type: none">• TCP、UDP 和 TLS 流量的負載均衡 (需要極高的性能)• 根據 IP 協議數據將流量路由到目標• 每秒可以處理數百萬個請求，同時保持超低延遲• 為處理突發和不穩定的流量模式進行了優化• 在傳輸層 (OSI 模型第 4 層) 運行 | <ul style="list-style-type: none">• HTTP、HTTPS、TCP 和 SSL 流量的負載均衡• 跨多個 EC2 實例進行負載均衡• Operates at both the 在應用層和傳輸層運行 |

Elastic Load Balancing 有三種類型：

- *Application Load Balancer (應用程式負載均衡器)* 在應用程式級別 (開放系統互相連線 (OSI)，模型第 7 層) 運行。它根據請求的內容將流量路由到目標 – Amazon Elastic Compute Cloud (Amazon EC2) 實例、容器、互聯網協定 (IP) 位址和 Lambda 函數。它是超文字傳輸協定 (HTTP) 和安全 HTTP (HTTPS) 流量實現高級負載均衡的理想之選。Application Load Balancer (應用程式負載均衡器) 提供面向現代化應用程式架構交付的高級請求路由，包括微服務和基於容器的應用程式。Application Load Balancer (應用程式負載均衡器) 通過確保始終使用最新的安全通訊端層/傳輸層安全性 (SSL/TLS) 密碼和協定，簡化並提高了應用程式的安全性。
- *Network Load Balancer (網路負載均衡器)* 在網路傳輸級別 (OSI 模型第 4 層) 運行，根據 IP 協定資料將連接路由至目標：EC2 實例、微服務和容器。它是傳輸控制協議 (TCP) 和使用者資料包通訊協定 (UDP) 流量實現負載均衡的理想之選。Network load Balancer (網路負載均衡器) 能夠在保持超低延遲的同時，每秒處理數百萬個請求。Network load Balancer (網路負載均衡器) 針對處理突發和不穩定的網路流量模式進行了優化。
- *Classic Load Balancer* 可以運行在應用程式級別和網路傳輸級別，在多個 EC2 實例之間提供基本的負載均衡。Classic Load Balancer 支援對使用 HTTP、HTTPS、TCP 和 SSL 的應用程式實施負載均衡。Classic Load Balancer 是一種較舊的實施方案。如果可能，AWS 建議您使用專用的 Application Load Balancer (應用程式負載均衡器) 或 Network

load Balancer（網路負載等化器）。

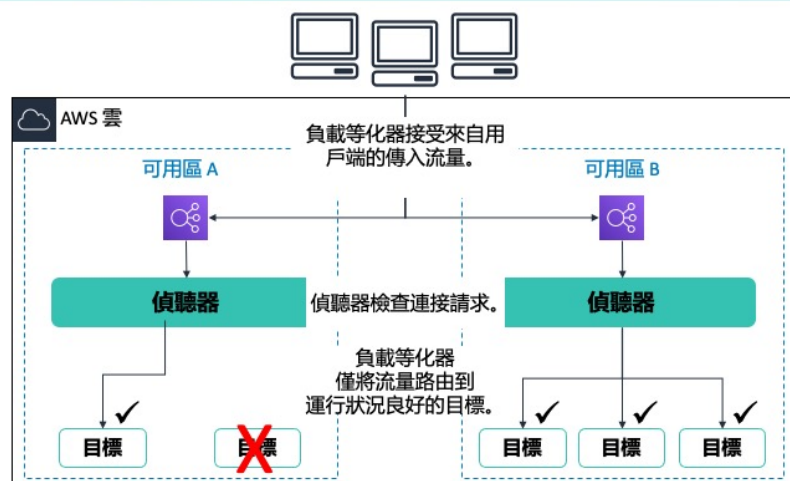
要瞭解有關三種類型的負載等化器之間差異的更多資訊，請參閱 Elastic Load Balancing 功能頁面上的[產品比較](#)。

Elastic Load Balancing 工作原理



- 對於 Application Load Balancer（應用程式負載等化器）和 Network load Balancer（網路負載等化器），您可以在目標群組中註冊目標，並將流量路由到目標群組。
- 對於 Classic Load Balancer，您可以在負載等化器中註冊實例。

負載等化器執行運行狀況檢查來監控已註冊目標的運行狀況。



負載等化器接受來自用戶端的傳入流量並將請求路由到一個或多個可用區中的已註冊目標（例如 EC2 實例）。

您可通過指定一個或多個偵聽器將您的負載等化器配置為接受傳入流量。偵聽器是用於檢查連接請求的進程。它配置了用於從用戶端連接到負載等化器的協定和埠號。同樣，它還配置了用於從負載等化器連接到目標的協定和埠號。

您還可以配置負載等化器以執行運行狀況檢查，這些檢查可用來監控已註冊目標的運行狀況，以便負載等化器僅將請求發送到運行情況良好的實例。當負載等化器檢測到運行狀況不佳的目標時，它會停止將流量路由到該目標。然後，當它檢測到目標運行狀況再次正常時，會繼續將流量路由到該目標。

負載等化器類型的配置方式具有一個關鍵區別。對於 Application Load Balancer（應用程式負載等化器）和 Network load Balancer（網路負載等化器），您可以在目標組中註冊目標，並將流量路由到目標群組。對於 Classic Load Balancer，您可以在負載等化器中註冊實例。

Elastic Load Balancing 使用案例



高度可用且容錯能力強的應用程式



容器化應用程式



彈性和可擴展性



Virtual Private Cloud (VPC)



混合環境



通過 HTTP(S) 調用 Lambda 函數

使用負載均衡器的原因有很多，其中包括：

- **讓您的應用程式實現高可用性和更好的容錯能力** – Elastic Load Balancing 可平衡多個可用區中正常運行的目標之間的流量。如果一個可用區內的一個或多個目標運行不正常，Elastic Load Balancing 會將流量路由至另一個可用區內運行正常的目標。目標恢復正常運行後，Load Balancing 會自動恢復傳輸到這些目標的流量。
- **對容器化應用程式進行自動負載均衡** – 現在，Elastic Load Balancing 的增強容器支援讓您可以在同一 EC2 實例上的多個埠之間進行負載均衡。您還可以利用與 Amazon Elastic Container Service (Amazon ECS) 的深度集成，該服務可提供完全託管的容器功能。您只需註冊具有負載均衡器的服務，Amazon ECS 會以透明方式管理 Docker 容器的註冊和註銷。負載均衡器會自動檢測埠，並自行動態重新配置。
- **自動擴展應用程式** – Elastic Load Balancing 可與 Amazon CloudWatch 和 Amazon EC2 Auto Scaling 結合使用，說明您擴展應用程式以滿足客戶需求。當任何一個 EC2 實例的延遲超過預先配置的閾值時，Amazon CloudWatch 警報可以觸發 EC2 實例佇列的自動擴展。然後，Amazon EC2 Auto Scaling 會預置新實例，您的應用程式將準備好處理下一個客戶請求。負載均衡器將註冊 EC2 實例，並根據需要將流量定向到該實例。
- **在 virtual private cloud (VPC) 中使用 Elastic Load Balancing** – 您可以使用 Elastic Load Balancing 在您的 VPC 中創建一個公共入口點，或在 VPC 中的應用程式各層之間路由請求流量。您可以向負載均衡器分配安全性群組，以控制向一系列允許的來源開放哪些埠。由於 Elastic Load Balancing 與 VPC 一起使用，所有現有的網路存取控制清單（網路 ACL）

和路由表均可繼續強化網路控制。當您在 VPC 中創建負載等化器時，可以指定該負載等化器是公開（預設）還是面向內部。如果選擇面向內部，則無需使用互聯網閘道來連接負載等化器，並且在負載等化器的網域名稱系統 (DNS) 記錄中將使用負載等化器的私有 IP 位址。

- **啟用混合負載平衡** – Elastic Load Balancing 讓您能夠使用同一負載等化器在 AWS 資源和本地資源之間進行負載均衡。例如，如果您必須在 AWS 資源和本地資源之間分配應用程式流量，則可以將所有資源註冊到同一個目標群組內，並將該目標群組與負載等化器關聯起來。或者，您也可以使用兩個負載等化器（其中一個用於 AWS 資源，另一個用於本地資源）在 AWS 資源和本地資源之間進行基於 DNS 的加權負載均衡。如果一個應用程式在 VPC 內，另一個應用程式在本地位置，您也可以使用混合負載均衡分別強化這些應用程式。只需將 VPC 目標放入一個目標群組，將本地目標放入另一個目標群組，然後使用基於內容的路由將流量路由到每個目標群組即可。
- **通過 HTTP(S) 調用 Lambda 函數** – Elastic Load Balancing 支援調用 Lambda 函數來處理 HTTP(S) 請求。這使用戶能夠從任何 HTTP 用戶端（包括 Web 瀏覽器）訪問無伺服器應用程式。您可以將 Lambda 函數註冊為目標，並利用對 Application Load Balancer（應用程式負載等化器）中基於內容的路由規則的支援，將請求路由到不同的 Lambda 函數。您可以將 Application Load Balancer（應用程式負載等化器）用作使用伺服器和無伺服器計算的應用程式的通用 HTTP 終端節點。您可以使用 Lambda 函數構建整個網站，也可以將 EC2 實例、容器、本機伺服器和 Lambda 函數組合在一起構建應用程式。

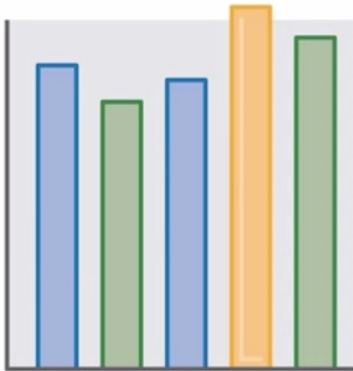
活動：Elastic Load Balancing



| | |
|-----------------------------------|---------------------------------------|
| 您必須支援將流量傳輸到容器化應用程式。 | Application Load Balancer (應用程式負載等化器) |
| 您的 TCP 流量極高且不可預測。 | Network load Balancer (網路負載等化器) |
| 您需要具有多個協定的簡單負載均衡。 | Classic Load Balancer |
| 您需要支援靜態或彈性 IP 位址，或 VPC 外部的 IP 目標。 | Network load Balancer (網路負載等化器) |
| 您需要每秒可處理數百萬次請求的負載等化器，同時保持低延遲。 | Network load Balancer (網路負載等化器) |
| 您必須支持 HTTPS 請求。 | Application Load Balancer (應用程式負載等化器) |

在本活動中，說出用於給定場景的的負載等化器的名稱。

負載等化器監控



- **Amazon CloudWatch 指標** – 用於驗證系統是否按預期運行，並在指標超出可接受範圍時創建警報以啟動操作。
- **訪問日誌** – 捕獲有關發送到負載等化器的請求的詳細資訊。
- **AWS CloudTrail 日誌** – 捕獲 AWS 服務中 API 交互的物件、內容、時間和地點。

您可使用以下功能監控負載等化器、分析流量模式及排查與負載等化器和目標相關的問題。

- **Amazon CloudWatch 指標** – Elastic Load Balancing 會將資料點發佈到 Amazon CloudWatch，以用於您的負載等化器和目標。利用 CloudWatch，您可以按一組有序的時間序列資料（也就是指標），來檢索關於這些資料點的統計資料。您可使用指標來驗證系統是否正常運行。例如，您可以創建 CloudWatch 警報來監控指定的指標，並在指標超出您的可接受範圍時啟動相應操作（例如向某個電子郵寄地址發送通知）。
- **訪問日誌** – 您可以使用訪問日誌來捕獲有關向負載等化器發出的請求的詳細資訊，並將這些詳細資訊作為日誌檔存儲在 Amazon Simple Storage Service (Amazon S3) 中。您可以使用這些訪問日誌分析流量模式，並解決與目標或後端應用程式相關的問題。
- **AWS CloudTrail 日誌** – 您可以使用 AWS CloudTrail 捕獲有關對 Elastic Load Balancing 應用程式設計發展介面 (API) 進行的調用的詳細資訊，並將這些詳細資訊作為日誌檔存儲在 Amazon S3 中。您可以使用這些 CloudTrail 日誌確定調用的發出方、已發出的調用、調用的發出時間和調用發出位置的源 IP 地址等。

第 1 部分要點



- Elastic Load Balancing 跨一個或多個可用區中的多個目標分配傳入的應用程式或網路流量。
- Elastic Load Balancing 支援三種類型的負載等化器：
 - Application Load Balancer (應用程式負載等化器)
 - Network load Balancer (網路負載等化器)
 - Classic Load Balancer
- ELB 提供實例運行狀況檢查、安全性和監控。

本模組中此部分要點包括：

- Elastic Load Balancing 可在一個或多個可用區的多個目標（如 Amazon EC2 實例、容器、IP 位址和 Lambda 函數）之間分配傳入的應用程式或網路流量。
- Elastic Load Balancing 支援三種類型的負載等化器：
 - Application Load Balancer (應用程式負載等化器)
 - Network load Balancer (網路負載等化器)
 - Classic Load Balancer
- Elastic Load Balancing 提供了幾個監控工具，用於連續監控和記錄審計和分析。

模組 10：自動擴展和監控

第 2 部分：Amazon CloudWatch



第 2 部分：Amazon CloudWatch

要高效使用 AWS，您需要深入瞭解您的 AWS 資源：

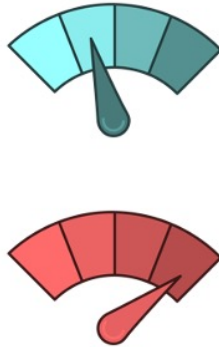
- 如何知道何時應當**啟動更多 Amazon EC2 實例**？
- **應用程式的性能或可用性**是否受到容量不足的影響？
- **實際使用的基礎設施**有多少？

要高效使用 AWS，您需要深入瞭解您的 AWS 資源。

例如，您可能需要瞭解：

- 何時應當啟動更多 Amazon EC2 實例？
- 應用程式的性能或可用性是否受到容量不足的影響？
- 實際使用的基礎設施有多少？

如何獲取相關資訊？



- 監控 –
 - AWS 資源
 - 在 AWS 上運行的應用程式
- 收集和跟蹤 –
 - 標準指標
 - 自訂指標
- 警報 –
 - 將通知發送到 Amazon SNS 主題
 - 執行 Amazon EC2 Auto Scaling 或 Amazon EC2 操作
- 事件 –
 - 定義規則以匹配 AWS 環境中的更改，並將這些事件路由到一個或多個目標函數或流以進行處理

您可以使用 Amazon CloudWatch 獲取相關資訊。

Amazon CloudWatch 是一種面向開發運營工程師、開發人員、網站可靠性工程師 (SRE) 和 IT 經理的監控和可觀測性服務。CloudWatch 可以即時監控您的 AWS 資源（和在 AWS 上運行的應用程式）。您可以使用 CloudWatch 收集和跟蹤指標，這些指標是您用於衡量資源和應用程式的變數。

您可以創建警報來監控您帳戶中的任何 Amazon CloudWatch 指標，並使用該警報自動將通知發送到 Amazon Simple Notification Service (Amazon SNS) 主題，或執行 Amazon EC2 Auto Scaling 或 Amazon EC2 操作。例如，您可以創建警報來監控 EC2 實例的 CPU 利用率、Elastic Load Balancing 請求延遲、Amazon DynamoDB 表輸送量、Amazon Simple Queue Service (Amazon SQS) 佇列長度，甚至是 AWS 帳單費用。您還可以為針對自訂應用程式或基礎設施的自訂指標創建警報。

您還可以使用 Amazon CloudWatch Events 來定義與傳入事件（或您 AWS 環境中的更改）匹配的規則，並將這些規則路由到目標進行處理。目標包括 Amazon EC2 實例、AWS Lambda 函數、Kinesis 流、Amazon ECS 任務、Step Functions 狀態機、Amazon SNS 主題、Amazon SQS 佇列和內置目標。CloudWatch Events 會在這些操作更改發生時感知到它們。CloudWatch Events 會回應這些操作更改，並根據需要採取糾正措施，方法是發送消息以回應環境、啟動函數、進行更改以及捕獲狀態資訊。

您可通過使用 CloudWatch 全面地瞭解資源使用率、應用程式性能和運行狀況。無需預先承諾，沒有最低消費限制；您只需為所用內容付費，費用會在每月月末按實際用量收取。

CloudWatch 警報



- 基於以下內容創建警報 –

- 靜態閾值
- 異常檢測
- 指標數學運算式

- 指定 –

- 命名空間
- 指標
- 統計數據
- 時間段
- 條件
- 其他配置
- 操作

The screenshot shows the AWS CloudWatch Alarm configuration interface. It includes a search bar for 'Statistic' with 'Average' selected, a 'Period' dropdown set to '5 minutes', and a 'Conditions' section. Under 'Threshold type', 'Static' is selected with the subtext 'Use a value as a threshold'. Below this, the condition is defined as 'Whenever CPUUtilization is... Define the alarm condition' with 'Greater' selected and '> threshold'. The 'then...' section shows 'Define the threshold value' set to '100'. At the bottom, there is a link for 'Additional configuration'.

您可以創建一個 CloudWatch 警報來監控單個 CloudWatch 指標，或基於 CloudWatch 指標監控數學運算式的結果。您可以根據靜態閾值、異常檢測或指標數學運算式來創建 CloudWatch 警報。

根據靜態閾值創建警報時，請為該警報選擇要監控的 CloudWatch 指標以及該指標的閾值。在指標超過閾值達到指定數量的評估期時，警報將變為“ALARM”（警報）狀態。

對於基於靜態閾值的警報，您必須指定：

- **命名空間** – 命名空間包含您所需的 CloudWatch 指標，例如 *AWS/EC2*。
- **指標** – 指標是您要衡量的變數，例如 *CPU 利用率*。
- **統計資料** – 統計資料可以是平均值，也可以是總計值、最小值或最大值、樣本計數、預定義的百分位數，或者自訂百分位數。
- **時間段** – 時間段是警報的評估期。評估警報時，每個時間段會聚合為一個數據點。
- **條件** – 指定靜態閾值的條件時，您可以指定該指標在何時大於、大於或等於、小於或等於或者小於閾值，您還可以指定閾值。
- **其他配置資訊** – 這包括評估期內觸發警報必須超出的資料點數量下限，以及 CloudWatch 在評估警報時處理丟失資料的方式。
- **操作** – 您可以選擇向 Amazon SNS 主題發送通知，或者執行 Amazon EC2 Auto Scaling 操作或 Amazon EC2 操作。

有關創建 CloudWatch 警報的更多資訊，請參閱 AWS 文檔中[使用警報](#)下的主題。

活動：Amazon CloudWatch



Amazon EC2

如果平均 CPU 利用率持續 5 分鐘高於 60%...

正確！



Amazon RDS

如果併發連接的數量超過 10 的情況持續了 1 分鐘...

正確！



Amazon S3

如果存儲桶的最大大小（以位元組為單位）約為 3 的情況持續了 1 天...

不正確。大約不是閾值選項。您必須指定 >、>=、<= 或 < 的閾值。



Elastic Load Balancing

如果 10 分鐘內正常運行的主機數量少於 5 個...

正確！



Amazon Elastic Block Store

如果 10 秒內讀取操作的數量大於 1000...

不正確。您必須指定一個統計資料（例如，平均量）。

在本活動中，看看您是否可以識別正確的 CloudWatch 警報。對於不正確的 CloudWatch 警報，看看您是否可以識別錯誤。

第 2 部分要點



- Amazon CloudWatch 可幫助您即時監控 AWS 資源以及在 AWS 上運行的應用程式。
- CloudWatch 可讓您 –
 - 收集和跟蹤標準以及自訂指標。
 - 設置警報以自動將通知發送到 SNS 主題，或執行 Amazon EC2 Auto Scaling 或 Amazon EC2 操作。
 - 定義與您的 AWS 環境中的更改匹配的規則，並將這些事件路由到目標進行處理。

本模組中此部分要點包括：

- Amazon CloudWatch 可幫助您即時監控 AWS 資源以及在 AWS 上運行的應用程式。
- CloudWatch 可讓您 –
 - 收集和跟蹤標準以及自訂指標。
 - 設置警報以將通知自動發送到 SNS 主題，或根據指標或運算式在多個時間段內相對於某閾值的值執行 Amazon EC2 Auto Scaling 或 Amazon EC2 操作。
 - 定義與您的 AWS 環境中的更改匹配的規則，並將這些事件路由到目標進行處理。

模組 10：自動擴展和監控

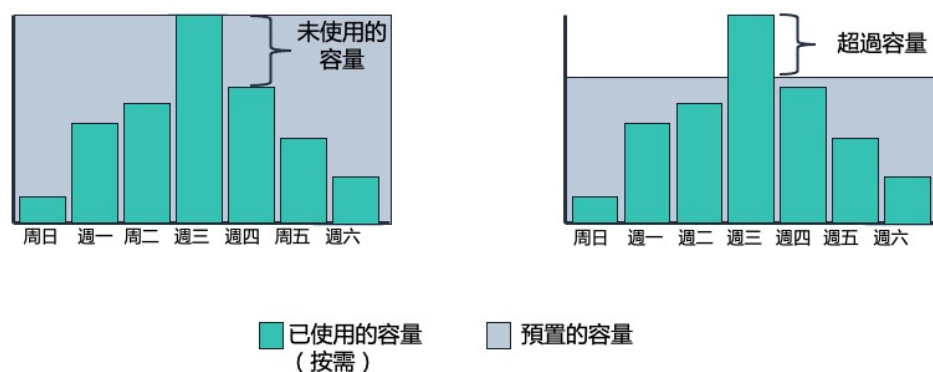
第 3 部分：Amazon EC2 Auto Scaling



第 3 部分：Amazon EC2 Auto Scaling

在 AWS 上運行應用程式時，您需要確保您的架構能夠擴展以處理不斷變化的需求。在本部分，您將學習如何使用 Amazon EC2 Auto Scaling 自動擴展 EC2 實例。

為什麼擴展很重要？

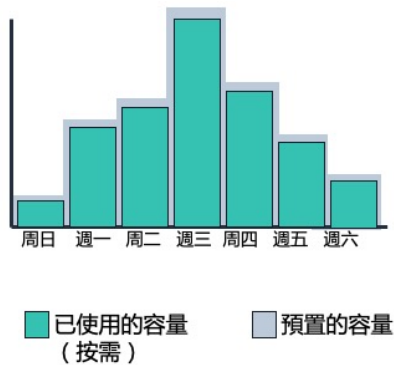


擴展是指增加或減少應用程式計算容量的能力。要瞭解擴展的重要性，請您設想有這樣一個工作負載，其資源需求不斷變化。在本示例中，星期三需要的資源容量最多，而星期日常需要的資源容量最少。

一個選擇是分配足夠的容量，以便始終滿足最高需求 – 在本例中為星期三。不過，這種情況意味著您運行的資源在一周的大部分時間內都得不到充分利用。這種選擇無法助您實現成本優化。

另一種選擇是通過分配較少的容量來降低成本。這種情況意味著您會在某些天內容量不足。如果您不能解決容量問題，您的應用程式可能無法發揮出應有的性能，甚至影響使用者正常使用。

Amazon EC2 Auto Scaling



- 說明您保持應用程式可用性
- 使您能夠根據定義的條件自動添加或刪除 EC2 實例
- 檢測受損的 EC2 實例和運行狀況不佳的應用程式，並在無需您干預的情況下替換實例
- 提供多種擴展選項 – 手動、計畫、動態或按需和預測性

在雲中，由於計算能力是一種程式設計資源，我們可以採取更靈活的方法來進行擴展。Amazon EC2 Auto Scaling 是一項 AWS 服務，通過此服務，您可以保持應用程式的可用性，並且根據您定義的條件自動添加或刪除 EC2 實例。您可以使用 EC2 Auto Scaling 的佇列管理功能維護佇列的運行狀況和可用性。

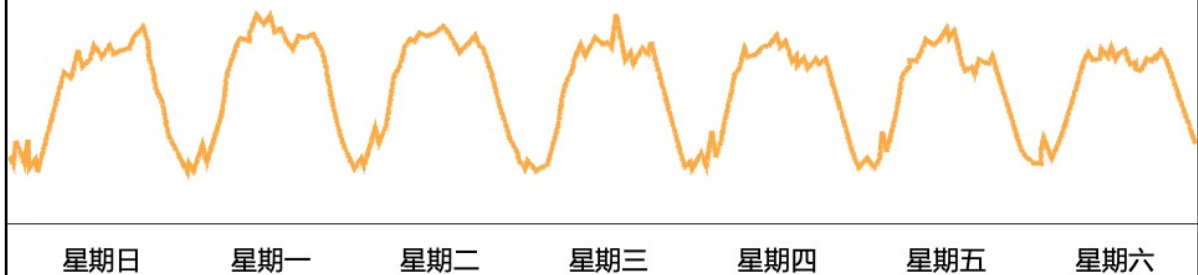
Amazon EC2 Auto Scaling 提供了多種擴展調整方式來滿足您的應用程式需求。您可以根據計畫手動添加或刪除 EC2 實例以回應不斷變化的需求，或者結合使用 AWS Auto Scaling 進行預測性擴展。動態擴展和預測性擴展可結合使用，以實現更快的擴展。

要瞭解有關 Amazon EC2 Auto Scaling 的更多資訊，請參閱 [Amazon EC2 Auto Scaling](#) 產品頁面。

Amazon.com 典型的每週流量



預置的容量

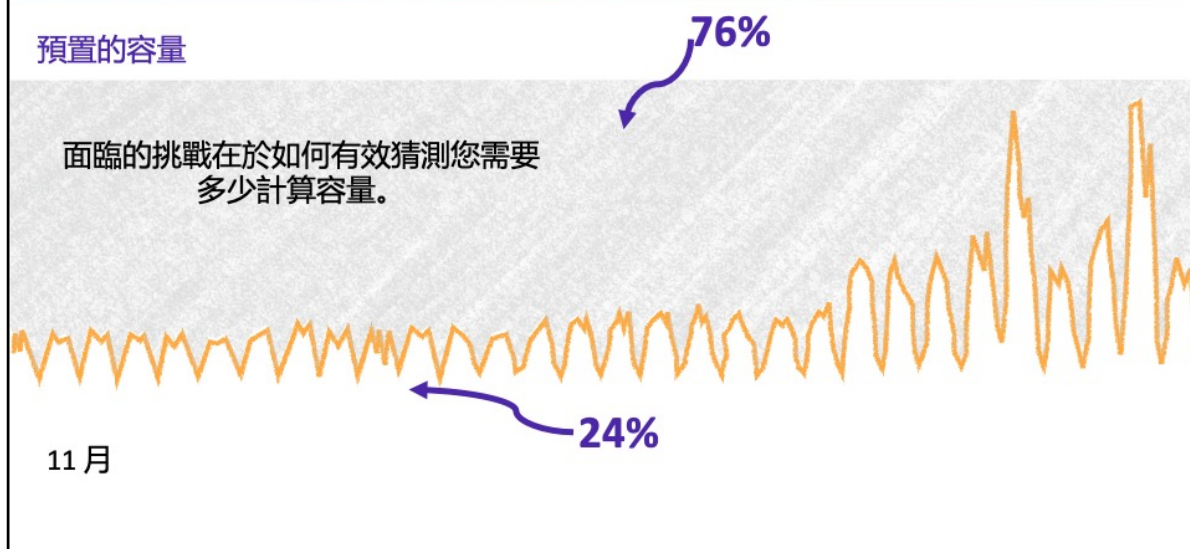


自動擴展對於可預測的工作負載很有用，例如零售公司 Amazon.com 的每週流量。

Amazon.com 11 月的流量

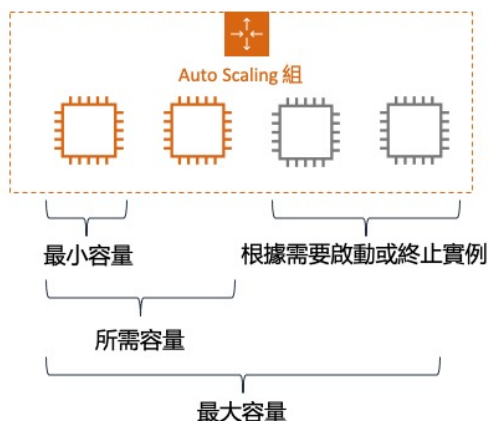
aws academy

預置的容量



自動擴展還適用於動態按需擴展。在 11 月底的“黑色星期五”和“網購星期一”，Amazon.com 會經歷季節性的流量高峰，這幾天是美國零售商的銷售旺季。如果 Amazon 預置固定容量以滿足最高使用量需求，那麼一年中的大部分時間將有 76% 的資源處於空閒狀態。容量擴展是支援不斷變化的服務需求的必要措施。如果不進行擴展，伺服器可能會因飽和而崩潰，從而使企業失去客戶信心。

Auto Scaling 組



Auto Scaling 組是 EC2 實例集合，這些集合被視為邏輯組以用於自動擴展和管理用途。

[Auto Scaling 組](#)是 Amazon EC2 實例集合，這些集合被視為邏輯組以用於自動擴展和管理用途。Auto Scaling 組的大小取決於您配置為**所需容量**的實例數量。您可以通過手動方式或使用自動擴展調整其大小以滿足需求。

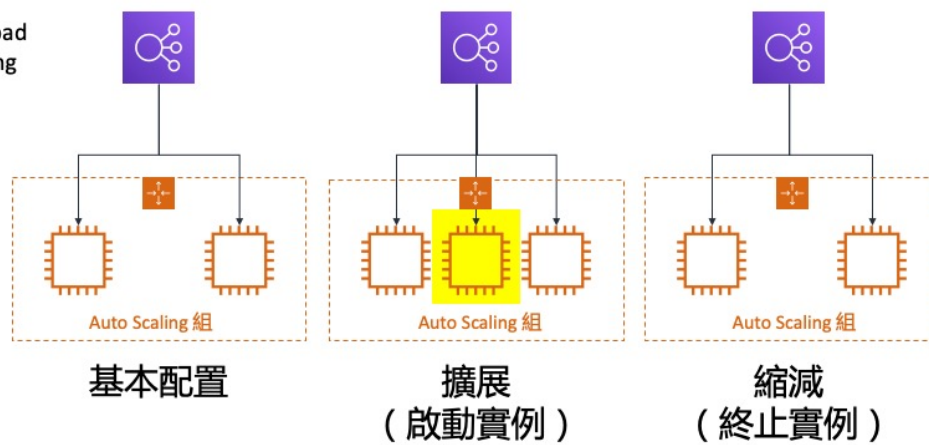
您可以指定每個 Auto Scaling 組內的最小實例數量，Amazon EC2 Auto Scaling 會確保您的組始終不低於這個數量。您可以指定每個 Auto Scaling 組內的最大實例數量，Amazon EC2 Auto Scaling 會確保您的組始終不高於這個數量。如果您在創建組時或之後的任何時間指定了**所需容量**，Amazon EC2 Auto Scaling 會調整組的大小，使組中具有指定的實例數。如果您指定了擴展策略，Amazon EC2 Auto Scaling 可以在應用程式需求增加或降低時啟動或終止實例。

例如，此 Auto Scaling 組的最小大小為 1 個實例，所需容量為 2 個實例，最大大小為 4 個實例。您制定的擴展策略是按照您指定的條件，在最大和最小實例數範圍內調整實例的數量。

擴展與縮減

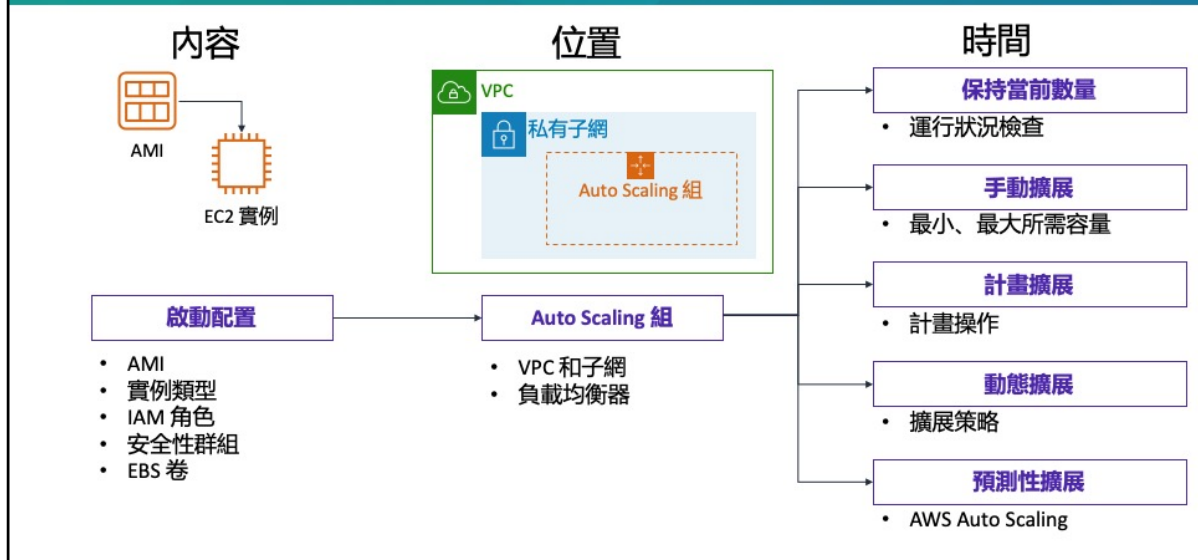
aws academy

Elastic Load
Balancing



使用 Amazon EC2 Auto Scaling 時，啟動實例稱為擴展，終止實例稱為縮減。

Amazon EC2 Auto Scaling 工作原理



為了啟動 EC2 實例，Auto Scaling 組使用**啟動配置**，即實例配置範本。您可以將啟動配置視為要擴展的**內容**。創建啟動配置時，您需要指定實例資訊。您指定的資訊包括 Amazon 系統映像 (AMI) 的 ID、實例類型、AWS Identity and Access Management (IAM) 角色、額外的存儲、一個或多個安全性群組以及任何 Amazon Elastic Block Store (Amazon EBS) 卷。

您定義 Auto Scaling 組的最大和最小實例數以及所需容量。然後，將其啟動到 VPC 內的子網中（您可以將其視為要擴展的**位置**）。Amazon EC2 Auto Scaling 與 Elastic Load Balancing 集成，讓您能夠將一個或多個負載均衡器附加到現有的 Auto Scaling 組。在您連接負載均衡器後，它會自動在組中註冊實例，並在各實例之間分配傳入流量。

最後，您需要指定擴展事件發生的**時間**。您有多個擴展選項：

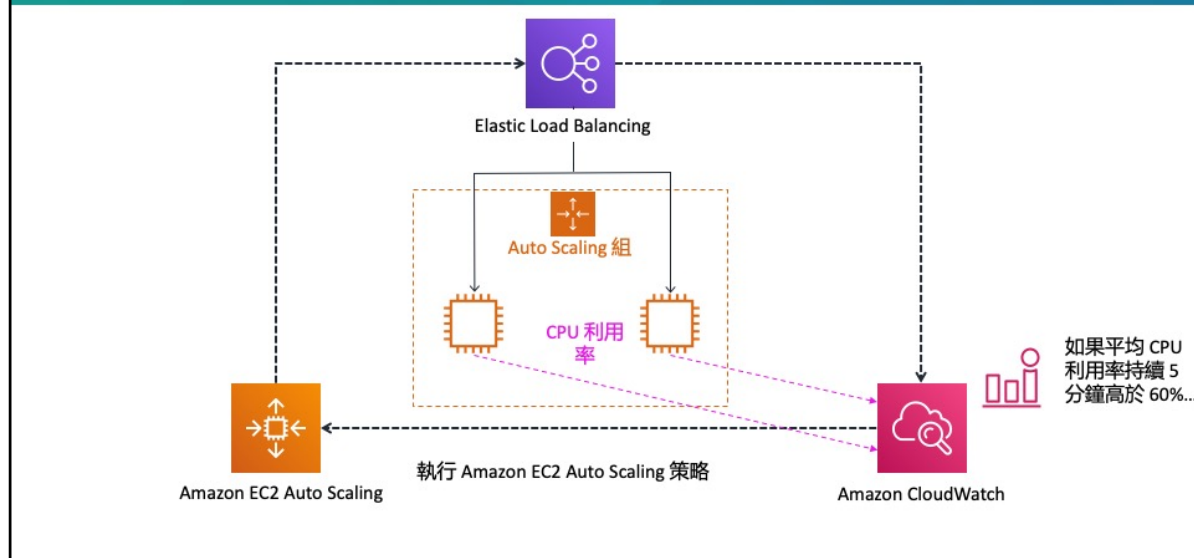
- **始終保持當前實例等級** – 您可以配置 Auto Scaling 組，使其始終保持指定的運行實例數。為了保持當前實例等級，Amazon EC2 Auto Scaling 將對 Auto Scaling 組內運行的實例執行定期運行狀況檢查。在找到未正常運行的實例時，Amazon EC2 Auto Scaling 將終止該實例並啟動新實例。
- **手動擴展** – 使用手動擴展，您只需指定 Auto Scaling 組的最大容量、最小容量或所需容量的變化。
- **計畫擴展** – 使用計畫擴展，擴展操作將按照日期和時間的函數自動執行。考慮到您確切地知道應在何時增加或減少組中的實例數量，這對於可預測的工作負載會非常有用。例如，假設每週的 Web 應用程式流量在星期三開始增加，星期四仍然保持較高水準，然後在星期五開始減少。您可以根據 Web 應用程式的可預測流量模式來計

畫擴展操作。要實施計畫擴展，您可以創建[計畫操作](#)。

- [動態按需擴展](#) – 一種更高級的資源擴展方法，可讓您定義參數以控制擴展過程。例如，您有一個當前在兩個實例上運行的 Web 應用程式，並希望在應用程式負載變化時將 Auto Scaling 組的 CPU 使用率保持在接近 50%。在根據條件變化進行擴展，但卻不知道條件何時改變時，可以使用這種方法。動態擴展為您提供了額外容量以處理流量高峰，而無需維護過多的空閒資源。您可以將 Auto Scaling 組配置為自動擴展來滿足這一需求。[擴展策略類型](#)決定了如何執行擴展操作。您可以將 Amazon EC2 Auto Scaling 與 Amazon CloudWatch 配合使用觸發擴展策略以回應警報。
- [預測性擴展](#) – 您可以將 Amazon EC2 Auto Scaling 與 AWS Auto Scaling 配合使用來實施預測性擴展，使用此方式時，容量可根據預測的需求進行擴展。預測性擴展使用從您的 EC2 的實際使用情況中收集的資料，而這些資料通過我們自己的觀察得出的數十億個資料點得到進一步豐富。然後，AWS 使用訓練有素的機器學習模型來預測您的預期流量（和 EC2 使用情況），包括每日和每週模式。該模型至少需要 1 天的歷史資料才能開始進行預測。每 24 小時重新評估一次，以創建接下來 48 小時的預測。預測過程中會產生可以驅動一組或多組自動擴展的 EC2 實例的擴展計畫。

要瞭解有關這些選項的更多資訊，請參閱 AWS 文檔中的[擴展 Auto Scaling 組的大小](#)。

實施動態擴展



實施動態擴展的一種常見配置是根據從 EC2 實例或負載等化器收集的性能資訊創建 CloudWatch 警報。當達到性能閾值時，CloudWatch 警報會觸發一次自動擴展事件來擴展或縮減 Auto Scaling 組中的 EC2 實例。

要瞭解其工作原理，請思考這樣一個示例：

- 您創建一個 Amazon CloudWatch 警報用於監控整個 EC2 實例佇列中的 CPU 利用率，並在整個佇列中的平均 CPU 利用率超過 60% 並持續 5 分鐘以上時執行自動擴展策略。
- Amazon EC2 Auto Scaling 會根據您創建的啟動配置將新的 EC2 實例產生實體到您的 Auto Scaling 組中。
- 添加新實例後，Amazon EC2 Auto Scaling 會調用 Elastic Load Balancing，在該 Auto Scaling 組中註冊新的 EC2 實例。
- 然後，Elastic Load Balancing 執行所需的運行狀況檢查，並開始將流量分配給該實例。Elastic Load Balancing 在 EC2 實例之間路由流量，並將指標饋送到 Amazon CloudWatch。

Amazon CloudWatch、Amazon EC2 Auto Scaling 和 Elastic Load Balancing 單獨使用也可發揮強大作用。但是，它們共同運行可變得更加強大，並增強了對應用程式處理客戶需求的方式的控制，同時提升了靈活性。



AWS Auto Scaling

- 監控您的應用程式並自動調整容量，從而以盡可能低的成本維持穩定、可預測的性能
- 提供簡單且功能強大的使用者介面，可讓您構建資源擴展計畫，其中包括：
 - Amazon EC2 實例和 Spot 佇列
 - Amazon Elastic Container Service (Amazon ECS) 任務
 - Amazon DynamoDB 表和索引
 - Amazon Aurora 副本

到目前為止，您已經瞭解了如何使用 Amazon EC2 Auto Scaling 擴展 EC2 實例。您還瞭解了可以將 Amazon EC2 Auto Scaling 與 AWS Auto Scaling 配合使用來執行預測性擴展。

AWS Auto Scaling 是一項獨立的服務，可用於監控您的應用程式。它會自動調整容量，從而以盡可能低的成本維持穩定、可預測的性能。該服務提供簡單且功能強大的使用者介面，可讓您構建資源擴展計畫，其中包括：

- Amazon EC2 實例和 Spot 佇列
- Amazon Elastic Container Service (Amazon ECS) 任務
- Amazon DynamoDB 表和索引
- Amazon Aurora 副本

如果您已經在使用 Amazon EC2 Auto Scaling 來動態擴展 EC2 實例，那麼現在可以將其與 AWS Auto Scaling 結合使用，為其他 AWS 服務擴展其他資源。

要瞭解有關 AWS Auto Scaling 的更多資訊，請參閱 [AWS Auto Scaling](#)。

第 3 部分要點



- 擴展讓您能夠快速回應資源需求的變化。
- Amazon EC2 Auto Scaling 通過自動添加或刪除 EC2 實例來維護應用程式可用性。
- Auto Scaling 組是 EC2 實例的集合。
- 啟動配置是實例配置範本。
- 動態擴展使用 Amazon EC2 Auto Scaling、CloudWatch 和 Elastic Load Balancing。
- AWS Auto Scaling 是獨立於 Amazon EC2 Auto Scaling 的一項服務。

本模組中此部分要點包括：

- 擴展讓您能夠快速回應資源需求的變化。
- Amazon EC2 Auto Scaling 說明您保持應用程式可用性，並讓您可以根據工作負載自動添加或刪除 EC2 實例。
- Auto Scaling 組是 EC2 實例的集合。
- 啟動配置是實例配置範本。
- 您可以使用 Amazon EC2 Auto Scaling、Amazon CloudWatch 和 Elastic Load Balancing 實施動態擴展。

AWS Auto Scaling 是一項獨立的服務，用於監控您的應用程式，並且會自動調整以下資源的容量：

- Amazon EC2 實例和 Spot 佇列
- Amazon ECS 任務
- Amazon DynamoDB 表和索引
- Amazon Aurora 副本

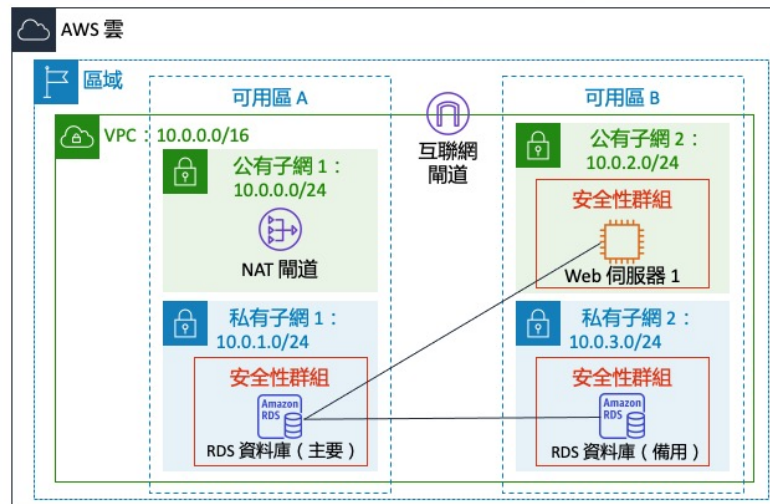
實驗 6： 擴展架構並實現負 載均衡

aws academy



現在，您將完成實驗 6：擴展架構並實現負載均衡。

實驗 6：場景



在本實驗中，您將使用 Elastic Load Balancing 和 Amazon EC2 Auto Scaling，以實現負載均衡並擴展您的基礎設施。先從給定的基礎設施入手。

實驗 6：任務

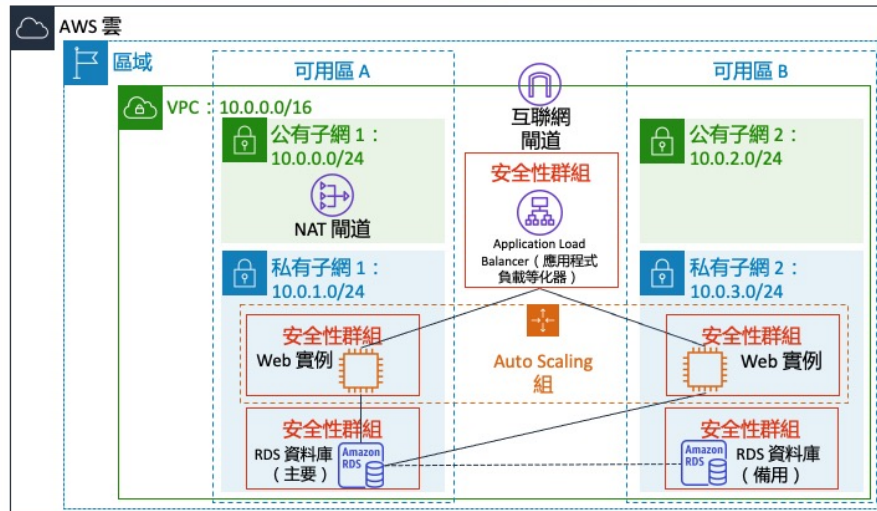


- 從正在運行的實例創建 Amazon 系統映射 (AMI)。
- 創建 Application Load Balancer (應用程式負載等化器) 。
- 創建啟動配置和 Auto Scaling 組。
- 自動擴展私有子網內的新實例。
- 創建 Amazon CloudWatch 警報並監控基礎設施性能。

在本實驗中，您將完成以下任務：

- 從正在運行的實例創建 Amazon 系統映射 (AMI)。
- 創建 Application Load Balancer (應用程式負載等化器) 。
- 創建啟動配置和 Auto Scaling 組。
- 自動擴展私有子網內的新實例。
- 創建 Amazon CloudWatch 警報並監控基礎設施性能。

實驗 6：最終成果



該圖總結了您完成實驗後將會構建的內容。



大約 30 分鐘



開始實驗 6：擴展架構 並實現負載均衡

現在可以開始實驗了。完成本實驗大約需要 30 分鐘。

實驗總結： 要點



在本實驗中，您完成了以下工作：

- 從正在運行的實例創建 Amazon 系統映射 (AMI)。
- 創建負載等化器。
- 創建啟動配置和 Auto Scaling 組。
- 自動擴展私有子網內的新實例。
- 創建 Amazon CloudWatch 警報並監控基礎設施性能。

模組 10：自動擴展和監控

模組總結



現在來回顧下本模組，並對知識測驗和對實踐認證考試問題的討論進行總結。

模組總結



總體來說，您在本模組中學習了如何：

- 說明如何使用 Elastic Load Balancing 在 Amazon Elastic Compute Cloud (Amazon EC2) 實例之間分配流量。
- 明確如何使用 Amazon CloudWatch 即時監控 AWS 資源和應用程式。
- 解釋 Amazon EC2 Auto Scaling 如何啟動和釋放伺服器來回應工作負載變化。
- 執行擴展和負載均衡任務以改善架構。

總體來說，您在本模組中學習了如何：

- 說明如何使用 Elastic Load Balancing 在 Amazon Elastic Compute Cloud (Amazon EC2) 實例之間分配流量。
- 瞭解如何使用 Amazon CloudWatch 即時監控 AWS 資源和應用程式
- 解釋 Amazon EC2 Auto Scaling 如何啟動和釋放伺服器來回應工作負載變化。
- 執行擴展和負載均衡任務以改善架構。

完成知識測驗



完成本模組的知識測驗。

您會使用哪項服務根據 Amazon CloudWatch 警報發送提醒？

- A. Amazon Simple Notification Service
- B. AWS CloudTrail
- C. AWS Trusted Advisor
- D. Amazon Route 53

查看答案選項，並根據之前突出顯示的關鍵字排除錯誤選項。

謝謝

© 2019 Amazon Web Services, Inc. 或其附屬公司。保留所有權利。未經 Amazon Web Services, Inc. 事先書面許可，不得複製或轉載本文的部分或全部內容。禁止因商業目的複製、出售或出售本文。如有對本課程的糾正或回饋意見，請發送電子郵件至：aws-course-feedback@amazon.com。如有其他任何問題，請與我們聯繫：<https://aws.amazon.com/contact-us/aws-training/>。所有商標均為各自所有者的財產。



感謝您的參與！