aws

最佳實務指南

Amazon Elastic Container Service

Amazon Elastic Container Service: 最佳實務指南

Table of Contents

Introduction

Amazon Elastic Container Service (Amazon ECS) 是具高可擴展性且快速的容器管理服務，可以在叢集上輕鬆執行、停止和管理容器。本指南涵蓋許多最重要的操作最佳實務，同時也解釋基於 Amazon ECS 應用程式如何運作的核心主題。目標是提供具體且可行的方法來操作和疑難排解 Amazon ECS 應用程式。

本指南將定期修訂以納入新的 Amazon ECS 最佳實務。如果您對本指南中的任何內容有任何疑問或意見，請在 GitHub 存放庫中提出問題。如需詳細資訊，請參閱「」亞馬遜 ECS 最佳實務指南(在 GitHub 上)。

- 最佳實務-網路
- 最佳實務-自動擴充和容量管理
- 最佳做法-永久性儲存
- 最佳實務-安全

最佳實務-網路

現代應用程序通常是由相互通信的多個分佈式組件構建的。例如，行動裝置或 Web 應用程式可能會與 API 端點進行通訊，而 API 可能會由透過網際網路通訊的多個微服務提供支援。

本指南提供建置網路的最佳實務，讓您的應用程式元件可以安全且可擴充的方式彼此通訊。

主題

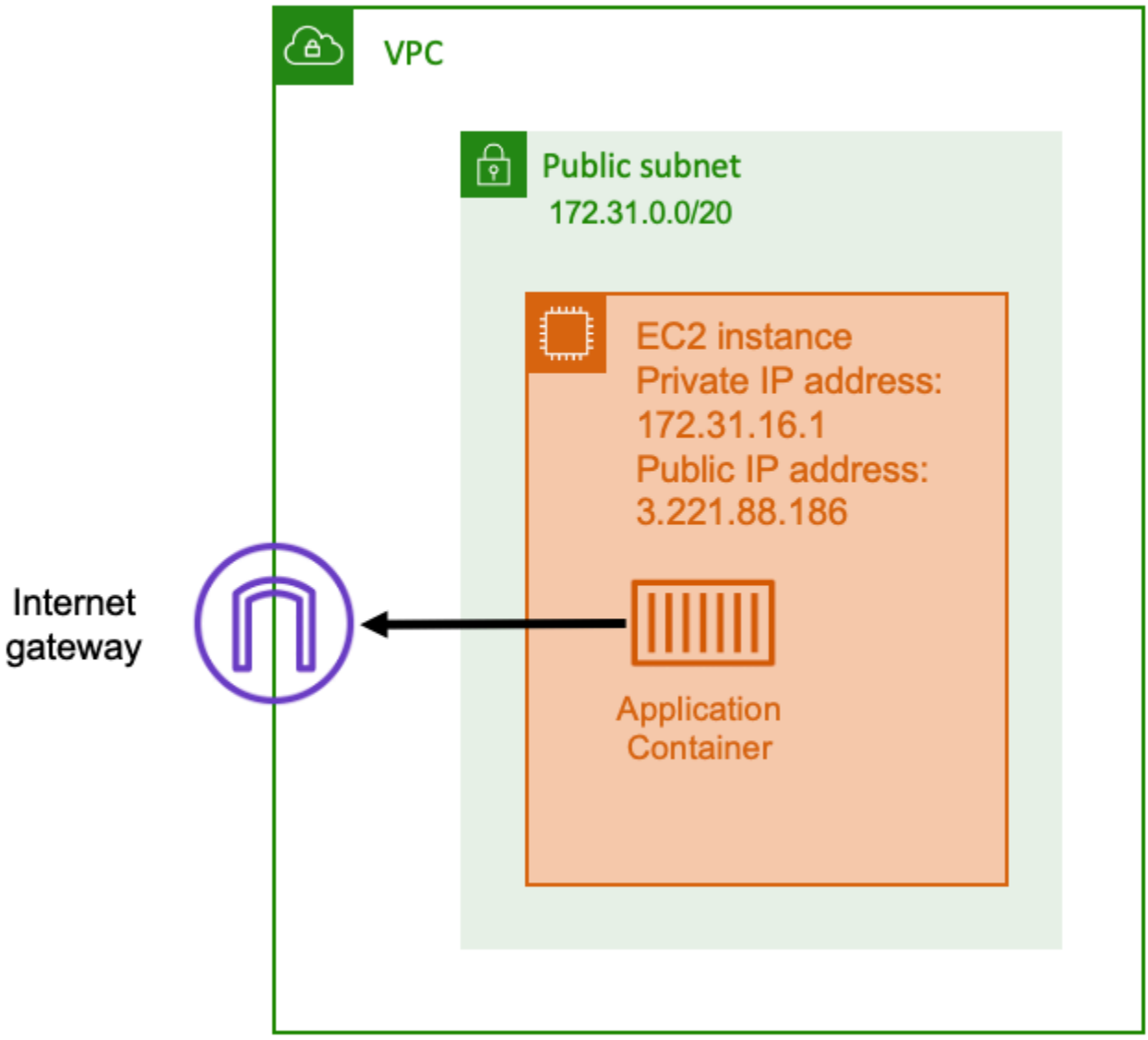
連線至網際網路

大多數容器化的應用程序至少有一些需要輸出訪問互聯網的組件。例如，行動應用程式的後端需要對推送通知的輸出存取權。

Amazon Virtual Private Cloud 有兩種主要方法，用於促進 VPC 和互聯網之間的通信。

主題

使用公用子網路和網際網路閘道



通過使用具有網際網路閘道路由的公有子網路，您的容器化應用程式可以在公有子網路上 VPC 內的主機上運行。運行您的容器的主機會分配一個公有 IP 地址。此公用 IP 位址可從網際網路路由傳送。如需詳細資訊，請參閱「」網際網路閘道中的Amazon VPC 使用者指南。

此網路架構有助於執行應用程式的主機與網際網路上其他主機之間的直接通訊。通訊是雙向的。這表示您不僅可以建立與網際網路上任何其他主機的輸出連線，而且網際網路上的其他主機也可能嘗試連線到您的主機。因此，您應該密切注意安全性群組和防火牆規則。這是為了確保網際網路上的其他主機無法開啟任何您不想開啟的連線。

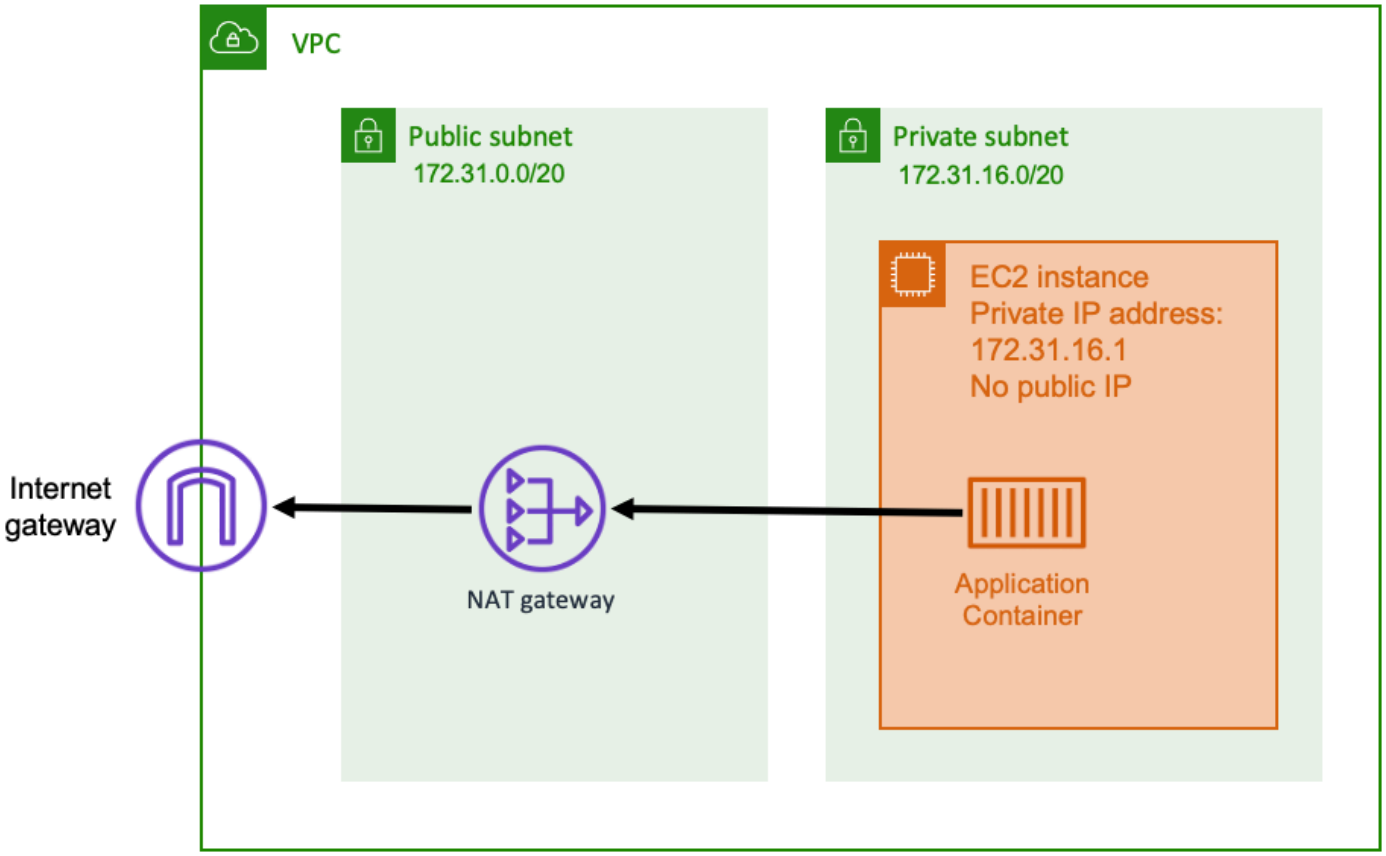
例如，如果您的應用程式在 Amazon EC2 上執行，請確定未開啟用於 SSH 存取的連接埠 22。否則，您的實例可能會接收來自互聯網上危險的機器人的持續 SSH 連接嘗試。這些機器人會透過公用 IP 位址進行拖網。在他們找到一個開放的 SSH 端口後，他們嘗試強制密碼來嘗試訪問您的實例。因此，許多組織會限制公用子網路的使用，而且喜歡在私用子網路內擁有大部分 (如果不是全部) 資源。

使用公用子網路進行網路適用於需要大量頻寬或最小延遲的公用應用程式。適用的使用案例包括視訊串流和遊戲服務。

在 Amazon EC2 上使用 Amazon ECS 以及在AWS Fargate。

- 使用 Amazon EC2 — 您可以在公有子網路上啟動 EC2 執行個體。Amazon ECS 使用這些 EC2 執行個體作為叢集容量，執行個體上執行的任何容器都可以使用主機的基礎公用 IP 地址進行輸出聯網。這適用於`host`和`bridge`網路模式。不過，`awsvpc`網路模式不提供公有 IP 地址的任務 ENI。因此，他們不能直接使用網際網路閘道。
- 使用 Fargate — 建立 Amazon ECS 服務時，請為服務的聯網配置指定公用子網路，並確保指派公有 IP 地址選項已啟用。每個 Fargate 任務都在公共子網絡中進行聯網，並且有自己的公共 IP 地址，用於與互聯網直接通信。

使用私人子網路和 NAT 閘道



藉由使用私人子網路和 NAT 閘道，您可以在私人子網路中的主機上執行容器化應用程式。因此，此主機具有可在 VPC 內路由的私有 IP 地址，但不可從互聯網路路由。這意味著 VPC 內的其他主機可以使用其私有 IP 地址連接到主機，但互聯網上的其他主機無法向主機進行任何輸入通信。

使用私有子網路，您可以使用網路位址轉譯 (NAT) 閘道讓私有子網路內的主機連線至網際網路。網際網路上的主機會收到似乎來自公用子網路內 NAT 閘道的公用 IP 位址的輸入連線。NAT 閘道負責作為互聯網和私有 VPC 之間的橋樑。基於安全理由，此設定通常是首選的，因為這表示您的 VPC 受到網際網路上的攻擊者直接存取。如需詳細資訊，請參閱「」NAT 閘道中的Amazon VPC 使用者指南。

此私有網路方法適用於您想要保護容器免於直接外部存取的案例。適用案例包括付款處理系統或儲存使用者資料與密碼的容器。您需要為您帳戶中建立和使用 NAT 閘道支付費用。NAT 閘道的每小時用量率也適用。基於備援目的，您每個可用區域中應該有一個 NAT 閘道。如此一來，單一可用區域的可用性喪失並不會影響您的輸出連線。因此，如果您的工作負載很小，則使用私人子網路和 NAT 閘道可能會更具成本效益。

在 Amazon EC2 上使用 Amazon ECS 以及在AWS Fargate。

- 使用 Amazon EC2 — 您可以在私有子網路上啟動 EC2 執行個體。在這些 EC2 主機上執行的容器會使用基礎主機網路，而輸出要求則會透過 NAT 閘道進行。
- 使用 Fargate — 建立 Amazon ECS 服務時，請為服務的聯網配置指定私有子網路，並且不要啟用指派公有 IP 地址選項。每個 Fargate 任務都在私有子網路中託管。其輸出流量會透過與該私人子網路相關聯的任何 NAT 閘道路由傳送。

從網際網路接收傳入的連線

如果您執行公用服務，則必須接受來自網際網路的輸入流量。例如，您的公用網站必須接受來自瀏覽器的輸入 HTTP 要求。在這種情況下，互聯網上的其他主機也必須啟動到應用程序主機的輸入連接。

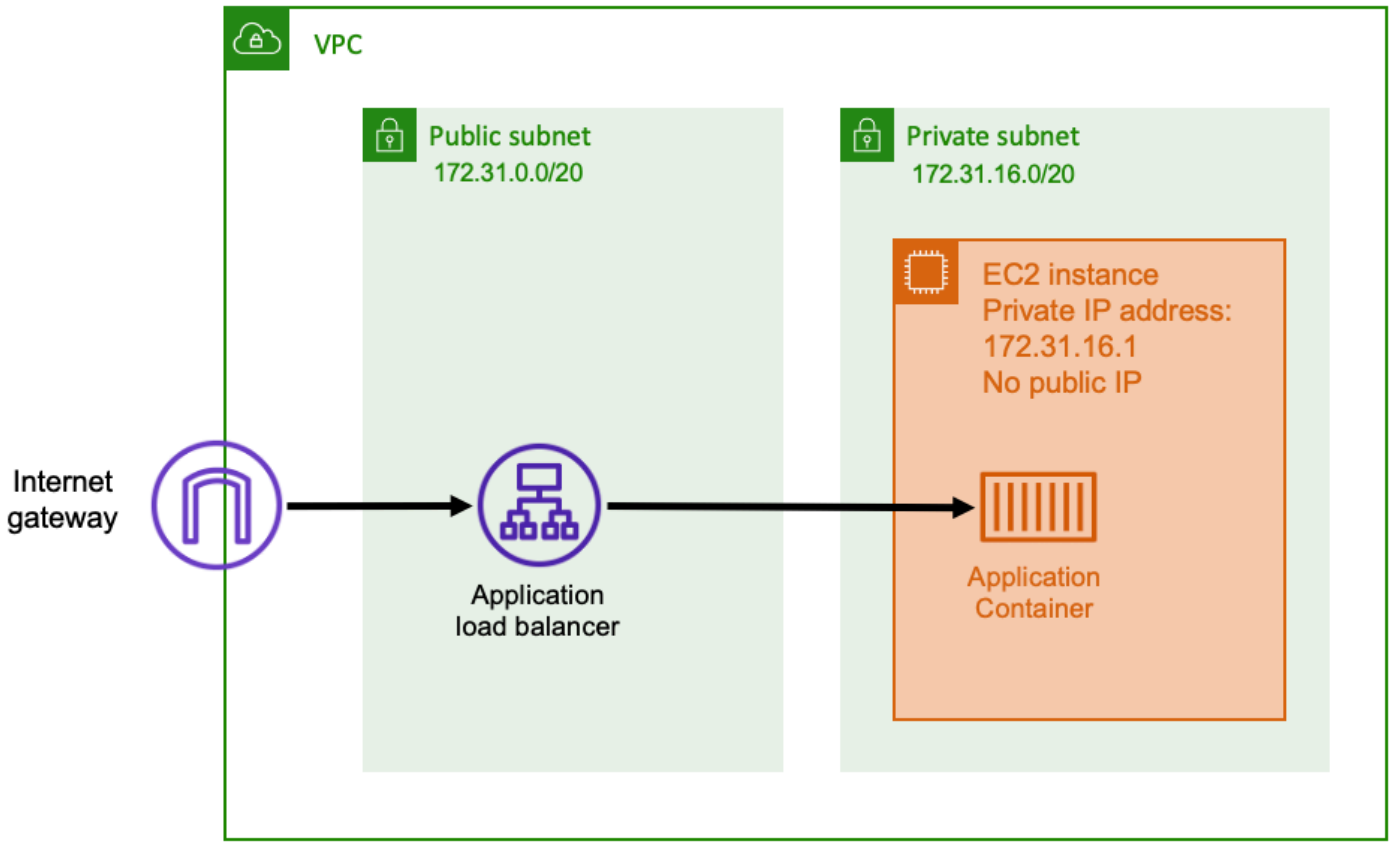
解決這個問題的一種方法是在具有公共 IP 地址的公共子網中的主機上啟動容器。不過，我們不建議使用此項目來進行大規模應用程式。對於這些，更好的方法是有一個位於互聯網和應用程序之間的可擴展輸入層。此方法可讓您使用任一AWS在本節中列出的服務作為輸入。

主題

- Application Load Balancer
- 網路負載平衡器
- Amazon API Gateway HTTP API

Application Load Balancer

Application Load Balancer 在應用程式層運作。這是開放系統互相連線 (OSI) 模型的第七層。這使得應用程序負載平衡器適用於公共 HTTP 服務。如果您有網站或 HTTP REST API，則 Application Load Balancer 是此工作負載適用的負載平衡器。如需詳細資訊，請參閱「」什麼是 Application Load Balancer？中的Application Load Balancer 使用者指南。



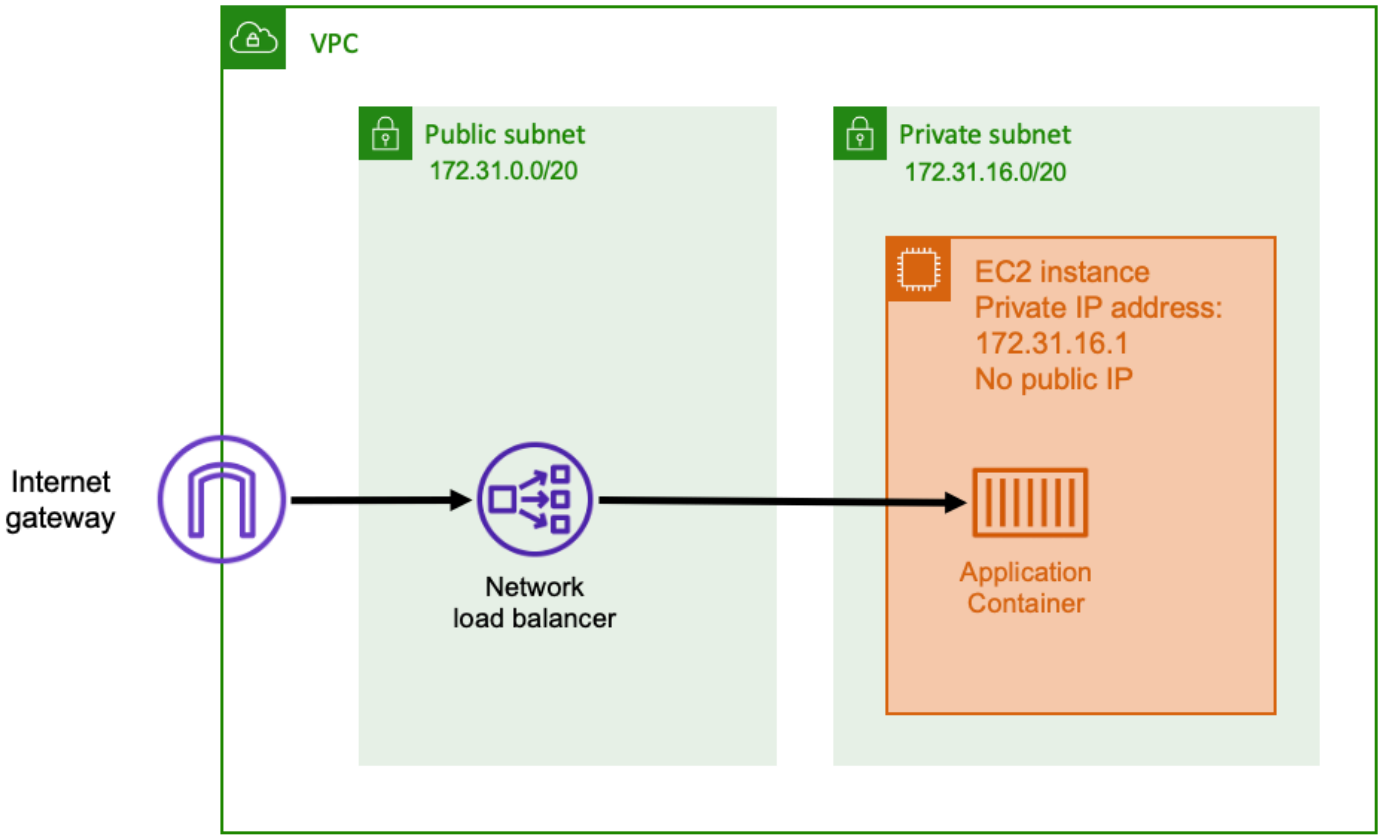
透過此架構，您可以在公用子網路中建立 Application Load Balancer，使其具有公用 IP 位址，並可以接收來自網際網路的輸入連線。當 Application Load Balancer 收到輸入連線，或更具體地說是 HTTP 要求時，它會使用其私人 IP 位址開啟應用程式的連線。然後，它通過內部連接轉發請求。

Application Load Balancer 具有下列優勢。

- SSL/TLS 終止 — Application Load Balancer 可以維持與用戶端通訊的安全 HTTPS 通訊和憑證。它可以選擇性地終止負載平衡器級別的 SSL 連接，這樣您就不必在自己的應用程序中處理證書。
- 進階路由 — Application Load Balancer 可以有多個 DNS 主機名稱。它還具有先進的路由功能，可根據諸如主機名稱或要求的路徑等度量，將傳入的 HTTP 請求發送到不同的目的地。這表示您可以使用單一 Application Load Balancer 作為許多不同內部服務的輸入，甚至可以在 REST API 的不同路徑上使用微服務。
- GRPC 的支持和網絡套接字-Application Load Balancer 可以處理不僅僅是 HTTP。它也可以負載平衡 GRPC 和基於網絡插座的服務，具有 HTTP/2 的支持。
- 安全性 — Application Load Balancer 可協助保護您的應用程式免受惡意流量侵害 它包含 HTTP 同步緩和措施等功能，並與 AWS Web 應用程式防火牆 (AWS WAF。AWS WAF可以進一步篩選出可能包含攻擊模式的惡意流量，例如 SQL 插入或跨網站指令碼。

網路負載平衡器

Network Load Balancer 可在開放系統互相連線 (OSI) 模型的第四層運作。它適用於需要端對端加密的非 HTTP 通訊協定或案例，但不具有 Application Load Balancer 相同的 HTTP 特定功能。因此，Network Load Balancer 最適合不使用 HTTP 的應用程式。如需詳細資訊，請參閱「」什麼是 Network Load Balancer？中的網路負載平衡器使用者指南。



當 Network Load Balancer 用作輸入時，其運作方式與 Application Load Balancer 類似。這是因為它是在公有子網路中建立的，而且具有可在網際網路上存取的公有 IP 地址。然後，Network Load Balancer 器打開到運行容器的主機的私有 IP 地址的連接，並將數據包從公共端發送到私有端。

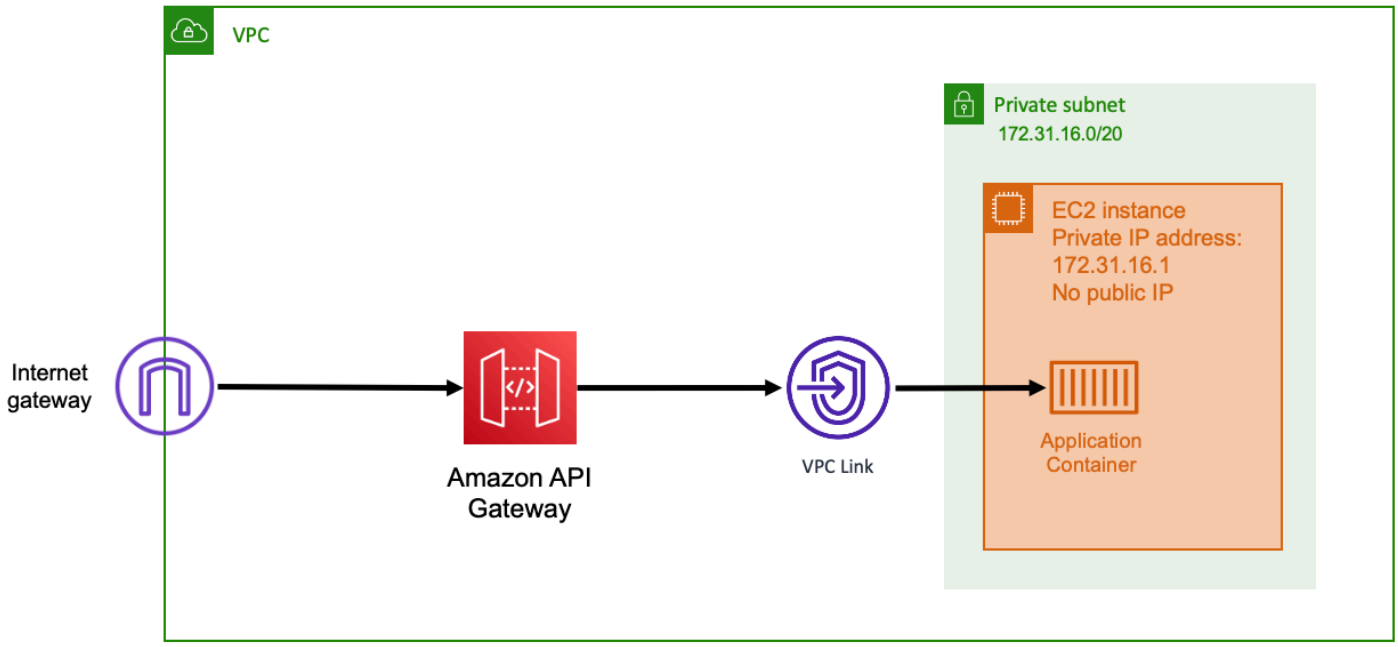
由於 Network Load Balancer 在網路堆疊的較低層級運作，因此它沒有與 Application Load Balancer 相同的功能集。但是，它確實具有以下重要功能。

- 端對端加密 — 由於 Network Load Balancer 在 OSI 模型的第四層運作，因此不會讀取封包的內容。這使得它適用於需要端對端加密的負載平衡通訊。
- TLS 加密 — 除了端對端加密之外，Network Load Balancer 也可以終止 TLS 連線。如此一來，您的後端應用程式就不必實作自己的 TLS。

- UDP 支援 — 由於 Network Load Balancer 在 OSI 模型的第四層運作，因此適用於非 HTTP 工作負載和 TCP 以外的通訊協定。

Amazon API Gateway HTTP API

Amazon API Gateway HTTP API 是一種較少入口的服務器，適合在請求卷或請求卷突然爆發的 HTTP 應用程序。如需詳細資訊，請參閱「」什麼是 Amazon API Gateway？中的API Gateway 開發人員指南。



Application Load Balancer 和 Network Load Balancer 的定價模型包含每小時的價格，可讓負載平衡器隨時接受傳入連線。相較之下，API Gateway 會分別收取每個要求的費用。這樣的效果是，如果沒有請求進來，則不會收取任何費用。在高流量負載下，Application Load Balancer 或 Network Load Balancer 可以以比 API Gateway 更便宜的每個請求價格來處理大量的請求。不過，如果您的整體要求數量很少或流量很低，則使用 API Gateway 的累積價格應該比支付小時費用來維護未充分利用的負載平衡器更具成本效益。

API Gateway 函式使用 VPC 連結，允許AWS受管理服務，使用其私有 IP 位址連線至 VPC 私有子網路內的主機。它可以檢測這些私人 IP 地址，通過查看AWS Cloud Map由 Amazon ECS 服務探索管理的服務探索記錄。

API Gateway 支援下列功能。

- SSL/TLS 終止

- 將不同的 HTTP 路徑路由到不同的後端微服務

除了上述功能之外，API Gateway 還支援使用自訂 Lambda 授權器，您可以使用這些授權器來保護您的 API 免於未經授權的使用。如需詳細資訊，請參閱「」欄位備註：使用 Amazon ECS 和 Amazon API 閘道的無伺服器容器型 API。

選擇網路模式

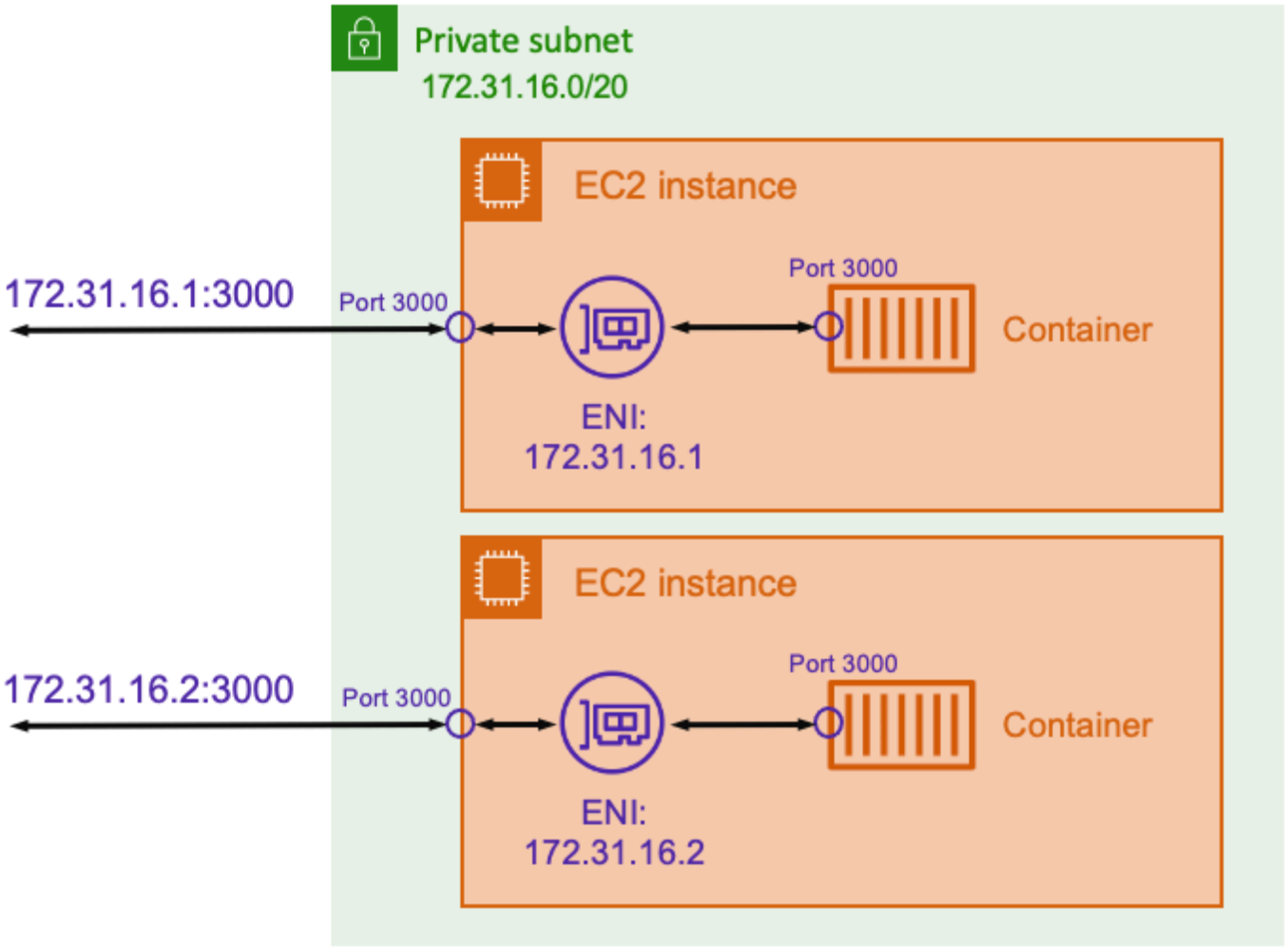
先前提到的架構輸入和輸出網路連線的方法，可套用至AWS，即使它們不在容器內。執行容器時，在AWS，您需要考慮另一層級的網絡。使用容器的主要優點之一是您可以將多個容器封裝到單一主機上。執行這項操作時，您需要選擇要如何將相同主機上執行的容器進行網路連線。以下是可供選擇的選項。

主題

- 主機模式
- 橋接模式
- AWVPC 模式

主機模式

所以此host網路模式是 Amazon ECS 支援的最基本網路模式。使用主機模式，容器的網絡直接綁定到正在運行容器的基礎主機。



假設您正在執行 Node.js 容器與快速應用程式偵聽連接埠3000類似於前面的圖表所示。當您host網路模式時，容器會使用基礎主機 Amazon EC2 執行個體的 IP 位址在連接埠 3000 上接收流量。我們不建議使用此模式。

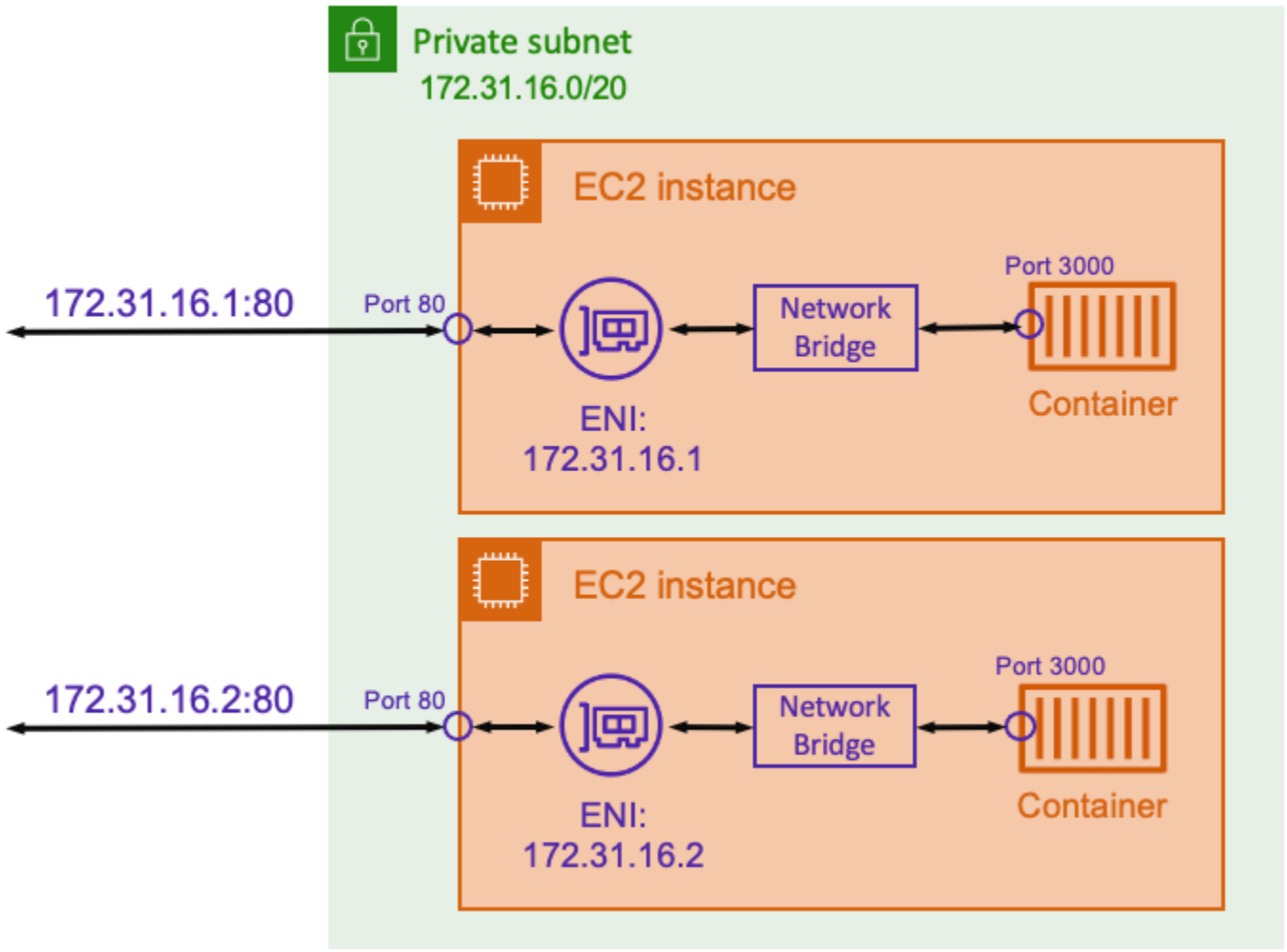
使用此網路模式有顯著的缺點。您不能在每台主機上執行任務的多個實例化。這是因為只有第一個任務可以綁定到 Amazon EC2 執行個體上的所需端口。當容器端口使用host網路模式。例如，如果應用程式需要偵聽特定的連接埠號碼，則無法直接重新對應連接埠號碼。相反地，您必須透過變更應用程式組態來管理任何連接埠衝突。

在使用host網路模式。此模式允許容器模擬主機，並允許容器連線到主機上的私人迴路網路服務。

所以此host網路模式僅支援在 Amazon EC2 執行個體上託管的 Amazon ECS 任務。在 Fargate 上使用亞馬遜 ECS 時不受支援。

橋接模式

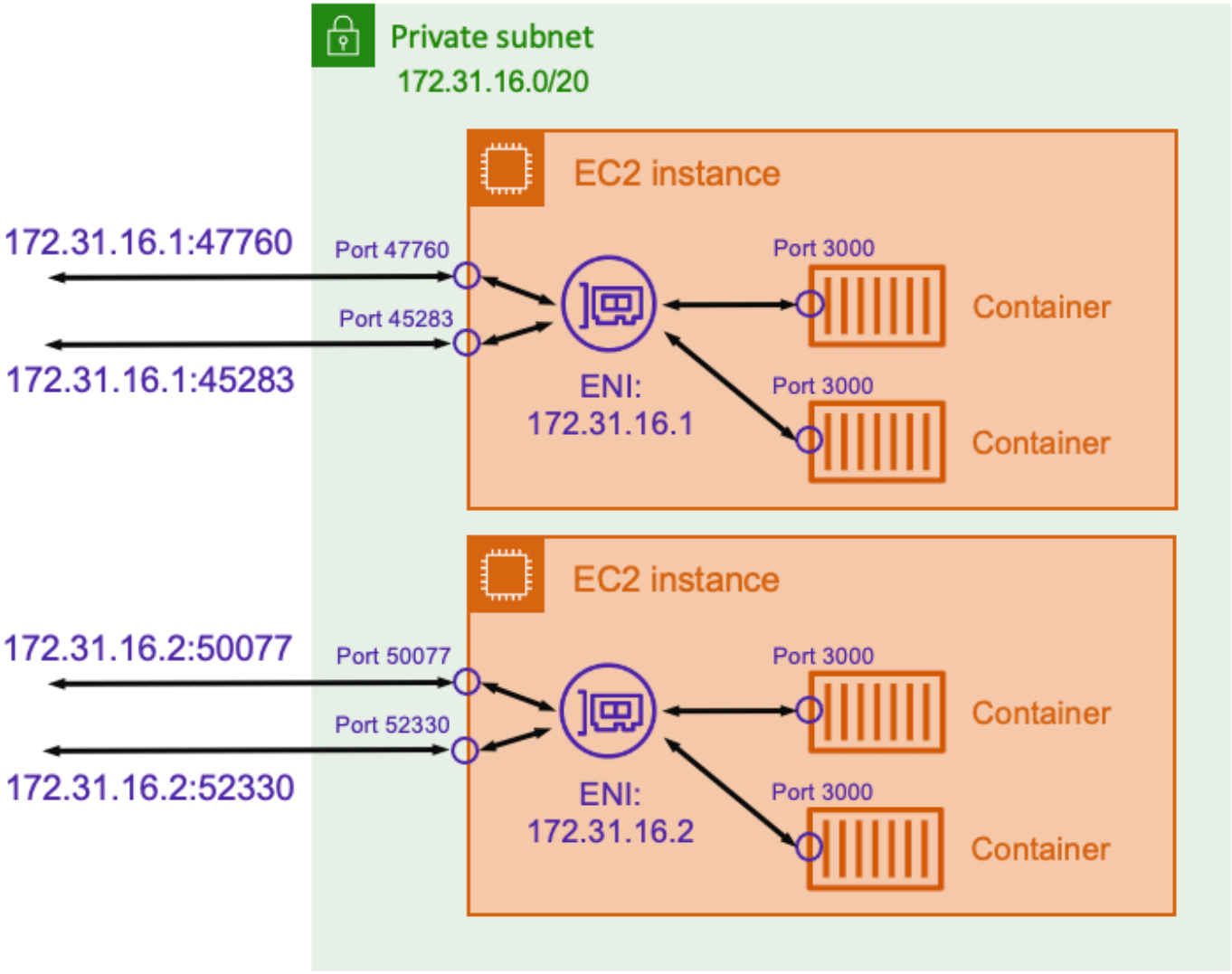
搭配bridge模式中，您正在使用虛擬網絡橋接器在主機和容器的網絡之間創建一個層。如此一來，您就可以建立連接埠對應，將主機連接埠重新對應至容器連接埠。對映可以是靜態或動態。



使用靜態連接埠對應，您可以明確定義要對應至容器連接埠的主機連接埠。使用上述範例，連接埠80正在映射到端口3000(位於容器)。若要與容器化應用程式進行通訊，您可以將流量傳送至連接埠80到 Amazon EC2 執行個體的 IP 地址。從容器化應用程序的角度來看，它可以看到端口3000。

如果您只想變更流量連接埠，則適合靜態連接埠對應。但是，這仍然具有與使用host網路模式。您不能在每台主機上執行任務的多個實例化。這是因為靜態連接埠對應只允許單一容器對應至連接埠 80。

若要解決此問題，請考慮使用bridge網路模式與動態連接埠對應，如下圖所示。



藉由不在連接埠對應中指定主機連接埠，您可以讓 Docker 從暫時連接埠範圍中選擇一個隨機、未使用的連接埠，並將其指派為容器的公用主機連接埠。例如，Node.js 應用程式偵聽連接埠`3000`可能會被分配一個隨機的高數字端口，例如`47760`(位於 Amazon EC2 主機)。這樣做意味著您可以在主機上運行該容器的多個副本。此外，每個容器都可以在主機上分配自己的端口。容器的每個副本都會在連接埠上接收流量`3000`。不過，傳送流量至這些容器的用戶端會使用隨機指派的主機連接埠。

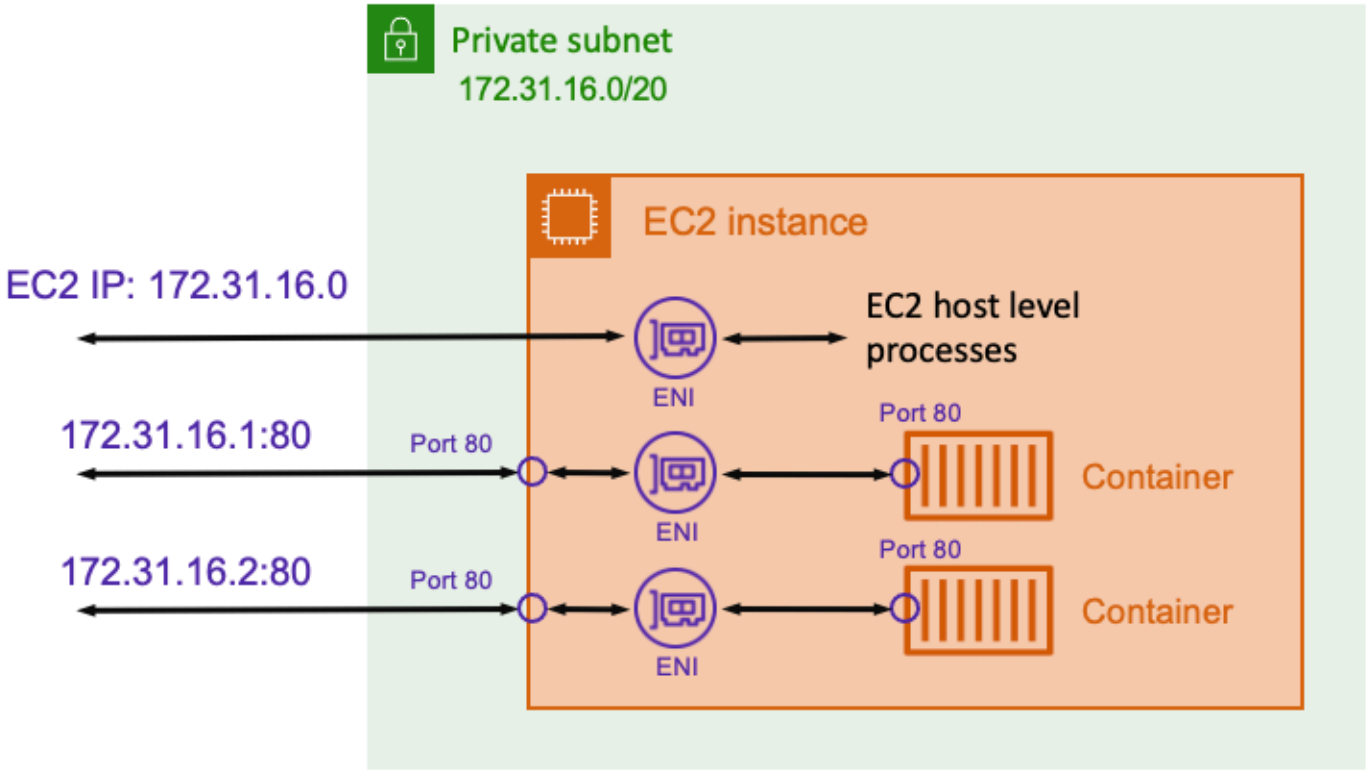
Amazon ECS 可協助您持續追蹤每個任務的隨機指派連接埠。它會自動更新負載平衡器目標群組並 AWS Cloud Map服務探索，以取得工作 IP 位址和連接埠的清單。這可讓您更輕鬆使用使用`bridge`模式與動態連接埠。

但是，使用bridge網路模式是很難將服務鎖定至服務通訊。由於服務可能會指派給任何隨機、未使用的連接埠，因此必須在主機之間開啟廣泛的連接埠範圍。不過，建立特定規則並不容易，讓特定服務只能與其他特定服務通訊。服務沒有特定的連接埠可用於安全性群組網路規則。

所以此bridge網路模式僅支援在 Amazon EC2 執行個體上託管的 Amazon ECS 任務。在 Fargate 上使用亞馬遜 ECS 時不支援此功能。

AWVPC 模式

使用awsvpc網路模式時，Amazon ECS 會為每個任務建立和管理彈性網路界面 (ENI)，並且每個任務會在 VPC 內接收自己的私有 IP 地址。此 ENI 是從底層主機 ENI 分開。如果 Amazon EC2 執行個體正在執行多個任務，則每個任務的 ENI 也是獨立的。



在上述範例中，Amazon EC2 執行個體會指派給 ENI。ENI 代表用於主機層級網路通訊之 EC2 執行個體的 IP 地址。每個任務也有一個對應的 ENI 和一個私人 IP 地址。因為每個 ENI 都是獨立的，所以每個容器都可以綁定到端口80在任務 ENI。因此，您不必跟蹤端口號碼。相反，您可以將流量發送到端口80在任務 ENI 的 IP 地址。

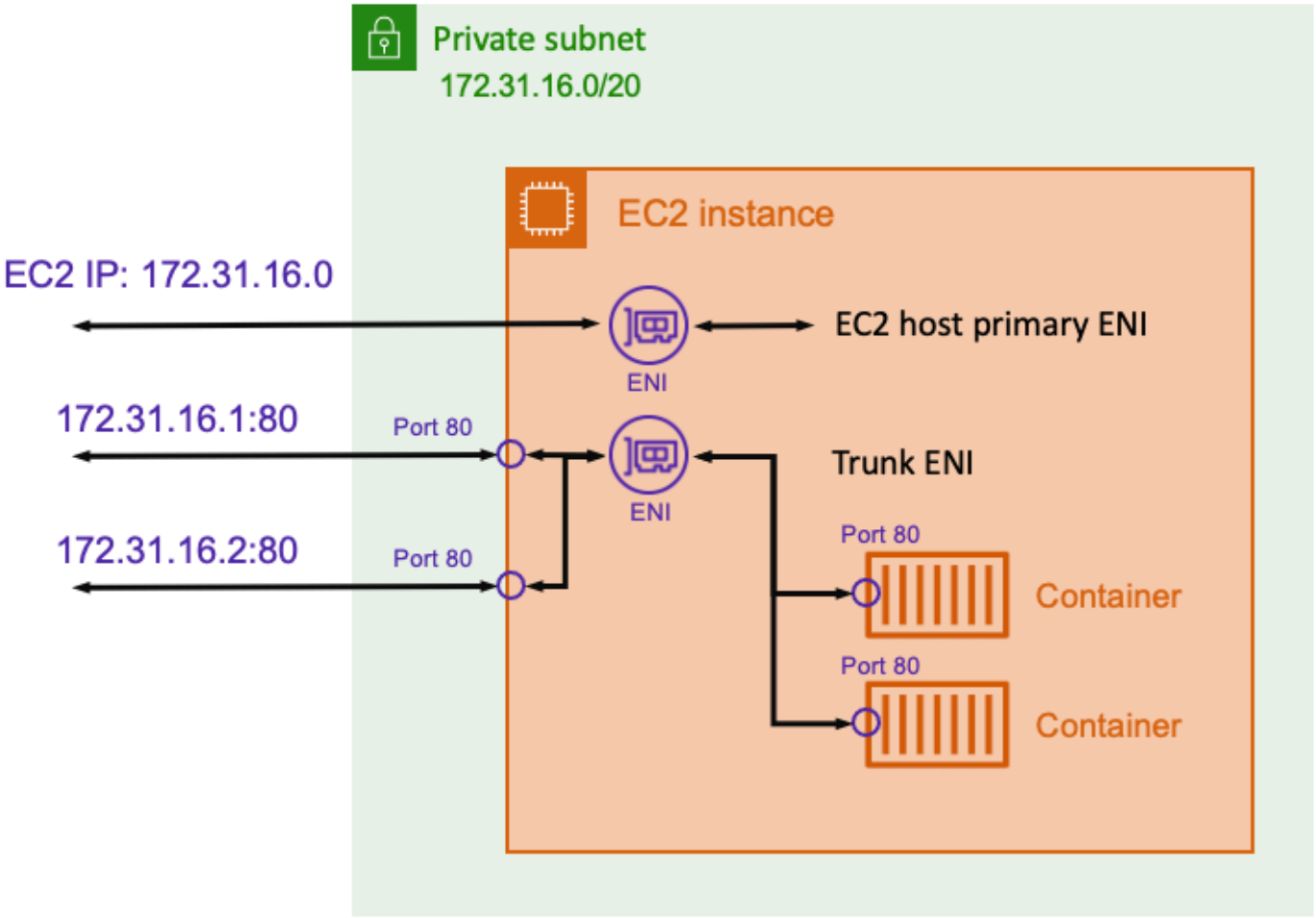
使用awsvpc網路模式的重要原因是每個工作都有個別的安全性群組來允許或拒絕流量。這表示您有更大的彈性，以更精細的層級控制工作與服務之間的通訊。您也可以將工作設定為拒絕來自同一主機上其他工作的傳入流量。

所以此awsvpc網路模式支援在 Amazon EC2 和 Fargate 上託管的 Amazon ECS 任務。請注意，在使用 Fargate 時，awsvpc網路模式是必要的。

當您使用awsvpc網路模式中，您應該注意幾個挑戰。

使用 ENI 中繼增加任務密度

最大的缺點是使用awsvpc網路模式與 Amazon EC2 執行個體上託管的任務是 EC2 執行個體對可以連接到它們的 ENI 數量有限。這會限制您可以在每個執行個體上放置多少工作。Amazon ECS 提供 ENI 中繼功能，可增加可用 ENI 的數量，以達到更高的任務密度。



使用 ENI 主幹時，預設會使用兩個 ENI 附件。第一個是執行個體的主要 ENI，用於任何主機層級處理程序。第二個是主幹 ENI，亞馬遜 ECS 創建。只有特定 Amazon EC2 執行個體類型才支援此功能。

考慮這個範例。如果沒有 ENI 主幹，c5.large執行個 vCPUs 只能裝載兩個工作。但是，使用 ENI 中繼，c5.large執行個體最多可以裝載十個工作。每個工作都有不同的 IP 位址和安全性群組。如需可用執行個體類型及其密度的詳細資訊，請參閱支援的 Amazon EC2 執行個體類型中的Amazon Elastic Container Service Container Service。

ENI 中繼在延遲或頻寬方面對運行時性能沒有影響。不過，它會增加工作啟動時間。您應該確保如果使用 ENI 主幹，則依賴於任務啟動時間的自動調整規則和其他工作負載仍然如預期般運作。

如需詳細資訊，請參閱「」彈性網路界面中繼中的Amazon Elastic Container Service Container Service。

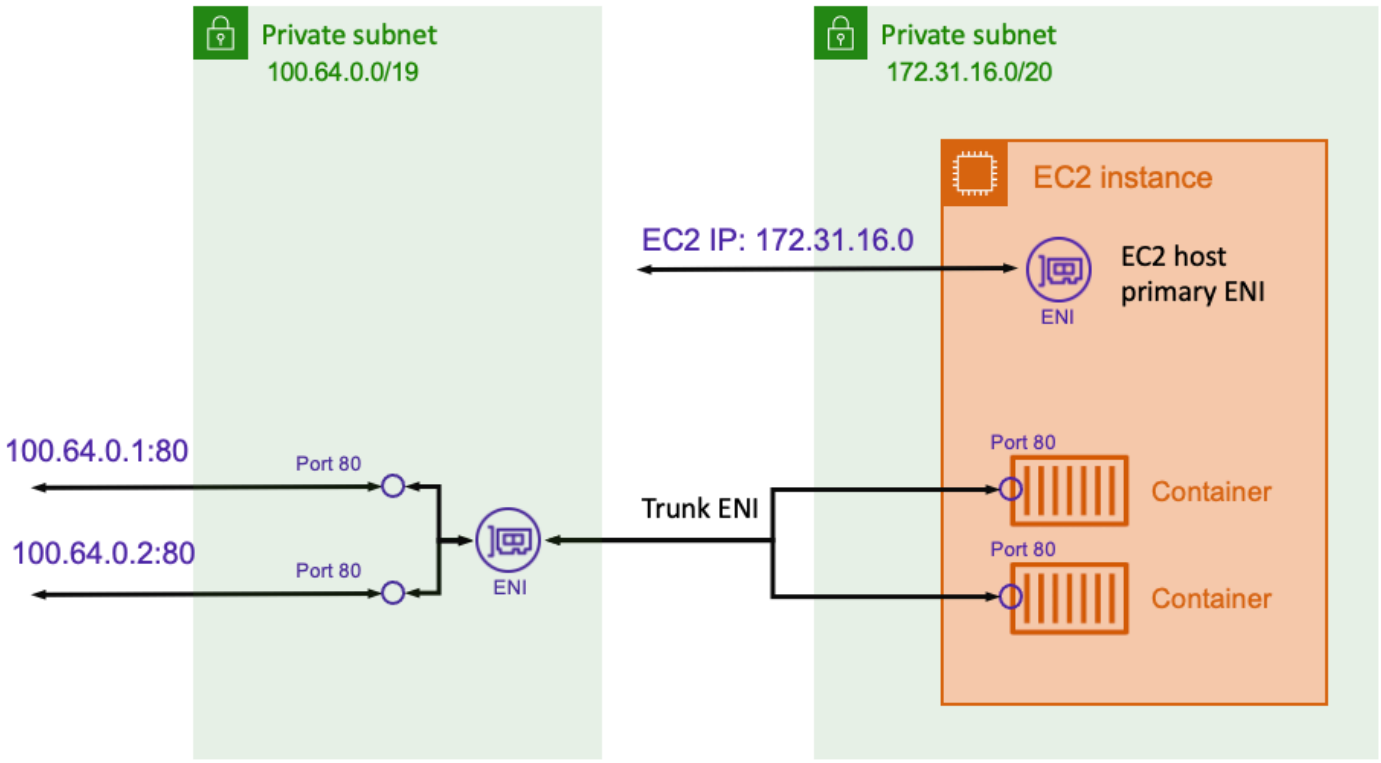
防止 IP 地址耗盡

透過為每項工作指派不同的 IP 位址，您可以簡化整體基礎結構，並維護提供絕佳安全性層級的安全性群組。不過，此組態可能會導致 IP 耗盡。

您AWS帳戶已預先佈建子網路，該子網路具有/20CIDR 範圍。這表示每個子網路都有 4,091 個可用的 IP 位址。請注意，數個 IP 位址/20範圍已留給 AWS 特定用途。考慮這個範例。您可以將您的應用程式分佈到三個可用區域中的三個子網路以提供高可用性。在這種情況下，您可以在三個子網路中使用大約 12,000 個 IP 位址。

使用 ENI 中繼，您啟動的每個 Amazon EC2 執行個體都需要兩個 IP 地址。一個 IP 位址用於主要 ENI，而另一個 IP 位址用於主幹 ENI。執行個體上的每個 Amazon ECS 任務都需要一個 IP 地址。如果您正在啟動極大的工作負載，可能會用盡可用的 IP 位址。這可能會導致 Amazon EC2 啟動失敗或任務啟動失敗。發生這些錯誤是因為如果沒有可用的 IP 位址，ENIS 無法在 VPC 內新增 IP 位址。

當您使用awsvpc網路模式，您應該評估您的 IP 位址需求，並確保您的子網路 CIDR 範圍符合您的需求。如果您已經開始使用具有小子網路的 VPC，且位址空間不足，則可以新增次要子網路。



透過使用 ENI 主幹，Amazon VPC CNI 可以配置為在與主機不同的 IP 位址空間中使用 ENI。通過這樣做，您可以為 Amazon EC2 主機和任務提供不重疊的不同 IP 地址範圍。在範例圖中，EC2 主機 IP 位址位於具有`172.31.16.0/20`IP 範圍。不過，在主機上執行的工作會在`100.64.0.0/19`範圍。透過使用兩個獨立的 IP 範圍，您不必擔心工作耗用太多 IP 位址，也不會為執行個體留下足夠的 IP 位址。

使用 IPv6 雙堆疊模式

所以此`awsvpc`網路模式與設定為 IPv6 雙堆疊模式的 VPC 相容。使用雙堆疊模式的 VPC 可透過 IPv4、IPv6 或兩者進行通訊。VPC 中的每個子網路可以同時具有 IPv4 CIDR 範圍和 IPv6 CIDR 範圍。如需詳細資訊，請參閱「」您 VPC 中的 IP 定址中的Amazon VPC 使用者指南。

您無法停用 VPC 和子網路的 IPv4 支援來解決 IPv4 耗盡問題。不過，有了 IPv6 支援，您可以使用某些新功能，尤其是僅限輸出網際網路閘道。僅限輸出網際網路閘道允許任務使用其公有路由至網際網路的輸出連線。但僅輸出網際網路閘道不允許從網際網路連線。如需詳細資訊，請參閱「」輸出限定網際網路閘道中的Amazon VPC 使用者指南。

連接到AWS從您的 VPC 內部服務

為了讓 Amazon ECS 正常運作，在每台主機上執行的 ECS 容器代理程式必須與 Amazon ECS 控制平面進行通訊。如果您要將容器映像儲存在 Amazon ECR 中，則 Amazon EC2 主機必須與 Amazon

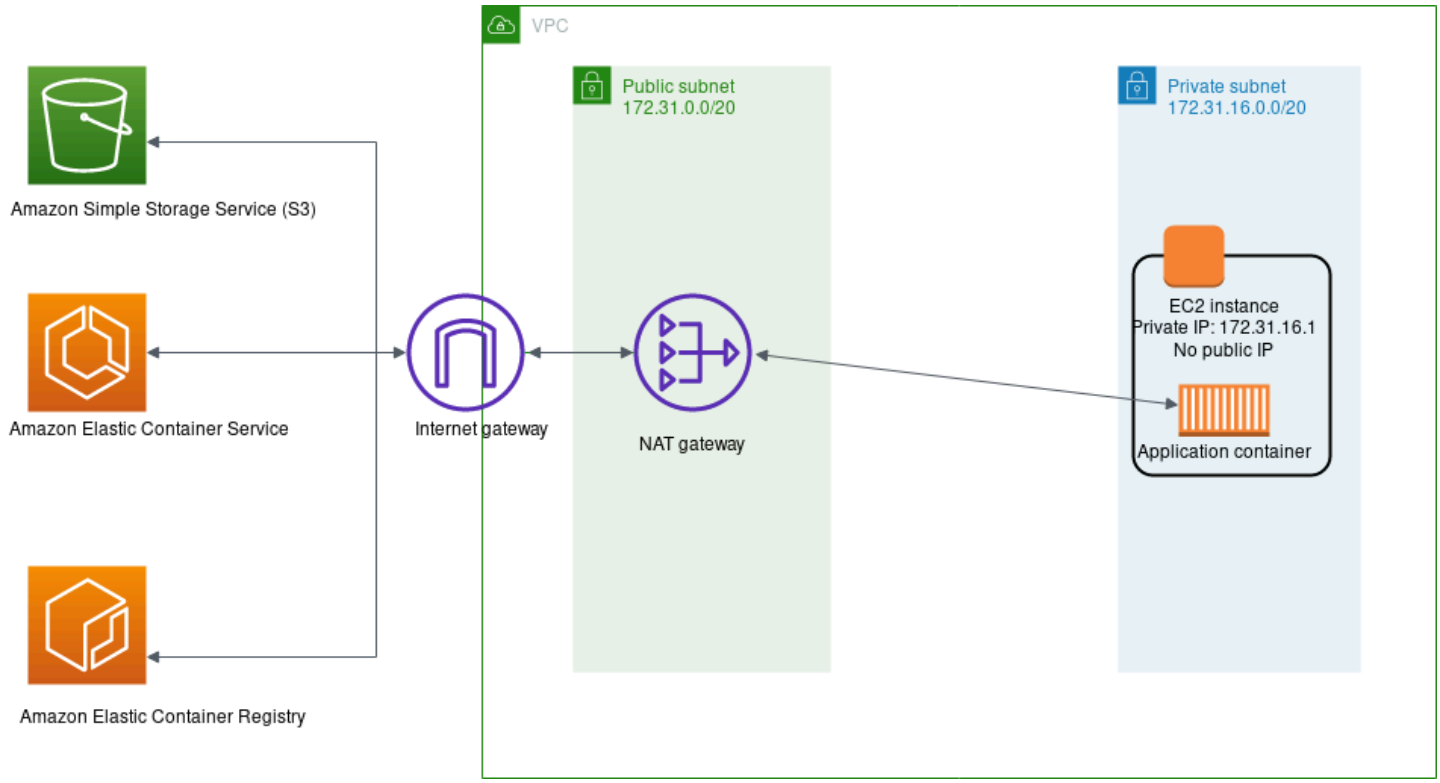
ECR 服務終端節點和儲存映像層的 Amazon S3 進行通訊。如果您使用其他AWS服務 (例如保留儲存在 DynamoDB 中的資料)，請仔細檢查這些服務是否也具有必要的網路支援。

主題

- NAT 閘道
- AWS PrivateLink

NAT 閘道

使用 NAT 閘道是確保 Amazon ECS 任務可存取其他AWS服務。如需有關此方法的詳細資訊，請參閱使用私人子網路和 NAT 閘道。



以下是使用此方法的缺點：

- 您無法限制 NAT 閘道可與哪些目的地通訊。您也無法限制後端輪胎可以通訊的目的地，而不會中斷 VPC 的所有輸出通訊。
- NAT 閘道會針對傳遞的每 GB 資料收費。如果您使用 NAT 閘道從 Amazon S3 下載大型檔案，或對 DynamoDB 進行大量資料庫查詢，則每 GB 頻寬需支付費用。此外，NAT 閘道支援 5 Gbps 的頻寬，並可自動擴展至 45 Gbps。如果您透過單一 NAT 閘道路由，需要極高頻寬連線的應用程式可能

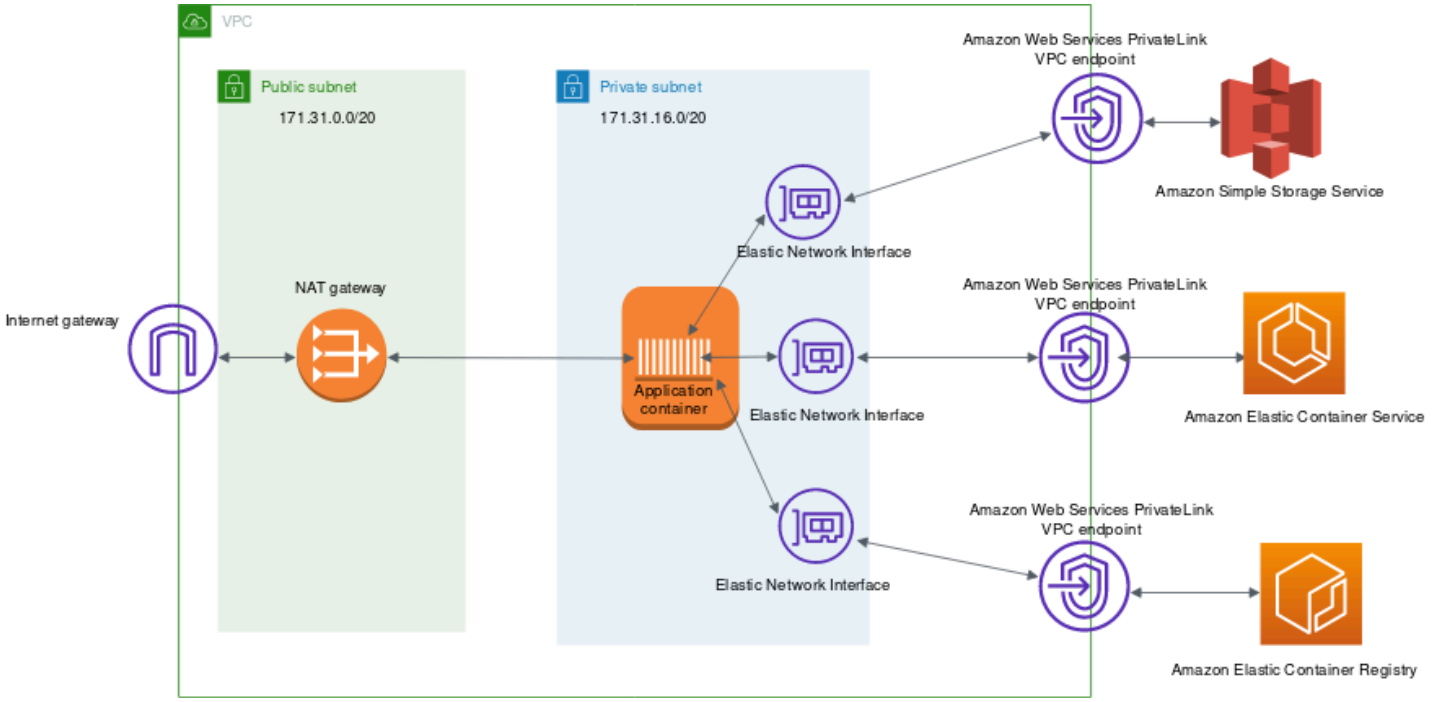
會遇到網路限制。因應措施是，您可以將工作負載分割為多個子網路，並為每個子網路提供自己的 NAT 閘道。

AWS PrivateLink

AWS PrivateLink提供 VPC 之間的私有連線，AWS服務和您的內部部署網路，而不會將您的流量暴露到公用網際網路。

其中一項用於實現此目標的技術是 VPC 端點。VPC 端點可讓您的 VPC 與受支援的 VPC 端點之間進行私人連線AWS服務和 VPC 端點服務。VPC 與另一個服務之間的流量都會保持在 Amazon 網路的範圍內。VPC 端點不需要網際網路閘道、虛擬私有閘道、NAT 裝置、VPN 連接或AWS Direct Connect 連線。VPC 中的 Amazon EC2 執行個體不需要公有 IP 地址，即可與服務中的資源通訊。

下圖顯示如何通訊AWS服務會在您使用 VPC 端點而不是網際網路閘道時運作。AWS PrivateLink在子網內部佈建彈性網絡接口（ENI），並且使用 VPC 路由規則通過 ENI 將任何通信發送到服務主機名，直接發送到目的地AWS服務。此流量不再需要使用 NAT 閘道或網際網路閘道。



以下是與 Amazon ECS 服務搭配使用的一些常見 VPC 端點。

- S3 閘道 VPC 端點
- VPC 端點
- Amazon ECS VPC 端點

- Amazon ECR VPC 端點

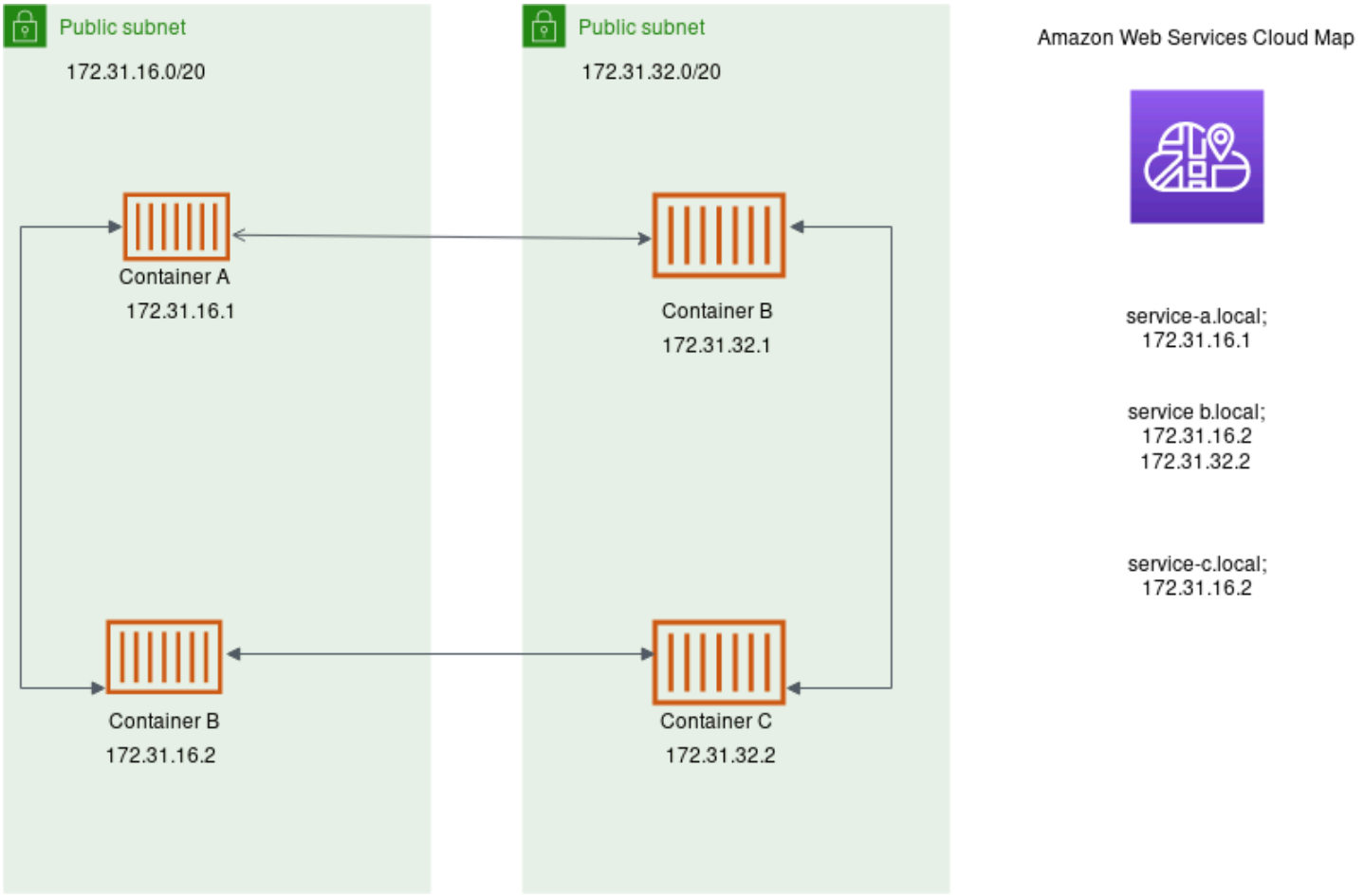
許多其他AWS服務支援 VPC 端點。如果您大量使用任何AWS服務，您應該查詢該服務的特定文件，以及如何為該流量建立 VPC 端點。

VPC 中的 Amazon ECS 服務之間的聯網

在 VPC 中使用 Amazon ECS 容器，您可以將整體應用程式溢出到不同的部分，這些部分可以在安全的環境中獨立部署和擴展。但是，確保 VPC 內外的所有這些零件都可以彼此通訊可能會很困難。有幾種方法用於促進溝通，所有方法都有不同的優點和缺點。

使用服務探索

服務對服務的通訊方法之一是使用服務探索的直接通訊。此方法可讓您使用AWS Cloud Map服務探索與 Amazon ECS 整合。使用服務探索，Amazon ECS 會將啟動的任務清單同步到AWS Cloud Map，它會維護一個 DNS 主機名稱，該主機名稱可解析為來自該特定服務的一或多個工作的內部 IP 位址。Amazon VPC 中的其他服務可以使用此 DNS 主機名稱，使用其內部 IP 位址直接將流量傳送到另一個容器。如需詳細資訊，請參閱「」服務探索中的Amazon Elastic Container Service Container Service。



在前面的圖表中，有三個服務。serviceA有一個容器，並與serviceB，其中有兩個容器。serviceB也必須與serviceC，它有一個容器。這三項服務中的每個容器都可以使用AWS Cloud Map，從需要通訊的下游服務尋找容器的內部 IP 位址。

這種服務對服務通訊的方法可提供低延遲。乍一看，它也很簡單，因為容器之間沒有額外的組件。交通直接從一個容器到另一個容器。

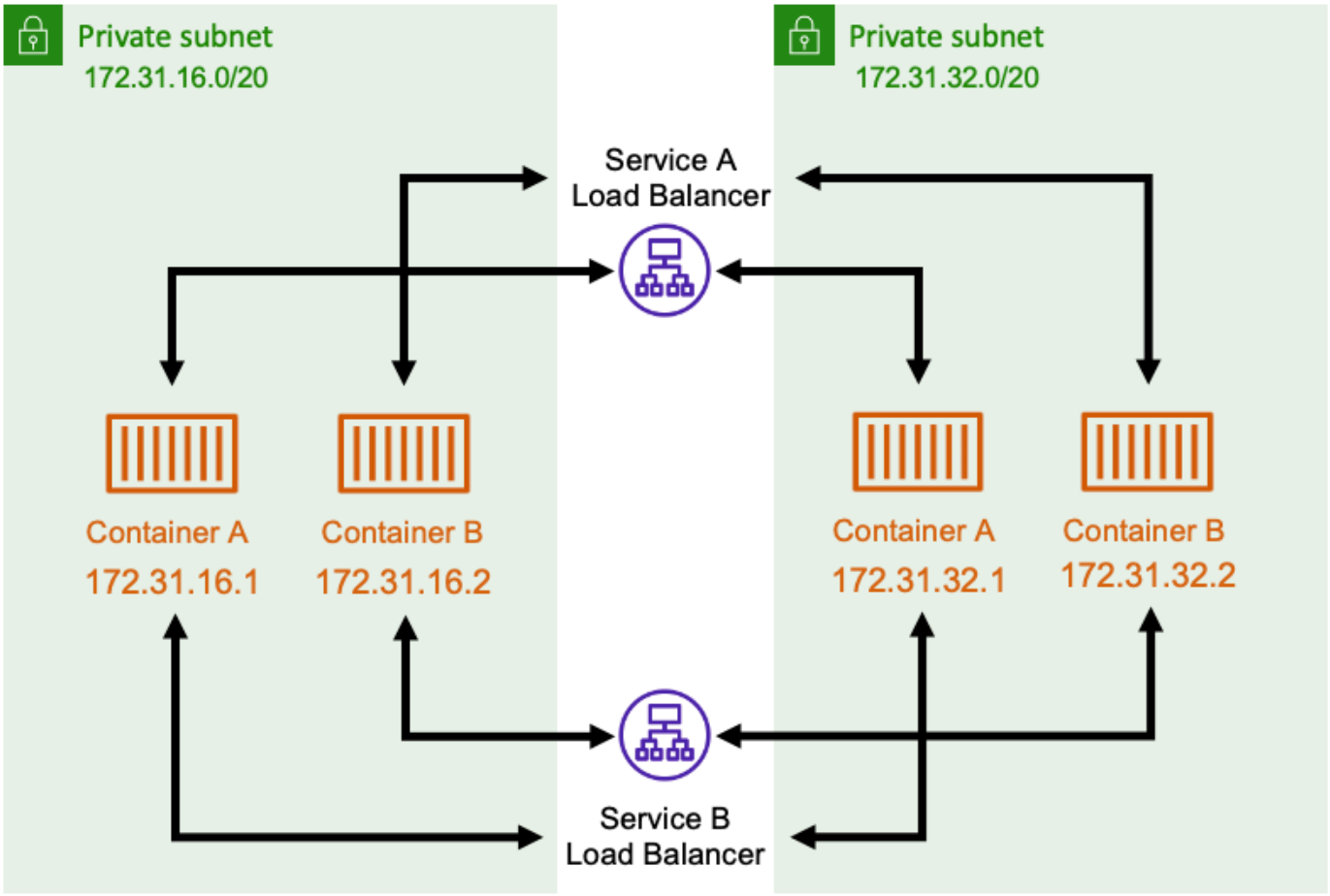
這種方法適用於使用awsvpc網路模式，其中每個工作都有自己的唯一 IP 位址。大多數軟件只支持使用 DNSA記錄，它們直接解析為 IP 地址。當您使用awsvpc網路模式時，每個工作的 IP 位址都是A記錄。但是，如果您使用的是bridge網路模式時，多個容器可以共用相同的 IP 位址。此外，動態連接埠對應會導致容器隨機指派該單一 IP 位址上的連接埠號碼。在這一點上，A記錄不再足以進行服務探索。您還必須使用SRV記錄。這種類型的記錄可以同時追蹤 IP 位址和連接埠號碼，但需要您適當地設定應用程式。您使用的某些預先建置的應用程式可能不支援SRV記錄。

另一個優點awsvpc網路模式的重要原因是每個服務都有唯一的安全性群組。您可以將此安全性群組設定為只允許來自需要與該服務交談的特定上游服務的連入連線。

使用服務探索直接服務對服務通訊的主要缺點是您必須實作額外的邏輯，才能重試並處理連線失敗。DNS 記錄具有存活期 (TTL) 期間，可控制快取的時間。需要一些時間才能更新 DNS 記錄和快取過期，以便您的應用程式可以取得最新版本的 DNS 記錄。因此，您的應用程序可能最終解析 DNS 記錄以指向另一個不再存在的容器。您的應用程序需要處理重試，並有邏輯忽略壞的後端。

使用內部負載平衡器

服務對服務通訊的另一種方法是使用內部負載平衡器。內部負載平衡器完全存在於 VPC 內部，只有 VPC 內部的服務才能存取。



負載平衡器會將備援資源部署至每個子網路，以維持高可用性。當從容器`serviceA`需要與來自`serviceB`，它會開啟與負載平衡器的連線。然後負載平衡器會從`service B`。負載平衡器可作為管理每個服務之間所有連線的集中位置。

如果從容器`serviceB`停止，則負載平衡器可以從集區中移除該容器。負載平衡器也會針對其集區中的每個下游目標執行健全狀況檢查，並可自動從集區移除不良目標，直到它們再次健全狀況為止。應用程序不再需要知道有多少下游容器存在。他們只是打開與負載平衡器的連線。

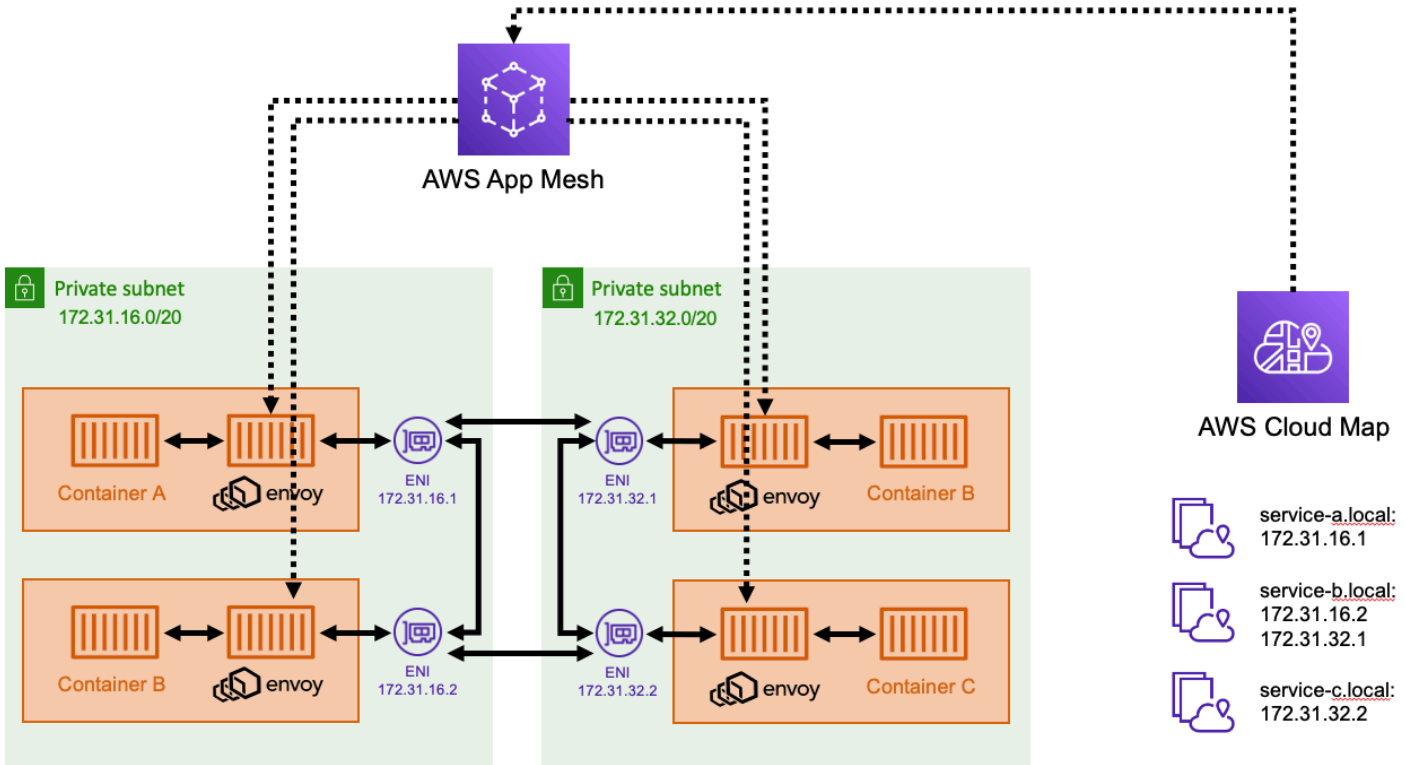
這種方法對所有網路模式都有利。負載平衡器可以在使用awsvpc網路模式，以及使用bridge網路模式。它將流量均勻分配到所有 IP 地址和連接埠組合，即使多個容器實際託管在同一個 Amazon EC2 執行個體上，只是在不同的連接埠上。

這種方法的一個缺點是成本。為了達到高可用性，負載平衡器需要在每個可用區域中擁有資源。這會增加額外的成本，因為負載平衡器支付額外費用，以及通過負載平衡器的流量量。

不過，您可以讓多個服務共用負載平衡器，以降低額外負荷成本。這特別適用於使用 Application Load Balancer 的 REST 服務。您可以建立路徑型路由規則，將流量路由傳送至不同服務。例如：/api/user/*可能會路由到屬於user服務，而/api/order/*可能會路由到相關聯的order服務。使用這種方法，您只需支付一個 Application Load Balancer，並為您的 API 提供一個一致的 URL。但是，您可以將流量分割為後端上的各種微服務。

使用服務網格

AWS App Mesh是一種服務網狀，可協助您管理大量服務，並更好地控制流量在服務之間路由的方式。App Mesh 功能做為基本服務探索和負載平衡之間的中介。使用 App Mesh，應用程式不會直接互動，但也不會使用集中式負載平衡器。相反地，你的任務的每個副本都附有一個特使代理附屬。如需詳細資訊，請參閱「」什麼是AWS App Mesh中的AWS App Mesh使用者指南。



在前面的圖表中，每個任務都有一個特使代理邊車。此側車負責代理任務的所有輸入和輸出流量。應用程式網狀控制平面會使用AWS Cloud Map以取得可用服務的清單，以及特定工作的 IP 位址。然後，App Mesh 將配置提供給特使代理旁車。此組態包含可連接的可用容器清單。特使代理旁車也會針對每個目標進行健康檢查，以確保目標可用。

此方法提供服務探索的功能，並且易於受管理的負載平衡器。應用程式不會在程式碼中實作盡可能多的負載平衡邏輯，因為 Envoy Proxy 附屬處理該負載平衡。您可以將 Envoy Proxy 設定為偵測失敗並重試失敗的要求。此外，它也可以設定為使用 MTLS 來加密傳輸中的流量，並確保您的應用程式正在通訊至已驗證的目的地。

Envoy Proxy 和負載平衡器之間有一些不同。簡而言之，使用特使代理，您必須負責部署和管理您自己的特使代理附屬代理。特使代理附屬使用您分配給 Amazon ECS 任務的部分 CPU 和記憶體。這會增加工作資源耗用的一些額外負荷，並在需要時維護和更新 Proxy 的額外操作工作負載。

App Mesh 和特使代理允許在任務之間達到極低的延遲。這是因為 Envoy 代理運行並配給每個任務。只有一個實例可以實例網絡跳轉，一個特使代理和另一個特使代理之間。這意味著與使用負載平衡器相比，網絡開銷也較少。使用負載平衡器時，有兩個網路跳轉。第一個是從上游工作到負載平衡器，第二個是從負載平衡器到下游工作。

跨網路服務AWS帳戶和 VPC

如果您是擁有多個團隊和部門的組織的一員，您可能會將服務獨立部署到共用AWS帳戶或與多個個別AWS帳戶。無論您部署服務的方式為何，我們都建議您補充網路元件，以協助路由 VPC 之間的流量。為此，幾個AWS服務可用來補充您現有的網路元件。

- AWS Transit Gateway — 您應該先考慮此聯網服務。此服務可作為中央集線器，用於在 Amazon VPC 之間路由連線AWS帳戶和內部部署網路。如需詳細資訊，請參閱「」什麼是傳輸閘道？中的亞馬遜 VPC 交通閘道指南。
- Amazon VPC 和 VPN 支援 — 您可以使用此服務建立站台對站台 VPN 連線，以便將現場部署網路連線到 VPC。如需詳細資訊，請參閱「」什麼是AWS Site-to-Site VPN?中的AWS Site-to-Site VPN使用者指南。
- Amazon VPC — 您可以使用 Amazon VPC 對等互連來協助您連接多個 VPC，無論是在同一個帳戶或跨帳戶。如需詳細資訊，請參閱「」什麼是 VPC 互連？中的Amazon VPC 對等指南。
- 共用 VPC — 您可以跨多個AWS帳戶。如需詳細資訊，請參閱「」使用共用 VPC中的Amazon VPC使用者指南。

最佳化和故障診斷

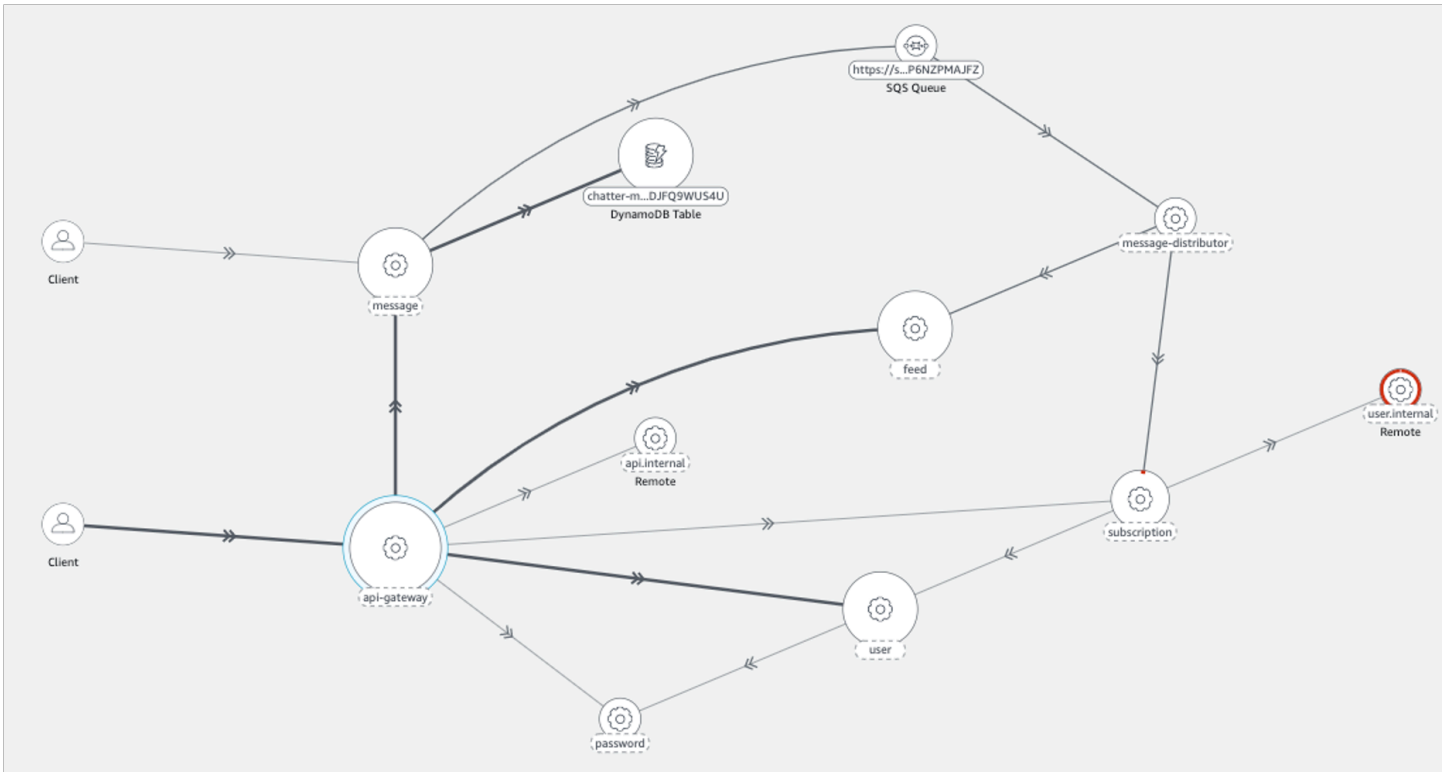
下列服務和功能可協助您深入瞭解網路和服務組態。您可以使用此資訊來排解網路問題，並進一步最佳化服務。

雲端容器洞見

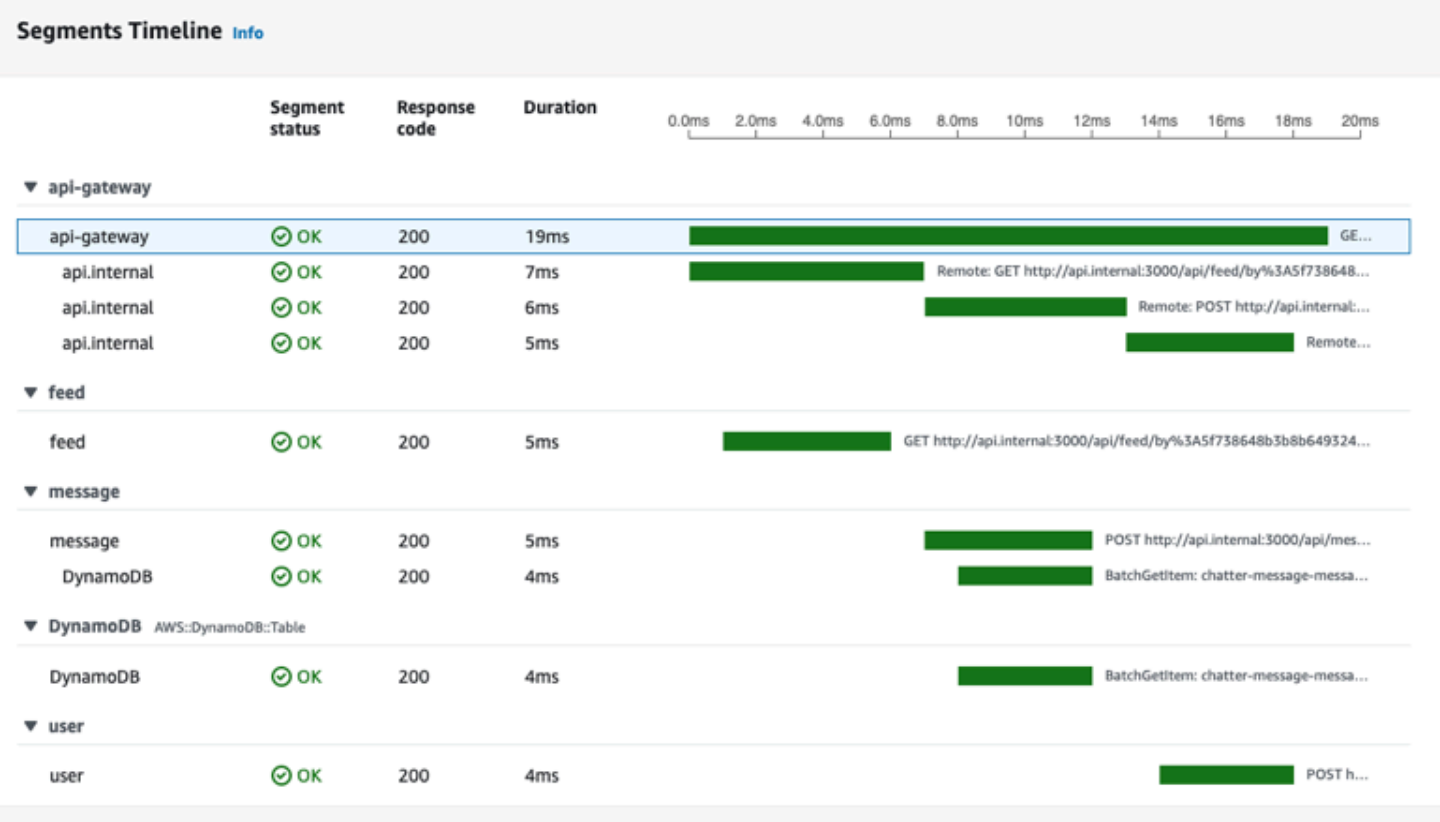
CloudWatch 容器洞見會從您的容器化應用程式和微型服務收集、彙總及總結指標和日誌。度量包括 CPU、記憶體、磁碟和網路等資源的使用率。它們可以在 CloudWatch 自動儀表板中使用。如需詳細資訊，請參閱「」在 Amazon ECS 上設定容器洞察中的Amazon CloudWatch 使用者指南。

AWS X-Ray

AWS X-Ray是一項追蹤服務，您可以用來收集應用程式所發出之網路要求的相關資訊。您可以使用 SDK 來測試應用程式，並擷取服務之間以及服務與AWS服務端點。如需詳細資訊，請參閱「」什麼是 AWS X-Ray中的AWS X-Ray開發人員指南。



您也可以AWS X-Ray您的服務如何互相聯網的圖表。或者，您也可以使用它們來探索每個服務對服務連結執行方式的彙總統計資料。最後，您可以深入探索任何特定交易，以查看代表網路呼叫的區段如何與該特定交易相關聯。



您可以使用這些功能來識別是否有網路瓶頸，或是網路中的特定服務未如預期般執行。

VPC 流程日誌

您可以使用 Amazon VPC 流程日誌來分析網路效能和偵錯連線問題。啟用 VPC 流量記錄後，您可以擷取 VPC 中所有連線的記錄。其中包括與 Elastic Load Balancing、Amazon RDS、NAT 閘道和其他金鑰相關聯的網路介面連線AWS您可能正在使用的服務。如需詳細資訊，請參閱「」VPC 流程日誌中的Amazon VPC 使用者指南。

網路調整秘訣

您可以微調幾個設定，以改善您的網路。

無檔案 Ulimit

如果您希望您的應用程序具有高流量並處理許多並發連接，則應該考慮允許的文件數的系統配額。當打開了很多網絡套接字時，每個套接字都必須用文件描述符表示。如果您的文件描述符配額太低，它會限制您的網絡套接字。這會導致連線失敗或錯誤。您可以針對 Amazon ECS 任務定義中的檔案數量更新容器特定配額。如果您在 Amazon EC2 上運行（而不是AWS Fargate)，那麼您可能還需要在基礎 Amazon EC2 執行個體上調整這些配額。

Syystl

另一個可調整設定類別是`sysctl`網路設定。您應該參考您所選擇的 Linux 發行版的特定設定。許多這些設定會調整讀取和寫入緩衝區的大小。在某些情況下執行具有大量容器的大型 Amazon EC2 執行個體時，這可以幫助您。

最佳實務-自動擴充和容量管理

Amazon ECS 用於執行各種規模的容器化應用程式工作負載。這包括極端的最小測試環境，以及以全球規模運作的大型生產環境。

使用亞馬遜 ECS，就像所有AWS服務，只需按實際用量付費。適當架構時，您可以讓應用程式在需要時僅消耗所需的資源，以節省成本。本最佳實務指南說明如何以符合服務層級目標的方式執行 Amazon ECS 工作負載，同時仍以符合成本效益的方式運作。

主題

決定任務大小

在 Amazon ECS 上部署容器時，最重要的選擇之一就是您的容器和任務大小。您的容器和工作大小對於擴充和容量規劃都是必不可少的。在 Amazon ECS 中，有兩個資源指標用於容量：CPU 和記憶體。CPU 的測量單位為完整 vCPU 的 1/1024 (其中 1024 個單位等於 1 個整個 vCPU)。記憶體以 MB 為單位測量。在您的作業定義中，您可以宣告資源保留與限制。

當您宣告保留區時，您會宣告工作所需的最小資源量。您的工作至少會收到所要求的資源量。您的應用程式可能會使用比您宣告的保留區更多的 CPU 或記憶體。但是，這受到您也聲明的任何限制。使用大於保留金額稱為大量批次。在 Amazon ECS 中，保證保留。例如，如果您使用 Amazon EC2 執行個體提供容量，則 Amazon ECS 不會在無法履行保留的執行個體上放置任務。

限制是您的容器或任務可以使用的 CPU 單元或內存量上限。任何嘗試使用更多的 CPU 超過此模擬會導致節流。任何嘗試使用更多內存都會導致容器被停止。

選擇這些值可能很具挑戰性。這是因為最適合您應用程式的值在很大程度上取決於應用程式的資源需求。負載測試您的應用程序是成功的資源需求規劃和更好地了解您的應用程序的要求的關鍵。

無狀態的應用程式

對於水平調整的無狀態應用程式 (例如負載平衡器後面的應用程式)，建議您先決定應用程式在提供要求時所耗用的記憶體數量。若要執行此操作，您可以使用傳統工具，例如ps或top或監控解決方案，例如 CloudWatch 容器深入解析。

決定 CPU 保留區時，請考慮如何調整應用程式以符合業務需求。您可以使用較小的 CPU 保留區，例如 256 個 CPU 單位 (或 1/4 vCPU)，以細微的方式向外擴充，將成本降至最低。但是，它們的擴展速度可能不夠快，無法滿足顯著的需求峰值。您可以使用較大的 CPU 保留區來更快速地進行擴充和擴充，因此更快地符合需求尖峰。不過，較大的 CPU 保留費用較高。

其他應用

對於無法水平擴充的應用程式 (例如單一工作者或資料庫伺服器)，可用容量和成本代表您最重要的考量。您應該根據負載測試指示您需要服務流量以符合服務層級目標，選擇記憶體和 CPU 的數量。Amazon ECS 可確保應用程式放置在具有足夠容量的主機上。

設定服務自動擴展

Amazon ECS 服務是任務的受管集合。每個服務都有相關的任務定義、所需的任務計數，以及選擇性的放置策略。Amazon ECS 服務自動擴展是透過 Application Auto Scaling 服務實作的。Application Auto Scaling 會使用 CloudWatch 度量做為調整度量的來源。它也會使用 CloudWatch 警示來設定何時向內或向外擴展服務的臨界值。您可以透過設定測量結果目標 (稱為目標追蹤擴展，或藉由指定臨界值 (稱為步驟擴展。設定「Application Auto Scaling 比例」之後，它會持續計算服務所需的適當工作計數。它也會在需要的任務計數應該變更時通知 Amazon ECS，方法是向外擴展或擴展。

若要有效地使用服務自動調整，您必須選擇適當的調整度量。我們在下節討論如何選擇指標。

描述您的應用程式

正確擴展應用程序需要知道應該擴展應用程序的條件，以及何時應該擴展應用程序。基本上，如果預測需求超出容量，應該向外擴充應用程式。相反地，應用程式可以在資源超過需求時進行調整，以節省成本。

識別使用率測量結果

若要有效擴充，識別指出使用率或飽和度的度量非常重要。此度量必須具有下列屬性，才能調整比例。

- 測量結果必須與需求相關。當資源保持穩定，但需求變更時，測量結果值也必須變更。當需求增加或減少時，量度應該增加或減少。

- 度量值必須按容量比例調整。當需求保持固定時，新增更多資源必須導致測量結果值的比例變更。因此，工作數目加倍應該會導致量度減少 50%。

識別使用率度量的最佳方式是透過預先生產環境 (例如臨時環境) 中的負載測試。商業和開放原始碼負載測試解決方案已廣泛提供。這些解決方案通常可以產生合成負載或模擬實際使用者流量。

若要開始負載測試的程序，您應該先為應用程式的使用率度量建置儀表板。這些度量包括 CPU 使用率、記憶體使用率、I/O 作業、I/O 佇列深度以及網路輸送量。您可以透過 CloudWatch 容器深入解析等服務收集這些指標。或者，通過使用 Prometheus 的亞馬遜託管服務以及格拉法納的亞馬遜託管服務來執行此操作。在此程序期間，請確定您收集並繪製應用程式回應時間或工作完成率的度量。

負載測試時，請從小量的請求或工作插入率開始。保持這個速率穩定數分鐘，以便您的應用程序熱身。然後, 慢慢地增加速率並保持穩定幾分鐘. 重複此週期，每次都會增加速率，直到應用程式的回應或完成時間太慢，無法滿足您的服務層級目標 (SLO) 為止。

負載測試時，請檢查每個使用率度量。隨著負載增加的度量是最佳使用率度量的最佳候選人。

接下來，識別達到飽和度的資源。同時，也請檢查使用率度量，以查看哪一個在高層級平面化。或者，檢查哪一個到達峰值，然後首先當機您的應用程式。例如，當您增加負載時，如果 CPU 使用率從 0% 增加到 70-80%，那麼在添加更多負載後保持在該負載，那麼可以肯定地說 CPU 已飽和。根據 CPU 架構，它可能永遠不會達到 100%。例如，假設記憶體使用率會隨著您增加負載而增加，然後當應用程式達到任務或 Amazon EC2 執行個體記憶體限制時，應用程式會突然當機。在此情況下，很可能是記憶體已完全耗用的情況。您的應用程序可能會消耗多個資源。因此，請選擇代表先耗盡之資源的度量。

最後，請在任務數量或 Amazon EC2 執行個體加倍後再次嘗試載入測試。假設關鍵度量增加或減少的速率與之前相同。如果是這種情況，則度量與容量成正比。這是一個很好的自動調整使用率度量。

現在考慮一下這個假設的情況。假設您載入測試應用程式，並發現 CPU 使用率最終達到 80%，每秒 100 個要求。當增加更多負載時，它不會使 CPU 使用率再提高。但是，它確實使您的應用程序響應更慢。然後，您再次執行負載測試，將工作數目加倍，但將速率保持在先前的尖峰值。如果您發現平均 CPU 使用率降至約 40%，則平均 CPU 使用率是調整量度的適當候選項。另一方面，如果增加工作數量後 CPU 使用率維持在 80%，則平均 CPU 使用率不是一個很好的縮放度量。在這種情況下，需要更多的研究來找到一個合適的指標。

一般應用程式模型和縮放屬性

各種軟件運行在AWS。許多工作負載都是本地生產，而其他工作負載則是以熱門的開放原始碼軟體為基礎 無論它們的起源在哪裡，我們都觀察到了一些常見的服務設計模式。如何有效地擴展取決於在很大程度上的模式。

高效率的 CPU 綁定伺服器

高效率的 CPU 繫結伺服器除了 CPU 和網路輸送量之外，幾乎不會使用任何資源。每個請求可以單獨由應用程序處理。請求不依賴於其他服務，如數據庫。該應用程序可以處理數十萬個並發請求，並且可以有效地利用多個 CPU 來執行此操作。每個要求是由具有低記憶體負荷的專用執行緒服務，或是有服務要求的每個 CPU 上執行的非同步事件迴圈。應用程序的每個副本同樣能夠處理請求。在 CPU 之前可能耗盡的唯一資源是網路頻寬。在 CPU 限制服務中，記憶體使用率 (即使在尖峰輸送量) 是可用資源的一小部分。

這種類型的應用程式適用於基於 CPU 的自動縮放。該應用程序在擴展方面享有最大的靈活性。它可以通過向其提供更大的 Amazon EC2 執行個體或 Fargate vCPUs 來垂直擴展。而且，它也可以透過新增更多複本來水平縮放。新增更多複本或增加執行個體大小的兩倍，相對於容量的平均 CPU 使用率減半。

如果您為此應用程式使用 Amazon EC2 容量，請考慮將其放置在運算優化的執行個體上，例如`c5`或`c6g`家庭。

高效率的記憶體繫結伺服器

高效率的記憶體繫結伺服器會配置大量的記憶體每個要求。在最大並行性 (但不一定是輸送量) 時，記憶體會在 CPU 資源耗盡之前耗盡。當請求結束時，與請求相關聯的內存被釋放。只要有可用的記憶體，就可以接受其他要求。

這種類型的應用程序適用於基於內存的自動擴展。該應用程序在擴展方面享有最大的靈活性。它可以通過向其提供更大的 Amazon EC2 或 Fargate 內存資源來垂直擴展。而且，它也可以透過新增更多複本來水平縮放。新增更多複本或使執行個體大小加倍，可能會將相對於容量的平均記憶體使用率減半。

如果您為此應用程式使用 Amazon EC2 容量，請考慮將其放置在記憶體優化執行個體，例如`r5`或`r6g`家庭。

某些記憶體繫結的應用程式不會釋放與要求結束時相關聯的記憶體，因此減少並行性並不會導致使用的記憶體減少。為此，我們不建議您使用記憶體為基礎的擴展。

以工作者為基礎的伺服器

以工作者為基礎的伺服器會逐一處理每個個別工作者執行緒的一個要求。工作者執行緒可以是輕量型執行緒，例如 POSIX 執行緒。它們也可以是重量較重的執行緒，例如 UNIX 處理序。無論它們是哪個線程，總是有應用程序可以支持的最大並發性。通常，並發限制按比例設置為可用的內存資源。如果達到並行限制，則會將其他要求放入待處理項目佇列中。如果積存佇列溢位，則會立即拒絕額外的連入要求。適合這種模式的常見應用程序包括 Apache Web 服務器和 Gunicorn。

要求並行性通常是調整此應用程式的最佳度量。因為每個複本都有並行限制，所以在達到平均限制之前向外擴充是很重要的。

獲取請求並發度量的最佳方式是讓您的應用程序將其報告給 CloudWatch。應用程式的每個複本都可以以高頻率將並行要求的數目發佈為自訂度量。我們建議將頻率設定為每分鐘至少一次。收集數個報告後，您可以使用平均並行作為縮放度量。此度量的計算方式是取得總並行處理並行處理並除以複本數目。例如，如果並行總數為 1000，而複本數為 10，則平均並行性為 100。

如果您的應用程式位於 Application Load Balancer 後面，您也可以使用`ActiveConnectionCount`度量做為調整度量中的因素。所以此`ActiveConnectionCount`度量必須除以複本數目，才能取得平均值。平均值必須用於縮放，而不是原始計數值。

若要讓這項設計發揮最佳效果，回應延遲的標準差應該在低要求率下很小。我們建議在低需求期間，大部分的要求都會在短時間內回覆，而且不會有太多需要比平均回應時間長得多的要求。平均回應時間應該接近 95 個百分位數的回應時間。否則，可能會發生佇列溢位。這會導致錯誤。我們建議您在必要時提供其他複本，以減少溢位的風險。

等候伺服器

等待服務器對每個請求進行一些處理，但它高度依賴於一個或多個下游服務才能運作。容器應用程序通常會大量使用下游服務，如數據庫和其他 API 服務。這些服務可能需要一些時間才能回應，特別是在高容量或高並行性案例中。這是因為這些應用程式在可用記憶體方面往往會使用較少的 CPU 資源及其最大並行性。

等待服務適用於記憶體繫結的伺服器模式或以工作者為基礎的伺服器模式，視應用程式的設計方式而定。如果應用程式的並行性僅受記憶體限制，則應使用平均記憶體使用率作為調整度量。如果應用程式的並行處理是以背景工作限制為基礎，則應使用平均並行處理作為縮放度量。

Java 伺服器

如果您的基於 Java 的服務器是 CPU 綁定的，並且與 CPU 資源成比例地擴展，那麼它可能適合高效的 CPU 綁定服務器模式。如果是這種情況，平均 CPU 使用率可能適合做為調整度量。然而，許多 Java 應用程序並不受 CPU 限制，因此它們具有挑戰性。

為獲得最佳效能，建議您盡可能將記憶體分配給 Java 虛擬機器 (JVM)。最新版本的 JVM（包括 Java 8 更新 191 或更高版本）會自動設置堆大小盡可能大以適應容器。這意味著，在 Java 中，內存利用率很少與應用程序利用成正比。隨著要求速率和並行性的增加，記憶體使用率保持不變。因此，我們不建議根據記憶體使用率調整 Java 架構的伺服器。相反地，我們通常建議調整 CPU 使用率。

在某些情況下，基於 Java 的服務器會在耗盡 CPU 之前遇到堆耗盡。如果您的應用程序容易在高並發情況下堆積耗盡，那麼平均連接是最佳的縮放度量。如果您的應用程式容易在高輸送量下堆積耗盡，則平均要求率是最佳的調整度量。

使用其他垃圾收集執行階段的伺服器

許多服務器應用程序是基於執行垃圾收集（如 .NET 和 Ruby）的運行時。這些伺服器應用程式可能符合先前所述的其中一種模式。然而，與 Java 一樣，我們不建議根據內存擴展這些應用程序，因為它們觀察到的平均內存使用率通常與吞吐量或並發無關。

對於這些應用程式，如果應用程式受 CPU 限制，我們建議您調整 CPU 使用率。否則，我們建議您根據負載測試結果，依平均輸送量或平均並行量進行調整。

Job 處理器

許多工作負載涉及非同步工作處理。它們包括不會即時接收要求的應用程式，而是訂閱工作佇列以接收工作。對於這些類型的應用程式，適當的調整度量幾乎總是佇列深度。佇列成長表示待處理工作超出處理能力，而空白佇列表示有比工作要做更多的容量。

AWS傳訊服務 (例如 Amazon SQS 和 Amazon Kinesis Data Streams) 提供可用於擴展的 CloudWatch 指標。對於 Amazon SQS，`ApproximateNumberOfMessagesVisible`是最佳度量。對於 Kinesis Data Streams，請考慮使用`MillisBehindLatest`指標，由 Kinesis 用戶端程式庫 (KCL) 發行。此度量應該在所有取用者之間平均值，然後再將其用於調整。

容量與可用

應用程式可用性對於提供無錯誤的體驗和最小化應用程式延遲至關重要。可用性取決於具有可存取且具有足夠容量來滿足需求的資源。AWS提供數種機制來管理可用性。對於 Amazon ECS 託管的應用程式，這些應用程式包括自動擴展和可用區域 (AZ)。自動調整會根據您定義的度量來管理工作或執行個體的數目，而可用區域則可讓您將應用程式託管在隔離但地理位置相當近的位置。

與任務規模一樣，容量和可用性存在某些權衡，您必須考慮。理想情況下，容量將完全符合需求。總是有足夠的容量來服務要求和處理工作，以滿足服務層級目標 (SLO)，包括低延遲和錯誤率。容量永遠不會太高，導致成本過高；也不會太低，導致高延遲和錯誤率。

自動調整是一個潛在的過程。首先，必須將即時指標傳遞至 CloudWatch。然後，它們需要進行彙總以進行分析，這可能需要長達數分鐘的時間，視量度的粒度而定。CloudWatch 會將指標與警示臨界值進行比較，以識別資源短缺或過量。若要防止不穩定，請設定警報，以便在鬧鐘響起前超過設定的臨界值數分鐘。它也需要一些時間來佈建新的任務和終止不再需要的任務。

由於系統中所描述的這些潛在延遲，因此您必須透過過過度佈建來維持一些預留空間。這樣做可以幫助容納短期突發的需求。這也有助於您的應用程式在不達到飽和度的情況下服務額外的請求。最好的做法是，您可以將縮放目標設定在使用率的 60-80% 之間。這有助於您的應用程式更好地處理額外需求的高載，而額外的容量仍在佈建過程中。

我們建議您過度佈建的另一個原因是，您可以快速回應可用區域失敗。AWS建議您從多個可用區域提供生產工作負載。這是因為，如果發生可用區域失敗，您在剩餘可用區域中執行的工作仍然可以滿足需求。如果您的應用程式在兩個可用區域中執行，則需要將一般工作計數加倍。如此一來，您可以在任何潛在故障期間立即提供容量。如果您的應用程式在三個可用區域中執行，建議您執行一般工作計數的 1.5 倍。也就是說，為普通服務所需的每兩個運行三個任務。

擴展速度最大化

自動調整是一個反應過程，需要時間才能生效。不過，有一些方法可協助將擴充所需的時間降至最低。

最小化影像大小。 較大的影像需要較長的時間才能從影像儲存庫下載並解壓縮。因此，保持較小的影像大小可減少容器啟動所需的時間。若要減少影像大小，您可以遵循以下特定建議：

- 如果您可以構建靜態二進製文件或使用 Golang，請構建您的圖像FROM從頭開始，並在結果圖像中僅包含您的二進制應用程序。
- 使用來自上游發行版廠商 (例如 Amazon Linux 或 Ubuntu) 的最小化基礎映像檔。
- 不要在最終圖像中包含任何構建工件。使用多階段構建可以幫助解決這個問題。
- 精簡RUN階段，盡可能。每個RUN階段會建立新的影像圖層，導致下載圖層的額外往返。單一RUN階段，該階段具有多個由&&的圖層少於具有多個RUN階段。
- 如果您想要在最終影像中包含資料 (例如 ML 推論資料)，請僅包含啟動和開始服務流量所需的資料。如果您在不影響服務的情況下從 Amazon S3 或其他儲存中隨選取資料，請將資料存放在這些地方。

讓您的影像保持近距離。 網路延遲越高，下載映像所需的時間越長。將您的圖像託管在相同的AWS您的工作負載所在的區域。Amazon ECR 是一個高效能的映像儲存庫，可在 Amazon ECS 提供的每個區域使用。避免在網際網路或 VPN 連結下載容器映像。在同一區域託管您的影像，可提高整體可靠性。它降低了不同區域中的網路連線問題和可用性問題的風險。或者，您也可以實作 Amazon ECR 跨區域複寫來協助解決這個問題。

降低負載平衡器運作狀態檢查臨界值。 負載平衡器會先執行健全狀況檢查，再將流量傳送至應用程式。目標群組的預設健全狀況檢查組態可能需要 90 秒或更長的時間。在此期間，它會檢查健康狀態和接收要求。降低運作狀態檢查間隔和臨界值計數可讓您的應用程式更快接受流量，並減少其他工作的負載。

考慮冷啟動效能。 某些應用程序使用運行時，例如 Java 執行剛剛在時間（JIT）編譯。編譯過程至少在啟動時可以顯示應用程序性能。解決方法是以不會造成冷啟動效能損失的語言重寫工作負載的延遲關鍵部分。

使用步驟擴展政策，而不是目標追蹤擴展政策。 Amazon ECS 任務有數個 Application Auto Scaling 選項。目標追蹤是最容易使用的模式。有了它，您只需要設定測量結果的目標值，例如 CPU 平均使用率。然後，自動縮放器會自動管理達到該值所需的任務數量。不過，我們建議您改用步驟縮放比例，以便您可以更快速地回應需求的變更。使用步驟縮放，您可以定義縮放測量結果的特定臨界值，以及超過臨界值時要新增或移除的工作數目。此外，更重要的是，您可以將臨界值警報違反的時間減至最少，以便快速對需求變化做出反應。如需詳細資訊，請參閱「」Service Auto Scaling中的Amazon Elastic Container Service 開發者指南。

如果您使用 Amazon EC2 執行個體提供叢集容量，請考慮下列建議：

使用較大的 Amazon EC2 執行個體和較快的 Amazon EBS 磁碟區。 您可以使用較大的 Amazon EC2 執行個體和較快的 Amazon EBS 磁碟區來提高映像下載和準備速度。在指定的 Amazon EC2 執行個體系列中，網路和 Amazon EBS 最大輸送量會隨著執行個體大小的增加而增加 (例如，來自`m5.xlarge`至`m5.2xlarge`。此外，您也可以自訂 Amazon EBS 磁碟區以增加其輸送量和 IOPS。例如，如果您使用的是`gp2`磁碟區，請使用提供更多基準輸送量的較大磁碟區。如果您使用的是`gp3`磁碟區，請在建立磁碟區時指定輸送量和 IOPS。

針對在 Amazon EC2 執行個體上執行的任務，使用橋接網路模式。 使用的任務`bridge`網路模式 Amazon EC2 啟動速度比使用`awsvpc`網路模式。時機`awsvpc`網路模式時，Amazon ECS 會在啟動任務之前將 elastic network interface (ENI) 附加至執行個體。這會引入額外的延遲。儘管如此，使用橋接網絡有幾個權衡。這些工作不會取得自己的安全性群組，而且負載平衡有一些含義。如需詳細資訊，請參閱「」負載平衡器目標群組中的Elastic Load Balancing 使用者。

處理需求衝擊

某些應用程式會遇到突然大幅衝擊的需求。發生這種情況的原因有很多：新聞事件、大型銷售、媒體事件或其他事件，這些事件會發生病毒，導致流量在非常短的時間內快速且顯著增加。如果未計劃，這可能會導致需求快速超出可用資源。

處理需求衝擊的最佳方式是預測他們並相應地計劃。由於自動調整可能需要一些時間，因此建議您在需求衝擊開始之前，先向外擴充應用程式。為了獲得最佳效果，我們建議您制定商務計劃，這些計劃涉及使用共用行事曆的團隊之間的緊密協同合作。規劃活動的團隊應該事先與負責申請的團隊緊密合作。這讓該團隊有足夠的時間來制定明確的排程計劃。他們可以安排容量，以便在活動前向外擴充，並在活動後擴充。如需詳細資訊，請參閱「」排程擴展功能中的Application Auto Scaling 使用者指南。

如果您有企業 Support 方案，請務必同時與您的技術客戶經理 (TAM) 合作。您的 TAM 可以驗證您的服務配額，並確保在事件開始之前提出任何必要的配額。如此一來，您就不會意外達到任何服務配額。它們也可以通過預熱服務（例如負載平衡器）來幫助您確保事件順利進行。

處理未定期的需求衝擊是一個更困難的問題。如果振幅足夠大的非排程衝擊，可能會迅速導致需求超出容量。它也可以超越自動調整反應的能力。準備意外衝擊的最佳方式是過度佈建資源。您必須擁有足夠的資源，才能隨時處理最大預期的流量需求。

預期未排定的需求衝擊時，維持最大容量可能會成本很高。若要減輕成本影響，請找出可預測即將發生大量需求衝擊的領先指標指標或事件。如果度量或事件可靠地提供重要的預先通知，當事件發生或度量超過您設定的特定臨界值時，立即開始向外延展程序。

如果您的應用程式容易發生突然的需求衝擊，請考慮在應用程式中新增高效能模式，以犧牲非關鍵功能，但保留客戶的重要功能。例如，假設您的應用程式可以從產生昂貴的自訂回應切換到服務靜態回應頁面。在此案例中，您可以大幅增加輸送量，完全不需要調整應用程式。

最後，您可以考慮打破整體服務，以更好地處理需求衝擊。如果您的應用程序是執行昂貴且擴展速度緩慢的單一服務，您可能可以提取或重寫性能關鍵部分，並將其作為單獨的服務運行。然後，這些新服務可以獨立於較不重要的元件進行擴充。彈性地將效能關鍵功能與應用程式的其他部分分開擴充，既可減少增加容量所需的時間，也有助於節省成本。

叢集容量

在本主題稍早，我們討論了如何使用縮放度量來向外擴充您的複本帳戶。您的工作也需要在資源上執行，包括 CPU 和記憶體資源。這又涉及到容量的主題。在 Amazon ECS 中，容量由兩個主要提供者提供：AWS Fargate和 Amazon EC2。

您可以透過多種方式為 Amazon ECS 叢集提供容量。例如，您可以啟動 Amazon EC2 執行個體，並在啟動時使用 Amazon ECS 容器代理程式向叢集註冊這些執行個體。不過，此方法可能很具挑戰性，因為您需要自行管理縮放比例。因此，建議您使用 Amazon ECS 容量提供者。他們為您管理資源擴展。有三種容量提供者：Amazon EC2，Fargate 和 Fargate 方點。

Fargate 和 Fargate 競價型容量供應商可為您處理 Fargate 任務的生命週期。Fargate 提供隨需容量，而 Fargate 現貨提供現貨容量。當任務啟動時，ECS 會為您提供 Fargate 資源。此 Fargate 資源附帶直接對應於您在任務定義中聲明的任務級限制的內存和 CPU 單元。每個任務都會收到自己的 Fargate 資源，在任務和計算 iresources 之間產生 1：1 的關係。

在遠方現場執行的任務可能會中斷。兩分鐘警告後會發生中斷 這些發生在需求繁重的時期。Fargate 競價型最適合容忍中斷的工作負載，例如批次作業、開發或臨時環境。它們也適用於高可用性和低延遲不是必要的任何其他場景。

您可以與 Fargate 隨選任務一起執行遠方之門現場任務。透過將它們一起使用，您可以以較低的成本獲得佈建「高載」容量。

ECS 也可以為您的任務管理 Amazon EC2 執行個體容量。每個 Amazon EC2 容量提供者都與您指定的 Amazon EC2 Auto Scaling 群組相關聯。當您使用 Amazon EC2 容量提供者時，ECS 叢集 Auto Scaling 會維持 Amazon EC2 自動擴展群組的大小，以確保可以放置所有排定的任務。

叢集容量最佳實務

將預留空間新增至您的服務，而不是容量提供者。 Amazon EC2 容量提供者提供目標容量值。如果將值設定為低於 100%，則 ECS 佈建的 Amazon EC2 執行個體數量超過適應您的任務所需的數量。準備好接受任務的多個 Amazon EC2 執行個體非常有幫助。但是，當您使用 Amazon Virtual Private Cloud 時，啟動新任務需要額外的時間才能下載映像並附加網路介面。這種增加的延遲可能會損害您的利潤。

因此，建議您執行下列動作。而不是減少容量提供者的目標容量，只要修改目標追蹤調整比例度量或服務的步驟調整閾值，即可增加服務中的複本數目。如需相關擴展政策的詳細資訊，請參閱。目標追蹤擴展政策或步驟擴展政策中的Amazon Elastic Container Service 開發者指南。Amazon EC2 容量提供者透過向 Auto Scaling 群組新增額外執行個體，佈建額外任務所需的容量。這有助於確保計算和應用程式資源在您需要時都可以使用。例如，它可以讓 ECS 服務中的任務數量增加一倍，以滿足立即 100% 的需求。

選擇 Fargate 任務大小

如果您在AWS Fargate，您必須在您任務定義中聲明任務 CPU 和記憶體限制。ECS 使用這些限制來決定要在其上執行工作的 Fargate 執行個體類型。您決定的限制必須大於或等於您宣告的任何保留。在大多數情況下，您可以將它們設定為在任務定義中宣告之每個容器的保留區總和。然後，也將數字四捨五入到最接近的 Fargate 實例大小。如需有關可用大小的詳細資訊，請參閱。任務 CPU 和記憶體中的Amazon Elastic Container Service 開發者指南。

選擇 Amazon EC2 執行個體類型

如果您使用 Amazon EC2 為 ECS 叢集提供容量，則可以從多種執行個體類型中選擇。所有 Amazon EC2 執行個體類型和系列都與 ECS 相容。

若要判斷您可以使用哪些執行個體類型，請先消除不符合應用程式特定需求的執行個體類型或執行個體系列。例如，如果您的應用程式需要 GPU，您可以排除任何沒有 GPU 的執行個體類型。但是，您也應該考慮其他要求。例如，考慮 CPU 架構、網路輸送量，以及執行個體儲存是否需求。接下來，檢查

每個執行個體類型所提供的 CPU 和記憶體數量。一般而言，CPU 和記憶體必須足夠大，才能容納至少一個您想要執行的工作複本。

您可以從與應用程式相容的執行個體類型中進行選擇。若執行個體較大，您可以同時啟動更多任務。而且，使用較小的執行個體，您可以更細微的方式向外擴充以節省成本。您不需要選擇適合叢集中所有應用程式的單一 Amazon EC2 執行個體類型。相反地，您可以建立多個「Auto Scaling 比例群組」。每個群組可以有不同的執行個體類型。然後，您可以為這些群組中的每一個建立 Amazon EC2 容量提供者。最後，在服務和工作的容量提供者策略中，您可以選取最適合其需求的容量提供者。

使用 Amazon EC2 Spot 和遠方 Spot

競價型容量可大幅節省隨需執行個體的成本。競價型容量是多餘的容量，價格遠低於隨需或預留容量。競價型容量適用於批次處理和機器學習工作負載，以及開發和預備環境。更一般地說，它適用於容許暫時停機的任何工作負載。

請瞭解下列後果，因為競價型容量可能無法一直使用。

- 首先，在需求極高的期間，競價型容量可能無法使用。這可能會導致 Fargate 競價型任務和 Amazon EC2 競價型執行個體的啟動延遲。在這些事件中，ECS 服務會重試啟動任務，而 Amazon EC2 Auto Scaling 群組也會重試啟動執行個體，直到所需容量可用為止。Fargate 和 Amazon EC2 不會以隨需容量取代競價型容量。
- 其次，當整體容量需求增加時，競價型執行個體和任務可能會在只有兩分鐘警告的情況下終止。傳送警告之後，如有必要，工作應該會在執行個體完全終止之前開始有序關機。這有助於減少錯誤的可能性。如需正常關機的詳細資訊，請參閱。使用 ECS 進行正常關機。

若要協助將競價型容量短缺降至最低，請考慮下列建議：

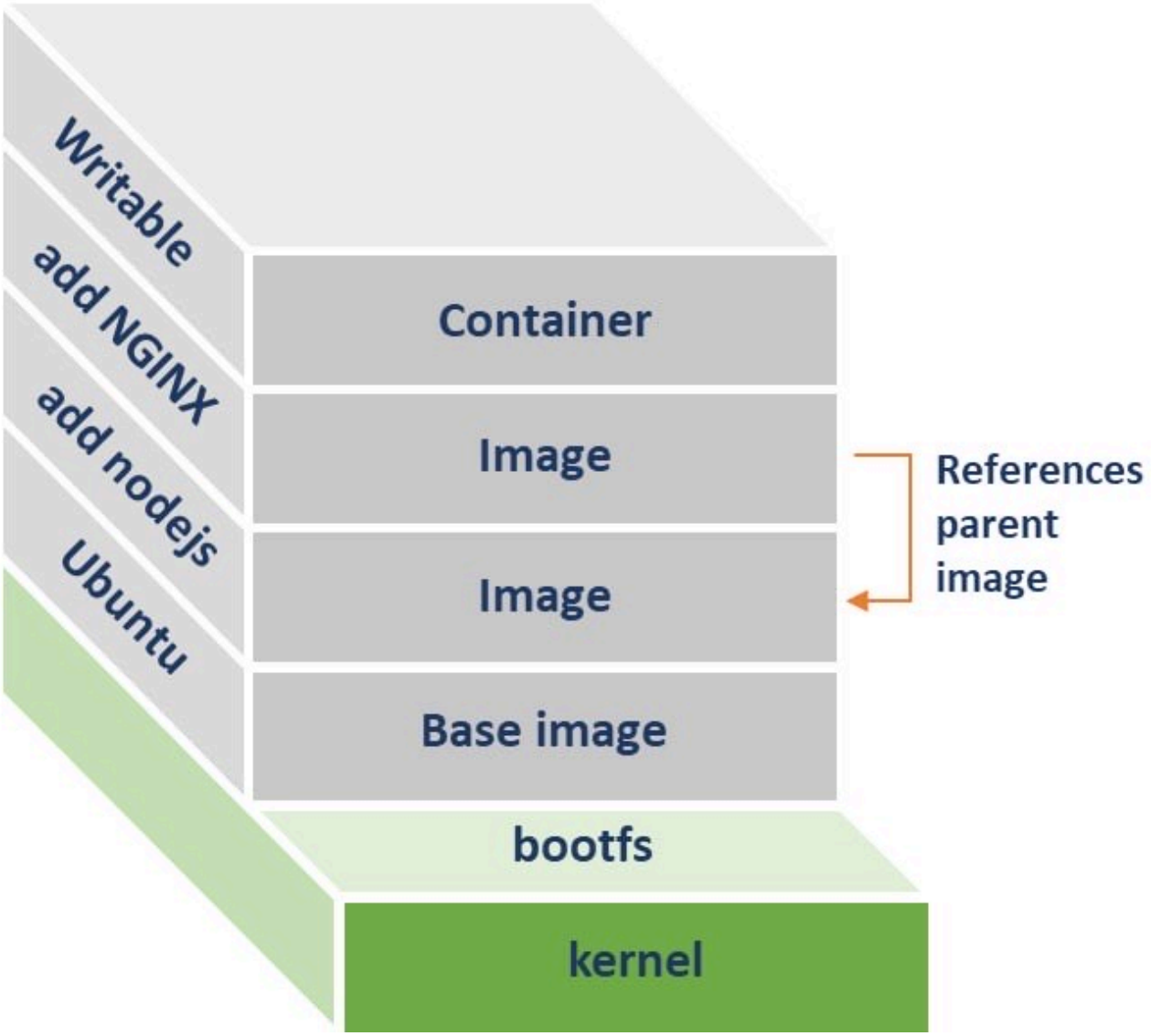
- 使用多個區域和可用區域。 競價型容量因區域和可用區域而異。您可以透過在多個區域和可用區域執行工作負載來改善競價型可用性。如果可能，請在執行工作和執行個體的區域中的所有可用區域中指定子網路。
- 使用多個 Amazon EC2 執行個體類型。 當您將混合執行個體政策與 Amazon EC2 Auto Scaling 搭配使用時，會在自動擴展群組中啟動多個執行個體類型。這可確保在需要時可以滿足競價型容量的請求。為了最大化可靠性並將複雜性降至最低，請在混合執行個體原則中使用大致相同 CPU 和記憶體數量的執行個體類型。這些例證可以來自不同層代，也可以是相同基礎例證類型的變體。請注意，這些功能可能會隨附您可能不需要的其他功能。這樣一個列表的例子可以包括 m4.5 大，m5 大，m5 大，m5n 大，m5n 大，m5n 大，m5dn 大和 m5ad.m。如需詳細資訊，請參閱「」具有多個執行個體類型和購買選項的 Auto Scaling 群中的Amazon EC2 Auto Scaling 用戶指南。

- 使用容量最佳化的競價型配置策略。 使用 Amazon EC2 競價型，您可以在容量和成本最佳化的配置策略之間進行選擇。如果您在啟動新執行個體時選擇了容量優化策略，Amazon EC2 競價會在所選可用區域中選擇具有最高可用性的執行個體類型。這有助於減少執行個體在啟動後不久終止的可能性。

最佳做法-永久性儲存

您可以使用 Amazon ECS 來大規模執行可設定狀態的容器化應用程式，方法是使用AWS儲存服務，例如 Amazon EFS BS、Amazon EBS 或適用於 Windows 檔案伺服器的 Amazon FSX，可為本質上的暫時容器提供資料持久性。術語資料持久性意味著數據本身超過了創建它的過程。資料持久性AWS是透過結合運算和儲存服務來達成的。與 Amazon EC2 類似，您也可以使用 Amazon ECS 將容器化應用程式的生命週期與使用和產生的資料分離出來。使用AWS儲存服務，即使任務終止後，Amazon ECS 任務也可以保留資料。

默認情況下，容器不會保存它們產生的數據。當容器被終止時，它寫入其可寫入層的數據將與容器一起銷毀。這使得容器適用於不需要在本地存儲數據的無狀態應用程序。需要數據持久性的容器化應用程序需要一個存儲後端，當應用程序的容器終止時，不會被銷毀。



容器影像是以一系列圖層建立的。每個圖層都代表 Docker 檔案中建立影像的指令。除了容器之外，每個圖層都是唯讀的。也就是說，當您建立容器時，會在底層層上加入可寫入的層。容器建立、刪除或修改的任何檔案都會寫入可寫入層。當容器終止時，也會同時刪除可寫圖層。使用相同影像的新容器具有自己的可寫入層。此圖層不包含任何變更。因此，容器的數據必須始終存儲在容器可寫層之外。

使用 Amazon ECS，您可以使用磁碟區執行可設定狀態的容器。Amazon ECS 本機與 Amazon EFS 整合，並使用與 Amazon EBS 整合的磁碟區。對於 Windows 容器，亞馬遜 ECS 與適用於 Windows 文件服務器的亞馬遜 FSX 整合以提供持久性存儲。

主題

- 為您的容器選擇正確的儲存類型

- Amazon EFS 磁碟區
- Docker 磁碟區
- Amazon FSx for Windows File Server

為您的容器選擇正確的儲存類型

在 Amazon ECS 叢集中執行的應用程式可以使用各種AWS儲存區服務和第三方產品，以便為可設定狀態的工作負載提供永續性 您應該根據應用程式的架構和儲存需求，為您的容器化應用程式選擇儲存後端。如需有關 的詳細資訊AWS儲存體服務，請參閱雲端儲存AWS。

對於包含 Linux 執行個體或與 Fargate 特搭配使用的 Amazon ECS 叢集，Amazon ECS 會與 Amazon EFS 和 Amazon EBS 整合以提供容器儲存體。Amazon EFS 和 Amazon EBS 之間最明顯的區別在於，您可以同時在數千個 Amazon ECS 任務上掛載 Amazon EFS 檔案系統。相反，Amazon EBS 磁碟區不支援並行存取。基於此，Amazon EFS 是水平擴展的容器化應用程式的建議儲存選項。這是因為它支持並發。Amazon EFS 會將您的資料備援存放在多個可用區域，並提供 Amazon ECS 任務的低延遲存取，無論可用區域為何。亞馬遜 EFS 支援在 Amazon EC2 和 Fargate 執行的任務。

假設您有一個應用程序，例如需要亞毫秒延遲的事務數據庫，並且在水平縮放時不需要共享文件系統。對於這類應用程式，我們建議使用 Amazon EBS 磁碟區進行永久性儲存。目前，Amazon ECS 僅支援在 Amazon EC2 上託管的任務的 Amazon EBS 磁碟區。亞馬遜 EBS 磁碟區的 Support 不適用於 Fargate 的任務。在將 Amazon EBS 磁碟區搭配 Amazon ECS 任務使用之前，您必須先將 Amazon EBS 磁碟區連接到容器執行個體，並從任務的生命週期分開管理磁碟區。

對於包含 Windows 執行個體的叢集，適用於 Windows 檔案伺服器的 Amazon FSX 會為容器提供永久性儲存體。Amazon FSx for Windows File Server 檔案系統支援異地同步備份部署。透過這些部署，您可以與跨多個可用區域執行的 Amazon ECS 任務共用檔案系統。

您也可以將 Amazon EC2 執行個體儲存用於使用繫結掛載或 Docker 磁碟區託管在 Amazon EC2 上的 Amazon ECS 任務的資料持續性。使用繫結掛載或 Docker 磁碟區時，容器會將資料儲存在容器執行個體檔案系統上。使用主機檔案系統進行容器儲存的其中一項限制是資料一次只能在單一容器執行個體上使用。這表示容器只能在資料所在的主機上執行。因此，只有在應用程式層級處理資料複寫的案例中，才建議使用主機儲存體。

Amazon EFS 磁碟區

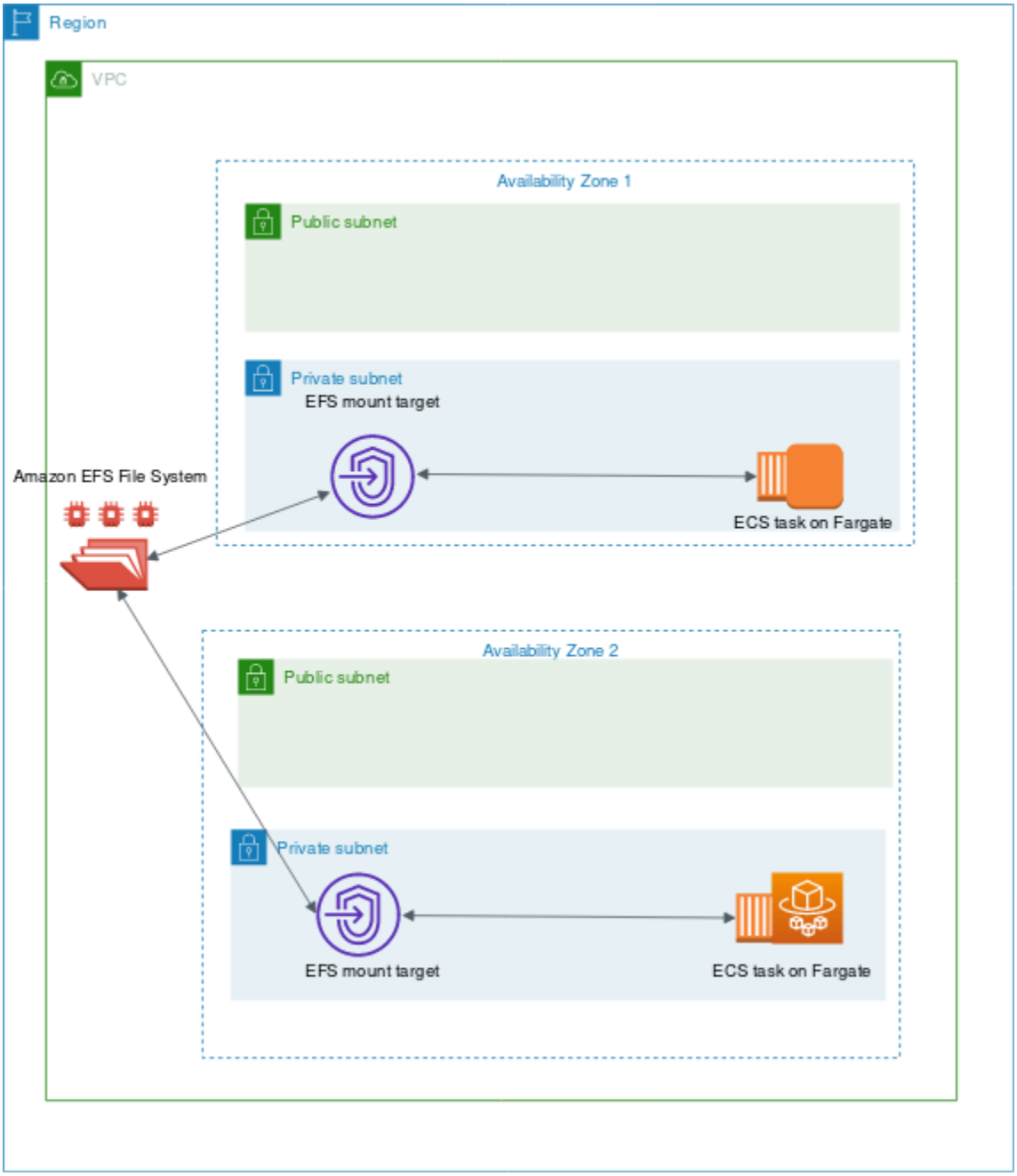
Amazon Elastic File System EFS) 提供簡單、可擴展的全受管的彈性 NFS 檔案系統。它是為了能夠根據需求擴充到 PB，而不會中斷應用程式。它可以在您新增和移除檔案時縮放或縮小。

您可以使用 Amazon EFS 磁碟區提供永久性儲存，在 Amazon ECS 中執行可設定狀態的應用程式。在 Amazon EC2 執行個體上或使用平台版本在 Fargate 上執行的 Amazon ECS 任務1.4.0和更新版本可以掛載現有 Amazon EFS 檔案系統。由於多個容器可以同時裝載和存取 Amazon EFS 檔案系統，因此無論其託管位置為何，您的任務都可以存取相同的資料集。

要在您的容器中掛載 Amazon EFS 檔案系統，您可以參考 Amazon ECS 任務定義中的 Amazon EFS 檔案系統和容器掛載點。以下是任務定義的程式碼片段，此程式碼片段會使用 Amazon EFS 進行容器儲存。

```
...
"containerDefinitions":[
    {
        "mountPoints": [
            {
                "containerPath": "/opt/my-app",
                 "sourceVolume": "Shared-EFS-Volume"
            }
    }
  ]
...
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "transitEncryption": "DISABLED",
        "rootDirectory": ""
      },
      "name": "Shared-EFS-Volume"
    }
  ]
```

Amazon EFS 會跨單一區域內的數個可用區域存放資料。Amazon ECS 任務會在其可用區域中使用 Amazon EFS 掛載目標來掛載 Amazon EFS 檔案系統。只有在 Amazon EFS 檔案系統在執行任務的可用區域中有掛載目標時，Amazon ECS 任務才能掛載 Amazon EFS 檔案系統。因此，最佳做法是在計劃在其中託管 Amazon ECS 任務的所有可用區域中建立 Amazon EFS 掛載目標。



如需詳細資訊，請參閱「」Amazon EFS 磁碟區中的Amazon Elastic Container Service 開發者指南。

安全性與存取控制

Amazon EFS 提供存取控制功能，可讓您使用這些功能確保儲存在 Amazon EFS 檔案系統中的資料安全無虞，且只能從需要的應用程式存取。您可以啟用靜態和傳輸中加密來保護資料的安全。如需詳細資訊，請參閱「」Amazon EFS 的資料加密中的Amazon Elastic File System 使用指南。

除了資料加密之外，您還可以使用 Amazon EFS 限制對檔案系統的存取。有三種方法可以實作 EFS 存取控制。

- 安全群組— 使用 Amazon EFS 掛載目標，您可以設定用來允許和拒絕網路流量的安全群組。您可以配置連接到 Amazon EFS 的安全群組，以允許連接到 Amazon ECS 執行個體的安全群組中的 NFS 流量 (連接埠 2049)，或在使用`awsvpc`網路模式，執行亞馬遜 ECS 任務。
- IAM— 您可以使用 IAM 限制存取 Amazon EFS 檔案系統。設定後，Amazon ECS 任務需要 IAM 角色來存取檔案系統，才能掛載 EFS 檔案系統。如需詳細資訊，請參閱「」使用 IAM 控制檔案系統資料存取中的Amazon Elastic File System 使用指南。

  IAM 政策還可以強制執行預先定義的條件，例如要求用戶端在連線到 Amazon EFS 檔案系統時使用 TLS。如需詳細資訊，請參閱「」用戶端的亞馬遜 EFS 條件金鑰中的Amazon Elastic File System 使用指南。
- Amazon EFS 存取點Amazon EFS 存取點是 Amazon EFS 檔案系統中的應用程式特定進入點。您可以針對透過存取點提出的所有檔案系統要求，使用存取點來強制執行使用者身分 (包括使用者的 POSIX 群組)。存取點也可以針對檔案系統強制執行不同的根目錄。這樣做可讓用戶端只能存取指定目錄或其子目錄中的資料。

考慮在 Amazon EFS 檔案系統上實作所有三種存取控制，以達到最大的安全性。例如，您可以將連接到 Amazon EFS 掛載點的安全群組設定為僅允許從與您的容器執行個體或 Amazon ECS 任務相關聯的安全群組進入 NFS 流量。此外，您可以將 Amazon EFS 設定為要求具有 IAM 角色才能存取檔案系統，即使連線來自允許的安全群組。最後，您可以使用 Amazon EFS 存取點強制執行 POSIX 使用者權限，並指定應用程式的根目錄。

以下任務定義程式碼片段示範如何使用存取點掛載 Amazon EFS 檔案系統。

```
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "authorizationConfig": {
          "acessPointId": "fsap-1234",
          "iam": "ENABLED"
        },
        "transitEncryption": "ENABLED",
        "rootDirectory": ""
      },
      "name": "my-filesystem"
    }
```

```
]
```

Performance

亞馬遜 EFS 提供兩種效能模式：一般用途和最大 I/O。一般用途適用於對延遲敏感的應用程式，例如內容管理系統和 CI/CD 工具。相較之下，Max I/O 檔案系統適用於工作負載，例如資料分析、媒體處理和機器學習。這些工作負載需要從數百甚至數千個容器執行平行作業，並且需要最高的彙總輸送量和 IOPS。如需詳細資訊，請參閱「」Amazon EFS 效能模式中的Amazon Elastic File System 使用指南。

某些對延遲敏感的工作負載需要最高 I/O 效能模式提供的較高 I/O 層級，以及一般用途效能模式提供的較低延遲。對於這類工作負載，我們建議建立多個一般用途效能模式的檔案系統。如此一來，只要工作負載和應用程式可以支援以下做法，您就可以將應用程式工作負載分攤到所有這些檔案系統中。

Throughput

所有 Amazon EFS 檔案系統都有相關聯的計量輸送量，這些輸送量取決於使用佈建輸送量或儲存在 EFS 標準或單一區域儲存類別中的資料量，適用於使用高載傳送量。如需詳細資訊，請參閱「」了解計量傳送量中的Amazon Elastic File System 使用指南。

Amazon EFS 檔案系統的預設輸送量模式是高載模式。使用成組分解模式時，檔案系統可用的輸送量會隨著檔案系統的成長而擴充或擴充。由於檔案型工作負載通常會突增，所以 Amazon EFS 的設計成突增，其他時間的傳輸量需要高傳輸量層級，所以 Amazon EFS 的設計成突增，以允許一段時間內的傳輸量層級。此外，由於許多工作負載都是大量讀取，因此讀取作業的計量比例與其他 NFS 作業 (例如寫入) 以 1:3 的比例計量。

對於每 TB 的 Amazon EFS 標準或 Amazon EFS 單區域儲存，所有 Amazon EFS 檔案系統都能提供每秒 50 MB 的一致基準效能。所有檔案系統 (不論大小為何) 都可以高載到 100 MB/s。具有 1TB 以上的 EFS 標準或 EFS 單一區域儲存的檔案系統，每 TB 都可以高載到每秒 100 MB。由於讀取作業是以 1:3 比率計量，因此每個 TiB 的讀取輸送量最多可驅動 300 MIBS/s。當您將資料新增至檔案系統時，檔案系統可用的最大輸送量會隨著 Amazon EFS Standard 儲存類別中的儲存而自動進行線性擴展。如果您需要的輸送量超過儲存資料量所能達到的輸送量，您可以將佈建輸送量設定為工作負載所需的特定數量。

檔案系統輸送量在連接到檔案系統的所有 Amazon EC2 執行個體之間共享。例如，可高載量達每秒 100 MB 的 1TB 檔案系統，可從單一 Amazon EC2 執行個體驅動每秒 100 MB 的檔案系統，每個執行個體都可以驅動每秒 10 MB。如需詳細資訊，請參閱「」Amazon EFS 效能中的Amazon Elastic File System 使用指南。

成本最佳化

Amazon EFS 為您簡化了擴展儲存。當您新增更多資料時，Amazon EFS 檔案系統會自動成長。尤其是 Amazon EFS高載傳送量模式中，Amazon EFS 上的輸送量會隨著標準儲存類別中的檔案系統大小來擴展。若要改善輸送量，而不需支付 EFS 檔案系統上佈建輸送量的額外成本，您可以與多個應用程式共用 Amazon EFS 檔案系統。使用 Amazon EFS 存取點，您可以在共用的 Amazon EFS 檔案系統中實作儲存隔離。如此一來，即使應用程式仍然共用相同的檔案系統，除非您授權，否則他們無法存取資料。

隨著資料成長，Amazon EFS 可協助您自動將不常存取的檔案移至較低的儲存類別。Amazon EFS 標準不常存取 (IA) 儲存類別可讓不會每天存取的檔案節省儲存成本。它執行此操作同時又維持 Amazon EFS 所提供的高可用性、高耐用性、彈性以及 POSIX 檔案系統存取。如需詳細資訊，請參閱「」Amazon EFS 儲存類別中的Amazon Elastic File System 使用指南。

考慮使用 Amazon EFS 生命週期政策，藉由將不常存取的檔案移至 Amazon EFS IA 儲存體，自動節省成本。如需詳細資訊，請參閱「」Amazon EFS 生命週期管理中的Amazon Elastic File System 使用指南。

建立 Amazon EFS 檔案系統時，您可以選擇 Amazon EFS 是否跨多個可用區域 (標準) 複寫您的資料，還是將資料以備援方式存放在單一可用區域中。與 Amazon EFS 標準儲存類別相比，Amazon EFS 單一區域儲存類別可以大幅降低儲存成本。考慮針對不需要異地同步備份恢復能力的工作負載使用 Amazon EFS One 區域儲存類別。您可以將不常存取的檔案移至 Amazon EFS 一區域不常存取，進一步降低 Amazon EFS 一區域儲存的成本。如需詳細資訊，請參閱「」Amazon EFS 不常存取。

資料保護

Amazon EFS 使用標準儲存類別，將您的資料跨多個可用區域存放在檔案系統的冗餘區域。如果您選取 Amazon EFS One 區域儲存類別，您的資料會以冗餘方式儲存在單一可用區域中。此外，Amazon EFS 的設計目的是在指定年份內提供 99.999999999% (11 9) 的耐用性。

與任何環境一樣，最佳做法是有備份並建置防止意外刪除的防護措施。對於 Amazon EFS 資料，最佳做法包括使用AWS Backup。使用 Amazon EFS One Zone 儲存類別的檔案系統預設會在建立檔案系統時自動備份檔案，除非您選擇停用此功能。如需詳細資訊，請參閱「」Amazon EFS 的資料保護中的Amazon Elastic File System 使用指南。

使用案例

Amazon EFS 提供平行共用存取，此存取會隨著檔案新增和移除來自動成長和縮減。因此，Amazon EFS 適合任何需要具有低延遲、高輸送量和寫入後讀取一致性等功能的儲存應用程式。對於水平擴展

且需要共用檔案系統的應用程式，Amazon EFS 是理想的儲存後端。資料分析、媒體處理、內容管理和 Web 服務等工作負載都是 Amazon EFS 使用案例。

Amazon EFS 可能不適合的一個使用案例是針對需要亞毫秒延遲的應用程式。這通常是交易式資料庫系統的要求。建議您執行儲存效能測試，以判斷使用 Amazon EFS 對於延遲敏感應用程式的影響。如果使用 Amazon EFS 時應用程式效能下降，請考慮使用 Amazon EBS io2 區塊 Express，該區塊 Express 會在 Nitro 執行個體上提供次毫秒、低變異 I/O 延遲。如需詳細資訊，請參閱「」Amazon EBS 磁碟區類型中的Amazon EC2 Linux 執行個體使用者指南。

如果某些應用程式的基礎儲存體意外變更，則會失敗。因此，Amazon EFS 不是這些應用程式的最佳選擇。相反，您可能更喜歡使用不允許從多個地方並發訪問的存儲系統。

Docker 磁碟區

Docker 磁碟區是 Docker 容器執行階段的一項功能，可讓容器透過從主機的檔案系統掛載目錄來保存資料。Docker 磁碟區驅動程式 (也稱為外掛程式) 用外部儲存系統 (例如 Amazon EBS) 來整合容器磁碟區。只有在 Amazon EC2 執行個體上託管 Amazon ECS 任務時才支援 Docker 磁碟區。

Amazon ECS 任務可以使用 Docker 磁碟區使用 Amazon EBS 磁碟區來保存資料。這是通過將 Amazon EBS 磁碟區附加到 Amazon EC2 執行個體，然後使用 Docker 磁碟區將磁碟區掛載到任務中來完成的。Docker 磁碟區可以在主機上的多個 Amazon ECS 任務之間共用。

Docker 磁碟區的限制在於任務使用的檔案系統與特定的 Amazon EC2 執行個體相關聯。如果實例因任何原因而停止，並且任務被放置在另一個實例上，數據將丟失。您可以將任務分配給實例，以確保關聯的 EBS 卷始終可用於任務。

如需詳細資訊，請參閱「」Docker 磁碟區中的Amazon Elastic Container Service 開發者指南。

Amazon EBS 磁碟區生命週期

容器儲存和 Amazon EBS 有兩種關鍵使用模式。第一個是當應用程序需要保存數據並防止數據丟失時，其容器終止。這種類型的應用程序的一個例子將是像 MySQL 一樣的事務數據庫。當 MySQL 任務終止時，另一個任務預計將替換它。在此案例中，磁碟區的生命週期與工作的生命週期分開。使用 EBS 保存容器數據時，最好使用任務位置約束將任務放置限制到附加 EBS 卷的單個主機上。

第二個是磁碟區的生命週期獨立於作業生命週期時。這對於需要高效能和低延遲儲存體，但在工作終止後不需要保留資料的應用程式特別有用。例如，處理大量資料的 ETL 工作負載可能需要高輸送量儲存體。Amazon EBS 適用於這種類型的工作負載，因為它提供高達 256,000 IOPS 的高效能磁碟區。當

任務終止時，取代複本可以安全地放置在叢集中的任何 Amazon EC2 主機上。只要工作能夠存取可滿足其效能需求的儲存後端，工作就可以執行其功能。因此，在此情況下不需要任務位置限制。

如果叢集中的 Amazon EC2 執行個體連接了多種類型的 Amazon EBS 磁碟區，您可以使用任務放置限制來確保任務放置在已連接適當 Amazon EBS 磁碟區的執行個體上。例如，假如叢集擁有某些具有gp2磁碟區，而其他人使用io1磁碟區。您可以將自定義屬性附加到具有io1磁碟區，然後使用工作位置限制來確保您的 I/O 密集型工作永遠放置在具有io1磁碟區。

如下所示AWS CLI命令用於將屬性置放到 Amazon ECS 容器執行個體上。

```
aws ecs put-attributes \
     --attributes name=EBS,value=io1,targetId=<your-container-instance-arn>
```

Amazon EBS 資料可用性

容器通常是短暫的，經常創建，並隨著應用程序水平擴展和向外而終止。最佳做法是，您可以在數個可用區域中執行工作負載，以改善應用程式的可用性。Amazon ECS 可讓您使用任務放置策略和任務放置限制來控制任務放置。當工作負載使用 Amazon EBS 磁碟區保留其資料時，其任務必須放置在與 Amazon EBS 磁碟區相同的可用區域中。我們也建議您設定位置限制，以限制可放置任務的「可用區域」。這可確保您的任務及其對應的磁碟區永遠位於相同的可用區域中。

執行獨立工作時，您可以使用可用區域屬性來設定位置限制，來控制要放置工作的可用區域。

```
attribute:ecs.availability-zone == us-east-1a
```

執行可在多個可用區域中執行的應用程式時，請考慮為每個可用區域建立不同的 Amazon ECS 服務。這可確保需要 Amazon EBS 磁碟區的任務永遠放置在與相關磁碟區相同的可用區域中。

我們建議在每個可用區域中建立容器執行個體，並使用啟動範本，並新增自訂屬性新增至執行個體，以區分這些執行個體與 Amazon ECS 叢集中的其他容器執行個體。建立服務時，請設定任務放置限制，以確保 Amazon ECS 將任務放置在正確的可用區域和執行個體。如需詳細資訊，請參閱「」任務位置限制範例中的Amazon Elastic Container Service 開發者指南。

Docker 磁碟區外掛程式

碼頭插件（如 Portworx）提供碼頭卷和亞馬遜 EBS 卷之間的抽象。當您需要磁碟區的任務啟動時，這些外掛程式可以動態建立 Amazon EBS 磁碟區。Portworx 也可以將磁碟區附加到新的主機，當容器終止，並且其後續複本會放置在不同的容器執行個體上。它也會在 Amazon ECS 節點之間和跨可用區域複寫每個容器的磁碟區資料。如需詳細資訊，請參閱「」Portworx。

Amazon FSx for Windows File Server

適用於 Windows 檔案伺服器的 Amazon FSX 提供完全受管、高度可靠且可擴展的檔案儲存，可透過業界標準的伺服器訊息區 (SMB) 通訊協定存取。它建置在 Windows 伺服器上，提供廣泛的系統管理功能，例如使用者配額、使用者檔案還原，以及 Microsoft 使用中目錄 (AD) 整合。它提供單一可用區和異地同步備份部署選項、完全受管的備份，以及對靜態和傳輸中的資料加密。

Amazon ECS 支援在 Amazon ECS 視窗任務定義中使用適用於 Windows 的 Amazon FSX 檔案伺服器，可使用稱為全域對映的 SMB 功能，透過 SMBv3 通訊協定作為掛接點。

若要設定適用於 Windows 檔案伺服器和 Amazon ECS 整合的 Amazon FSX，Windows 容器執行個體必須是作用中目錄網域服務 (AD DS) 上的網域成員，該網域成員由AWS Directory Service for Microsoft Active Directory、內部部署使用中目錄或 Amazon EC2 上自行託管的使用中目錄。AWS Secrets Manager用來儲存敏感資料，例如用來對應 Windows 容器執行個體上共用的 Active Directory 認證的使用者名稱和密碼。

若要為您的容器使用 Amazon FSx for Windows File Server 檔案系統磁碟區，您必須在任務定義中指定磁碟區和掛載點組態。以下是任務定義的程式碼片段，此程式碼片段會將 Amazon FSx for Windows File Server 用於容器儲存裝置。

```
{
 "containerDefinitions": [{
  "name": "container-using-fsx",
  "image": "iis:2",
  "entryPoint": [
   "powershell",
   "-command"
  ],
  "mountPoints": [{
   "sourceVolume": "myFsxVolume",
   "containerPath": "\\mount\\fsx",
   "readOnly": false
  }]
 }],
 "volumes": [{
  "fsxWindowsFileServerVolumeConfiguration": {
   "fileSystemId": "fs-ID",
   "authorizationConfig": {
    "domain": "ADDOMAIN.local",
    "credentialsParameter": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:SecretName"
```

```
    },
    "rootDirectory": "share"
   }
  }]
}
```

如需詳細資訊，請參閱「」Amazon FSx for Windows File Server 磁碟區中的Amazon Elastic Container Service 開發者指南。

安全性與存取控制

Amazon FSx for Windows File Server 檔案系統 FSx for Windows File Server 檔案系統中儲存的資料安全且只能從需要的應用程式存取。

資料加密

Amazon FSx for Windows File Server 支援兩種形式的檔案系統加密。它們是傳輸中的數據加密和靜態加密。對應至支援 SMB 通訊協定 3.0 或更新版本的容器執行個體上的檔案共用支援傳輸中的資料加密。建立 Amazon FSX 檔案系統時，會自動啟用靜態資料加密。當您存取檔案系統時，Amazon FSX 會使用 SMB 加密自動加密傳輸中的資料，無需修改應用程式。如需詳細資訊，請參閱「」Amazon FSx 的資料加密中的Amazon FSx for Windows File Server 使用指南。

使用視窗 ACL 的資料夾層級存取控制

Windows Amazon EC2 執行個體使用使用中目錄登入資料存取 Amazon FSX 檔案共用。它使用標準的 Windows 存取控制清單 (ACL) 來進行精細的檔案和資料夾層級的存取控制。您可以為共用內的特定資料夾建立多個認證，每個認證都會對應至特定工作。

在下列範例中，任務可以存取資料夾App01使用保存在 Secrets Manager 中的認證。它的 Amazon Resource Name（ ARN ）是1234。

```
"rootDirectory": "\\path\\to\\my\\data\App01",
"credentialsParameter": "arn-1234",
"domain": "corp.fullyqualified.com",
```

在另一個範例中，工作可以存取資料夾App02使用保存在 Secrets Manager 中的認證。其 ARN 是6789。

```
"rootDirectory": "\\path\\to\\my\\data\App02",
"credentialsParameter": "arn-6789",
```

```
"domain": "corp.fullyqualified.com",
```

使用案例

容器不是設計來保存資料。不過，某些容器化的 .NET 應用程式可能需要本機資料夾做為永續性儲存，以儲存應用程式輸出。Amazon FSx for Windows File Server 會在容器中提供本地資料夾。這可讓多個容器在同一個由 SMB 共用支援的檔案系統上讀寫。

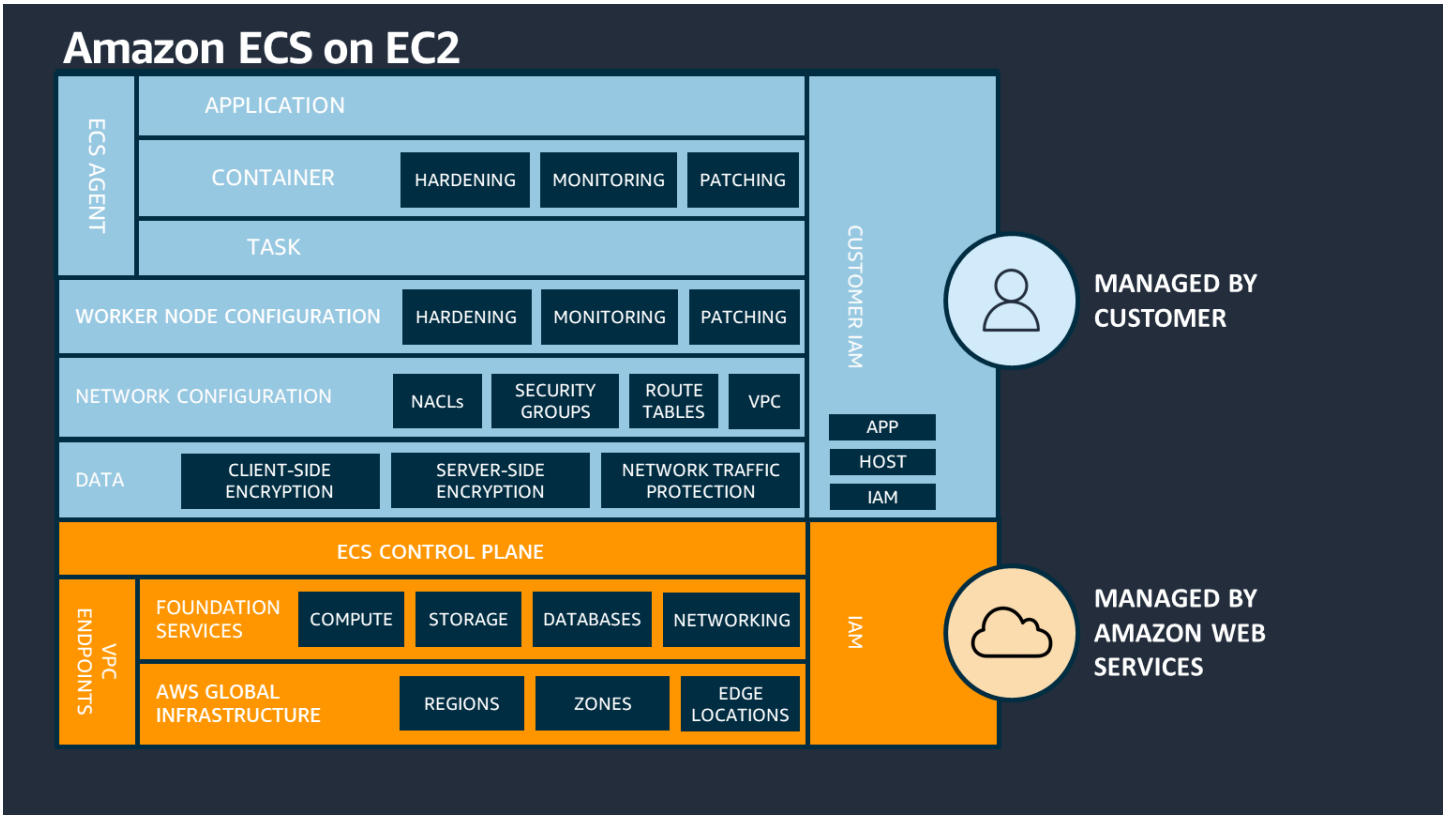
最佳實務-安全

本指南提供安全性和合規性建議，以保護您的資訊、系統和其他依賴於 Amazon ECS 的資產。它還引入了一些風險評估和緩解策略，您可以使用這些策略來更好地控制專為 Amazon ECS 叢集及其支援的工作負載而建置的安全控制。本指南中的每個主題都以簡短的概觀開始，後面是您可以用來保護 Amazon ECS 叢集的建議和最佳實務清單。

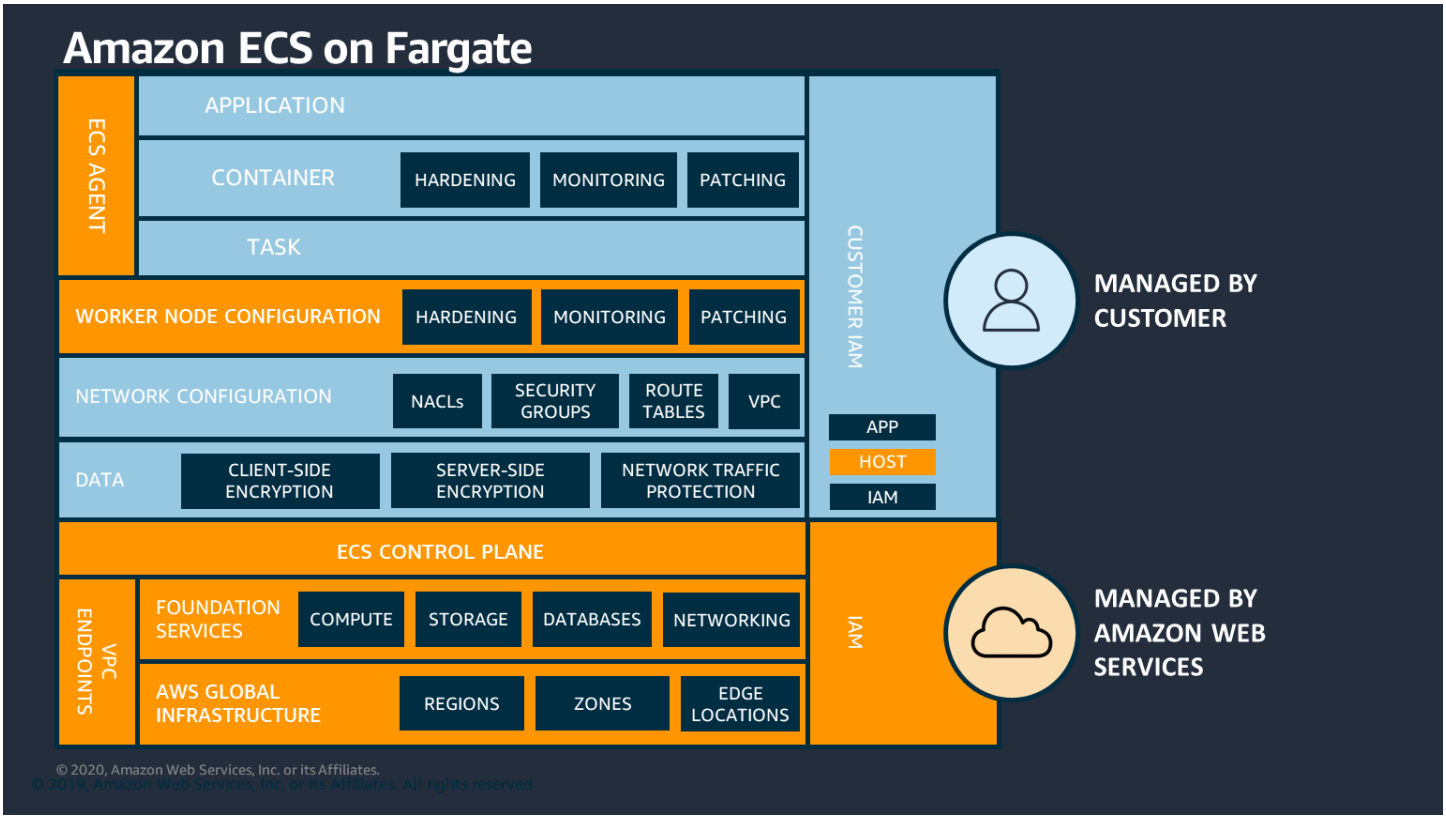
主題

共同責任模型

像 Amazon ECS 這樣的受管服務的安全性和合規性是您和AWS。一般來說，AWS負責雲端的安全性，而客戶則負責「在」雲端的安全性。AWS負責管理 Amazon ECS 控制平面，包括提供安全可靠服務所需的基礎設施。而且，您主要負責本指南中的主題。這包括資料、網路和執行階段安全性，以及記錄和監視。



關於基礎架構安全性，AWS承擔更多責任AWS Fargate資源，而不是其他自我管理的執行個體。有了FargateAWS會管理雲端中基礎執行個體的安全性，以及用來執行工作的執行階段。Fargate 也會代表您自動擴充您的基礎設施。



在將服務擴展到雲端之前，您應該了解您需要負責哪些安全性和合規性。

如需共用責任模型的詳細資訊，請參閱共同責任模型。

AWS Identity and Access Management

您可以使用AWS Identity and Access Management(IAM) 來管理和控制對AWS服務和資源，用於驗證和授權目的。更具體地說，透過這項服務，您可以控制對AWS資源，使用套用至 IAM 使用者、群組或角色的政策。在這三個中，IAM 使用者是可以存取您的資源的帳戶。而且，IAM 角色是一組可以由已驗證身分所假設的權限，該身分與 IAM 以外的特定身分無關。如需詳細資訊，請參閱「」IAM 中的政策和權限？。

管理對亞馬遜 ECS 的存取

您可以透過建立和套用 IAM 政策來控制對 Amazon ECS 的存取。這些策略是由一組適用於特定資源的動作所組成。政策的動作定義了允許或拒絕的操作清單 (例如 Amazon ECS API)，而資源控制動作適用於哪些 Amazon ECS 物件。您可以將條件新增至原則以縮小其範圍。例如，您可以撰寫原則，只允許對具有特定標籤集的工作執行動作。如需詳細資訊，請參閱「」Amazon ECS 如何與 IAM 搭配運作中的Amazon Elastic Container Service 開發者指南。

Recommendations

我們建議您設定 IAM 角色和政策時執行下列動作。

遵循最低權限存取的政策

建立規定範圍的原則，以允許使用者執行其指定的工作。例如，如果開發人員需要定期停止工作，建立只允許該特定動作的原則。下列範例只允許使用者停止屬於特定`task_family`在具有特定 Amazon Resource Name (ARN) 的叢集上。在條件下引用 ARN 也是使用資源級權限的一個例子。您可以使用資源層級許可來指定動作套用到的資源。

> **Note**
>
> 在政策中參考 ARN 時，請使用新的較長 ARN 格式。如需詳細資訊，請參閱「」Amazon Resource Name (ARN) 和 ID中的Amazon Elastic Container Service 開發者指南。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:StopTask"
      ],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"
        }
      },
      "Resource": [
        "arn:aws:ecs:<region>:<aws_account_id>:task-definition/<task_family>:*"
      ]
    }
  ]
}
```

讓叢集資源做為系統管理界限

範圍太窄的原則可能會造成角色擴散，並增加系統管理負荷。而不是建立限定在特定工作或服務範圍的角色，只建立叢集範圍的角色，並使用叢集做為主要系統管理界限。

透過建立自動化管線，將最終使用者與 Amazon ECS API 隔離

您可以透過建立自動將應用程式封裝和部署到 Amazon ECS 叢集的管線來限制使用者可以使用的動作。這會有效地將建立、更新和刪除任務的工作委派給管線。如需詳細資訊，請參閱「」教學課程：使用 CodePipeline 的亞馬遜 ECS 標準部署中的AWS CodePipeline使用者指南。

為增加一層安全使用原則條件

當您需要新增的安全性層時，請在原則中新增條件。如果您正在執行特殊權限的作業或需要限制可針對特定資源執行的一組動作，這會很有用。下列範例原則在刪除叢集時需要多因素授權。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DeleteCluster"
      ],
      "Condition": {
        "Bool": {
          "aws:MultiFactorAuthPresent": "true"
        }
      },
    "Resource": ["*"]
    }
  ]
}
```

套用至服務的標籤會傳播至屬於該服務一部分的所有工作。因此，您可以建立範圍限定在具有特定標籤的 Amazon ECS 資源的角色。在下列政策中，IAM 主體會以`Department`和一個標籤值`Accounting`。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:RunTask"
```

```
            ],
            "Resource": "arn:aws:ecs:*",
            "Condition": {
                "StringEquals": {"ecs:ResourceTag/Department": "Accounting"}
            }
        }
    ]
}
```

定期稽核對 Amazon ECS API 的存取

使用者可能會變更角色。變更角色之後，先前授與他們的權限可能不再套用。請確定您稽核擁有 Amazon ECS API 存取權的人員，以及該存取權是否仍然有保證。考慮將 IAM 與使用者生命週期管理解決方案整合，該解決方案會在使用者離開組織時自動撤銷存取。如需詳細資訊，請參閱「」Amazon ECS 安全性稽核指南中的Amazon Web Services 一般參考。

將 IAM 角色搭配 Amazon ECS 任務使用

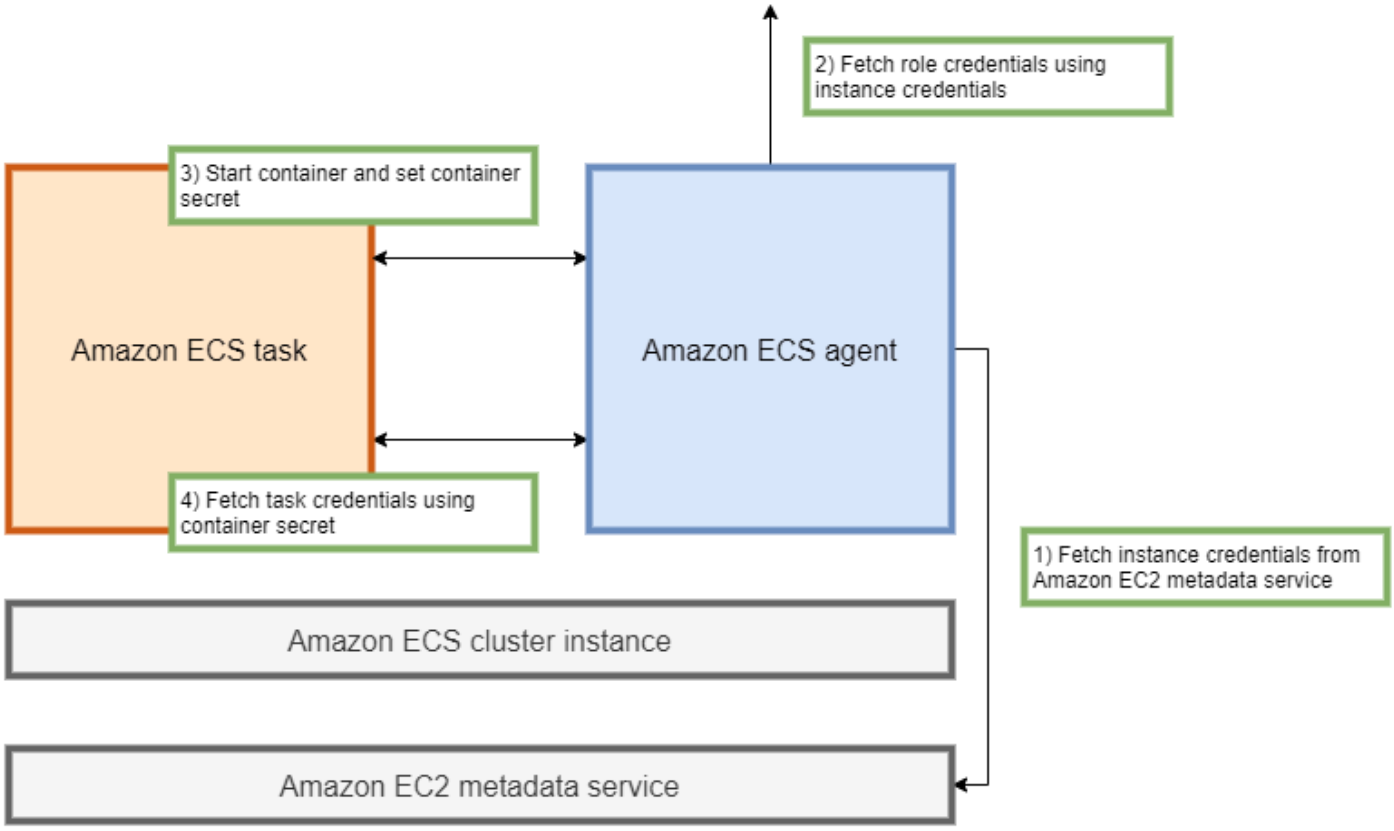
建議您將 IAM 角色指派給任務。其角色可以與執行的 Amazon EC2 執行個體的角色區分。為每個任務指定一個角色與最低權限存取的原則一致，並允許對動作和資源進行更精細的控制。

為任務指派 IAM 角色時，您必須使用下列信任政策，以便每項任務都可以採用與 EC2 執行個體所使用的 IAM 角色不同的 IAM 角色。這樣，你的任務不會繼承你的 EC2 實例的作用。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

當您將任務角色新增至任務定義時，Amazon ECS 容器代理程式會自動建立具有唯一認證識別碼的令牌 (例如，`12345678-90ab-cdef-1234-567890abcdef`) 來執行任

務。然後，此權杖和角色認證會新增至代理程式的內部快取。代理程式會填入環境變數AWS_CONTAINER_CREDENTIALS_RELATIVE_URI具有認證識別碼的 URI 的容器中 (例如/v2/credentials/12345678-90ab-cdef-1234-567890abcdef。



您可以手動從容器內擷取臨時角色登入資料，方法是將環境變數附加到 Amazon ECS 容器代理程式的 IP 位址，然後執行curl命令的結果字符串上。

```
curl 192.0.2.0$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

預期的輸出如下：

```
{
 "RoleArn": "arn:aws:iam::123456789012:role/SSMTaskRole-SSMFargateTaskIAMRole-
DASWWSF2WGD6",
 "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
 "Token": "IQoJb3JpZ2luX2VjEEM/Example==",
 "Expiration": "2021-01-16T00:51:53Z"
}
```

較新版本的AWSSDK 會自動從AWS_CONTAINER_CREDENTIALS_RELATIVE_URI環境可變AWSAPI呼叫。

輸出包含一組存取金鑰組，其中包含一組秘密存取金鑰 ID 和您的應用程式用來存取金鑰組。AWS的費用。它也包括一個令牌AWS用來確認認證有效。依預設，使用工作角色指派給工作的認證有效期為六小時。之後，Amazon ECS 容器代理會自動輪替它們。

任務執行角色

任務執行角色是用來授與 Amazon ECS 容器代理程式呼叫特定AWS會代表您執行 API 動作。例如，當您使用AWS Fargate，Fargate 需要一個 IAM 角色，允許它從亞馬遜 ECR 中提取圖像並將日誌寫入 CloudWatch Logs。當任務參考儲存在AWS Secrets Manager，例如圖像拉秘密。

> **Note**
>
> 如果您以經過驗證的用戶身份提取圖像，則不太可能受到碼頭集線器的拉動速率限制。如需詳細資訊，請參閱容器執行個體的私有登錄檔驗證。
> 通過使用亞馬遜 ECR 和亞馬遜 ECR 公共，您可以避免由 Docker 施加的限制。如果您從 Amazon ECR 提取影像，這也有助於縮短網路提取時間，並減少流量離開 VPC 時的資料傳輸變更。

> **Important**
>
> 當您使用 Fargate 時，您必須使用repositoryCredentials。無法設定 Amazon ECS 容器代理程式環境變數ECS_ENGINE_AUTH_TYPE或ECS_ENGINE_AUTH_DATA或修改ecs.config文件以獲取 Fargate 託管的任務。如需詳細資訊，請參閱「」任務的私有登錄檔身份驗證。

Amazon EC2 容器執行個體角色

Amazon ECS 容器代理程式是在 Amazon ECS 叢集中的每個 Amazon EC2 執行個體上執行的容器。它在亞馬遜 ECS 之外使用init命令，可在作業系統上使用。因此，無法透過任務角色授與權限。相反，必須將權限指派給代理程式執行的 Amazon EC2 執行個體。範例中的動作清單AmazonEC2ContainerServiceforEC2Role原則必須授與ecsInstanceRole。如果您不這樣做，您的執行個體就無法加入叢集。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTags",
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:UpdateContainerInstancesState",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

在此政策中，`ecr`和`logs`api 動作允許執行個體上執行的容器從 Amazon ECR 提取影像並將日誌寫入 Amazon CloudWatch。所以此`ecs`動作可讓代理程式註冊和取消註冊執行個體，以及與 Amazon ECS 控制平面進行通訊。其中，`ecs:CreateCluster`動作是選擇性的。

服務連結角色

您可以使用 Amazon ECS 的服務連結角色來授與 Amazon ECS 服務許可，代替您呼叫其他服務 API。Amazon ECS 需要建立和刪除網路介面、向目標群組註冊和取消註冊目標的權限。它也需要建立及刪除縮放政策所需的許可。這些權限會透過服務連結角色授予。第一次使用服務時，會代表您建立此角色。

> **Note**
>
> 如果您不小心刪除這個服務連結角色，則可重新建立該角色。如需說明，請參閱「」建立服務連結角色。

Recommendations

我們建議您在設定任務 IAM 角色和政策時執行下列動作。

封鎖存取 Amazon EC2 中繼資料

當您在 Amazon EC2 執行個體上執行任務時，強烈建議您封鎖對 Amazon EC2 中繼資料的存取，以防止容器繼承指派給這些執行個體的角色。如果您的應用程序必須調用AWSAPI 動作，請改為將 IAM 角色用於工作。

若要防止工作在橋模式存取 Amazon EC2 中繼資料時，請執行下列命令或更新執行個體的使用者資料。有關更新實例的用戶數據的更多說明，請參閱AWSSupport 文章。如需任務定義橋接模式的詳細資訊，請參閱工作定義網路模式。

```
sudo yum install -y iptables-services; sudo iptables --insert FORWARD 1 --in-interface
 docker+ --destination 192.0.2.0/32 --jump DROP
```

若要讓這項變更在重新開機後持續存在，請執行下列針對您的 Amazon Machine 映像 (AMI) 特定的命令：

- Amazon Linux 2

```
sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now
 iptables
```

- Amazon Linux

```
sudo service iptables save
```

對於使用awsvpc網路模式，請將環境變數ECS_AWSVPC_BLOCK_IMDS至true中的/etc/ecs/ecs.configfile.

您應該設置ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST變數設定為false放在 ecs-agent 配置文件中，以防止在host網路存取 Amazon EC2 中繼資料。

使用awsvpc網路模式

使用網路awsvpc網路模式，以限制不同任務之間或您的任務和在 Amazon VPC 內執行的其他服務之間的流量。這會增加額外的安全層。所以此awsvpc網路模式為在 Amazon EC2 上執行的任務提供任

務級網路隔離。它是上的默認模式AWS Fargate。這是您可以用來將安全性群組指派給工作的唯一網路模式。

使用 IAM 存取建議程式來調整角色

我們建議您移除任何從未使用過或已有一段時間未使用過的動作。這可以防止不必要的存取發生。若要執行此動作，請檢閱 IAM 存取建議程式產生的結果，然後移除從未使用過或最近未使用過的動作。您可以按照下列步驟執行此操作。

執行下列命令，以產生報告，顯示參考原則的上次存取資訊：

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

使用JobId在輸出中運行以下命令。在您執行這項操作後，您便可以檢視報告的結果。

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

如需詳細資訊，請參閱「」IAM 存取 Advisor。

監控AWS CloudTrail針對可疑活動

您可以將監控AWS CloudTrail進行任何可疑活動。大部分AWSAPI 呼叫會記錄至AWS CloudTrail作為事件。他們被分析AWS CloudTrail深入解析，您會收到任何可疑行為的警示。writeAPI 呼叫。這可能包括呼叫音量的尖峰。這些警示包括發生異常活動的時間，以及造成 API 的最高識別 ARN 等資訊。

您可以識別具有 IAM 角色的工作所執行的動作，位於AWS CloudTrail查看事件的userIdentity屬性。在以下範例中，arn包含假定角色的名稱，s3-write-go-bucket-role，後面是任務的名稱，7e9894e088ad416eb5cab92afExample。

```
"userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA36C6WWEJ2YEXAMPLE:7e9894e088ad416eb5cab92afExample",
    "arn": "arn:aws:sts::123456789012:assumed-role/s3-write-go-bucket-
role/7e9894e088ad416eb5cab92afExample",
    ...
}
```

> **Note**
>
> 在 Amazon EC2 容器執行個體上執行假定角色的任務時，Amazon ECS 容器代理程式會將請求記錄到位於`/var/log/ecs/audit.log.YYYY-MM-DD-HH`格式。如需詳細資訊，請參閱「」任務 IAM 角色記錄和記錄線索的 Insights 事件。

網路安全

網路安全性是涵蓋數個子主題的廣泛主題。其中包括傳輸中的加密、網路分段和隔離、防火牆、流量路由和可觀度。

傳輸中加密

加密網路流量可防止未經授權的使用者在透過網路傳輸資料時攔截和讀取資料。使用 Amazon ECS，可以透過以下任何方式來實作網路加密。

- 使用服務網狀 (TLS)：

  搭配AWS App Mesh，您可以在使用網狀端點部署的 Envoy Proxy 之間設定 TLS 連線。兩個範例是虛擬節點和虛擬閘道。TLS 憑證可以來自AWS Certificate Manager(ACM)。或者，它可以來自您自己的私人憑證授權單位。

  - 啟用 Transport Layer Security (TLS)
  - 啟用AWS App Mesh使用 ACM 憑證或客戶提供的憑證
  - TLS ACM 逐步解說
  - TLS 檔案演練
  - Envoy
- 使用 Nitro 執行個體：

  根據預設，下列 Nitro 執行個體類型之間的流量會自動加密：
  C5dn、G4、I3en、I3dn、M5dn、R5dn、R5dn、R5dn、R5dn、R5dn、R5dn、R5dn、R5dn、R5dn、R5
  流量透過傳輸閘道、負載平衡器或類似的中介路由時，不會加密流量。

  - 傳輸中加密
  - 2019 年開始的新會計
  - 這個演講來自 2019 年的重要資訊
- 搭配 Application Load Balancer 使用伺服器名稱指示 (SNI)：

Application Load Balancer (ALB) 和 Network Load Balancer (NLB) 支援伺服器名稱指示 (SNI)。透過使用 SNI，您可以將多個安全應用程式放在單一接聽程式後面。為此，每個人都有自己的 TLS 證書。建議您使用AWS Certificate Manager(ACM)，然後將它們新增至接聽程式的憑證清單。所以此AWS負載平衡器會搭配 SNI 使用智慧憑證選擇演算法。如果用戶端提供的主機名稱符合憑證清單中的單一憑證，負載平衡器會選擇該憑證。如果用戶端提供的主機名稱符合清單中的多個憑證，負載平衡器會選取用戶端可支援的憑證。範例包括自我簽署憑證或透過 ACM 產生的憑證。

- 使用 Application Load Balancer 的 SNI
- 使用 Network Load Balancer 的 SNI

- 使用 TLS 憑證的端對端加密：

  這牽涉到部署 TLS 憑證與工作。這可以是自我簽署憑證或來自信任憑證授權單位的憑證。您可以參考憑證的密碼來取得憑證。否則，您可以選擇執行向 ACM 發出憑證簽署要求 (CSR) 的容器，然後將產生的密碼掛接至共用磁碟區。

  - 使用搭配 Amazon ECS 第 1 部分的 Network Load Balancer，維護傳輸層安全性到您的容器
  - 維護傳輸層安全 (TLS) 一路到容器第 2 部分：使用 AWS Private Certificate Authority

任務聯網

下列建議將考慮到 Amazon ECS 的運作方式。Amazon ECS 不使用覆疊網路。相反地，工作會設定為在不同的網路模式下運作。例如，設定為使用`bridge`模式從每台主機上運行的 Docker 網絡獲取不可路由的 IP 地址。設定為使用`awsvpc`網路模式會從主機的子網路取得 IP 地址。設定為`host`網路使用主機的網路介面。`awsvpc`是慣用的網路模式。這是因為它是您可以用來將安全性群組指派給工作的唯一模式。這也是唯一可用於AWS Fargate任務。

任務的安全群組

我們建議您設定任務，以使用`awsvpc`網路模式。將任務配置為使用此模式後，Amazon ECS 代理程式會自動佈建並將彈性網路介面 (ENI) 附加到任務。佈建 ENI 時，工作會註冊在AWS安全群組。安全群組做為虛擬防火牆，您可以使用它來控制傳入及傳出流量。

服務網格與相互傳輸層安全性 (MTLS)

您可以使用服務網格，例如AWS App Mesh來控制網路流量。根據預設，虛擬節點只能與其設定的服務後端通訊，例如虛擬節點將與之通訊的虛擬服務。如果虛擬節點需要與網格外的服務進行通訊，您可以使用`ALLOW_ALL`輸出篩選器，或在外部服務的網格內建立虛擬節點。如需詳細資訊，請參閱「」Kubernetes 出口如何逐步解說。

App Mesh 也可讓您使用相互傳輸層安全性 (MTLS)，其中用戶端和伺服器都使用憑證進行互相驗證。然後，用戶端與伺服器之間的後續通訊會使用 TLS 加密。透過在網狀中的服務之間要求 MTLS，您可以確認流量來自受信任的來源。如需詳細資訊，請參閱下列主題：

- MTLS 身份驗證
- MTLS 秘密探索服務 (SDS) 逐步解說
- MTLS 檔案逐步解說

AWS PrivateLink

AWS PrivateLink是一種網路技術，可讓您為不同的AWS服務，包括亞馬遜 ECS。在 Amazon VPC 沒有連接 Internet Gateway (IGW) 且沒有其他路由到網際網路的沙箱化環境中，端點是必要的。使用AWS PrivateLink可確保對 Amazon ECS 服務的呼叫會保留在 Amazon VPC 內，且不會周遊網際網路。如需如何建立AWS PrivateLink終端節點和其他相關服務的詳細資訊，請參閱Amazon ECS 介面 Amazon VPC 端點介面。

> **Important**
>
> AWS Fargate任務不需要AWS PrivateLink端點。

Amazon ECR 和 Amazon ECS 都支援端點政策。這些原則可讓您縮小對服務 API 的存取權限。例如，您可以為 Amazon ECR 建立端點政策，該政策僅允許映像推送到登錄中，特別是AWS帳戶。這樣的政策可用於防止資料透過容器映像洩漏，同時仍允許使用者推送至授權的 Amazon ECR 登錄。如需詳細資訊，請參閱「」使用 VPC 端點政策。

下列政策允許所有AWS主參與者，以僅針對您的 Amazon ECR 存放庫執行所有動作：

```
{
  "Statement": [
    {
      "Sid": "LimitECRAccess",
      "Principal": "*",
      "Action": "*",
      "Effect": "Allow",
      "Resource": "arn:aws:ecr:region:your_account_id:repository/*"
    },
  ]
}
```

您可以通過設置一個條件來進一步增強這一點，該條件使用新的PrincipalOrgID屬性。這可以防止 IAM 主體推送和拉動圖像，該主體不屬於AWS Organizations。如需詳細資訊，請參閱「」aws:PrincipalOrgID。

我們建議將相同的政策套用至com.amazonaws.*region*.ecr.dkr與com.amazonaws.*region*.ecr.api端點。

Amazon ECS 容器代理設定

Amazon ECS 容器代理程式組態檔包含數個與網路安全相關的環境變數。ECS_AWSVPC_BLOCK_IMDS和ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST用於阻止任務對 Amazon EC2 元數據的訪問。HTTP_PROXY用於設定代理程式，以透過 HTTP 代理路由以連線至網際網路。如需設定代理程式和 Docker 執行階段以透過 Proxy 路由的指示，請參閱「」。HTTP 代理組態。

> **Important**
>
> 這些設定在您使用AWS Fargate。

Recommendations

建議您在設定 Amazon VPC、負載平衡器和網路時執行下列動作。

在適用情況下使用網路加密

您應該在適用的情況下使用網路加密。如果資料包含持卡人資料，例如 PCI DSS，某些合規性計劃會要求您加密傳輸中的資料。如果您的工作負載有類似的需求，請設定網路加密。

現代瀏覽器在連接到不安全的網站時警告用戶。如果您的服務是由面向公眾的負載平衡器提供，請使用 TLS/SSL 加密從客戶端瀏覽器到負載平衡器的流量，並在必要的情況下重新加密到後端。

使用**awsvpc**網路模式和安全性群組，當您需要控制工作之間或工作與其他網路資源之間的流量

您應該使用。awsvpc網路模式和安全性群組，當您需要控制工作之間以及工作與其他網路資源之間的流量時。如果您的服務位於 ALB 後面，請使用安全性群組，只允許使用與 ALB 相同的安全性群組來自其他網路資源的輸入流量。如果您的應用程式位於 NLB 後面，請將任務的安全群組設定為僅允許來自 Amazon VPC CIDR 範圍的輸入流量以及指派給 NLB 的靜態 IP 位址。

安全群組也應該用來控制 Amazon VPC 內的任務和其他資源之間的流量，例如 Amazon RDS 資料庫。

需要嚴格隔離網路流量時，在不同的 Amazon VPC 中建立叢集

當網路流量需要嚴格隔離時，您應該在不同的 Amazon VPC 中建立叢集。避免在不需要符合這些需求的工作負載的叢集上執行具有嚴格安全性需求的工作負載。強制執行嚴格的網路隔離時，請在不同的 Amazon VPC 中建立叢集，並使用 Amazon VPC 端點選擇性地將服務公開給其他 Amazon VPC。如需詳細資訊，請參閱「」Amazon VPC 端點。

設定AWS PrivateLink端點 (若有保固)

您應該將設定為AWS PrivateLink端點端點。如果您的安全政策阻止您將 Internet Gateway (IGW) 附加到 Amazon VPC，請設定AWS PrivateLink端點以及其他服務 (例如亞馬遜 ECR)、AWS Secrets Manager和 Amazon CloudWatch。

使用 Amazon VPC 流量日誌分析長時間執行任務的流量

您應該使用 Amazon VPC 流量日誌來分析長時間執行任務的流量。使用的任務`awsvpc`網路模式會取得自己的 ENI。如此一來，您就可以使用 Amazon VPC 流量日誌監控往返個別任務的流量。Amazon VPC 流量日誌 (v3) 的最新更新會使用流量中繼資料 (包括 vpc ID、子網路 ID 和執行個體 ID) 來豐富日誌。此中繼資料可用來協助縮小調查範圍。如需詳細資訊，請參閱「」Amazon VPC 流程日誌。

> **Note**
>
> 由於容器的暫時性，流程記錄可能不一定是分析不同容器或容器與其他網路資源之間流量模式的有效方法。

秘密管理

應用程式經常使用秘密 (例如 API 金鑰和資料庫認證) 來存取其他系統。它們通常包含使用者名稱和密碼、憑證或 API 金鑰。這些機密的存取應限制為使用 IAM 並在執行階段注入容器的特定 IAM 主體。

秘密可以無縫地注入到容器AWS Secrets Manager和 Amazon EC2 Systems Manager 參數存放區。這些秘密可以在您的任務中被引用為以下任何一個。

1. 它們被引用為使用`secrets`容器定義參數。
2. 它們被引用為`secretOptions`如果您的日誌記錄平台需要驗證。如需詳細資訊，請參閱「」日誌組態選項。

3. 它們被引用為使用`repositoryCredentials`容器定義參數，如果從中提取容器的登錄需要驗證。從 Docker 集線器提取影像時，請使用此方法。如需詳細資訊，請參閱「」任務的私有登錄檔身份驗證。

Recommendations

我們建議您在設定秘密管理時執行下列動作。

使用AWS Secrets Manager或 Amazon EC2 Systems Manager 參數存放區，以儲存秘密材料

您應該將 API 金鑰、資料庫憑證和其他秘密資料安全地儲存在AWS Secrets Manager或在 Amazon EC2 Systems Manager 參數存放區中做為加密參數。這些服務是相似的，因為它們都是使用AWS KMS來加密敏感資料。AWS Secrets Manager，也包含自動旋轉秘密、產生隨機秘密，以及在AWS帳戶。如果您認為這些重要功能，請使用AWS Secrets Manager否則請使用加密的參數。

> **Note**
>
> 參考來自秘密的任務AWS Secrets Manager或 Amazon EC2 Systems Manager 參數存放區需要任務執行角色，其中包含授與 Amazon ECS 存取所需秘密的政策，如果適用的話，AWS KMS密鑰，用於加密和解密該密碼。

> **Important**
>
> 在任務中引用的秘密不會自動旋轉。如果您的密碼發生變更，您必須強制執行新的部署或啟動新工作以擷取最新的密碼值。如需詳細資訊，請參閱下列主題：
>
> - AWS Secrets Manager：將數據注入為環境變量
> - Amazon EC2 Systems Manager 參數存放區：將數據注入為環境變量

從加密的 Amazon S3 儲存貯體擷取資料

因為環境變量的值可能會在日誌中無意中洩漏，並且在運行`docker inspect`，您應該將秘密存放在加密的 Amazon S3 儲存貯體中，並使用任務角色限制對這些秘密的存取。當您執行此操作時，您的應用程式必須寫入才能從 Amazon S3 儲存貯體讀取秘密。如需說明，請參閱「」設定 Amazon S3 儲存貯體的預設伺服器端加密行為。

使用邊車容器將秘密安裝到磁碟區

由於環境變數的資料洩漏風險較高，因此您應該執行一個側邊容器，從AWS Secrets Manager並將其寫入共用磁碟區。此容器可以運行和通過使用應用程序容器之前退出Amazon ECS 容器訂購。當您執行這項操作時，應用程式容器隨後會裝載寫入秘密的磁碟區。與 Amazon S3 儲存貯體方法一樣，您的應用程式必須寫入才能從共用磁碟區讀取秘密。因為磁碟區的範圍是工作，磁碟區會在工作停止之後自動刪除。如需附屬容器的範例，請參閱ABS 秘密側注射器專案。

> **Note**
>
> 在 Amazon EC2 上，寫入秘密的磁碟區可以使用AWS KMS客戶受管金鑰。在 上AWS Fargate，磁碟區儲存會使用服務管理的金鑰自動加密。

其他資源

- 在 Amazon ECS 任務中將秘密傳遞給容器
- 分庭是一個包裝函數，用於在 Amazon EC2 Systems Manager 參數存放區存放秘密

Compliance

使用 Amazon ECS 的合規責任取決於資料的機密性、您公司的合規目標，以及適用法律和法規。

AWS提供下列資源，以協助合規：

- 安全性與合規性快速入門指南：這些部署指南討論在上部署以安全及合規為重心基準環境的架構考量和步驟。AWS。
- HIPAA 安全與合規白皮書：本白皮書說明公司如何使用AWS以建立符合 HIPAA 規範的應用程式。
- AWS合規計劃範圍內的服務：此清單包含AWS服務範圍中的具體合規計劃。如需詳細資訊，請參閱「」AWS合規計劃。

支付卡產業資料安全標準 (PCI DSS)

遵守 PCI DSS 時，請務必瞭解持卡人資料 (CHD) 在環境中的完整流程。CHD 流程決定 PCI DSS 的適用性、定義持卡人資料環境 (CDE) 的界限和元件，以及 PCI DSS 評估的範圍。精確判定 PCI DSS 範圍是定義安全狀態並最終成功評估的關鍵。客戶必須具備範圍判定程序，以確保其完整性並偵測與範圍之間的變化或偏差。

容器化應用程式的暫時性，在稽核設定時提供額外的複雜性。因此，客戶必須保持對所有容器組態參數的認知，以確保在整個容器生命週期的所有階段都能滿足合規性需求。

如需有關在 Amazon ECS 上實現 PCI DSS 合規的其他資訊，請參閱下列白皮書。

- 在 Amazon ECS 上架構 PCI DSS 合規
- PCI DSS 範圍界定和分段的架構 AWS

HIPAA (美國 Health 保險流通與責任法案)

將 Amazon ECS 搭配處理受保護健康資訊 (PHI) 的工作負載使用，無需額外設定。Amazon ECS 充當協調流程服務，可協調 Amazon EC2 上的容器啟動。它不會與正在協調的工作負載內的資料一起運作或運作。符合 HIPAA 法規和AWS透過 Amazon ECS 啟動的容器存取「商業夥伴增補合約」，PHI 應在傳輸過程中和靜態狀態進行加密。

各種靜態加密機制可用於每個AWS儲存選項，例如 Amazon S3、亞馬遜 EBS 和AWS KMS。您可以部署覆疊網路 (例如 VNS3 或 Wed Net)，以確保在容器之間傳輸的 PHI 完整加密，或提供冗餘的加密層。完整的記錄也應該啟用，並且所有容器日誌都應該導向到 Amazon CloudWatch。如需詳細資訊，請參閱「」HIPAA 安全與合規架構。

Recommendations

您應該儘早與企業內的合規性計劃擁有者聯繫，並使用AWS共同責任模型，找出符合性控制所有權，以便利用相關的合規性計畫。

記錄和監控

日誌和監控是維護 Amazon ECS 和AWS解決方案。AWS提供多種工具來監控 Amazon ECS 資源並回應潛在事件：

- Amazon CloudWatch 警示
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CloudTrail 日誌

您可以在任務中設定容器，將日誌資訊傳送給 Amazon CloudWatch Logs。若您使用的是AWS Fargate啟動類型，您可以檢視容器的日誌。如果您使用的是 Amazon EC2 啟動類型，可以在一個便利的位置檢視容器的不同日誌。這也會防止容器日誌佔用容器執行個體上的磁碟空間。

如需 Amazon CloudWatch Logs 的詳細資訊，請參閱監控 Amazon EC2 執行個體的日誌中的Amazon CloudWatch 使用者指南。如需自任務將容器日誌傳送到 Amazon CloudWatch Logs 的指示，請參閱「」。使用awslogs記錄驅動程式。

使用流利位元的容器記錄

AWS提供了流利位元影像，其中包含 Amazon CloudWatch Logs 和 Amazon Kinesis Data Firehose 的外掛程式。此影像提供將日誌路由到 Amazon CloudWatch 和 Amazon Kinesis Data Firehose 目的地 (包括 Amazon S3、Amazon Elasticsearch Service 和 Amazon Redshift) 的功能。我們建議您使用 Fluent Bit 作為日誌路由器，因為它的資源使用率低於 Fluentd。如需詳細資訊，請參閱「」流利位元的 Amazon CloudWatch Logs和Amazon Kinesis Data Firehose 適用於流利位元。

所以此AWS的流利位映像可在以下位置使用：

- 亞馬遜 ECR 亞馬遜 ECR 公共畫廊
- Amazon ECR 存放庫(在大多數高可用性區域)
- Docker Hub

以下顯示 Docker CLI 使用的語法。

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:tag
```

例如，您可以將最新的AWS使用此 Docker CLI 命令的流利位圖像：

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:latest
```

另請參閱以下部落格文章，以取得更多關於 Fluent Bit 及相關功能的詳細資訊：

- 適用於亞馬遜 EKS 的流利位元 AWS Fargate
- 使用流利位元集中式容器記錄
- 建置可擴充的記錄解決方案彙總器AWS Fargate、流利和 Amazon Kinesis Data Firehose

自訂日誌路 FireLens 亞馬遜 ECS

使用 Amazon ECS 適用的 FireLens ens，您可以使用任務定義參數將日誌路由到AWS服務或AWS合作夥伴目的地，以儲存和分析日誌。FireLens 可與 Fluentd 和 Fluent Bit 搭配使用。我們提供AWS流利位圖像。或者，您也可以使用自己的 Fluentd 或流利位元映像。

在使用 Amazon ECS 的 FireLens 時，您應該考慮下列條件和考量：

- 適用於 Amazon ECS 的 FireLens 支援在AWS Fargate和 Amazon EC2。
- 支援 Amazon ECS 的火箭鏡頭AWS CloudFormation範本。如需詳細資訊，請參閱「」AWS::ECS::TaskDefinition FirelensConfiguration中的AWS CloudFormation使用者指南。
- 對於使用bridge網路模式，具有 FireLens 配置的容器必須在任何依賴該模式的應用程式容器啟動之前啟動。若要控制容器開始的順序，請在任務定義中使用相依性條件。如需詳細資訊，請參閱「」容器相依性。

AWS Fargate 安全性

我們建議您在使用時考慮下列最佳實務。AWS Fargate。

使用AWS KMS來加密臨時存儲

您應該將您的短暫存儲加密AWS KMS。對於託管在AWS Fargate使用平台版本1.4.0或更新版本中，每個任務都會收到 20 GB 的暫時性儲存體。儲存容量無法調整。對於 2020 年 5 月 28 日或之後啟動的這類任務，會以使用受管加密金鑰的 AES-256 加密演算法加密其暫時性儲存。AWS Fargate。

範例：啟動 Amazon ECS 任務AWS Fargate平台版本 1.4.0，具有暫時性儲存加密

以下命令將在上啟動 Amazon ECS 任務AWS Fargate平台 1.4 版。由於此任務是作為 Amazon ECS 叢集的一部分啟動，因此它會使用 20 GB 的自動加密的暫時儲存。

```
aws ecs run-task --cluster clustername \
  --task-definition taskdefinition:version \
  --count 1
  --launch-type "FARGATE" \
  --platform-version 1.4.0 \
  --network-configuration
 "awsvpcConfiguration={subnets=[subnetid],securityGroups=[securitygroupid]}" \
  --region region
```

用於核心系統呼叫追蹤的 SYS_PTRACE 功能

Docker 會提供新增或移除容器中的 Linux 功能的預設設定。如需有關可用功能的詳細資訊，請參閱執行階段權限與 Linux 功能中的Docker 執行文件中)。

在上啟動的工作AWS Fargate僅支援新增SYS_PTRACE核心功能。

請參閱以下教學影片，說明如何透過 Sysdig 使用此功能法爾科專案。

#ContainersFromTheCouch-為您的AWS Fargate使用系統追蹤功能的工作

在前面的視頻中討論的代碼可以在 GitHub 上找到這裡。

工作與容器安全性

您應該將容器影像視為對抗攻擊的第一條防線。不安全，構造不良的映像可能會允許攻擊者逃離容器的邊界並獲得對主機的訪問權限。您應該執行以下操作，以減輕這種情況發生的風險。

Recommendations

我們建議您在設定任務和容器時執行下列動作。

建立最少或使用無散佈影像

首先從容器映像中移除所有無關的二進位檔案。如果您使用來自 Docker Hub 的不熟悉映像，請檢查映像以參考每個容器圖層的內容。您可以使用應用程序，如潛水若要執行此作業。

或者，您可以使用無散佈影像，只包含您的應用程式及其執行階段相依性。它們不包含軟件包管理器或 shell。無散佈影像可改善「訊號到掃描器的雜訊，並減少建立來源的負擔，滿足您所需。」 如需詳細資訊，請參閱 GitHub 文件，無散佈。

Docker 有一個機制，用於從名為刮。如需詳細資訊，請參閱。使用創建一個簡單的父映像刮中的 Docker 文檔。使用像 Go 這樣的語言，您可以創建一個靜態鏈接的二進製文件並在 Docker 文件中引用它。以下範例示範如何完成此操作。

```
############################
# STEP 1 build executable binary
############################
FROM golang:alpine AS builder
# Install git.
# Git is required for fetching the dependencies.
RUN apk update && apk add --no-cache git
WORKDIR $GOPATH/src/mypackage/myapp/
COPY . .
# Fetch dependencies.
# Using go get.
RUN go get -d -v
# Build the binary.
RUN go build -o /go/bin/hello
```

```
############################
# STEP 2 build a small image
############################
FROM scratch
# Copy our static executable.
COPY --from=builder /go/bin/hello /go/bin/hello
# Run the hello binary.
ENTRYPOINT ["/go/bin/hello"]
This creates a container image that consists of your application and nothing else,
 making it extremely secure.
```

上一個範例也是多階段組建的範例。從安全性的角度來看，這些類型的組建是有吸引力的，因為您可以使用它們來最小化推送到容器登錄的最終映像的大小。容器映像沒有建置工具和其他無關的二進位檔案，藉由減少映像的攻擊面，改善您的安全性狀態。如需多階段組建的詳細資訊，請參閱建立多階段組建。

掃描影像是否有弱點

與其虛擬機器對應部門類似，容器映像可能包含具有弱點的二進位檔案和應用程式程式庫，或隨著時間而開發弱點。防止漏洞攻擊的最佳方式是使用影像掃描器定期掃描影像。儲存在 Amazon ECR 中的影像可透過推送或隨選掃描 (每 24 小時一次)。Amazon ECR 目前使用克萊爾，這是一個開放原始碼映像掃描解決方案。掃描影像後，結果會記錄到亞馬遜事件橋中的亞馬遜 ECR 事件串流。您也可以在 Amazon ECR 主控台中查看掃描結果，或呼叫描述影像搜尋結果API。影像與`HIGH`或`CRITICAL`弱點應該被刪除或重建。如果已部署的映像發生弱點，應儘快取代該弱點。

Docker 桌面邊緣 2.3.6.0或更新版本可以scan本機映像。掃描是由斯尼克，一種應用程序安全服務。當發現漏洞時，Snyk 會利用 Dockerfile 中的漏洞識別層和相依性。它還建議安全的替代方案，例如使用較少弱點的較薄基本映像，或將特定套件升級到較新版本。藉由使用 Docker 掃描，開發人員可以先解決潛在的安全性問題，再將其映像推送至登錄。

- 使用 Amazon ECR 和AWS Security Hub說明如何從亞馬遜 ECR 中顯示漏洞信息AWS Security Hub，並透過封鎖對易受攻擊影像的存取來自動修復。

移除影像中的特殊權限

存取權限旗標`setuid`和`setgid`允許使用可執行文件的擁有者或組的權限運行可執行文件。移除具有這些存取權限的所有二進位檔案，因為這些二進位檔案可用來提升權限。考慮刪除所有 shell 和實用程序，如`nc`和`curl`，可用於惡意目的。您可以使用`setuid`和`setgid`存取權限，請使用下列命令。

```
find / -perm /6000 -type f -exec ls -ld {} \;
```

要從這些文件中刪除這些特殊權限，請將以下指令添加到您的容器映像。

```
RUN find / -xdev -perm /6000 -type f -exec chmod a-s {} \; || true
```

建立一組策劃的影像

不要讓開發人員建立自己的映像，而是為組織中的不同應用程式堆疊建立一組經過審核的映像。藉由這樣做，開發人員可以放棄學習如何撰寫 Docker 檔案，並專注於撰寫程式碼。隨著變更合併到程式碼庫中，CI/CD 管線可以自動編譯資產，然後將其儲存在成品儲存庫中。最後，在將工件推送到 Docker 註冊表（如 Amazon ECR）之前，將工件複製到適當的映像中。至少你應該創建一組基本圖像，開發人員可以從. 您應避免從 Docker Hub 拉取映像檔。您並不總是知道圖像中的內容，而前 1000 張圖像中的五分之一有可能性。這些映像檔及其弱點的清單可以在https://vulnerablecontainers.org/。

掃描應用程式套件和程式庫中的弱點

開放原始碼程式庫的使用現在很常見。與作業系統和作業系統套件一樣，這些程式庫可能會有漏洞。作為開發生命週期的一部分，這些程式庫應該在發現嚴重弱點時進行掃描和更新。

Docker 桌面執行使用 Snyk 的本機掃描。它也可以用來尋找開放原始碼程式庫中的弱點和潛在授權問題。它可以直接整合到開發人員工作流程中，讓您能夠減輕開放原始碼程式庫所造成的風險。如需詳細資訊，請參閱下列主題：

- 開源應用程序安全工具包含用於偵測應用程式中弱點的工具清單。
- Docker 掃描備忘表

執行靜態程式碼分析

您應該在構建容器映像之前執行靜態代碼分析。它會針對您的原始程式碼執行，並用來識別可能被惡意程式碼的程式碼，例如錯誤注入。SonarQube是靜態應用程序安全測試（SAST）的熱門選項，並支持各種不同的編程語言。

以非 root 使用者身分執行容器

您應以非根使用者身分來執行容器。根據預設，容器會以`root`使用者，除非`USER`指令包含在您的碼頭文件中。Docker 指派的預設 Linux 功能會限制可以執行為`root`，但只是微乎其微。例如，以`root`仍然不允許存取裝置。

作為您的 CI/CD 管道的一部分，您應該 lint Docker 文件來查找USER指令，如果它丟失，則構建失敗。如需詳細資訊，請參閱下列主題：

- Dockerfile-lint是 RedHat 的一個開源工具，可用於檢查文件是否符合最佳實踐。
- 哈多林是構建符合最佳實踐的 Docker 映像的另一個工具。

使用唯讀根檔案系統

您應該使用唯讀根檔案系統。在預設情況下，容器的根檔案系統是可寫入的。當您將容器設定為RO（只讀）根文件系統，它強制您明確定義可以保存數據的位置。這會減少您的攻擊面，因為除非特別授予權限，否則無法寫入容器的檔案系統。

> **Note**
>
> 擁有唯讀根檔案系統可能會導致某些作業系統套件發生問題，這些套件期望能夠寫入檔案系統。如果您打算使用唯讀根檔案系統，請事先徹底測試。

設定具有 CPU 和記憶體限制的任務 (Amazon EC2)

您應該設定具有 CPU 和記憶體限制的工作，以將下列風險降至最低。任務的資源限制為任務中的所有容器可保留的 CPU 和內存量設置上限。如果未設定任何限制，則工作可以存取主機的 CPU 和記憶體。這可能會導致部署在共用主機上的工作可能會使系統資源的其他工作餓死的問題。

> **Note**
>
> 開啟 Amazon ECSAWS Fargate工作會要求您指定 CPU 和記憶體限制，因為它會使用這些值進行計費。Amazon ECS Fargate 的一項任務佔用所有系統資源並不是問題，因為每項任務都是在自己的專用執行個體上執行的。如果您沒有指定記憶體限制，Amazon ECS 會為每個容器配置至少 4MB。同樣地，如果沒有為任務設定 CPU 限制，則 Amazon ECS 容器代理程式會為其指派至少 2 個 CPU。

搭配 Amazon ECR 使用不可變標籤

使用 Amazon ECR，您可以且應該使用具有不可變標籤的映像設定。這可防止將變更或更新版本的影像推送到具有相同標籤的影像儲存庫。這可以防止攻擊者在使用相同標籤的映像上推送受影響版本的映像。通過使用不可變標籤，您可以有效地強制自己為每次更改推送帶有不同標籤的新映像。

避免以特許身分執行容器 (Amazon EC2)

您應該避免以特權身份運行容器。對於背景，容器以`privileged`會在主機上以延伸權限執行。這意味著容器繼承了分配給`root`在主機上。它的使用應該受到嚴格限制或禁止。我們建議設定 Amazon ECS 容器代理程式環境變數`ECS_DISABLE_PRIVILEGED`至`true`來防止容器作為`privileged`在特定主機上，如果`privileged`不需要。或者，您可以使用AWS Lambda掃描您的工作定義，以便使用`privileged`參數。

> **Note**
>
> 將容器運行為`privileged`不支援 Amazon ECSAWS Fargate。

從容器中移除不必要的 Linux 功能

以下是指派給 Docker 容器的預設 Linux 功能清單。如需每個功能的詳細資訊，請參閱Linux 功能概觀。

```
CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_FOWNER, CAP_FSETID, CAP_KILL,
CAP_SETGID, CAP_SETUID, CAP_SETPCAP, CAP_NET_BIND_SERVICE,
CAP_NET_RAW, CAP_SYS_CHROOT, CAP_MKNOD, CAP_AUDIT_WRITE,
CAP_SETFCAP
```

如果容器不需要上面列出的所有 Docker 內核功能，請考慮將它們從容器中刪除。如需每個 Docker 核心功能的詳細資訊，請參閱核心功能。您可以執行下列操作，找出正在使用中的功能：

- 安裝 OS 套裝服務程式碼封面-NG並執行`pscap`公用程式來列出每個處理程序正在使用的功能。
- 您也可以使用Capsh來解密進程正在使用哪些功能。
- 請參閱Linux 功能以了解詳細資訊。

使用客戶受管金鑰 (CMK) 加密推送至 Amazon ECR 的映像

您應該使用客戶受管金鑰 (CMK) 來附加推送到 Amazon ECR 的映像。推送到 Amazon ECR 的映像會在靜態時自動使用AWS Key Management Service(AWS KMS) 受管金鑰。如果您寧願使用自己的金鑰，Amazon ECR 現在支援AWS KMS使用客戶管理的金鑰 (CMK) 加密。使用 CMK 啟用伺服器端加密之前，請先檢閱靜態加密。

執行時間安全

執行階段安全提供容器執行時的主動保護。這個想法是檢測並防止惡意活動發生您的容器。

使用安全運算 (seccomp），您可以防止容器化應用程序對底層主機操作系統的內核進行某些 syscalls。雖然 Linux 作業系統有幾百個系統呼叫，但其中大部分是不需要執行容器。藉由限制容器可以製作哪些 syscalls，您可以有效地減少應用程式的攻擊面。

要開始使用 seccomp，您可以使用`strace`來生成堆棧跟踪，以查看您的應用程序正在進行哪個系統調用。您可以使用如`syscall2seccomp`從堆疊追蹤收集的資料建立 seccomp 設定檔。如需詳細資訊，請參閱「」搦淺和系統呼叫 2 秒。

與 SELinux 安全模組不同，seccomp 無法彼此隔離容器。但是，它可以保護主機內核免受未經授權的系統調用。它通過攔截系統調用，並只允許那些被允許列出的系統調用。Docker 具有一個defaultseccomp 設定檔，適用於大多數一般用途工作負載。

> **Note**
>
> 您也可以為需要其他許可的事物建立自己的設定檔。

ArperMAppArmor 是一個類似於 seccomp 的 Linux 安全模塊，但它限制了容器的功能，包括訪問文件系統的部分。它可以在`enforcement`或`complain`模式。由於建立 ApperAppArmor 定檔可能具有挑戰性，所以建議您使用像bane。如需 HarbAppArmor 的詳細資訊，請參閱官方AppArmor(憑證已建立！) 頁面上的名稱有些許差異。

> **Important**
>
> 虛 AppArmor 只適用於 Linux 的 Ubuntu 和 Debian 發行版。

Recommendations

我們建議您在設定執行階段安全性時採取下列動作。

為執行階段防禦使用第三方解決方案

為執行階段防禦使用第三方解決方案。如果您熟悉 Linux 安全性的運作方式，請建立和管理節目和虛擬 AppArmor 定檔。兩者都是開源項目。否則，請考慮改用不同的協力廠商服務。大多數使用機器學習功能來封鎖或警示可疑活動。如需可用的協力廠商解決方案清單，請參閱AWS Marketplace適用於容器。

使用節段原則新增或移除 Linux 功能

使用 seccomp 可更好地控制 Linux 功能，並避免系統呼叫檢查錯誤。Seccomp 的運作方式是系統呼叫篩選器，可撤銷執行特定系統呼叫或使用特定系統呼叫的權限。

AWS合作夥伴

您可以使用以下任一AWS合作夥伴產品，為您的 Amazon ECS 工作負載新增額外的安全性和功能。如需詳細資訊，請參閱「」Amazon ECS 合作夥伴。

Aqua Security

您可以使用Aqua Security來保護從開發到生產環境的雲端原生應用程式。Aqua Cloud 原生安全平台與您的雲端原生資源和協調工具整合，提供透明且自動化的安全性。它可以即時防止可疑活動和攻擊，並協助執行政策並簡化法規遵循。

Palo Alto Networks

Palo Alto Networks為雲端和整個開發和軟體生命週期中的主機、容器和無伺服器基礎架構提供安全性和保護。

扭鎖是由帕洛阿爾托網絡提供，並且可以與亞馬遜 ECS FireLens 集成。有了它，您就可以存取高準確度的安全性記錄檔和事件，這些事件無縫彙總為數個AWS服務。這些包括 Amazon CloudWatch，Amazon Athena 和 Amazon Kinesis Kinesis。Twistlock 可保護部署在AWS容器服務。

西斯迪格

您可以使用西斯迪格，在生產案例中執行安全且符合規範的雲端原生工作負載。Sysdig Secure DevOps 平台具有內嵌的安全性和合規性功能，可保護您的雲端原生工作負載，並提供企業級的擴充性、效能和自訂功能。

Amazon ECS 最佳實務指南的文件歷程記錄

下表說明 Amazon ECS 最佳實務指南的文件版本。

update-history-change	update-history-description	update-history-date
安全最佳實務	為 Amazon ECS 工作負載新增安全管理的最佳實務。	2021 年 5 月 26 日
自動擴充和容量管理最佳實務	為 Amazon ECS 工作負載新增自動擴展和容量管理的最佳實務。	2021 年 5 月 14 日
永久儲存最佳實務	為 Amazon ECS 工作負載新增永久儲存的最佳實務。	2021 年 5 月 7 日
網路最佳實務	為 Amazon ECS 工作負載新增聯網管理的最佳實務。	2021 年 4 月 6 日
初始版本	Amazon ECS 最佳實務指南的初始版本	2021 年 4 月 6 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。