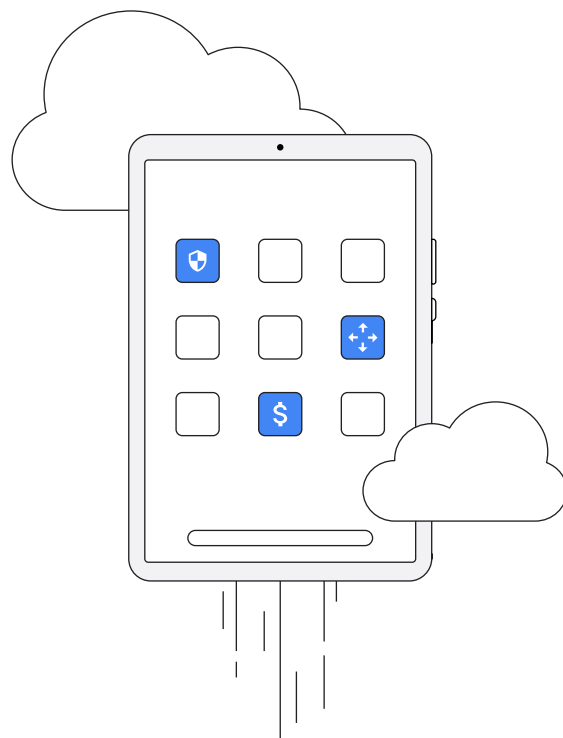


容器化是基礎架構 的未來趨勢

科技和新創公司紛紛迅速採用代管容器平台。透過這樣的平台，這些公司能在維護基礎架構上投入較少工程資源，將更多資源投入發展藍圖中的優先事項，藉此改善業務表現，例如推動成長、取得競爭優勢和提高獲利能力等。



如何減少基礎架構管理工作，
加速上市

第 2 頁

為何應考慮使用採開放標準
的代管服務

第 5 頁

如何讓遷移發揮最大效益

第 9 頁

第 1 章

如何減少基礎架構管理工作， 加速上市

雖然現今的科技和新創公司大多都在雲端上營運，但只有少數實際發揮這麼做的所有優勢。如果您在雲端上營運，但不使用 [Kubernetes](#)，您大概是採用特定廠商提供的解決方案，輔以自己建構及維護的自訂工具。此外，由於執行工作負載的 [虛擬機器](#) (VM) 使用率偏低，因此也有效率不彰的情形。您也可能面臨平台制約的問題。

在管理及修補 VM 方面，您目前使用幾項工具？

您如何升級應用程式？VM 使用情況為何？您目前的做法可能不是很有效率。您的 VM 架構有弱點，導致系統失效 (發生服務中斷或擴充性問題等)，成本超出您的掌控，或是基礎架構無法因應以下業務需求：

- 將最小可行性產品 (MVP) 重構或重建為可擴充的解決方案
- 擴大採用其他雲端服務供應商，符合法規或客戶期望
- 擴展服務的地理區域，縮短延遲時間，為更廣大的客群提供更優質的體驗
- 加強端對端安全防護機制
- 改善客戶體驗 (例如服務可用性)

舊有架構和技術債可能會拖累業務發展，因此我們發現許多機構都改用 Kubernetes。由代管容器組成的現代化架構可讓您掌握經過實證的模式，使基礎架構能安全可靠地運作。您可以加快上市速度並提高生產效率，同時兼顧穩定性和安全性，且能吸引最優秀的技術人才投入創新工作。

您也須注意自己是否受制於特定系統，而無法加入 Kubernetes 社群及相關的體系。這個社群與體系的創新步伐相當穩定，奠定了今日的業界標準和最佳做法。此外，人才招聘方面目前面臨多項挑戰，但有一件事在這些挑戰下更顯重要，您必須決定工程師最終應把時間運用在哪種工作上：是要維護基礎架構同時建構及維護自訂工具，還是要完成貴機構的優先要務，推動業務成長。您目前使用的架構或許可以運作，但在規劃發展藍圖時，您可能會希望未來不會出現技術債和下列跨平台問題：

- 端對端加密
- 觀測能力 (記錄檔、指標、自動記錄功能)
- 政策管理及執行
- 高可用性和自動容錯移轉
- 降低成本

Kubernetes 是採開放原始碼且各平台通用的服務，提供各項常見工具，且全都立即可用，能保護這項服務結合多數系統管理員的心血結晶，將日積月累的 bash 指令碼和最佳做法全部納入單一系統，方便您 API 存取。各項流程均採自動化機制，介面經過簡化，沒有繁複細節，且立即可用。Kubernetes 可以免除「程式碼式基礎架構」的大多數問題，同時將您的平台轉換為「資料式基礎架構」。您不必編寫或維護程式碼，也不需詳述做法，只要告訴 Kubernetes 您的目標即可，因此能節省大量的管理時間。

容器是實現連續運算架構的最佳方法



如果想要執行傳統工作負載，可先將應用程式與 VM 分開，並放到容器中，再透過 Kubernetes 在現代化平台上執行。利用容器映像檔封裝軟體，可簡化 VM 的升級作業。您現在可以拆分 VM 和應用程式的生命週期管理作業，移除 Ruby、Python 和 Java 等項目來簡化 VM，並移至合適的位置，即開發人員應用程式附近，以便集中一處掌控各項作業，並將機器保持在裸機狀態。



代管型運算平台可將雲端服務轉化為平台式服務 (PaaS)，讓您享有容器的強大功能與彈性，以及無伺服器的便利。不必管理伺服器、設定叢集和進行維護，不僅保有掌控權，也能同時獲得莫大優勢。



如果不太需要控制工作負載的叢集設定，您可以讓服務佈建及管理叢集的基礎架構，包括節點和節點集區。這樣一來，您只須支付工作負載的費用，不必針對叢集付費，免除管理叢集的麻煩、改善安全性，還可能節省大量費用。



如果應用程式偏向雲端原生，無伺服器模式能幫助您更輕鬆地達成相同的效果，且不需基礎架構，還可用來端對端託管應用程式、資料和分析功能。無伺服器平台可讓您以最簡單的方式在全代管環境中執行容器，同時享有安全性、效能和擴充性，並遵循最佳做法。

以下將說明您的業務可能受到的影響，以及讓貴機構效率領先的方法。

第 2 章

為何應考慮使用採開放標準的代管服務

如果您有多雲端需求，不妨運用可簡化相關事務的技術

部分企業認為多雲端營運模式是必要的解決方案。資料引力確實存在，如果您的客戶遍布全球，您一定會發現，有些客戶希望能在實際存放資料的位置附近執行運算作業，盡可能降低延遲時間和網路費用。

在這種情況下，多雲端可以拓展您的潛在市場，但，您最後都必須支援其他供應商的代管服務和資料。也有公司將多雲端視為降低風險的策略，無論是何種情況，關鍵都在於正確採用多雲端模式，而符合業界標準的解決方案這時就能派上用場。



01

首先：**確保您能盡可能重複運用各項工作流程**。工作流程是指啟用工作自動化功能，在資料庫與運算作業間建立資料流程架構。因此，使用開放原始碼相當重要，如果您選取的資料庫並非開放原始碼資料庫，就會難以導入資料流程架構和自動化功能。建議您在任何雲端供應商提供的服務上，使用 Postgres 通訊協定 (例如 [Cloud Spanner](#)) 或代管的 Postgres 資料庫 (例如 [Cloud SQL](#))。

02

其次：在運算方面，Kubernetes 可大幅節省部署和自動化的時間，因此在挑選技術時，**請確保這些技術能跨越供應商設定的界線運作**，像是跨區域或可用區運作，或是在不同的專案或帳戶中，甚至是地端及雲端部署環境中運作。您不必浪費時間為每個雲端供應商的服務建構及維護個別的基礎架構 (例如在 AWS、Google 和 Azure 各採用一個基礎架構)，光是要設法讓這些基礎架構保持一致，就會讓工程師不堪負荷。只要建構一次並部署至多個雲端環境，需要時就能以集中的方式統一更新。Kubernetes 這類運算堆疊可為客戶帶來極大優勢，在採用多雲端模式時可享有高效率，而且在加入新的雲端服務供應商時也不需重新規劃。

03

第三：**風險管理**。如果可在其他環境中執行堆疊，當雲端服務供應商中斷服務或與您的業務有競爭時，您就能降低風險。為了遵守法規，機構會挑選供應商，確保業務能夠持續執行。舉例來說，如果有後備服務供應商，即便某個區域的服務中斷，也不會停機。

採用開放標準的多雲端遷移作業，效果往往相當良好。以 Kubernetes 為例，這項服務提供各供應商通用的 API，可執行、設定及部署應用程式，並整合安全性政策和網路等。可將 Kubernetes 想像成一個多雲端作業系統，這項服務可做為「抽象層」，把多數主要雲端服務供應商間的繁雜差異隱藏起來。

採用全代管服務，降低營運負擔

Kubernetes 的使用方式十分彈性，首先您當然可以選擇自己執行 Kubernetes。Kubernetes 是開放原始碼專案，您可以自行下載，並花數年的時間整合至雲端供應商的服務或偏好的環境中。

不過，如果您覺得這麼做無法妥善利用時間，則可考慮使用代管的 Kubernetes 服務。如果您使用的是 AWS，可以採用 EKS；如果您使用的是 Azure，可以採用 AKS；如果您使用 Google Cloud，則可採用 [Google Kubernetes Engine \(GKE\)](#)。上述所有選項都提供通用的 Kubernetes API，因此在您的團隊建構好自己的工具與工作流程後，您可以在不同的供應商之間重複使用這些工具和工作流程。

不過，並不是所有代管服務都相同。Kubernetes 的成效取決於採用的基礎架構，而 GKE 是成熟的全代管 Kubernetes 自動化調度管理服務，可彌補基礎架構中的不足。這項服務提供完全整合的 IaaS，範圍涵蓋佈建 [Tau VM](#) (成本效益比同類的一般用途產品高 42%)¹、跨多個可用區自動調度資源、升級，以及建立及管理 GPU 和 TPU，方便您用於機器學習、儲存空間磁碟區與安全性憑證。您只需要將應用程式放到容器中，並根據需求選擇系統即可。

如果您已選擇 AWS 的 VM 雲端服務，該怎麼做？是否需要繼續採用 EKS？整體來說，Kubernetes 的功能對所有雲端服務供應商來說並無差異，都是採用 Kubernetes API，但其中還有叢集、工作站節點、安全性政策等各個項目。在這方面，GKE 就明顯具有優勢。

舉例來說，如果您仍需其他叢集，可以連結至 [GKE Connect](#)，在同一個地方管理、查看所有叢集，並進行疑難排解和偵錯，也可集中管理憑證等。GKE 是領先業界的 Kubernetes 服務，不僅控制層、多區域或多可用區皆具高可用性，端對端的管理能力更令人稱道。GKE 也可以使用集中代管的 [多叢集 Ingress](#) 服務，在多個叢集及區域中運用全域負載平衡器。

如果您有意使用 Kubernetes API，但卻不想佈建、升級叢集及調動資源呢？對於大多數工作負載來說，使用 [GKE Autopilot](#) 就不需完成叢集基礎架構的相關作業 (包括節點和節點集區)，而且只需支付工作負載的費用。GKE Autopilot 能為您提供標準的 Kubernetes API，這個 API 有強大的預設安全性設定，可讓您專注處理工作負載，不必為叢集費心。

¹結果是依據 SPECrate@2017_int_base 在以下環境中執行的預估值：另外兩大雲端供應商的實際工作環境 VM，以及正式發布前的 Google Cloud Tau VM (三個環境均使用供應商建議的編譯器)。SPECrate 是 Standard Performance Evaluation Corporation 的商標。詳情請見 www.spec.org。

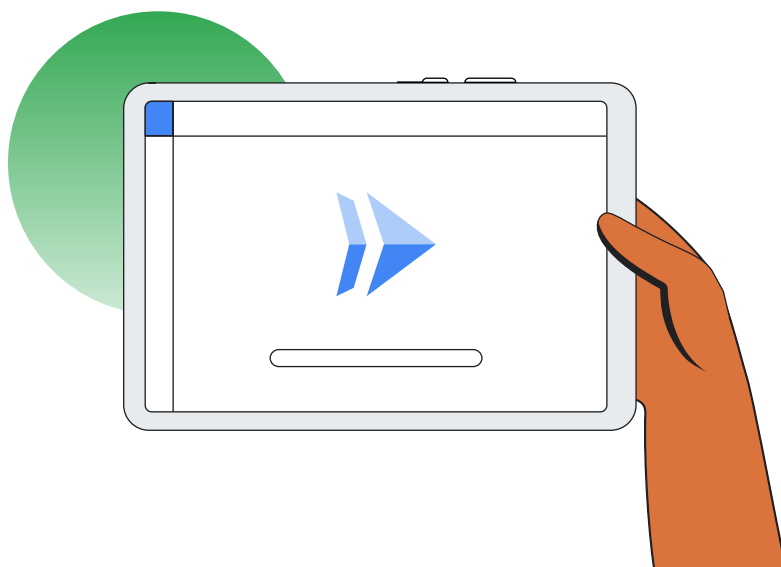
運用無伺服器技術簡化應用程式供應程序

Kubernetes 將機構從 VM 移到一組新的抽象化機制中，方便您將作業自動化並專注在應用程式上。不過對於特定工作負載 (例如網路和行動應用程式、REST API 後端、資料處理、工作流程自動化)，使用無伺服器模式可進一步簡化部署項目，並調整到最佳狀態。

假設您使用的是 AWS Lambda，這個熱門的無伺服器平台可編寫「函式即服務」，並連結至各種事件。但由於您最終會連線至資料庫，且需處理安全疑慮，因此這些函式可能會變得越來越複雜，部分函式甚至比一般應用程式還要大。如果應用程式已無法透過簡易的函式即服務處理，或是想以無伺服器方式執行現有的應用程式，又該怎麼做？

不同於需要重新編寫應用程式的傳統無伺服器平台，[Cloud Run](#) 提供的方法可讓您重複使用先前投資的現有容器化應用程式。雖然 GKE 是代管服務，但您還是必須做出一些重要決定：要在哪些可用區執行應用程式、儲存記錄檔的位置、如何管理不同應用程式版本間的流量、註冊網域名稱，以及負責管理安全資料傳輸層 (SSL) 憑證。

Cloud Run 讓您不用做出這些決定，即可執行更多傳統工作負載，並完全停用「將資源調度率降至零」功能，避免冷啟動。如果您的應用程式必須持續執行，Cloud Run 也能助您一臂之力，且支援其他傳統要求，例如 NFS、WebSocket 和虛擬私有雲整合功能。但與大多數傳統的無伺服器平台一樣，Cloud Run 結構明確，且提供內建流量管理和自動調度資源等功能。



第 3 章

如何讓遷移發揮最大效益

思考您的遷移方式

假設您完全沒有使用容器，且想知道該從何處著手，歡迎參考下列實用做法，瞭解如何容器化。

01

採用容器的第一個理由是為了解決封裝問題。如今有許多工作與下列用途有關：創造可重現的成果、瞭解軟體內部的情況，或是業界所謂的「[軟體安全供應鏈](#)」。我們認為達成這個目標的理想方法，就是利用容器映像檔保存應用程式的程式碼、執行階段及依附元件。容器的好處是可以部署至 VM、降低跨機器部署及維護軟體的複雜性。

03

採用容器的第三個理由是效率。除了維護工作之外，多數機構的 CPU 使用率只有 5% 至 10%，記憶體和儲存空間的使用率更低，因而浪費大量資源。許多團隊都透過自動調度資源和分裝等方式建構更多自訂工具，希望能縮小差距，但也耗費更多作業時間，導致費用超支，並帶來昂貴的雲端帳單。

02

採用容器的第二個理由是自動化調度管理。管理 VM 會帶來大量的管理和維護開銷，如果您像大多數公司一樣，團隊須執行數十或數百個步驟管理基礎架構，即使採用自動化機制，仍需持續維護自動化工具。以上情況假設您使用 Terraform、Puppet、Chef 和 Ansible 等業界標準自動化工具，如果您是使用自行開發或自訂的工具，維護開銷可能更高。

很少有機構能成功建立自己的自動化調度管理平台，因此許多機構都選用 Kubernetes、Mesos 或 Nomad 等開放原始碼平台。不過，如果您希望平台能為您帶來某些助益，例如減少維護事務、採用業界標準最佳做法，以及與其他雲端服務供應商深度整合，就必須採用 GKE 等代管服務，發揮容器的潛在價值。



順利完成遷移作業的基礎概念

此時，您可能會有疑問：遷移至容器真的需要長時間停機，且完全沒有價值嗎？您最不想要的就是暫停日後的開發作業，對吧？

為了回答這個疑問，我們來看看該如何利用您目前累積的雲端經驗，讓遷移作業更加順利。

將應用程式從 VM 遷移至容器，對您而言可能並非易事，如果期間還必須把所有運算資源遷移至新的雲端服務供應商，則又更加困難。但其實您可以先從 VM 開始，一次遷移一個應用程式，再將一或兩個應用程式移到 Kubernetes，不必一次全部遷移。這樣一來，這些應用程式就能在同一個網路中同時運作並與 VM 通訊。您可以從一個平台逐漸移至另一個平台，不用一次完成全面性的遷移。

少數應用程式可能比較適合繼續使用 VM，例如無法有效運用 Kubernetes 的大型資料庫。不過這不成問題，您可以雙管齊下。但我們發現，將應用程式和開放原始碼專案移至 Kubernetes 環境後，大多數客戶都能獲益良多。您可優先遷移在移至 GKE 後會帶來最大效益的應用程式，不必一次移動所有項目。

您目前可能使用 Linux VM，而 Kubernetes 仍會在這些 Linux VM 上執行，有利您簡化及統一工作流程，整合您的基礎架構藍圖，省去一連串自行開發的指令碼和自動化功能。只要使用 Kubernetes 等業界標準工具，就能讓新進團隊成員更輕鬆地上手，不必費時指導所有特殊做法。

在這個階段中，您可以緩慢但務實地開始採用容器，為團隊省下處理管理作業的時間，以及運算相關的費用支出。

準備好邁出下一步了嗎？

我們探討了許多關於容器和雲端遷移作業的事項，無論您已全面採用 Kubernetes，只是想尋找更好的服務，或是希望能透過不同的做法，滿足現有應用程式的需求，並支援未來發展，都能在本文中找到答案。無論您採取的是何種應用程式開發途徑，選用代管容器即可大幅降低基礎架構費用，同時幫助團隊充分發揮能力，專心建構出色產品。容器化是基礎架構的未來趨勢，您應確保您的工程師和企業擁有所需的基礎架構與工具，獲得最大成功。



Google Cloud 領先業界的代管服務能協助您充分發揮現代化應用程式的效益，如要進一步瞭解，歡迎與我們聯絡。

與專家聯絡



延伸閱讀

- [以 Kubernetes 實現雲端原生架構：快速展開您的 GKE 開發旅程](#)
- [雲端中的容器：更有效率的應用程式開發及部署方式](#)
- [與 Eric Brewer 共同探討 Kubernetes 的過去、現在與未來](#)
- [以雲端原生方式重新建立架構：這是大規模提高開發人員生產力的革命性方法](#)

