

4.1 現代化雲端基礎架構 (Modernizing Infrastructure in the Cloud)

4.1.1 在雲端運行計算工作負載的好處 (The benefits of running compute workloads in the cloud)

- **總擁有成本 (TCO)：** 降低 IT 成本，按需付費，長期承諾有折扣。
- **可擴展性 (Scalability)：** 根據需求彈性調整資源，無需大量前期投資。
- **可靠性 (Reliability)：** 高度可靠性和正常運行時間，多個數據中心和自動監控修復。
- **安全性 (Security)：** 除了實體數據中心安全外，還包括數據加密、身份和訪問管理、網絡安全、虛擬私有雲和監控服務。
- **靈活性 (Flexibility)：** 可隨時選擇和調整最符合需求的雲服務。
- **抽象化 (Abstraction)：** 雲端供應商管理硬體、軟體以及部分安全和網絡，簡化使用者操作。
- **加速上市時間 (Faster time to market)：** 無需自行開發和維護基礎設施。
- **創新平台 (Platform for innovation)：** 提供最新的技術和工具。

4.1.2 虛擬機 (Virtual machines)

- **虛擬化 (Virtualization)：** 允許多個系統 (虛擬機/VM) 在同一硬體上運行，共享資源。
- **Compute Engine：** Google Cloud 的基礎設施即服務產品，用於創建和運行虛擬機，無需前期投資。
- 每個虛擬機都包含完整操作系統的功能。
- 可通過 Google Cloud 控制台、Google Cloud CLI、Terraform 或 Compute Engine API 創建虛擬機實例。
- **計費方式：** 按秒計費，最低一分鐘，持續使用有折扣。
- **承諾使用折扣 (Committed use discounts)：** 承諾使用一年或三年可享折扣。
- **可搶佔/現貨虛擬機 (Preemptible/Spot VMs)：** 成本較低，但可能被系統終止，適合可中斷和重啟的工作負載。現貨虛擬機比可搶佔虛擬機提供更多功能，例如沒有最大運行時間限制，但目前價格相同。
- 可選擇預定義或自定義機器類型。

4.1.3 容器 (Containers)

4.1.4 管理容器 (Managing containers)

- **Kubernetes：** 開源平台，用於管理容器化工作負載和服務，方便協調、擴展、部署和回滾。
- **Google Kubernetes Engine (GKE)：** Google 托管的 Kubernetes 服務，由計算引擎實例組成的集群，可自定義機器類型、節點數量和網絡設置，提供 API 和 Web 控制台。

- **GKE Autopilot**：全面管理集群基礎設施並提供按 Pod 計費的模式。
- **Cloud Run**：完全管理的無伺服器平台，用於部署和運行容器化應用程式，自動處理擴展和管理，適合無狀態應用程式。
- **GKE vs Cloud Run**：GKE 適合需要大量控制和運行複雜應用程式的情況；Cloud Run 適合需要簡單、完全管理、可快速擴展和縮減的無伺服器平台的情況。
- **案例**：Ubie 使用 GKE Autopilot 降低了 20% 的基礎設施成本，並簡化了維護和升級任務。

4.1.5 無伺服器運算 (Serverless computing)

無伺服器運算的定義

- 無伺服器不代表沒有伺服器，而是資源會根據需要在背景自動配置，使用者只需在運行查詢或應用程式時付費。
- 企業提供功能代碼，雲端供應商負責其他所有事情。
- **函數即服務 (Function as a service)**：一些功能是對特定事件的響應。
- 無伺服器運算的優勢：
 1. **降低運營成本**：基礎設施和維護由雲提供商負責，應用程式所有者無需投入基礎設施和管理資源。
 2. **自動擴展**：根據應用程式需求自動擴展計算資源，雲提供商管理擴展過程，應用程式所有者只需支付實際使用的資源費用。
 3. **更快的上市時間**：消除了對基礎設施設置和配置的需求，減少了部署應用程式所需的時間，使應用程式所有者可以專注於編寫代碼和快速部署新功能。
 4. **減少開發成本**：開發過程簡化，開發人員可以專注於應用程式邏輯而不是底層基礎設施。
 5. **提高彈性和可用性**：雲提供商自動管理基礎設施的故障轉移和災難恢復能力。
 6. **按使用付費的定價模式**：應用程式所有者只需為使用的計算資源付費，減少未使用資源的成本並幫助優化成本。
- **案例**：[Mashme.io](#) 使用 Google Kubernetes Engine 降低延遲，並實現持續集成和部署。

總結：這份教材詳細說明了多種在 Google Cloud 上運行和管理應用程式的方式，包含虛擬機、容器和無伺服器運算，並比較了它們的優缺點和適用情境。

4.2 現代化雲端應用程式 (Modernizing Applications in the Cloud)

4.2.1 現代雲應用程式開發的優勢 (The benefits of modern cloud application development)

- **架構 (Architecture)：** 使用微服務架構，獨立部署、可擴展和可維護的組件，加速產品上市。
- **部署 (Deployment)：** 部署到雲端，可使用管理或部分管理的服務，釋放人力資源。
- **成本 (Cost)：** 按需付費，有效利用資源，開發人員可使用預構建的 API 和其他工具加速開發和部署。
- **可擴展性 (Scalability)：** 可輕鬆擴大或縮小以滿足用戶需求。
- **高可用性和彈性 (High availability and resilient)：** 內建負載均衡和自動故障轉移，提供強大的監控和管理工具。

4.2.2 在雲端重新託管舊式應用程式 (Rehosting legacy applications in the cloud)

- **重新託管 (Rehosting/Lift and shift)：** 將應用程式從本地環境移動到雲環境，而不做任何更改。
- **優點：** 繼承雲計算的優勢，如成本節省、可擴展性、可靠性和安全性。
- **缺點：** 複雜性、風險和供應商鎖定。
- **Google Cloud 解決方案：**
 - **Google Cloud VMware Engine：** 將現有的 VMware 工作負載遷移到雲端，無需重新設計應用程式或重新調整操作，並可使用其他 Google Cloud 服務。
 - **Bare Metal Solution：** 在專用裸機伺服器上運行 Oracle 工作負載。

4.2.3 應用程式介面 (APIs)

- **API 定義：** 一組讓不同軟體程式相互通信的指令集，提供標準化和可預測的數據交換和互動方式。
- **API 的應用：** 從社交媒體平台到移動應用和網路服務，節省開發時間和精力。
- **Google 提供的 API：** 例如網站搜索、Google 地圖數據和翻譯等。
- **API 的商業機會：**
 - 創建新的產品和服務。
 - 產生新的收入流。
 - 創建合作夥伴關係。

4.2.4 Apigee API 管理 (Apigee API Management)

- **Apigee：** Google Cloud 的 API 管理服務，提供增強的規模、安全性和自動化。
- **Apigee 的優點：**
 - API 安全性：身份驗證、授權和數據加密。
 - API 使用情況追蹤和分析：實時分析和歷史報告。
 - API 開發和部署：可視化 API 編輯器和測試沙箱。
 - API 版本管理、API 文檔和 API 限流。

- **案例：** AccuWeather 使用 Apigee 管理 API，客製化 API 消耗，通過可自定義的開發者門戶吸引開發者，並通過內置分析監控流量。

4.2.5 混合雲和多雲 (Hybrid and multi-cloud)

- **混合雲 (Hybrid cloud)：** 本地或私有雲基礎設施與公共雲服務的組合。
- **多雲 (Multi-cloud)：** 組織使用多個公共雲提供商。
- **選擇混合雲或多雲的原因：**
 - 整合特定公共雲的優勢。
 - 部分系統保留在本地，部分移至雲端，創建獨特適合組織需求的環境。
 - 只將特定工作負載移至雲端，受益於雲服務的靈活性、可擴展性和更低的計算成本。
 - 添加專門的服務，如機器學習、內容緩存、數據分析、長期存儲和物聯網。
- **Google Cloud 的解決方案：**
 - **GKE Enterprise：** 管理的生產就緒平台，用於在多個雲環境中運行 Kubernetes 應用程式，提供一致的方式來管理 Kubernetes、集群、應用程式和服務。
 - **GKE Enterprise 的優點：**
 - 多雲和混合雲支持：可在 Google Cloud、AWS、Azure 和其他公共雲上運行 Kubernetes 集群。
 - 集中管理：單一集中控制台管理 Kubernetes 集群、應用程式、安全性、合規性、網絡和負載均衡。
 - 監控和日誌記錄：提供豐富的工具，用於監控和維護整個網絡的應用程式一致性。

總結： 這份教材說明了現代化應用程式開發的各種方式，包含微服務、重新託管、API、API 管理，以及混合雲和多雲策略，並提供相關的 Google Cloud 解決方案和案例。