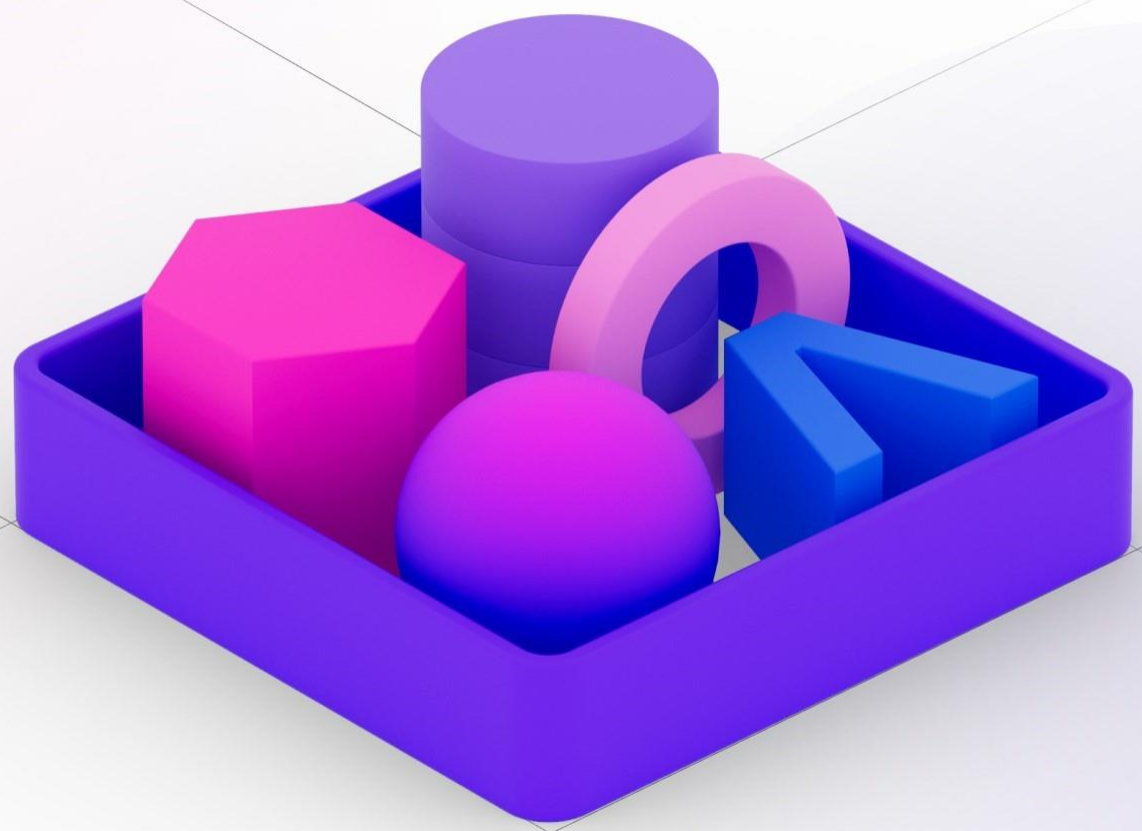
STUDY4
為 學 習 而 生

Build School

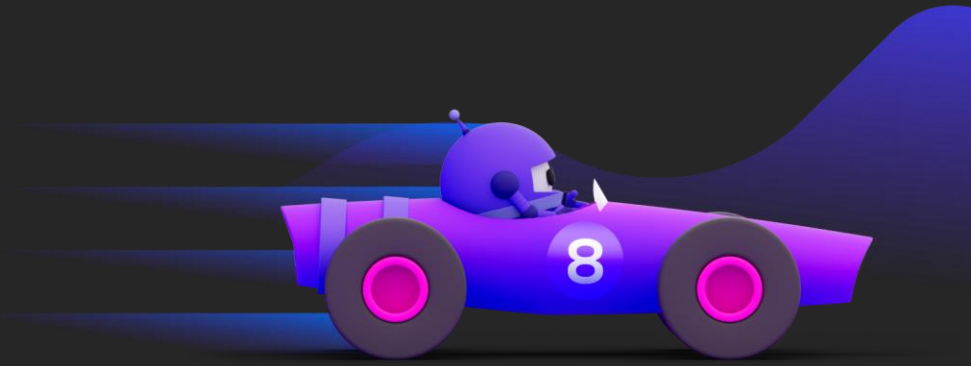
.NET Conf
TAIWAN



STUDY4 為學習而生 Build School

組合 Azure PaaS 的應用與看法

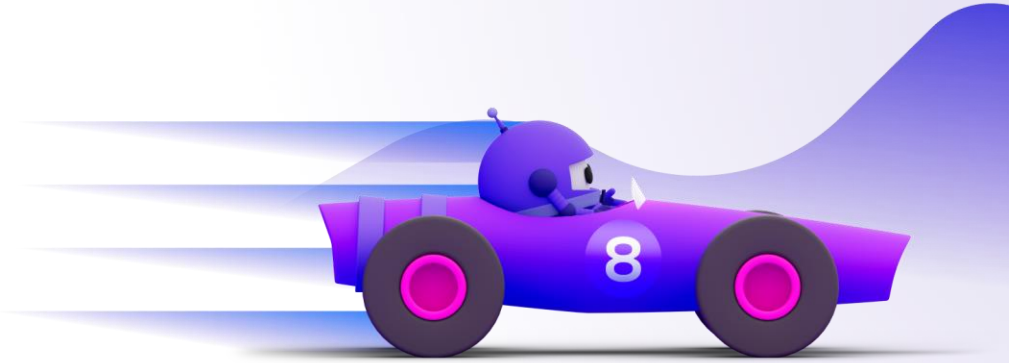
Edward Kuo



STUDY4 為學習而生 Build School

最適合開發人員的裝備

組合 Azure PaaS 的應用與看法



關於我

Kingston Technology Senior IT Manager
Microsoft Regional Director
Microsoft Azure MVP

- Agile Summit 講師
- DevOps Days Taipei 講師
- Microsoft 大型研討會 講師
- 大專院校 人工智慧 客座講師
- 企業 DevOps 講師
- 台灣微軟技術社群講師
- Azure Bootcamp 講師
- .NET Conf 講師

開始前...

- 普遍面對Cloud轉型起手式
- PaaS帶來另一種思維

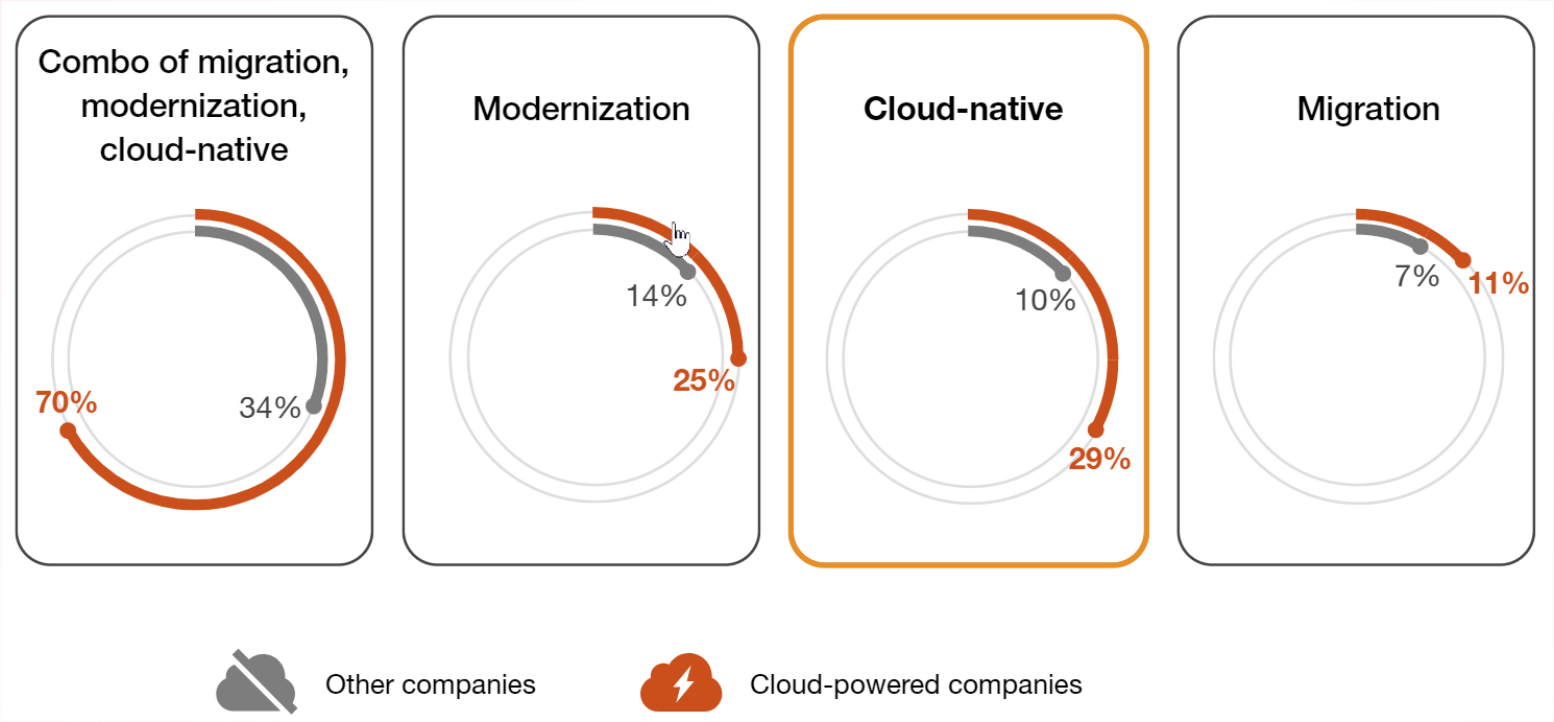
STUDY4 為學習而生 Build School

Cloud-Native意義

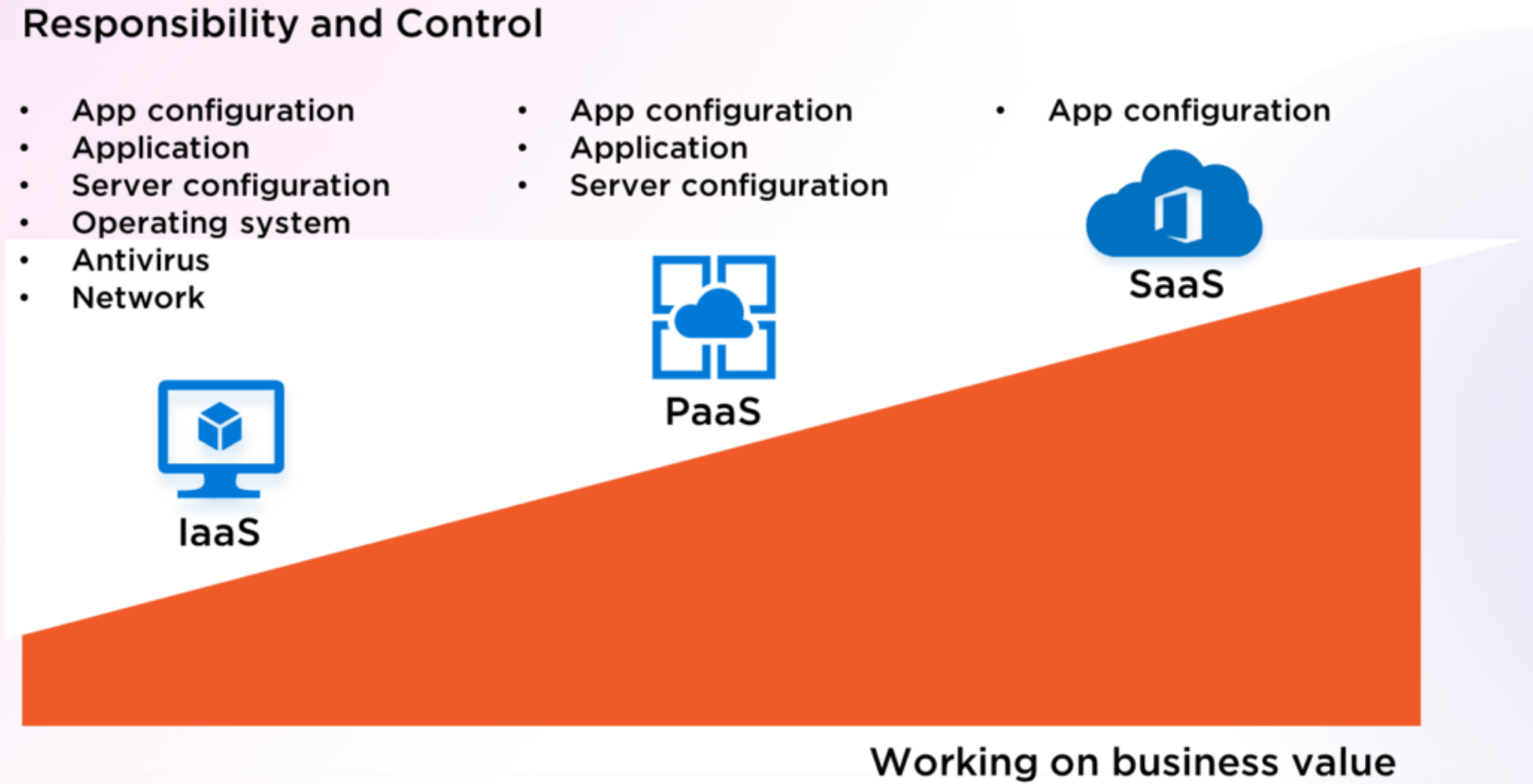
雲端應用要從IaaS觀念轉型 PaaS思維

雲端轉型成功關鍵

企業管理諮詢機構PwC，2023發布企業雲端調查報告（Cloud Business Survey），「企業上雲成功的關鍵其一是根據業務目標來決定採行何種上雲做法，而不依循舊有遷雲做法，先將工作負載搬到雲端，然後才開始進行現代化與發展雲原生」



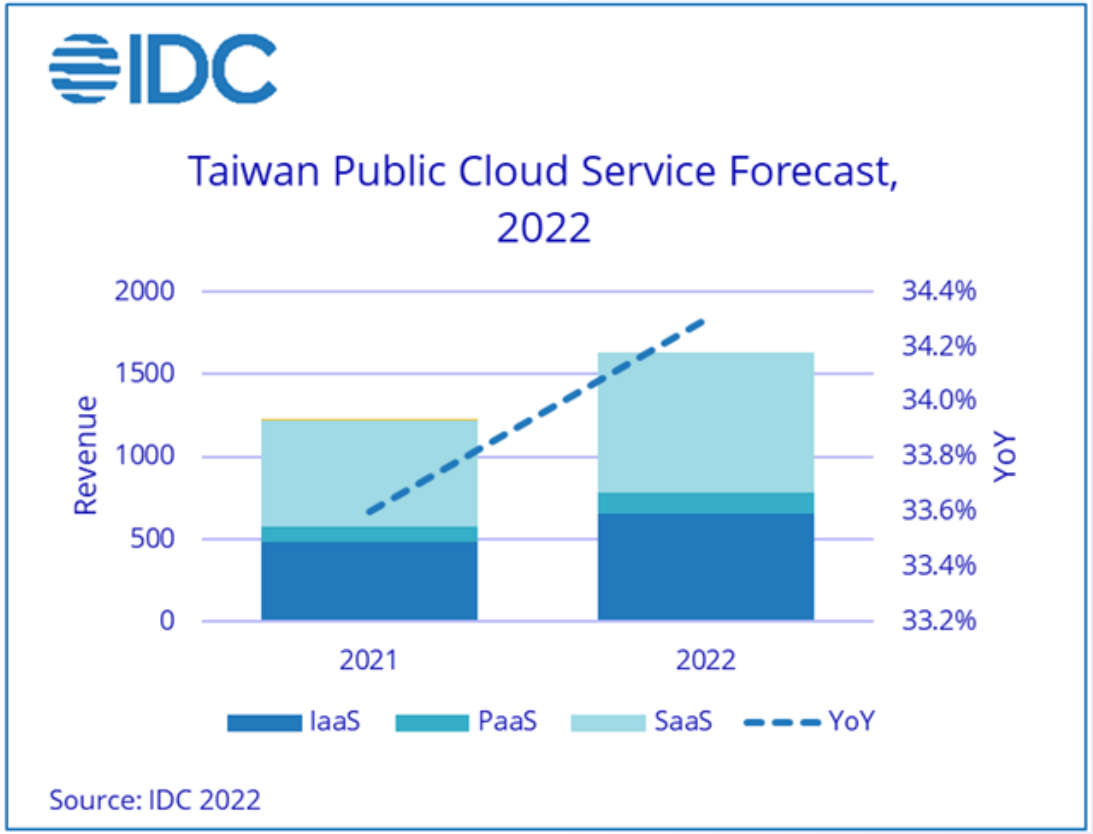
因致力於商業價值


Responsibility and Control
• App configuration
• Application
• Server configuration
• Operating system
• Antivirus
• Network
IaaS
• App configuration
• Application
• Server configuration
PaaS
• App configuration
SaaS
Working on business value

VM 遷移到雲=雲端轉型?

伺服器虛擬化從企業內移到雲端機房，多數把雲端機房做為自家機房的延伸

- 商業效益沒有增加，卻多出一筆雲端費用
- 應用系統在地端會發生的問題，雲端依舊存在
- 總體維運的壓力並沒有減少
- 沒有因上雲而增加商業的彈性和敏捷性
- 導致系統架構更加複雜沒彈性



Why VM模式占多數...

- 早期要推廣雲概念，最快方式就是推薦客戶採用雲端機房概念
- 雲廠商想讓企業短時間消耗雲經費，才有業績
- 雲廠商認為可以完全消滅企業端IT機房達到全雲化
- 多數雲導入廠商對於雲服務(PaaS)技術不熟，IaaS是最好販賣的商品
- 多數企業IT採購資源被掌握在Infra團隊，思維以VM角度觀念為主
- 開發者不想要重新建構架構或變更系統
- 許多非常老舊系統，已經無法改成PaaS，但被迫要直接運行在雲上

Cloud-Native 定義

- 雲原生指的是構建和運行在公共雲或混合雲環境中的應用系統
- 部署在雲提供的平台服務之上，無需管理底層基礎設施
- 強調以雲為中心的敏捷開發和部署模式，利用雲的彈性...等優勢，構建可靠、可擴展的應用系統

Cloud-Native 的效益

- 最大化運用雲端特性和優勢
- 應用程式設計以雲原生概念進行設計
- 最大程度發揮彈性、可擴展性和可靠性
- 關注商業需求、創意與快速達成企業願景

「寵物服務模型」和「牛群服務模型」

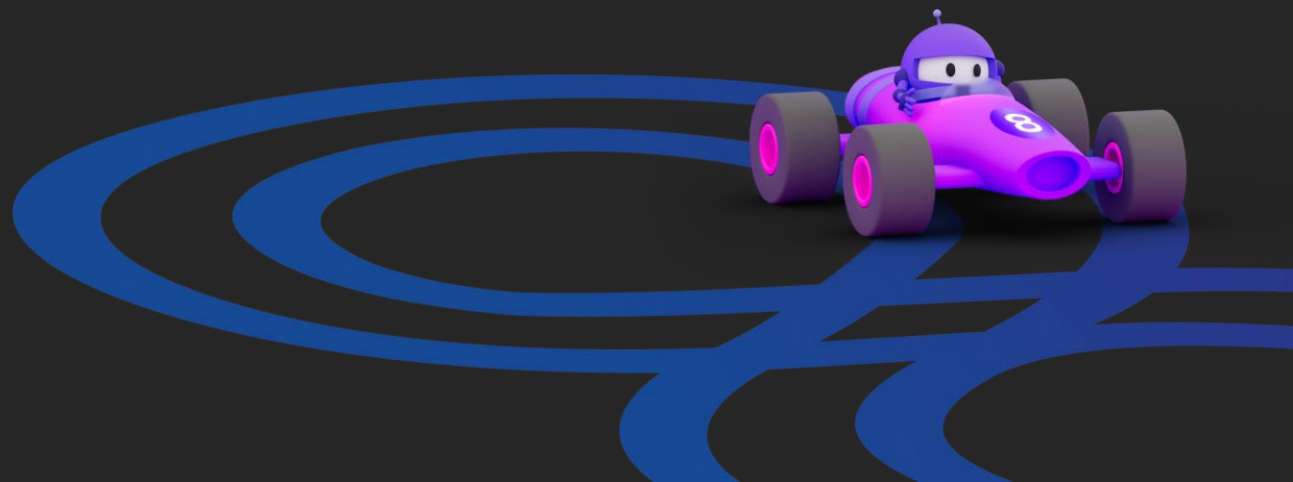
- 寵物服務模型：將伺服器視為寵物，每個伺服器都有獨特的名稱，需要個別照顧和修復。這些伺服器可以是主機、獨立伺服器、負載均衡器、資料庫系統等
- 牛群服務模型：將伺服器視為牛群，每個伺服器都有識別號，當一個伺服器出現問題時，可以被替換，而且伺服器之間幾乎相同。這些伺服器可以是網頁伺服器陣列、NoSQL集群、搜尋集群...等

Cloud-Native採用牛群服務模型，依據需求彈性縮減或擴增資源結構，不受所在機器的限制

STUDY4 為學習而生 Build School

Platform-as-a-Service 應用

我們建構全PaaS 架構的服務案例



Platform-as-a-Service優勢與契機

- 像積木，取得符合需求的服務進行組合
- 用PaaS為不同領域(服務)構建服務雲
- 專注邏輯和商業價值開發，不需要顧慮基礎設施的管理和維護
- 減少建置、安裝軟體和維護伺服器時間或成本
- 自動提供原生監控健康狀況的服務
- 降低自行裝載風險，和處理發生的問題
- 降低重建服務時間和試錯的風險與成本
- 降低建置、維運和管理基礎建設的人力

不捨棄企業機房

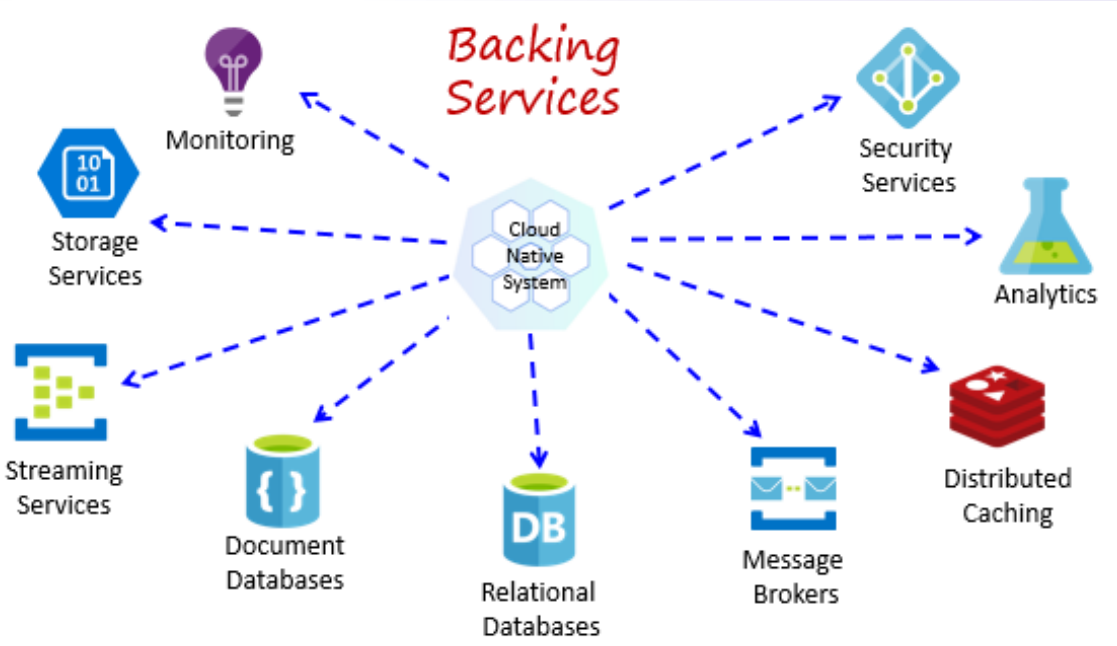
當你還有企業機房，什麼狀況
會讓你想使用雲端資源?

多數作法...

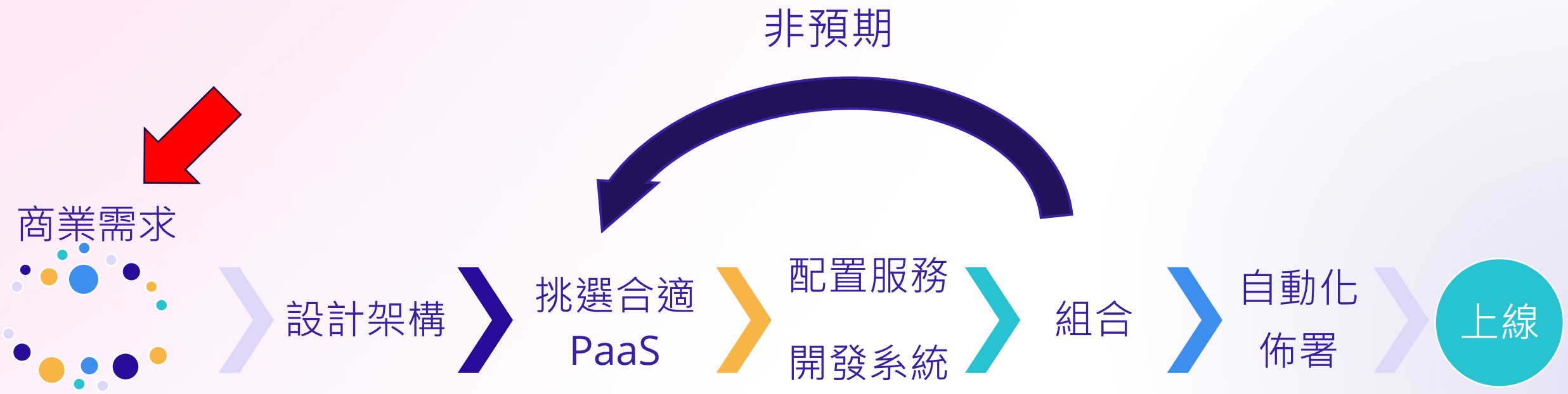
- 聽信廠商認為所有企業端的系統都可以上雲
- 沒有定義好雲端應用策略的框架或架構
- 沒有定義該上雲系統，或是需維持在地端不上雲的建置策略
- 用過多VM，發現成本過高而下雲

我們在PaaS的設計準則

- 最大化的運用PaaS資源進行組合和開發
- 不允許系統被部署在虛擬機或用虛擬機安裝資料庫
- 不同系統服務之間的連接必須(也)只能用API溝通
- API採用API First設計模式
- API一律使用JWT作為令牌交換的方法
- 只允許自動化佈署，無法透過人工方式部署

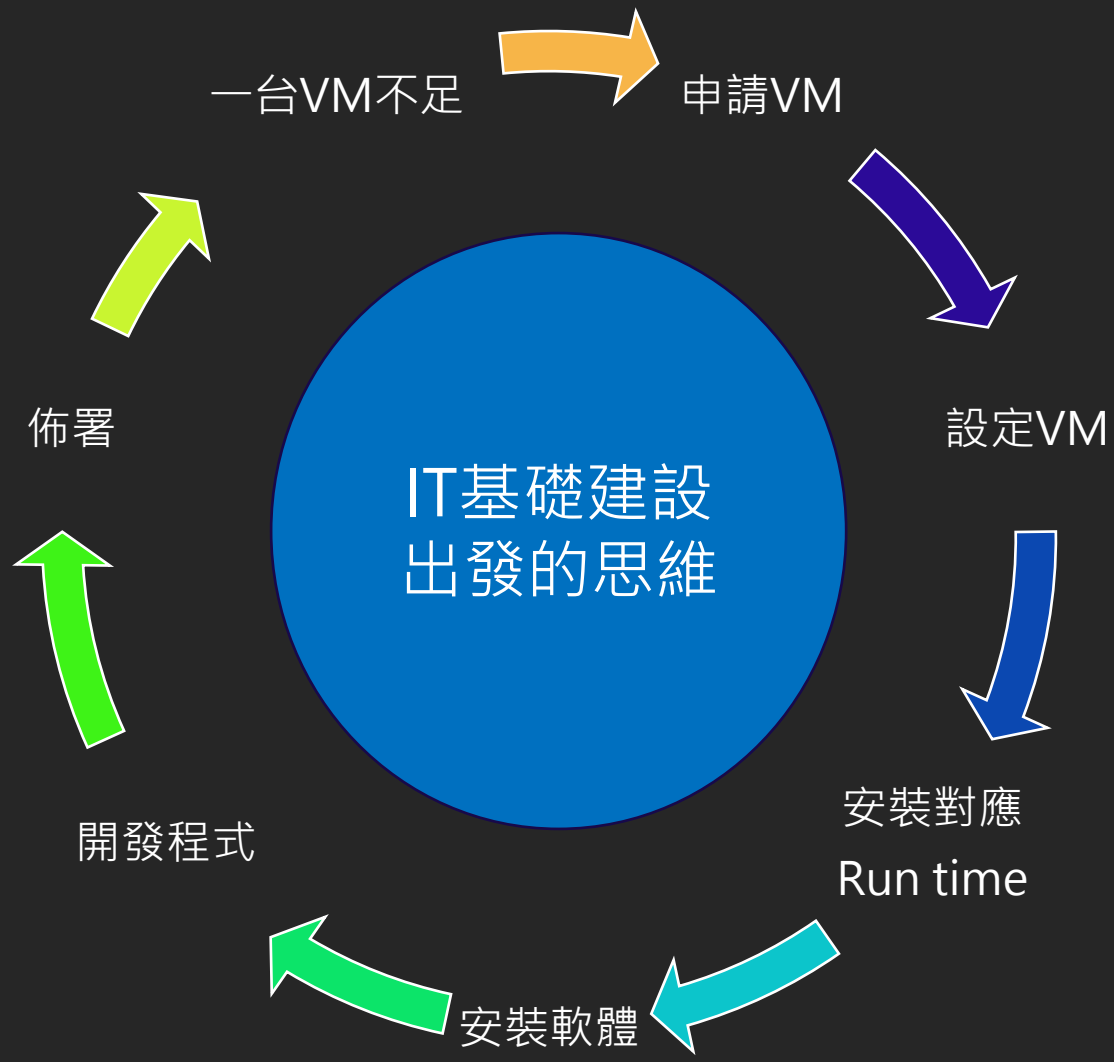


選擇PaaS步驟

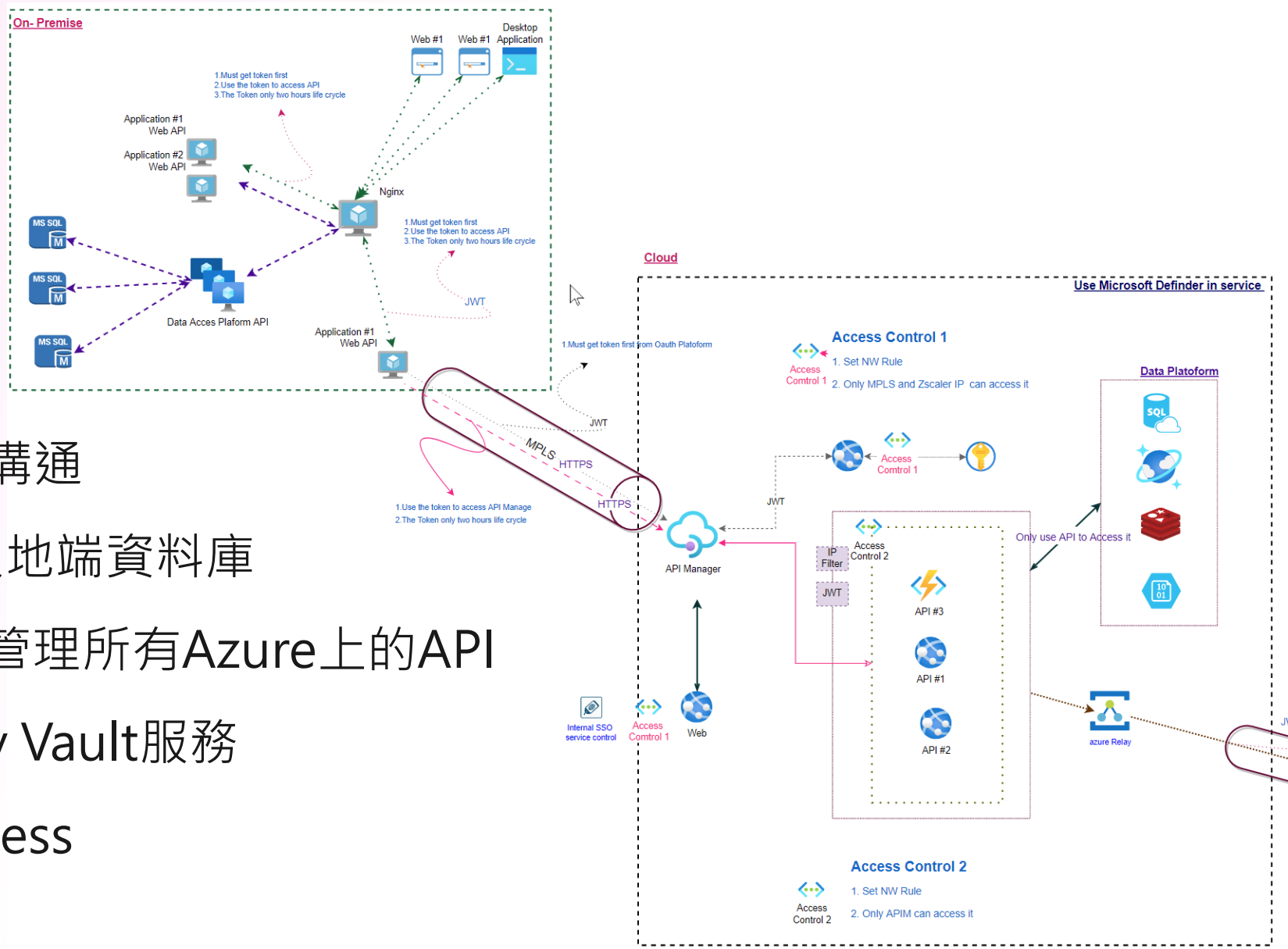


傳統做法...

- 必須完整確認商業需求對應的硬體規格
- 依照需求規格，等待VM申請或是採購
- 需求規格異動導致資源不足，必須再申請
- 安裝各項可運行系統之軟體或是SDK
- 限制商業系統架構的擴充、彈性與敏捷性



應用架構之一



- 建構ER與Azure 資料中心溝通
- 雲端系統只能透過API存取地端資料庫
- Azure API Management管理所有Azure上的API
- 加密金鑰保存在Azure Key Vault服務
- 資料庫使用PaaS或Serverless

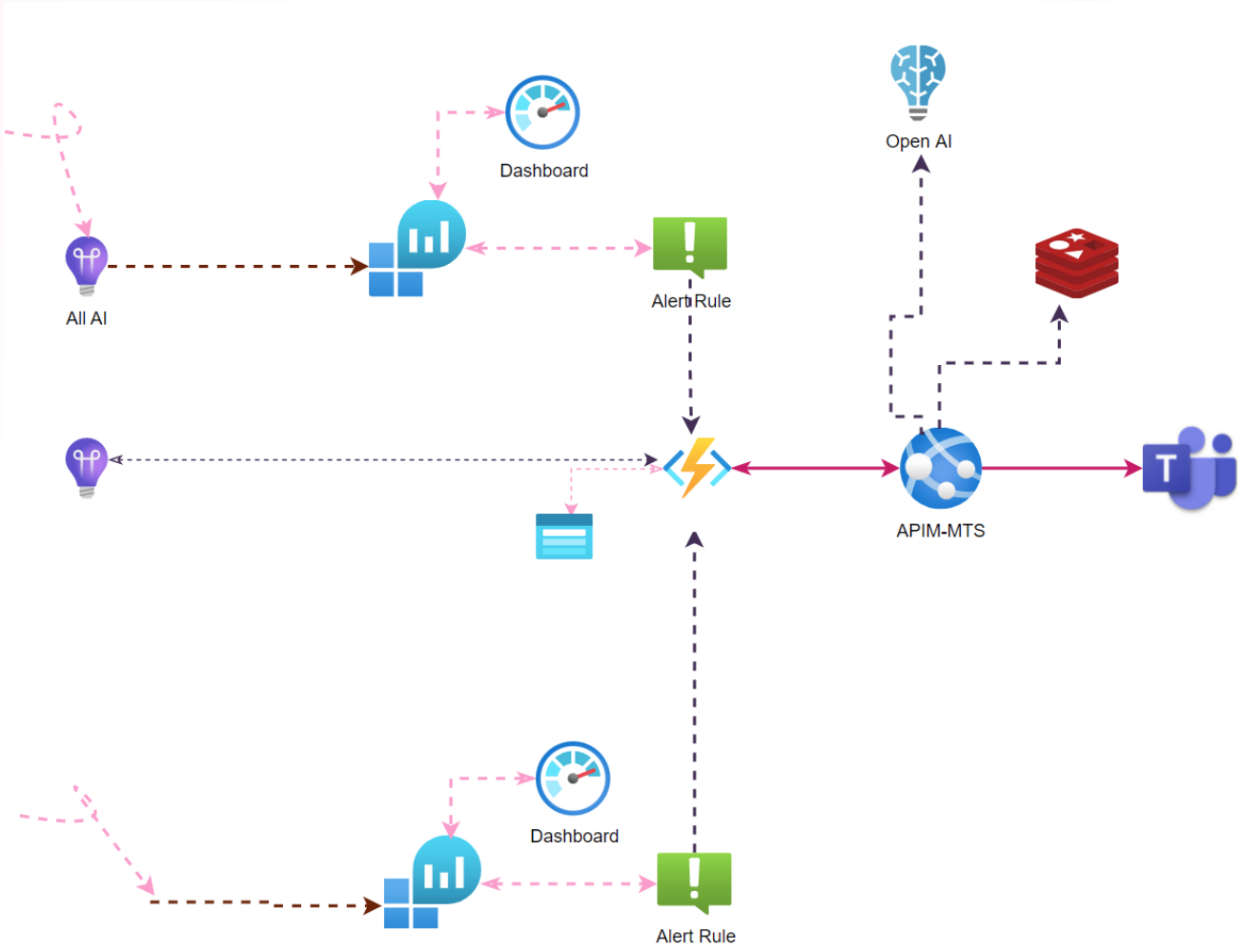
試想...

- 使用IaaS是否可以做到?
- 建置多少VM才能滿足?
- SQL Server / No SQL 叢集建置
- 成本經濟效益
- 需求變動導致轉換成本

應用架構之二

蒐集企業內部和雲端系統運作資訊的平台，並能快速查詢與分析資訊和產生維運必須之儀表板

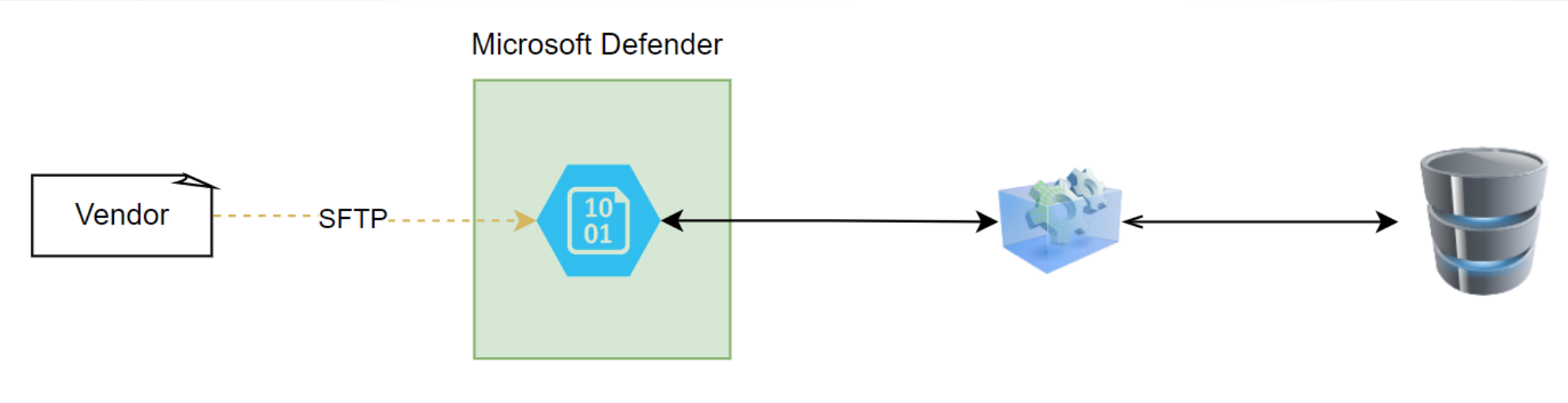
Application的Exception說明與解決方法，會透過Azure OpenAI 反饋



應用架構之三

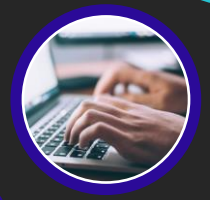
Azure Blob建立SFTP服務收傳送檔案，解析後存放到資料庫

- 快速建立SFTP服務，並能簡易設定帳戶權限
- Azure Function / Logic App解析檔案內容，存放到資料庫
- Microsoft Defender 保護檔案安全



IaaS是否可以做到...?

風險



彈性

人力

經濟效益

時間

PaaS 優於 IaaS

使用 PaaS 而非 IaaS。 IaaS 就像有一盒零件(VM、VNET、Public IP...)。 可以建置任何專案，但必須自行組合。

PaaS 選項更容易設定和管理。 不需要設定虛擬機器（VM）或虛擬網路。 也不需要處理維護工作，例如:安裝修補程式和更新。

而不是執行 ...	請考慮使用 ...
Active Directory	Microsoft Entra ID
Elasticsearch	Azure 認知搜尋
Hadoop	Azure HDInsight
IIS	Azure App Service
MongoDB	Azure Cosmos DB
Redis	Azure Cache for Redis
SQL Server	Azure SQL Database
檔案共用	Azure NetApp Files

DevOps團隊的優勢

- 鬆散耦合架構、封裝良好的架構
- 團隊能自主判斷何種服務對商業目標最好，選擇他們適用的服務
- 重要在於團隊變更設計或是服務，而不用仰賴其他團隊變更設定或系統
- 輕量化變更管理流程
- 輕量化概念是由團隊間作管控，從程式碼、基礎建設與資料庫變更
- 作為開發流程一部分，開發團隊是否有權創建並變更規格，而不需要經過非相關人員的流程

架構與技術目的在於幫助開發人員達到更好的成果，運用雲服務使結果成為可能

STUDY4 為學習而生 Build School

Application Migration to Azure PaaS

.NET 8 Code Assessment

評估轉移到Azure的.NET系統

- 使用 dotnet-appcat工具，它會使用進階分析技術瞭解任何 .NET 應用程式的結構和相依性
- 目前會識別移轉至 Azure App、AKS 和 Azure Container Apps 的潛在問題
- 工具安裝如下:
  - Visual Studio :Azure Migrate application and code assessment - Visual Studio Marketplace
  - .NET 工具 : dotnet tool install -g dotnet-appcat

識別相依性的種類


Database


File system


Network resources

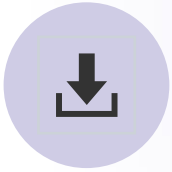

Authentication


Windows Identity


SMTP


Message queues


Caching


Session state

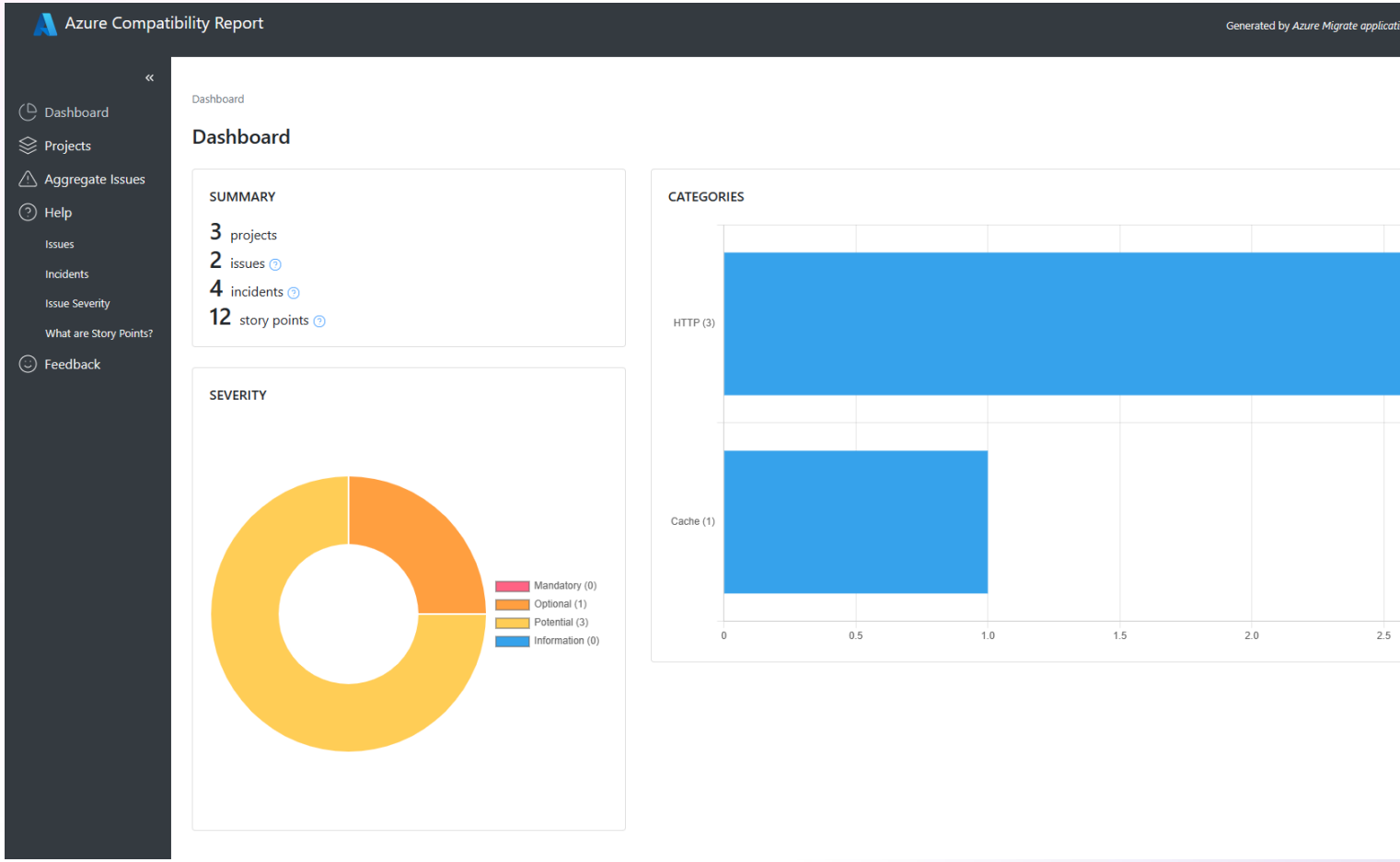

Secret management


Non-HTTP protocols


Application Request Routing

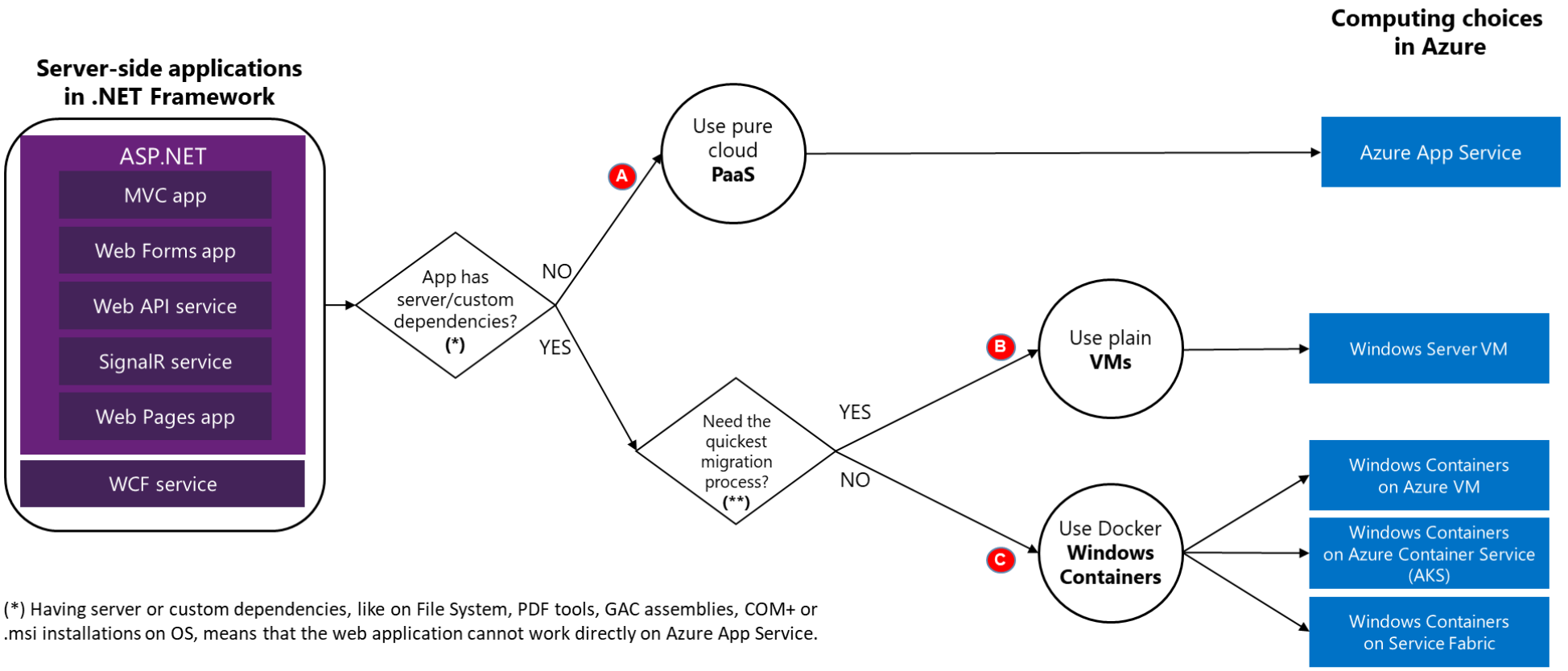
掃描結果

- 報表會以儀表板格式呈現其結果。
- 主要儀表板會顯示 [摘要] 區段，其中包含掃描結果、嚴重性和問題與事件的類別。



請先嘗試選項 A，但選項 B 是最容易執行的路徑

Azure compute decision tree for existing .NET Framework apps migration



(*) Having server or custom dependencies, like on File System, PDF tools, GAC assemblies, COM+ or .msi installations on OS, means that the web application cannot work directly on Azure App Service.

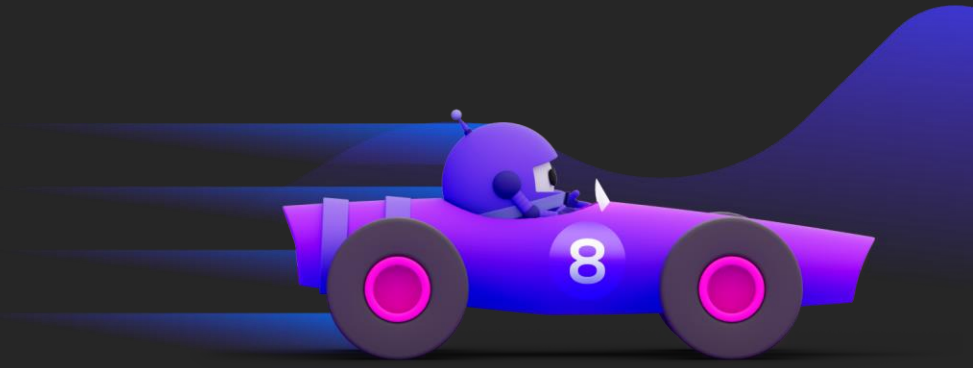
(**) Using Docker with Windows Containers is not the quickest migration path because it requires certain learning curve and minor changes in code, even when it won't require to re-architect your application. There are many deployment benefits, when using Docker Containers compared to plain Azure VMs, which make your applications *Cloud DevOps-Ready*.
On the other hand, plain VMs are familiar, pretty similar to on-premises servers/VMs and quick to migrate to.

NET 8 與Azure PaaS更多整合

.NET 8.0在Azure上構建高性能、韌性的服務帶來的新特性，包括Native AOT、Open Telemetry支持等，並提升.NET開發體驗

PaaS的資訊安全

PaaS資安設計與企業內部資安的思維是不同



安全責任

每個雲端模型（IaaS、PaaS 或 SaaS）都有不同的安全控制所有權，和實施這些控制的不同方法。

PaaS的安全責任是由廠商負責

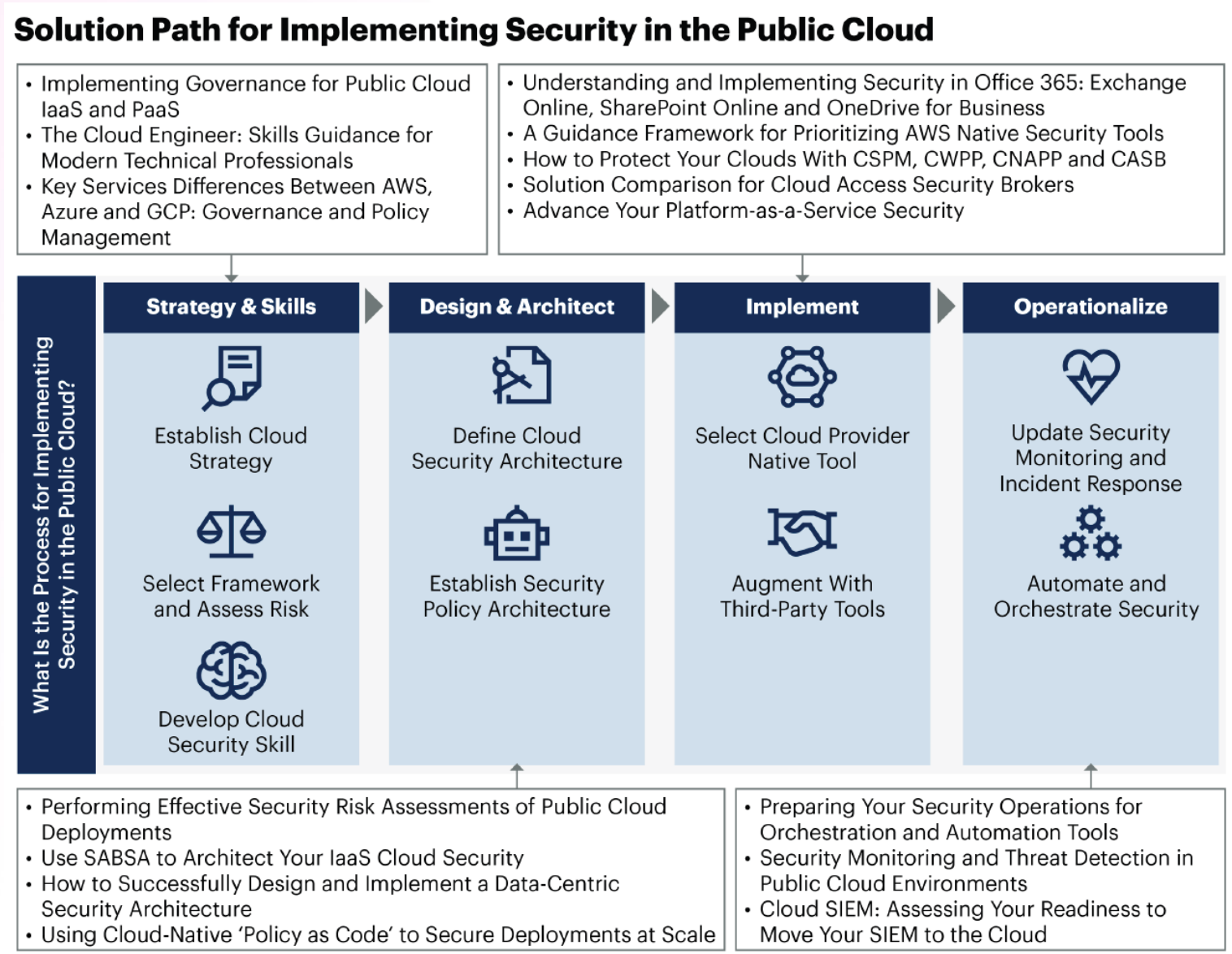
~From Gartner

雲安全解決方案

企業遷移到雲端不再是問題。問題是「企業如何保持敏捷並利用技術推動業務成長，同時保持安全性」

資訊安全人員除確保資訊安全外，首要任務必須是保持敏捷性，同時確保業務能夠專注於最擅長的事情，而不是因為安全策略而受到干擾。

~From Gartner



Source: Gartner
757969_C

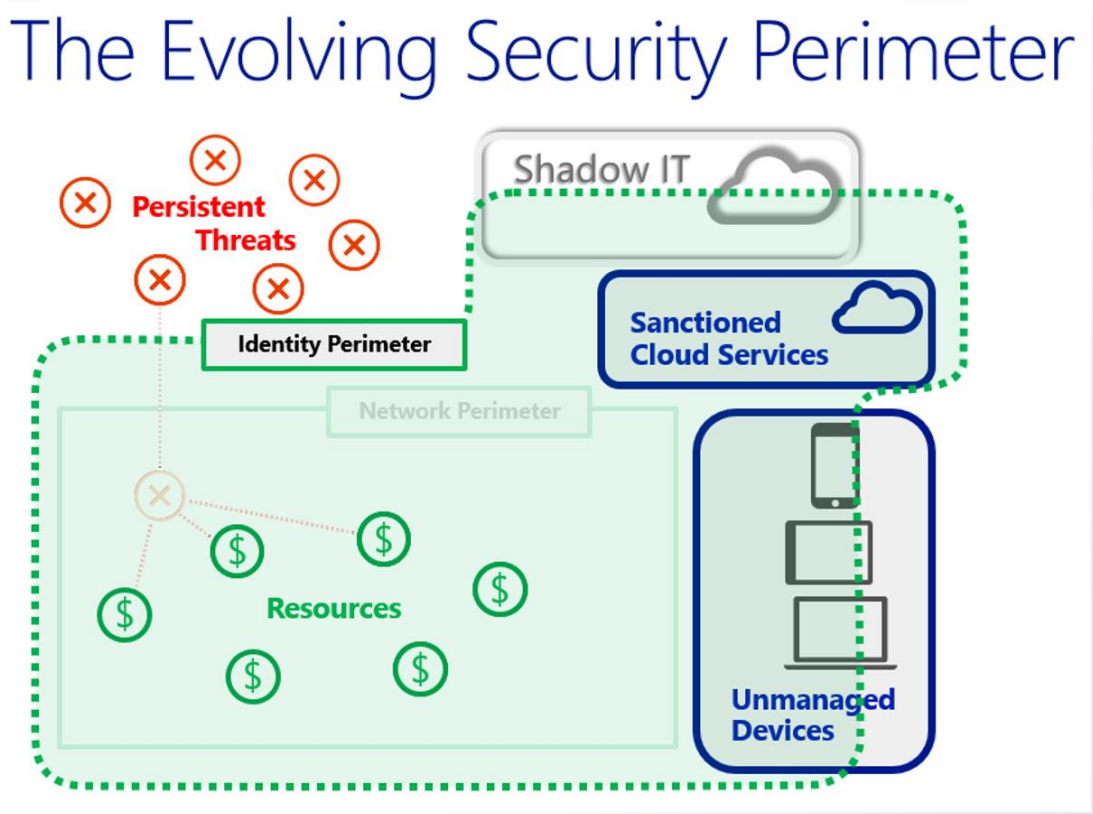
事實上...

- 多數企業負責資訊安全人員思維並未轉型到雲資訊安全思維
- 把PaaS當作IaaS或是企業內VM架構設計資訊安全
- 過度或不正確的資訊策略導致喪失PaaS原有彈性和優勢
- PaaS權責從原本Infra (資安)轉至開發人員或雲平台，導致不安感

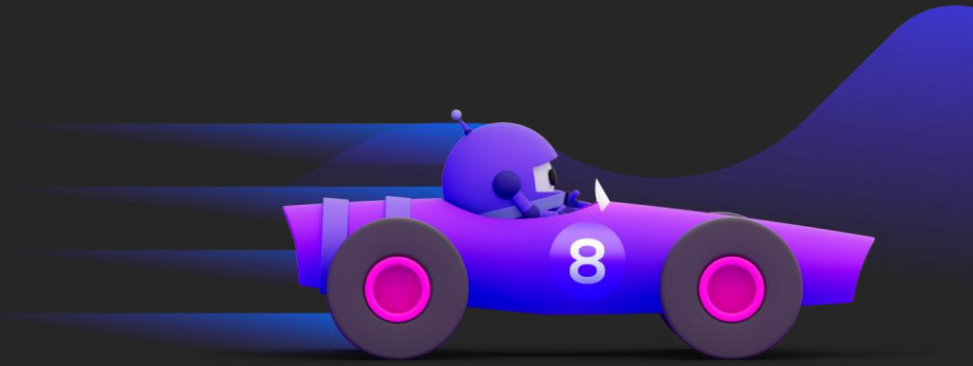
PaaS安全性考量

「雲端運算的五個基本特徵之一是廣泛的網路存取，這使得以網路為中心的思維變得不那麼相關。在雲端運算的目標中，讓用戶能夠不受地點限制地存取資源。對於大多數用戶來說，他們的位置通常是在網際網路上。

從安全的角度來看，安全邊界已經從網路邊界演變為身分邊界。安全性的重點不再是保衛的網路，而是更多地關注保護資料並管理應用程式和使用者的安全。關鍵的差異在於你希望將安全性更貼近對你公司重要的事物」



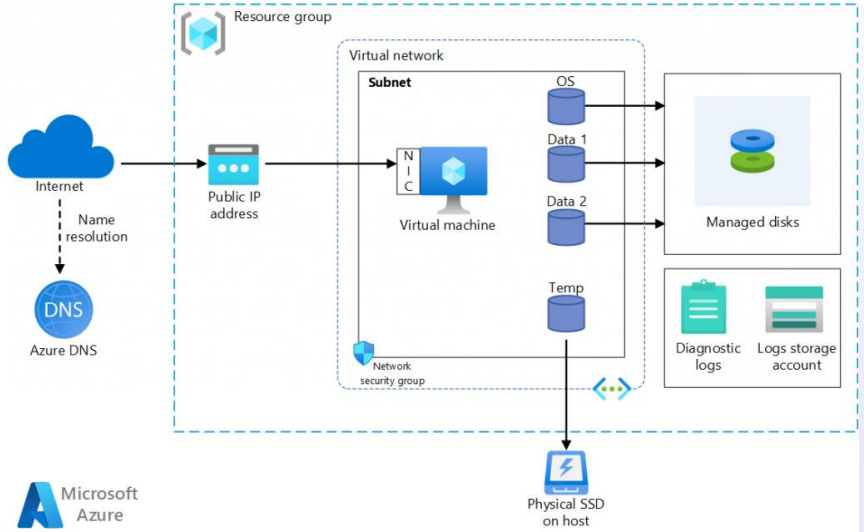
總結

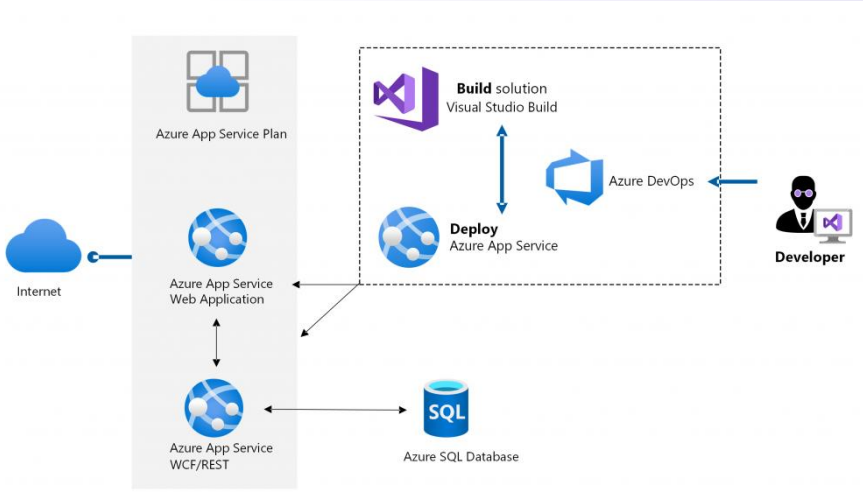


PaaS帶來的好處

我們雲費用價格變化

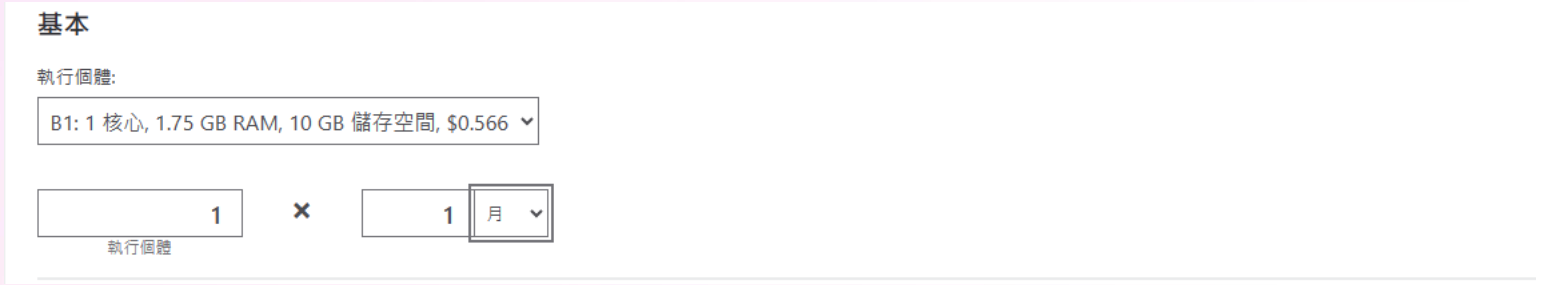
- PaaS月費比IaaS下降30~50%
- PaaS省去不少建置和維運成本
- PaaS依據服務等級與應用搭配雲服務規格
- 像是Azure Redis，則降低基礎配置和維運成本
- 省掉備份和備援的額外儲存成本
- 費用中心化，避免大鍋飯方式浪費資源
- 從商業情境拆分並搭配合適的雲服務





案例..

- 使用.NET 8，使用Linux版本方式計算，Windows價格差異更大

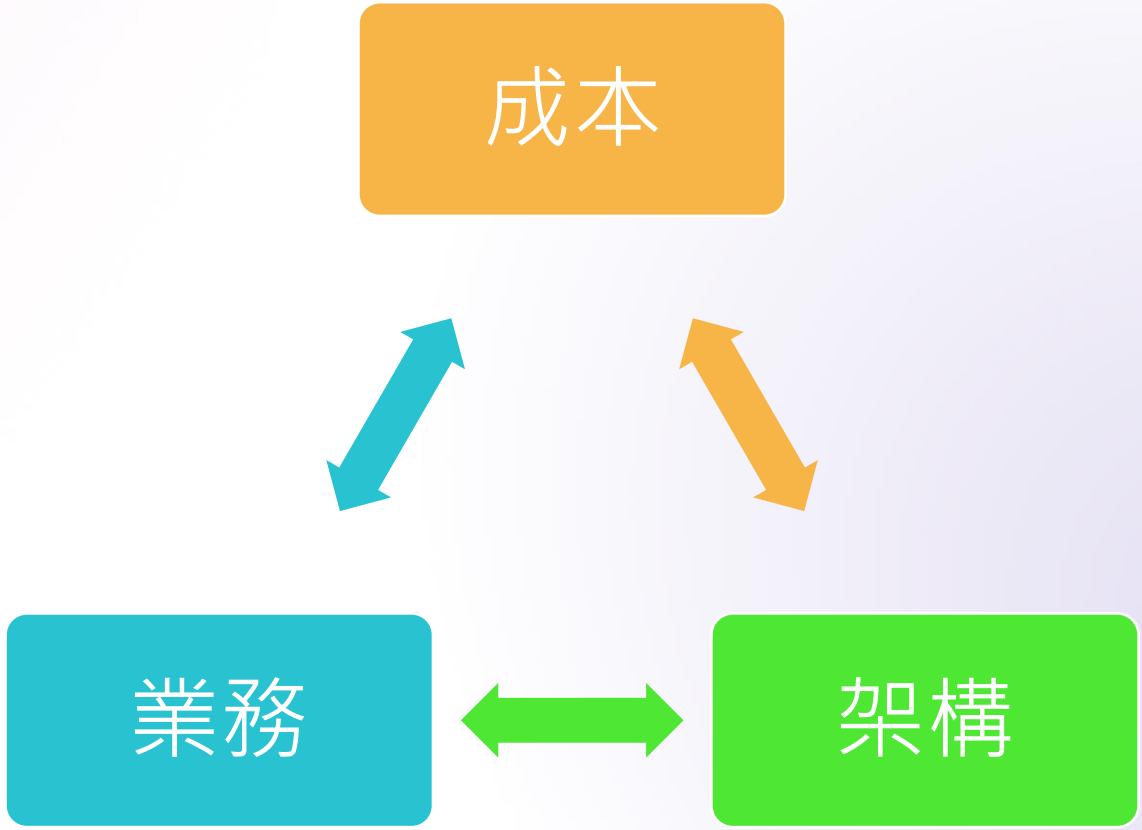


$APP\ Service = 413$

$VM + 128\ G\ SSD$ +儲存交易=830.58

成本、業務與架構三者搭配

- 架構設計是一系列權衡的集合
- 確保系統的最終成本與業務保持一致
- 將成本視為一種非功能性需求
- 無法觀測的系統將帶來無法估量的成本
- 成本觀測架構實現成本控制
- 成本優化是一個迭代的过程



是否會被雲廠商鎖定

- 雲廠商鎖定的風險被誇大，試圖避免只會增加整體的複雜性、風險和成本
- 企業內部和使用第三方機房或托管服務提供商，都存在鎖定的情況像是 Dell EMC、HP、Lenovo、、NetApp、Cisco、Arista、Red Hat、VMware、Microsoft、Atlassian..等「綁架」
- 雲端的靈活性、廣泛性和深度，降低不同之間雲切換的成本
- 應用程式託管是一個部署問題，容器技術將可以解決此問題
- 管理平台服務的鎖定問題，良好的程式碼設計可以隔離服務依賴性
- 故意選擇多個雲端避免鎖定成本是昂貴
- 試圖整合不同雲端間的服務可能會帶來更多的成本和技術衝突

優勢

- 團隊能「共享整個軟體開發和維運生命週期」的所有權
- 專注商業目標、軟體設計和提供服務
- 減少Infrastructure層面帶來的阻礙與影響
- 面對快速的市場變化
- 提升程式碼品質、系統架構穩定與降低維運成本
- 隨點即用 & 降低成本

挑戰

- 打破原始組織工作權責
- 傳統Infrastructure組織的思維改變
- 企業內部資安防禦的概念轉換
- 雲端資安和地端的資安，考慮面向是不同
- 開發人員從傳統開發思維轉換成雲端開發思維
- 面對PaaS服務的管理、非自願的版本更新與淘汰的挑戰
- 極為特殊的需求或客製化將無法使用

Don’t...

「多數IT團隊僅憑以往的工作經驗，他們就認為當前的技術、架構或語言永遠是最佳選擇。這可能會產生一種錯誤的安全感，阻礙對現狀的質疑，更會打擊對可能更加高績效、更具成本效益或可擴展性更強的新選項的探索」

Resources

Download .NET 8
aka.ms/get-dotnet-8

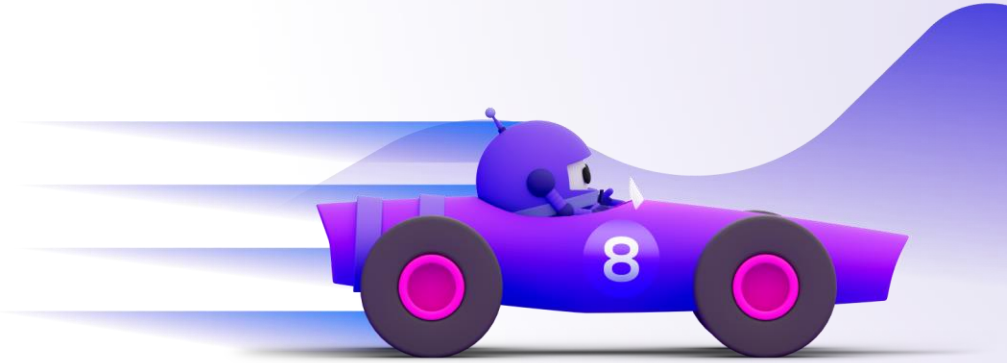
Lorem ipsum
hyperlink

Lorem ipsum
hyperlink

Lorem ipsum
hyperlink

Download .NET 8

https://aka.ms/get-dotnet-8



特別感謝

.NET Conf
TAIWAN

Microsoft

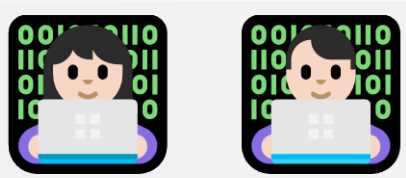
tsmc

Build School

成功

HackMD

MVP Microsoft® Most Valuable Professional



STUDY4
為 學 習 而 生

以及各位參與活動的你們