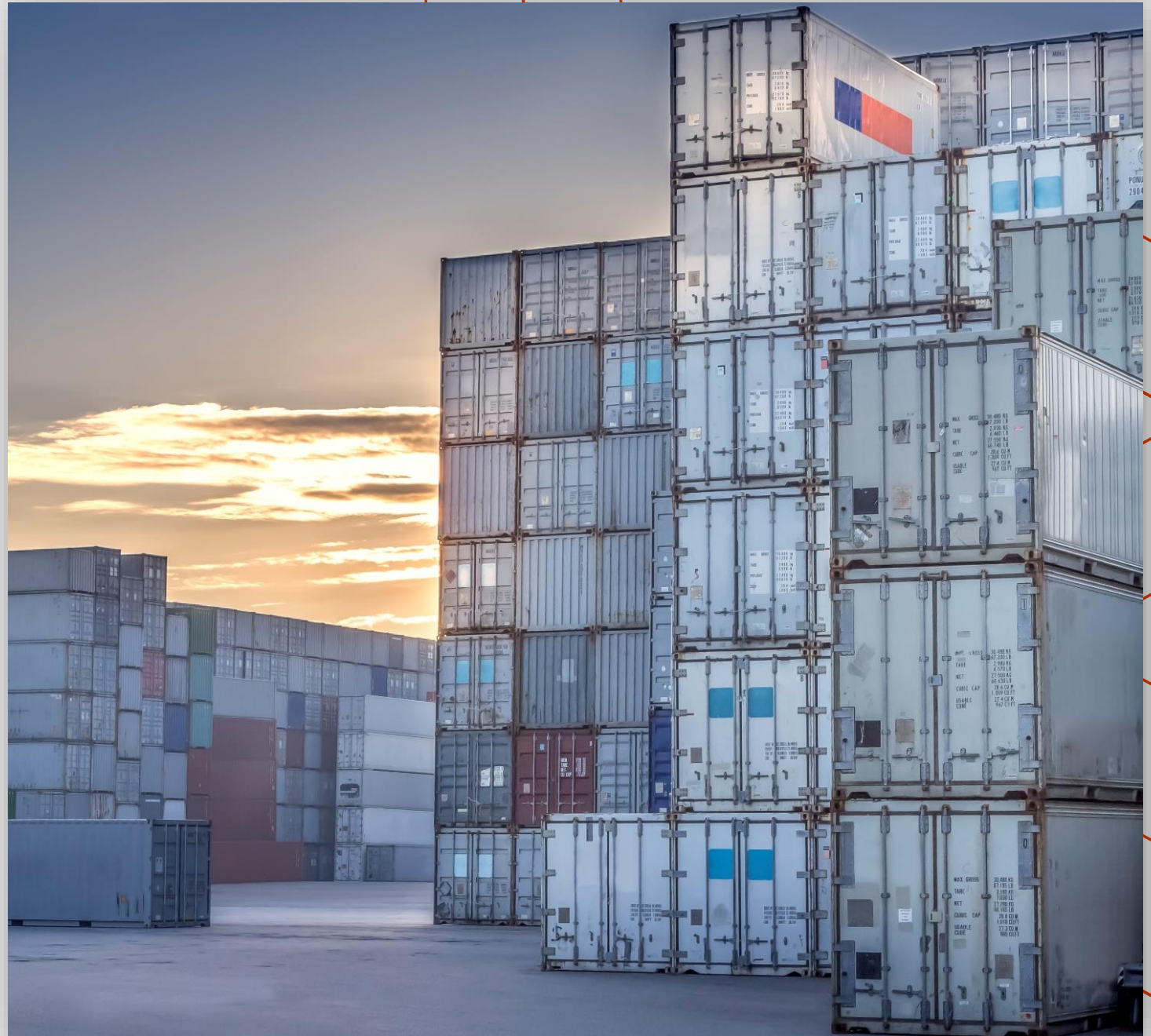
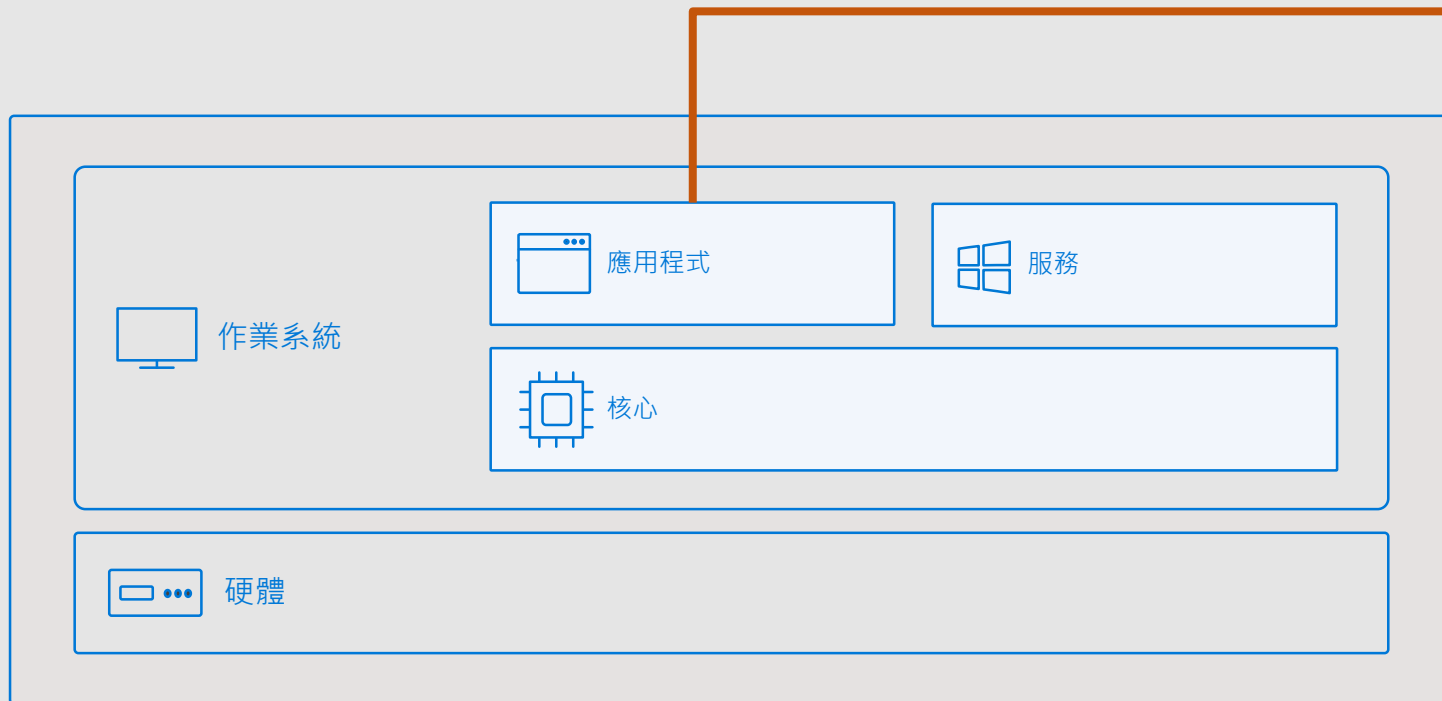


所以... 什麼是容器?



架構

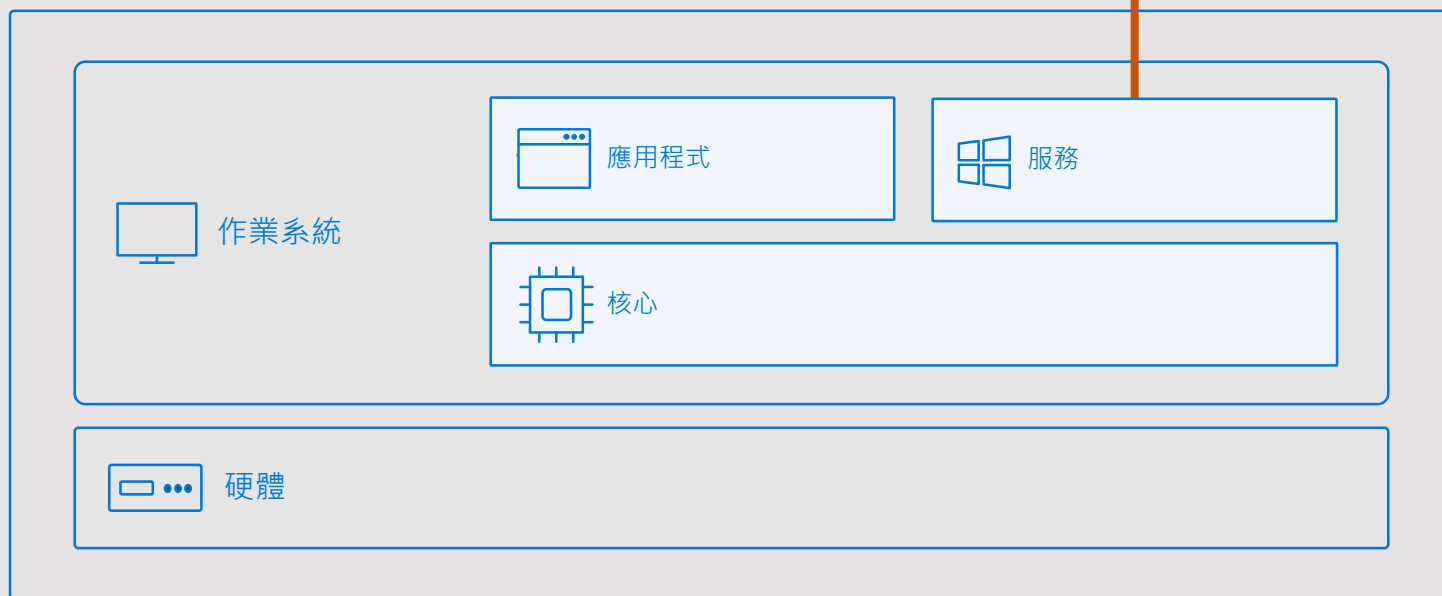
傳統



- 你的應用程式，可能由各種程式語言所撰寫：
 - Python
 - NodeJS
 - .NET full
 - .NET core

架構

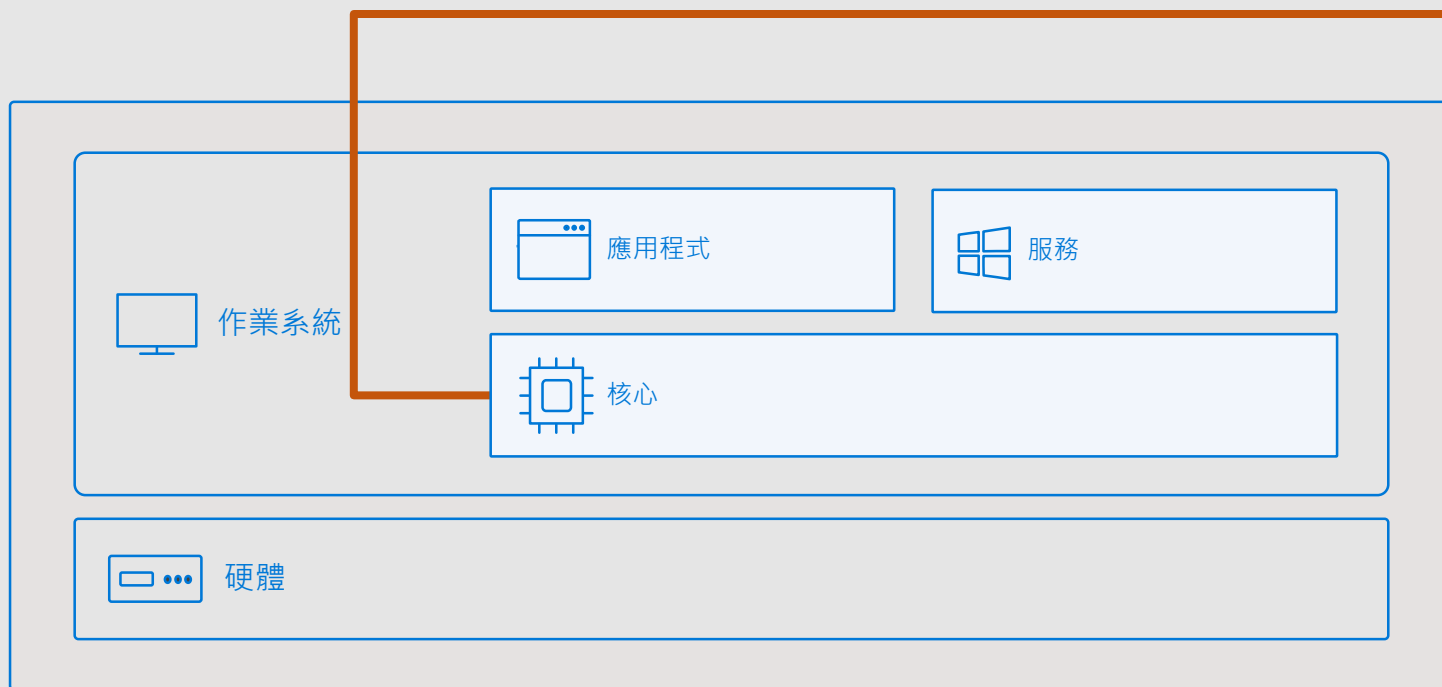
傳統



- 長期執行的程式
- 提供應用程式功能呼叫
- 可做為與作業系統核心溝通的介面

架構

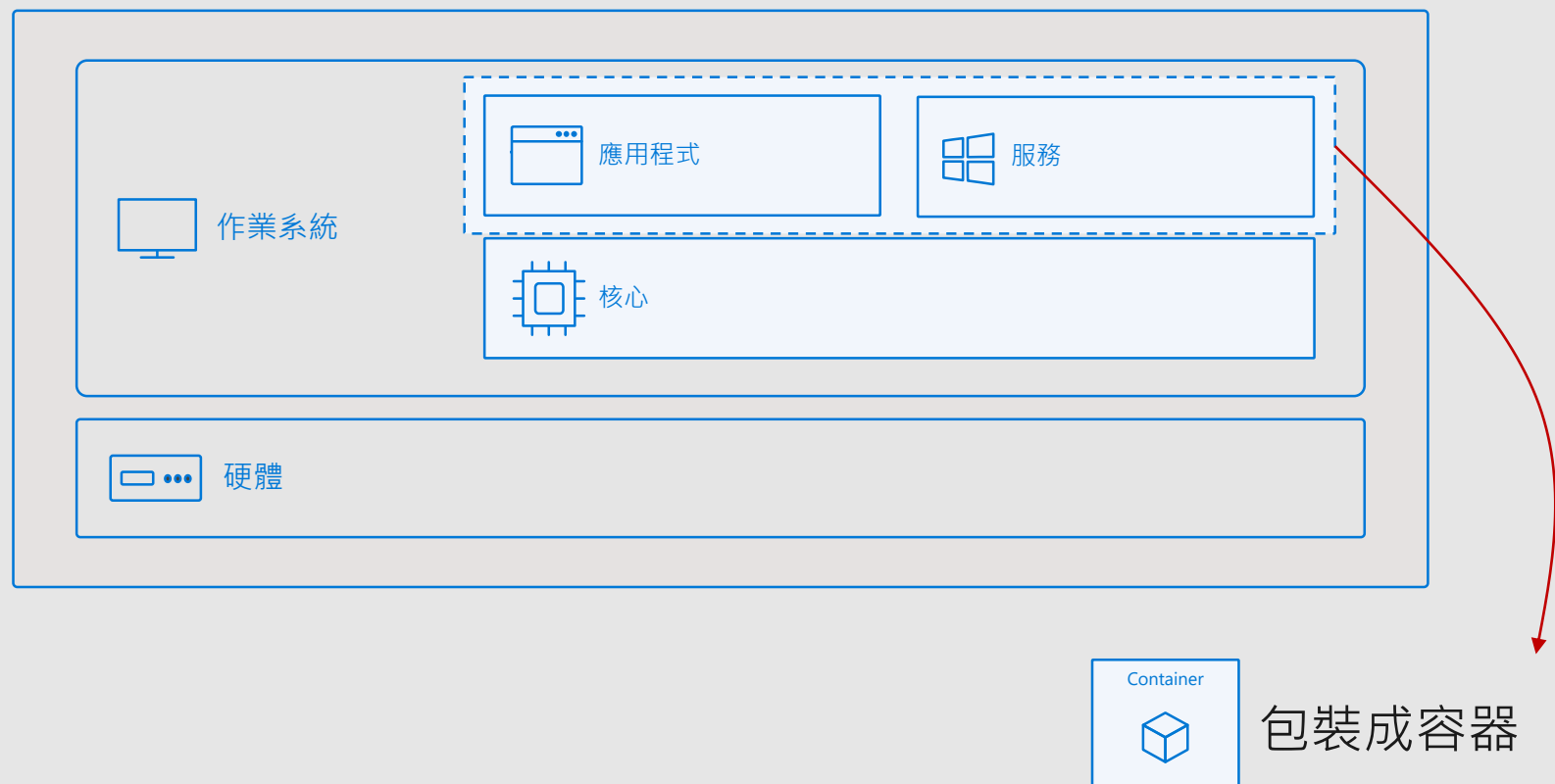
傳統



- 提供你的應用程式低階的功能呼叫
- 應用程式透過核心與硬體溝通
- 管理系統資源
- 硬體驅動程式

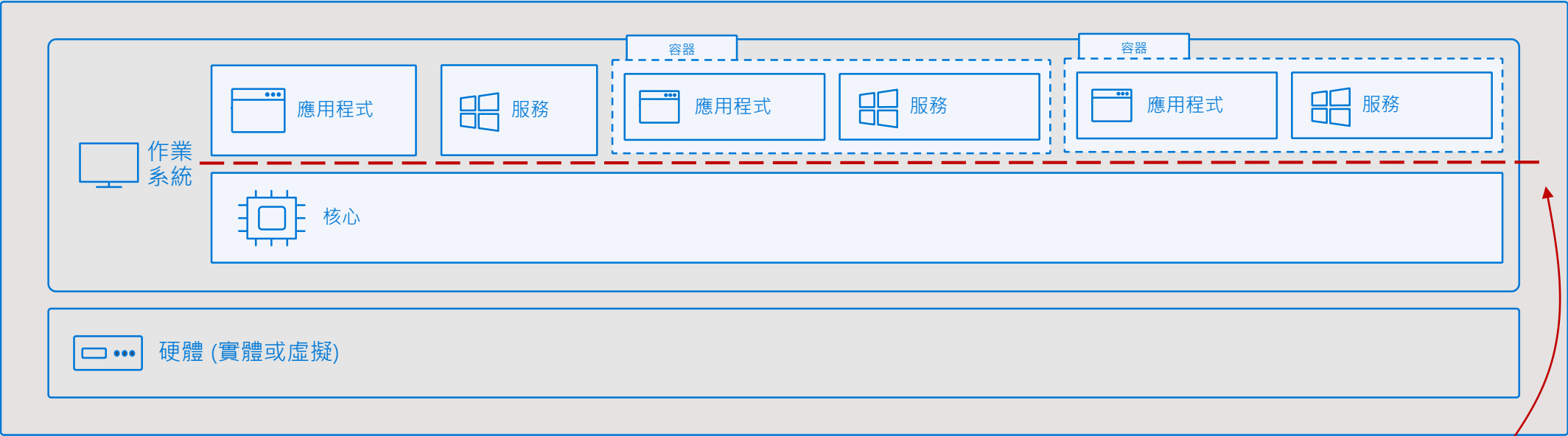
架構

傳統 -> 容器



架構

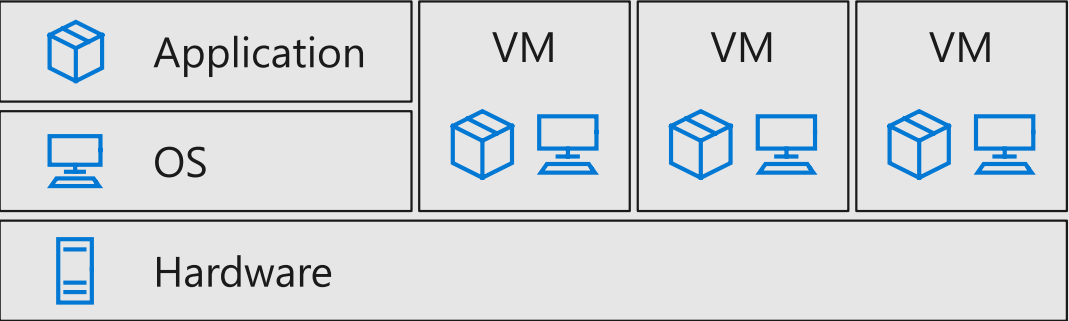
容器



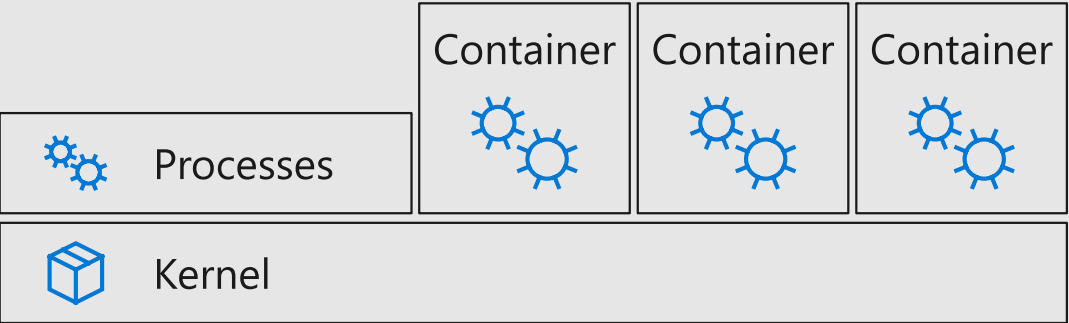
虛擬化界線

什麼是容器？

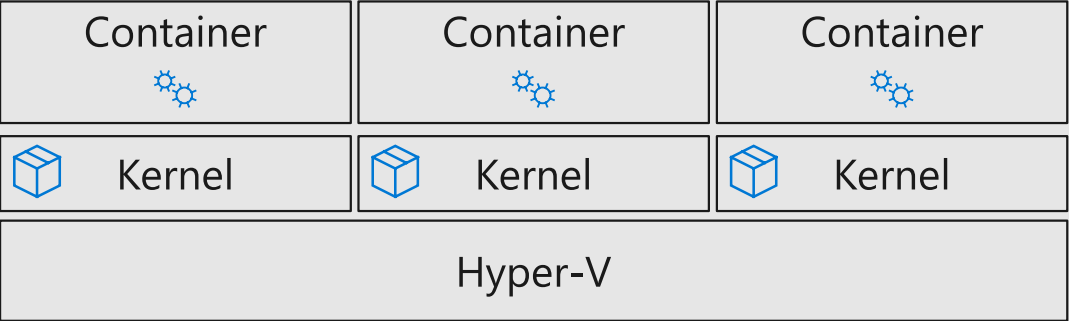
傳統虛擬機器 = 硬體虛擬化



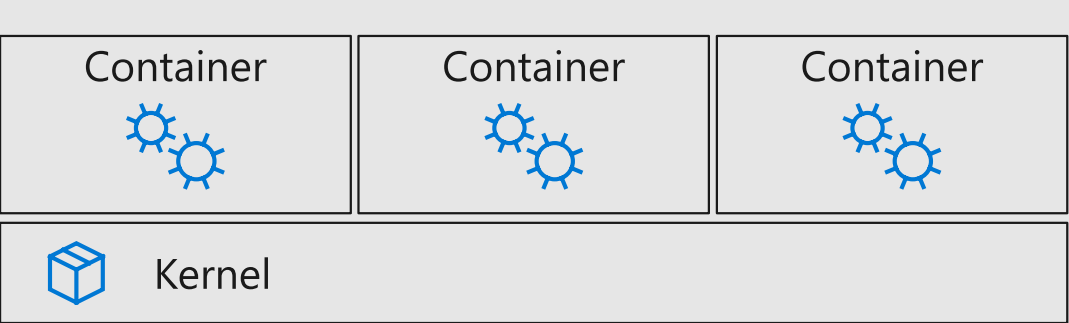
容器 = 作業系統虛擬化



Hyper-V 容器 — 隔離 + 效能



Windows Server 容器 — 最大化速度與密度



Windows 作業系統映像

Nano Server

為雲應用程式而生

支援 .NET Core

映像大小：94 MB

Windows Server Core

應用程式相容性

支援完整的 .NET Framework

映像大小：1.4 GB

Windows

所有現有應用程式的平台

具有大部份的 Windows 作業系統元件

映像大小：3.5 GB



如何在生產環境管理容器？



Azure Kubernetes Service

不需要是 Kubernetes 專家就可以建置與管理 Kubernetes 叢集

全自動更新、上補丁

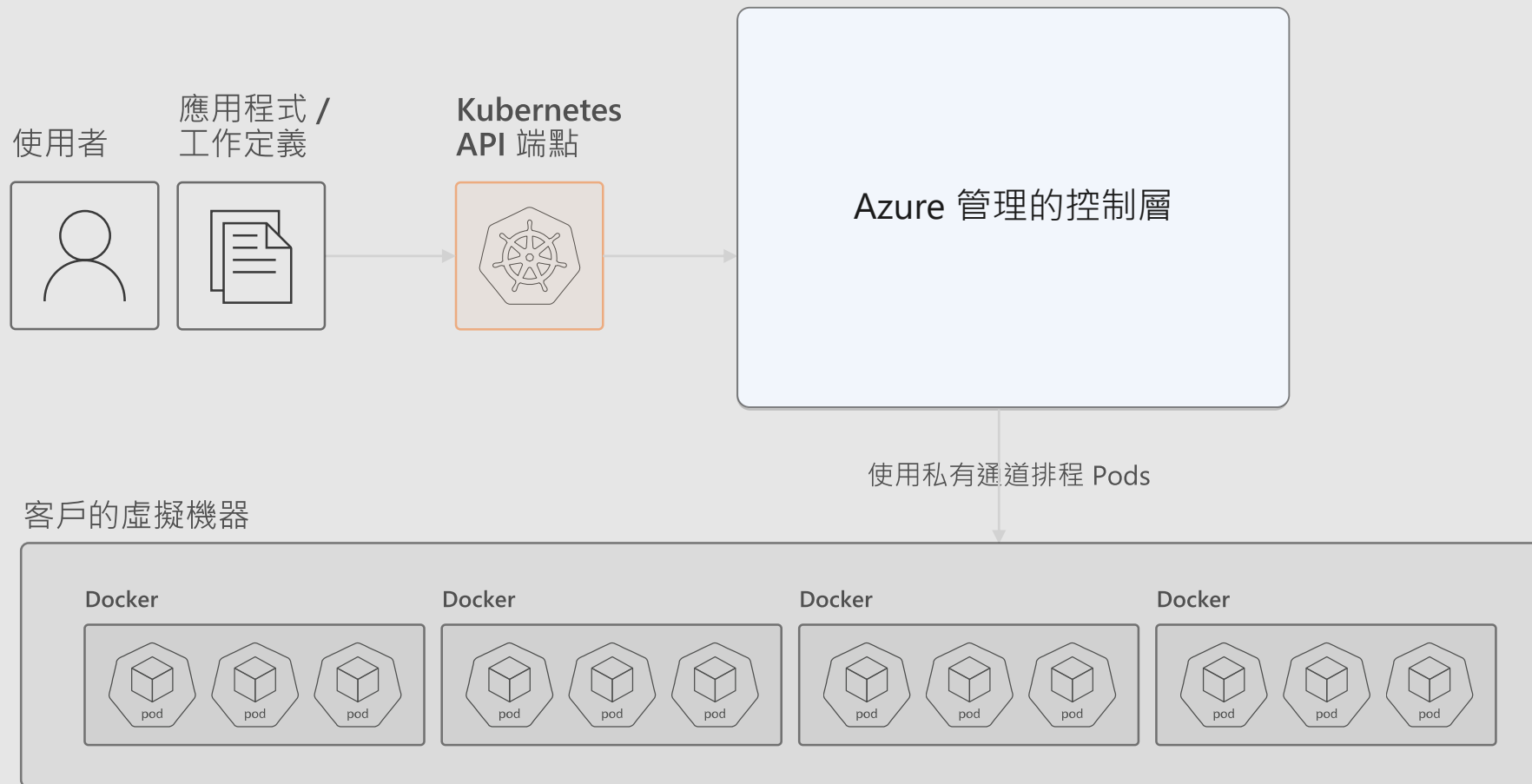
高可靠性與可用性

簡單安全的叢集規模調整

自我修復

API 伺服器監控

透過控制層，前端使用完全不受版本
改變影響



Microsoft Azure

混合式資料中心

混合雲

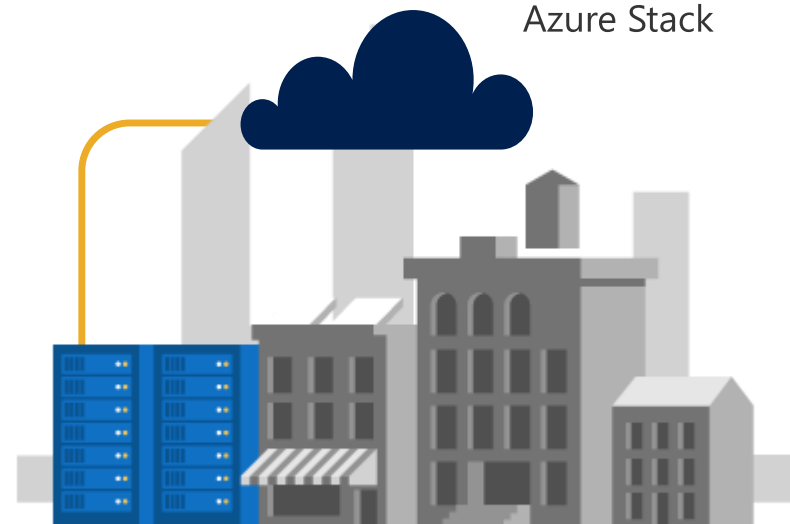
WSSD (Windows Server 軟體定義)

SQL/.NET/容器



地端

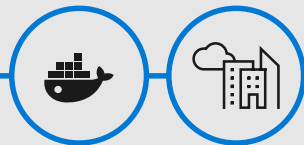
Azure Stack



Web 容器應用程式

容易部署與執行以容器為基礎的各種規模 Web 應用程式

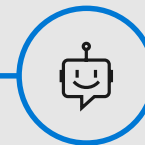
加速開發循環



緊密整合 Docker Hub 與
Azure Container Registry



內建 CI/CD 與部署位置

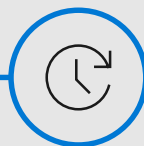


智慧性診斷、疑難排解
與遠端偵錯

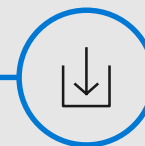
完整授控的平台



自動規模調整
與負載平衡



高可用性、自動更新



備份與復原

彈性與選擇

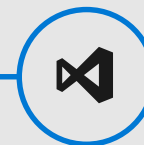
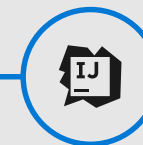


使用 CLI、PowerShell (Windows)
、入口網站或 ARM 範本來部署



單一 Docker 映像、
多容器、Docker Compose、Kubernetes

只支援 Linux



IntelliJ, Jenkins, Maven
Visual Studio family

建立 CI/CD 管線



建立 CI/CD 管線

啟用持續部署

VSTS Team 專案

App Service 的 CI/CD webhook

Azure Container Registry

App Service 的部署位置與位置切換