2026 年 | 指南

打造可投入正式環境運作的 Kubernetes 平台所需的條件

了解為何「自行建置 Kubernetes」會在實際執行環境中失效，以及平台團隊需要採取哪些不同的做法。



NUTANIX

目錄

第一章　挑戰：要打造正式環境就緒的平台，為何光有 Kubernetes 還不夠 3

第二章　策略性轉向：內部開發人員平台與黃金路徑 5

第三章　Nutanix Kubernetes 平台 6

第四章　針對具狀態工作負載的企業級資料服務與管理 9

第五章　平台團隊的營運成果 10

第一章
挑戰：要打造正式環境就緒的平台，為何光有 Kubernetes 還不夠

自行建置平台的生命週期

大多數組織並非一開始就打算建置一個複雜的內部平台，而是隨著一個又一個 YAML 檔案的累積，不知不覺間就陷入其中。在初始建置階段的「蜜月期」，組合各種開放原始碼元件看似能充分發揮生產力，團隊也能專注於推動專案交付上。每遇到一個新問題，都像是一道有趣的謎題，可以用新穎的方法加以解決；然而，隨著平台的規模擴大，情況也開始有所不同。

最初僅是一個簡單的協調工具，如今已演變為由網路、網狀網路、可觀測性及身分驗證等相互依存的工具所構成的脆弱網路。每次 Kubernetes® 更新，都會引發一連串相依性問題的骨牌效應。到了持續營運階段，團隊的大量時間都耗費在相容性測試上，而管理整個機群的組態偏移則成為團隊的全職工作。

從開發人員身上卸下的認知負荷並未消失，只是轉嫁到了平台工程師身上。這支團隊非但沒有為使用者打造一條「黃金路徑」，反而深陷於維持系統運作的循環之中，不斷為了維運而犧牲藍圖中的創新。造成正式環境就緒能力落差的問題，並不在於 Kubernetes 的協調功能本身，而是如何妥善管理其周邊所有元件的完整生命週期。

功能缺口：超越協調與生命週期維護

要建置一個完整的企業級容器平台，所需的功能必須遠遠超出核心 Kubernetes 協調範圍。為了為開發人員提供穩定的環境，平台工程師必須手動整合並管理由各式各樣元件組成的技術堆疊。

鑑於 CNCF 生態系中已有超過 1,200 個專案，各組織在工具選型與整合方面面臨極大的複雜性。每項新增至技術堆疊的工具，無論是用於 GitOps、可觀測性或資安，都會成為一項永久性的維護需求，其中包含：

- **進階網路技術：**L4/L7 負載平衡、DNS 及網路政策強制執行
- **輸入閘道：**HTTP 路由、SSL/TLS 終端處理及外部存取管理
- **容器登錄：**私有映像託管與生命週期管理
- **資料服務：**具狀態儲存、儲存遷移，以及具備複寫功能的容器原生備份/還原，以確保業務永續性
- **政策管理：**跨叢集強制執行「政策即程式碼」的集中式角色導向存取控制（RBAC）
- **安全性：**密碼管理、供應鏈管控及零信任網路架構
- **可觀測性：**跨叢集的指標、日誌與追蹤
- **自動化與部署：**GitOps 工作流程 (Argo CD / Flux) 加上用於建置/測試/部署的 CI/CD 管道整合
- **多叢集管理：**機群佈建、升級、政策部署及營運管控，橫跨混合雲/多雲環境
- **FinOps/成本管理工具：**跨團隊與跨叢集的資源使用狀況可見性與成本最佳化
- **GPU 存取：**管理 AI/ML 工作負載的 GPU 存取與排程
- **服務網格：**流量管理、服務間安全性 (mTLS) 以及工作負載層級的可觀測性

手動管理這些元件的生命週期會消耗大量資源。一個可實際執行環境平台，需要手動整合 20 多個各不相同的元件。由於每個專案每年會發布 3 至 4 次，平台團隊每年必須處理超過 100 次升級，這已構成實證的負擔，且每次升級都需進行獨立的相容性測試與回歸測試。如果缺乏一種能將整個堆疊視為經過驗證的整體，再加以升級的單一平台，工程師就只能持續將心力投入底層基礎架構管理，而無法專注於提供架構層面的價值。

支援期間與 API 的脆弱性

Kubernetes 以快速的節奏發布次要版本，而上游對特定次要版本的修補程式支援期約為 12 個月 (通常視為端到端約 14 個月)，因此平台團隊最終必須持續進行升級作業。

由於上游 Kubernetes 並非以預先整合的企業級套件形式提供，因此每次升級都必須進行全面的相容性測試。這包括稽核已被取代及移除的 API、更新清單與運算子，並驗證關鍵附加元件 (CNI、CSI、輸入控制器、政策及可觀測性元件) 是否仍與目標版本相容。這往往會導致「升級債務」，團隊因無法承擔系統停機的風險，或缺乏資源來妥善驗證所有相互依賴關係，而延遲部署關鍵的安全修補程式。

跨環境的碎片化

許多組織主要在地端環境中運作，然而，越來越多組織被要求支援同時涵蓋公有雲和邊緣節點的混合環境。若缺乏統一且不依賴特定廠商的 API，這些環境便會變成彼此隔離的運作孤島。這種碎片化現象通常迫使企業必須成立多個團隊來支援混合維運，包括獨立的工程團隊、地端團隊以及公有雲團隊，因為管理工作流程和自動化指令碼無法在不同服務供應商之間移植。

維持這些團隊的主要影響在於，營運開銷與技術複雜度將大幅增加，因為組織必須重複投入資源，才能在不同的基礎架構上管理相同的工作負載。這種缺乏一致性的情況，使得難以強制執行統一的安全策略，導致叢集機群淪為一堆彼此不連貫的環境。各團隊需要一套解決方案，能夠透過單一運作模式，在任何環境中標準化叢集的建置與安全防護方式，藉此減少對備援專門團隊的需求，並確保無論工作負載在何處執行，都能達到正式環境所需的就緒標準。

具狀態工作負載韌性與資料保護

Kubernetes 原本是為無狀態服務而設計的，其狀態與常駐資料會被推送到外部基礎架構中。當組織將任務關鍵型資料庫、鍵值儲存系統，或其他長期儲存方案遷移至機群時，儲存便成為最大的瓶頸，而平台團隊則被迫手動彌合 Kubernetes 與傳統儲存陣列之間的差距。

在雲原生架構中，部署企業級資料服務仍是一大挑戰。雖然 Kubernetes 能有效處理無狀態工作負載的擴展，但要保護具狀態工作負載，則需要進行複雜的整合以支援多種協定，包括區塊儲存、檔案儲存以及相容於 S3 的物件儲存。若缺乏具備應用程式感知的原生儲存整合功能 (該功能需支援 Metro 或非同步複寫)，要達到任務關鍵型工作負載所需的復原時間，對大多數平台團隊而言，仍是導致在持續營運階段面臨工作倦怠感的主要原因。

不斷攀升的成本與營運債務

開放原始碼、DIY 平台最初的吸引力，往往會被維護成本以及維持其運作所需的手動工作量所掩蓋。當平台是逐塊搭建時，工程團隊的工時便會從開發人員服務的建置，轉而投入到修補、測試及排除碎片化技術堆疊問題的無止盡循環中。

隨著機群規模擴大，營運負擔會呈現非線性增長，進而形成一種困境：團隊的資源將被生命週期管理及故障排除與維修工作所耗盡。最終，這種營運債務會成為瓶頸，迫使最優秀的工程師將精力投入於維護基礎架構，而非創新的高價值服務。

第二章
策略性轉向：內部開發人員平台與黃金路徑

許多企業正在探索一種由專業團隊管理與精心規劃的平台體驗，以擺脫持續維護分散且自行組建的技術堆疊所帶來的負擔。

目標是讓團隊將重心從低階元件整合，轉移到交付內部開發人員平台 (IDP) 上。平台工程團隊打造了一條通往實際執行環境的「黃金路徑」，作為簡化與標準化創新的最簡便途徑。

為了發揮實效，內部開發人員平台的基礎應為：

- **開放且可擴展：**基於純粹的上游 Kubernetes 建構，以確保與 CNCF 生態系統完全相容性，並避免受到專有技術的限制
- **跨環境的一致性：**為符合採用 Kubernetes 的策略性目標，Kubernetes 平台必須具備一致性與可擴展性，以便在各種環境 (包括地端、公有雲、混合雲及邊緣運算) 中提供統一的可見性
- **AI 就緒型基礎架構：**超越單純的容器，將 GPU 等資源及動態資源分配 (DRA) 作為標準平台服務來處理
- **以開發人員為中心：**內建針對叢集機群的修補、更新、升級及測試自動化功能，以實現集中式可觀測性，並讓開發人員能專注於交付應用程式程式碼與推動創新
- **資料服務：**資料服務與管理堆疊，包含用於儲存影像與多媒體的物件儲存、用於儲存結構化使用者資料的 SQL/NoSQL 資料庫，以及用於儲存日誌的檔案儲存。這確保了在運算資源之外，還能具備必要的資料常駐與儲存資源

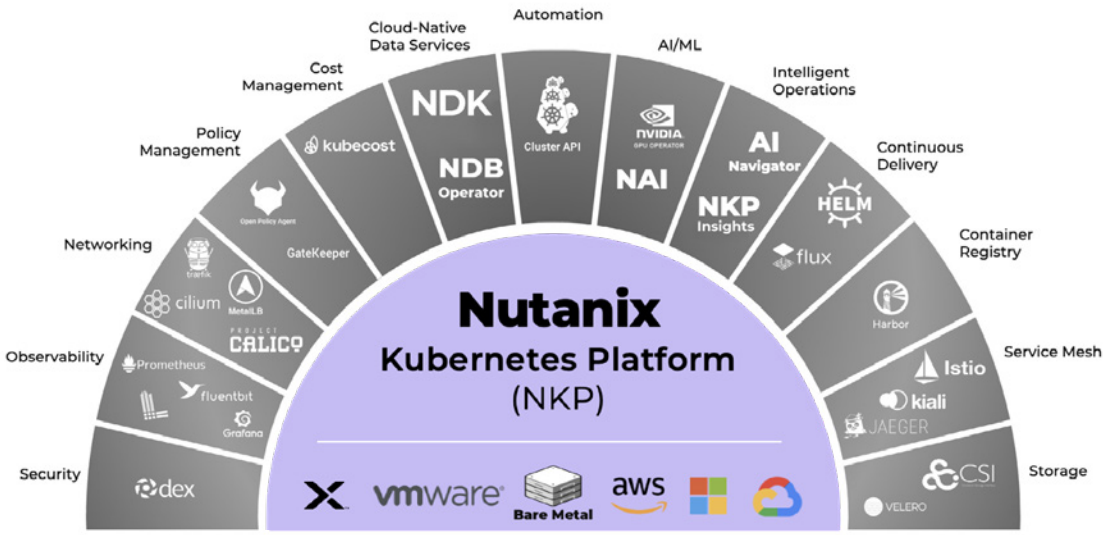
Nutanix 協助平台工程團隊部署一個基於純上游 Kubernetes 的企業級平台，並提供自助服務環境，讓開發人員能從程式碼提交一路快速推進至正式環境，無需因手動佈建基礎架構而延誤進度。

第三章
Nutanix Kubernetes 平台

Nutanix Kubernetes 平台 (NKP) 是一個完整、開放且企業級的平台，能為雲原生應用程式提供韌性、安全性及持續營運階段所需的管理能力。NKP 旨在透過一套立場鮮明的技術堆疊，消除 DIY Kubernetes 的整合門檻，該堆疊標準化了叢集的建置、升級、安全防護及監控方式，並能透過單一運作模式，在地端、邊緣及公有雲環境中部署機群。

關鍵功能包括：

- **企業級平台：**此全堆疊平台提供在實際執行環境中部署與執行容器化應用程式所需的所有元件，包括監控、安全性、入口閘道、持續交付及儲存。
- **開放性：**完全符合 CNCF 上游規範，不僅可存取經驗證的 CNCF 專案完整目錄，更能自由整合任何替代工具。
- **簡化生命週期管理 (LCM)：**只需升級單一平台，即可自動完成叢集部署、升級、作業系統以及附加元件的升級，而所有這些項目均已通過升級測試與相容性測試。
- **可觀測性與深入解析：**整合式日誌記錄、監控與警示功能。NKP Insights 與 AI Navigator 提供異常偵測功能，包含根本原因分析、對話式疑難排解介面，以及相關建議，幫助你更快解決問題。
- **適用於可重複建置的黃金範本：**NKP 提供標準化的「黃金路徑」及自動化功能，可在任何環境中實現一致且可重複的叢集組建。這有助於客戶避免隨著組織擴大容器部署規模時，因手動管理而經常出現的組態偏移問題。
- **透過原生整合實現加速交付：**原生 CI/CD 管道整合以及基於 GitOps 的協調機制，讓開發人員能夠從程式碼提交一路快速推進至正式環境，無需面對 DIY 平台所造成的基礎架構瓶頸。
- **跨環境的可攜性：**確保運作一致性，並可在任何地方執行工作負載。
- **將營運成本降至最低：**透過統一管理不同環境 (雲端、地端、邊緣) 中的叢集，NKP 有助於減少管理分散工具及備援團隊所產生的開銷。
- **Nutanix 企業級技術支援：** 提供可靠且持續的技術支援，淨推薦分數 (NPS) 高達 90 以上。



機群管理

- **宣告式機群協調：**NKP 運用 Cluster API 模式，為叢集生命週期管理提供統一且宣告式的介面，並透過 Kubernetes 控制器，自動將基礎架構在地端、雲端及邊緣環境中調整至預期狀態。
- **集中式機群管理：**透過單一控制面板，標準化機群中叢集的建立、設定、升級及監控流程。
- **一致的變更部署：**無需手動介入，即可將應用程式與組態更新推送至多個叢集。
- **整合式平台工具集：**NKP 包含一個經過驗證的 CNCF 專案目錄，以及網路、可觀測性、政策管理與安全性等平台元件，讓團隊能減少將工具拼湊在一起及重新測試相容性所花費的時間。
- **AI 驅動的診斷與運作：**NKP Insights 與 AI Navigator 扮演自動化可靠性工程師的角色，持續依據最佳做法對叢集進行掃描，在異常導致服務中斷前即予以偵測；同時提供基於自然語言的對話式介面，協助進行故障排除，進而縮短平均修復時間 (MTTR)。

透過單一平台，隨時隨地執行

組織可在各種環境中部署 NKP。NKP 可在地端無縫運作，包括 Nutanix 本身的 Nutanix 雲端基礎架構 (NCI) 平台、虛擬機、裸機伺服器、公有雲、邊緣據點，甚至實體隔離的據點。在 Nutanix 基礎架構上運行 NKP，可提供獨特的整合功能，不僅能簡化部署流程、加速營運，更能開啟一整套全方位的資料服務存取權限。此外，Nutanix 平台的分散式資料庫架構提供了額外的彈性層，進一步鞏固了 NKP 的基礎架構。

安全性與實體隔離支援

NKP 的設計內建了安全機制與各項功能，可支援客戶的合規計畫。它將身分識別、存取控制、政策強制執行、網路分割以及安全升級做法進行標準化，並包含對受限環境及實體隔離環境的支持。

- **集中式存取**
  - **身分驗證：**支援單一登入 (SSO) 及聯合身分驗證模式，以確保跨叢集的身分與存取權限保持一致。
  - **RBAC 與加密：**利用 Kubernetes 原生 RBAC 及加密功能，以滿足企業安全需求，並減少依叢集為單位的存取模式。
  - **合規支援：**提供相關功能，協助客戶在適用情況下符合 FIPS 140-2 等規範要求。
- **政策強制執行與網路控制**
  - **「政策即程式碼」(OPA)：**利用政策強制執行機制 (OPA Gatekeeper)，在不影響交付速度的前提下，一致地套用各項標準，例如准入控制與基準安全規則。
  - **網路控制：**支援雲原生網路方案 (Cilium/Calico) 以及 Kubernetes 網路政策，以進行 Pod/服務層級的流量控制。
  - **適用於 mTLS 的服務網格：**支援服務網格功能，以便在需要時啟用 mTLS，以確保服務間的安全性。
- **安全的生命週期與經過驗證的運作**
  - **生命週期一致性：**將核心平台安全元件部署並維護為經過驗證且版本一致的平台應用程式，以減少版本不一致及升級風險。
- **實體隔離/受限環境中的運作**
  - **離線封裝與派送：**支援離線封裝方案，以減少將影像/圖表移入受限網路時所需的人工操作。
  - **本機登錄檔選項**：啟用本機登錄檔模式，以便在環境內部儲存和管理映像檔。
  - **代理伺服器與離線彈性：**可與現有代理伺服器或本機登錄黨配合使用，因此實體隔離環境的設定無需進行一次性自訂操作。
  - **軍用級 DevSecOps：**
    - 經過預先掃描的發行版本與安裝套件，可確保部署安全且經過驗證。
    - 單一登入 (SSO) 與集中式角色導向存取控制 (RBAC)，以實現統一的存取控制。
    - 支援多租戶架構，以實現安全的工作負載隔離。
    - VPC 整合與網路政策可提供精細的流量控制。
    - 透過加密通訊來保護傳輸中的資料。
    - 透過政策強制執行與異常偵測，以支援主動式資安措施，並協助客戶符合合規要求。

雲原生技術展示系列

在此播放清單中深入瞭解 Kubernetes 的部署與管理，主題涵蓋自動化、安全性，以及簡化持續營運等。

Nutanix Kubernetes 平台（NKP）

了解 NKP 如何簡化平台工程，並在各種環境中提供一致的使用體驗。

準備好深入了解了嗎？

歡迎探索雲原生技術資源中心，取得技術部落格、操作教學影片，以及經過驗證的設計方案。

第四章
針對具狀態工作負載的企業級資料服務與管理

適用於 Kubernetes 的 Nutanix 資料服務 (NDK) 透過針對具狀態 Kubernetes 工作負載，提供具備應用程式感知的複寫與災難復原功能，將企業級儲存擴展至 Kubernetes，使平台團隊無需整合獨立的儲存或資料服務解決方案，即可保護資料並復原應用程式。開發人員現在可以在部署流程中，定義應用程式層級的快照與複寫排程。

這對平台團隊帶來以下好處：

- **適用於 Kubernetes 的企業級儲存解決方案：**Nutanix 統一儲存 (NUS) 為所有資料提供整合式永續性儲存體，包括對區塊、檔案、物件及資料庫的支援。其橫向擴展架構符合 Kubernetes 應用程式的建置與執行方式，確保跨叢集的擴展過程更為簡便。
- **資料庫即服務 (DBaaS)：**提供資料庫即服務 (DBaaS) 支援能力，可快速且可重複地部署 SQL、NoSQL 及向量資料庫。透過使用 NDB Kubernetes Operator，開發人員可運用「基礎架構即程式碼」的方式，直接在 NKP 內自動執行資料庫的佈建、修補程式安裝及備份作業，讓資料中心或邊緣環境也能享有與公有雲資料庫服務同樣的便利性。
- **NDK 可透過 NKP 應用程式目錄取得：**平台團隊可直接透過 NKP 啟用 NDK，輕鬆將應用程式感知複寫與災難復原納入 Kubernetes 平台的標準功能。
- **應用程式與命名空間層級的保護：**基於政策的備份、快照、複寫及災難復原協調機制，有助於降低 RPO/RTO 風險。
- **多據點治理支援能力：**透過跨叢集的一致性政策，實現地理上分散的資料保護，以滿足合規與治理需求。

適用於 Kubernetes 的 Nutanix 資料服務（NDK）

了解 Nutanix 如何確保應用程式韌性，並簡化雲原生營運管理。

第五章
平台團隊的營運成果

精簡化的生命週期管理

NKP 可為各類基礎架構供應商自動化執行 Kubernetes 的部署、擴展及升級作業。平台團隊無需將每次 Kubernetes 升級都視為一次客製化整合作業，而是可以善用核心平台功能之間已確認相容性的平台，大幅簡化升級流程。

此舉可限制跨叢集的版本差異、簡化硬體更新週期之間的協調、將升級負債降至最低，並使機群更容易保持修補程式最新狀態，同時避免引入不必要的營運風險。

整個機群的營運一致性

NKP 透過標準化 Kubernetes 的部署、設定與治理方式，實現跨叢集與環境的一致運作模式。無論叢集是在地端、公有雲或邊緣環境中執行，平台團隊都能在整個機群中套用相同的生命週期工作流程、安全控制措施及政策邊界。

這種一致性可以減少叢集之間的偏移，並確保無論工作負載在何處執行，或叢集最初以何種方式建立，都能達到正式環境所需的就緒標準。

更快速的軟體交付

NKP 旨在透過預先規劃的叢集和黃金映像，快速實現價值，並大幅縮短標準化重複部署所需的時間。團隊無需再將數十個獨立的開放原始碼元件進行整合並持續重新驗證，而是基於一個具備一致生命週期的完整、整合式平台層來運作。透過減少對手動相依關係追蹤、版本相容性測試以及特定環境指令碼的依賴，NKP 釋放了平台資源，使其能專注於更高價值的工作。NKP 能透過統一的自助服務體驗，協助縮短產品上市時間，讓開發人員能夠輕鬆、即時地使用關鍵工具與資料服務。

準備好深入了解了嗎？

歡迎探索雲原生技術資源中心，取得技術部落格、操作教學影片，以及經過驗證的設計方案。

NUTANIX

info@nutanix.com | www.nutanix.com | @nutanix

©2026 Nutanix, Inc. 版權所有。Nutanix、Nutanix 標誌以及文中提及的所有 Nutanix 產品與服務名稱，均為 Nutanix, Inc. 於美國及其他國家/地區的註冊商標或商標。
Kubernetes 是 Linux Foundation 於美國及其他國家/地區的註冊商標。文中提及的所有其他品牌名稱僅供識別之用，且可能為其各自擁有者的商標。
FM-CN-What It Takes to Build a Production-Ready Kubernetes Platform-FY27Q1.-zh_TW-100226