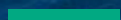


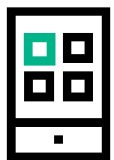


**Hewlett Packard
Enterprise**

HPE Ezmeral Container Platform



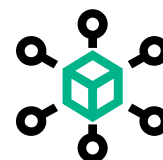
對企業而言容器技術面臨的挑戰與基本要求



如何規劃**K8S**的混合式基礎架構，並且可以統一納管？



如何搬移現有的服務至容器環境？



如何規劃須後端儲存的環境與應用？



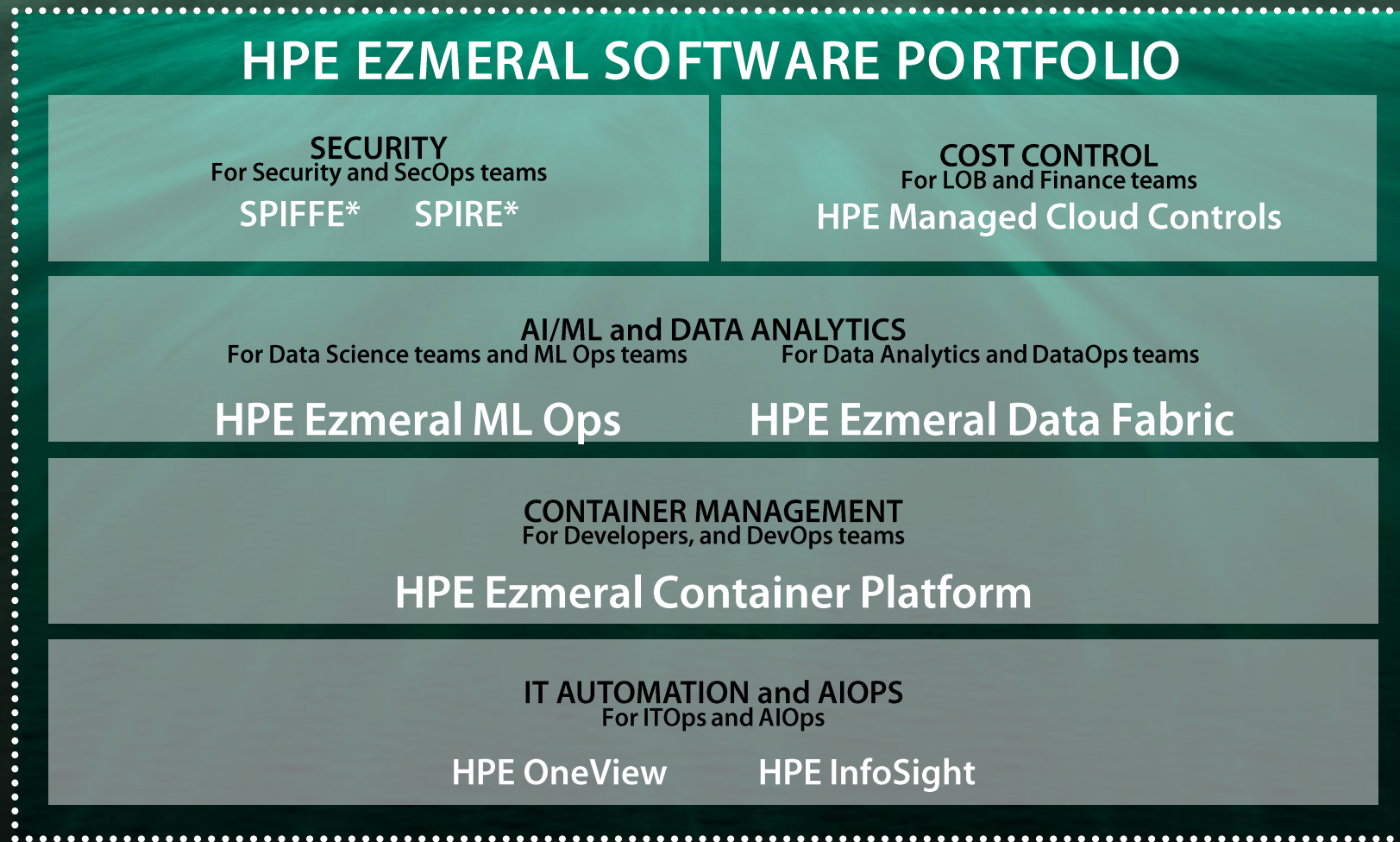
如何提供企業級的安全與最佳化的成本效益

HPE Ezmeral

用於運行，管理，控制和保護運行您的業務的應用程序和數據的軟體-推動整個組織中從邊際到雲的數據驅動的數位轉型



HPE Ezmeral overview



HPE GreenLake

*Open source projects



Introducing the HPE Ezmeral Container Platform

業界第一個企業級容器平台，專為帶有開源Kubernetes原生的雲應用和分散式非原生的雲應用而設計，可在本地端、公有雲和邊際的任何位置實現真正的混合雲操作。

Cloud native and non-cloud native apps



HPE Ezmeral Container Platform



Any hardware



Any cloud

- 將容器的速度和效率帶到雲微服務應用和新應用程式
- 使用裸機安裝容器降低OS License成本，通過企業級安全性降低風險
- 通過一鍵式容器部署更快交付與發佈
- 為Kubernetes提供跨雲與跨基礎設施管理

Leveraging differentiated IP from recent HPE acquisitions

Open source Kubernetes plus software innovations from BlueData and MapR



kubernetes



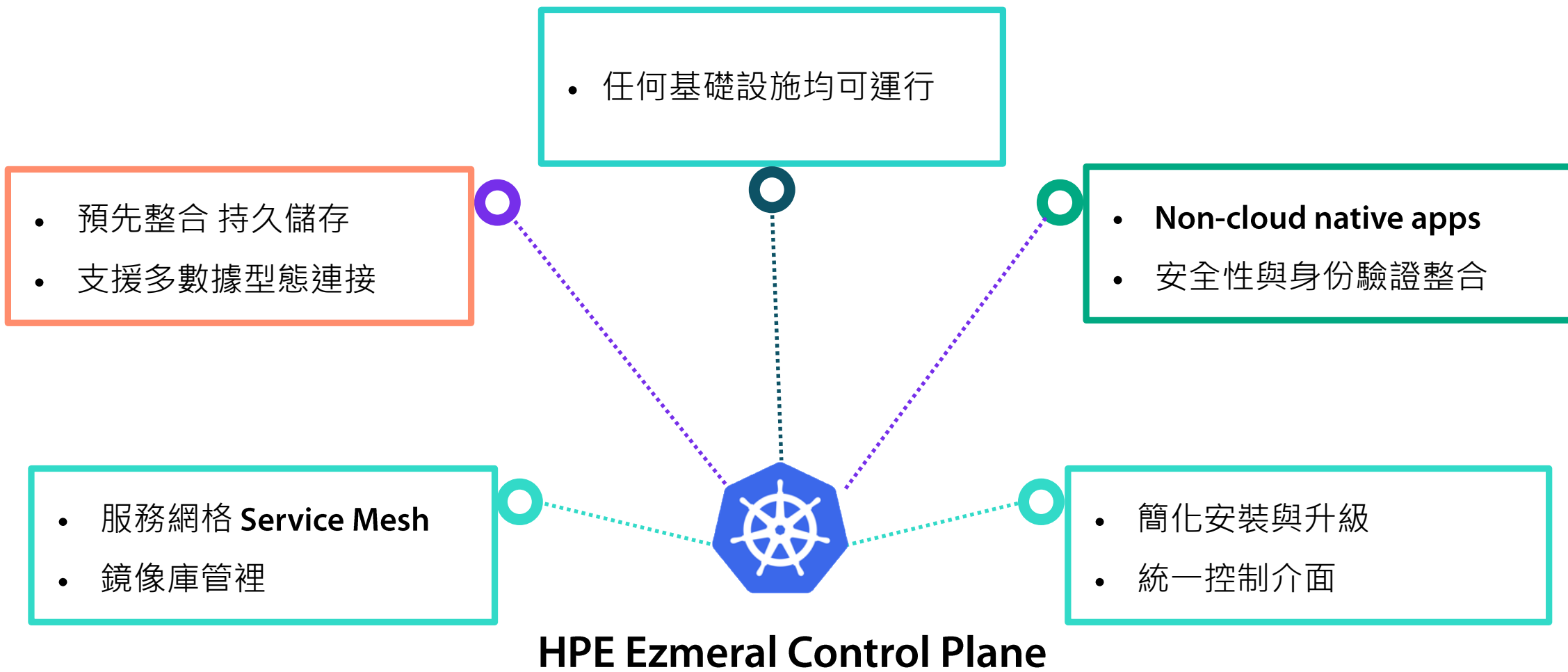
Open source standard for container orchestration—with rapidly growing adoption

Enterprise container-based software platform—initially focused on AI/ML and analytics

Data fabric for Kubernetes—with persistent storage and universal data access for containers

Bluedata with kubernetes orchestration

多租戶、多叢集管理, 100% open source upstream Kubernetes



HPE Ezmeral Data Fabric for Kubernetes

High performance, unified view of all data from edge to core to cloud

Enterprise data persistence

多租戶架構與控制

單一介面數據存取



GLOBAL
NAMESPACE

Edge | Core | Cloud
DISTRIBUTED FILE
SYSTEM



MULTI-PROTOCOL

POSIX, NFS, HDFS, S3,
REST, JSON, NoSQL,
KAFKA

Effortless data scale

Cold, Warm, Hot Data

資料分層



MULTI-
TEMPERATURE

Cold | Warm | Hot



CLOUD SCALE

Secure and portable

安全取得資料

不受限任何基礎設施



SECURE

Common security and
governance model

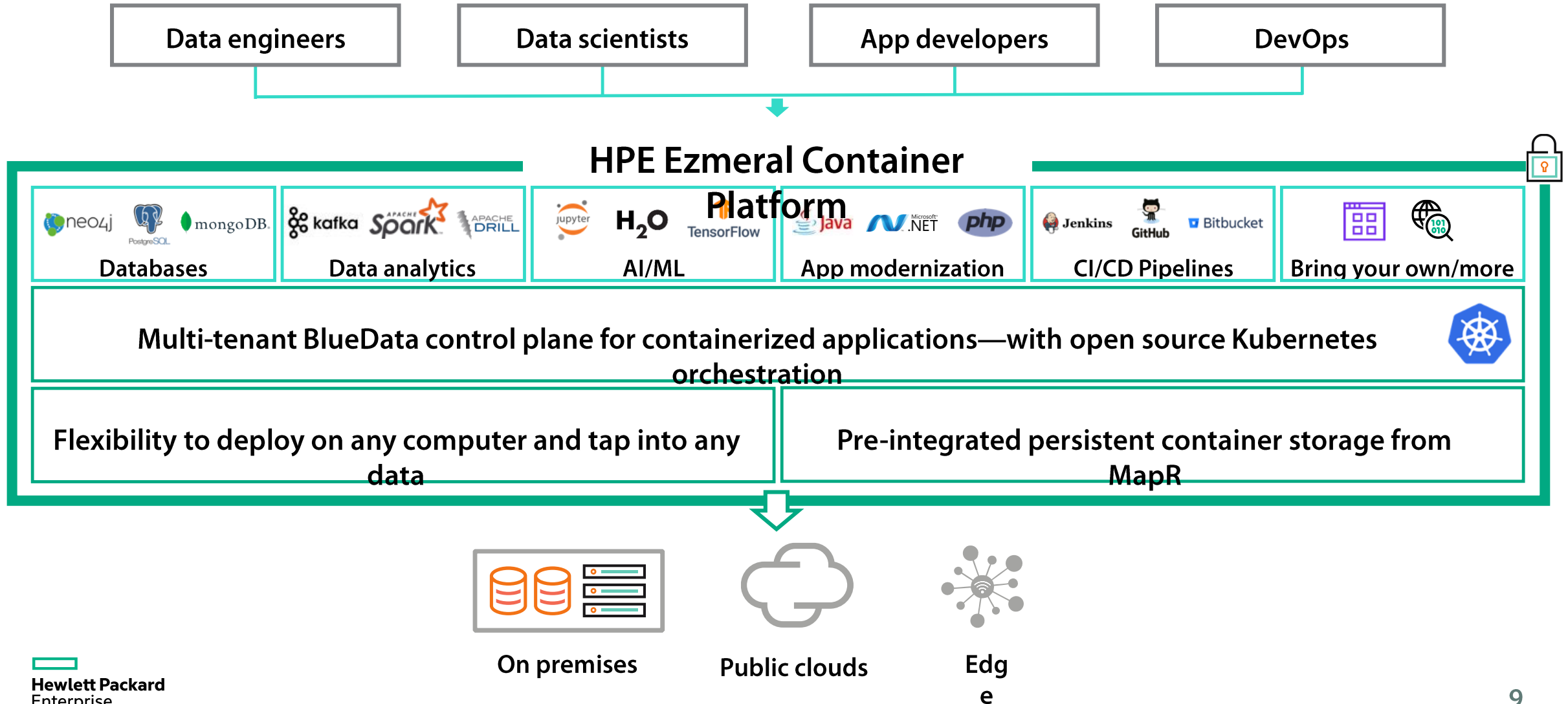


DATA AND APP
MOBILITY

Edge | Core | Cloud

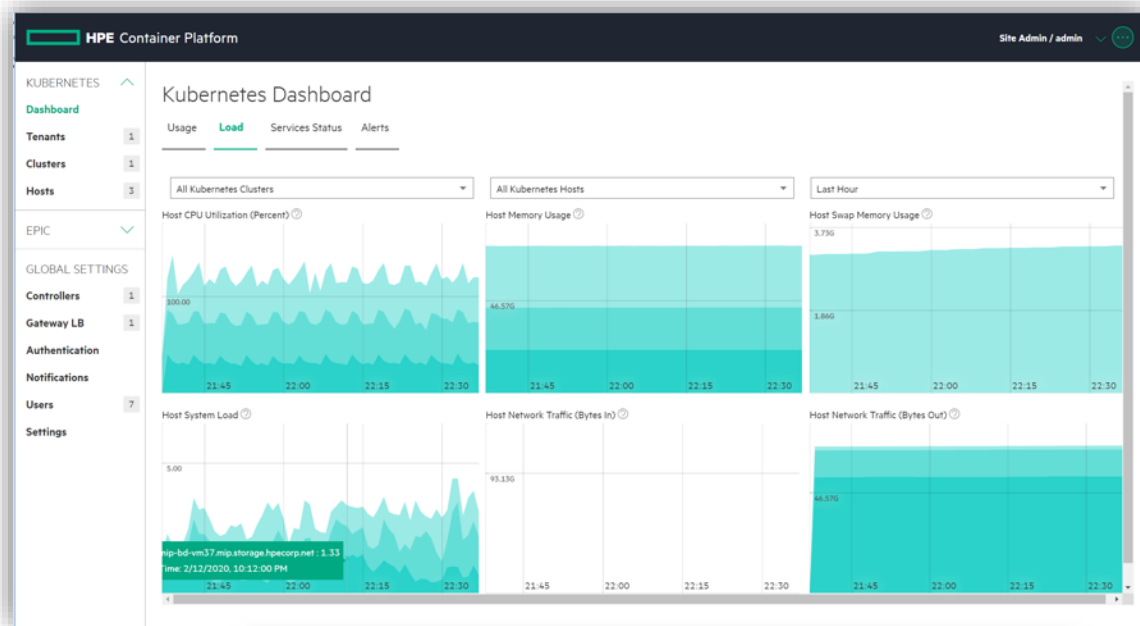
HPE Ezmeral Data Fabric

HPE Ezmeral Container Platform



容器平台『視覺管理』

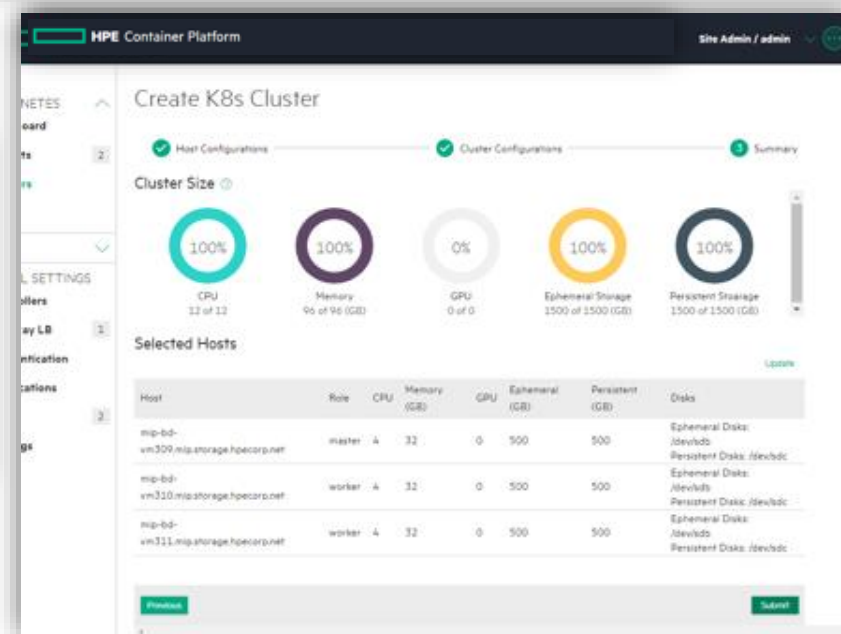
- Kubernetes元件狀態管理
- 可配置電子郵件發送警報
- 單一介面查看Kubernetes Pod資源使用狀態



Kubernetes Dashboard

Usage Load **Services Status** Alerts

Name	BD Agent	Monitoring Collector	Disk Pressure	Docker Daemon	Kube API Server	Kube Controller	Kube Proxy	Kube Scheduler	Kubelet	Memory Pressure	Network	FileServer	MountPoint	PosixClient	Warden
mip-bd-vm36.mip.storage.hpccorp.net	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
mip-bd-vm37.mip.storage.hpccorp.net	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
mip-bd-vm38.mip.storage.hpccorp.net	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●



容器平台『大規模部署』

- 大規模叢集管理以利故障排除
- 簡單易操作的UI介面
- 可支援超過**1000+**大規模部署

Kubernetes Cluster

		Refresh Kubernetes Manifest		Kubernetes Manifest Log		Create Kubernetes Cluster		Import Kubernetes Cluster	
☐	Cluster Name	Version	Hosts	Type	Resources	Details	Status	Actions	
☐	K8sC1	1.17.5	✓ master (5) ✓ worker (800)		CPU Cores: 1690 Memory (GB): 12290 GPUs: 0 Ephemeral Storage (GB): 402500 Persistent Storage (GB): N/A	Created At: Thu Jul 30 2020 19:13:00 Created by: admin	● ready	🗑️ ✎ 📁 ⬇️ 📁 ⬆️ 👤 🔗	

Kubernetes Dashboard

Usage Load **Services Status** Alerts

Tenants 2
Clusters 1
Hosts 805

EPIC
GLOBAL SETTINGS
Controllers 3
Gateway LB 1
Authentication
Notifications
Users 3
Settings

Name	BD	Disk	Docker	Kube API	Kube	Kube	Kube	Kubelet	Memory	Network
ip-10-0-158-255.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-99-222.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-73-36.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-70-60.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-66-130.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-50-232.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-47-30.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-32-109.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-241-65.ec2.internal	●	●	●	●	●	●	●	●	●	●
ip-10-0-225-225.ec2.internal	●	●	●	●	●	●	●	●	●	●

Rows per page: 10 1-10 of 805

容器平台『擴縮節點』『版本管理』『滾動式更新』

■ Kubernetes多版本管理

■ 依據租戶資源限制

Create New K8s Tenant

Tenant Name

Tenant Description

K8s Cluster

Adopt Existing Namespace ☒ (Optional)

Existing Namespaces

Is Namespace Owner ☒ (Optional)

Map Services To Gateway ☒ (Optional)

Quotas

External Authentication

External User Groups (Optional)

Member

Create Kubernetes Cluster

1 Host Configurations 2 Cluster Configurations 3 Summary

Kubernetes Version*

Pod Network Range

Service Network Range

Pod DNS Domain

HPE CSI Driver for Kubernetes

HPE Nimble Storage ☐

Kubernetes Root CA Certificate Browse

Kubernetes Root CA Private Key Browse

Previous Next

容器平台『資源分配』『單一窗口對外提供服務』

- 依據需求分配給各團隊
- 網管人員統一管理服務

Quotas

Edit K8s Tenant

Tenant Name

k8s-ct2

Tenant Description

k8s-ct2

K8s Cluster

k8s-1

Adopt Existing Namespace

☐

(Optional)

Specified Namespace Name

k8s-ct2

(Optional)

Is Namespace Owner

☒

(Optional)

Map Services To Gateway

☒

(Optional)

Enable Istio Service Mesh

☒

(Optional)

Mutual TLS mode

permissive

AI/ML Project

☐

Maximum Cores

20

Maximum Memory (GB)

100

Maximum Ephemeral Storage (GB)

20

GPU Devices

5

Tenant Storage(GB)

105

istent Storage (GB)

110

HPE Container Platform

(Acting as Tenant Admin) DeployPipelines / admin

Dashboard

Users0

DataTaps1

FsMounts1

Applications

Service Endpoints

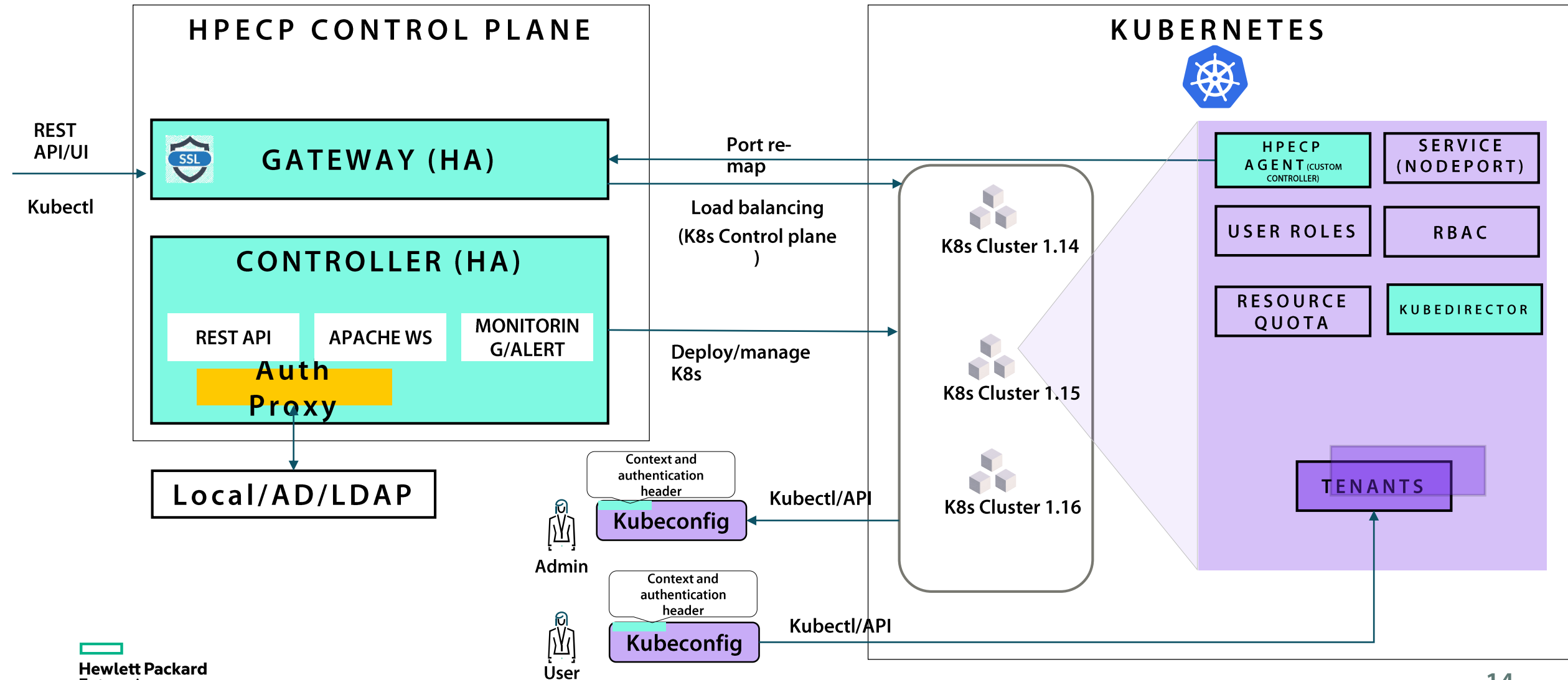
Kubernetes Service Endpoints

Name	Details	Services	Ports	Gateway Mappings
istio-ingressgateway		status-port	15020	mip-bd-vm180.mip.storage.hpecorp.net:10006
		http2	80	mip-bd-vm180.mip.storage.hpecorp.net:10007
		https	443	mip-bd-vm180.mip.storage.hpecorp.net:10008
		tcp	31400	mip-bd-vm180.mip.storage.hpecorp.net:10009
		kiali	15029	mip-bd-vm180.mip.storage.hpecorp.net:10010
		prometheus	15030	mip-bd-vm180.mip.storage.hpecorp.net:10011
		grafana	15031	mip-bd-vm180.mip.storage.hpecorp.net:10012
		tracing	15032	mip-bd-vm180.mip.storage.hpecorp.net:10013
		tls	15443	mip-bd-vm180.mip.storage.hpecorp.net:10014
s-kdss-65g6f-0	KubeDirectorCluster ID: centos7x-instance KubeDirectorApp: ID: centos7x Name: CentOS 7.5 Role: vanilla_centos	ssh	22	mip-bd-vm180.mip.storage.hpecorp.net:10019

Connected to terminal

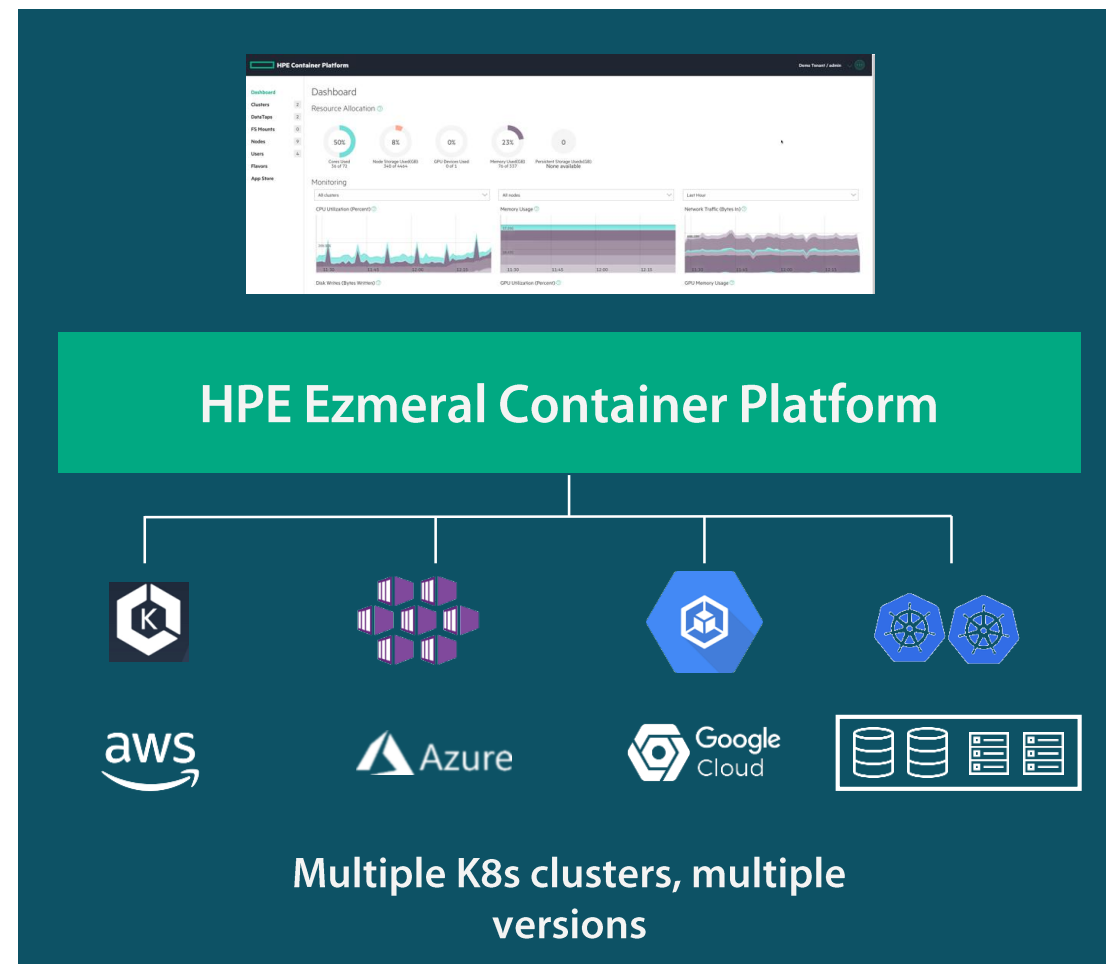
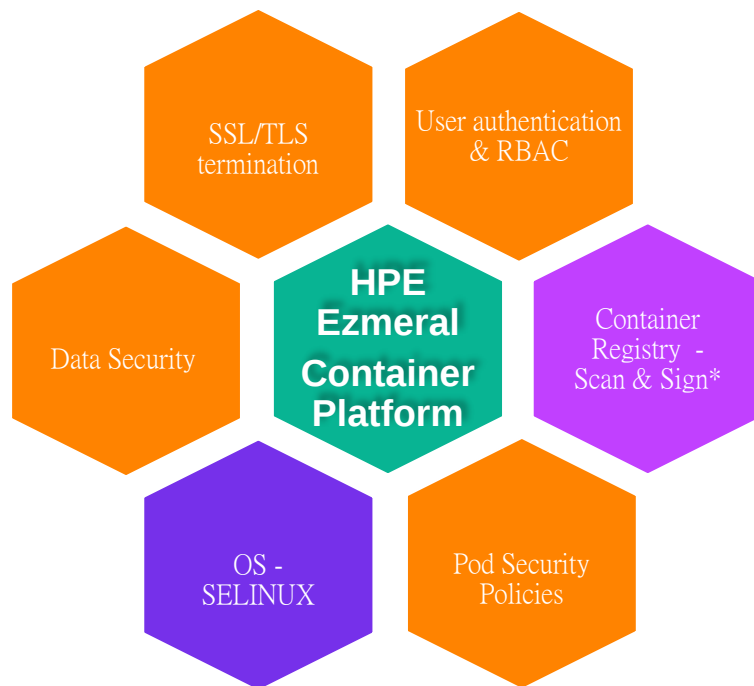
Terminate

簡易好維運的『高可用架構』與『集中式管理』



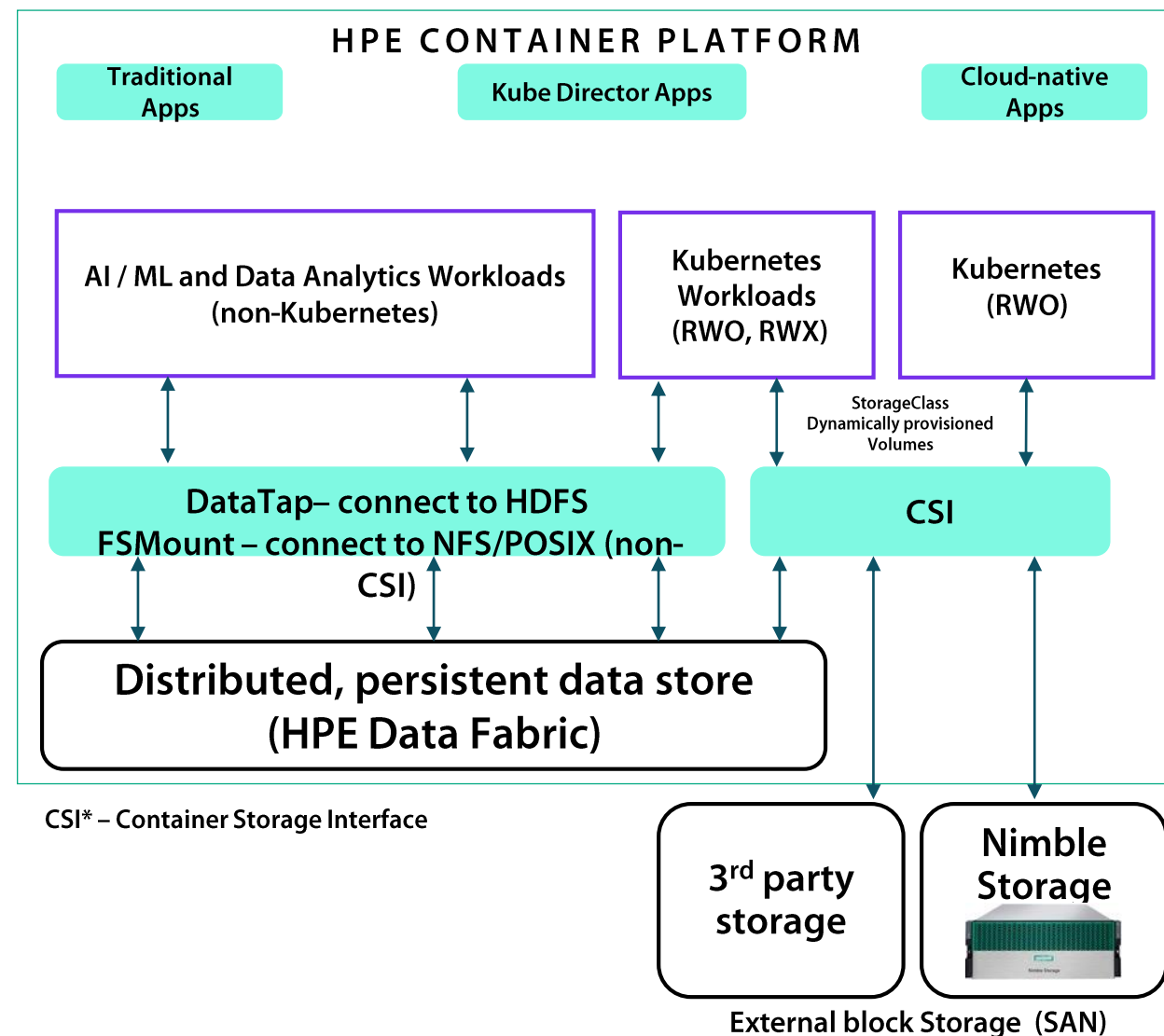
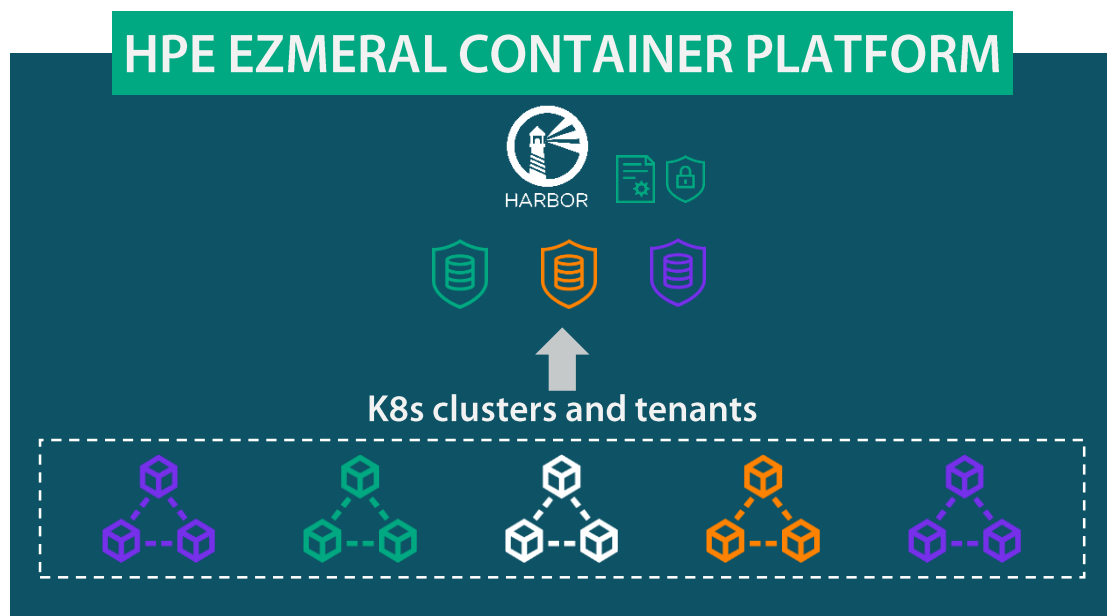
容器平台『企業等級安全性』『跨平台管理』

- 整合安全性登入與控制
- 支援多平台安全管理Kubernetes



容器平台『鏡像管理』『儲存管理』

- 支援多種儲存介面
- 容器服務存取外部資料無縫接軌
- 依據租戶鏡像庫統一管理



REST API

- 協助維運人員的API 介面
- Swagger 快速索引API
- Link: <http://<controller ip>/apidocs>

Code	Description
200	List of Kubernetes cluster services was successfully fetched.

Media type

Controls Accept header.

Example Value | Schema

```
{
  "_embedded": {
    "k8sservices": [
      {
        "service_name": "string",
        "namespace": "string",
        "endpoints": [
          {
            "proxy_service_name": "k8s:13:tenant-abc12:100:32055",
            "protocol": "string",
            "service_port": 0,
            "gateway_host": "string",
            "gateway_port": 0
          }
        ]
      }
    ],
    "_links": {
      "self": {
        "href": "/api/v2/cluster/1/action_history/action_28/node_status/bluedata-11"
      }
    }
  },
  "_links": {
    "self": {
      "href": "/api/v2/cluster/1/action_history/action_28/node_status/bluedata-11"
    }
  }
}
```

 **Swagger**
Supported by SMARTBEAR

Select a definition
Sessionless API

HPE Ezmeral Container Platform Sessionless API ^{v2} ^{OAS3}

/apidocs/epic_sessionless.json

Operations available when not logged in.

Servers

Authorize 

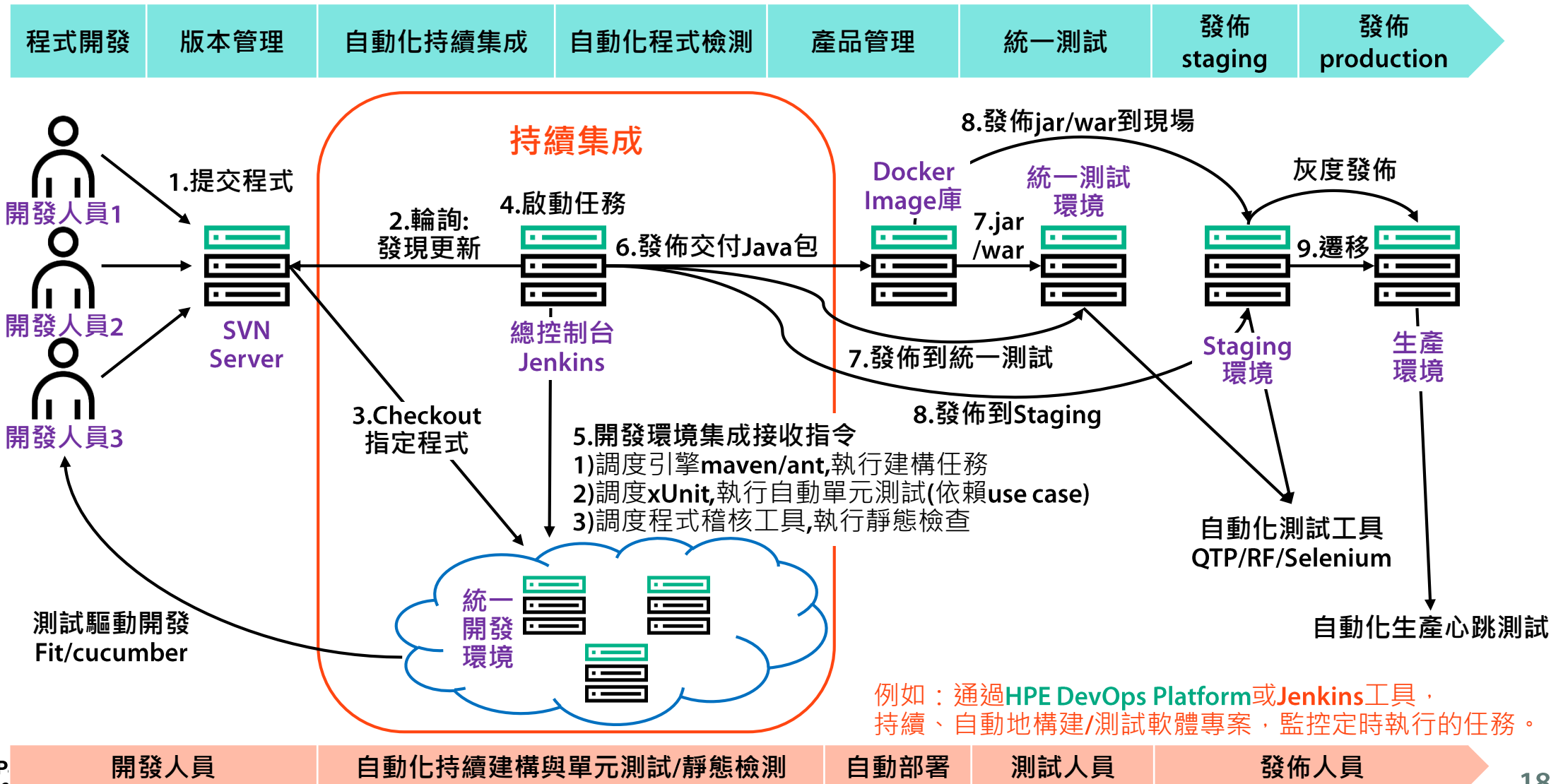
config

GET	/config	
GET	/config/auth	

每個服務都該考量引入DevOps理念

CI/CD是軟體發展的實踐

讓團隊在DevOps基礎上收到回饋並加以改進，不必等到開發的後期才尋找和修復問題。



Service Mesh

- 快速佈建網格服務
- 建立服務間連線規則

Overview

Graph

Applications

Workloads

Services

Istio Config

Namespace: k8s-ct2

App graph

Display

Find...

Hide...

Nov 19, 01:12:46 AM ... 01:13:46 AM

```
graph LR; unknown((unknown)) --> kiali1((kiali)); kiali1 --> kiali2((kiali)); kiali2 --> prometheus[prometheus.istio-system.svc.cluster.local]; kiali2 --> kubernetes[kubernetes.default.svc.cluster.local]; kiali2 --> istiod[istiod.istio-system.svc.cluster.local];
```

Legend

1 2

Graph tour

Last 1m

Every 15s

Host Configurations

Current Graph: 1 app (1 versions), 4 services, 5 edges

Incoming	Outgoing	Total
HTTP (requests per second):		
Total	%Success	%Error
1.69	100.00	0.00

0 25 50 75 100

OK 3xx 4xx 5xx

Create Kubernetes Cluster

Select from the list of applications

HPE Nimble Storage

Harbor container image registry

Istio

Enable packaged apps

Previous

Create New K8s Tenant

Tenant Name

Tenant Description

K8s Cluster

Adopt Existing Namespace

Specified Namespace Name

Is Namespace Owner

Map Services To Gateway

Enable Istio Service Mesh

AJML Project

什麼是服務網格 (Service Mesh) ？

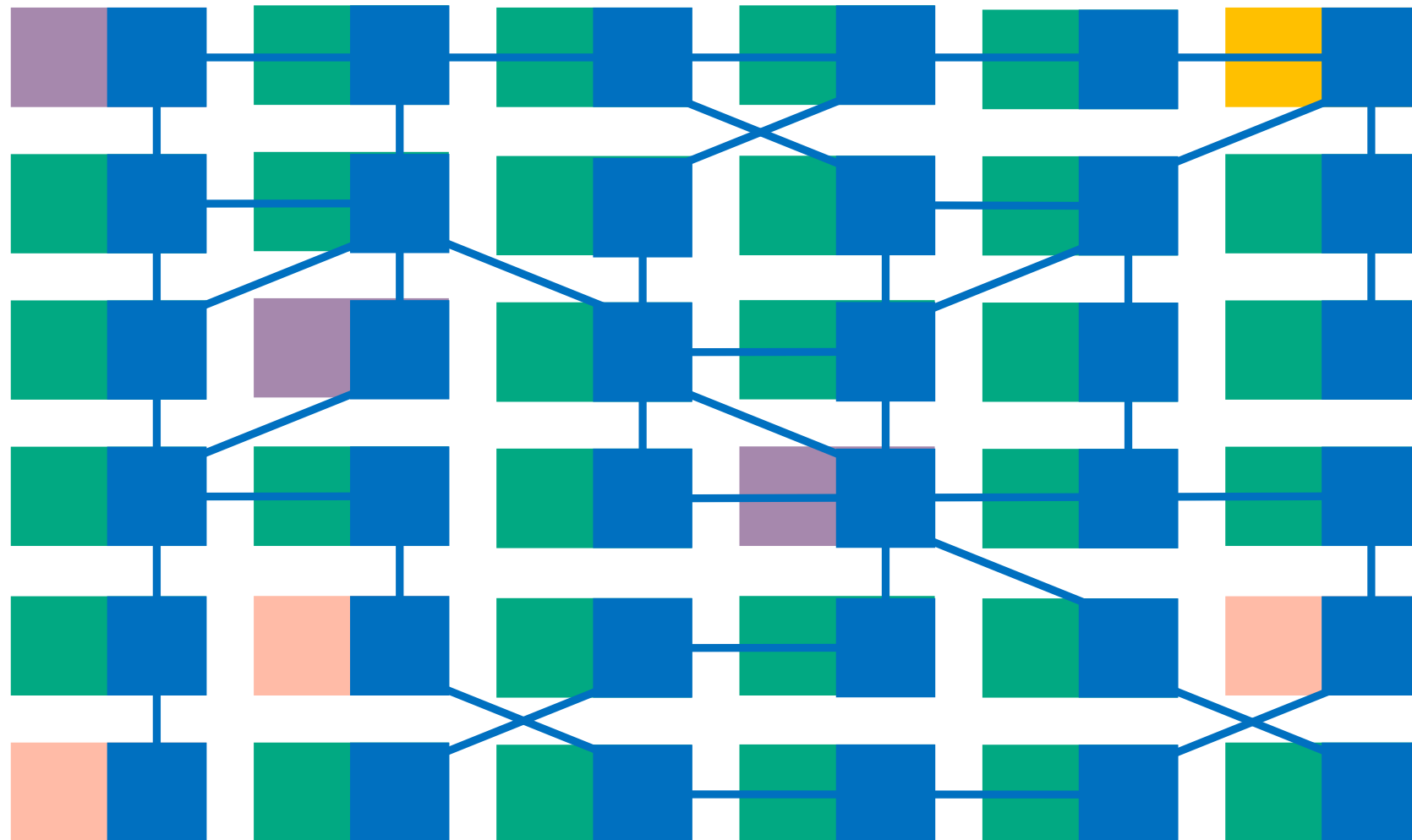
解決開發和運維人員從單體應用走向微服務架構的轉型過程中面臨的諸多挑戰

■ 主要問題

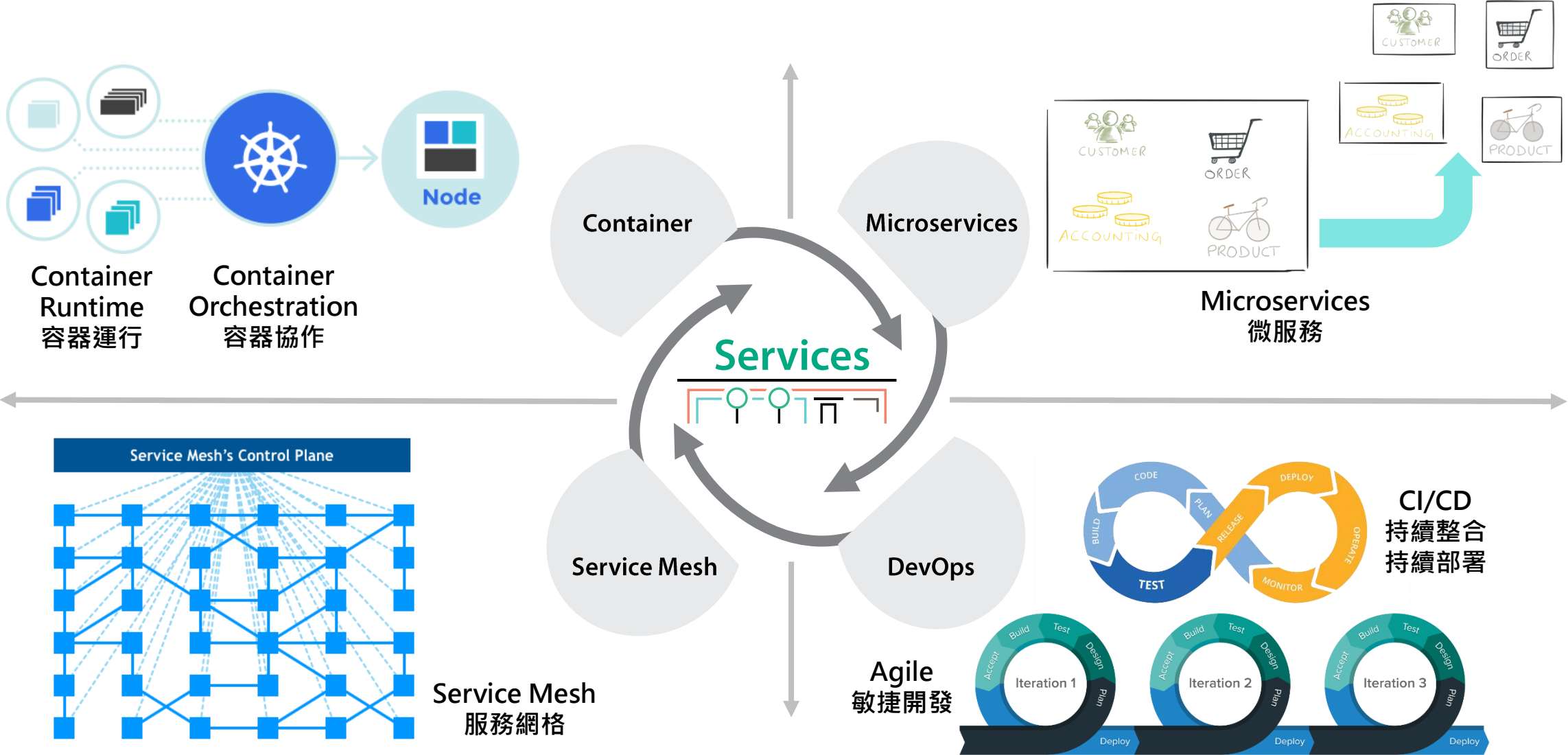
- 服務註冊與發現
- 負載均衡
- 故障恢復
- 存取控制
- 證書管理
- 指標收集和監控
- 運維需求

■ 解決方案

- 利用**sidecar**代理
- 獨立的元件及框架
- 專注於服務間的通信
- 與開發語言無關
- 已充分整合**Kubernetes**



服務為導向的容器基礎架構結合相關方案實現了企業的期望



一切都是為了提供服務...

更迅速的服務交付

更靈活的服务管理

更彈性的服務能力



Cloud-Native Application

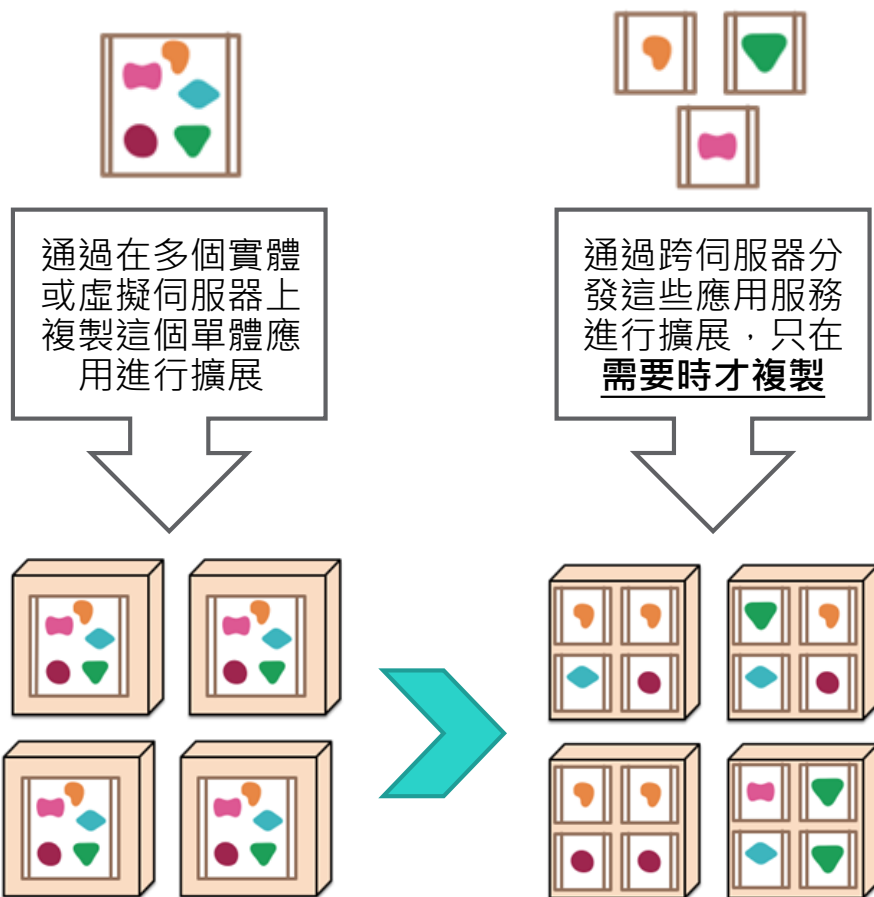
- 容器化封裝 (Container packaged)
- 動態管理 (Dynamically managed)
- 微服務導向 (Microservices oriented)

- 程式維護成本？
- 人員培養成本？
- 問題修復成本？
- 技術架構演進成本？
- 系統擴展成本？
- 其他成本...？



成功案例：微服務概念與考量

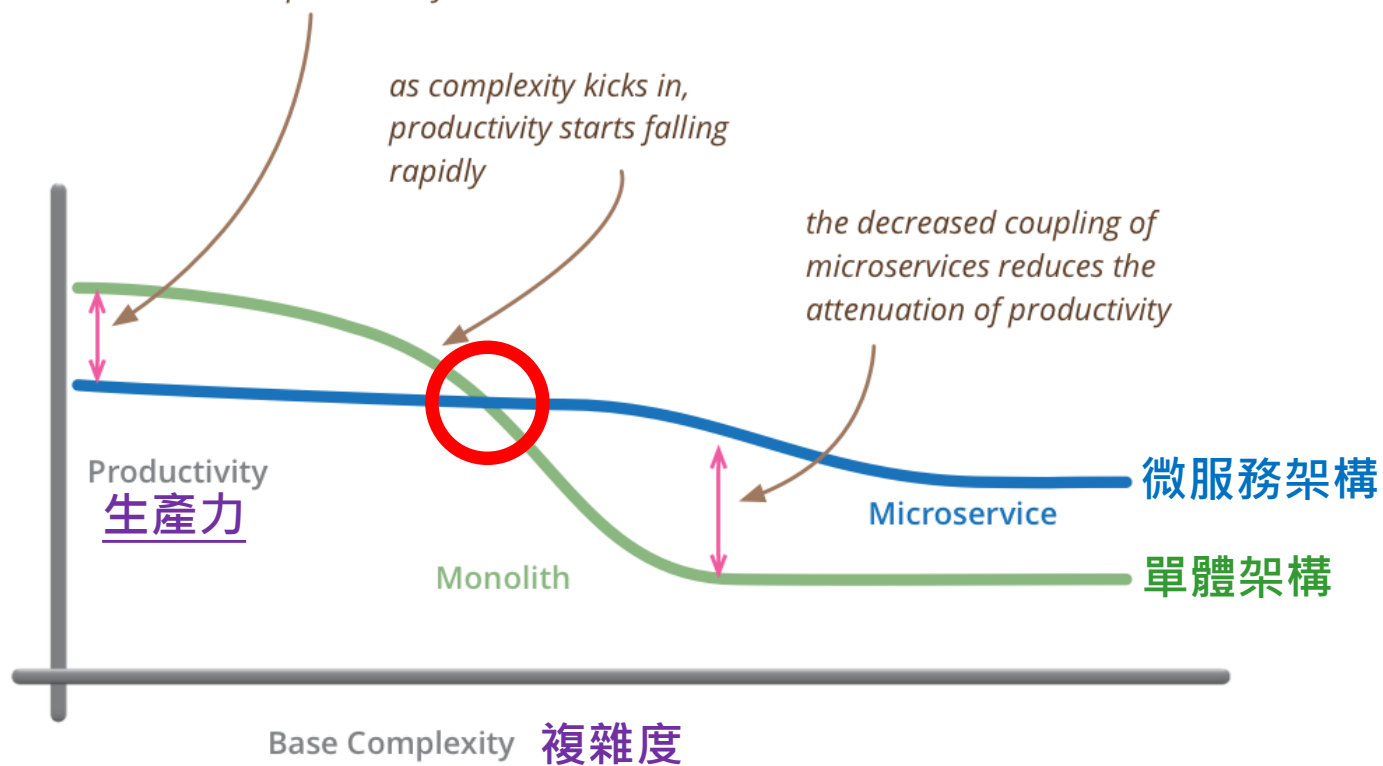
微服務架構風格是一種將一個單一應用程式開發為一組小型服務的方法，每個服務運行在自己的進程中，服務間通信採用羽量級通信機制(通常用HTTP資源API)。這些服務圍繞業務能力構建並且可通過全自動部署機制獨立部署。這些服務共用一個最小型的集中式的管理，服務可用不同的語言開發，使用不同的資料存儲技術。



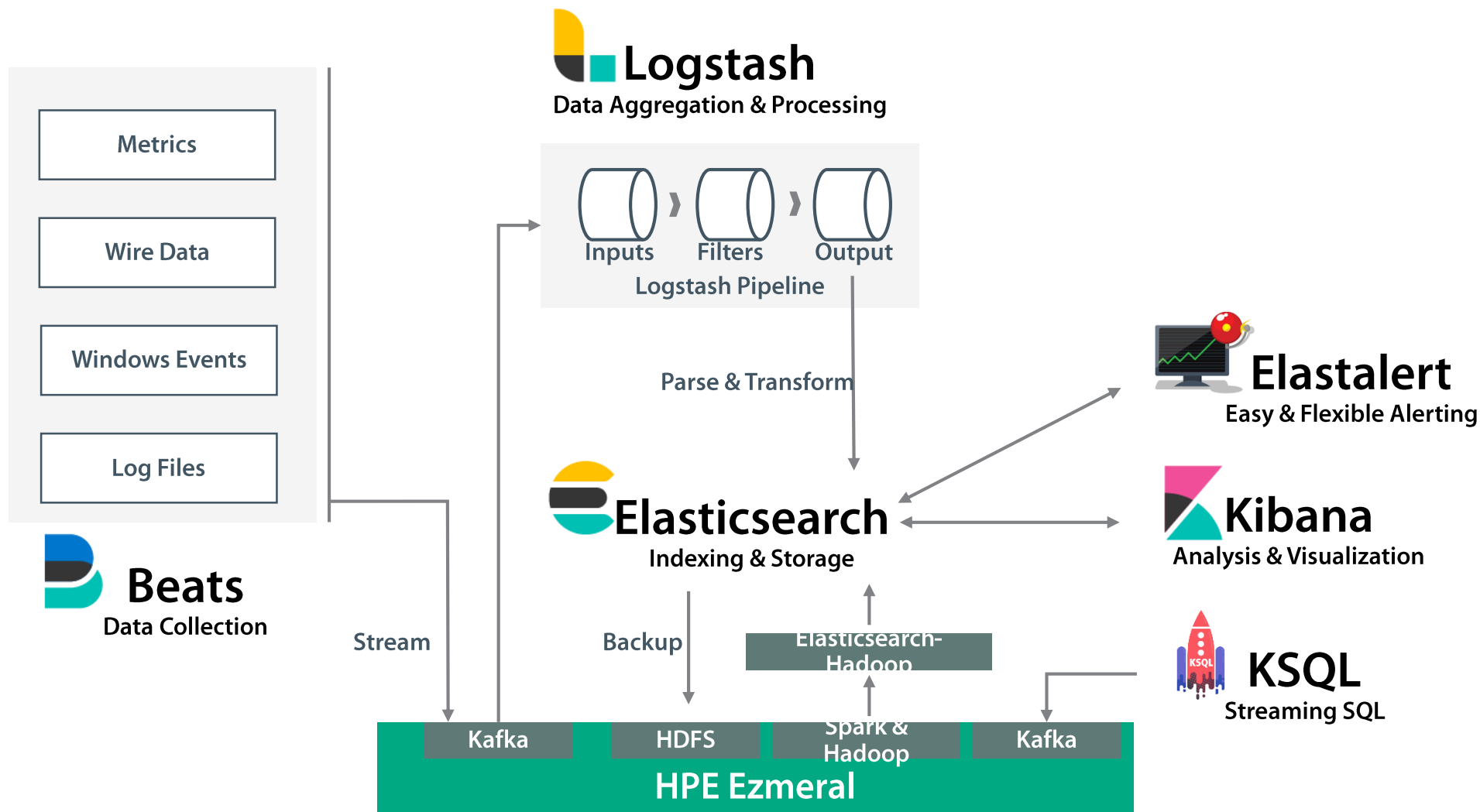
for less-complex systems, the extra baggage required to manage microservices reduces productivity

—— 摘自馬丁.福勒先生

<http://martinfowler.com/microservices>

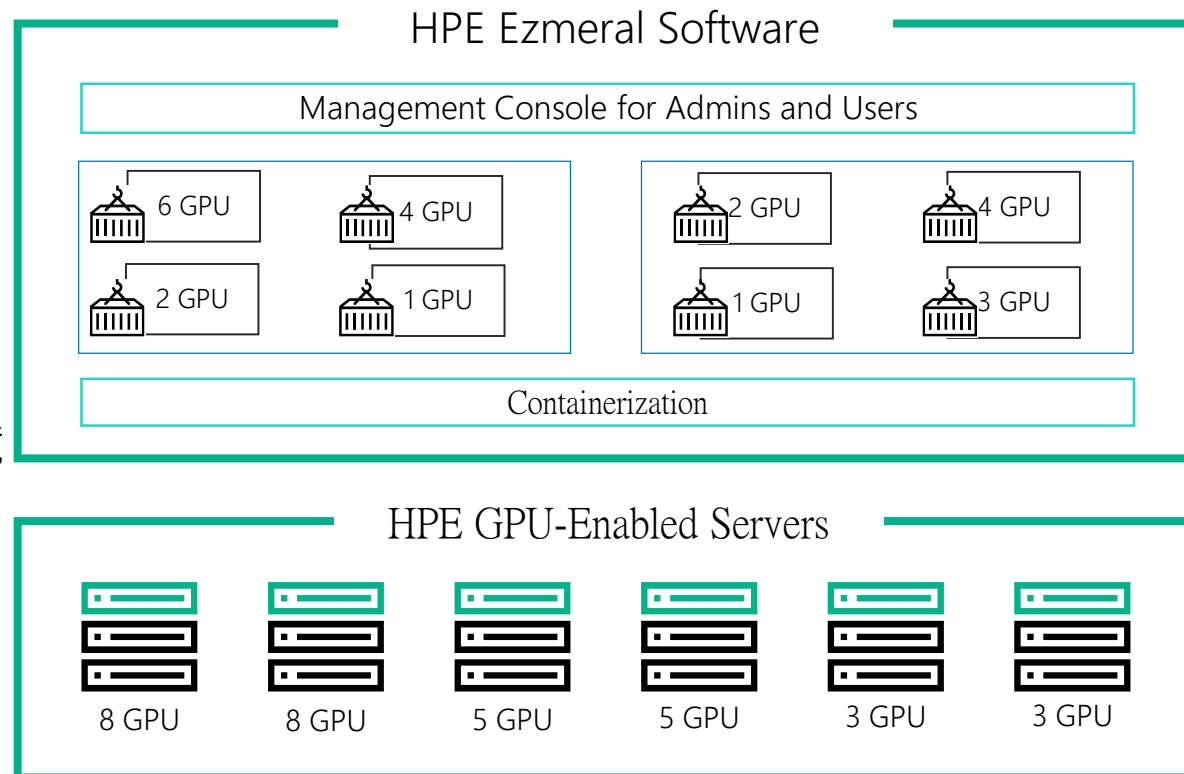


成功案例：ELK Stack Log收集

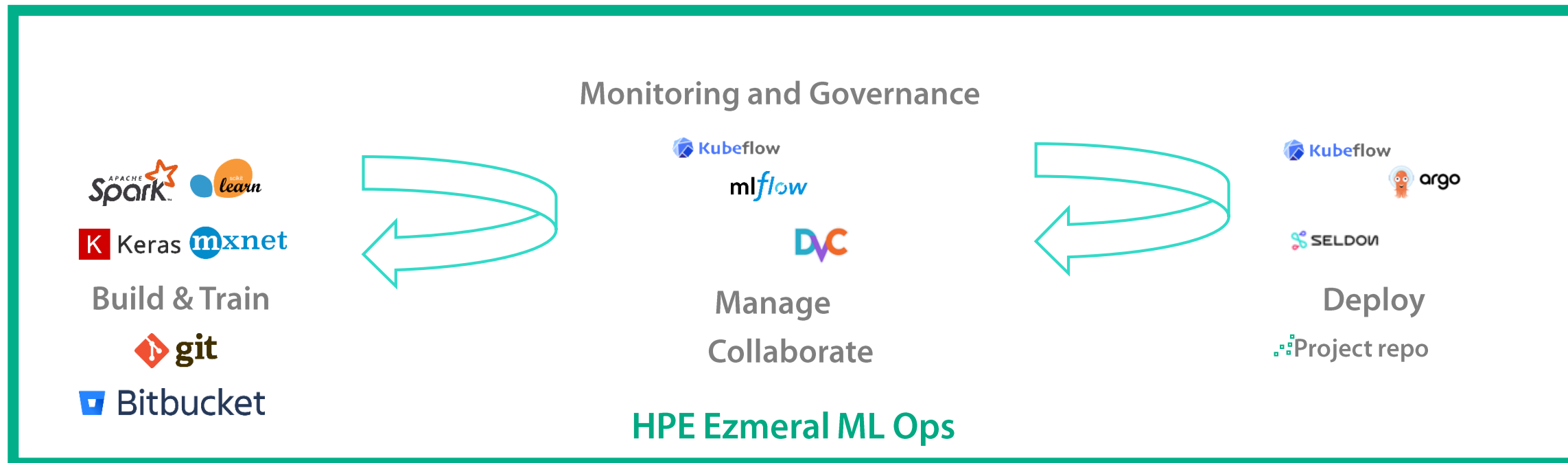


成功案例：GPU as a Service

- 為多個資料科學家專案分配和追蹤**CPU**資源配額
- 多租戶架構整合**AD**與**LDAP**安全性隔離
- 滿足各種工作需求分配**GPU**資源
- 啟動與停止容器使得未使用之資源釋放
- **GPU**資源池,不限制任何伺服器,虛擬化與雲端環境



未來趨勢：HPE Ezmeral MLOps



HPE如何以容器技術協助企業客戶邁向成功？

○ 容器技術降低傳統應用雲端化的遷移難度

將應用和資源池整合在一起，可以不對網路和存儲資源等進行改造，降低了傳統的應用構架雲端化改造難度。應用系統只要實現資料的分割，就能很快速的實現自己容器的設計和構建，形成生產力。

○ 容器技術簡化應用的開發，測試和部署

將應用打包成registry存儲的(registry-stored)型態，僅通過一行命令就可以完成元件的部署。無論你想將服務部署在哪裡，容器都可以從基礎上簡化服務部署的工作，實現開發，測試，部署，與上線的標準化。

○ 容器技術快速可用和彈性擴展

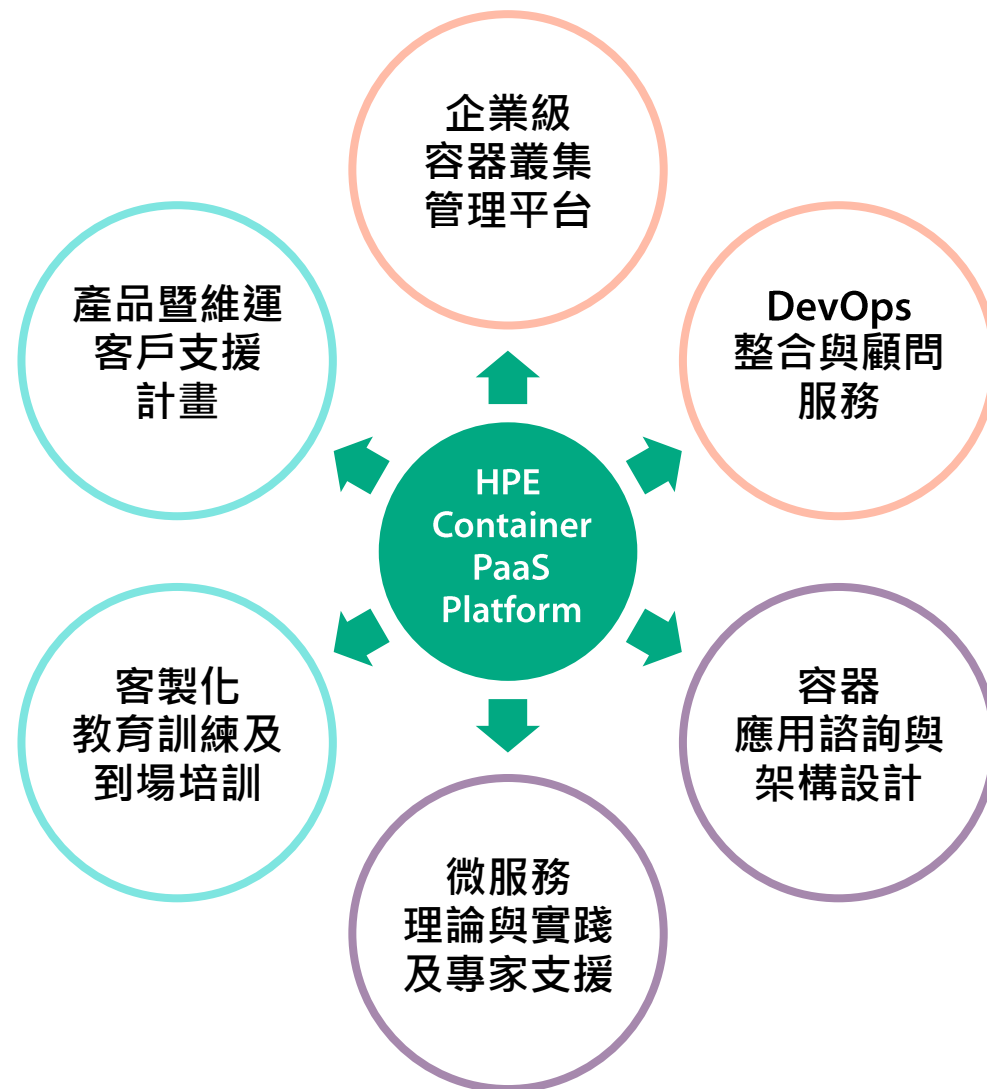
對作業系統的資源進行再次抽象，通過這種方式，打包好的服務可以秒級的時間內啟動，並且可以自由的調整服務集群的大小。

○ 容器技術結合微服務化，對業務在實現SLA分級控制

通過使用不同SLA的容器，可以從資源層面區分業務的SLA控制。例如關鍵的業務使用小機與企業級資料庫，普通業務使用x86與一般資料庫，從系統穩定性，性能和資料的安全性方面進行分級。

○ 容器技術提高系統對業務劇烈波動的適應性

新型態業務的波動性加大，並且要求業務24小時高連續性。按容器管理模式構建的系統，其標準化，模組化，靈活性，可以快速的形成生產能力，動態提高對業務劇烈波動的適應能力。





Hewlett Packard
Enterprise

THANK YOU

