

Oracle Database World

Rethinking Database Architecture to Embrace Modern App Dev

重新思考資料庫架構，促進現代化
新一代資料導向應用程式開發

Rick Chuang 莊復貴

首席雲端顧問

甲骨文



現代化的應用系統開發

應用系統開發的模式正處於快速轉型階段

現代化應用系統開發的模式正在簡化和加速應用程序開發

Deploying Apps



As a Service Building Blocks Containers



Kubernetes Cloud

Architecting Apps






Microservices Events Data Mesh






API Driven Low Code

Data Models and Data Analytics



Machine Learning Documents Graph Analysis



Text Search Spatial Processing Blockchain



現代化的應用系統開發

可是,這些應用系統開發的模式可以**極大地複雜化資料庫架構**

為了讓這些應用系統開發的模式可以成功,必須重新考慮資料庫架構

這個簡報說明 Oracle's **Decentralized Database Architecture** 如何擁抱現代化的應用系統開發

Deploying Apps






As a Service Building Blocks Containers





Kubernetes Cloud

Architecting Apps






Microservices Events Data Mesh






API Driven Low Code

Data Models and Data Analytics



Machine Learning Documents Graph Analysis



Text Search Spatial Processing Blockchain

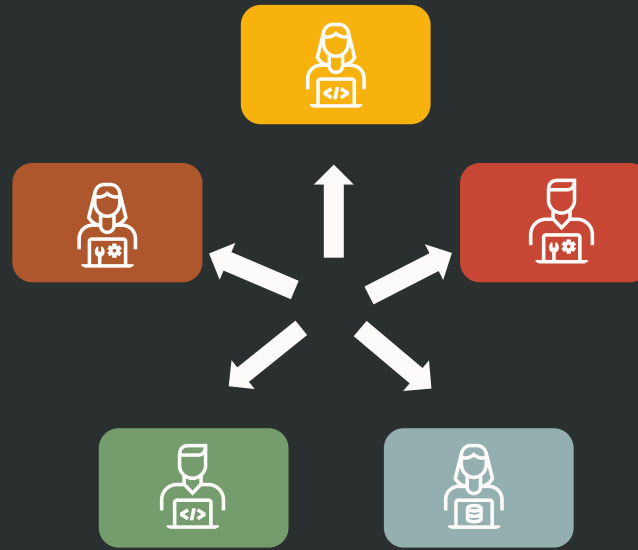


分權、去中心化的應用系統跟應用系統開發團隊

現代化的應用系統開發轉型超越技術變革

他們改變了技術團隊的**組織方式**

它們是應用程序、應用程序開發團隊和數據**去中心化**的更廣泛宏觀趨勢的一部分



集中式 vs 去中心化的資訊系統

集中式和去中心化組織各有優缺點

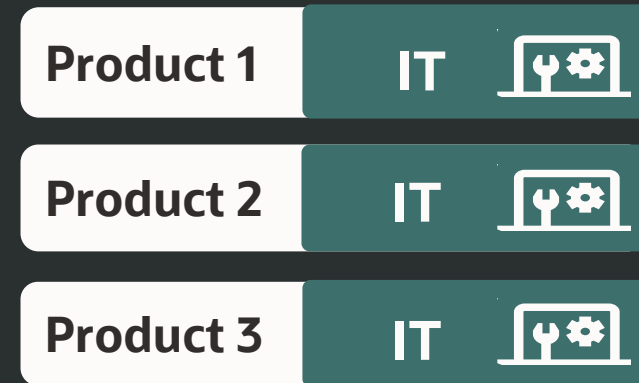
集中式 I.T. 效益

效率、一致性、風險降低



去中心化 I.T. 效益

專注、結果導向、敏捷



I.T. 組織向來偏向支持團隊和技術的集中化

用於集中化的 甲骨文數據技術

甲骨文以其領先的支持集中式架構跟組織的數據技術而聞名

整合和混合工作負載



Multitenant Database, Exadata, OLTP and analytics in one

容錯可用性



RAC, Data Guard, Flashback, Online Data Reorg, Online Patching, MAA

最高安全性



Data Safe, Advanced Security, Encryption, Roles, Privileges ...

透明的擴充能力



RAC, Exadata, Parallel SQL, Native Sharding

支持全面的數據模式



JSON, text, graph, spatial, blockchain

強大的分析能力



In-memory analytics, machine learning, data partitioning

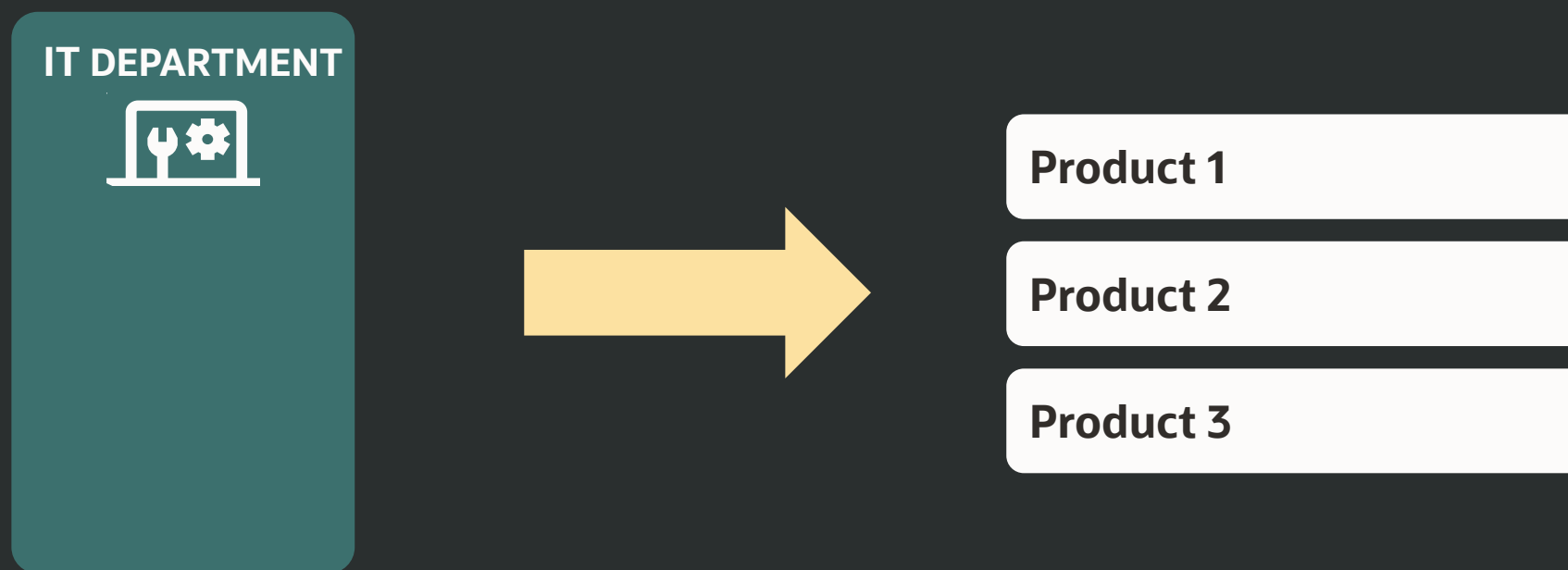
These technologies are focused on centralized architectures but are key for decentralized also

I.T. 趨勢是去中心化

最近 I.T. 已經轉向更分散的組織模式

- 目標: 更小、更敏捷、更專注的團隊，擁有端到端的解決方案

現代應用開發技術支持了這種去中心化的方向



去中心化技術組織- DevOps

應用開發簡化的技術使功能性 I.T. 團隊將被合併為 **DevOps** 並按產品分散

Development



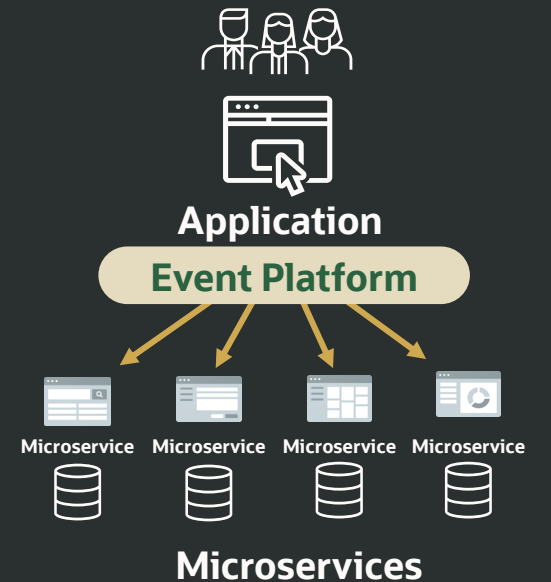
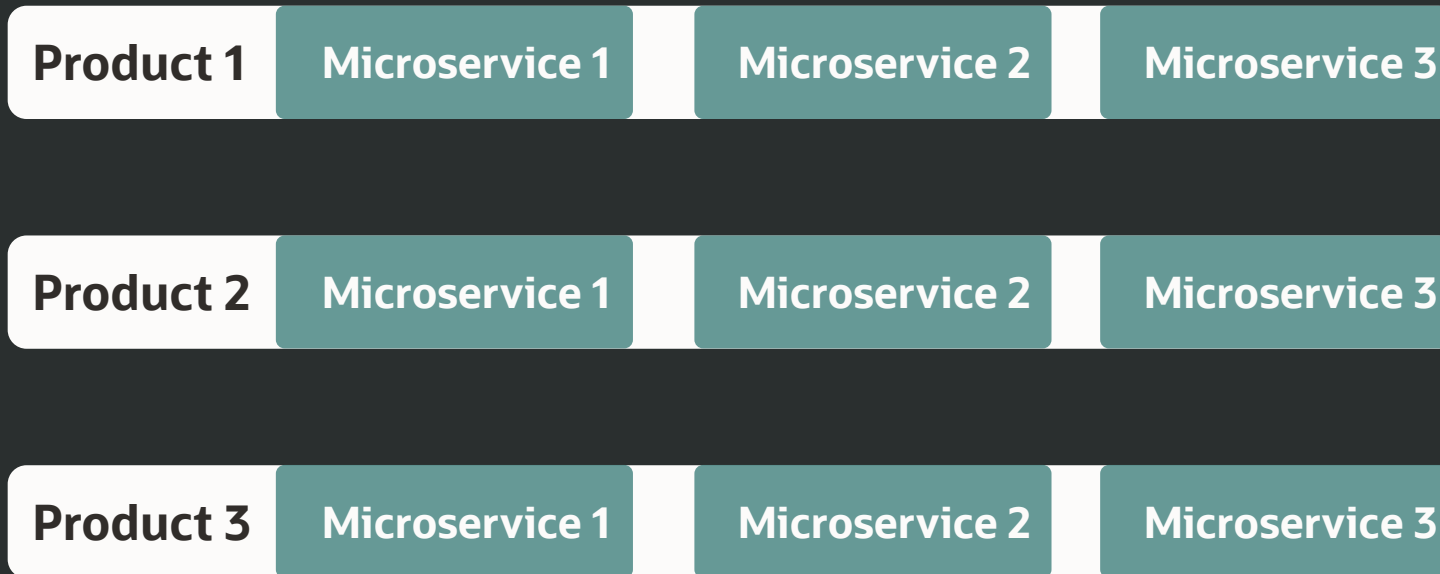
Operations



DevOPS

去中心化技術組織- 微服務

API 跟 Event Driven 應用開發技術使團隊能夠進一步分散，專注於產品特定的微服務而不是完整的應用

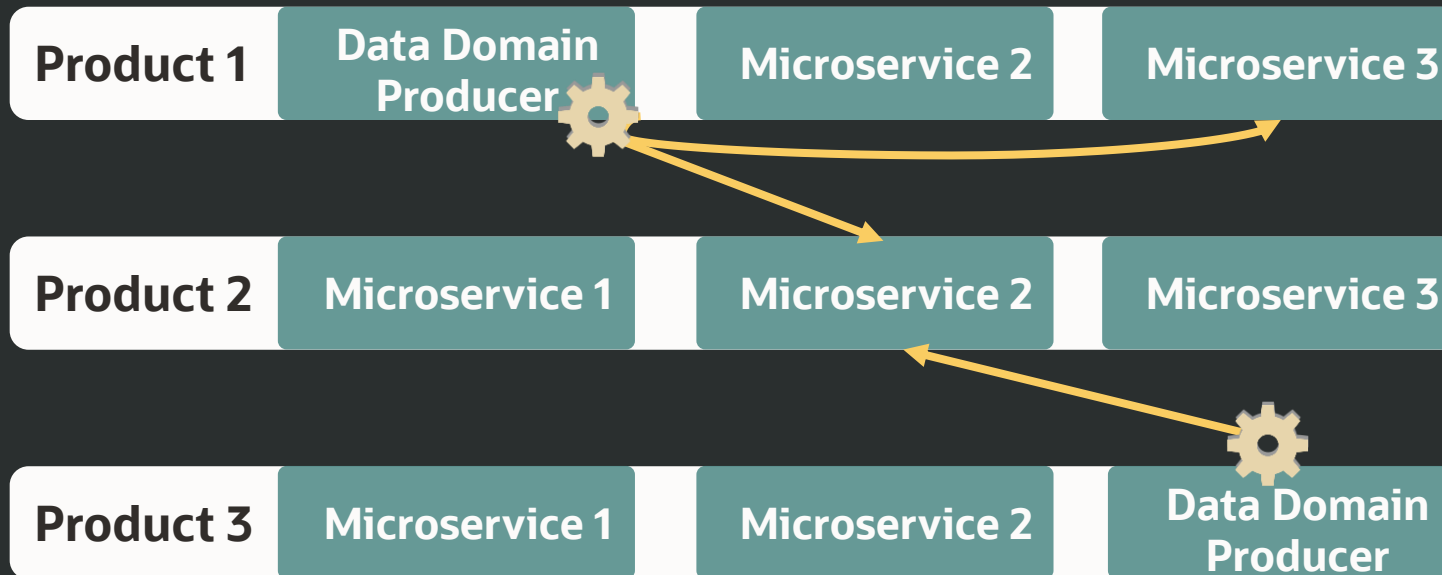


去中心化技術組織—分散式數據網格(Data Mesh)

分散式數據網格架構也促進了去中心化

在分散式數據網格中，Domain數據“作為產品”發布給消費者

格式和發佈歸**去中心化的領域專家**所有，而不是中心化的技術專家



Data Mesh



去中心化 – Data(數據)

隨著應用程序和應用程序開發團隊變得更加分散，數據也變得更加分散

- 這會使數據庫架構複雜化



IT 去中心化的權衡

分散的 IT 更加以**結果為中心**，可以**更快地交付解決方案**

- 然而,它具有需要**減輕**和**最小化**的固有的權衡



IT 去中心化會造成**浪費**

- 重複勞動和技術擴散
- 數據的碎片化和發散



IT 去中心化會增加**風險**

- 安全風險、可用性風險、品質風險

最佳的集中式和分散式

甲骨文策略:為集中式和分散式架構提供最佳數據技術



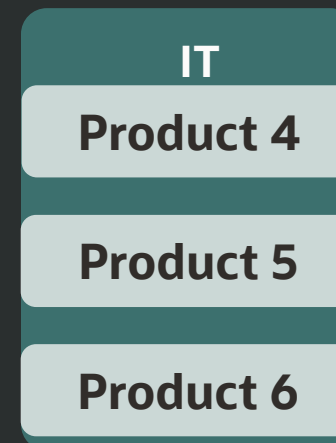
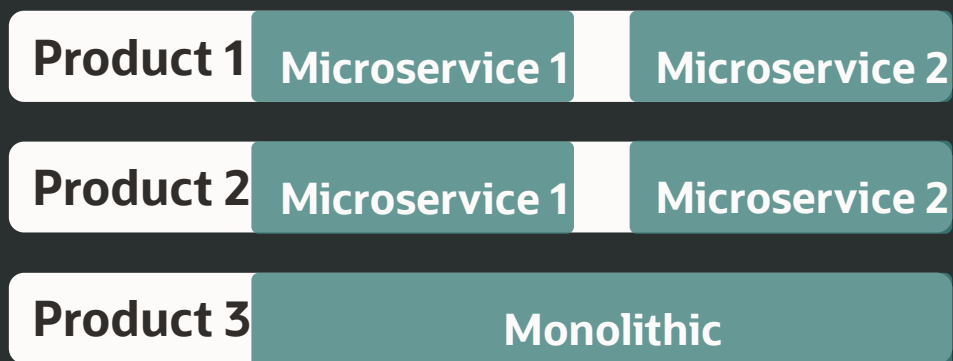
增加集中化的敏捷性



減少去中心化的浪費和風險

讓 I.T. 領導者根據業務需求選擇替代方案或兩者的混合

- 不是基於底層技術的限制



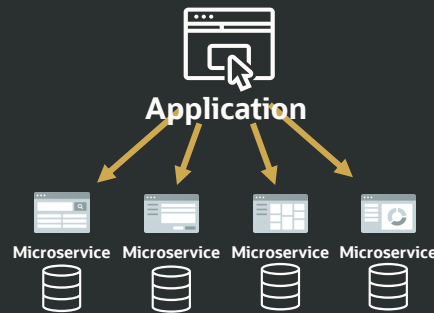
本次演講的重點- 去中心化

本演示內容介紹了 Oracle 的數據技術，該技術可促進**集中式和分散式應用程序開發**

然而，我們將**專注**於我們的**去中心化數據庫架構**，它促進了開發營運、微服務和分散式數據網格等應用開發技術，因為這些技術對**數據架構**尤其具有挑戰性



DevOPS



Microservices



Data Mesh



第一個部分

Oracle 如何簡化分散部署

Deploying Apps



As a Service
Building Blocks



Containers



Kubernetes



Cloud

Architecting Apps



Microservices



Events



Data Mesh



API Driven



Low Code

Data Model and Data Analytics



Machine
Learning



Documents



Graph
Analysis



Text
Search



Spatial
Processing



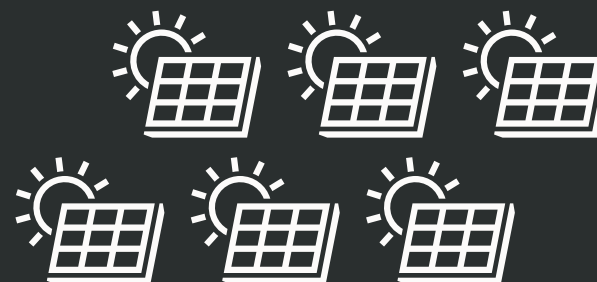
Blockchain

技術去中心化是由簡單實現的

去中心化需要**易於部署和操作**的技術



電廠的複雜性要求集中發電



太陽能電池板的簡單性使發電容易分散



數據技術即服務

同樣，技術構建區塊的交付像 **API 驅動服務** 一樣簡單易用，可實現技術去中心化

Oracle 的分散式資料庫架構使分散式應用開發從根本上更易於部署和使用



Data Tech
As a Service



In-Database
Containers



Docker for
Database



Kubernetes
for Database



Database as
Cloud Service



Autonomous
Database

In-Database Containers

容器通過實現邏輯系統而不是物理系統的快速和輕量級部署來簡化部署

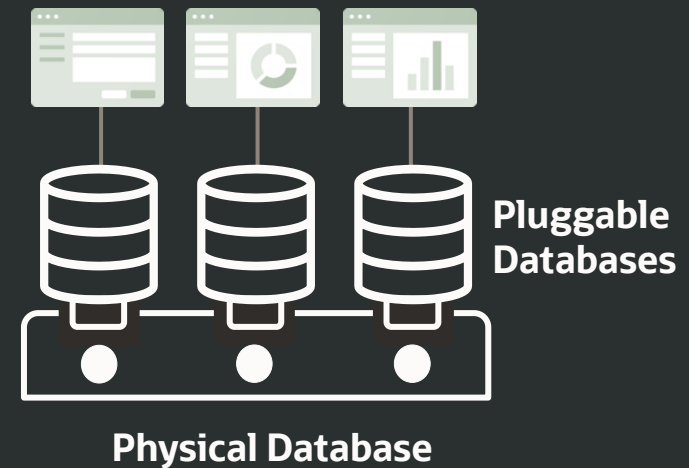
Oracle 在資料庫中原生實現了容器概念

您可以部署輕量級 **pluggable container databases**

- 而不是為每個微服務部署和管理物理數據庫，

Many Pluggable Databases 可以部署在 **single physical database** 來簡化管理

Pluggable databases 可以很容易地複製和移動 physical databases



Unlimited Pluggable Databases Now Available in Database 21c XE

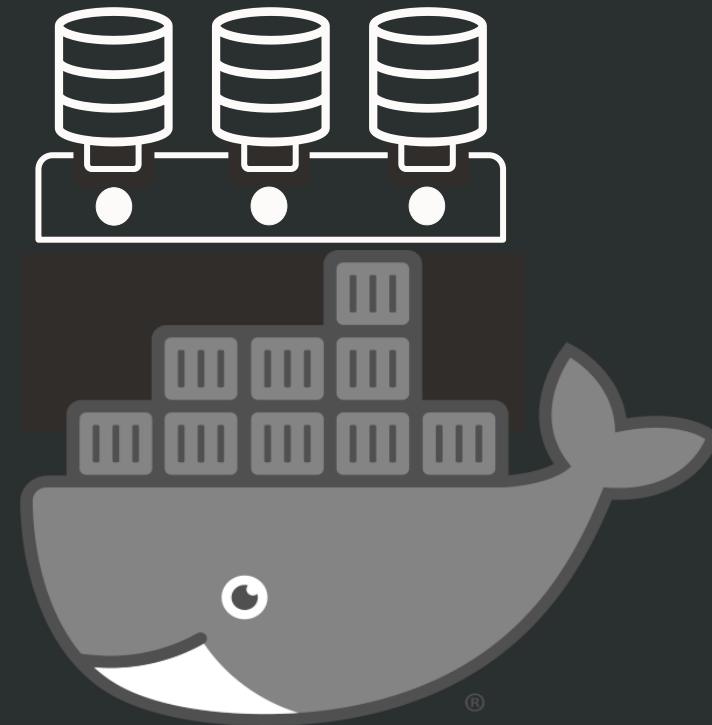
Docker Containers

你現在可以很簡單的使用 Docker Containers 來**簡化部署實際的資料庫**

甲骨文提供預先建好的 Docker Container 映像檔包括 單一 Instance, RAC, 跟 Sharding

- <https://container-registry.oracle.com>

Oracle 在 GitHub 上發布了用於建立這些映像的 Docker 文件您可以使用 Docker 文件為您的特定資料庫版本和更新檔構建映像檔



Kubernetes

Oracle 提供了使用 Kubernetes **簡化資料庫部署跟排程**的能力

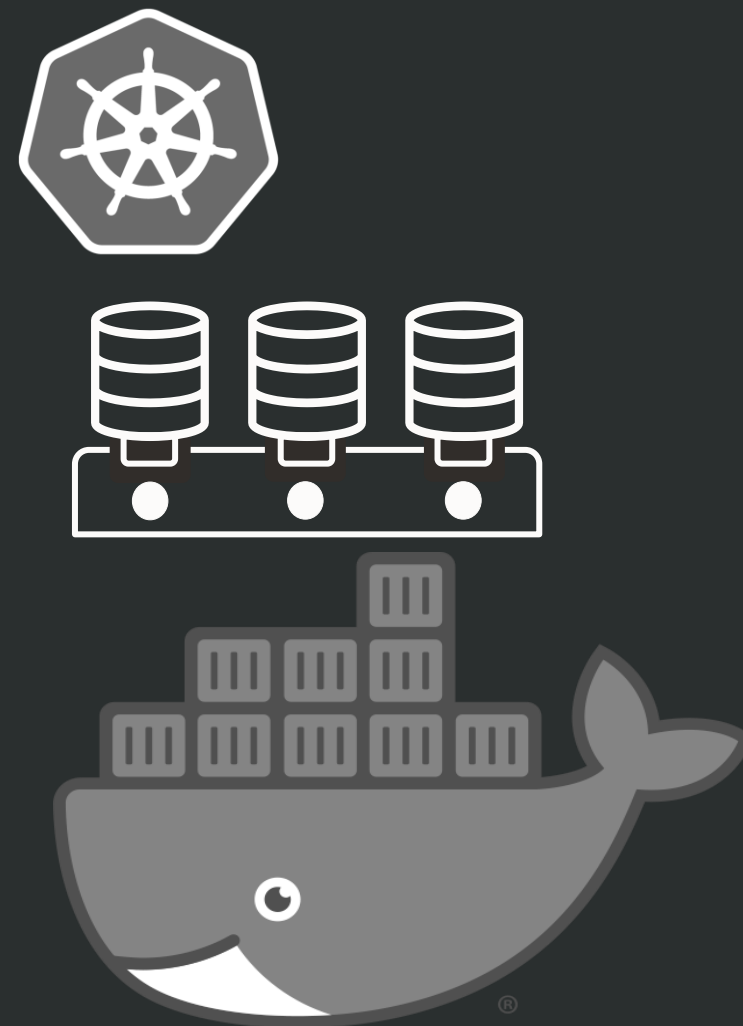
Oracle Kubernetes Operators 排程可插拔容器數據庫:

- Create, delete, clone, scale, patch, etc.

Oracle 在 GitHub 上提供了描述數據庫結構的 Helm Charts

- 啟用複雜的配置，如Pluggable Database

Kubernetes 工具和儀表板可以監控和跟踪完整的應用程序



Exadata Database Service

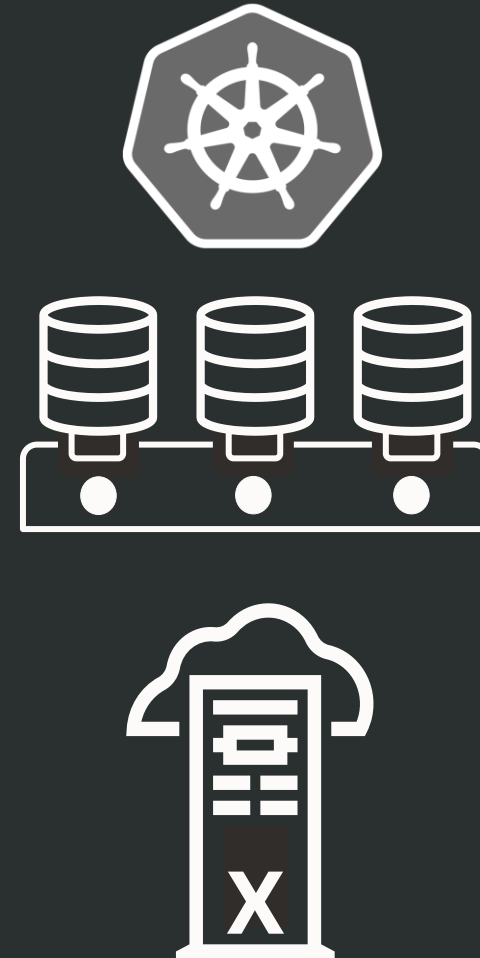
您可以使用 Exadata 資料庫服務實施完整的資料庫即服務模型

- 在世界上最好的資料庫平台上運行– Exadata

從硬體到自動化和優化所有資料庫基礎架構

提供用戶控制的數據庫自動化、彈性擴展和按使用付費

可以部署在公有雲, 或 Cloud@Customer 在你的資料中心



Autonomous Database

為實現極致簡單，使用 Oracle 自主資料庫使您的資料庫實現自主駕駛、自主保護和自主修復

Oracle 接管通用資料庫管理，允許 DevOps 專注於應用程序特定的數據模型、安全性、性能等。

可以部署在公有雲, 或 Cloud@Customer 在你的資料中心

JSON 格式的數據服務可以使用 **Autonomous JSON Database**





In-Database
Containers



Docker for
Database



Kubernetes
for Database



Database as
Cloud
Service



Autonomous
Database

大方向

Oracle 的去中心化資料庫架構
使分散式應用程序易於部署和
操作

第二部分

Oracle 如何讓構建去中心化應用程序變得簡單

Deploying Apps



As a Service
Building Blocks



Containers



Kubernetes



Cloud

Architecting Apps



Microservices



Events



Data Mesh



API Driven



Low Code

Data Model and Data Analytics



Machine
Learning



Documents



Graph
Analysis



Text
Search



Spatial
Processing



Blockchain



簡化去中心化應用程序開發

我們獨特的分散式資料庫架構讓使用分散式應用程序開發技術變得簡單



去中心化應用程序是如何實現的

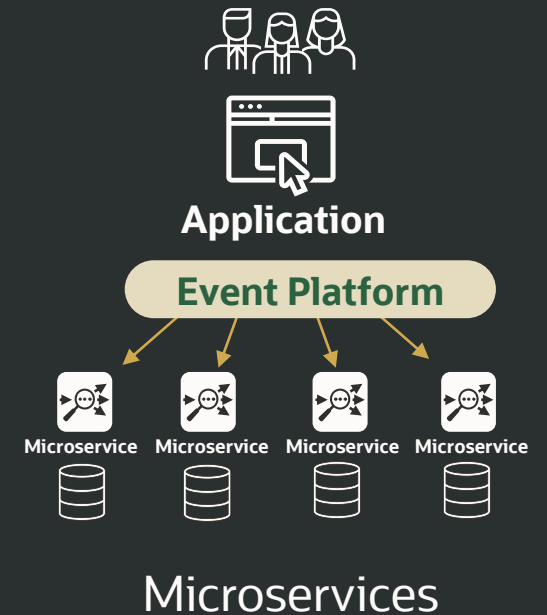
去中心化 App 使用**非同步事件**來啟動應用和跨微服務動作

- 舉例, 發佈一個 “*process this new order*” 事件可以造成多個為服務或應用程式去執行動作

事件是使用 如Kafka 等事件平台發布的

用戶使用 Kafka 的一些痛點越來越明顯

- Oracle 提供了從架構上解決這些痛點的獨特解決方案



去中心化應用程序的問題

事件處理程序通常由開發人員寫死- **coded events**

- 事件處理程序實現操作、更改資料庫狀態並產生其他事件

確保事件**處理程序保持跨服務的數據一致性**很困難，因為它們必須處理諸如：

- 程序和網絡故障、立即變更、從備份恢復和災難故障轉移
- 隨著時間的推移，任何資料差異都會導致額外的差異



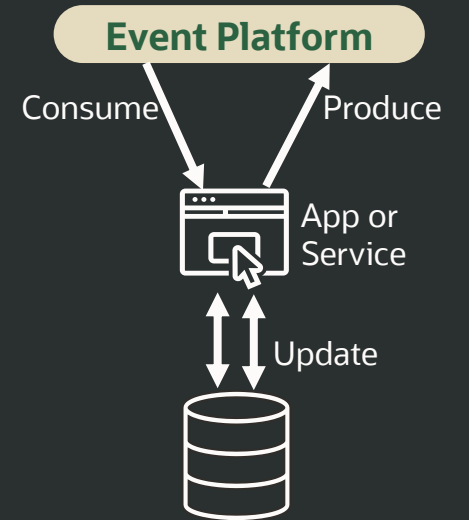
事件系統到資料庫一致性

即使我們忽略跨服務**一致性**，在有狀態事件平台和**單個**有狀態數據庫之間保持一致性也很困難

大多數事件平台要求開發人員將**一致性和故障恢復寫死**到每個事件處理程序中

對於簡單的發布或使用情況，開發人員必須手動跟踪事件使用的狀態並將其存儲在資料庫中

對於涉及多個事件和資料庫操作的更複雜的情況，保持一致性需要大量且容易出錯的工作



ACID 和交易的好處

原子**交易**的好處**很好理解**但經常被**遺忘**

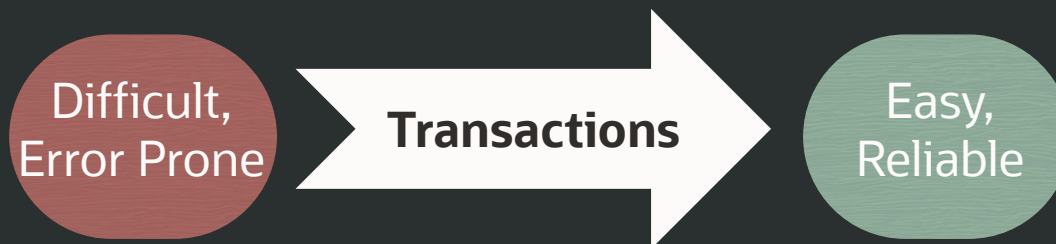
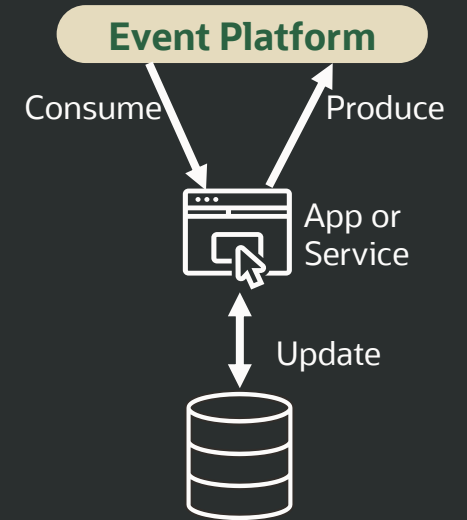
NoSQL 資料庫最初聲稱交易是不必要且有害的

- 然而，NoSQL 資料庫的用戶通過經驗了解到，在應用程式中編碼數據一致性容易出錯且耗時

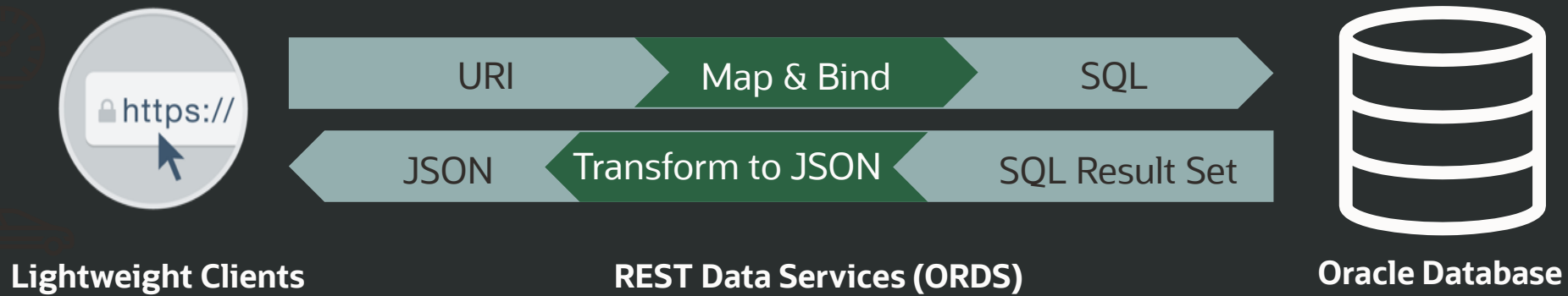
較新版本的 Kafka 支持交易—**但僅限於活動平台內**

Kafka 適用於不支持交易的同質性服務之間的事件

對於 Oracle 資料庫內或跨 Oracle 資料庫的同類事件，使用**資料庫原生交易事件**更簡單、更可靠



API 導向架構- 資料庫原生 REST



去中心化的應用系統常使用 REST APIs 作為溝通的方式

Oracle Rest Data Services 自動產生一個 REST API 在任何一個 SQL statement 或 stored procedure 之上

- 免費的功能支援 Open API, OAuth, Pagination, Filtering, LOBs, connection pooling
- **統一 App APIs 跟 Data APIs** 使去中心化應用程序更易於開發
- 讓你可以建構 **Data-as-a-Service**

Low-Code - Database Native APEX



Oracle's free APEX low-code IDE 顯著簡化 App Dev 實現 App Dev 去中心化
With APEX, 企業應用程序可以生成而不是只是編碼

APEX 統一應用層和資料層

- 通過消除編碼和管理中間層的麻煩來提高敏捷性
- 不再需要應用層狀態或連接管理，或將應用類型映射到資料庫類型

Low-code 簡單性使精通技術的領域專家能夠創建自己的應用程序或分散式數據網格“數據產品”，進一步分散技術

APEX Apps 使用 Rest Data Services 與其他去中心化應用程序溝通連結

Over 2 Million APEX Apps Exist Today
3K New APEX Apps Developed Every Day

Geo Distribution – Database Native Sharding

Oracle's 原生 Sharding 使應用程序可以輕鬆提供

- 地理數據分佈或大規模

將集中式數據庫分解為較小的分散式數據庫群

- 分片式可以位於不同的國家以滿足數據主權

分片式完全隔離-提供數據和用戶的線性可擴展性

- Routes SQL based on shard key, or runs cross shard SQL
- Online addition and reorganization of shards

分片式的所有好處與成熟的所有好處SQL Database

Centralized DB



Many Decentralized Shards

Used by Many of the Largest Internet Providers and Telecoms in the World



大願景

去中心化的應用程序非常簡單
和可靠

使用 Oracle 分散式資料庫架構



第三部分

Oracle 如何在分散式應用程序中輕鬆使用 現代資料模型和工作負載

Deploying Apps



As a Service
Building Blocks



Containers



Kubernetes



Cloud

Architecting Apps



Microservices



Events



Data Mesh



API Driven



Low Code

Data Model and Data Analytics



Machine
Learning



Documents



Graph
Analysis



Text
Search



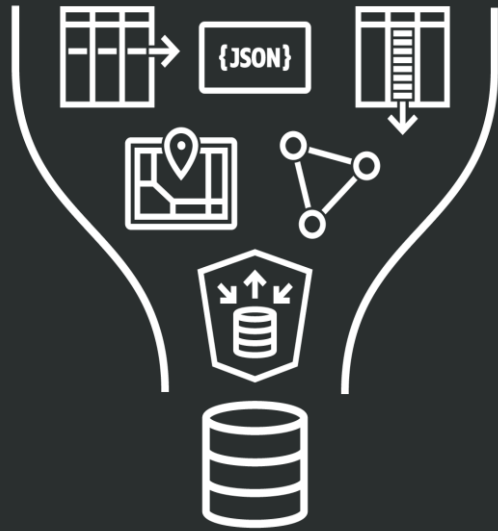
Spatial
Processing



Blockchain

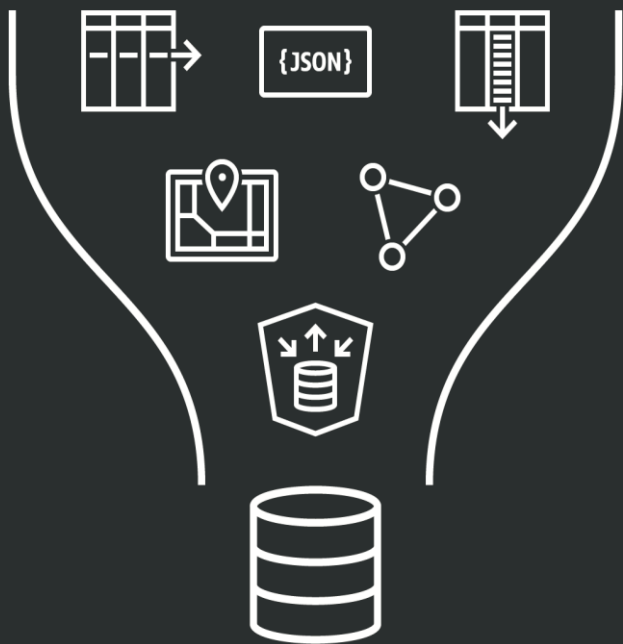
使用現代數據模型和工作負載

Oracle 是一種融合數據庫，可為所有數據模型和工作負載提供同類最佳的支持



- Relational Data
- JSON and XML Documents
- Machine Learning
- Graph analysis
- Text search
- Spatial query
- Blockchain

消除單一用途數據存儲的複雜性



所有數據類型和工作負載都是內建的

Just call SQL 來使用

與多個單一用途數據庫相比，單個融合數據庫在部署、保護、擴展和操作方面的工作量和風險要小得多

- 還避免了必須在數據庫之間移動和協調數據

一樣的SQL, REST APIs 跟和交易維運跨所有數據類型和工作負載運行

它大大地簡化了 App Dev 和 Ops，使分散的小型團隊能夠使用現代數據類型和工作負載



大方向

通過消除單一用途數據存儲引入的
複雜性，

融合式資料庫大大地簡化了去中心
化的應用程序開發

Oracle 如何簡化現代 分散式應用程序的示範



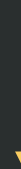
線上商店的示範

讓我們看一個線上商店的例子

傳統上，應用程式的所有方面都將位於同一地點/位置

- 在一個巨大的應用中
- 針對一個巨大資料庫運行

Online Store



Operational



線上商店 - 去中心化微服務架構

線上商店的微服務架構看起來非常不同

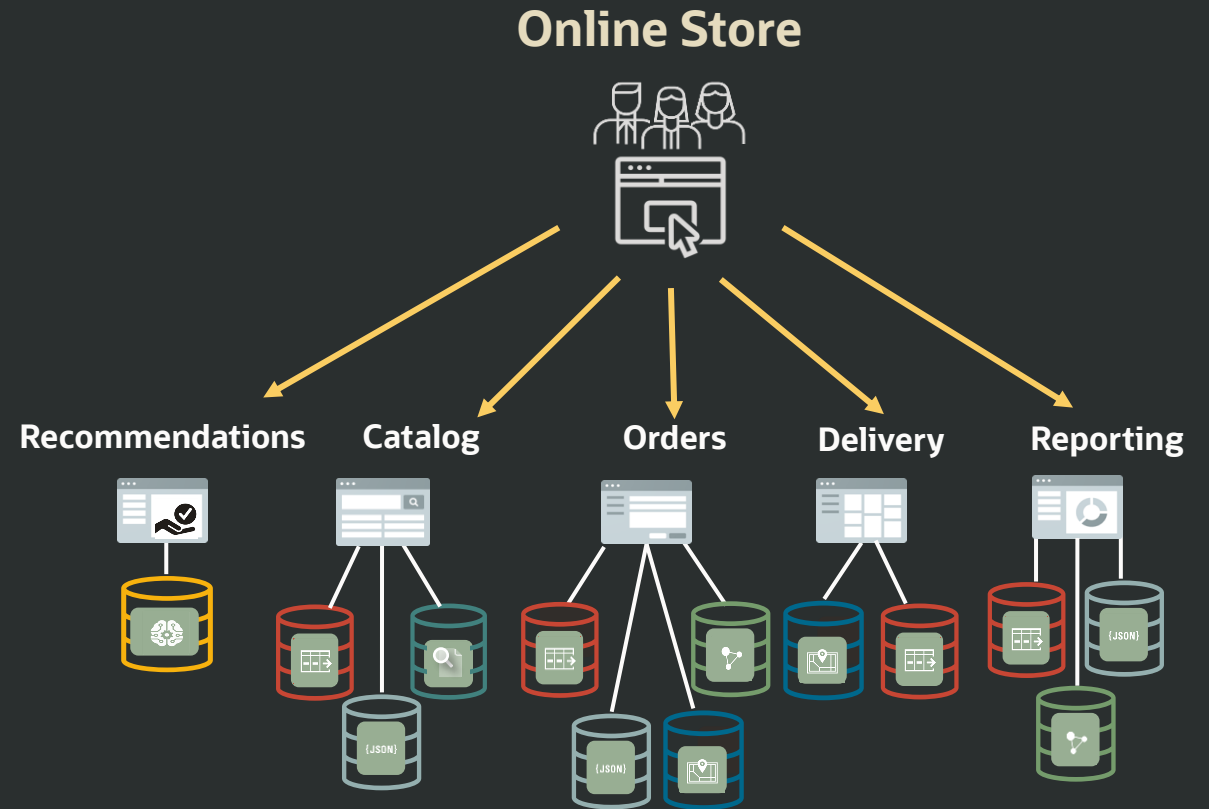
應用架構和數據庫架構是去中心化的

應用系統拆成五種微服務

- 推薦系統
- 型錄
- 訂單
- 送貨
- 分析

每個微服務使用多個單一用途的資料庫-

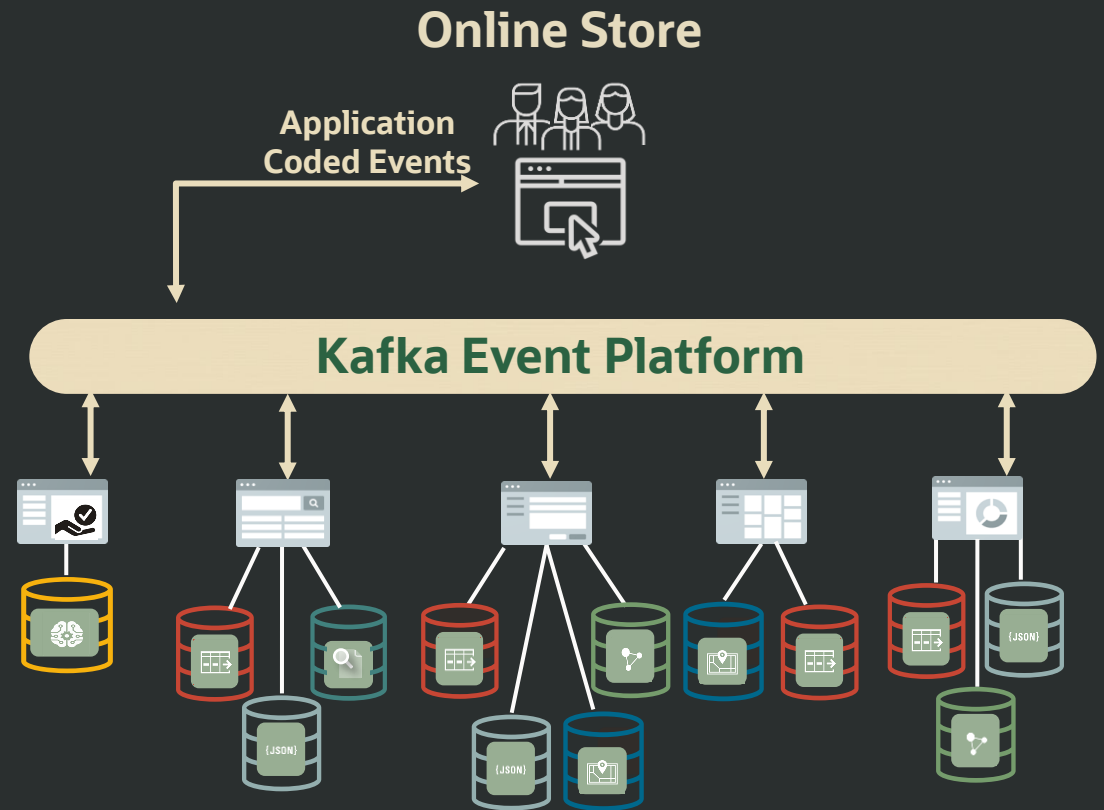
每一個用於每個數據模型或工作負載



Coded Events for App Actions

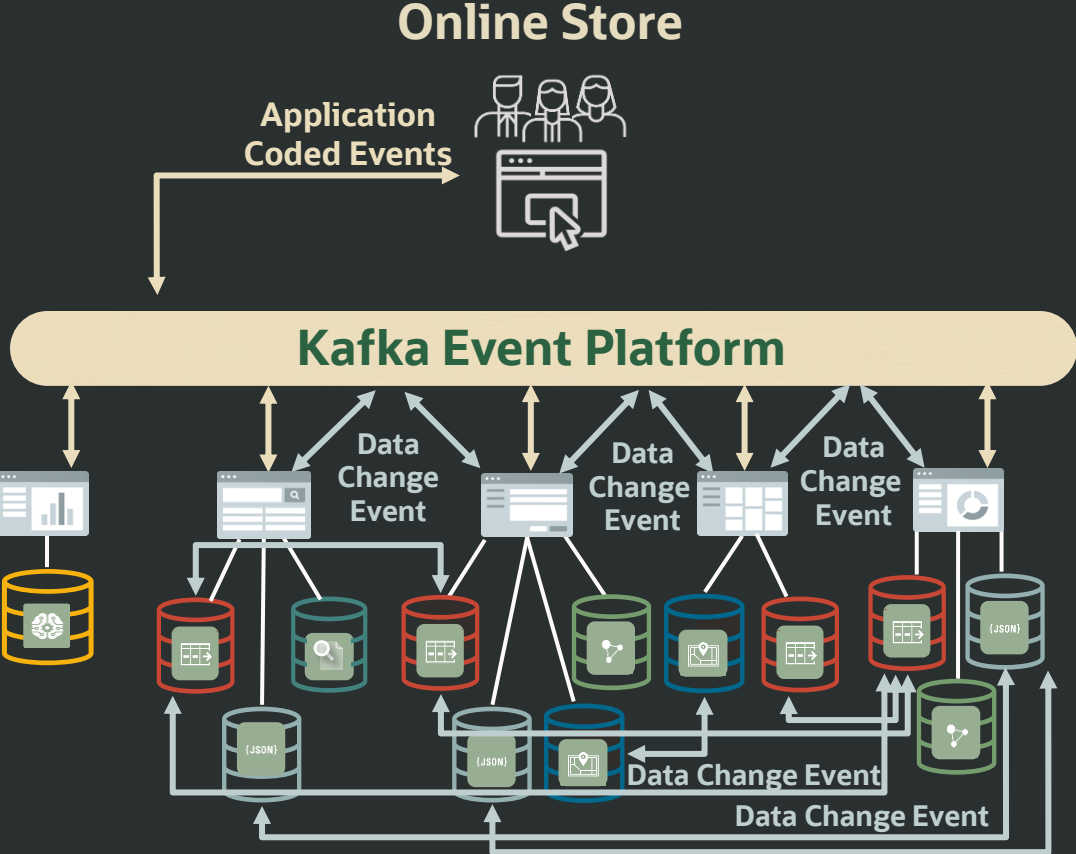
非同步編碼事件用於實現跨微服務的應用程序邏輯或操作

使用 Kafka 等事件平台在應用程序級別傳遞編碼事件



Coded Events for Data Changes

編碼事件還用於在微服務之間傳遞數據更改



Oracle 如何簡化這種微服務架構?

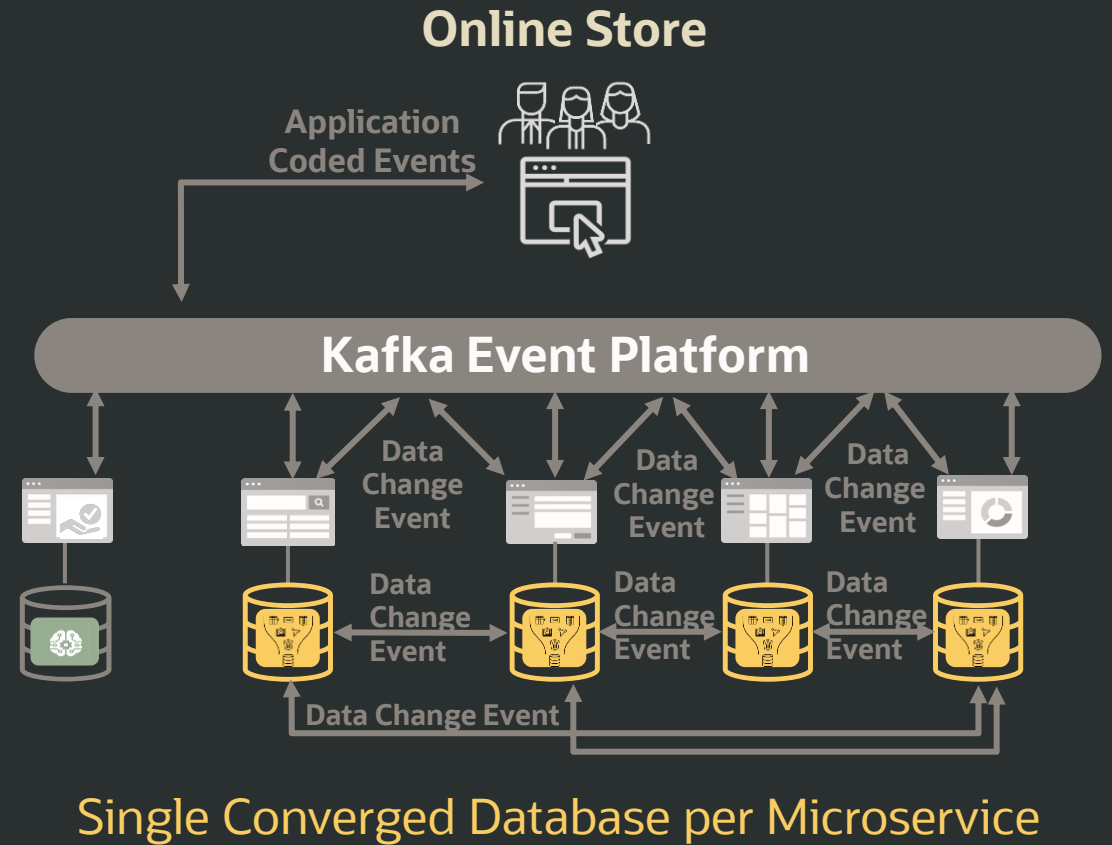


在一個微服務中聚合多個資料庫

步驟一

用單個融合資料庫替換微服務中的多個單一用途資料庫

- 使用單一資料存儲的微服務的優化、保護、擴展和故障排除要容易得多



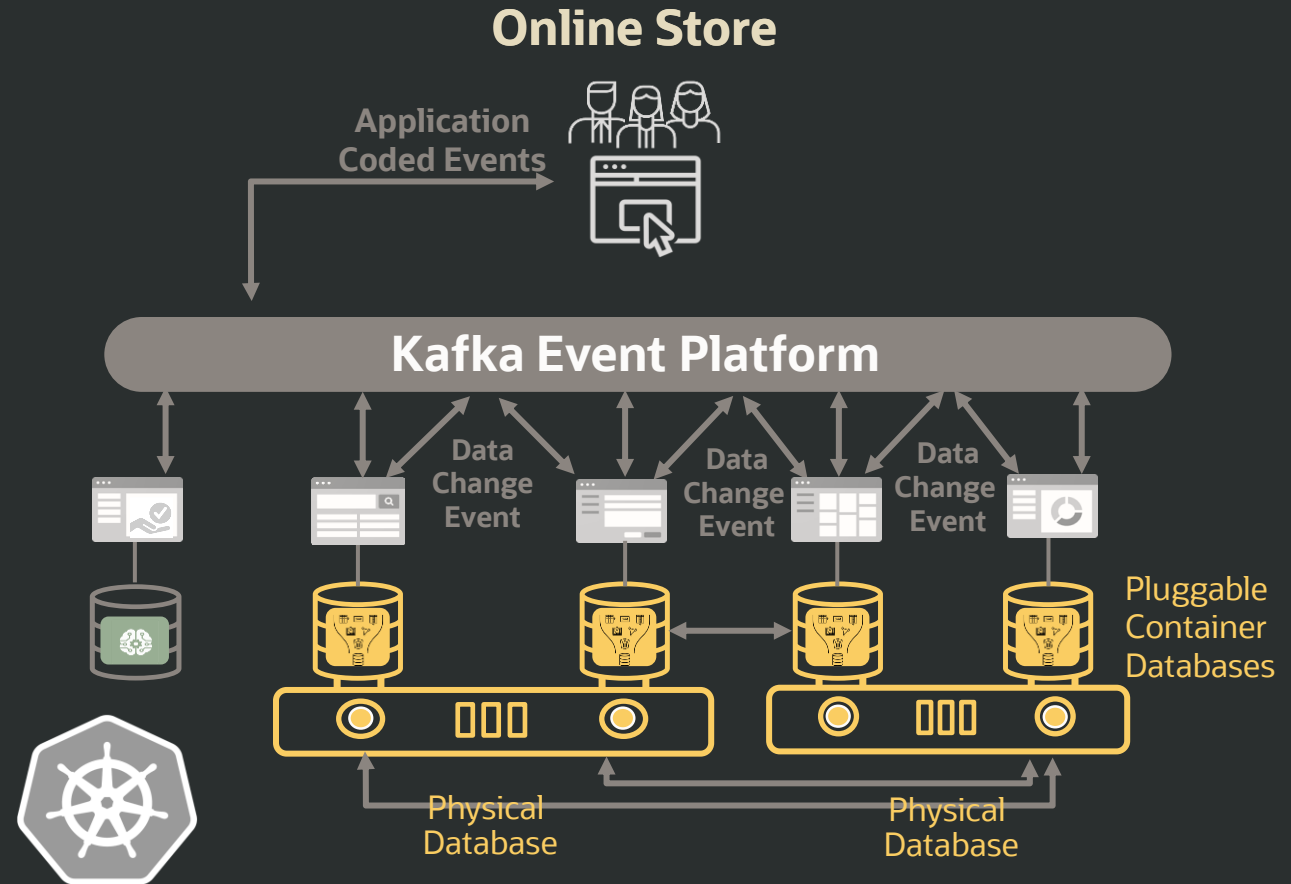
容器化微服務數據庫

步驟二

使用可插拔資料庫將融合資料庫容器化為更少的物理資料庫

- 將多個資料庫合而為一

使用 **Kubernetes Operators** 編排可插拔資料庫操作



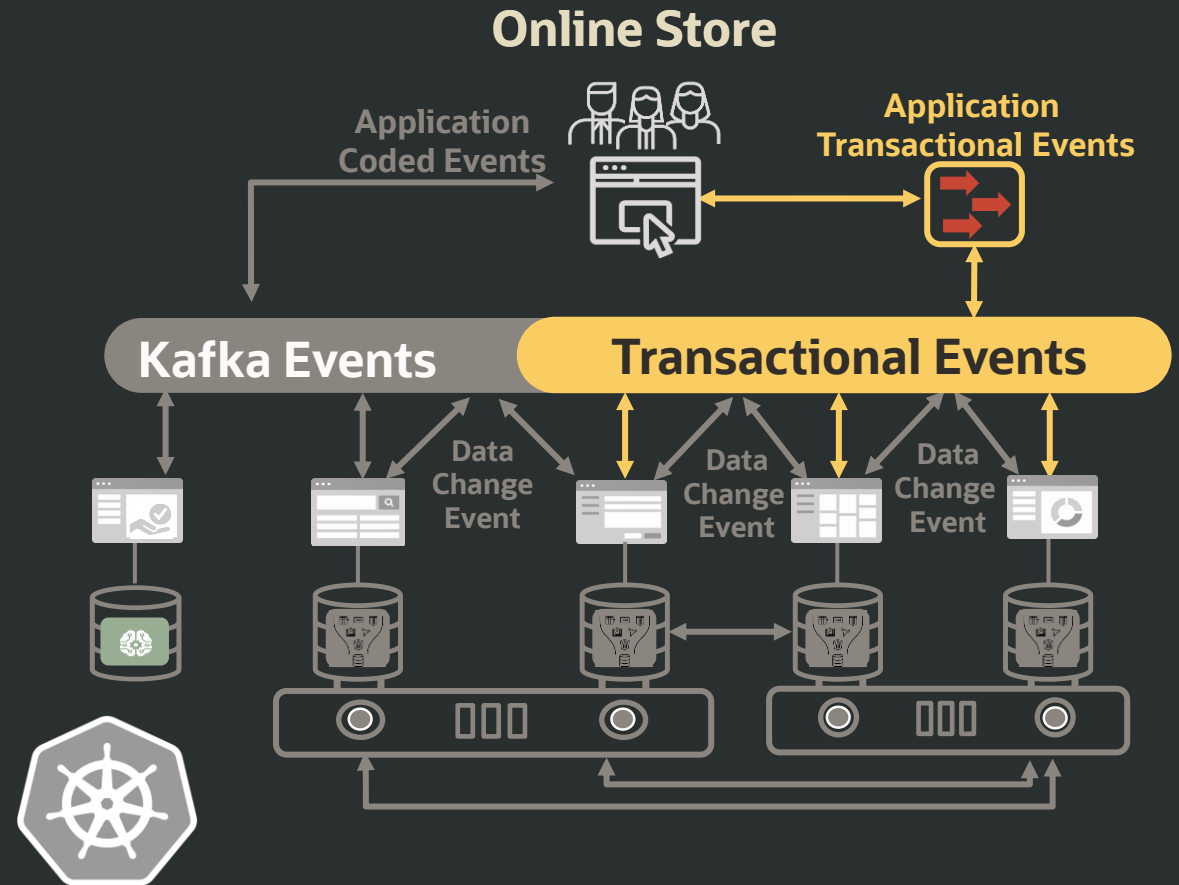
使用 Transactional Event Queues

步驟三

Implement Coded Events with **Transactional Event Queues** where possible

- Events and database changes execute **in one atomic transaction**

繼續使用 Kafka 處理異構事件



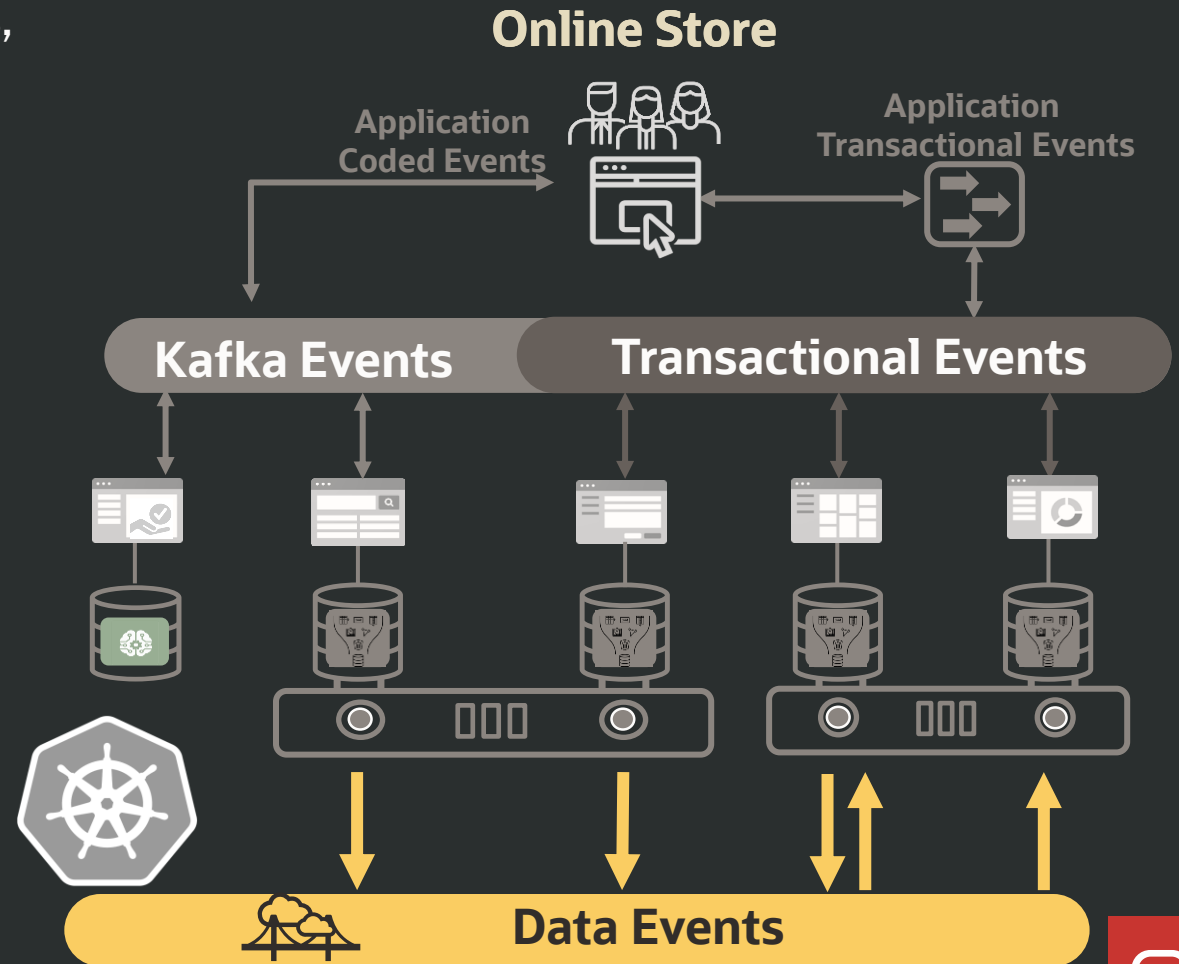
使用 GoldenGate

步驟四

使用 Oracle GoldenGate to **transactionally** publish and consume **data events** between Microservices, Apps, or Clouds

- 無需手動編碼捕獲和應用狀態更改

一個應用程序可以同時擁有編碼事件和資料事件
GoldenGate 可以發布資料事件到Kafka

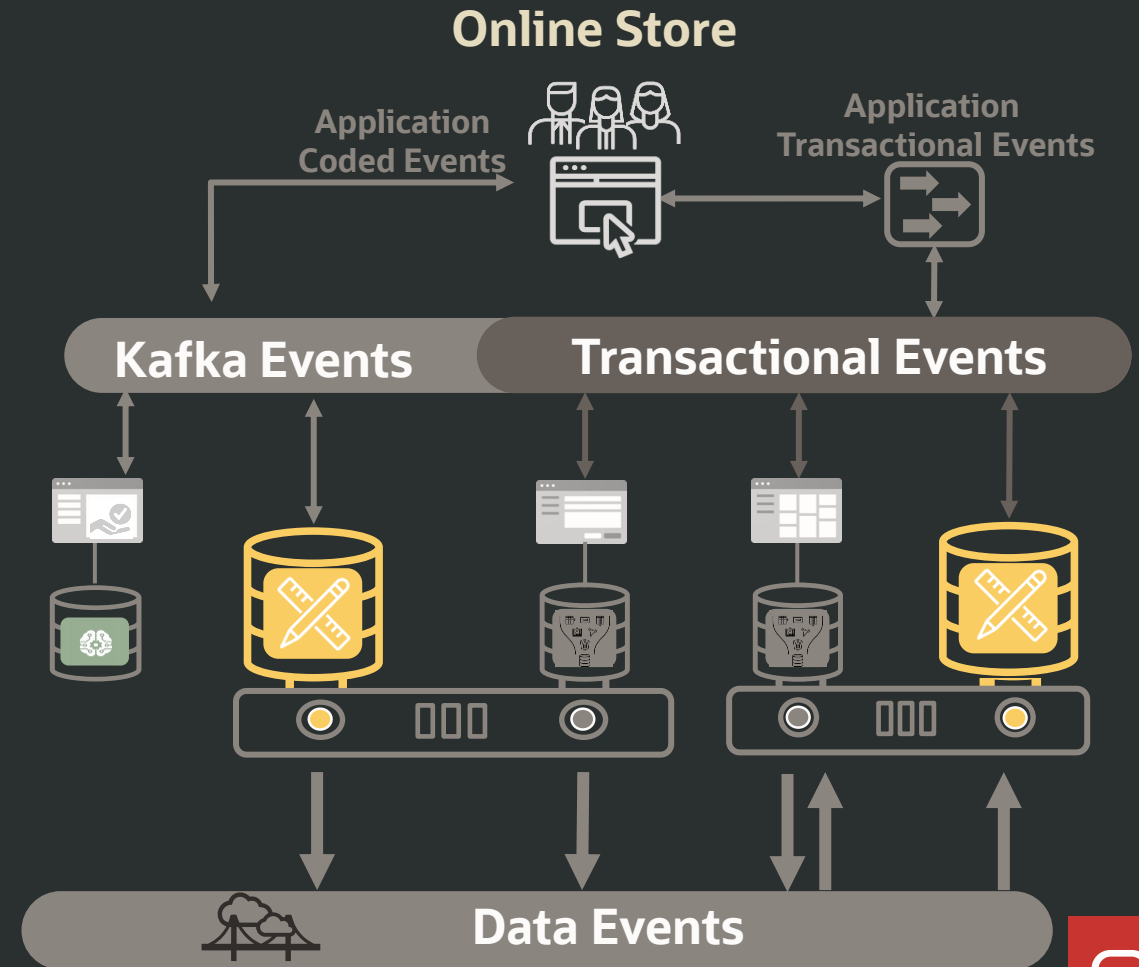


使用 Low-Code APEX 產生應用系統

步驟五

使用Low-Code APEX 生成大多數企業應用程序或微服務，而不是手動寫死它們

也消除了單獨的應用層和數據層的複雜性

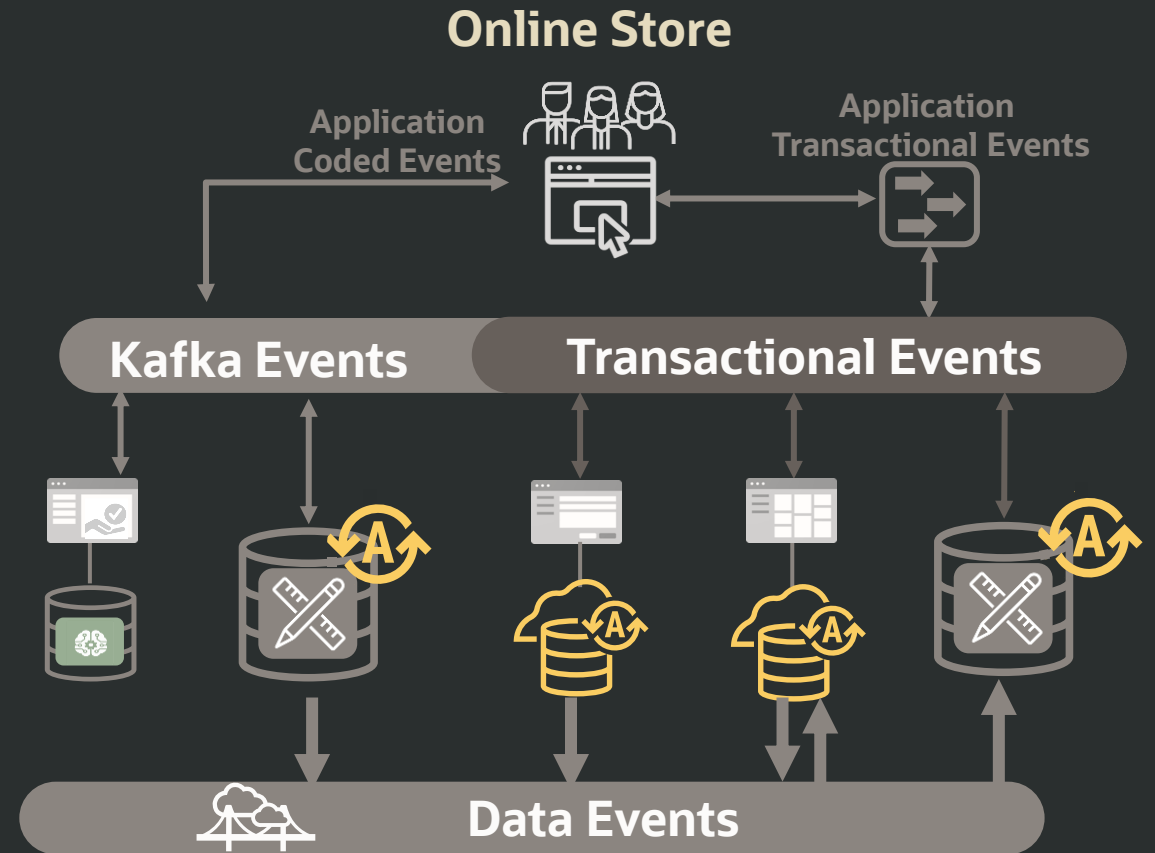


使用雲技術部署和管理資料庫 最後步驟

使用 Oracle Exadata Cloud services 自動化和優化整個
數據庫基礎架構

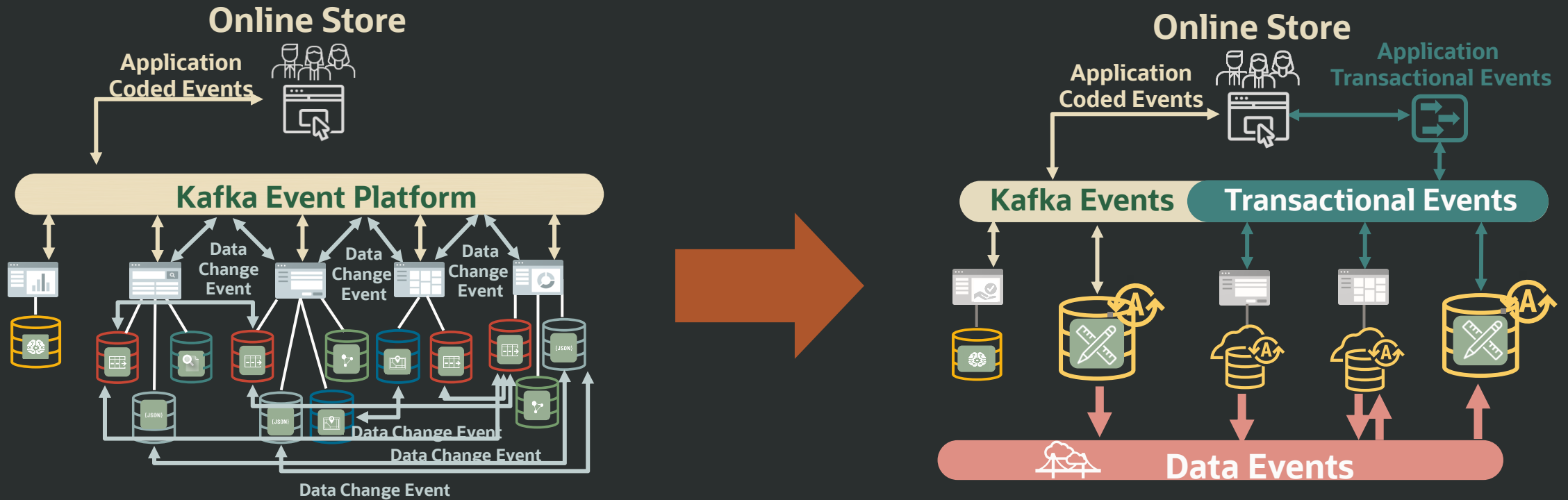
- 在公有雲或自家機房

使用 **Autonomous Database** 讓你的資料庫 自主管理,
自主修復, 自主保護



簡化後的線上商店

如您所見，重新思考數據庫架構以擁抱現代 App Dev 會導致應用程序變得更加簡單和強大



More Info: <https://developer.oracle.com/microservices/>



重要訊息

現代範式簡化和加速應用程序開發

- 促進應用程序和開發團隊的去中心化
- 然而,分散的應用程序開發使資料庫架構複雜化

Oracle's 策略 –重新思考資料庫架構:

- 同時促進集中式和分散式應用程序開發
- 為既有集中化資料庫帶來敏捷性來支持去中心化的應用
- 降低去中心化的複雜性、浪費和風險

Simplify App Deployments



In-Database Containers



Docker for Database



Kubernetes for Database



Database as Cloud Service



Autonomous Database

Simplify App Architecture



Transactional Event



Data Event



Native REST



Database Native Low Code



Sharding

Simplify Data Models and Data Analytics



Converged Database



ORACLE

OSR & Free Training



Oracle Support Rewards 專案



在 Database 維護期間內的附加價值



可使用 Oracle Cloud Infrastructure (OCI) 累積 Support Rewards



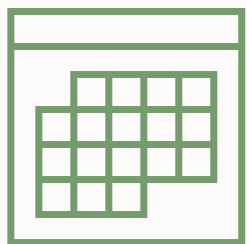
使用累積的 **Support Rewards** 抵扣 Database 維護費用

依據 Oracle 雲 (OCI) 實際使用量之

25 %

回饋金額用於抵扣 Database 維護費用

Oracle 雲 (OCI) 實際使用量越多 – 回饋越多



Oracle Support Rewards

在每個月月底基於 OCI 雲服務一整個月消費金額賺取

範例

Database 維護費用

原本需支付的維護費用

\$450

OCI 一整個月的消費金額

\$1000

Oracle Support Reward 產生的回饋金額

\$250

新的維護費用

\$200



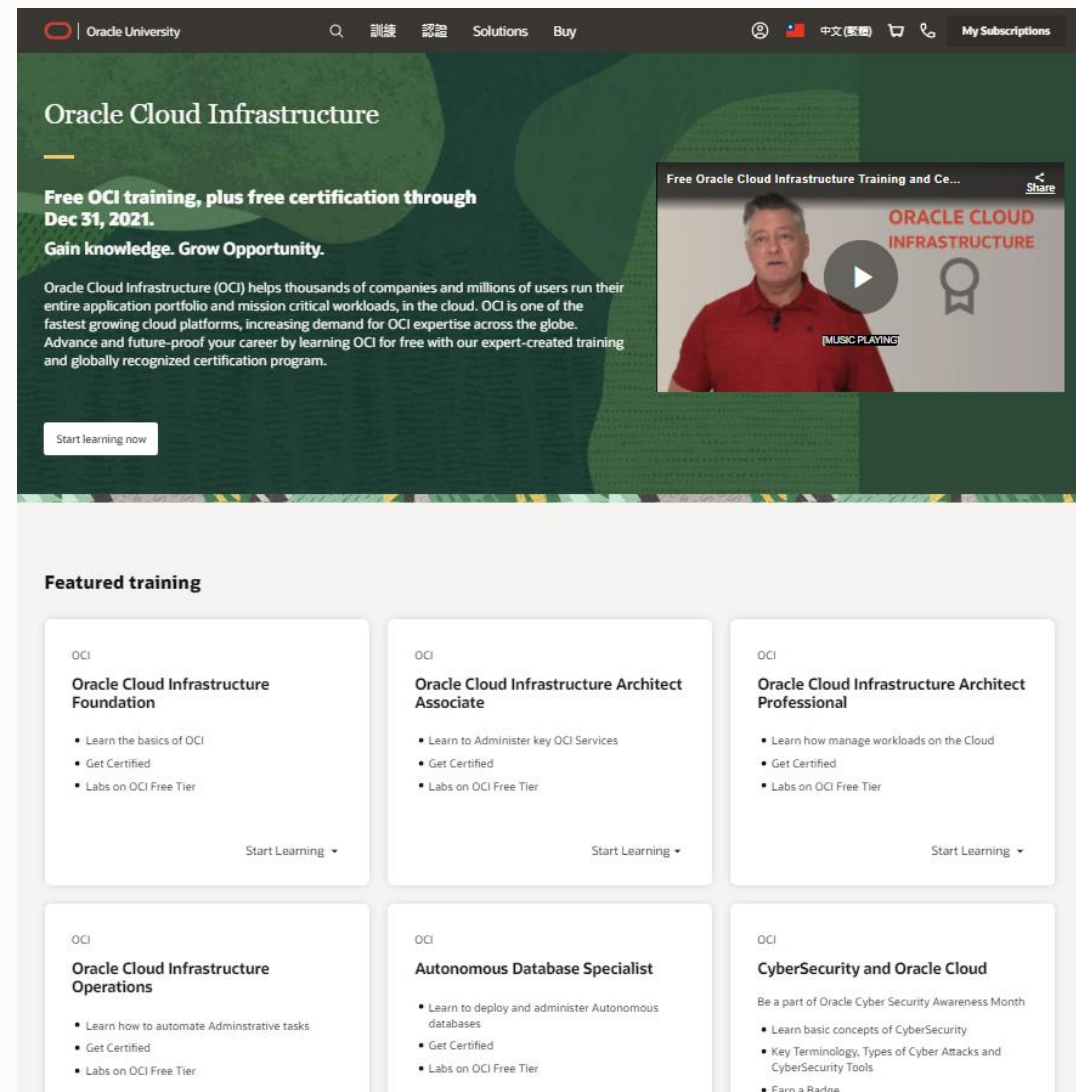
藉由使用 Oracle 雲 (OCI)，客戶有機會可以將 Database 維護費用降至 0*

* 不適用於抵扣 Database 維護費用所需繳納的稅金



免費 Oracle Cloud Infrastructure 教育訓練課程及認證考試

- 幫助增加客戶對 OCI 服務的了解和後續使用。
- 客戶可以免費參加我們針對所有學習階段和角色的 OCI 雲端課程培訓內容
- 包括 培訓課程及認證考試。
 - OCI 教育訓練課程
 - 資料管理訓練課程
 - **OCI 免費認證考試 從 2021/9/8 至 2021/12/31.**



The screenshot shows the Oracle University website with a dark green header. The main content area features a large banner for "Free OCI training, plus free certification through Dec 31, 2021." Below the banner, there is a "Start learning now" button. A video player is embedded on the right side of the banner, showing a man in a red shirt speaking. Below the banner, there is a "Featured training" section with six cards, each representing a different OCI training course. Each card includes the course title, a list of bullet points, and a "Start Learning" button.

Featured training

- OCI Oracle Cloud Infrastructure Foundation**
 - Learn the basics of OCI
 - Get Certified
 - Labs on OCI Free Tier
- OCI Oracle Cloud Infrastructure Architect Associate**
 - Learn to Administer key OCI Services
 - Get Certified
 - Labs on OCI Free Tier
- OCI Oracle Cloud Infrastructure Architect Professional**
 - Learn how manage workloads on the Cloud
 - Get Certified
 - Labs on OCI Free Tier
- OCI Oracle Cloud Infrastructure Operations**
 - Learn how to automate Administrative tasks
 - Get Certified
 - Labs on OCI Free Tier
- OCI Autonomous Database Specialist**
 - Learn to deploy and administer Autonomous databases
 - Get Certified
 - Labs on OCI Free Tier
- OCI CyberSecurity and Oracle Cloud**
 - Be a part of Oracle Cyber Security Awareness Month
 - Learn basic concepts of CyberSecurity
 - Key Terminology, Types of Cyber Attacks and CyberSecurity Tools
 - Earn a Badge



Thank You



ORACLE