

# OCI 与云原生应用程序相关的容器服务介绍

公益讲座11：00准时开始，请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布，已入群客户请勿重复入群！



20-20

数据库和云讲座群



甲骨文云技术公众号

ORACLE

# OCI 与云原生应用程序相关的容器服务介绍

Container Instances、Managed Kubernetes、Oracle Functions

---

**Xin Zhang**  
SE Hub, JAPAC

Jan. 2023



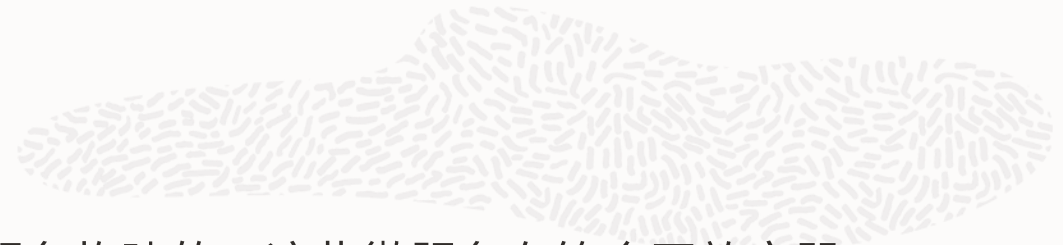
# Agenda



- 什么是云原生应用程序
- 甲骨文云上的多种容器服务
  - OCI 容器实例(Container Instances)
  - Oracle Container Engine for Kubernetes (OKE)
    - OKE 管控节点(Managed Nodes)
    - OKE 虚拟节点(Virtual Nodes)
  - 无服务器 Oracle Functions
- 容器服务的选择



# 云原生应用程序

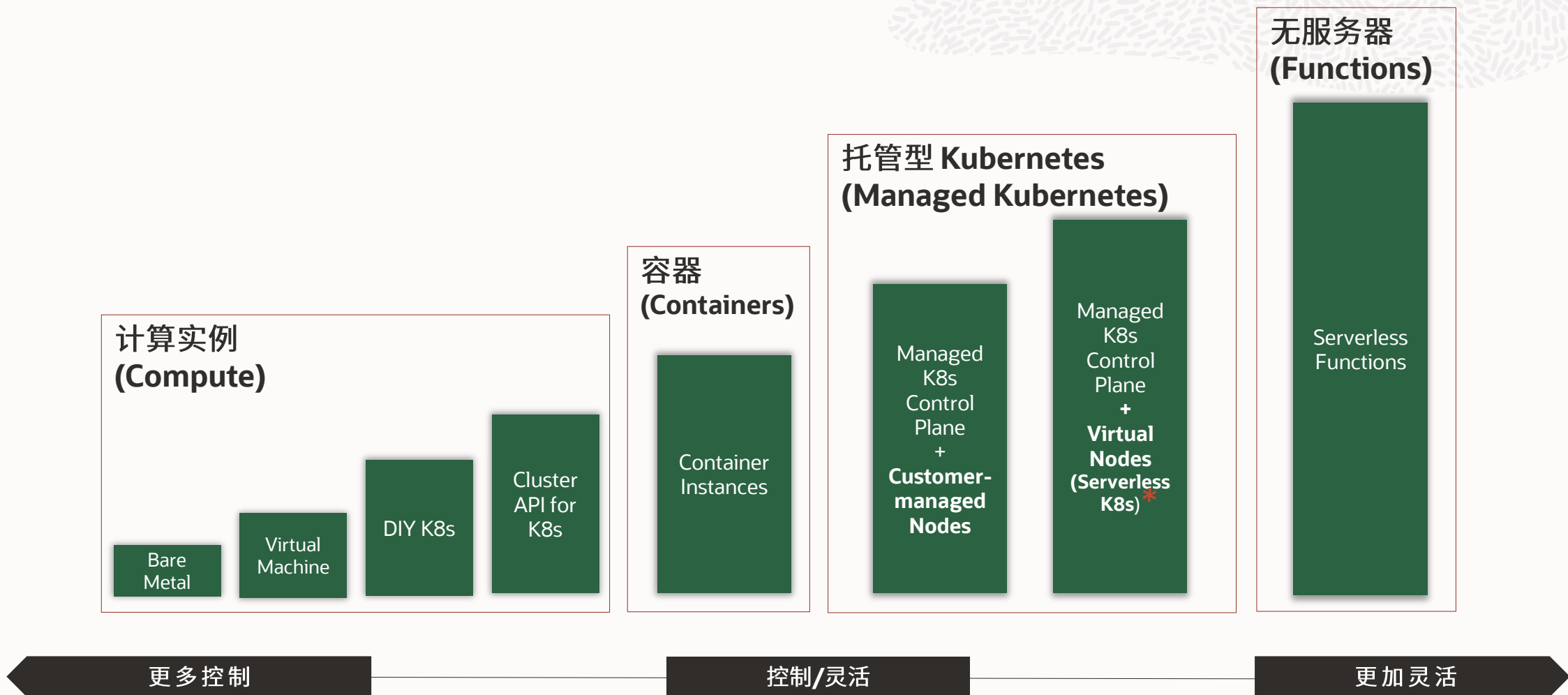


- 通常，（Cloud Native）云原生应用程序是作为一组微服务构建的，这些微服务在符合开放容器（Open Container Initiative）倡议的容器（如 Containerd）中运行，并且可以在 **Kubernetes** 中进行编排，并使用 DevOps 和 Git CI 流水线进行管理和部署。
- 使用容器的优点是能够将执行所需的所有软件打包到一个可执行包中。
- 容器在虚拟化环境中运行，该环境将包含的应用程序与其环境隔离开来。

来源（机器翻译）：[云原生计算 - 维基百科 \(wikipedia.org\)](https://www.wikipedia.org)



# 甲骨文云上的多种容器服务



\*: 待发布



# Container 和 Kubernetes 服务

在甲骨文云上运行容器工作负载



甲骨文云 容器实例

**(OCI Container Instances)**

以简单、快速、安全的方式运行容器，无需管理虚拟机

新发布



甲骨文云 托管型Kubernetes服务

**(OCI Container Engine for Kubernetes(OKE))**

管控节点

**(Managed Nodes)**

客户管控的节点

虚拟节点

**(Virtual Nodes)**

无服务器 Kubernetes

待发布



# OCI 容器实例(Container Instances)

# 甲骨文云容器实例(OCI Container Instances)

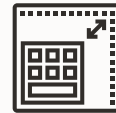
以简单、快速、安全的方式在OCI中运行容器



用于容器的  
无服务器计算实例



快速启动应用程序



多达128个vCPU和  
1024GB内存\*

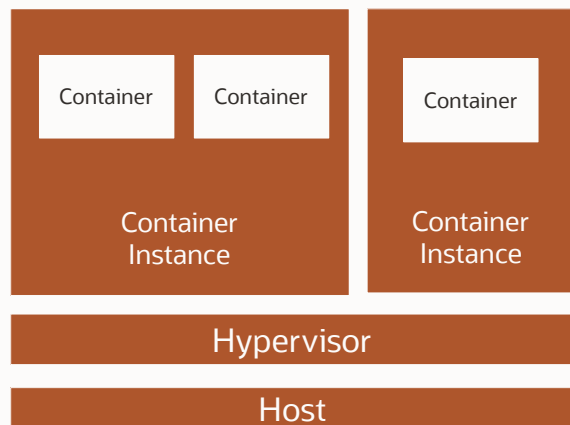
\* 由E3/E4 Flex Shapes支持的CPU/内存



与“常规”计算实例的价  
格相同



# 主要特点



- 甲骨文管理
- 用户管理

- 无服务器计算实例 – 无需管理服务器
  - 选择合适的计算实例配置；可选E4/E3 Flex，计划后续增加更多选项
  - 可以灵活指定CPU/内存资源
  - 可以分配计算实例配置所提供的所有资源，以支持高负载的应用程序
  - 包括15GB的临时存储
- 简单、快速、灵活
  - 使用控制台、CLI、API、Terraform轻松启动
  - 每个实例有一个或多个容器
  - 从甲骨文云容器镜像库(OCIR)或外部容器库拉取容器镜像
  - 可选配置 – 优雅关机、重启策略、环境变量、启动选项等
- 安全、网络和可观察性
  - 强大的隔离性 – 各实例之间不共享内核和资源
  - 通过虚拟云网络(VCN)进行安全通信，可选择分配公共IP
  - 使用资源认证(resource principal)的IAM访问控制甲骨文云上的其他服务
  - 在控制台查看日志或使用API拉取



# 使用场景



- 不需要容器编排（如 Kubernetes）的容器化应用
  - APIs 和 web 应用程序
  - 临时工作负载
    - CI/CD 流水线作业
    - 开发/测试环境
    - 数据/媒体处理
    - 自动化任务
  
- 直接在服务器/虚拟机上运行的独立容器工作负载
  
- 将旧版应用程序容器化，直到分解为云原生应用程序

注意：要在 Kubernetes 上运行应用程序而不想管理基础设施，请使用 OKE 虚拟节点(Virtual Nodes)服务



# 容器实例(Container Instances)演示(1/7)

## ➤ 添加基本信息

可以灵活指定CPU/内存资源

<input type="text" value="1"/>	<input type="text" value="64"/>	*
或者		
<input type="text" value="1024"/>	<input type="text" value="1"/>	

名称  
myblog

配置 折叠

CI.Standard.E4.Flex  
Processor 2.25 GHz AMD EPYC™  
7J13 (Milan) ✓

CI.Standard.E3.Flex  
Processor 2.25 GHz AMD EPYC™  
7742 (Rome)

选择 OCPU 数

1 16 32 48 64

选择内存量 (GB)

1 256 512 768 1024

\*注意：甲骨文云 10CPU=2VCPU



# 容器实例(Container Instances)演示(2/7)

## ➤ 配置容器

### 映像

中心: 外部注册表  
工作版本: mysql:5.7.40 选择映像

### 环境变量

键	值	
<input type="text" value="MYSQL_ROOT_PASSWORD"/>	<input type="text" value="123456"/>	×
<input type="text" value="MYSQL_DATABASE"/>	<input type="text" value="wordpress"/>	×
<input type="text" value="MYSQL_USER"/>	<input type="text" value="oracle"/>	×
<input type="text" value="MYSQL_PASSWORD"/>	<input type="text" value="123456"/>	×





# 容器实例(Container Instances)演示(4/7)



## ➤ 配置容器

容器 - 2

Name 可选

映像

中心: 外部注册表  
工作版本: wordpress:php8.2

选择映像



# 容器实例(Container Instances)演示(5/7)



## ➤ 配置容器

环境变量

键	值	
WORDPRESS_DB_HOST	127.0.0.1	×
WORDPRESS_DB_USER	oracle	×
WORDPRESS_DB_PASSWORD	123456	×
WORDPRESS_DB_NAME	wordpress	×
WORDPRESS_TABLE_PREFIX	myblog	×

+ 其他变量



# 容器实例(Container Instances)演示(6/7)

[容器实例](#) > [容器实例详细信息](#)

➤ 创建成功



活动

## myblog

编辑

启动

停止

重新启动

更多操作 ▾

实例信息

标记

### 一般信息

可用性域: AD-1

容错域: -

OCID: ...2hzvka [显示](#) [复制](#)

启动时间: 2023年1月13日周五 UTC 09:13:32

上次更新时间: 2023年1月13日周五 UTC 09:14:36

区间: ██████████

配置: ██████████

正常关闭超时 (秒) :-

容器重新启动策略: 始终

### 主要 VNIC

名称: ██████████

OCID: ...dfdz6a [显示](#) [复制](#)

公共 IP 地址: ██████████.113 [复制](#)

专用 IP 地址: ██████████ [复制](#)

MAC 地址: 02:00:17:01:36:A8

网络安全组: 无 [编辑](#) ⓘ

子网: ██████████

网: ██████████

专用 DNS 记录: 启用

主机名: ██████████

内部 IP 地址: ██████████

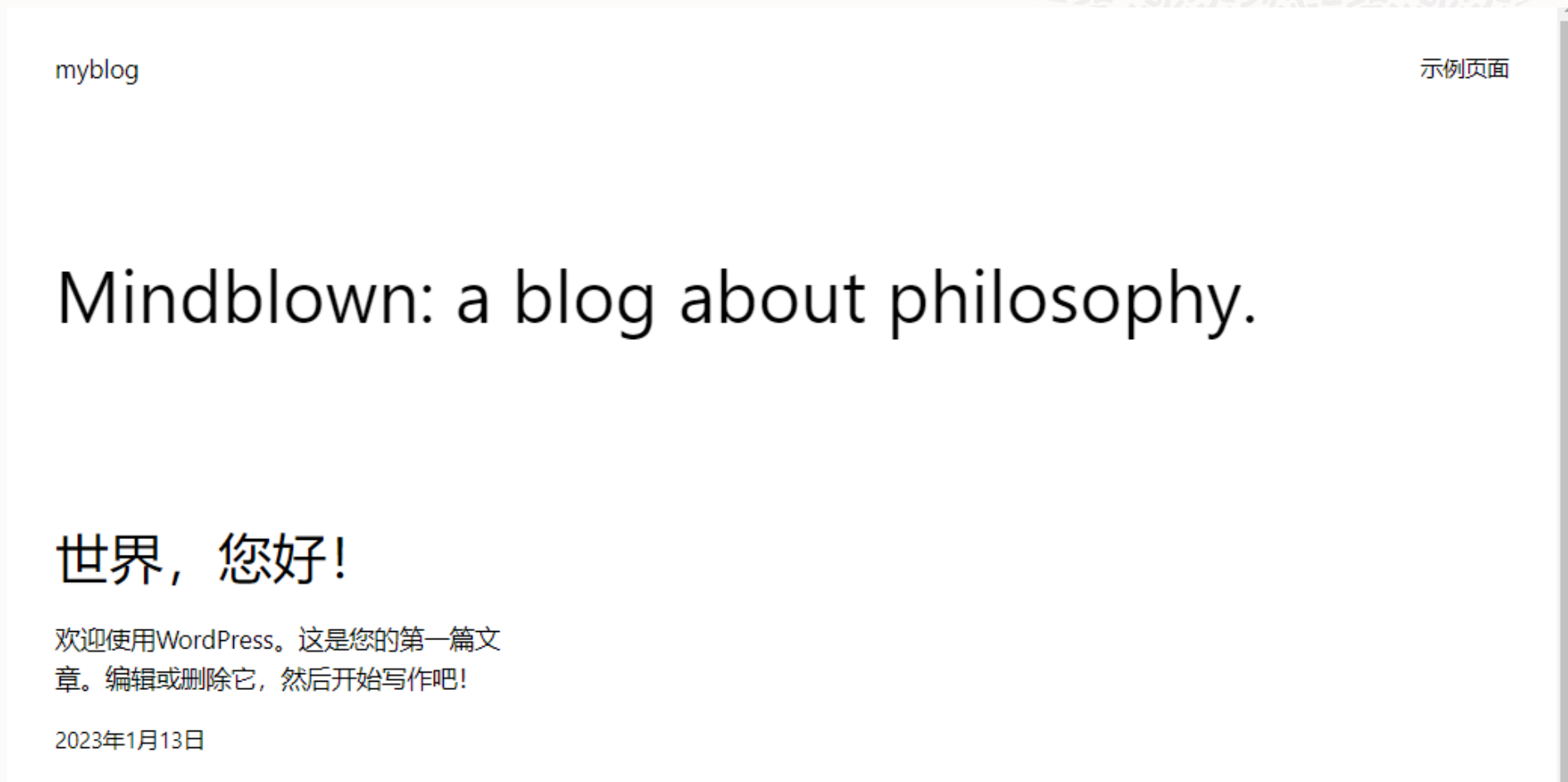
FQDN: ██████████





# 容器实例(Container Instances)演示(7/7)

➤ 访问wordpress



# Oracle Container Engine for Kubernetes (OKE)

# Oracle Container Engine for Kubernetes (OKE)



# OKE 提供管控节点和虚拟节点2种模式



甲骨文云 托管型Kubernetes服务

(OCI Container Engine for  
Kubernetes(OKE))

管控节点

(Managed Nodes)

客户管控的节点

虚拟节点

(Virtual Nodes)

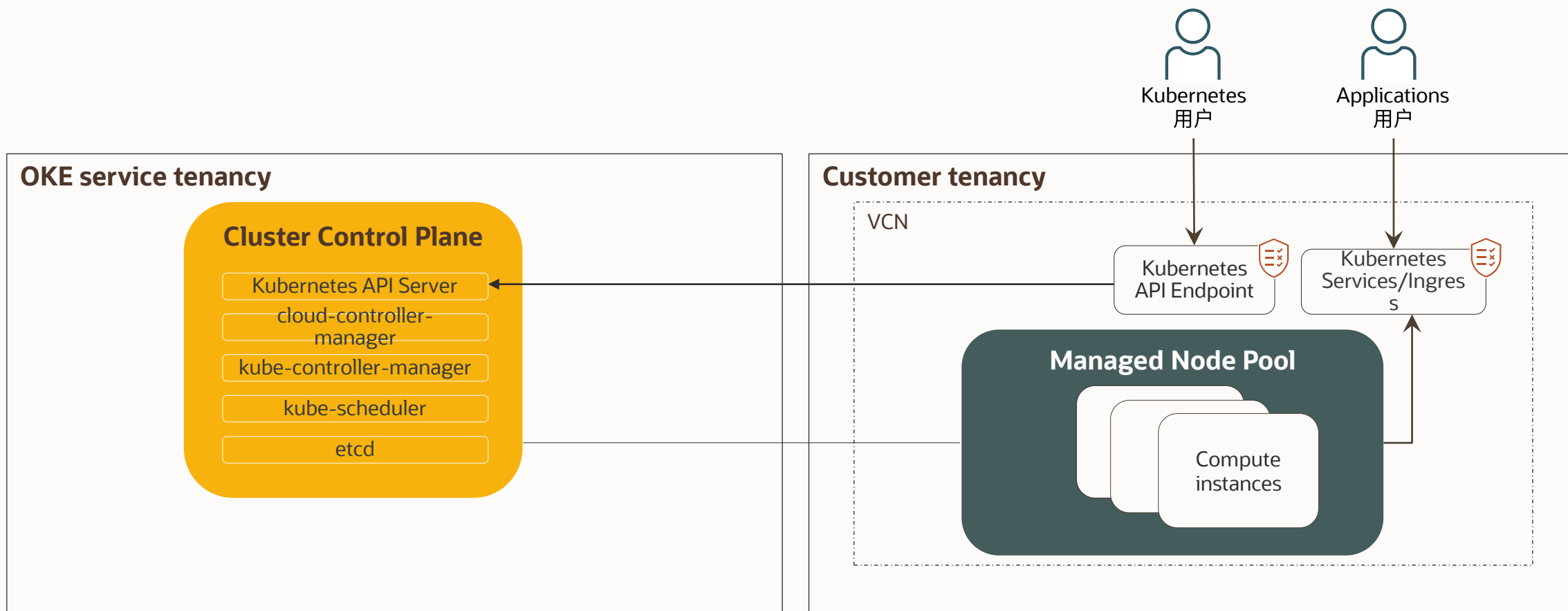
无服务器 Kubernetes

待发布



# OKE 管控节点(Managed Nodes)

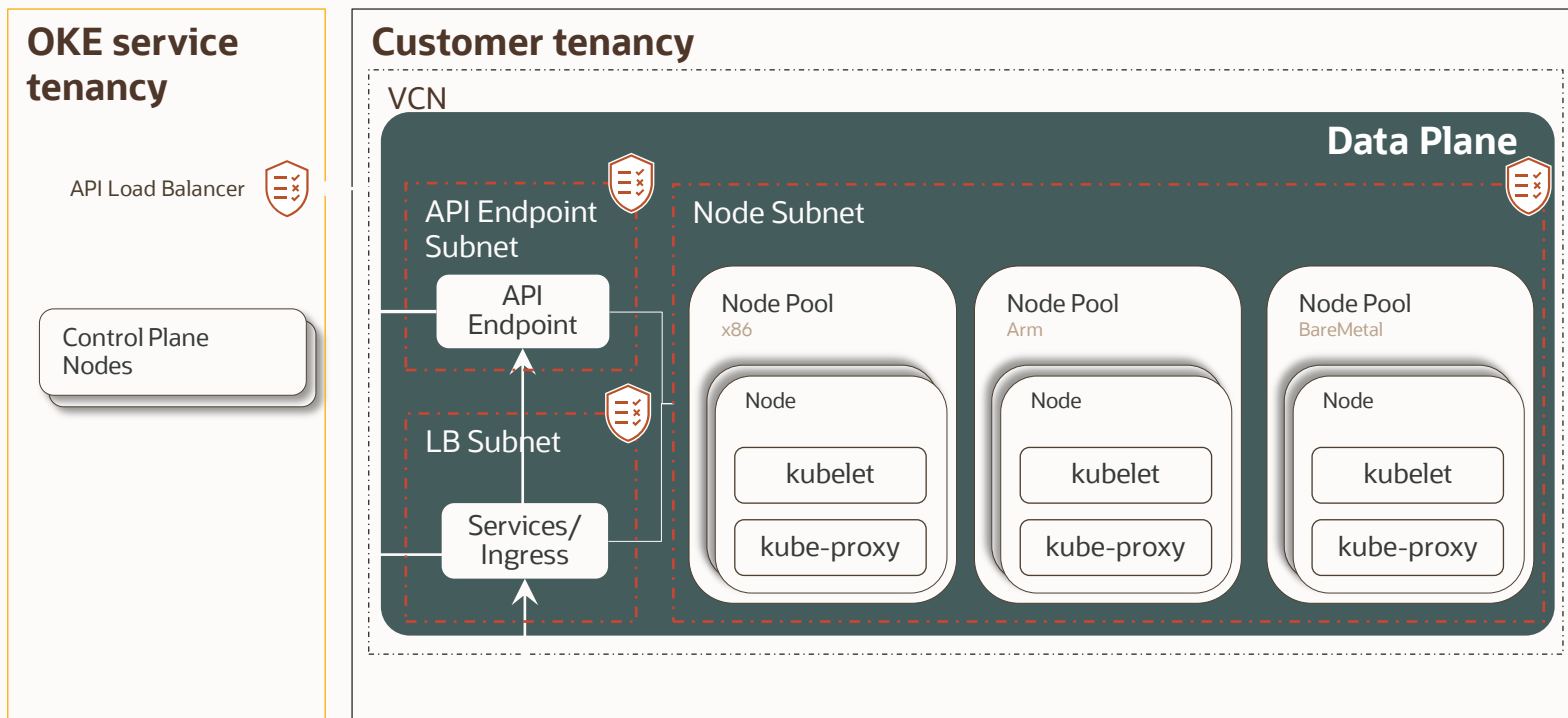
# OKE 管控节点(Managed Nodes)



Kubernetes 集群控制平面由 OKE 管理。集群数据平面在客户的租户中运行。



# 高度被管理的 Kubernetes 工作节点



## 灵活的数据平面

- ✓ 异构节点
- ✓ 支持Cluster Autoscaler
- ✓ 轻松升级管理平面
- ✓ 轻松升级工作节点
- ✓ 节点镜像定制



# 托管型Kubernetes(OKE)演示(1/5)

## ➤ 创建集群 – 快速创建

### 创建集群 [帮助](#)

#### 快速创建

选择“快速创建”选项将创建新的集群，同时创建新的网络资源。新的网络资源包括用于 Kubernetes API 端点的一个区域子网、用于 worker 节点的一个区域子网，以及用于负载均衡器的另一个区域子网。 ✓

#### 定制创建

选择“定制创建”选项将创建新的集群，可以指定要在其中使用的现有网络资源。这包括选择现有公共子网或专用子网（将在其中托管 Kubernetes API 端点、worker 节点和负载均衡器）。

**i** **新资源包括：**

- 虚拟云网络 (VCN)
- Internet 网关 (IG)
- NAT 网关 (NAT)
- 服务网关 (SGW)
- Kubernetes 集群
- Kubernetes worker 节点和节点池

提交 取消





# 托管型Kubernetes(OKE)演示(2/5)

## ➤ 创建集群 – 快速创建

### 创建集群 (快速)

**1 创建集群**  
**2 复查**

名称  
demo-cluster

区间  
[模糊显示]

Kubernetes 版本 ⓘ  
v1.24.1

Kubernetes API 端点

<b>专用端点</b> 创建的 Kubernetes 集群将在专用子网上托管	<b>公共端点</b> 创建的 Kubernetes 集群将在自动分配了公共 IP 地址的公共子网上托管 ✓
---	---

Kubernetes worker 节点

<b>专用 worker</b> 创建的 Kubernetes worker 节点将在专用子网中托管 ✓	<b>公共 worker</b> 创建的 Kubernetes worker 节点将在公共子网中托管
---	---



# 托管型Kubernetes(OKE)演示(3/5)



## ➤ 创建集群 – 快速创建

The screenshot shows the configuration interface for a Kubernetes cluster shape. At the top, a dropdown menu is set to "VM.Standard.E4.Flex". Below this, a text box explains that the number of OCPU units can be customized, with other resources scaling proportionally. Two sliders are present: "选择 OCPU 数" (Select OCPU count) with a value of 1, and "内存量 (GB)" (Memory in GB) with a value of 16. Below the sliders, the network bandwidth is set to 1.0 Gbps and the maximum number of VNICs is 2. At the bottom, the "节点计数" (Node count) is set to 3. A "显示高级选项" (Show advanced options) link is visible at the bottom left of the configuration area.

Shape ⓘ

VM.Standard.E4.Flex

您可以定制分配给弹性配置的 OCPU 数。其他资源将按比例缩放。[了解有关弹性配置的更多信息。](#)

选择 OCPU 数

1 16 32 48 64 1

内存量 (GB) ⓘ

1 256 512 768 1024 16

网络带宽 (Gbps): 1.0 ⓘ      最大 VNIC 总数: 2 ⓘ

节点计数 ⓘ

3

[显示高级选项](#)



# 托管型Kubernetes(OKE)演示(4/5)

- 创建成功 – 访问集群

## 访问集群 [帮助](#)

### Cloud Shell 访问权限

通过 Cloud Shell 使用 Kubectl 远程管理集群。 ✓

### 本地访问权限

使用 kubectl 和 Kubernetes 仪表盘在本地管理集群。

通过 Cloud Shell 管理集群。

- 1 启动 Cloud Shell
- 2 要通过 VCN 本机公共端点访问您的集群的 kubeconfig，请复制以下命令：

```
$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1. 复制
```

[了解有关 Cloud Shell 的更多信息](#)

关闭



# 托管型Kubernetes(OKE)演示(5/5)



- 创建成功 – 访问集群

```
Cloud Shell
[用户名]@cloudshell:~ (ap-chuncheon-1)$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1.ap-chuncheon-1 --file $HOME/.kube/config --region ap-chuncheon-1 --token-version 2.0.0 --kube-endpoint PUBLIC_ENDPOINT
Existing Kubeconfig file found at /home/[用户名]/.kube/config and new config merged into it
[用户名]@cloudshell:~ (ap-chuncheon-1)$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
10.0.10.2	Ready	node	11m	v1.24.1
10.0.10.222	Ready	node	11m	v1.24.1
10.0.10.95	Ready	node	11m	v1.24.1

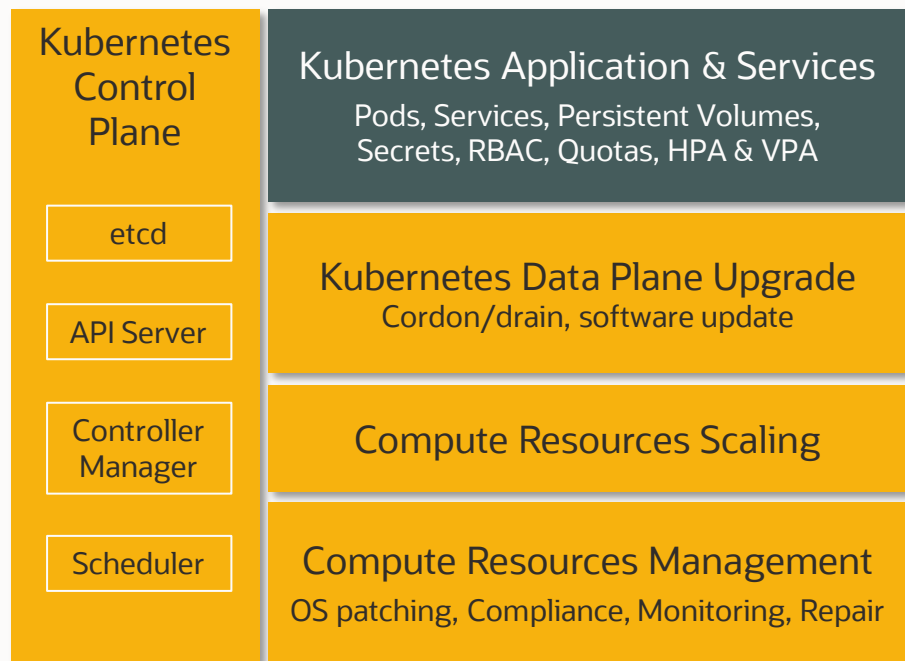


# OKE 虚拟节点(Virtual Nodes)

在不管理任何基础设施的情况下运行 Kubernetes

# OKE 虚拟节点(Virtual Nodes)

## Virtual Nodes 提供 无服务器Kubernetes体验



- Oracle管理
- 用户管理

### 简化的 **Kubernetes** 操作

- 无需基础架构管理
- 简化资源扩展
- 无缝的 Kubernetes 升级

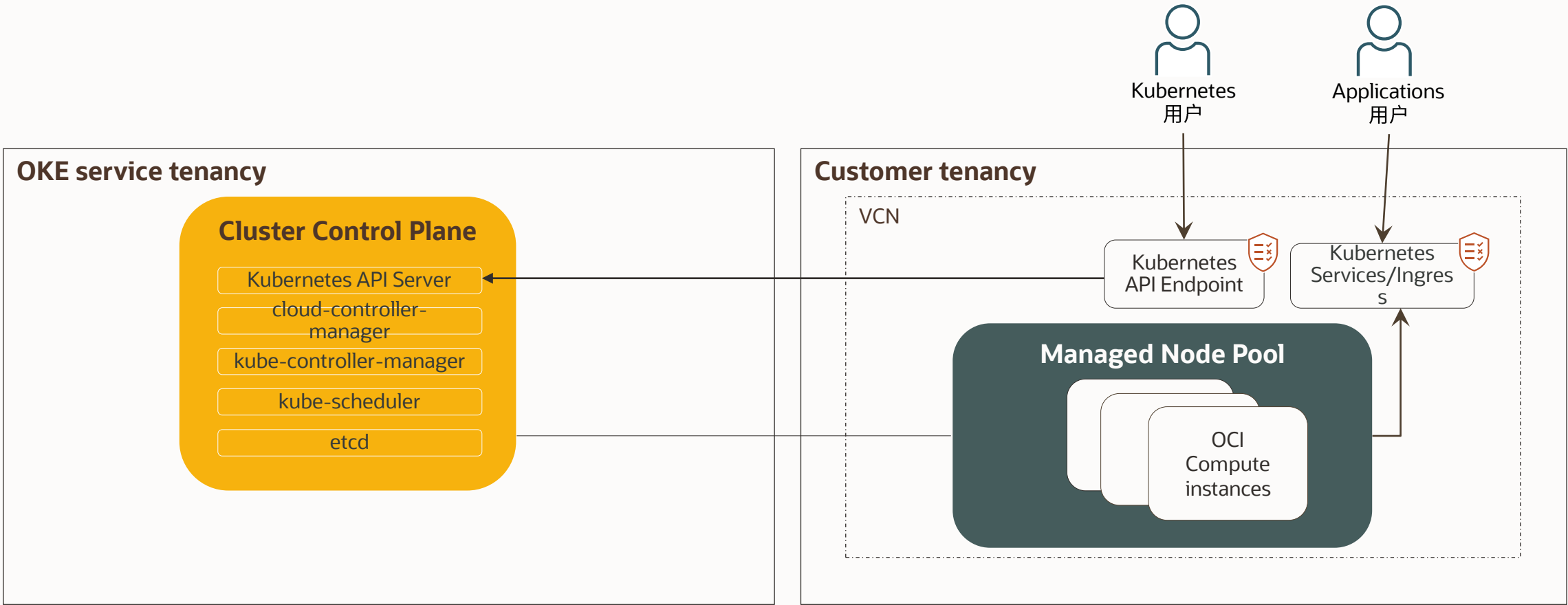
### 灵活支持您的应用需求

- 处理器配置选择和高度垂直可扩展性
- 每个 Pod 的强隔离
- VCN 安全 (Routing, NSG, Flowlogs)

### 成本优化

- 按计算价格为 Kubernetes Pods 使用的资源付费 (待公布, 请以正式发布的价格信息为准)

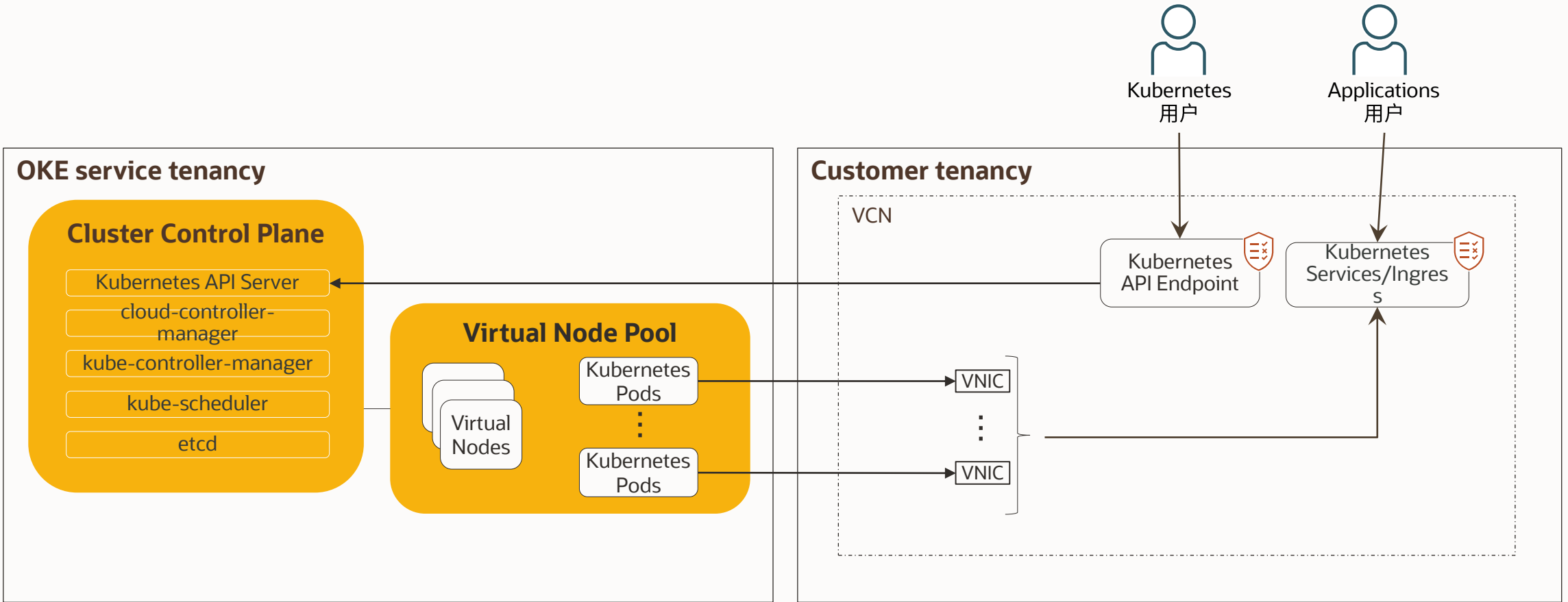
# OKE 管控节点(Managed Nodes)



Kubernetes 集群控制平面由 OKE 管理。集群数据平面在用户的租户中运行。



# OKE 虚拟节点(Virtual Nodes)



Kubernetes 集群控制平面和数据平面由 OKE 管理。



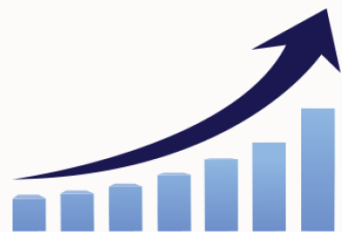


## 使用 OKE 虚拟节点(Virtual Nodes)消除数据平面操作

用户责任	OKE 管控节点(Managed nodes)	OKE 虚拟节点(Virtual Nodes)
工作节点监控	Yes	No
工作节点缩放	Yes	No
工作节点操作系统打补丁	Yes	No
<b>Kubernetes 升级</b>	半托管 / 完全托管(roadmap)	完全托管并与集群升级同步



# 使用场景



## 工作负载用例

- 高度可扩展的无状态应用程序  
例: Web 应用程序与 Autonomous Database 的结合, 在 Autonomous Database 中管理状态
- 高度可扩展的有状态应用程序  
例: Spark
- Jobs、Cron jobs、batch 等。
- 工作流(Workflows)  
例: Argo Workflows
- 缩放到 0 个的应用程序  
例: Knative

## 操作用例

- 通过内置弹性处理使用高峰
- 无需管理基础设施
- 将突增的工作负载到虚拟节点(Virtual node)池中
- 多租户集群
- 按应用程序级计费 (待公布, 请以正式发布的价格信息为准)

# 无服务器 Oracle Functions

完全管理的、高度可扩展的、只在运行时收费的无服务器执行环境



# 什么是无服务器?

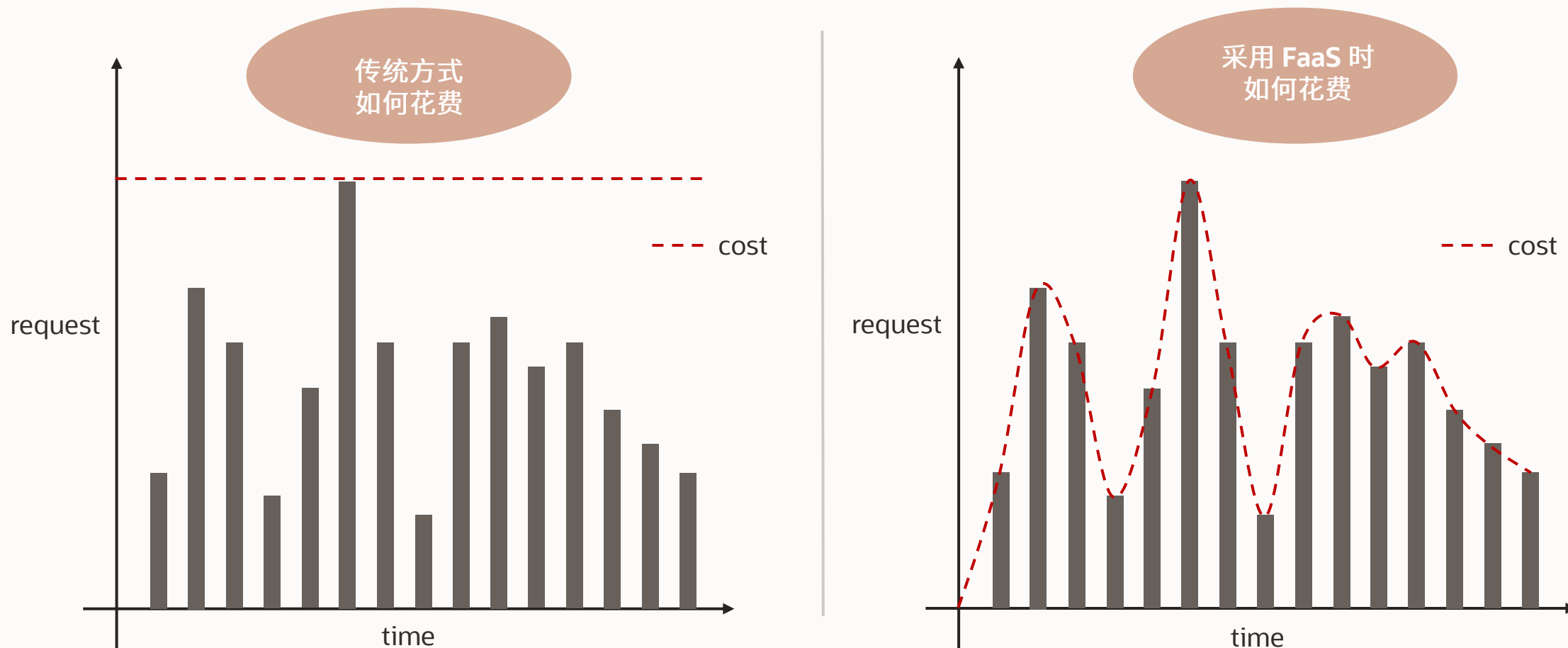


- 在需要时根据需要使用云中的灵活计算资源
- 开发人员不必了解服务器 (= 基础设施)
  - 无需管理服务器，而不是服务器不存在
  - 专注于为用户创造价值



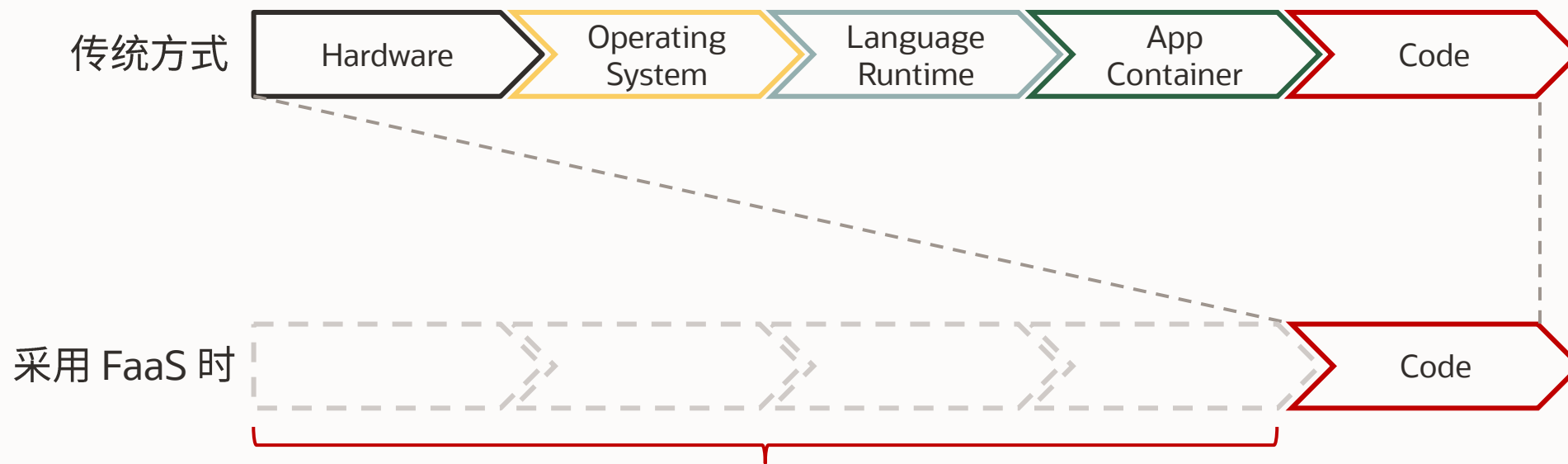
# 采用 FaaS 的好处(1/3) – 降低成本

即用即付，在需要时根据需求分配资源，这更有可能降低成本



## 采用 FaaS 的好处(2/3) – 缩短了开发的准备时间

无需构建基础设施，使实施者能够专注于实现应用程序和为用户提供价值

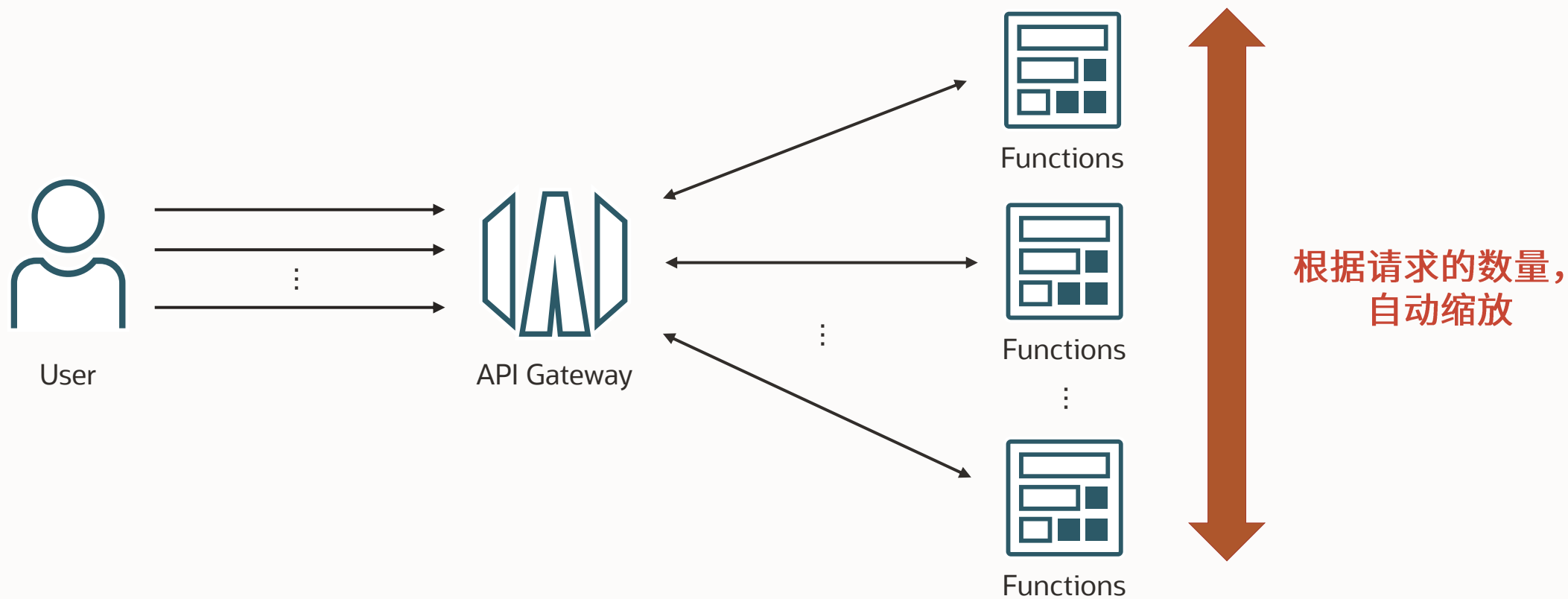


可以用于改进 UX 和功能!



## 采用 FaaS 的好处(3/3) – 可扩展性

自动缩放以匹配请求数，便于灵活响应请求增加或减少



# 无服务器 Oracle Functions

## ➤ 特点

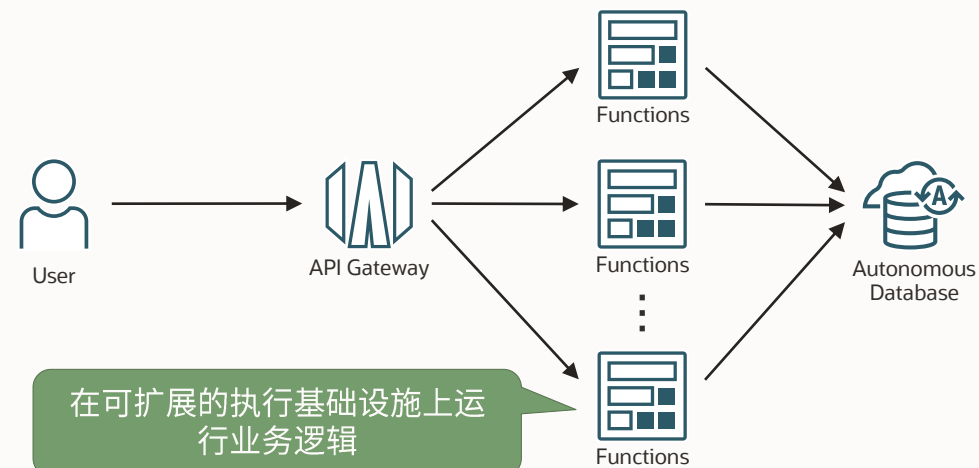
- 完全托管的无服务器执行基础设施 (FaaS)
- 基于开源 Fn Project 项目，无供应商锁定
- 支持多种编程语言作为FDK (Java、Python、Node、Go、Ruby、C#)
- 支持长达 5 分钟的运行时间
- 通过预配(Provisioned Concurrency)的上下文显著缩短首次运行时间
- 提供简单的开发流程和测试方式

## ➤ 价格 (截至2022年12月)

- 前 200 万个请求是免费的，之后每个请求 US\$0.00000002
- 前 400,000 GB 免费，此后每 GB-s = \$0.00001417

(※对于应用了预配传输的未使用Oracle Functions，将收取 25% 的资源使用费)

用作Web, Mobile, IoT业务逻辑执行的基础设施





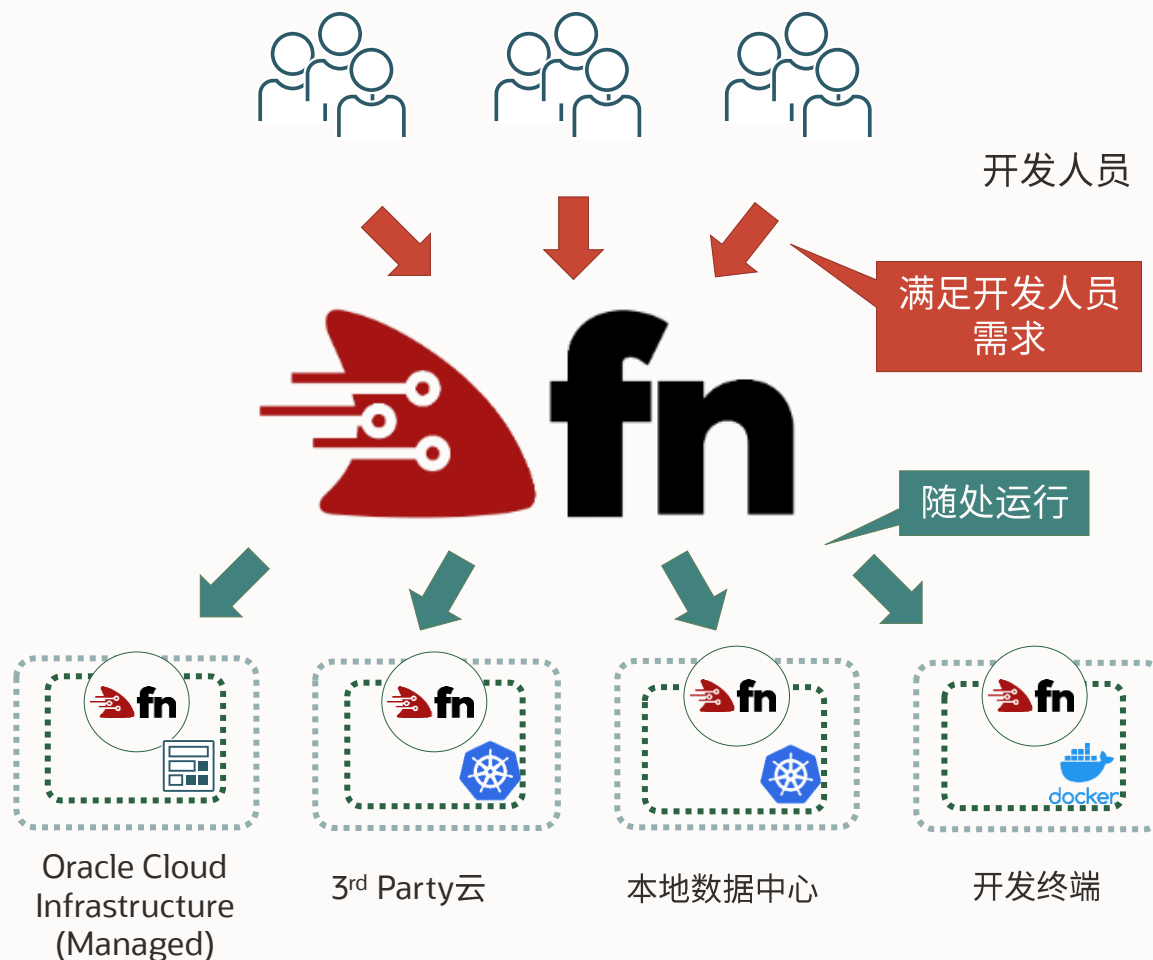
# 基于容器的开源“Fn 项目”

## 基于 OSS 的生态系统

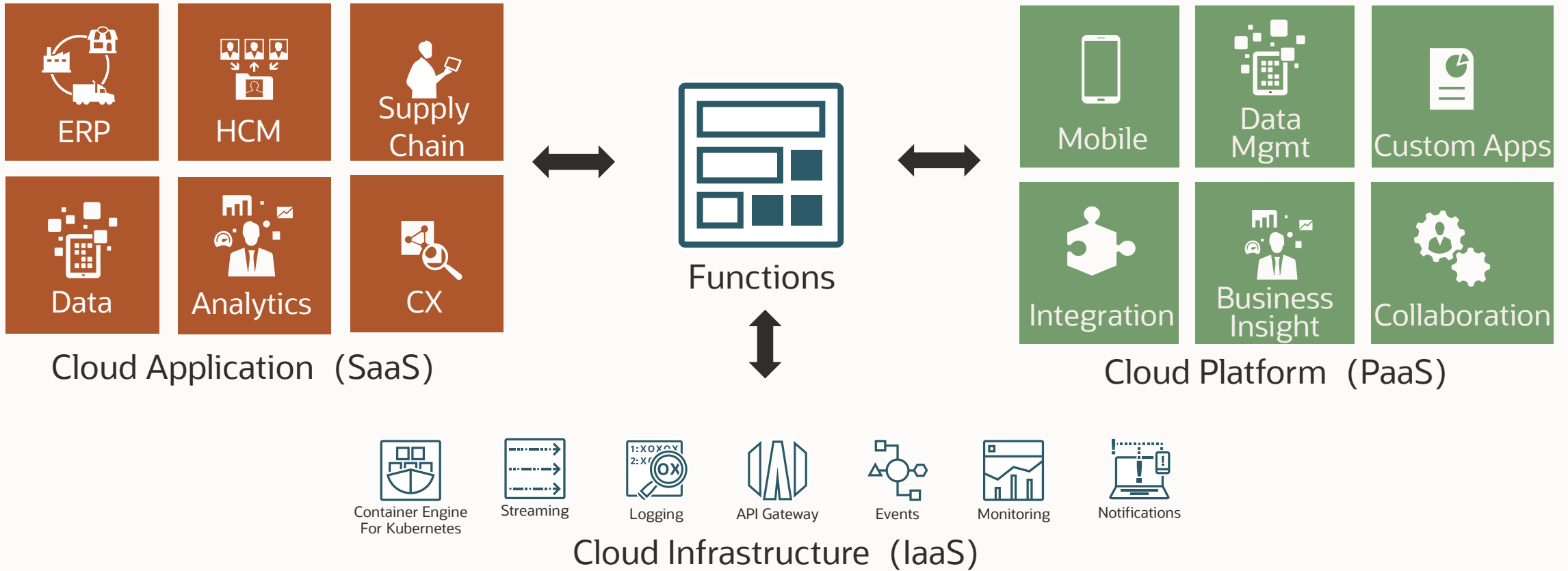
- <https://fnproject.io/>
- 积极反映开发人员的需求
- 无需担心供应商锁定

## 不受场所约束的可移植性

- 从云到本地数据中心都可以运行
- 开发 PC 也能正常运行

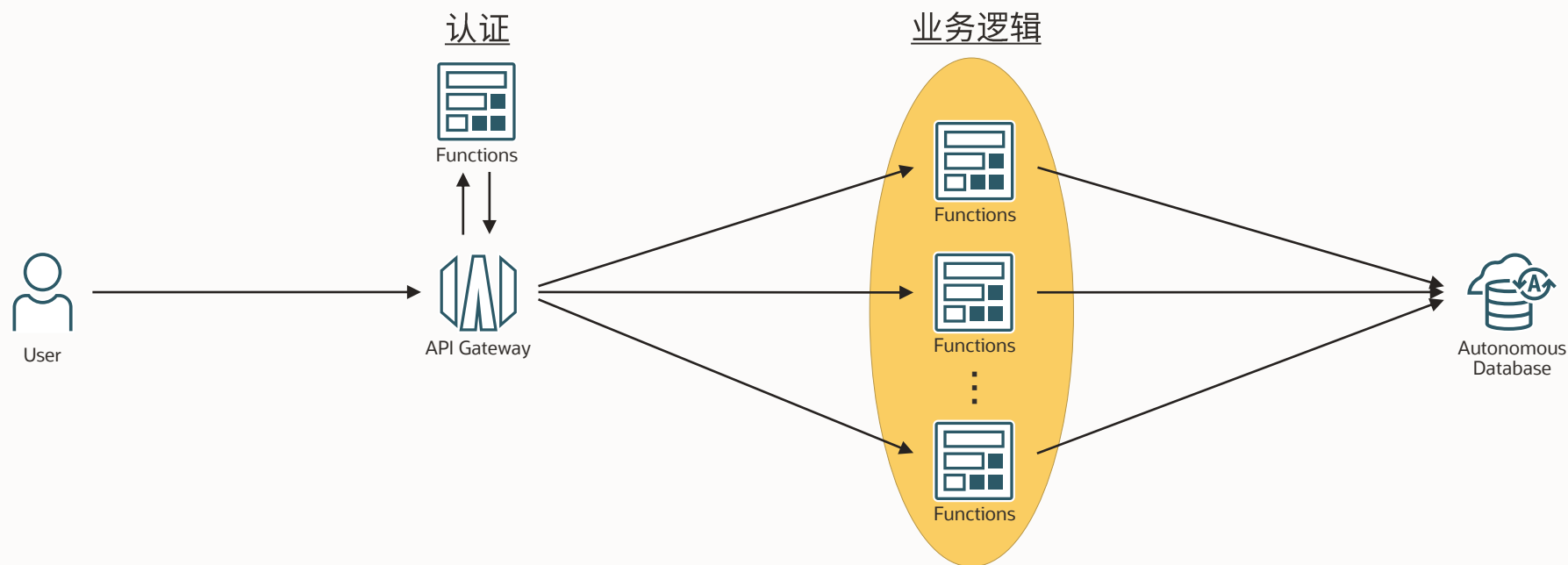


# Oracle Functions 通过与 Oracle 云服务的连接实现各种用例



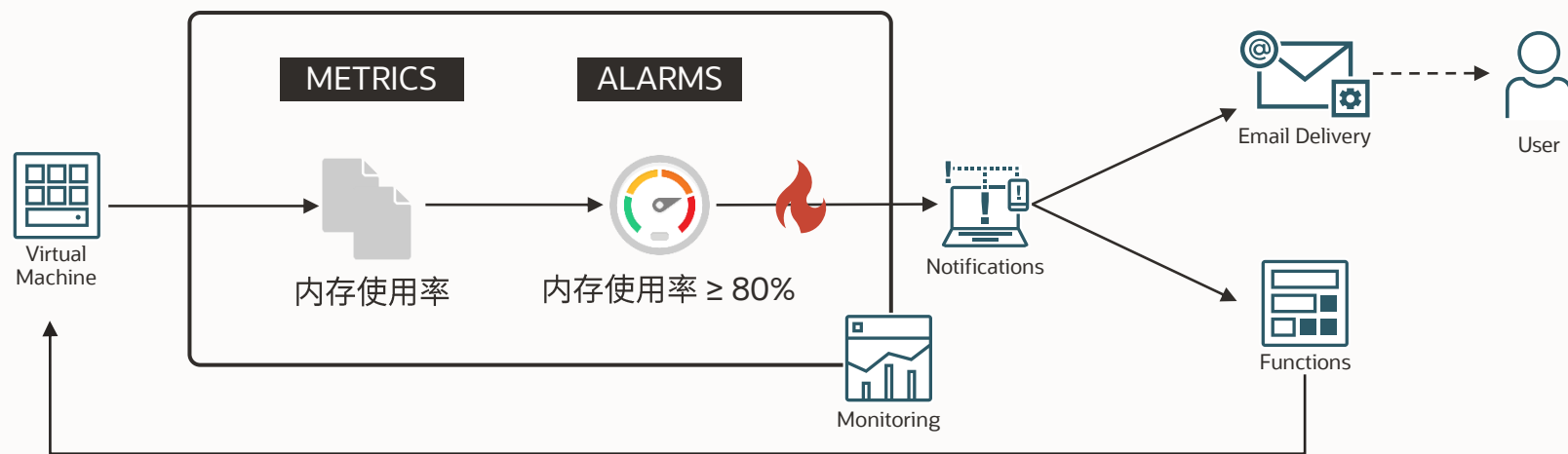
# 使用场景1：执行Web、Mobile、IoT业务逻辑

- 在可扩展性极佳的执行平台上实现业务逻辑。例如：
  - 实施 SaaS 的扩展、单体应用程序的分割、新应用程序的开发等。



## 使用场景2：操作自动化

- 使用 Functions + OCI SDK 自动化一些手动操作
- 由于只对执行时所使用的资源计费，因此在降低成本的同时实现了高效运营

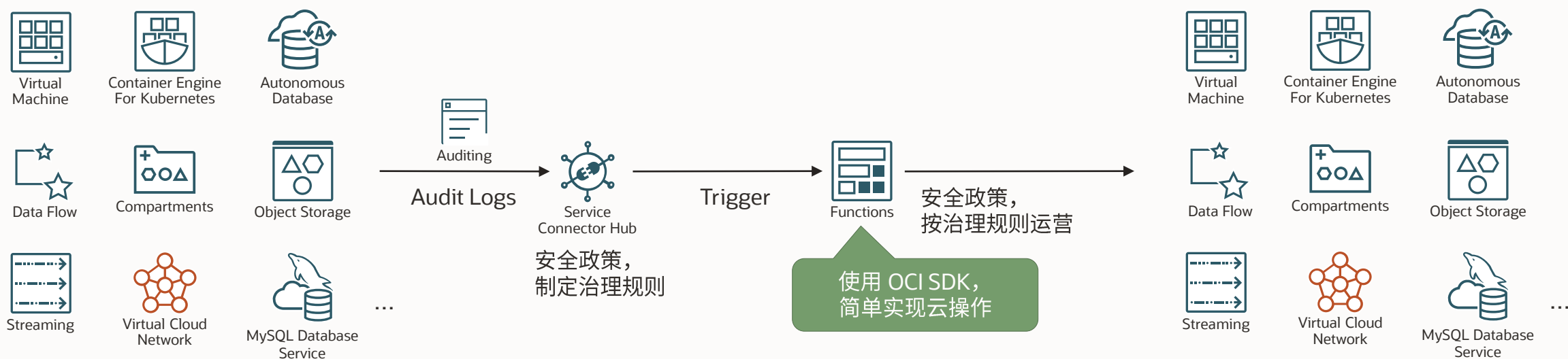


变更计算实例配置(Scale Up)  
VM.Standard2.1 → VM.Standard2.2



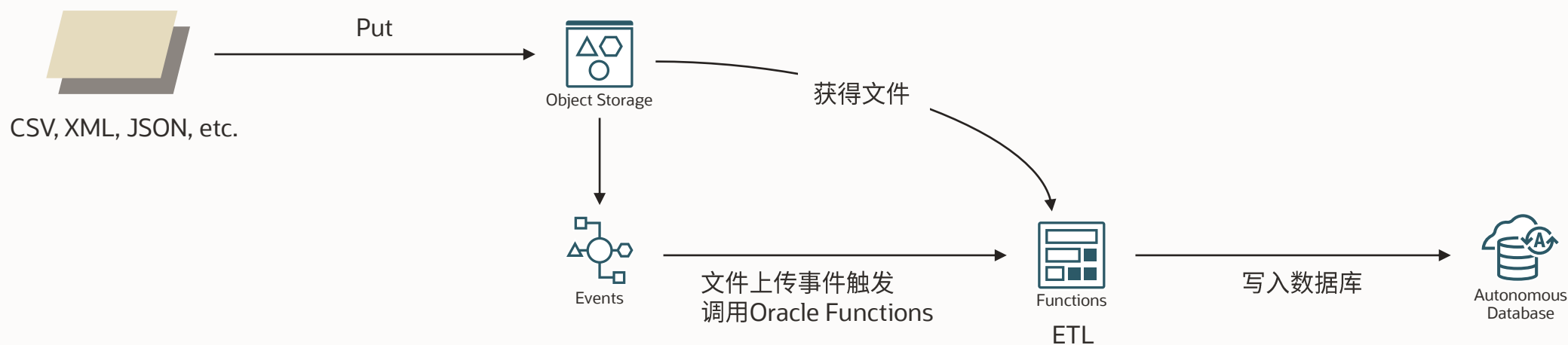
# 使用场景3：执行安全策略和治理规则

- 实施审计日志应用公司安全策略和治理规则。例如：
  - 如果创建了一个公网IP的实例，则关闭该实例
  - 如果创建了具有公开可见性的存储桶，则将其删除
  - 等等



## 使用场景4：事件驱动的 ETL 处理

- 使用文件上传触发 Oracle Functions 并执行ETL处理
- 当您想通过轻量级处理和按需付费定价来降低成本时尤其有效



# 无服务器本地开发演示

1. 安装 fn cli

```
curl -LSs https://raw.githubusercontent.com/fnproject/cli/master/insta
```

2. 启动 fn 服务器

```
fn start
```

3. 为本地开发设置注册表

```
fn update context registry fndemouser
```

4. 创建 Function

```
fn init --runtime go gofn; cd gofn
```

5. 修改 func.go (此演示未做任何修改)

6. 创建 App

```
fn create app goapp
```

7. 发布 Function 到 App

```
fn --verbose deploy --app goapp --local
```

8. 使用 CLI 调用 Function

```
fn invoke goapp gofn  
{"message":"Hello World"}
```



# 容器服务的选择





# 容器服务的选择

## OCI 容器实例 (Container Instances)

- 在几秒钟内运行容器，无需管理任何服务器
- 适用于不需要 Kubernetes 编排的工作负载
- 简单，不需要 Kubernetes 技能

## OKE 虚拟节点 (Virtual Nodes)

- Kubernetes 编排
- 容器由 Virtual Nodes 执行，这些节点为 Kubernetes 节点提供服务
- 消除管理、扩展、升级节点基础架构和排除故障的运营开销
- 需要较少的 Kubernetes 技能

## OKE 管控节点 (Managed Nodes)

- Kubernetes 编排
- 容器由 OCI 计算实例执行，生命周期通过 OKE API 进行管理
- 您可以根据需要控制工作节点的配置
- 需要 Kubernetes 技能

## 无服务器 (Oracle Functions)

- 在几秒钟内运行容器，无需管理任何服务器
- 业务处理可以在短时间内执行完成
- 执行频度不确定，不执行时无需付费
- 根据请求的数量，自动缩放
- 简单，不需要 Kubernetes 技能




ORACLE

ORACLE  
甲骨文

# Oracle数据在线重组 及重定义

## 实战演练工作坊系列(四)




张西东

- 资深解决方案工程师
- 数据库性能调优专家
- 10年+数据库及应用系统运维经验

### 内容简介

在线重组和重新定义功能为管理员提供了前所未有的灵活性来修改表物理属性和转换数据和表结构，同时允许业务完全访问数据库。

- 修改表物理属性并转换数据和表结构，同时允许用户完全访问数据库
- 提高数据可用性
- 使应用程序升级过程更轻松、更安全、更快捷。



直播时间: 2月3日 11:00 - 12:00  
扫描二维码注册并安装手机Zoom进入直播  
Zoom ID: 976 6962 5763 密码: 98039717



数据库和云讲座群

20-20



甲骨文云技术公众号



技术专家1V1深入交流

