

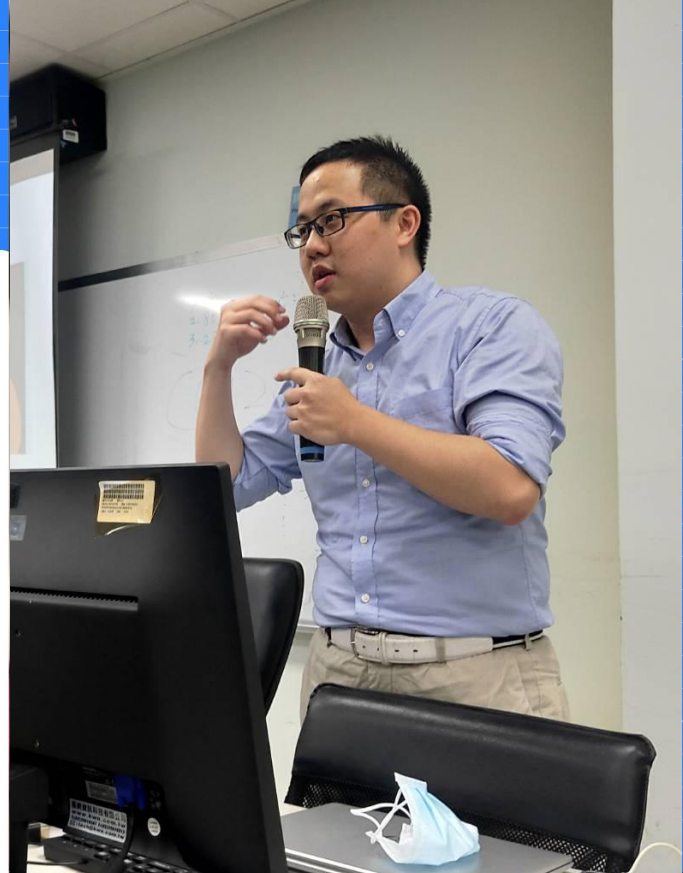
devfest 2022

```
book-nav-toggle'  
dden="" fixed="" aria-label='Hide  
'Hide side navigation'
```

The Evolution of the Serverless Computing - Cloud Function & Cloud Run

linktr.ee/youjun

 Google Developer Groups
Taipei



Youjun Yu(余佑駿)
Sr. Data Scientist, Nextlink Technology

AGENDA

1. Brief intro Cloud Function & Cloud Run
2. Serverless, microservice, event-driven
3. Challenges we will encounter & How to overcome
4. What's new features these year
5. Feature - Concurrency
6. Demo
7. QA

linktr.ee/youjun_talk

bit.ly/devfest_taipei_2022_serverless



devfest
2022

```
" class='time talk-ended single-  
' class='talk-name'>...</th>  
 class='description'>...</th>
```

Brief intro Cloud Function & Cloud Run



Cloud Run

Serverless 應用與容器

<https://cloud.run>

Serverless Anywhere

跨平台 & 跨雲



Cloud Functions

Serverless
已不僅是 FaaS

Serverless 函數

<https://cloud.google.com/functions>

Serverless Compute

Develop, deploy and scale applications fast and securely in a fully managed environment

維運模式



Auto-scaling



Fully Managed

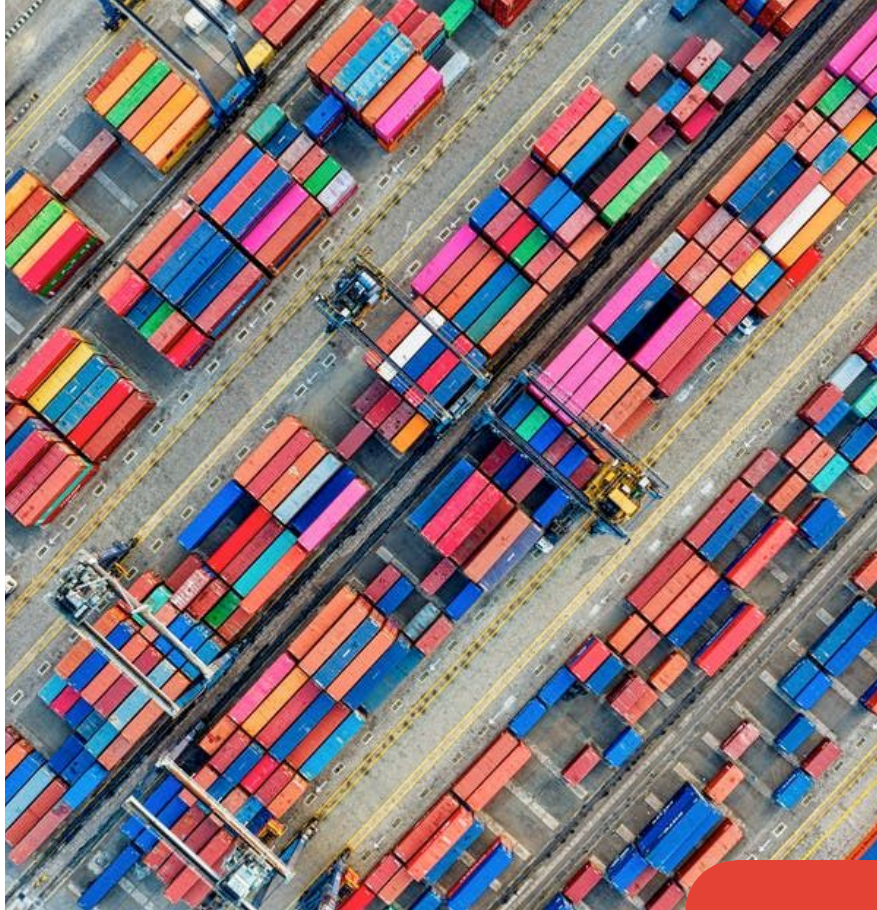
開發模式



Speed to Market



容器 Serverless 體驗如何？



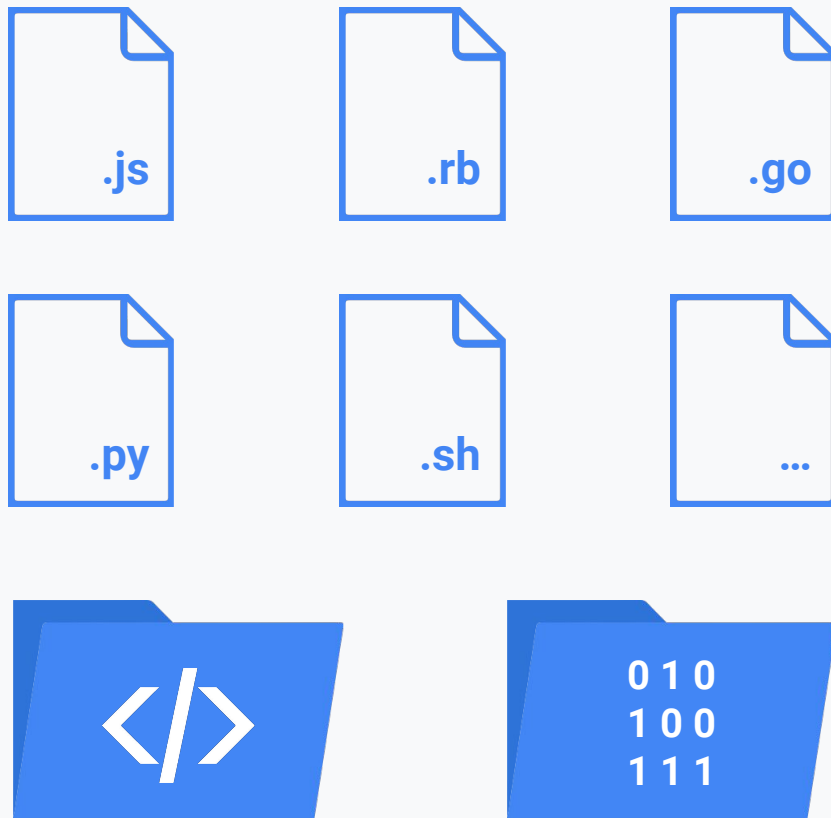
Google Cloud Serverless 容器服務

成長 3 倍

from April 2020 to April 2021

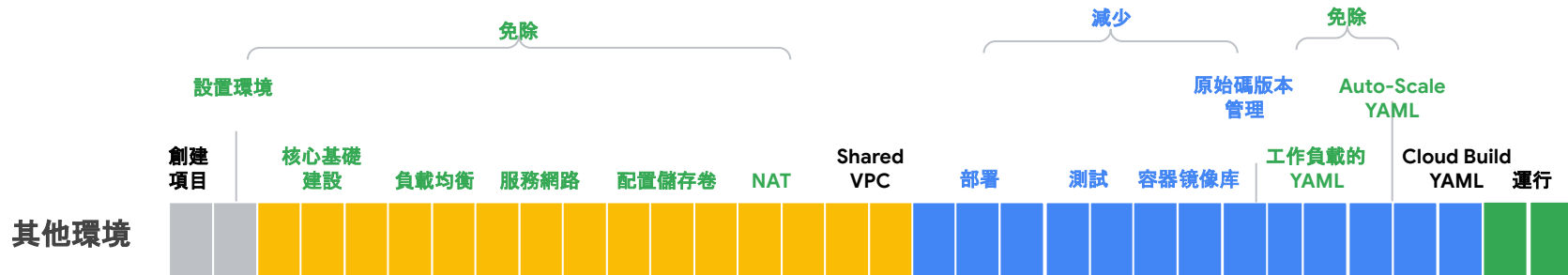
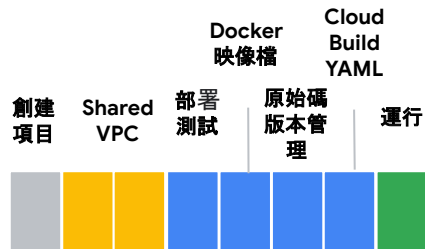
Container

- 任何語言
- 任何程式庫
- 任意程式文件，包括傳統應用程式
- 基礎容器映像檔
- 輕鬆遷移



使用 Cloud Run

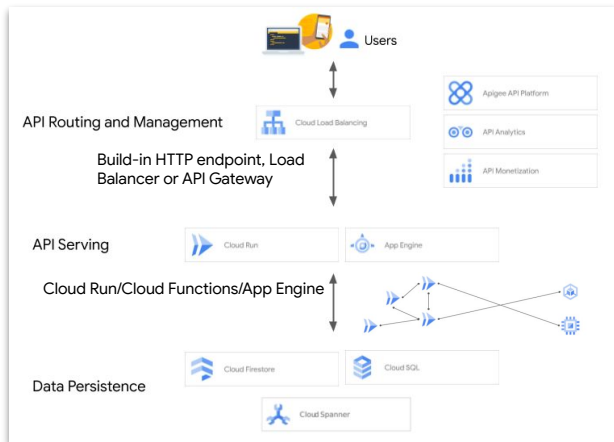
應用程式推出市場速度提高 8 倍 請求回應毫秒級



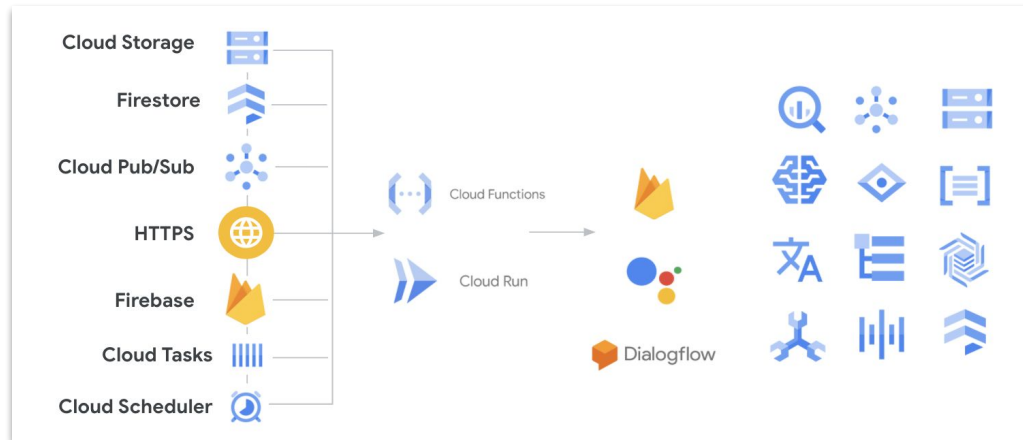
- 啟動
- 設置計算資源
- CI/CD
- 運行

Serverless 常見場景

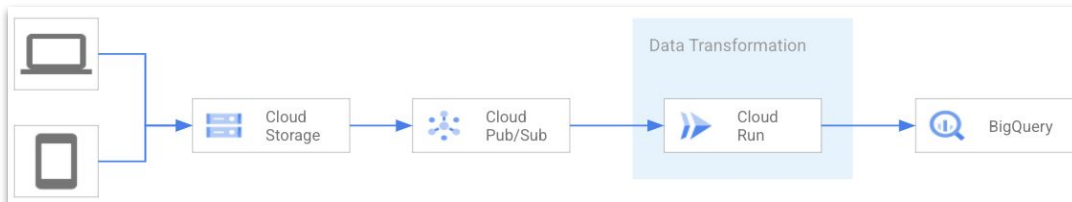
1. Web / 行動後台 / 微服務 API



2. 串接雲端服務



3. 串流資料處理和批次處理



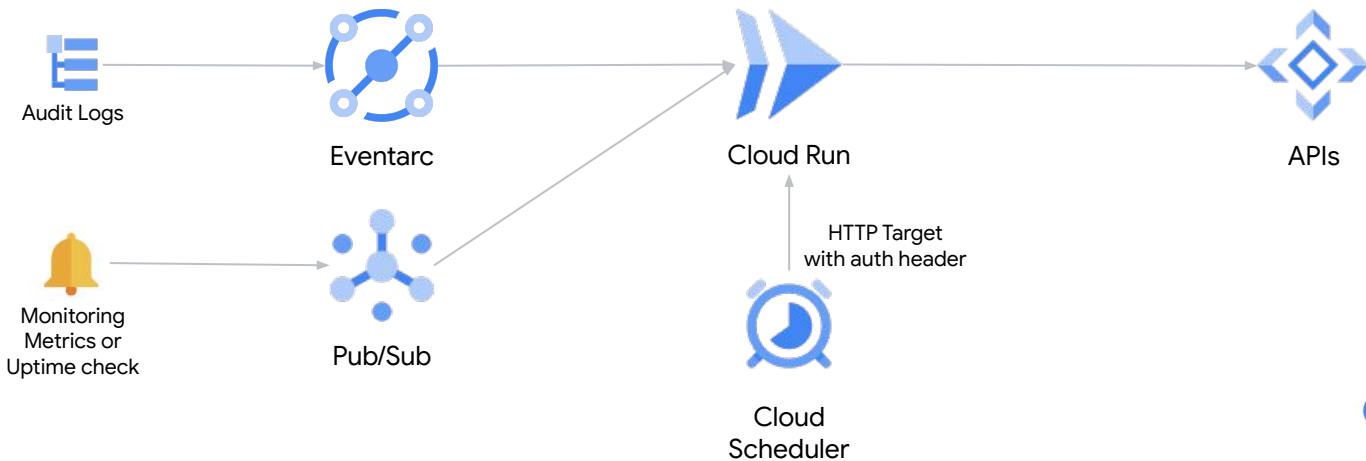
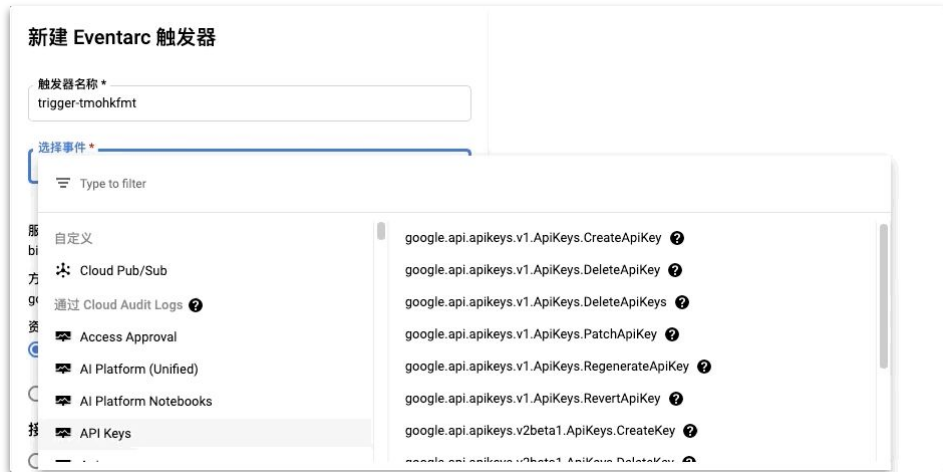
4. 維運自動化

- 🔔 監控警告與事件，驅動自動化回應
例如定時拉起一個叢集
- 🔧 工作流程 & 服務自動化調度

60多項觸發事件來源

應用場景舉例：

- 自動校正策略和配置；
- 自動啟動新 Cluster, 執行資料處理任務
- 監控指標超過警告值自動警告通知
- 修改資料庫自動觸發任務



Cloud Run / Cloud Functions 內建 HTTPS 入口

- 可設置認證 或 不認證
- 可設置面向公共網路 或 只面向 VPC 私有網路
- 不需配置 API Gateway 節省成本
- 支持自定義網域名 Custom domain

HTTPS



Frontend

```
IAM:  
role: "roles/run.invoker"  
member: "allUsers"
```

devfest
2022

`class="time talk-ended single-`
`class="talk-name">...`
`class="description">...`

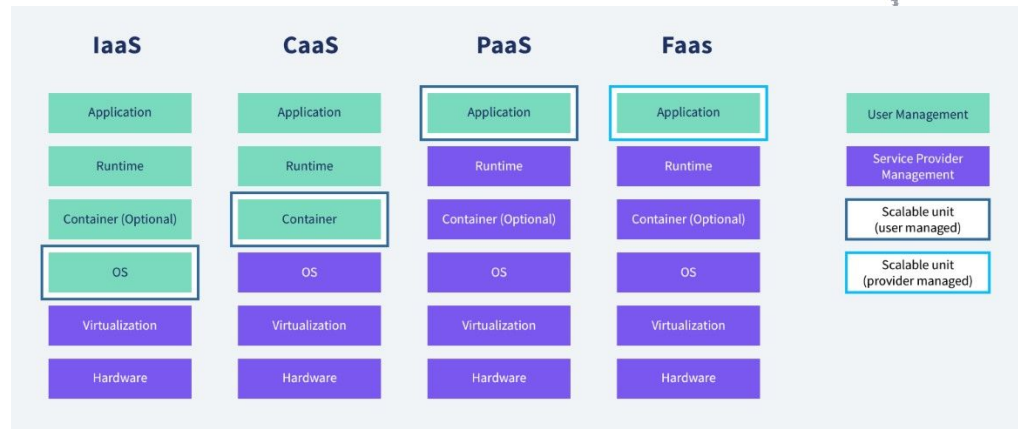
Serverless, Microservice, Event-driven



```
class= devsite-book-nav-toggle
-haspopup='menu' hidden='' fixed='' aria-label='Hide
navigation' data-title='Hide side navigation'
-expanded='true'><span class='material-icons
' /></span></button>
```

“ Serverless ”

Serverless computing is a method of providing backend services on an as-used basis. A serverless provider **allows users to write and deploy code without the hassle of worrying about the underlying infrastructure.**

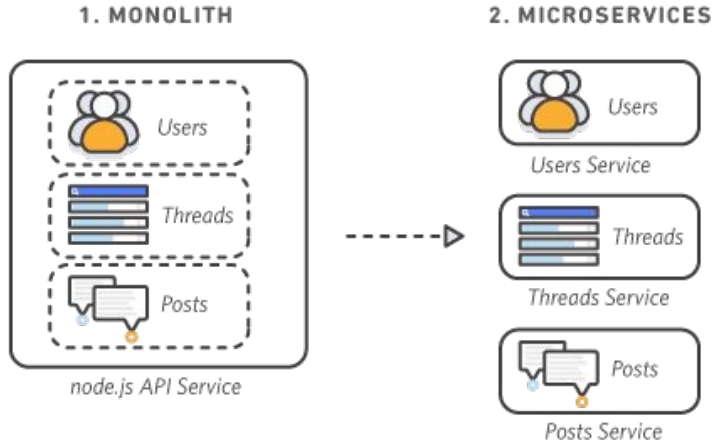


A company that gets backend services from a serverless vendor is charged based on their computation and do not have to reserve and pay for a fixed amount of bandwidth or number of servers, as the service is auto-scaling. **Note that despite the name serverless, physical servers are still used but developers do not need to be aware of them.**



```
class= devsite-book-nav-toggle
-haspopup='menu' hidden='' fixed='' aria-label='Hide
navigation' data-title='Hide side navigation'
-expanded='true'><span class='material-icons
' /></span> /button>
```

“ microservice ”

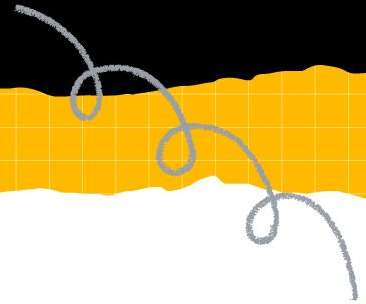


Microservices are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs. These services are owned by small, self-contained teams.

Microservices architectures **make applications easier to scale and faster to develop, enabling innovation and accelerating time-to-market for new features.**



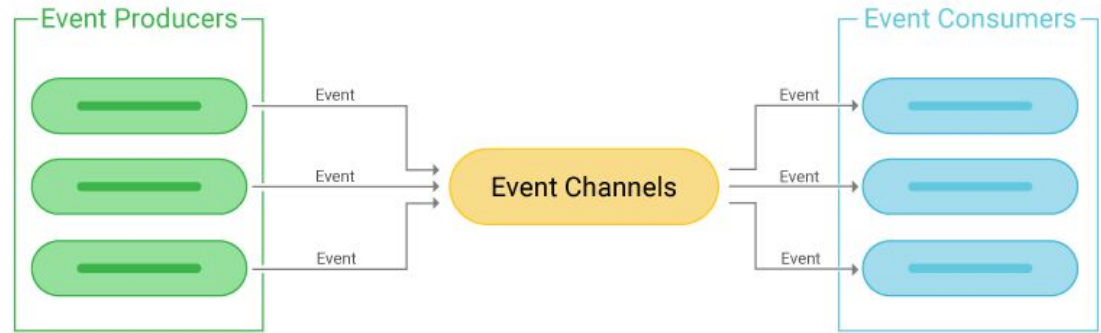
```
on class= devsite-book-nav-toggle
-haspopup='menu' hidden='' fixed='' aria-label='Hide
navigation' data-title='Hide side navigation'
-expanded='true'><span class='material-icons
' /></span></button>
```



```
class= 'device-book-nav-toggle  
-haspopup='menu' hidden=''' fixed=''' aria-label='Hide  
navigation' data-title='Hide side navigation'  
-expanded='true'><span class='material-icons  
</span></button>
```

“ event-driven ”

Event-driven architecture is a modern application design pattern that is based on **principles of loose-coupling and message-driven interaction.**



In a computing context, an event is any identifiable occurrence that has significance for system hardware or software. As such, events include both user-generated actions like mouse clicks and keystrokes and system-generated events such as program loading.

Event-driven programming is an approach rather than a type of language. **Event-driven processing can improve responsiveness, throughput and flexibility.**



devfest
2022

```
" class='time talk-ended single-  
' class='talk-name'>...</div>  
' class='description'>...</div>
```

Challenges we will encounter & How to overcome



客戶對 Serverless 有更高的要求

“我們不希望被綁定在單一雲上面”

“開發更快、門檻更低”

“如何利用已有的二進位程式集？如何安裝XX擴充？”

“我的應用無法接受cold start”

“Serverless function的timeout時間過短”

“能否支援XX開發語言”

“支援高concurrency, 並且能處理百萬以上的QPS”

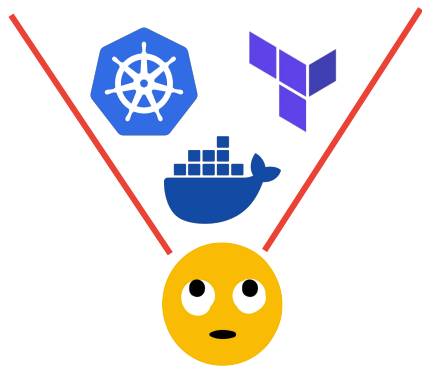
“我要遷移到xx雲, xx環境”



企業的挑戰



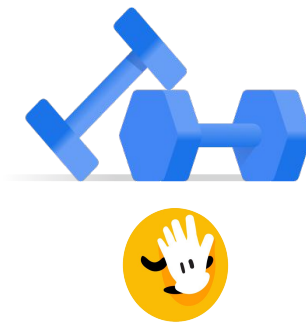
Fast Time to Market
快速交付



Endless... to Learn
一邊Coding, 一邊學新工具

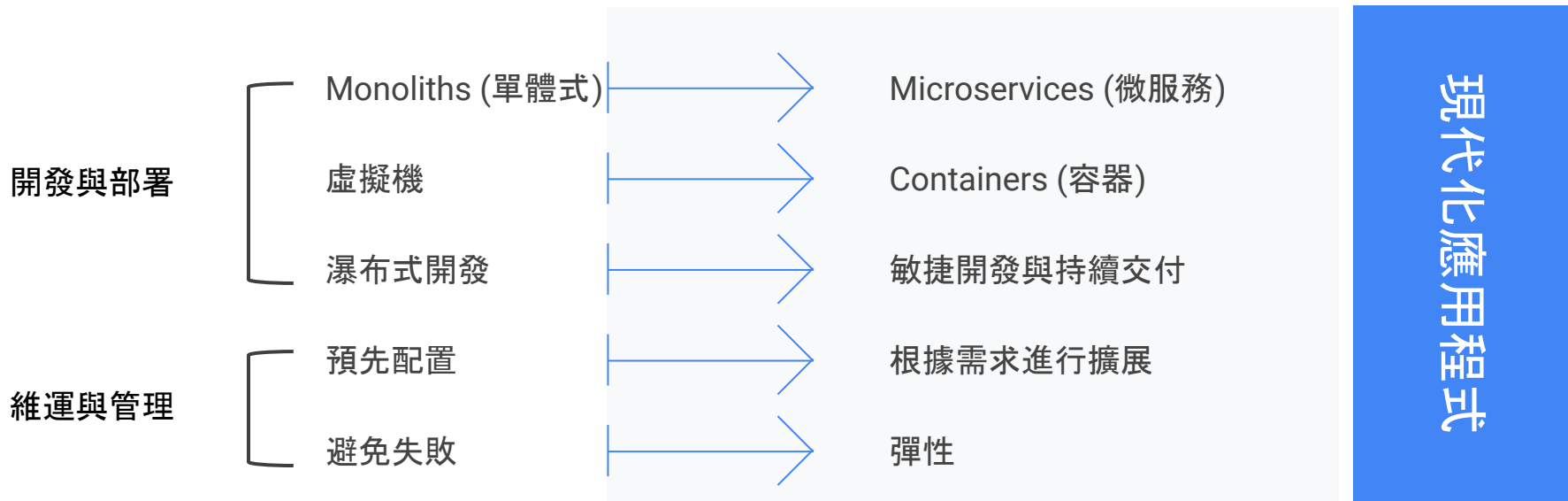


Cost Effective
高性價比

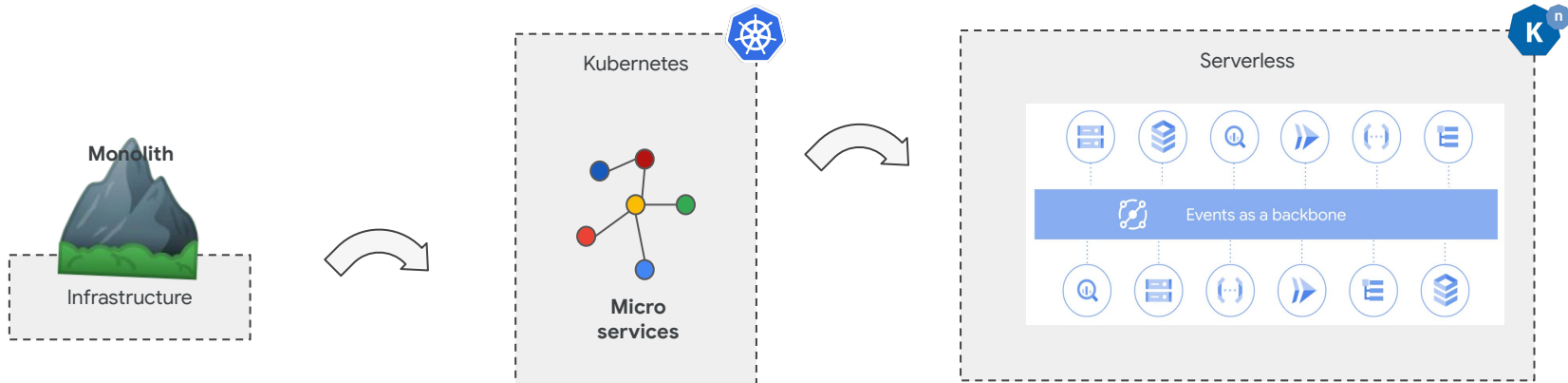


Burst Requests
突發的大量需求

現代化應用程式建立在新的模式之上



單體式應用程式朝向微服務

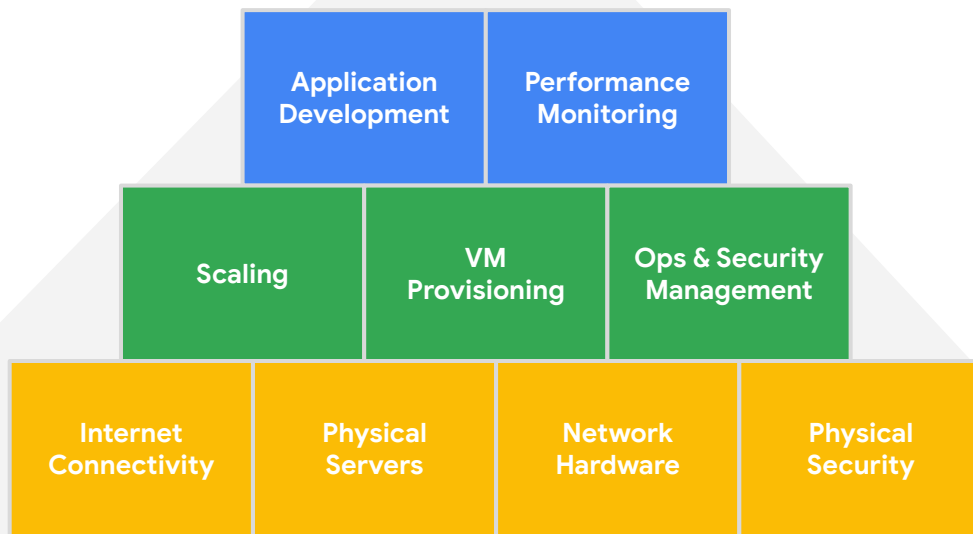


- Large release units
- Strong coupling
- Creaky automation
- Coarse tests

- Smaller release units
- Looser coupling
- Better automation
- Smaller-grained tests

- Lowest coupling
- On-Demand Scalable
- Best automation
- Finest granular tests

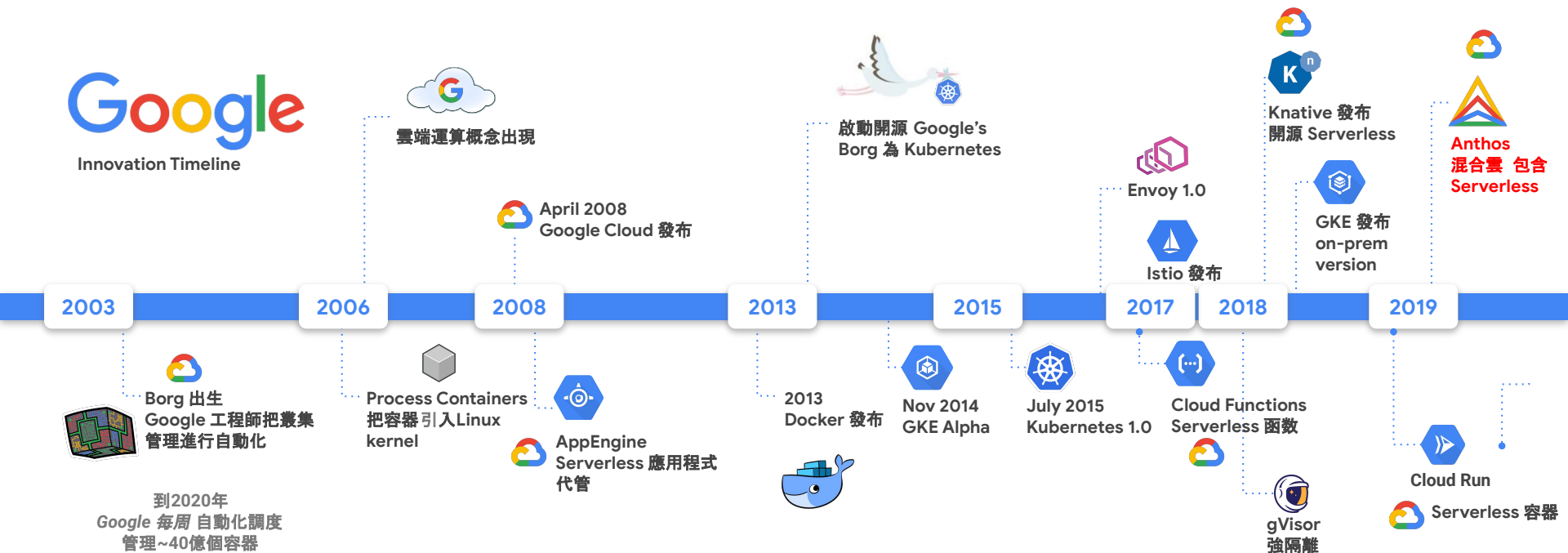
完成一個應用程式 你所需要關注的事



Google 超過 18 年的技術創新



Innovation Timeline



devfest
2022

```
" class='time talk-ended single-  
' class='talk-name'>...  
class='description'>...
```

What's new features these year

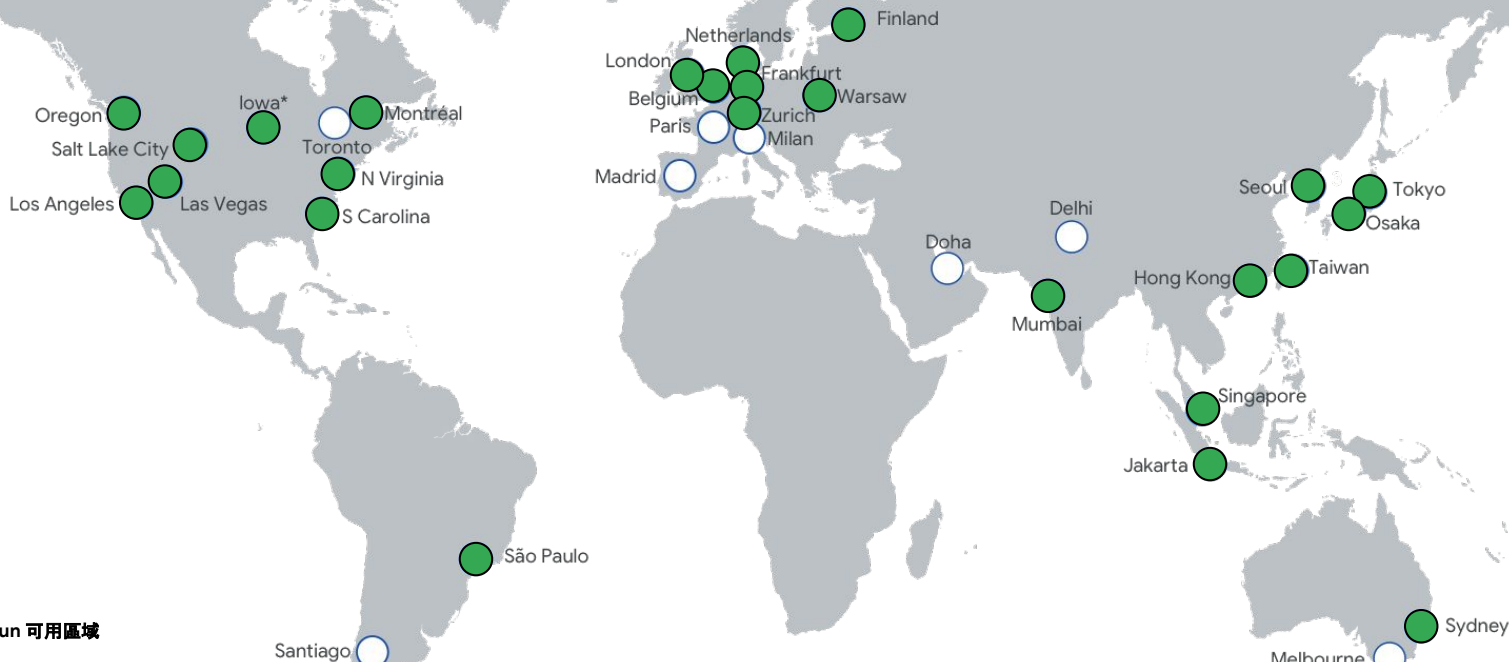


新一代 Serverless



Serverless 運算寬廣的應用程式支援能力

“我是全球業務，該怎麼分布？”



“全球都部署伺服器成本太高，也很難調度？”
用 Serverless 可以輕鬆很多

全球分布式 Serverless

全球負載平衡

通過外部 HTTP(S) 負載均衡



https://example.com

暴露一個 **global endpoint**

路由請求到就近的運算服務



https://example.com

GCP Project

Google前端

一個 IP



Cloud
Load
Balancing

美中



Cloud Run
Cloud Functions
us-central1

荷蘭



Cloud Run
Cloud Functions
eu-west1

日本



Cloud Run
Cloud Functions
asia-northeast1



Cloud CDN
Cloud Armor (anti-DDoS/WAF)

多功能性

Cloud Run 更大, 更長運行的實例

2x



記憶體 to 8GB Max and more...

2x



CPU to 4 vCPUs Max

4x



要求逾時 提升到 60 分鐘



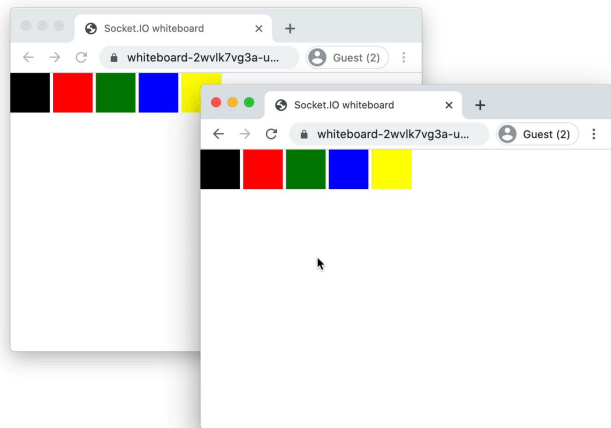
想加就“任性”加 (還會更多...)



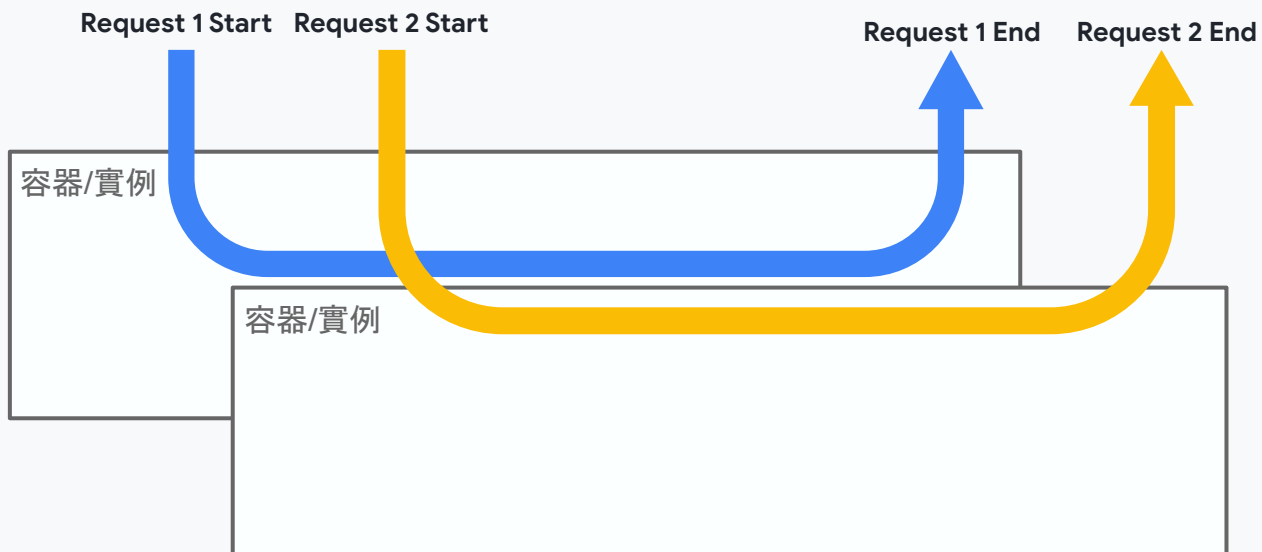
多功能性

Cloud Run WebSocket, HTTP/2 and gPRC 雙向串流服務

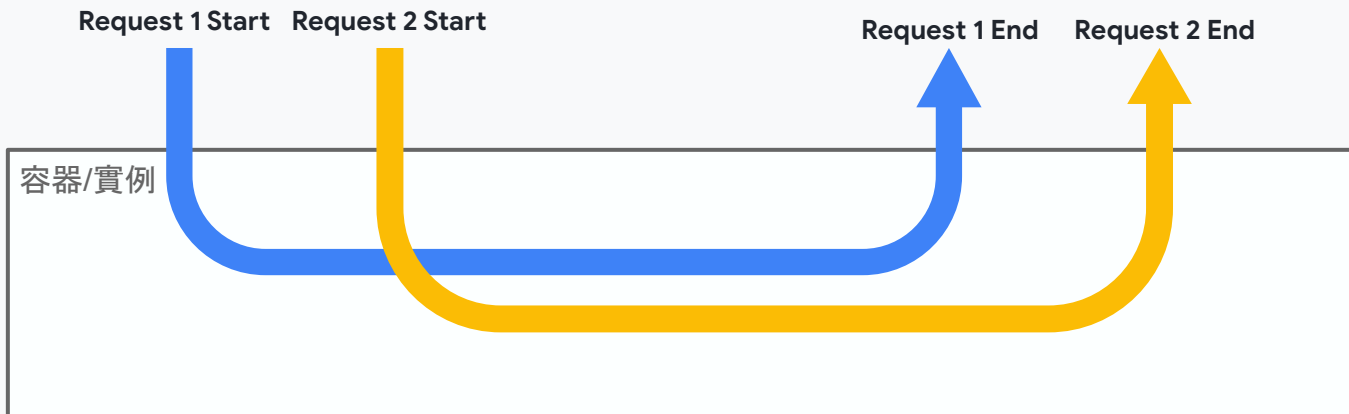
- ▶ 客戶端可以建立 WebSocket, HTTP/2, gPRC 連接到 Cloud Run 容器實例
- ▶ 每個 Cloud Run 實例可以處理最大 250 並行連接 (處理能力與你的業務有關係)
- ▶ 連接可以保持最長 1 小時
- ▶ 一個連接的生命周期內與對應的實例是相連的, 但跨連接則不保證



並行度 = 1

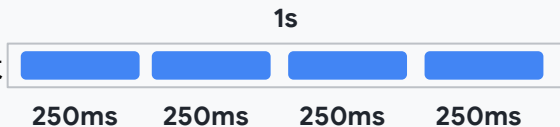


並行度 > 1 ~ 250



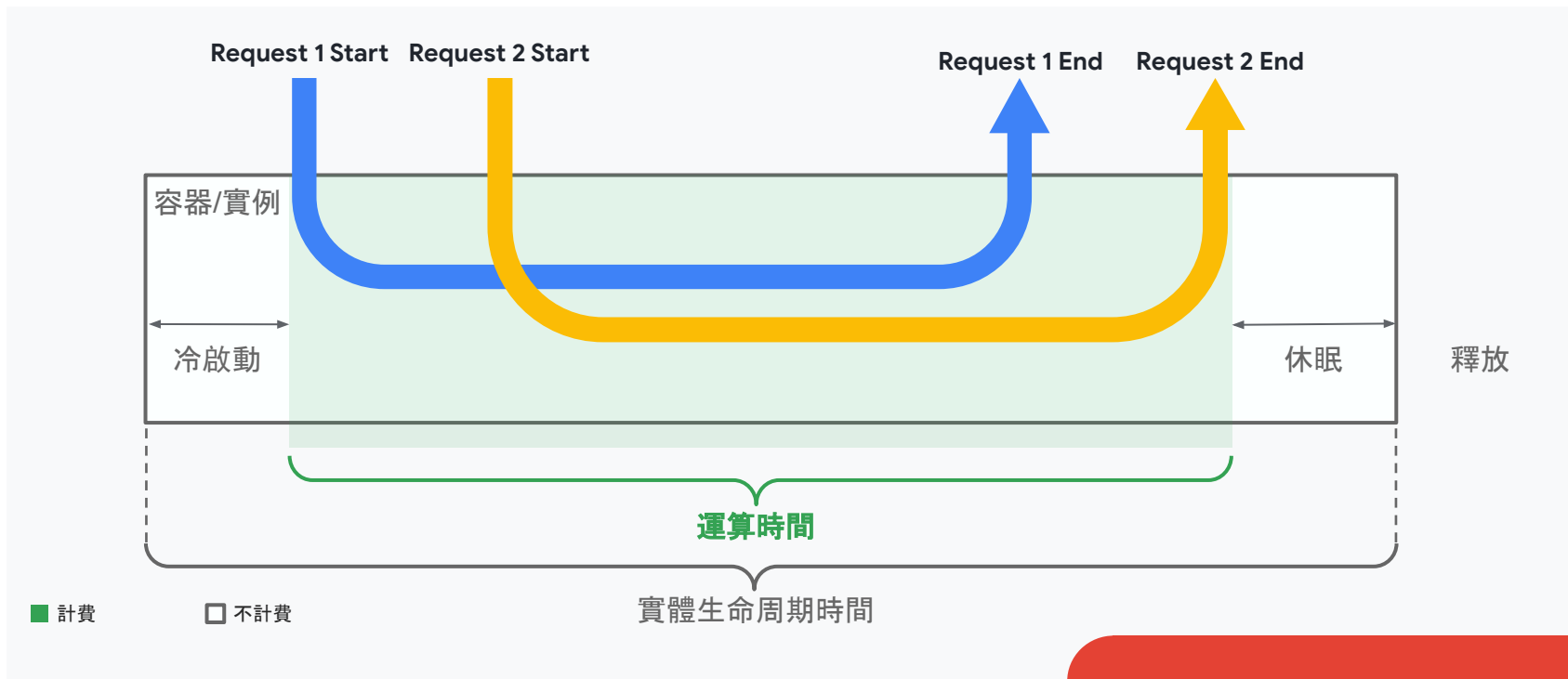
example:

250 並行/實例 x 1000 實例(默認) = 25 萬並行連接
平均每個請求 250ms, 1 個並行連接 1 秒處理 4 請求
25 萬 x 4 = 1 百萬 RPS



1 並行換算成 4 RPS

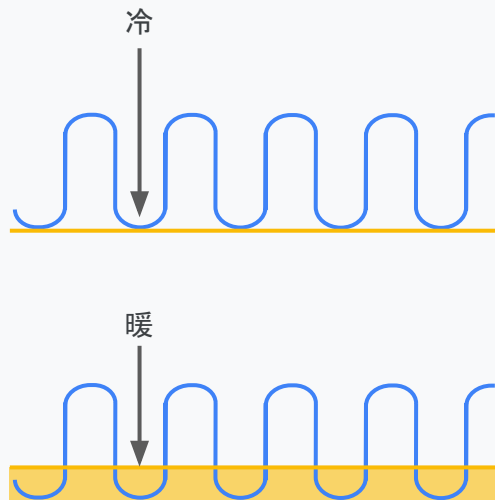
實體生命週期



多功能性

最小實例功能 : No Prewarm Needed Cloud Run & Cloud Functions

- ▶ 如果沒有流量，過一段時間之後，實例數量縮減到 0
- ▶ 此時有新流量到達，自動擴展 0=>1，此時為 冷啟動
- ▶ 可以選擇設置一定數量的最小實例，即暖實例
- ▶ 暖實例的費用 CPU 只為正常的 10% + 記憶體費用



實體生命週期與費用

Cloud Run 承諾折扣

▶ 承諾每小時的固定用量

▶ 可以享受對應用量的折扣

▶ 最少一年

▶ 17% 折扣

▶ 直接在 Cloud 控制台就可以使用

▶ 對同一個 Region 的所有 Cloud Run 服務有效



Current: Request-based

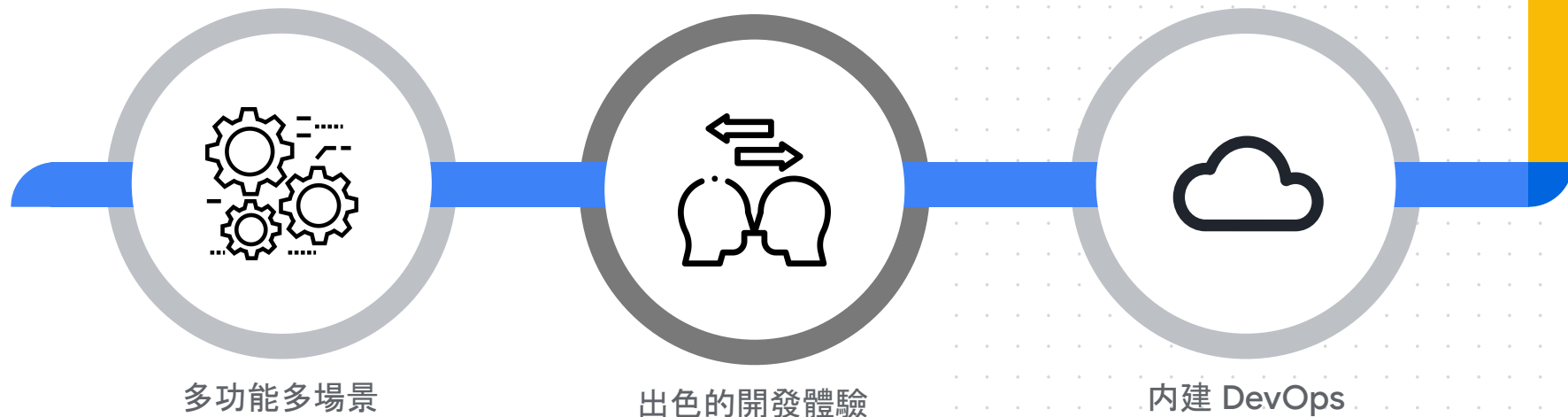


New option: Instance-based



```
$ gcloud beta run services update my-service --no-cpu-throttling
```

新一代 Serverless



Cloud Run 開發體驗

Cloud Run 拉取映像檔快取在服務中，並代管運行

但容器映像檔是怎麼來的？

需要用 Dockerfile Build 出一個容器映像檔

定期檢查容器映像檔是否有過期依賴，避免安全漏洞

好複雜 我不就想執行個應用程式嘛

現在使用 Cloud Run，你可以用原始碼直接部署，其他的讓雲端服務去做

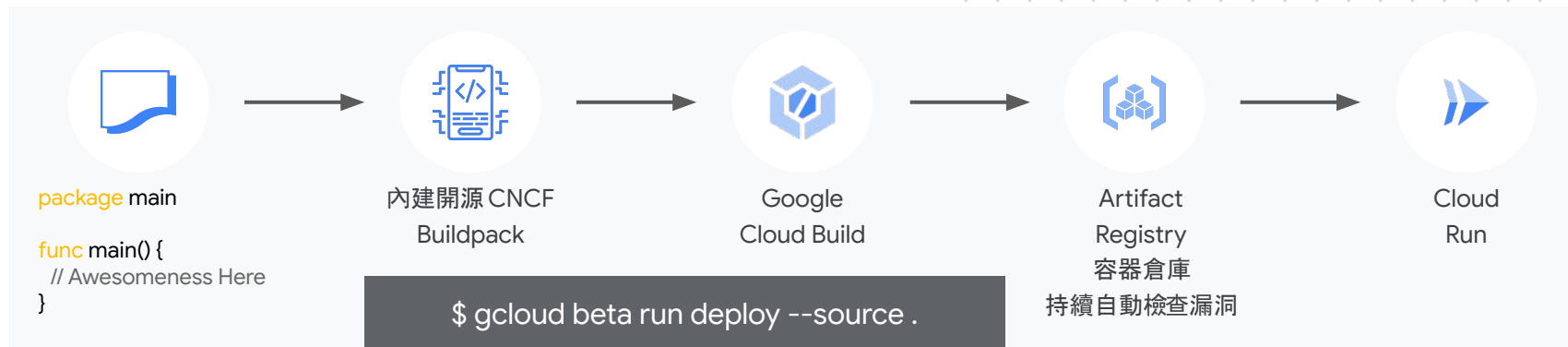


Cloud Run 開發體驗

從原始碼直接部署到 Cloud Run，無需 dockerfile

github.com/GoogleCloudPlatform/buildpacks

支援: Go 1.10+, Node.js 10+ Python 3.7+, Java 8 & 11, .NET Core 3.1+

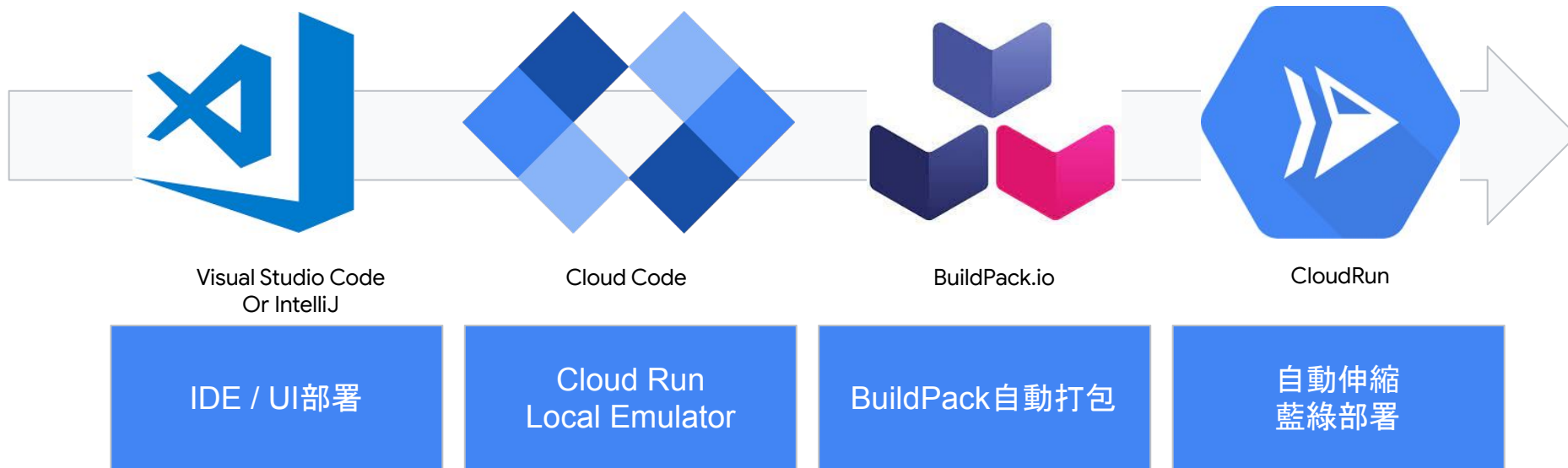


- Cloud Shell / Editor - 免費的雲端開發 (IDE) 和維運環境
- Cloud Code 在本地 IDE 中輕鬆編寫、調試和部署容器應用



開發者所見即所得：

- 只要Code + Deploy (不需要Dockerfile和YAML的開發體驗)
- 無Vendor Lock-in
- 並支援各種主流語言 (支援: Go 1.10+, Node.js 10+ Python 3.7+, Java 8 & 11, .NET Core 3.1+)
- 非常適合新手上路的開發朋友



98% 的 Cloud Run 新用戶都可以只用 5 minutes 完成從範例容器, build 到部署應用

開發體驗評估報告

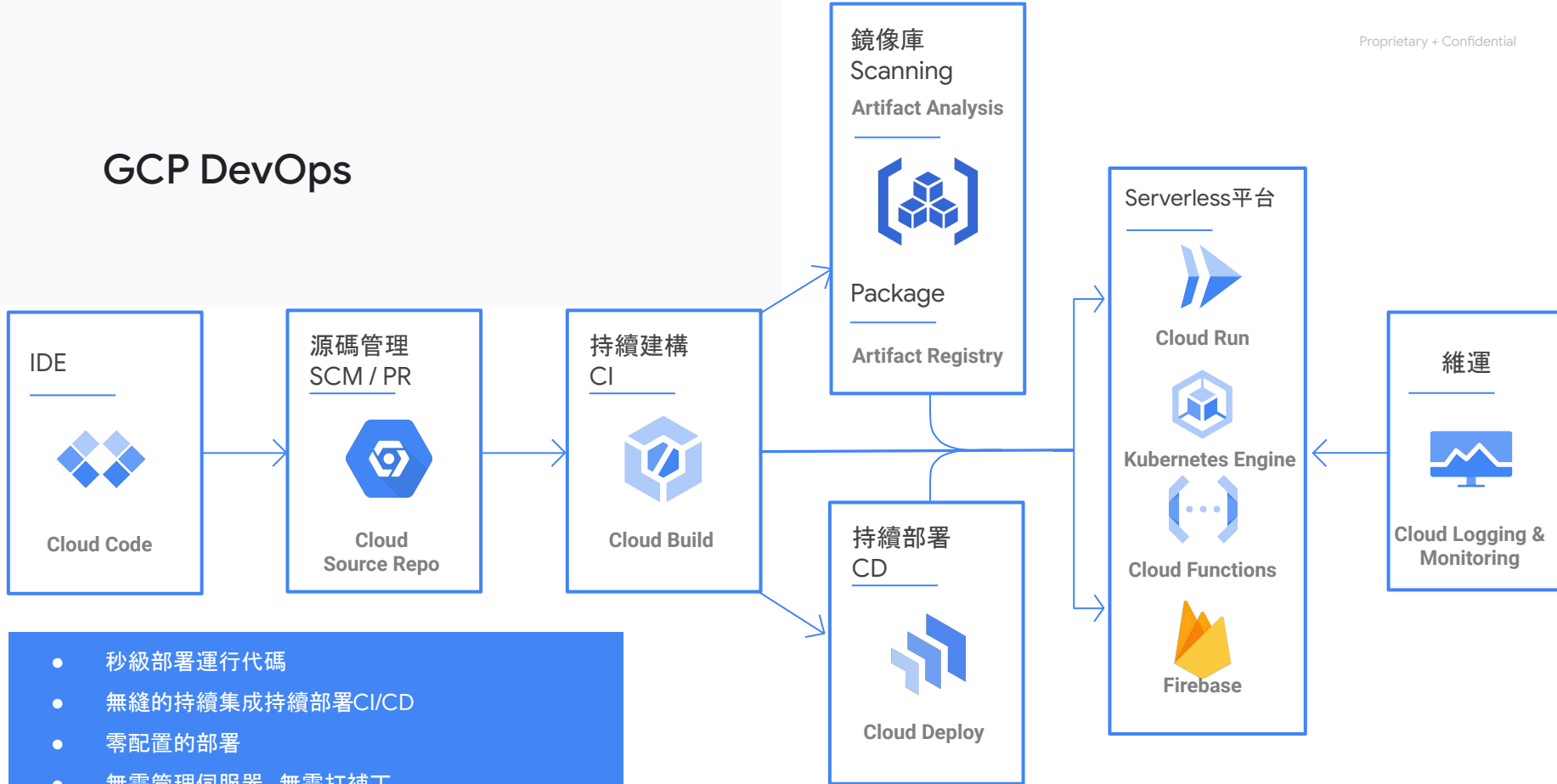
	Cloud Run	A vendor	B vendor
開發者體驗得分	85%	64%	65%
系統可用性量表評分	79	59	67
推薦指數得分	50	-13	27

*Usability Benchmark Research from User Research International Jan 2021

新一代 Serverless

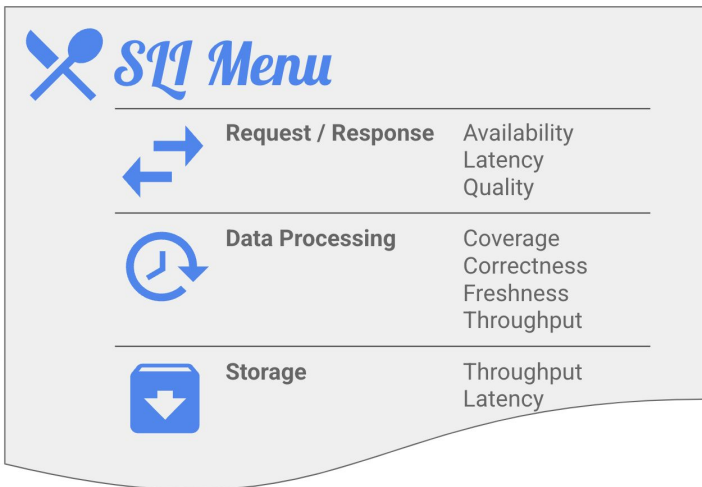


GCP DevOps



- 秒級部署運行代碼
- 無縫的持續集成持續部署CI/CD
- 零配置的部署
- 無需管理伺服器, 無需打補丁
- 自動擴展應對突發流量, 無需配置和監控
- 簽名部署

內建 SRE SLO儀表板 with CloudRun Services



Google Cloud Platform | lesser-weevil | Search Products, resources, docs (/)

Cloud Run | Services | CREATE SERVICE | MANAGE CUSTOM DOMAINS | COPY | DELETE | RELEASE NOTES | SHOW INFO PANEL

SERVICES | JOBS | PREVIEW

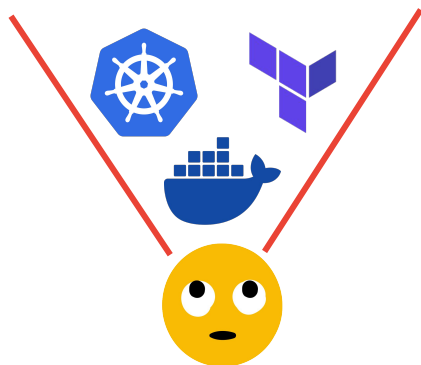
Filter Filter services

<input type="checkbox"/>	<input type="radio"/>	Name ↑	Req/sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
<input type="checkbox"/>	<input checked="" type="radio"/>	has-data	0.02	us-central1	Allow unauthenticated	All	SECURITY	Mar 17, 2022	johnbryan@google.com
<input type="checkbox"/>	<input checked="" type="radio"/>	hello	0	us-central1	Require authentication	All	SECURITY	Dec 16, 2021	natjo@google.com
<input type="checkbox"/>	<input checked="" type="radio"/>	hello-natjo	0	us-central1	Allow unauthenticated	All	SECURITY	Mar 4, 2022	natjo@google.com
<input type="checkbox"/>	<input checked="" type="radio"/>	hello2	0	us-central1	Allow unauthenticated	All	SECURITY	Mar 4, 2022	johnbryan@google.com
<input type="checkbox"/>	<input checked="" type="radio"/>	hola	0.04	us-central1	Allow unauthenticated	All	SECURITY	Mar 2, 2022	johnbryan@google.com
<input type="checkbox"/>	<input checked="" type="radio"/>	kyles-chonky-service	0	us-central1	Require authentication	All	SECURITY	Mar 4, 2022	kgrady@google.com
<input type="checkbox"/>	<input type="radio"/>	slo-availability-test	0	us-central1	Allow unauthenticated	All	SECURITY	Mar 18, 2022	johnbryan@google.com
<input type="checkbox"/>	<input checked="" type="radio"/>	testing	0	us-central1	Allow unauthenticated	All	SECURITY	Apr 8, 2022	johnbryan@google.com

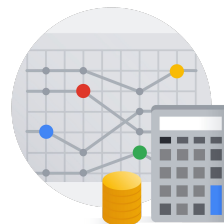
GCP Serverless的價值



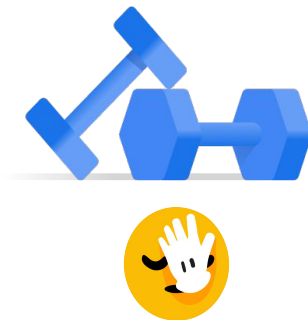
Fast Time to Market
快速交付



Coding + Deploy
兩步驟起手式



以秒計費
有用才算錢



Rapid Scaling
/Concurrency
應付突發的大量需求

devfest
2022

`class='time talk-ended single-`
`class='talk-name'>...`
`class='description'>...`

Feature - Concurrency

以前 Cloud Function 做不到，
現在(2nd gen)站在 Cloud Run 的肩膀上都行了



內建藍綠部署機制

Specify % traffic between revisions

Blue / Green deployments

Get URLs for specific revisions

```
# Gradual rollout
```

```
$ gcloud run deploy myservice \  
  --image gcr.io/project/image:f5bd774 \  
  --no-traffic \  
  --tag green
```

```
$ curl https://green---myservice-12345-us.a.run.app
```

```
$ gcloud run services update-traffic myservice \  
  --to-tags green=1
```

```
$ gcloud run services update-traffic myservice \  
  --to-tags green=10
```

```
$ gcloud run services update-traffic myservice \  
  --to-tags green=50
```

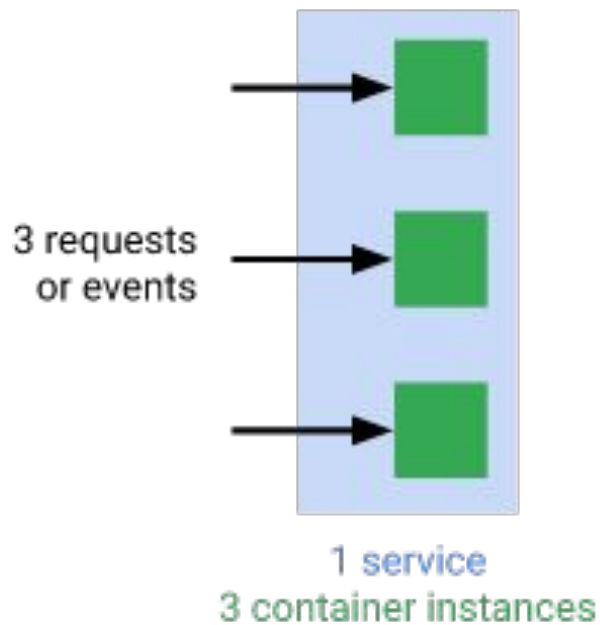
```
$ gcloud run services update-traffic myservice \  
  --to-tags green=100
```

```
# Rollback
```

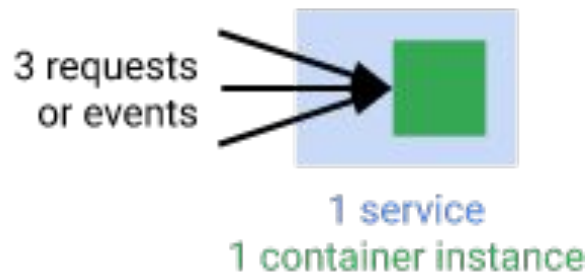
```
$ gcloud run services update-traffic myservice \  
  --to-revisions my-service-0002-joy=100
```

並行機制

concurrency = 1



concurrency = 80



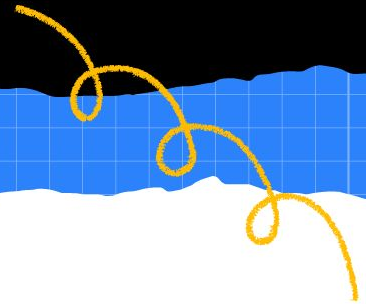


Demo Time

<https://codelabs.developers.google.com/codelabs/cloud-starting-cloudfunctions-v2>



```
on class= devsite-book-nav-toggle
-haspopup='menu' hidden='' fixed='' aria-label='Hide
navigation' data-title='Hide side navigation'
-expanded='true'><span class='material-icons
' /></span></button>
```



Thank You

