Microsoft

# Reactor

MS Learn 學堂
Azure Static Web Apps 入門

Friday 25th February | 3:30 PM HK

**Kinfey Lo**

Cloud Advocate | Microsoft

# Hello , All !

## Kinfey Lo – (盧建暉)

**Microsoft Cloud Advocate**

Former Microsoft MVP , Xamarin MVP and Microsoft RD, with more than 10 years of experience in cloud native, artificial intelligence, and mobile applications, providing application solutions for education, finance, and healthcare. Microsoft Iginte, Teched conference lecturer, Microsoft AI hackathon coach, currently at Microsoft, preaching technology and related application scenarios for technicians and different industries.

I love programming (Python , C# , TypeScript , Swift , Rust , Go )

Focus on artificial intelligence, mobile applications, cloud native

**Github :** https://github.com/lokinfey      **Open Source Project :** https://github.com/SciSharp/TensorFlow.NET

**Email :** kinfeylo@microsoft.com    **Blog :** https://blog.csdn.net/kinfey

**Twitter :** @Ljh8304

記得關注我YouTube 頻道

Microsoft
Reactor

# Azure 資源索取



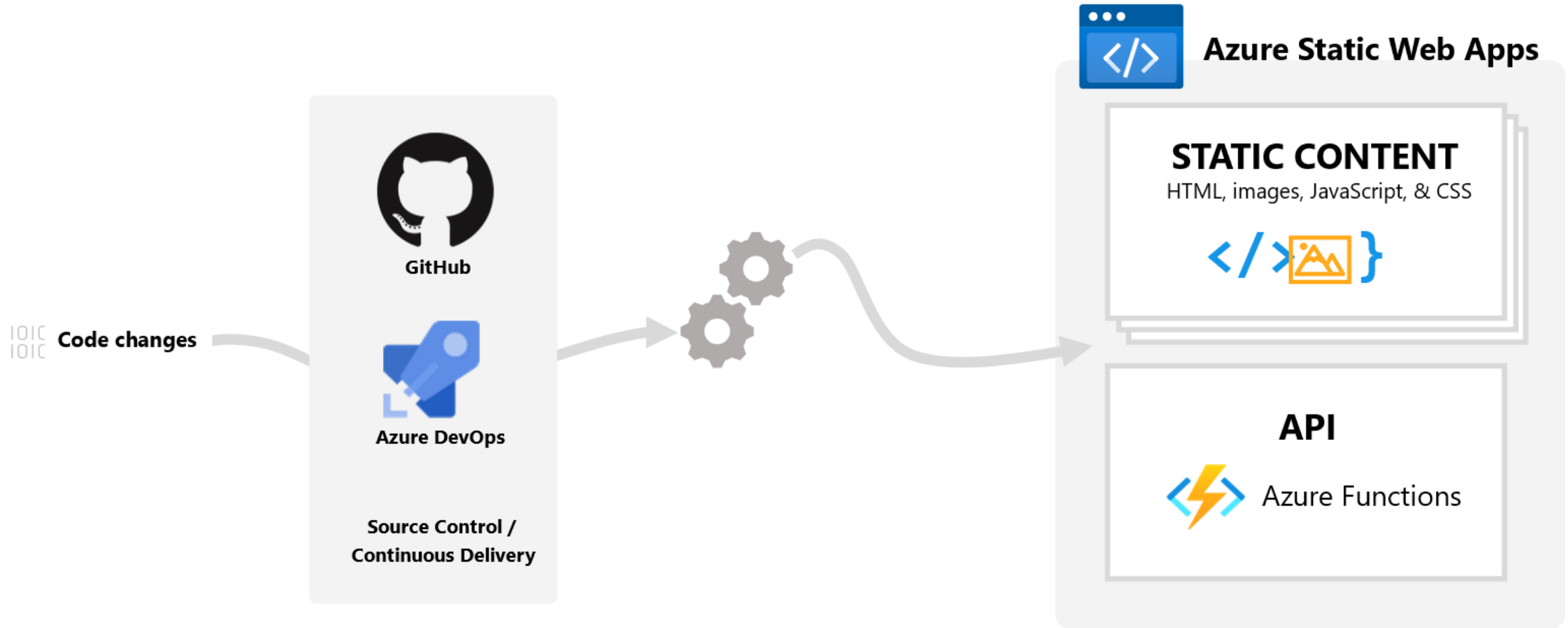Free Azure Plan  https://azure.com/free



Free Azure Student Plan  https://aka.ms/StudentGetAzure
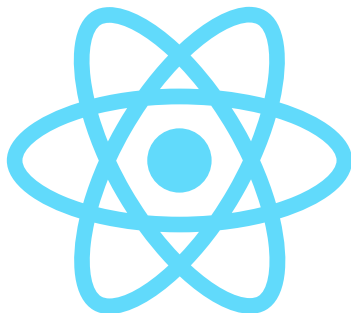
# Static web app

# Azure Static web apps

靜態 Web 應用程序通常使用不需要服務器端渲染的庫和框架（如 Angular、React、Svelte、Vue 或 Blazor）構建。

# GitHub Actions and Azure Static Web Apps



```yaml
  - name: Build And Deploy
      id: builddeploy
        uses: Azure/static-web-apps-deploy@v0.0.1-preview
        with:

azure_static_web_apps_api_token: ${{ secrets.AZURE_STATIC_WEB_APPS_API_TOKEN }}
        repo_token: ${{ secrets.GITHUB_TOKEN }}
        action: "upload"
          app_location: "azure-static-web-app/client"
          api_location: "azure-static-web-app/api"
          output_location: ".github/workflows"
```

Database          API          Application          Solution          Summary

# Azure Static Web Apps

- Globally distributed hosting

- Serverless APIs

- CI/CD integration

- Custom domains + Free SSL

- VS Code + local CLI

- Staging environments

- Built-in auth + custom routes

- Enterprise ready
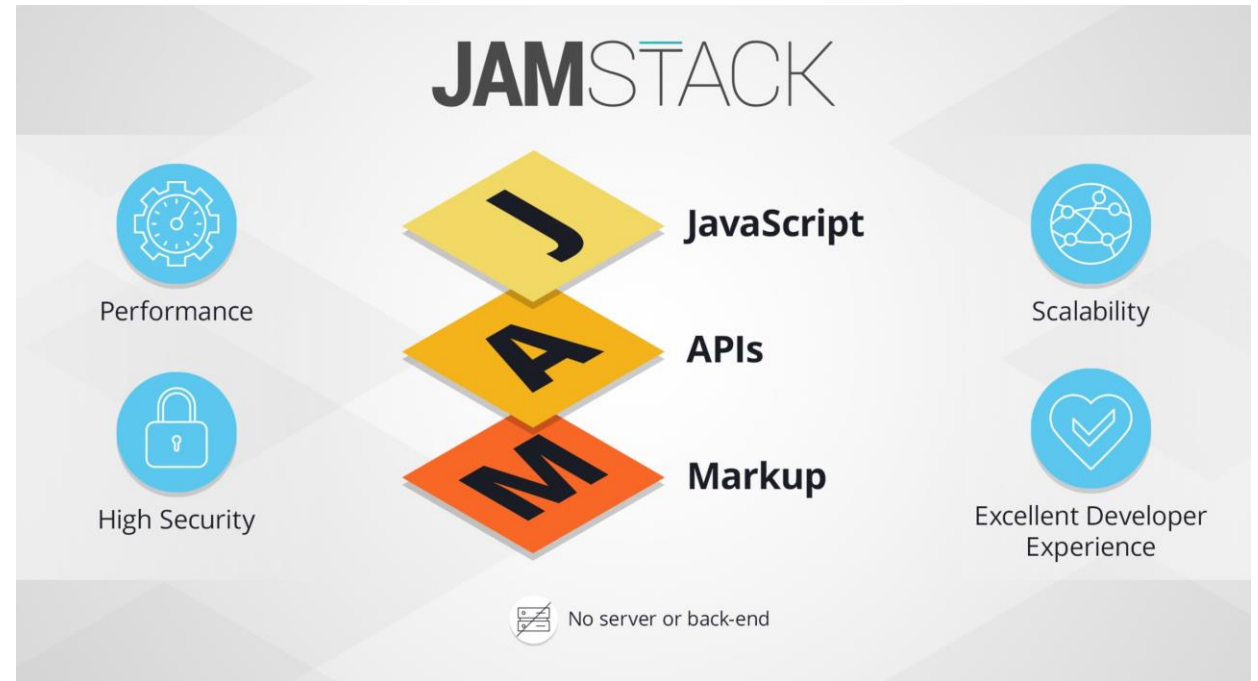
# 使用边种服務?

· Traditional web app (ASP.NET, Express.js, Django)
  **Azure App Service**

· All static content and want full control over setting up CDN, etc
  **Static website hosting on Azure Storage**

· Turnkey service for building, deploying, running Jamstack apps
  **Azure Static Web Apps**

# Plan comparison

| | Free | Standard |
|---|---|---|
| Price | Free | $9 per app per month |
| Included bandwidth | 100GB per subscription | 100GB per subscription |
| Bandwidth overage | – | $0.20 per GB |
| Custom domains | 2 | 5 |
| Custom authentication | – | ✔ |
| Bring your own Azure Functions app | – | ✔ |
| Staging environments | 3 | 10 |
| Deployment size | 250MB | 500MB |
| SLA | – | 99.95% |

# 標準計劃——自帶功能應用

- Link any function app to your static web app, accessible at /api

- Use Static Web Apps auth in function app

- Use bring your own functions app to:
    - Write functions in any language
    - Use Premium plan
    - Use non-http triggers

- Currently only supports production environment

# Standard plan – custom authentication

- Use your own app registration:
  - AAD
  - GitHub
  - Facebook
  - Google
  - Twitter
  - Apple
  - any OpenID Connect provider

- Add "auth" section in configuration

```json
{
  "auth": {
    "twitter": {
      "registration": {
        "consumerKeySettingName": "<TWITTER_CONSUMER_KEY>",
        "consumerSecretSettingName": "<TWITTER_CONSUMER_SECRET>"
      }
    },
    "identityProviders": {
      "openIdConnectProviders": {
        "myProvider": {
          "registration": {
            "clientIdSettingName": "<MY_PROVIDER_CLIENT_ID>",
            "clientCredential": {
              "secretSettingName": "<MY_PROVIDER_CLIENT_SECRET>"
            },
            "openIdConnectConfiguration": {
              "wellKnownOpenIdConfiguration":
                "https://<MY_ID_SERVER>/.well-known/openid-configuration"
            }
          },
          "login": {
            "nameClaimType":
              "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",
            "scope": [],
            "loginParameterNames": []
          }
        }
      }
    }
  }
}
```
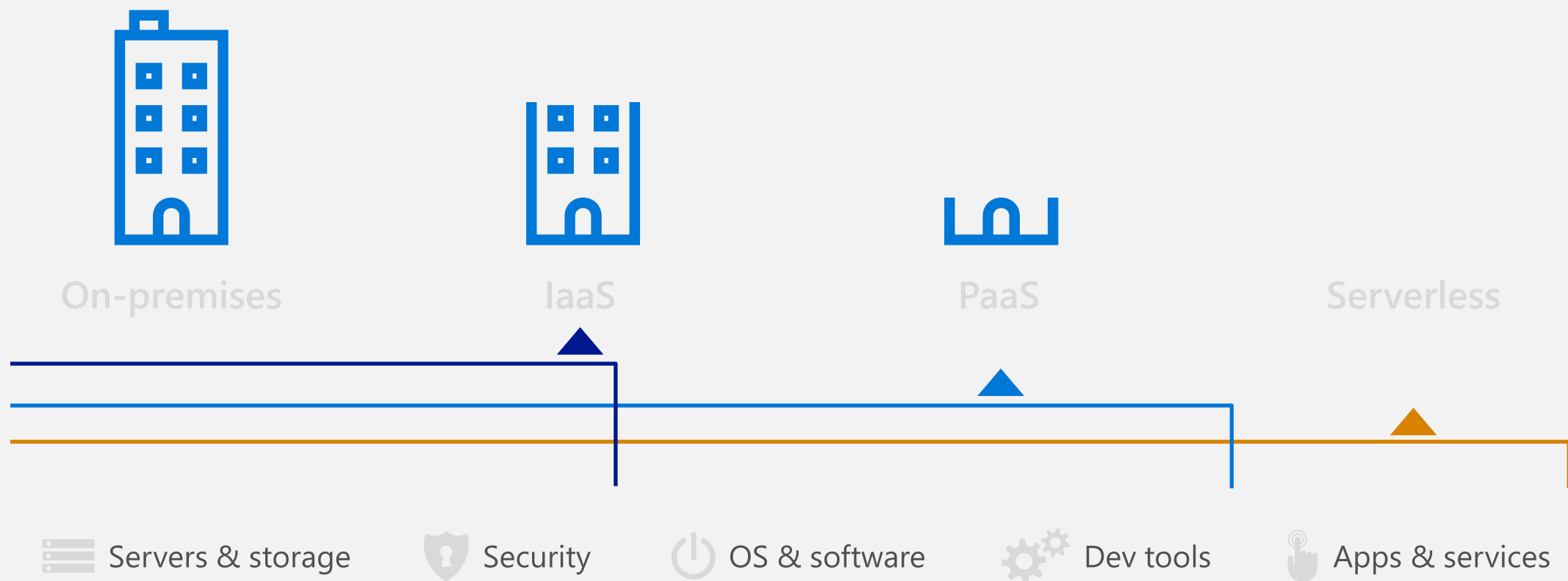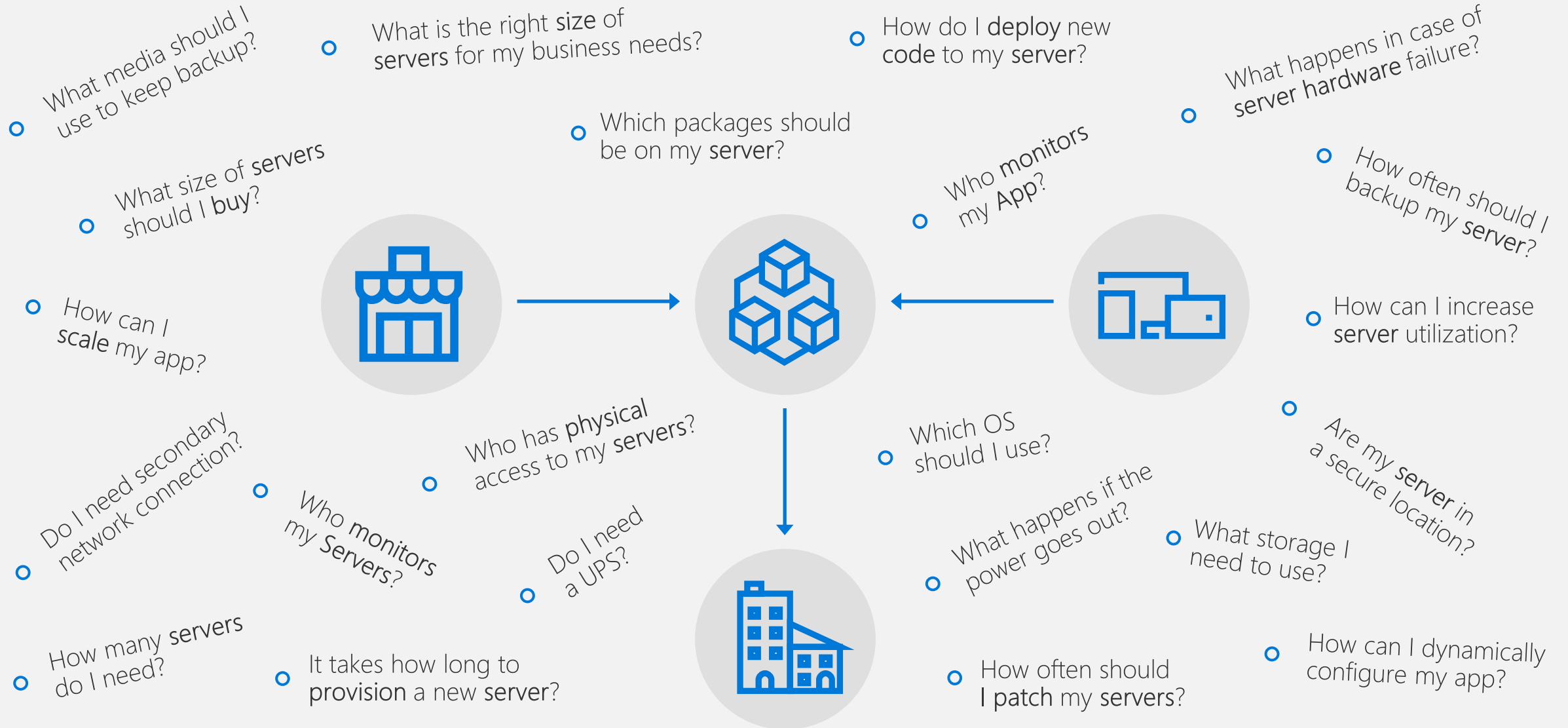
# 创建简单的static web app

Demo

# 雲發展史

越來越先進的雲技術導致公司將越來越多的 IT 活動委託給服務提供商

On-premises          IaaS          PaaS          Serverless

Servers & storage     Security     OS & software     Dev tools     Apps & services

# Before cloud...

What media should I use to keep backup?

What is the right **size** of **servers** for my business needs?

How do I **deploy** new **code** to my **server**?

What happens in case of **server hardware** failure?

Which packages should be on my **server**?

What size of **servers** should I **buy**?

Who **monitors** my **App**?

How often should I backup my **server**?

How can I **scale** my app?

How can I increase **server** utilization?

Who has **physical** access to my **servers**?

Which OS should I use?

Do I need secondary network connection?

Are my **server** in a secure location?

Who **monitors** my **Servers**?

Do I need a UPS?

What happens if the power goes out?

What storage I need to use?

How many **servers** do I need?

It takes how long to **provision** a new **server**?

How often should I **patch** my **servers**?
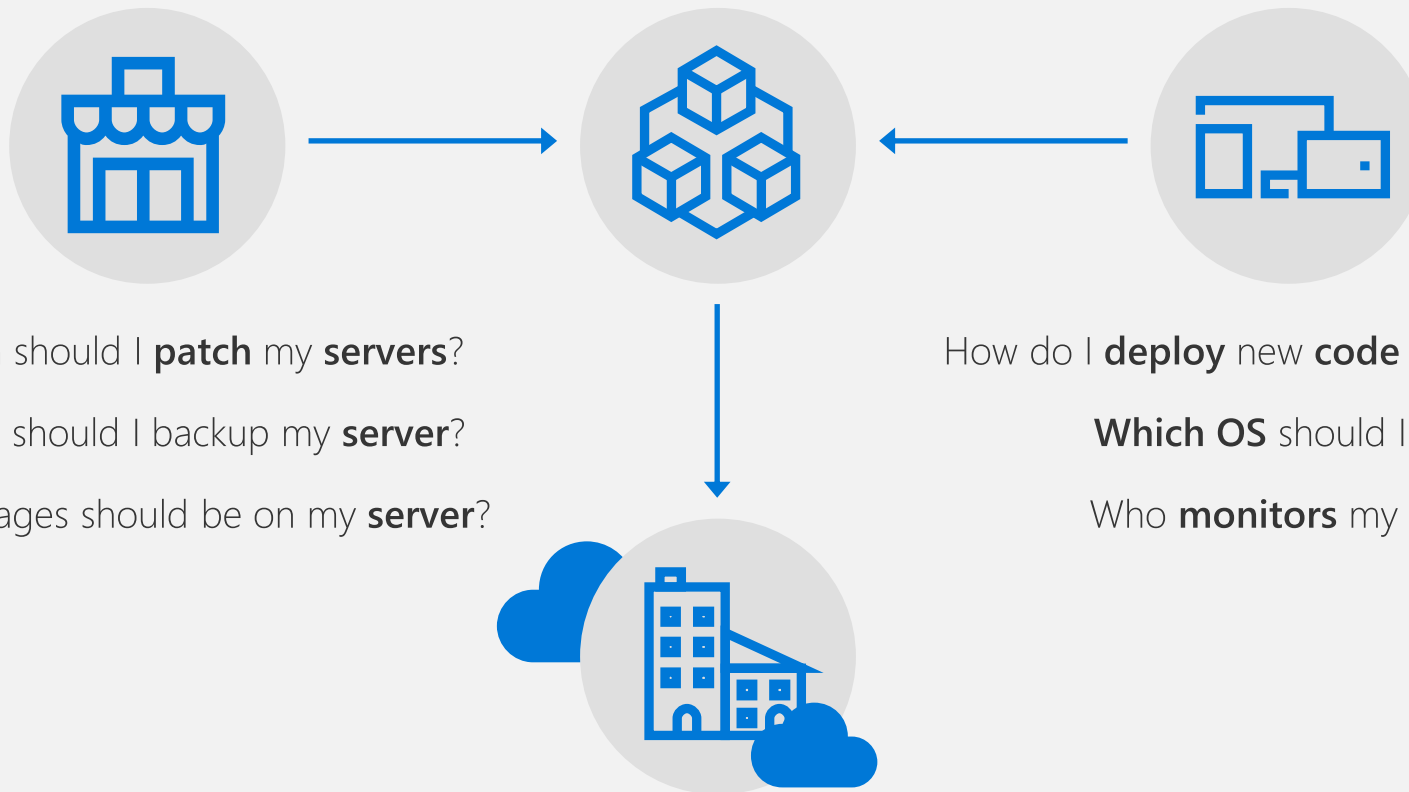
How can I dynamically configure my app?

# ......然後 IaaS 為數字業務奠定了賭注......

What is the right **size** of **servers** for my business needs?

How can I increase **server** utilization?

How many **servers** do I need?

How can I **scale** my app?



How often should I **patch** my **servers**?

How often should I backup my **server**?

Which packages should be on my **server**?

How do I **deploy** new **code** to my **server**?

**Which OS** should I use?

Who **monitors** my App?

# …然後是 PaaS，對數字化轉型至關重要

What is the right size of **"servers"** for my business needs?

How can I increase **"server"** utilization?

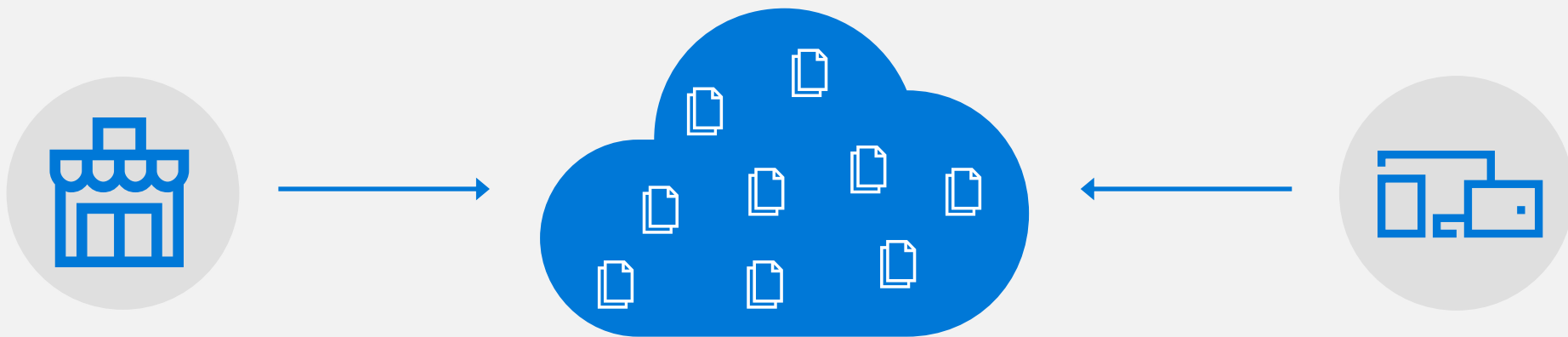How many **"servers"** do I need?

How can I **scale** my app?

# Serverless:當今下一代應用程序的平台

How do I **architect** my app to become Serverless?

# 有什麼好處?

**重點**

解決業務問題——而不是與無差別的繁重工作相關的技術問題

**效率**

更短的上市時間
固定成本轉換為可變成本
更好的服務穩定性
更好的開發和測試
管理減少浪費

**靈活性**

簡化的啟動體驗
更輕鬆的旋轉意味著更大的靈活性
更容易的實驗
按照你的節奏擴展——不要在第一天押注農場
自然適合微服務

# **What is Azure Functions?**

基於事件的無服務器計算體驗，可加速應用程序開發

- **Azure Functions = FaaS++**

- 集成編程模型
  - 使用內置觸發器和綁定來定義函數何時被調用以及它連接的數據

- 增強的開發體驗
  - 使用您喜歡的編輯器或易於使用的基於 Web 的界面（包括監控）在本地進行編碼、測試和調試

- 託管選項靈活性
  - 在不影響開發體驗的情況下選擇更適合您的業務需求的部署模型

# Azure Functions is an **open-source** project
Functions runtime and all extensions are fully open source



https://github.com/Azure/Azure-Functions

# 與 Azure 生態系統完全集成

Functions is the center piece of the Serverless platform

**Development**

**Platform**

| IDE support |
| --- |
| Integrated DevOps |
| Local development |
| Monitoring |
| Visual debug history |

### Event Grid

Manage all events that can trigger code or logic

### Functions

Execute your code based on events you specify

### Logic Apps

Design workflows and orchestrate processes

| Database | Storage | Analytics | Intelligence | Security | IoT |
| --- | --- | --- | --- | --- | --- |

# 创建简单的Azure Function

Demo

# Azure Function + Static Web App

Demo

# MS Learn Module 推薦



https://aka.ms/AzureHKStudy002

Microsoft

# Reactor

Thank You!

Q&A

Microsoft
Reactor