STUDY4 為學習而生 Build School

Azure Kubernetes Services 永續性軟體工程設計方針

Phil Huang <pichuang@cloudnative.tw>
CNCF Ambassador / Sr. Cloud Solution Architect, Microsoft Taiwan
2023/12/10

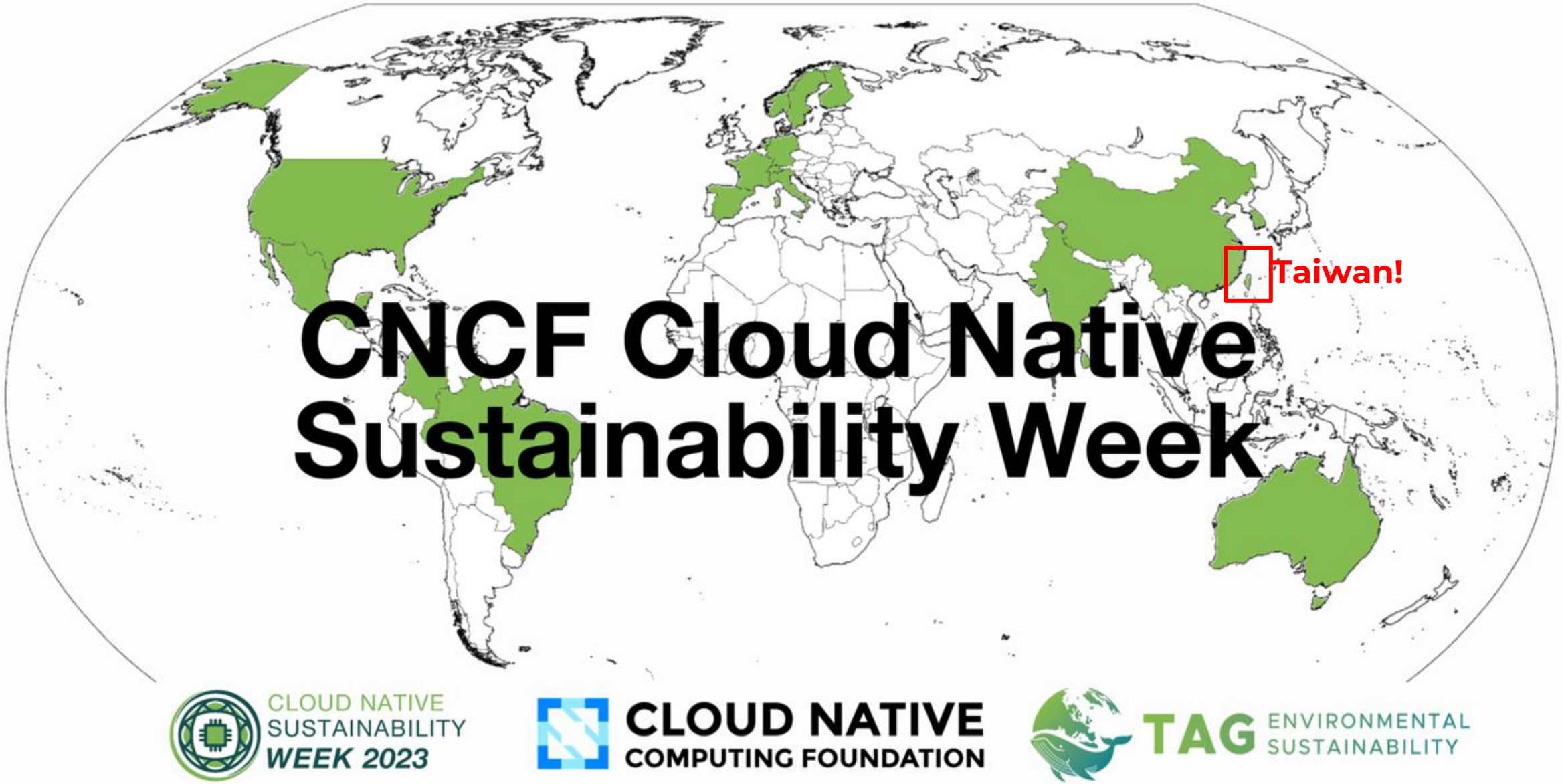
Phil Huang 黃秉鈞

- CNCF Ambassador 雲端原生基金會大使
- Cloud Native Taiwan User Group 成員
- Microsoft 資深雲端解決方案架構師
- 部落格: blog.pichuang.com.tw

"(自稱) IT 環保人員"

Ref: https://www.linkedin.com/in/pichuang-tw/


下列何者可以減少碳排放量?
多選題
A: 採用 CDN
B: 採用 Serverless
C: 使用 Cluster Autoscaler
D: 拔掉網路線



CNCF Cloud Native Sustainability Week

CLOUD NATIVE SUSTAINABILITY WEEK 2023

CLOUD NATIVE COMPUTING FOUNDATION

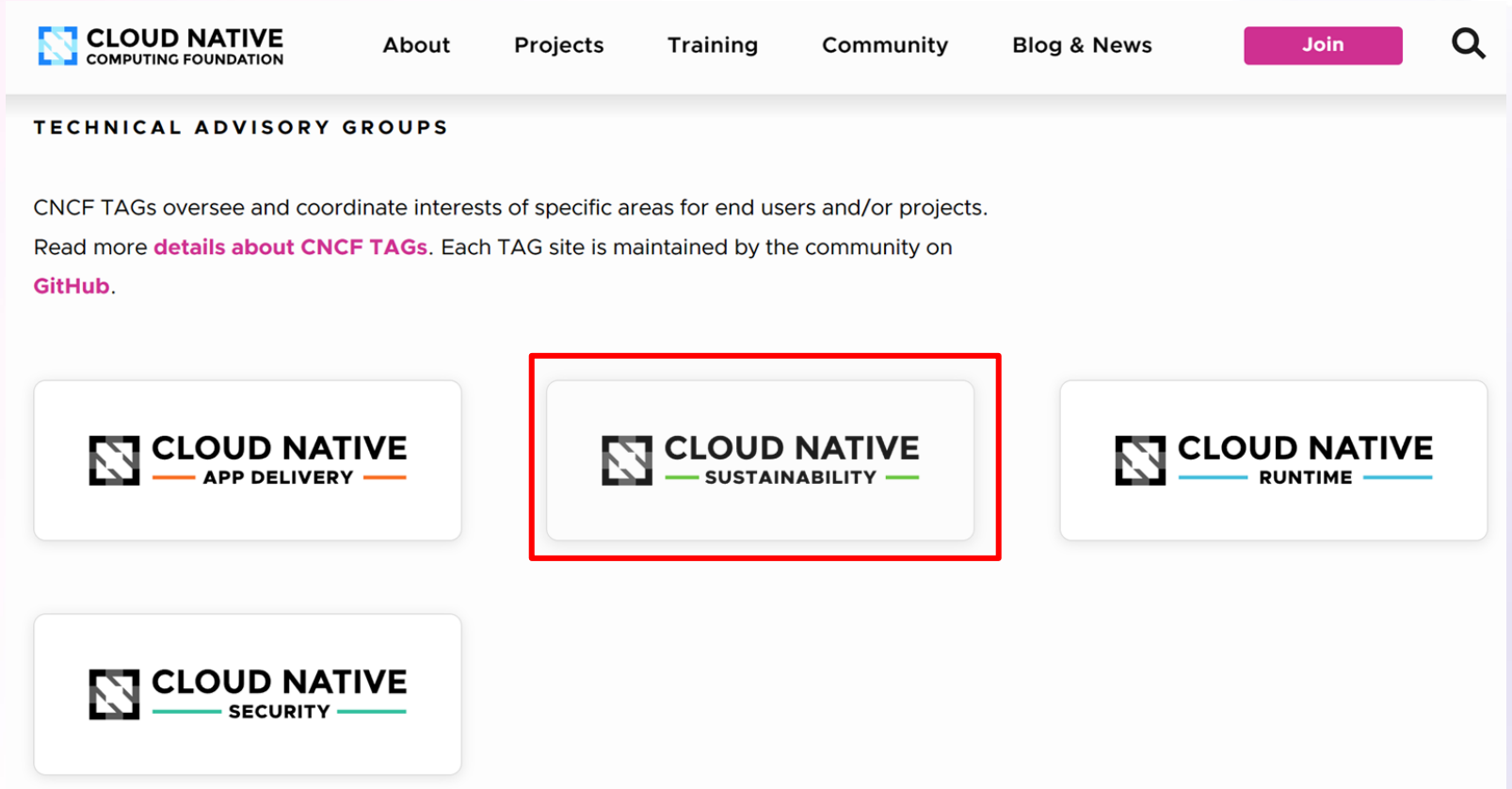
TAG ENVIRONMENTAL SUSTAINABILITY

https://tag-env-sustainability.cncf.io/cloud-native-sustainability-week/

Cloud Native Sustainability

TAG ENVIRONMENTAL SUSTAINABILITY

- 提供目前雲端原生永續發展藍圖 (Cloud Native Sustainability Landscape)
- 提高參與者的碳意識
- CNCF 基金會以 TAG Environmental Sustainability 做為提倡組織



https://tag-env-sustainability.cncf.io/

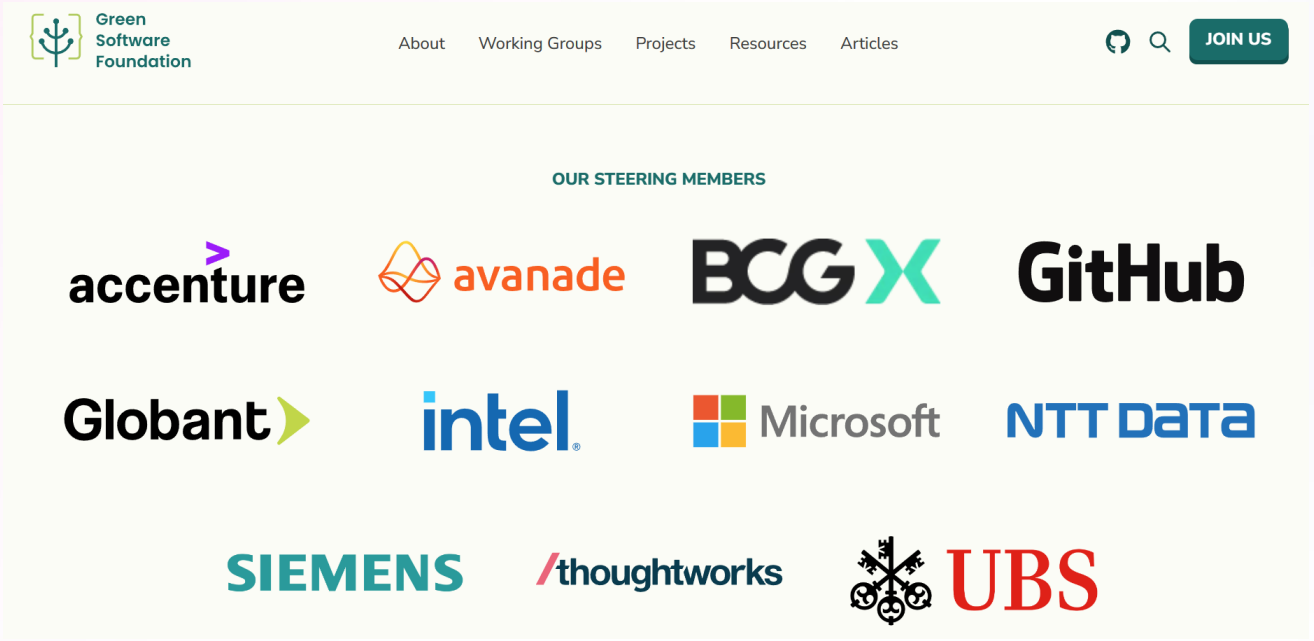
永續發展提倡單位 Sustainability Initiatives

1. **Green Software Foundation**
   - 規格: SCI (Software Carbon Intensity) Specification
   - 實踐: Green Software Patterns
   - Awesome Green Software
2. **CNCF TAG Environmental Sustainability**
3. **Linux Foundation Energy**
4. **Cloud Carbon Footprint**
5. **Energy Efficient High Performance Computing Working Group (EE HPC WG)**
6. **Open Compute Project (OCP)**
   - Heat Reuse

https://learn.greensoftware.foundation/measurement/

Green Software Foundation (GSF)

- 由 Accenture + GitHub + Microsoft + Thoughtworks 於 2021 年一同成立
- 使命: 建立由人員、標準、工具和最佳實踐組成的值得信賴的生態系統，用於新增和**建立綠色軟體 (Green Software)**



https://blogs.microsoft.com/blog/2021/05/25/accenture-github-microsoft-and-thoughtworks-launch-the-green-software-foundation-with-the-linux-foundation-to-put-sustainability-at-the-core-of-software-engineering/

Green Software 的平行宇宙

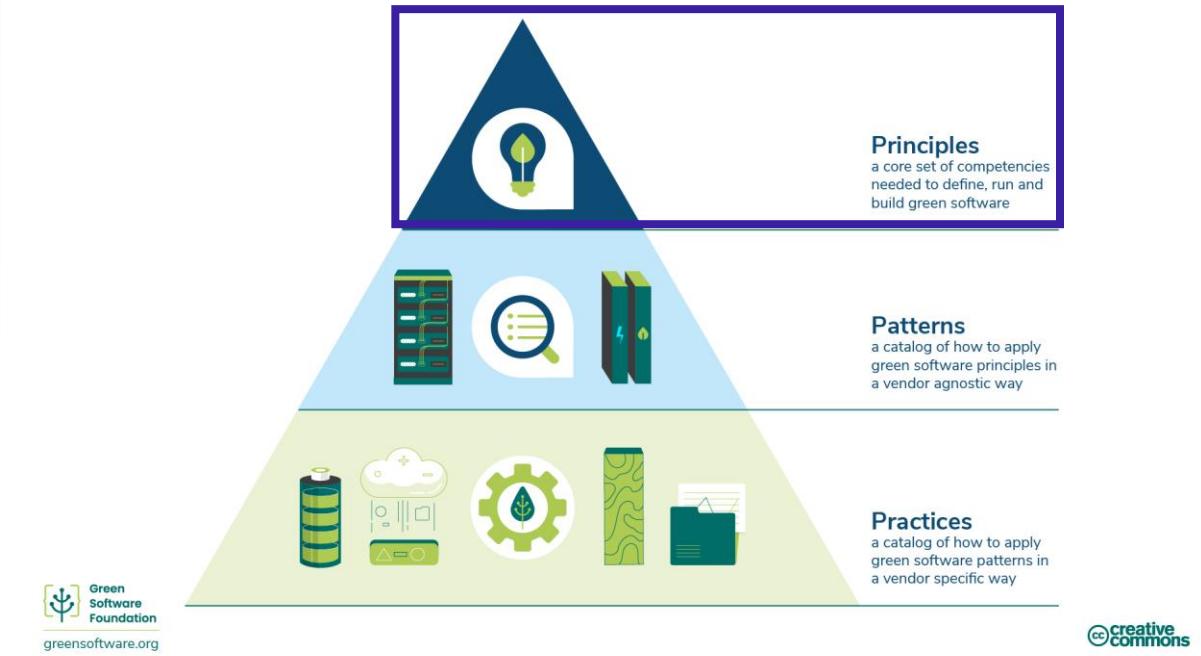
- Green Software 主要是指針對軟體運行的時候，對於環境友好和具有環境永續性
  1. 運行中減少能源消耗
  2. 提升運算效率
  3. 注重低碳排

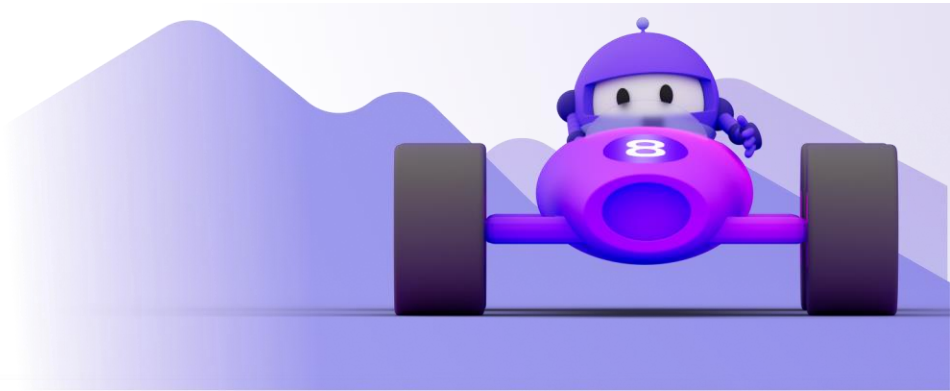
- Green Software **並不涉及**下列討論
  - 可攜式軟體
  - 免費軟體
  - 自由軟體

STUDY4 為學習而生 Build School

Green Software Principle

Green Software Foundation





Green Software Principles 綠色軟體 3 大原則

Green Software Principles

Energy Efficiency

Consume the least amount of electricity possible

節能

Hardware Efficiency

Use the least amount of embodied carbon possible

提高利用率

Carbon Awareness

Do more when the electricity is clean and less when it's dirty

提升碳意識

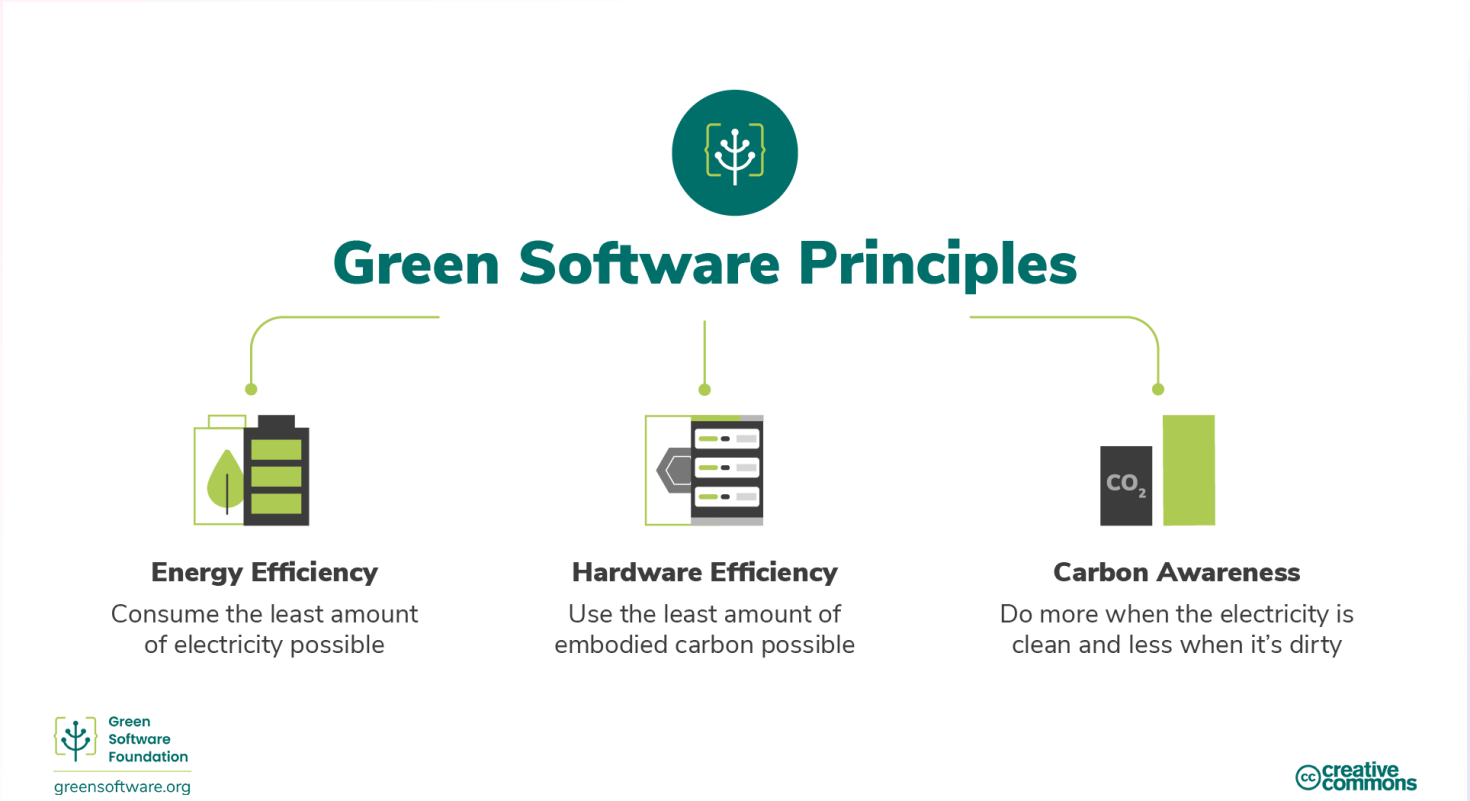
Green Software Foundation

greensoftware.org

Sustainable Software Engineering (SSE) 永續性軟體工程好處

"Sustainability is enough, all by itself, to **justify** our work"

"永續性本身就足以證明我們的工作是正確的"



- 優化程式
- 減少成本
- 提高硬體利用
- 提升硬體生命週期
- 更好預測成本
- 減少碳足跡

https://techcommunity.microsoft.com/t5/green-tech-blog/a-visual-guide-to-sustainable-software-engineering/ba-p/2130034

Software Carbon Intensity (SCI) 軟體碳排強度

$$SCI = ((E * I) + M) \text{ per } R$$

E: 軟體系統消耗的能源 (Energy)，單位是 kWh

I: 碳排強度 (Intensity)，單位是 gCO2/kWh

M: 為了運行軟體而產生的硬體碳排放量 (Embodied)

R: 功能單位 Functional Unit，如 per Seat / per API call

Green Software Foundation

https://learn.greensoftware.foundation/measurement/

Sustainable Software Engineering (SSE) 永續性軟體工程 6 個準則

$$SCI = ((E * I) + M) \text{ per } R$$

1. **碳排放效率 (Carbon Efficiency)**: 撰寫出較低碳排數值 **(E, Energy)** 的服務，但服務功能性需求 (Functional Requirement) 不變，穩定性 (Stability) 也不變
2. **碳意識 (Carbon Awareness)**: 選擇碳排較低 **(I, Intensity)** 的資料中心或機房位置
3. **用電效率 (Electricity Efficiency)**: 盡量降低硬體 **(M)** 電力消耗，如非使用需求下，降頻 CPU
4. **硬體效率 (Hardware Efficiency)**: 延長硬體 **(M)** 生命週期來分攤碳成本
5. **測量 (Measurement)**: 透過測量持續改善，採用 **SCI** 公式計算軟體碳成本
6. **氣候承諾 (Climate Commitments)**: 遵循國際共識碳中和標準 PAS 2060、淨零 Net Zero、ISO 14064 等

https://learn.microsoft.com/zh-tw/training/modules/sustainable-software-engineering-overview/

如何計算 Software Carbon Intensity 軟體碳強度?

$$SCI = ((E * I) + M) \text{ per } R$$

1. 呼叫第三方 API 資訊
2. 人工計算 CPU / Memory 消耗量

呼叫第三方 API 資訊，如 WattTime / Electricity Maps

1. 硬體可從硬體廠商或 spec.org 查
2. 雲端資訊可從第三方 API 資訊或雲端公開資訊

看 Scale out/in 計算時的單位為何，per Instance / per Pod

Green Software Foundation

https://sci-guide.greensoftware.foundation/

計算 SCI Score 範例

$$SCI = ((E * I) + M) \text{ per } R$$

1. Bound: 了解應用服務邊界，意旨你要有架構圖
2. Scale: 了解服務擴展最小單位，如 Node Pool 或 API Request
3. Measurements: 量化既有的利用率，透過既有監控軟體或第三方服務獲得
4. Report: 計算出 SCI 分數

Node	E, Energy (kwh) (monthly)	I, Emission Factor (gCo2/kwh)	E*I, Usage Emissions (gCo2)	M, Embodied Emissions (gCo2)	SCI = (E*I)+M Total Emissions (gCo2)
VM1	3.79	350.861	1,328.78	763.33	2,092.11
VM2	3.79	350.861	1,328.78	763.33	2,092.11
VM3	3.74	350.861	1,313.62	763.33	2,076.96
VM4	3.74	350.861	1,313.62	763.33	2,076.96
VM5	3.74	350.861	1,313.62	763.33	2,076.96
VM6	2.92	350.861	1,025.64	763.33	1,788.97
VM7	2.92	350.861	1,025.64	763.33	1,788.97
VM8	2.92	350.861	1,025.64	763.33	1,788.97
DB	2.38	350.861	836.17	763.33	1,599.51
Reporting	4.10	350.861	1,439.93	763.33	2,203.27
App Gateway	-		-		979.24

Total Requests in a month 890k per API Request

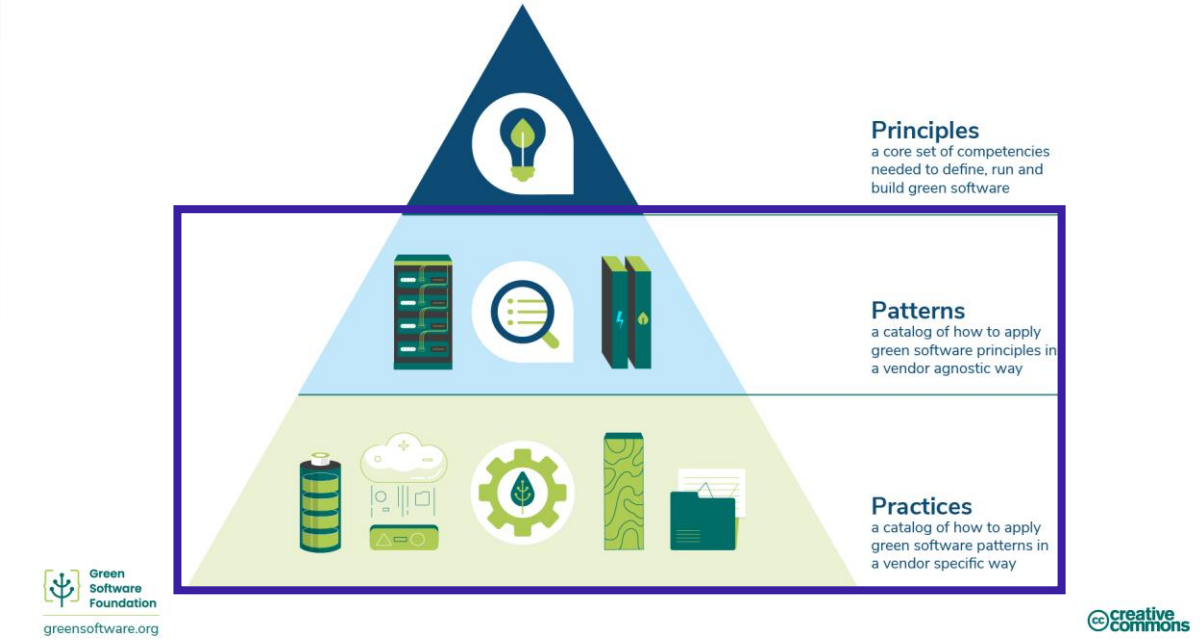
SCI Score per Request = 0.025 gCO2

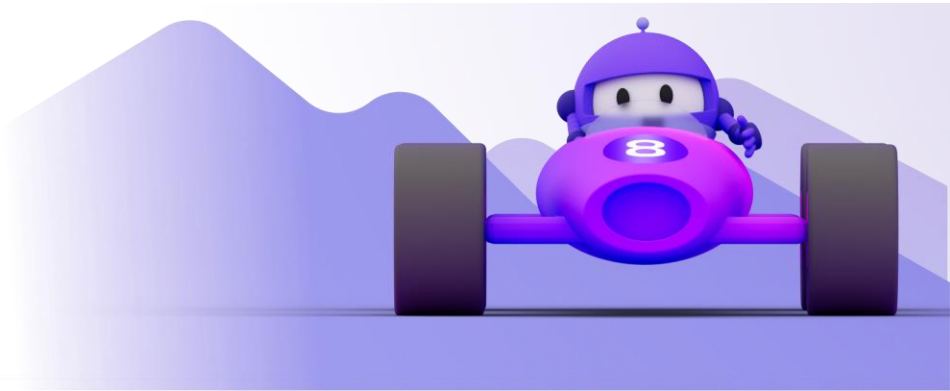
https://greensoftware.foundation/articles/how-accenture-implemented-the-sci-specification-score-to-track-software-emissions

STUDY4 為學習而生 Build School

Green Software Patterns & Practices

Green Software Foundation





為何使用 CDN 可以達到減碳的效果?

- 符合 Green Software Patterns 所列之模式 **Cache static data**
- 故採用 Azure Front Door 做為架構設計的一環是符合 GSF 建議實踐

$$SCI = ((E * I) + M) \text{ per } R$$

降低 Packet 傳輸距離，所需總電量下降

通過減少所經設備運行總量，降低整體隱藏碳排放

https://patterns.greensoftware.foundation/

在 Azure Kubernetes Service 上的永續性軟體工程做法

分類	綠色軟體設計模式	適用於 Application Workload	適用於 AKS Cluster	代表服務/專案/術語	E	I	M
應用程式設計	為邏輯元件的獨立調整而設計	Yes		Dapr Framework	⏬		⏬
	為事件驅動調整而設計	Yes		KEDA	⏬		⏬
	以無狀態設計為目標	Yes		無狀態設計	⏬		⏬
應用程式平台	啟用叢集和節點自動更新		Yes	Auto Upgrade	⏬		⏬
	安裝支援的附加元件和延伸模組	Yes	Yes	KEDA + Dapr	⏬		⏬
	在適用情況下將工作負載容器化	Yes		Draft	⏬		⏬
	使用能源效率高的硬體		Yes	Ampere Altra Arm / Karpenter	⏬		
	符合延展性需求，並利用自動調整和高載功能		Yes	Cluster Autoscaler	⏬		⏬
	在非上班時間關閉工作負載和節點集區	Yes	Yes	Nodepool Stop	⏬		⏬
作業程序	刪除未使用的資源	Yes	Yes	Image Cleaner	⏬		⏬
	標記資源	Yes	Yes	Resource Tag			
儲存體	最佳化儲存使用量	Yes	Yes	Storage Type	⏬		⏬
網路與連線能力	選擇最接近使用者的區域		Yes	Azure Region	⏬		⏬
	減少節點之間的網路周遊		Yes	Proximity Placement Group	⏬		⏬
	使用服務網格進行評估		Yes	Service Mesh	⏬		⏬
	最佳化記錄收集	Yes	Yes	Reduce Log Size	⏬		⏬
	快取靜態資料	Yes	Yes	Azure Front Door	⏬		⏬
安全性	評估是否要使用 TLS 終止	Yes	Yes	Azure Front Door / Application Gateway	⏬		⏬
	使用雲端原生網路安全性工具和控制項	Yes	Yes	Web Application Firewall (WAF)	⏬		⏬
	掃描弱點	Yes	Yes	Microsoft Defender for Cloud			⏬

https://learn.microsoft.com/zh-tw/azure/aks/concepts-sustainable-software-engineering

網路與連線能力: 最佳化紀錄收集

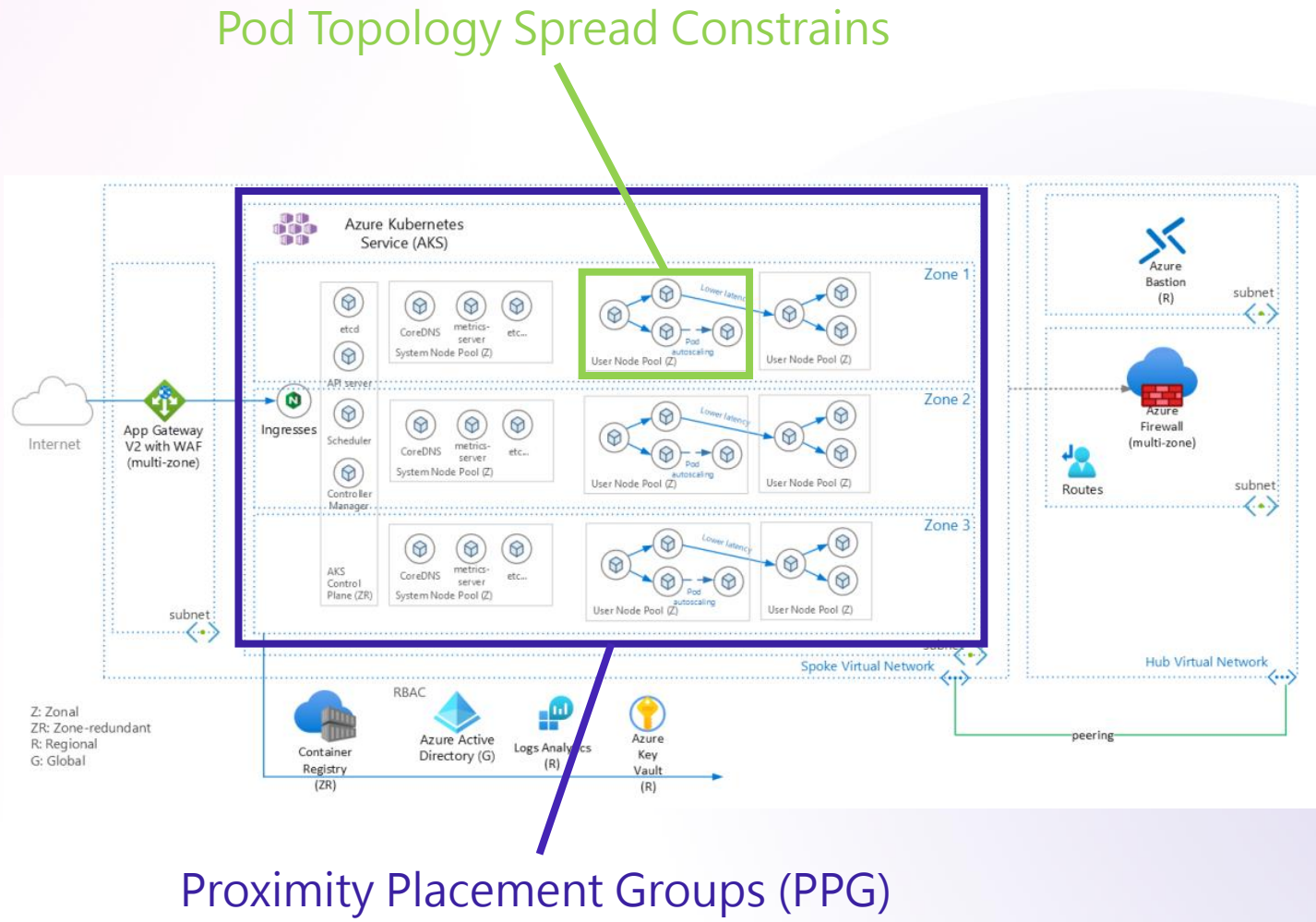
- 主要目的
  - 減少收集無用的資訊
  - 功能保持不變
- 精算封包的狀況下，縮減 83% 儲存和網路傳輸成本，
  389 Bytes -> 64 Bytes

"一起當 IT 客家人"

https://techcommunity.microsoft.com/t5/azure-network-security-blog/optimizing-azure-firewall-logging-costs/ba-p/3908958

網路與連線能力: 減少節點之間的網路周遊

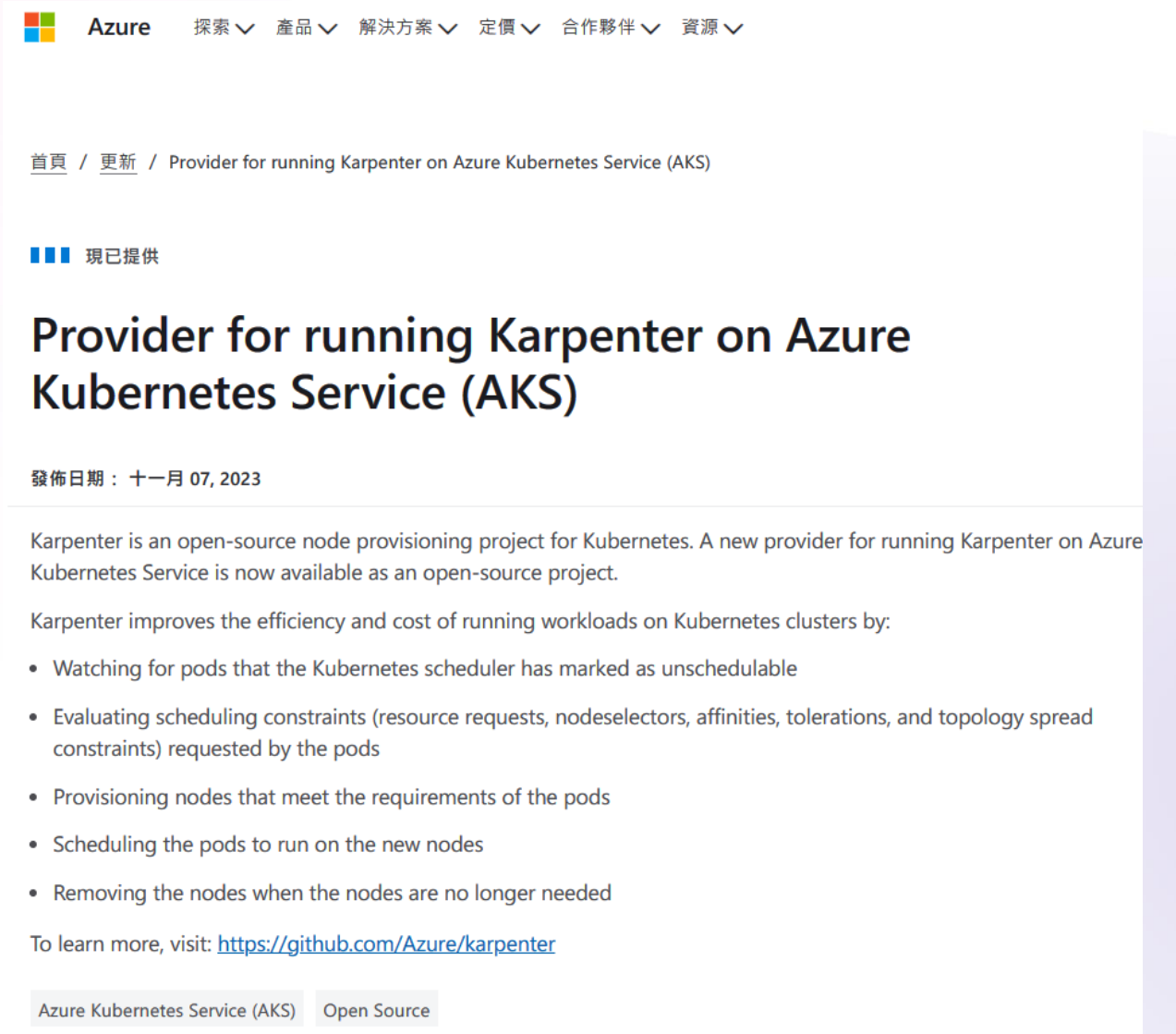
- Azure Kubernetes Services 的 Node Pool 是採用 VMSS (Virtual Machine Scale Set) 的類型
- 服務放置策略的改善方式
  1. Cluster Level: AKS 可用 Proximity Placement Groups (PPG)
  2. Pod Level: Kubernetes 可搭配使用 Pod 拓撲分布約束 (Pod Topology Spread Constraints)



https://learn.microsoft.com/en-us/azure/aks/reduce-latency-ppg
https://kubernetes.io/docs/concepts/scheduling-eviction/topology-spread-constraints/

應用程式平台: 使用能源效率高的硬體

- AKS Kapenter
- 值得關注能力
  - 類同 Cluster Autoscaler，可以自動縮容 Instances Unit
  - 比 Cluster Autoscaler 更好的是會可以動態選擇 Instance Type



Azure 探索 產品 解決方案 定價 合作夥伴 資源

首頁 / 更新 / Provider for running Karpenter on Azure Kubernetes Service (AKS)

現已提供

Provider for running Karpenter on Azure Kubernetes Service (AKS)

發佈日期： 十一月 07, 2023

Karpenter is an open-source node provisioning project for Kubernetes. A new provider for running Karpenter on Azure Kubernetes Service is now available as an open-source project.

Karpenter improves the efficiency and cost of running workloads on Kubernetes clusters by:

- Watching for pods that the Kubernetes scheduler has marked as unschedulable
- Evaluating scheduling constraints (resource requests, nodeselectors, affinities, tolerations, and topology spread constraints) requested by the pods
- Provisioning nodes that meet the requirements of the pods
- Scheduling the pods to run on the new nodes
- Removing the nodes when the nodes are no longer needed

To learn more, visit: https://github.com/Azure/karpenter

Azure Kubernetes Service (AKS) Open Source

https://azure.microsoft.com/en-us/updates/provider-for-running-karpenter-on-azure-kubernetes-service-aks/

應用程式平台: 符合延展性需求，並利用自動調整和高載功能

$$SCI = ((E * I) + M) \text{ per } R$$

API Request

```
curl --request POST --url
https://beta4.api.climatiq.io/compute/azure/instance \
--header 'Authorization: Bearer API_KEY' \
--data '{
  "region": "uk_west",
  "instance": "h8",
  "duration": 24,
  "duration_unit": "h"
}'
```

AKS Nodepool 擴縮容時，單位為 Node，故 3 台同時運行的時候，R 為 per 3 Node

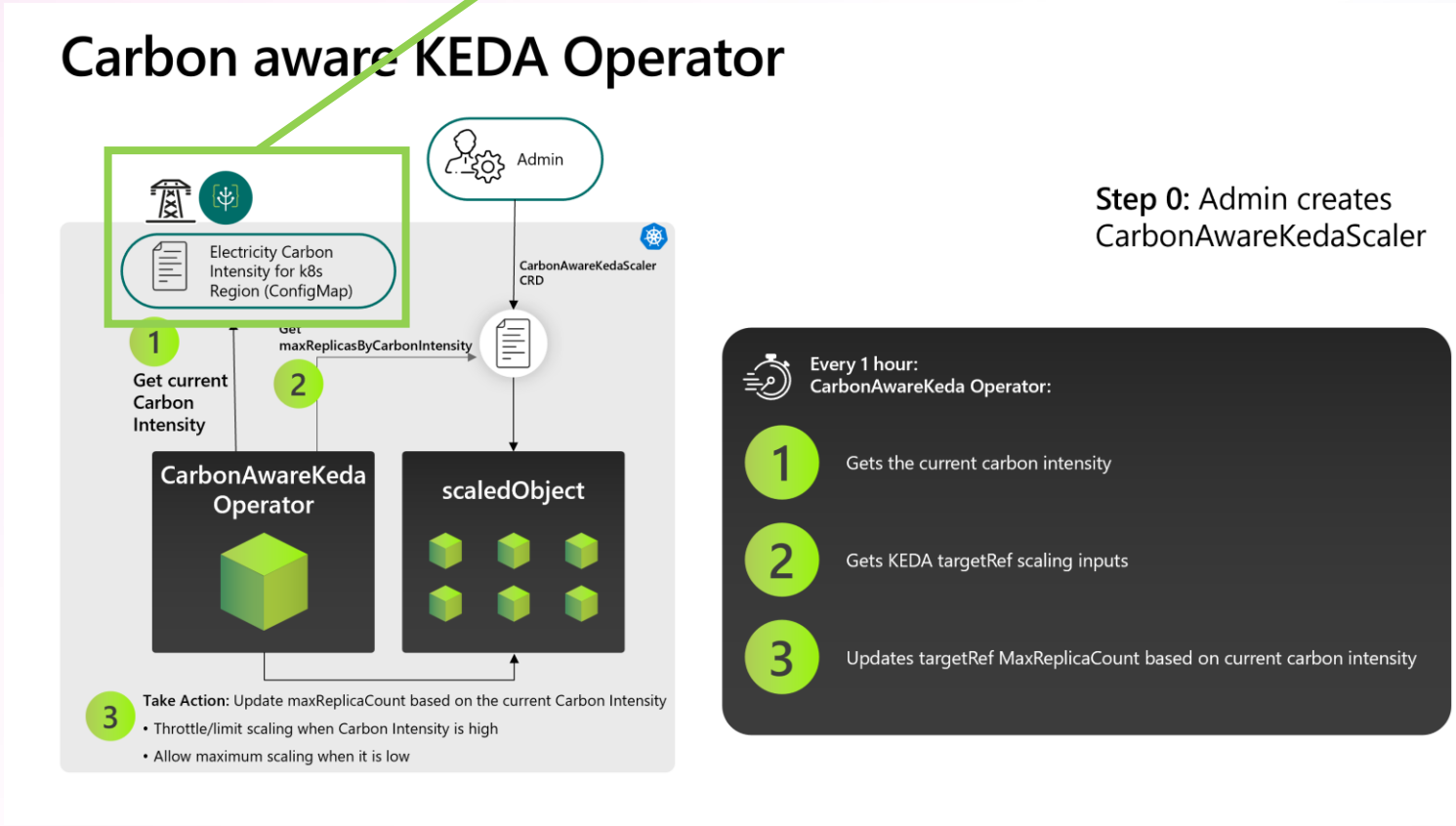
API Response


```
{
    "total_co2e": 0.7437,
    "total_co2e_unit": "kg",
    "memory_estimate": {
        "co2e": 0.1382,
        "co2e_unit": "kg",
        "co2e_calculation_method": "ar5",
        "co2e_calculation_origin": "source",
        "emission_factor": {
            "name": "Electricity supplied from grid",
            "activity_id": "electricity-supply_grid-source_supplier_mix",
```

Carbon Aware KEDA Operator

$$SCI = ((E * I) + M) \text{ per } R$$

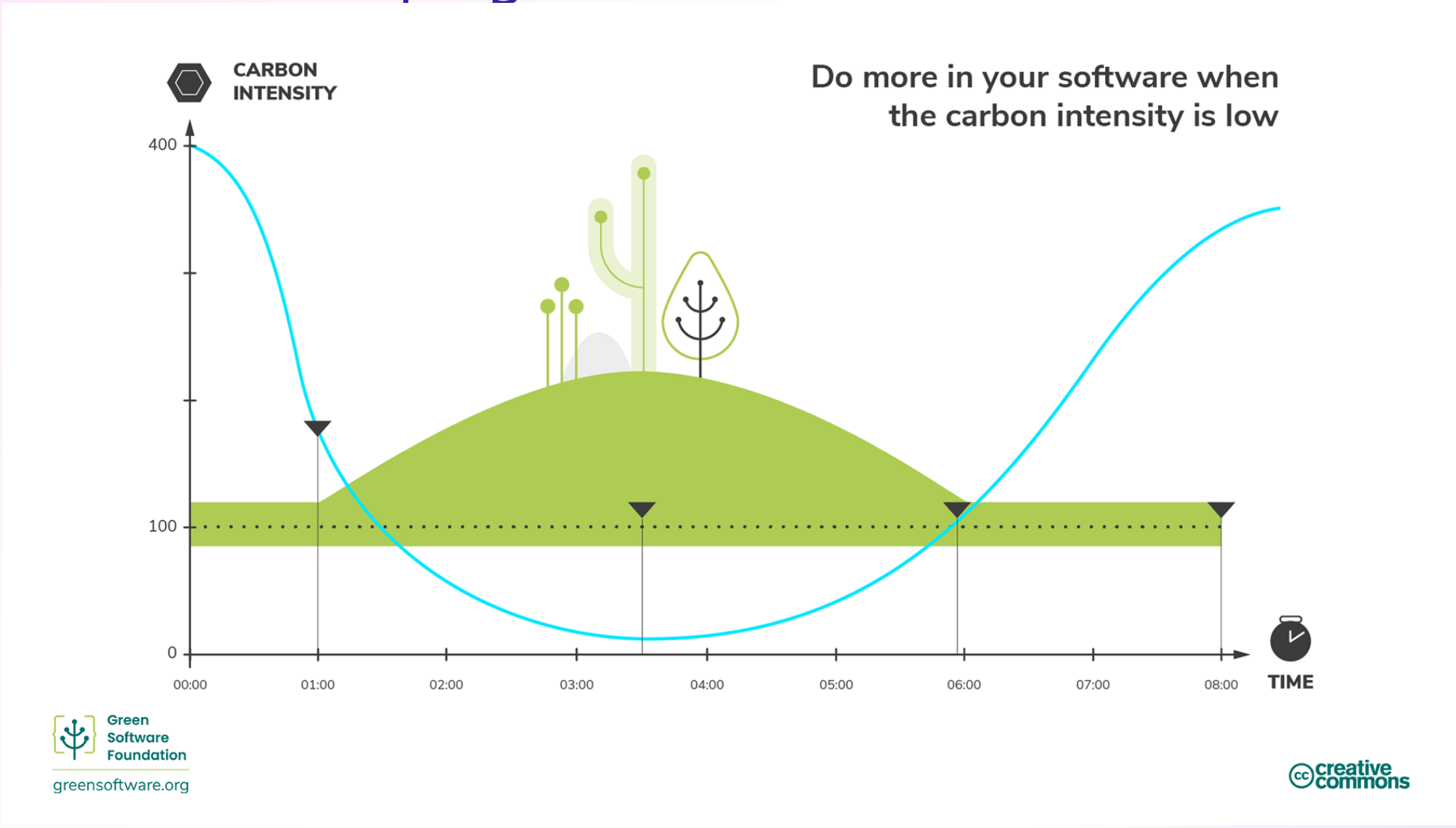
I: 碳排強度 (Intensity)，單位是 gCO2/kWh



https://github.com/Azure/carbon-aware-keda-operator

1. 形成需求 Demand Shaping

- 如果碳排強度低，提升運算需求
- 如果碳排強度高，降低運算需求



https://learn.greensoftware.foundation/carbon-awareness/

ELECTRICITY MAPS

FAQ We're hiring! Open Source Blog Get our data

台灣

總計 2023年11月

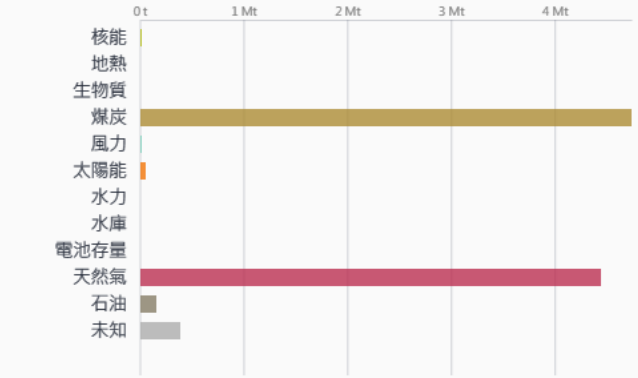
514 g 碳強度 gCO_2eq/kWh

19% 低碳能源

12% 可再生能源

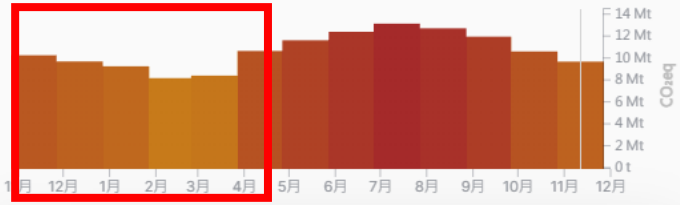
用電量 碳排放量

Total carbon emissions by source



Emissions in the last 12 months

Get hourly historical, live, and forecast data with Electricity Maps API



顯示過去資料 2023年11月

24 小時 30 天 12 個月 6 年

11月 12月 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 11月

產出 消耗

國家 區域

台灣

2023年11月

514 g 碳強度

19% 低碳能源

12% 可再生能源

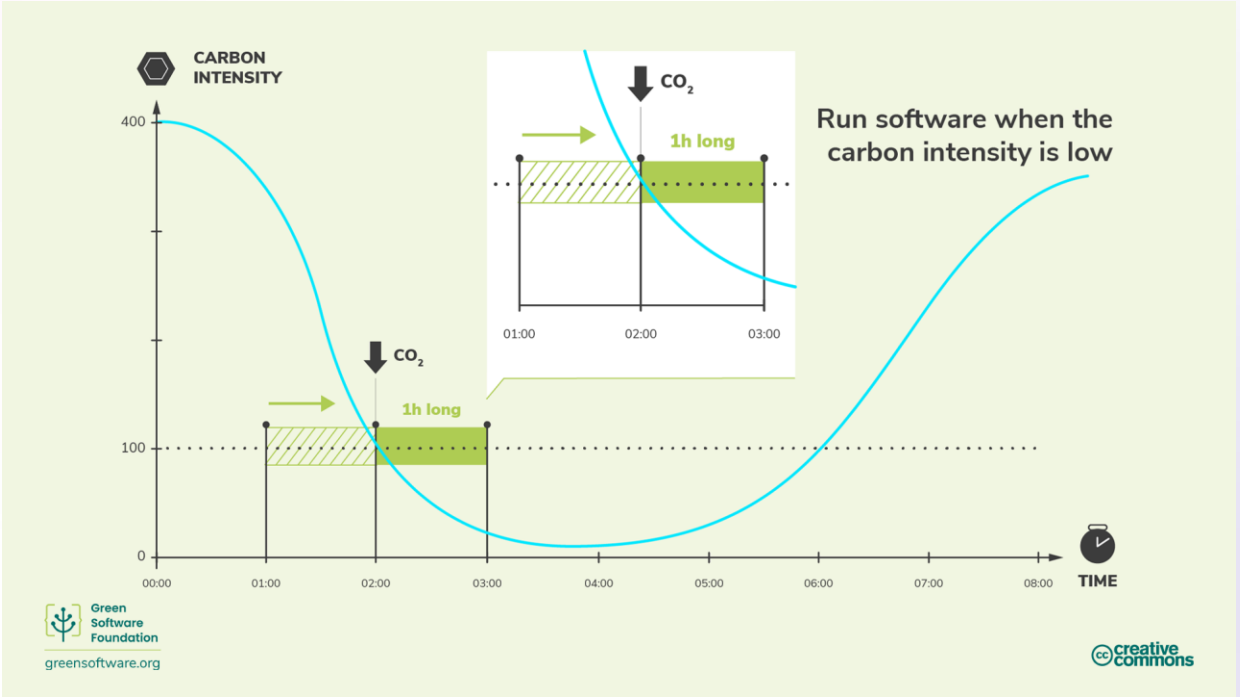
碳強度 (gCO_2eq/kWh)

0 300 600 900 1200 1500

2. 需求轉移 Demand Shifting

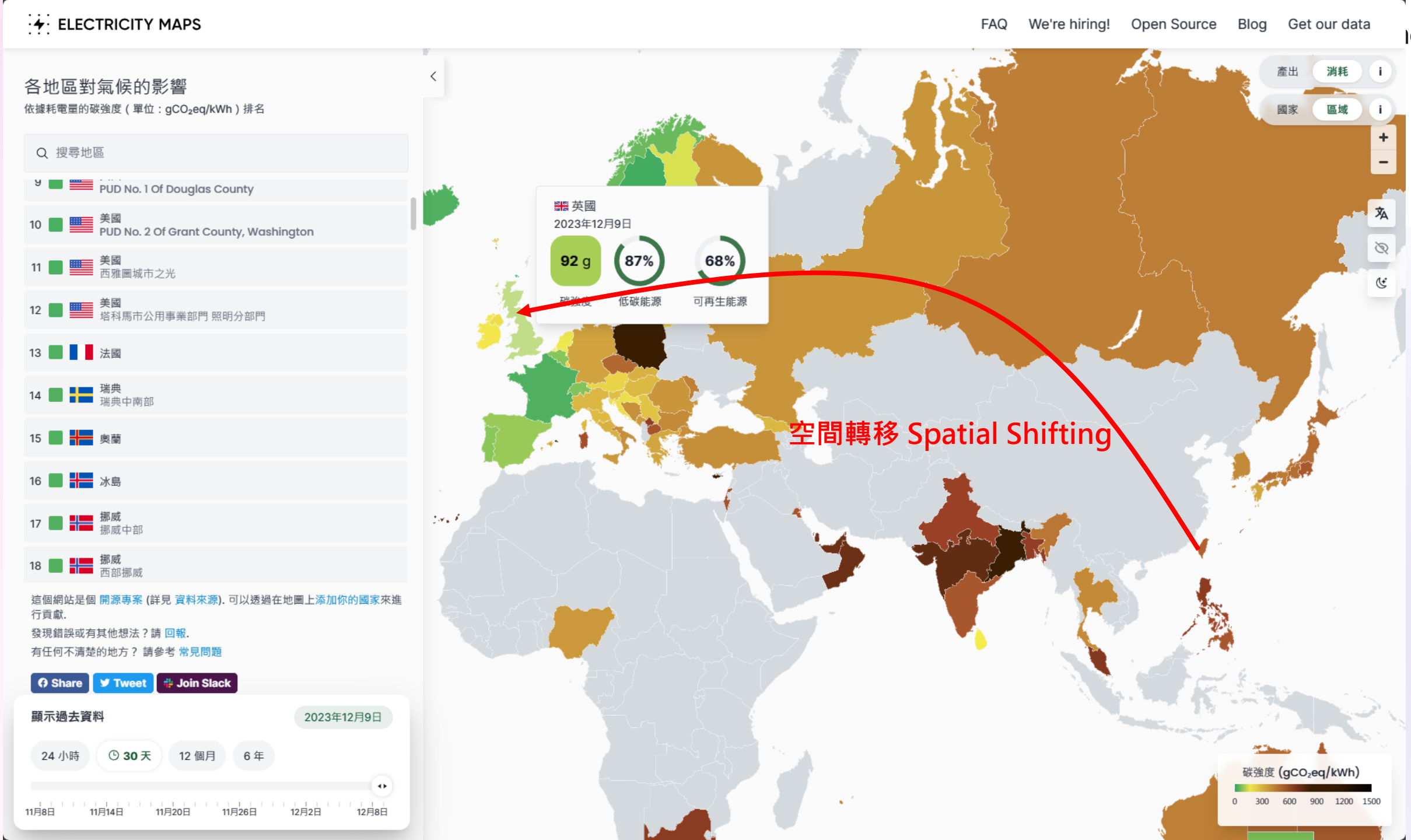
空間轉移 Spatial Shifting

轉移到碳排強度更低的區域運行，例如跟著四季或發展綠電優異的地區移動

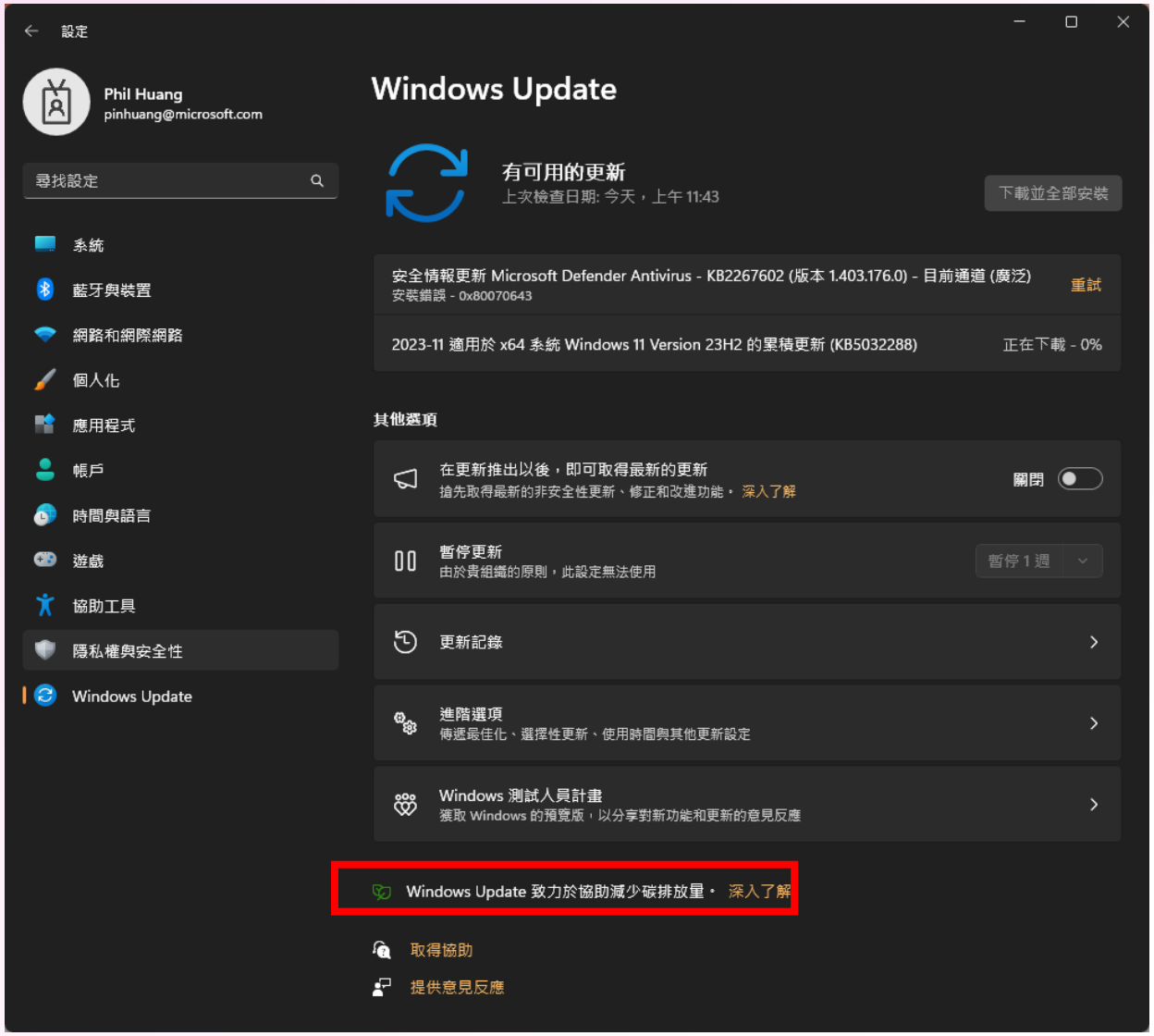


時空轉移 Temporal Shifting

轉移到碳排強度更低的時間再運行，譬如 Microsoft Carbon Aware Windows 計畫，選擇在碳排強度最低的時候進行 Windows Update

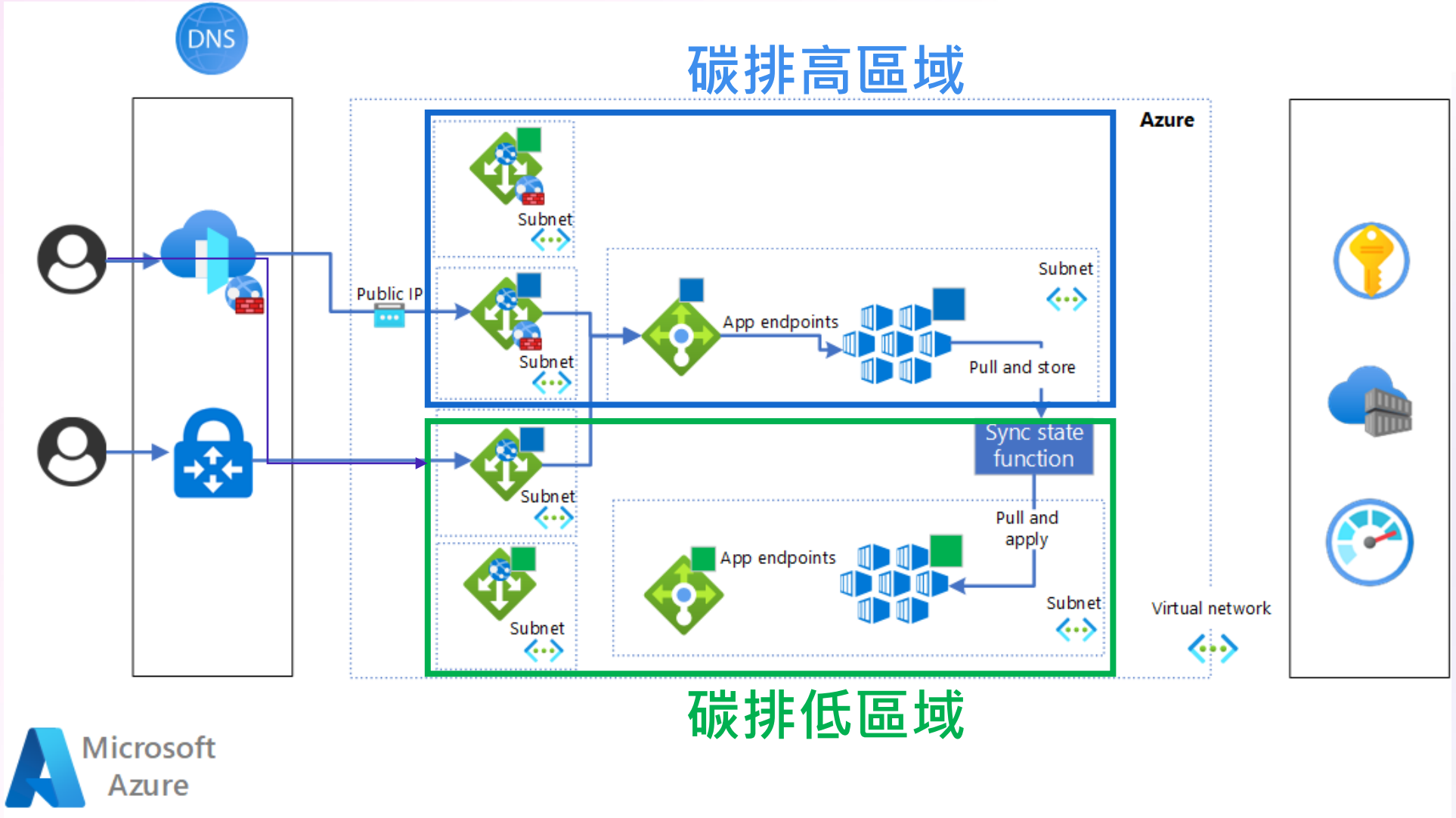

ELECTRICITY MAPS
FAQ
We're hiring!
Open Source
Blog
Get our data
各地區對氣候的影響
依據耗電量的碳強度（單位：gCO₂eq/kWh）排名
搜尋地區
9 PUD No. 1 Of Douglas County
10 美國 PUD No. 2 Of Grant County, Washington
11 美國 西雅圖城市之光
12 美國 塔科馬市公用事業部門 照明分部門
13 法國
14 瑞典 瑞典中南部
15 奧蘭
16 冰島
17 挪威 挪威中部
18 挪威 西部挪威
這個網站是個 開源專案 (詳見 資料來源). 可以透過在地圖上添加你的國家來進行貢獻.
發現錯誤或有其他想法？請 回報.
有任何不清楚的地方？ 請參考 常見問題
Share
Tweet
Join Slack
顯示過去資料
2023年12月9日
24 小時
30 天
12 個月
6 年
11月8日
11月14日
11月20日
11月26日
12月2日
12月8日
英國
2023年12月9日
92 g
87%
68%
低碳能源
可再生能源
空間轉移 Spatial Shifting
產出
消耗
國家
區域
碳強度 (gCO₂eq/kWh)
0
300
600
900
1200
1500

Microsoft Carbon Aware Windows



Ref: https://support.microsoft.com/en-us/windows/windows-update-is-now-carbon-aware-a53f39bc-5531-4bb1-9e78-db38d7a6df20

Azure Kubernetes Service 藍綠部署 - 永續性詮釋



https://learn.microsoft.com/zh-tw/azure/architecture/guide/aks/blue-green-deployment-for-aks

個人心得

$$SCI = ((E * I) + M) \text{ per } R$$

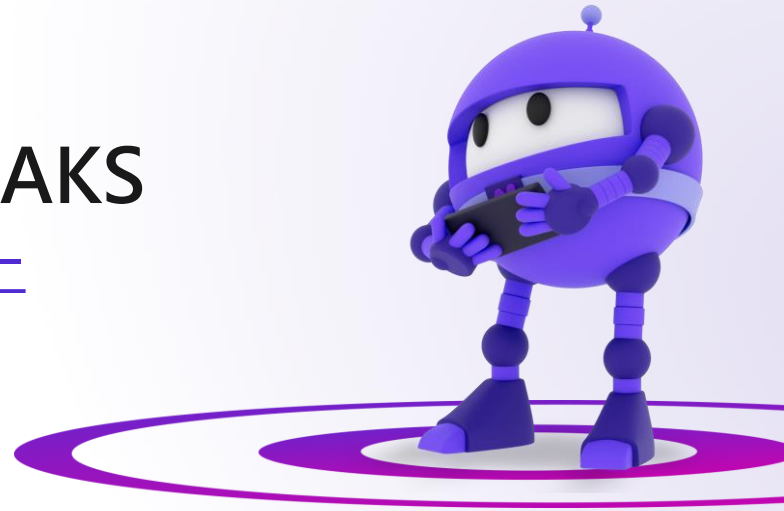
1. 永續軟體工程執行起來有相當難度，尤其是量測的部分
2. 要持續獲得正確的量測數據，需要額外去採購 Dataset
3. 設計初期，盡量讓 Azure Kubernetes Services 保持著架構修改的彈性及融合 GSF 所提供的建議
4. 運行中期，盡量保持持續監控系統使用率，如 CPU / Memory Avg. 因為與運行成本大部分正相關
5. 程式可以複製，但地球只有一個 (OnlyOneEarth)

Resources

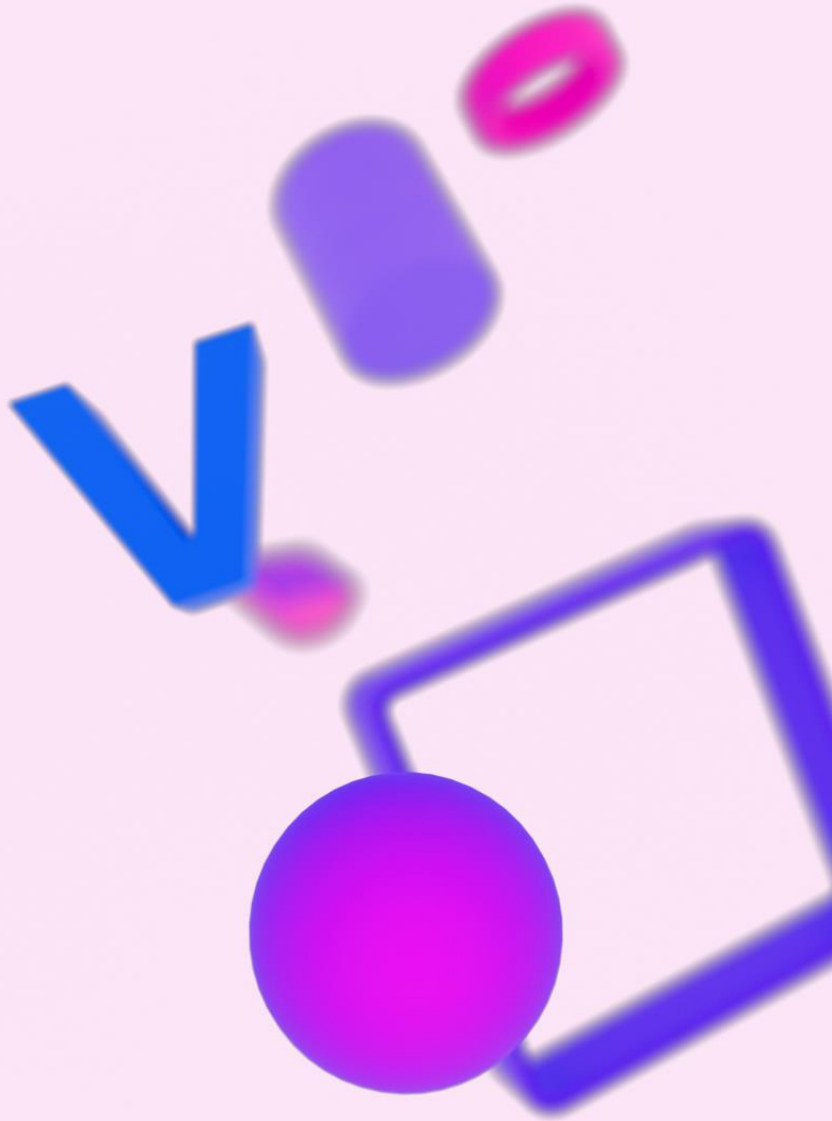
Green Software Foundation
greensoftware.foundation

Well-Architected Framework - Sustainability Workload
learn.microsoft.com/en-us/azure/well-architected/sustainability

Sustainable software engineering practices in AKS
learn.microsoft.com/en-us/azure/aks/concepts-sustainable-software-engineering



謝謝聆聽



STUDY4 Build School

黃秉鈞

Azure Kubernetes Services 永續性軟體工程設計方針

今年 KubeCon NA 2023 的主題演講之一是探討雲端原生環境永續性 (Environmental Sustainability) 的原則和實踐，其中提到了希望大家在使用雲端原生的技術的時候，也能夠關注計算資源利用率和延伸的環境影響，設計出更加符合當前環境永續性的架構。

本演講將會基於 Azure Kubernetes Services 來簡介採用現今 Green Software Foudantion 所提出的原則之下，如何設計出更加符合環境永續性的架構。

12/9 - 10 台大社會科學院

Microsoft tsmc 成功 HackMD

.NET Conf TAIWAN

STUDY4 學習而生 Build School