分享如何在 GCP 上
搭個 K3s 來學習 K8s
然後透過 Argo CD 佈署我的應用

Allen Jiang
2023.06.17

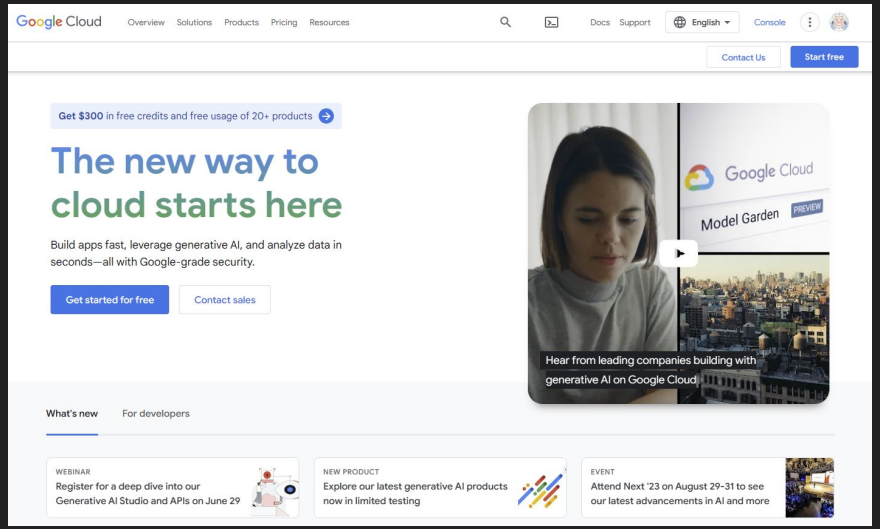



大綱

- Server 的演化史
  a. 為什麼雲端越來越夯
  b. 什麼是容器, 容器化有什麼好處？
- 介紹什麼是 GCP
- 介紹什麼是 IaC 與 GitOps
- 如何搭建 K3s 叢集
- K8s 基本概念與 Kubectl 基本操作
- 在叢集上安裝 Argo CD
  a. 什麼是 CI/CD
  b. 宣告式與指令式的差異
  c. 使用 Argo CD 佈署你的應用

什麼是 Google Cloud Platform (GCP)？

- Google 家的雲端服務
- 服務有很多類型，最常用的就是在上面開虛擬機器跑一些酷東西
- 要 $$，pay as you go，初次使用有 300 美金的扣打可以用



GCP 的優點

- 有台灣地區可供選擇，你可以在台灣開機器，機器連線的速度 Hen 快

使用 IaC 工具來操作 GCP 資源

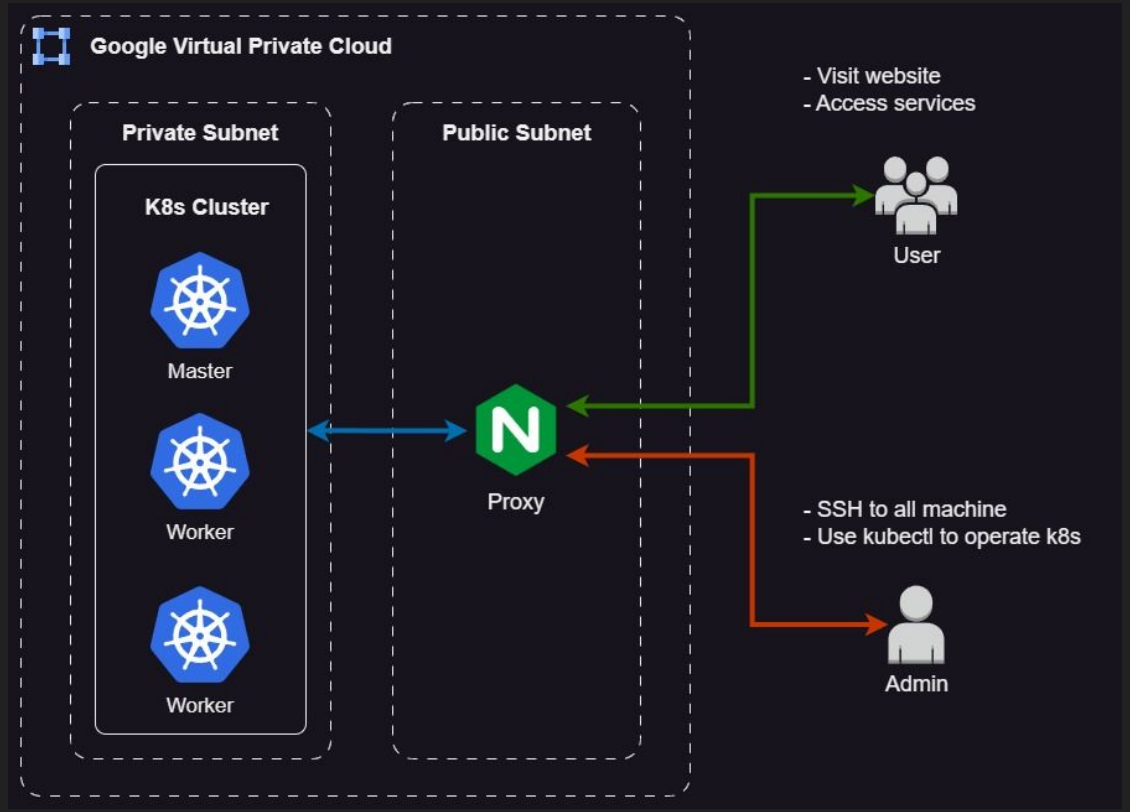
- 避免資源建立後忘記刪除，建議使用 Terraform (或 Pulumi) 來管理雲端資源
- 在使用 Terraform 之前，我們需要使用 gcloud CLI 來設定權限

HashiCorp Terraform

Pulumi

* IaC is **Infrastructure as Code**

架構


Google Virtual Private Cloud
Private Subnet
K8s Cluster
Master
Worker
Worker
Public Subnet
Proxy
- Visit website
- Access services
User
- SSH to all machine
- Use kubectl to operate k8s
Admin

Demo - 在 GCP 上建立資源

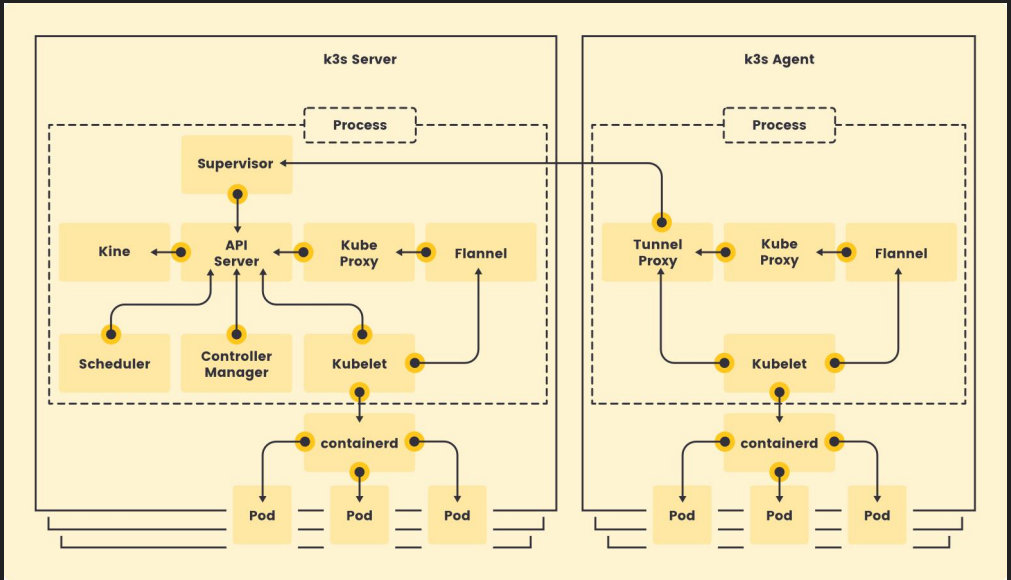
- 安裝 gcloud CLI 並設定給 Terraform 使用的憑證 (Certificate)
  - 可參考我放在 GitHub 上的筆記
- 使用 Terraform 在 GCP 上建立我們需要的機器

Type	Name	Private IP	Public IP	OS
e2-micro	proxy-server	10.0.0.10	處於量子糾纏	Ubuntu 22.04
e2-medium	k3s-server	10.0.1.10		Ubuntu 22.04
e2-medium	k3s-agent-1	10.0.1.11		Ubuntu 22.04
e2-medium	k3s-agent-2	10.0.1.12		Ubuntu 22.04

什麼是 K3s？

- 為 K8s (kubernetes) 的輕量版本，記憶體使用率較低
- 為 CNCF (Cloud Native Computing Foundation) 的 Sandbox 專案
- 主要在用 IoT 這種資源較為受限的環境
- K3s 名稱怎麼來的？
  - K8s 原詞 kubernetes 總共有 10 個字，頭尾的字母保留不算，中間共有 8 個字，因此得命 k8s
  - K3s 的設計初衷，是希望在記憶體使用上只有 k8s 的一半，所以 10 / 2 = 5，根據 k8s 的命名方式，5 個字的的詞，如果頭尾字母保留不算，中間會有 3 個字母，k3s 的名稱便是由此而來

K3s 的架構



API Server：提供 k3s 的 RESTful API 服務，kubectl 就是透過此 API 操作 k3s 上的資源

Kubelet：在每個 node 上運行，確保 pod 中的容器正常運行

Scheduler：K3s 上的調度員，會根據node 上的硬體資源決定 pod 要在哪個 node 上面運行

Controller Manager：包含多個控制器，控制器會不斷的對叢集狀態進行調節，使其與期望狀態一致

Containerd：開源的容器執行期(runtime)，負責管理容器的生命週期，如創建容器與停止容器

Pod：K3s 中的最小單位，可以運行一個或多個容器(但建議 pod 內只運行 1 個容器)

Demo - 搭建 K3s 叢集

1. 首先使用 SSH 遠端連線到 k3s-server 上安裝 k3s server
2. 將 kube config 放到本地環境上
3. 更新 proxy 的設定，讓我們可以在本地環境上使用 kubectl 操作 GCP 上的 k3s
4. 接下來連線到 k3s-agent-1 與 k3s-agent-2 上安裝 k3s agent

* Demo 流程文件請參考這裡

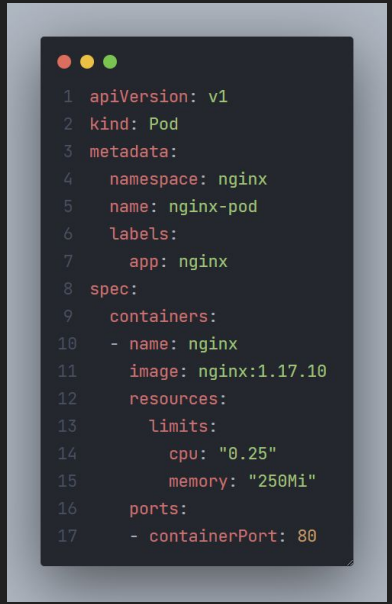
什麼是 Kubectl？

- K8s 提供的官方工具，可以讓管理者直接與 K8s 互動，例如建立、查看或是刪除資源
- K3s 也同樣可以使用

```
# 查看所有的 namespaces
kubectl get namespaces

# 查看特定 namespace 下的所有 resources
kubectl get all -n <namespace>

# 查看特定 namespace 下的所有 services
kubectl get services -n <namespace>

# 查看特定 namespace 下的所有 deployments
kubectl get deployments -n <namespace>

# 查看特定 namespace 下的所有 pods
kubectl get pods -n <namespace>

# 查看某一個 pod 的詳細資訊
kubectl describe pod <pod_name> -n <namespace>
```

使用 IaC 管理你的 K3s 資源

- 可以使用 JSON 或是 YAML 檔案來敘述你的 k3s 資源
- 使用指令 `***kubectl apply -f <file>***` 讀取檔案並在 k3s 中建立資源



```yaml
apiVersion: v1
kind: Pod
metadata:
  namespace: nginx
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.17.10
    resources:
      limits:
        cpu: "0.25"
        memory: "250Mi"
    ports:
    - containerPort: 80
```

```yaml
apiVersion: v1
kind: Service
metadata:
  namespace: nginx
  name: nginx-service
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30090
  selector:
    app: nginx
```

K3s 中的各種資源類型 (Kind)

- **Pod**：前面講過了 😏
- **ReplicaSet**：負責確保指定數量的 Pod 在運行，它提供了彈性伸縮的基本機能
- **Deployment**：進一步管理 ReplicaSet，提供更新策略 (如滾動更新)，和回滾功能 (功能上完全可以取代 ReplicaSet)
- **Service**：提供穩定的網路接口給 Pod，讓 Pod 中的服務可以被外部訪問
- **CofingMap**：提供非敏感設定資料給 Pod，可以用來設定容器中的環境變數
- **Secret**：與 ConfigMap 功能類似，但我是敏感的 😘，此外，我還是明碼 😈
- **PersistentVolume**：提供儲存區域，可以是本地硬碟，也可以是雲端存儲服務 (Google Cloud Storage、AWS S3)

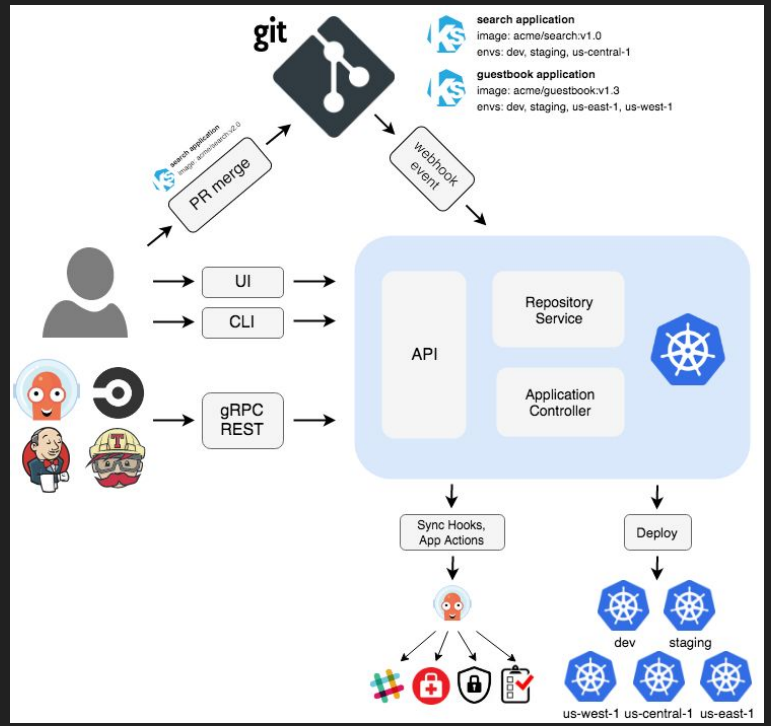
什麼是 Argo CD？

- Argo CD 是用於 k8s 上的宣告式 (Declarative) GitOps 持續交付工具
- 為 CNCF 的 Graduated 專案

Argo CD 的運作流程

- 需要在 k3s 叢集中佈署 Argo CD 服務
- 基於 pull 的佈署模式，Argo CD 會週期性地拉取 Git 存儲庫中的配置清單 (manifest)，**配置清單會敘述我們期望在 k3s 中運行的服務狀態**，也就是所謂的**期望狀態**
- 會將叢集中服務的**實際狀態與期望狀態進行比較**，如果實際狀態不符合期望狀態，就會更新服務的實際狀態以匹配期望狀態

Argo CD 的運作流程



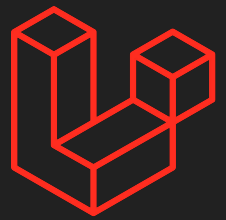
Demo - 在 K3s 上部署 Argo CD

- 在 k3s 中佈署 Argo CD
- 更新 proxy 的設定，使 Argo CD Web UI 可以對外連線
- 嘗試部署一個 nginx server 到 k3s 中

* Demo 流程文件請參考這裡

最終 Demo - 使用 Argo CD 佈署我的 Laravel 應用

- 將我的網頁應用配置清單上傳到 GitHub 上
- 讓 Argo CD 讀取 GitHub 上的配置清單，然後在 k3s 中佈署 …
- 等等 …



等等，Secret 不能上版控吧？

- Secret 裡面放了很多第三方服務的 secret key 還有資料庫的密碼
- Secret 使用明碼儲存。如果丟上版本控制，就好比你去人擠人的百貨公司裸奔
- 無法將 Secret 上傳到 GitHub，那 Argo CD 要怎麼部署？

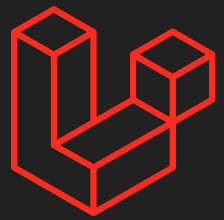
什麼是 Sealed Secret？

- Sealed secret 為 Bitnami Labs 開發的工具，主要有兩個部分
  - 運行在 k3s 上的 sealed secret controller
  - 客戶端加密工具 kubeseal
- 使用 kubeseal **將 secret 加密變成 sealed secret** 後，再放入版本控制中
- 將 sealed secret 佈署到 k3s 中之後，**sealed secret controller 會自動將你的 sealed secret 解密，並生成解密後的 secret**

* 介紹文件請參考這裡

真 ● 最終 Demo - 使用 Argo CD 佈署我的 Laravel 應用

- 在 k3s 上佈署 sealed secret controller
- 使用 kubeseal 加密 secret
- 將我的網頁應用配置清單上傳到 GitHub 上
- 讓 Argo CD 讀取 GitHub 上的配置清單，然後在 k3s 中佈署網頁應用



Bonus - Reloader

- 發現更新 ConfigMap 與 Secret 後，Pod 還是使用舊版的
- 如何確保 Pod 會使用最新版本的 ConfigMap 與 Secret？
- 檢查 ConfigMap 與 Secret 是否有更新，如果有的話，會更新與之相依的 Pod

* 介紹文件請參考這裡

分享結束，感謝你的聆聽 😁