

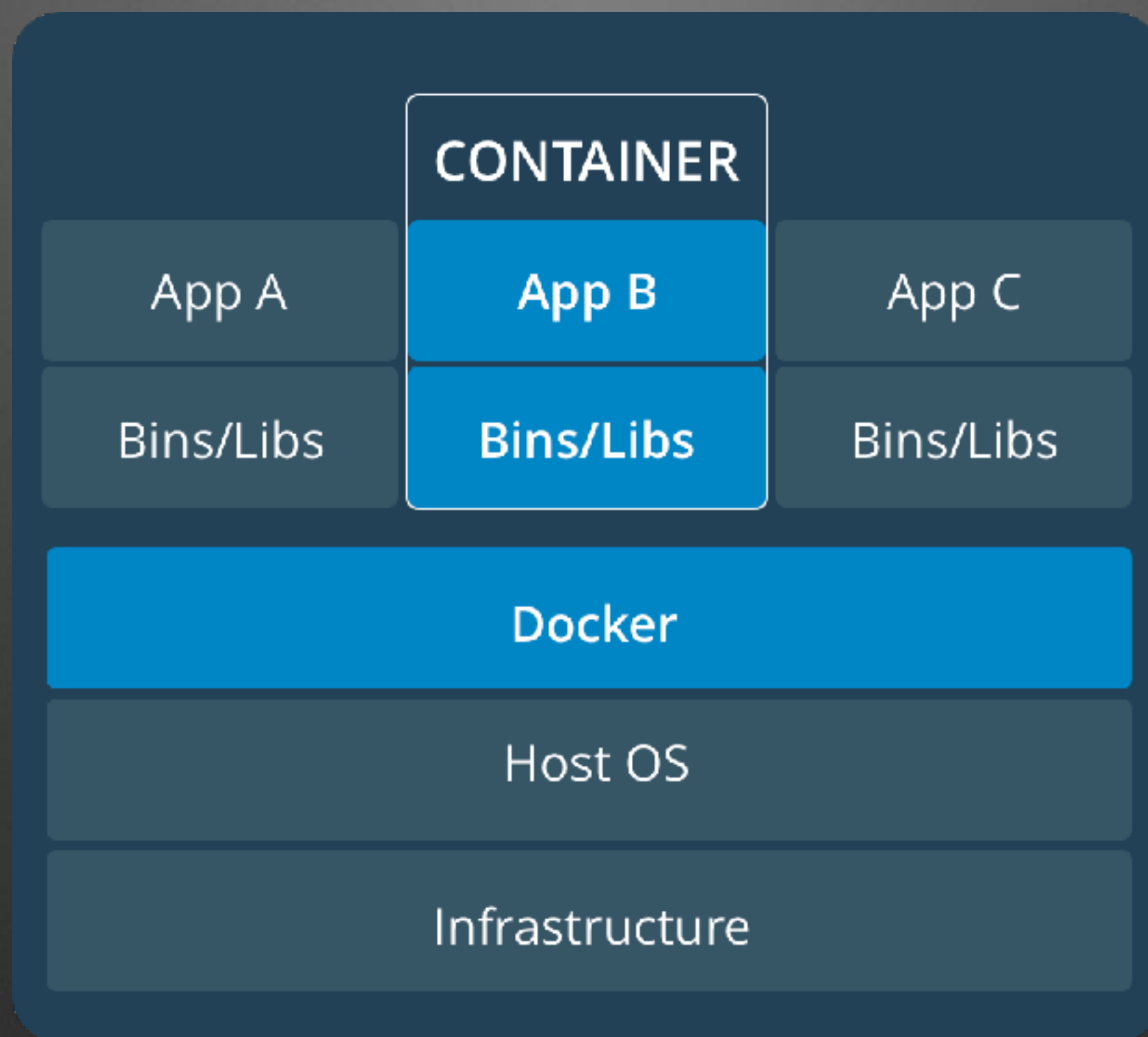
Kubernetes Tutorial

PLANET SCALE 、 NEVER OUTGROW 、 RUN ANYWHERE

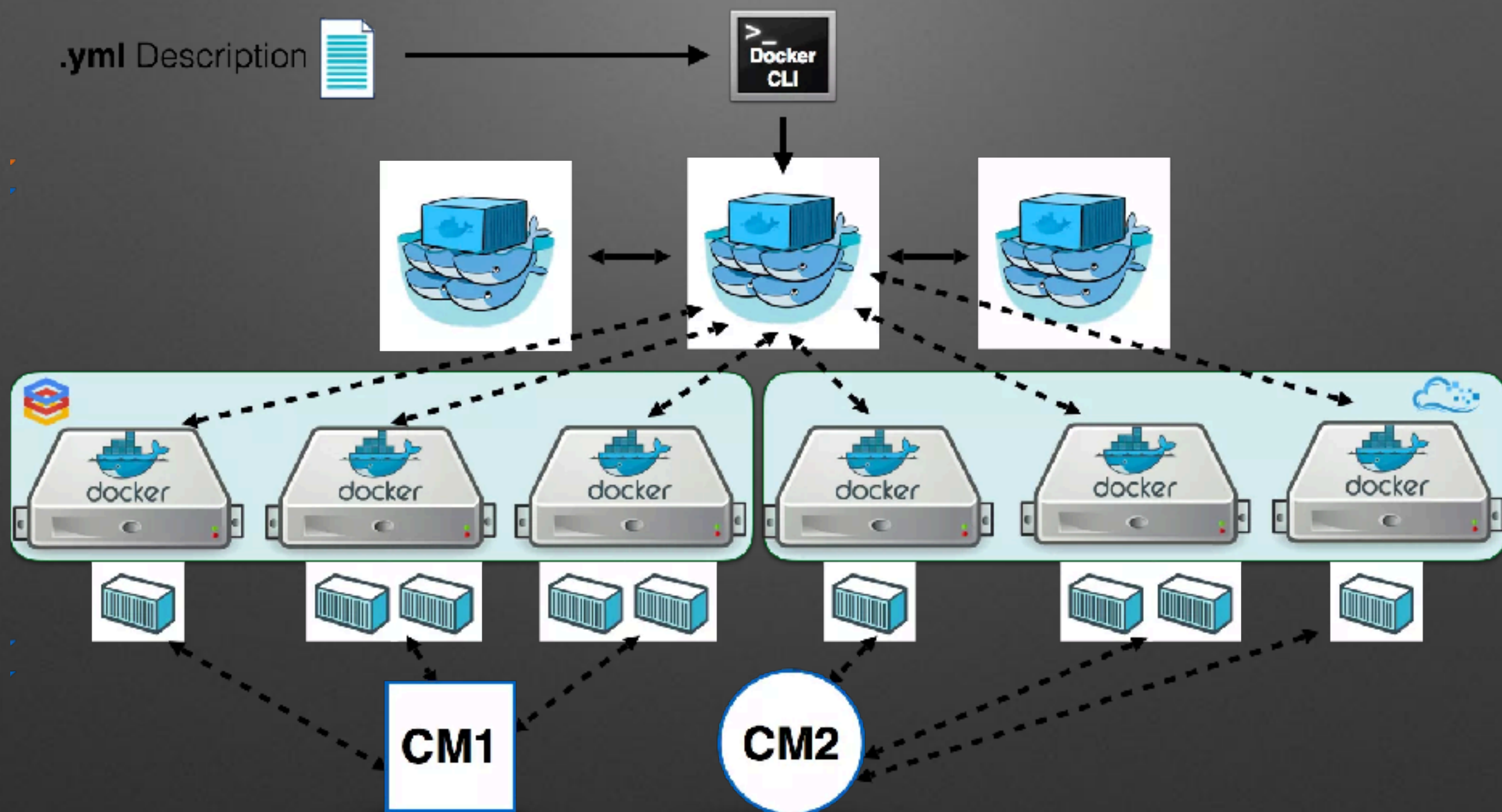
OutLine

- 一、為何需要容器叢集管理工具
- 二、容器叢集管理工具簡介
- 三、Azure Kubernetes Service
- 四、Kubernetes 簡介
- 五、Helm簡介
- 六、Reference

一、為何需要容器叢集管理工具



一、為何需要容器叢集管理工具



二、容器叢集管理工具

- Docker Swarm
 - Docker
- Fleet
 - Core OS
- Mesos
 - Apache
- Kubernetes (k8s)
 - Google

三、Azure Kubernetes Service

- 3-1 Azure cli
- 3-2 Azure Kubernetes Service (AKS)簡介

三、Azure Kubernetes Service

- 3-1 Azure cli
 - 從終端機登入Azure
 - az login
 - 查詢自己的azure帳號有哪些訂閱
 - az account list -o table
 - 切換訂閱
 - az account set -s [SubscriptionId]
 - 查詢自己的azure帳號目前是使用哪個訂閱
 - az account show -o table

三、Azure Kubernetes Service

- 3-2 AKS簡介
 - 主要會有兩個ResourceGroup
 - 放置AKS叢集管理服務 (free)
 - 放置AKS相關的Infatructrue服務，例如：
Node Server, LoadBalancer和Azure Disk等

三、Azure Kubernetes Service

- 3-2 AKS簡介
 - 建立aks指令
 - `az group create --name AKSRG --location japaneast`
 - `az aks create --resource-group AKSRG --name demo-aks --node-count 3 --dns-name-prefix AKSRG --generate-ssh-keys --node-vm-size Standard_DS5_v2 --kubernetes-version 1.9.9 --no-wait`

三、Azure Kubernetes Service

- 3-2 AKS簡介
 - 連結aks指令
 - az aks install-cli
 - az aks get-credentials --resource-group AKSRG --name demo-aks
 -

三、Azure Kubernetes Service

- 3-2 AKS簡介
 - 建立aks dashboard指令
 - cd 至2-k8sTutorial資料夾
 - kubectl create -f dashboard.yaml
 - 利用vs code 開啟aks的dashbaord

四、Kubernetes簡介

- 4-1 What is Kubernetes
- 4-2 Kubernetes Architecture
- 4-3 Why use Kubernetes
- 4-4 Kubernetes Objects
- 4-5 Yaml檔設計方式
- 4-6 常用Commands

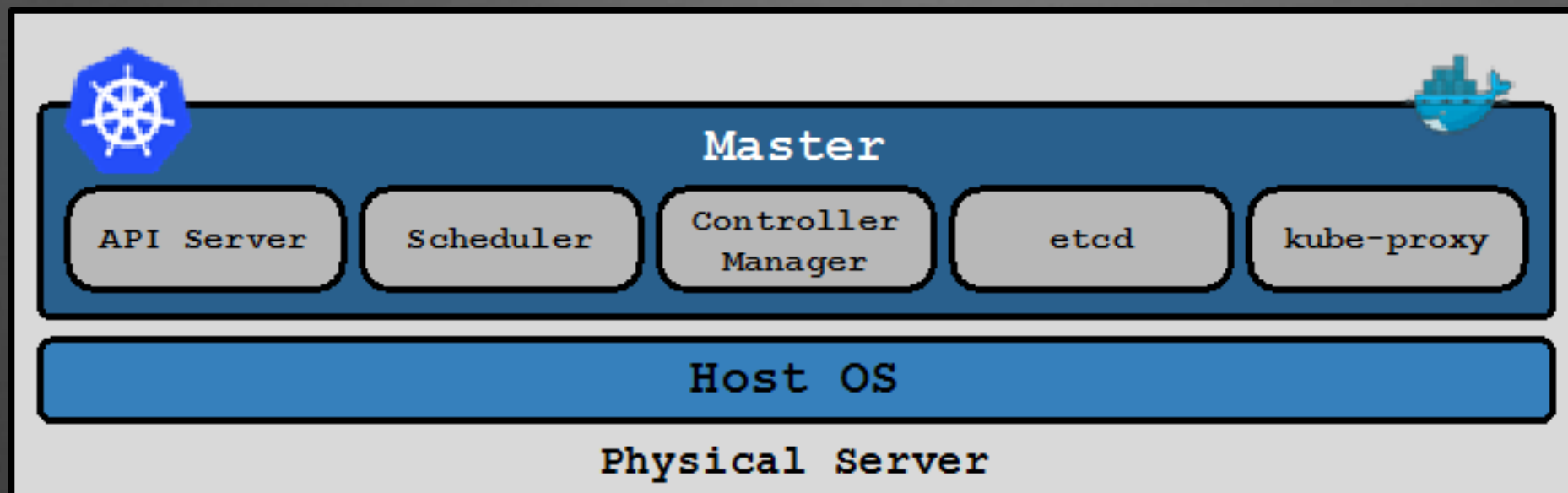
四、Kubernetes簡介

- 4-1 What is Kubernetes
 - 自動化部署、管理和負載平衡容器的開源軟體
 - 分散式系統架構

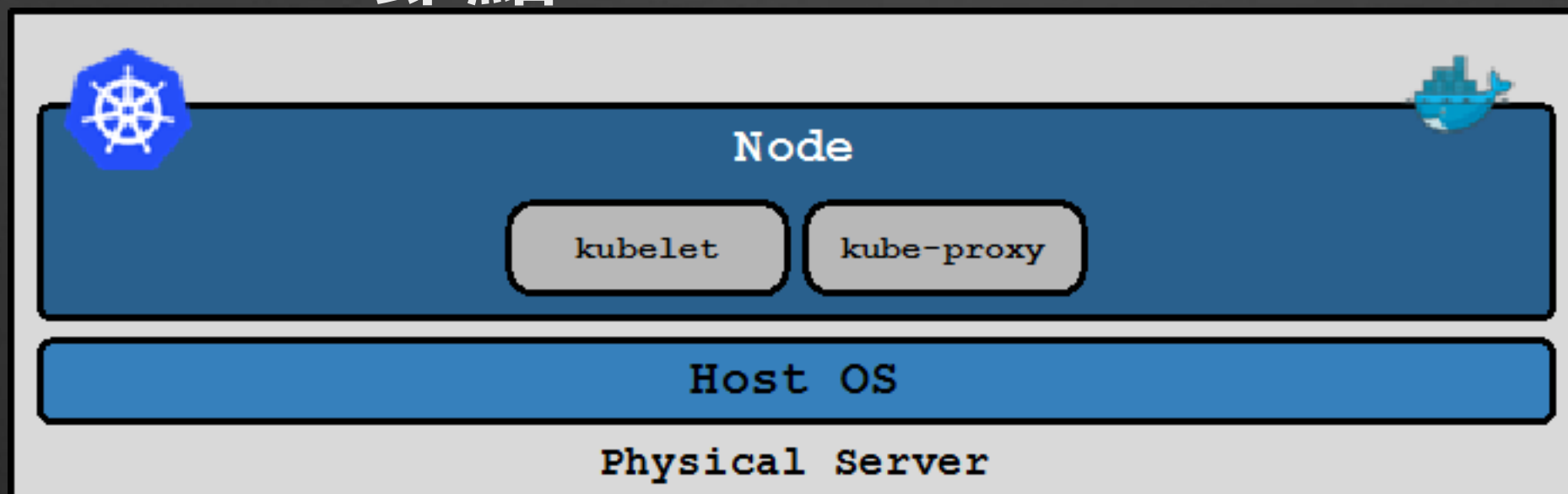
四、Kubernetes簡介

- 4-2 Kubernetes Architecture

- Master節點



- Node節點



四、Kubernetes簡介

- 4-3 Why use Kubernetes
 - 管理功能特性
 - 多租戶管理界面
 - 管理者可以定義多個系統角色 (Cluster Role)，各自擁有不同的叢集權限
 - 操作特定的命名空間 (namespace) 的資源

四、Kubernetes簡介

- 4-3 Why use Kubernetes
 - 管理功能特性
 - 平台可擴充性
 - 叢集運算資源不足時，可以輕易地擴充運算節點。
 - 擴充的節點需安裝相對應版本之Kubernetes元件和Docker。

四、Kubernetes簡介

- 4-3 Why use Kubernetes

- 管理功能特性

- 多元儲存介面

- 雲端儲存服務

- NFS檔案共享系統

四、Kubernetes簡介

- 4-3 Why use Kubernetes
 - 應用程式特性
 - 滾動式升級 (Rolling Update)
 - 應用服務擴展性 (Service Scalability)
 - 自動化負載平衡 (Service Discovery and Load Balancing)
 - 自動重啟失能服務 (Self-healing)
 - 容器網路管理 (Network Policy)

四、Kubernetes簡介

- 4-4 Kubernetes Objects
 - Pod
 - Replication Controller
 - Service
 - Label
 - Namespace
 - Node

四、Kubernetes簡介

- 4-4 Kubernetes Objects
 - Pod
 - 許多相關容器的組合，Pod所包含的容器都運作在同一台宿主機 (Node)中
 - Pod中所有容器使用同一個網路命名空間、IP位址和通訊埠，並可共用Volume
 - Pod是k8s中建立、排程和管理的最小單位

四、Kubernetes簡介

- 4-4 Kubernetes Objects
 - Replication Controller
 - 管理Pod的備份，保障k8s的應用程式可以正常執行
 - 確保任何時候k8s叢集中有指定數量的Pod在執行，如果少於指定數量的Pod的備份，Replication Controller會啟動新的Pod備份，反之則刪除多餘的備份以保障數量不變

四、Kubernetes簡介

- 4-4 Kubernetes Objects

- Service

- 真實應用服務的抽象

- Pod對外服務的單一存取介面

四、Kubernetes簡介

- 4-4 Kubernetes Objects
 - Label
 - 區分Pod, Replication Controller和Service
 - Label是Service和Replication Controller連結Pod的基礎。
 - 例如：Label名稱為dev的Service只會連線到Label名稱為dev的Pod。

四、Kubernetes簡介

- 4-4 Kubernetes Objects
 - Namespace
 - 區分k8s物件集合的資源集合，以方便管理和隔離不同應用的物件集合。
 - 例如：將Label名稱為dev的Pod、Replication Controller、Service等物件放到Namespace為DevNamespace名稱的資源集合裡。

四、Kubernetes簡介

- 4-4 Kubernetes Objects
 - Node
 - k8s的操作單元，用來分配給Pod進行綁定，並提供運算資源給Pod。

四、Kubernetes簡介

- 4-5 Yaml檔設計方式

- 自行設定的參數值需以小寫為主，中間可以-連結

```
apiVersion: [API版本編號]
```

```
kind: [物件類型]
```

```
metadata:
```

```
  name: [物件名稱]
```

```
  namespace: [Namespace名稱]
```

```
  labels:
```

```
    [LabelKey]app: LabelName
```

```
spec: [物件相關規格]
```

四、Kubernetes簡介

- 4-6 常用Commands
 - 使用方式概述
 - Pod
 - Replication Controller
 - Service

四、Kubernetes簡介

- 4-6 常用Commands
 - 使用方式概述
 - kubectl [options]
 - 建立NameSpace
 - kubectl create namespace [YourNameSpace]
 - kubectl config set-context demo-aks --namespace= [YourNameSpace]

四、Kubernetes簡介

- 4-6 常用Commands

- Pod (Hands-on 1-PodOverView)

- 建立Pod

- `kubectl create -f pod-overview.yaml`

- 更新特定Label的Pod，如果不存在則建立Pod

- `kubectl apply -f pod-overview.yaml`

- 取代特定Label的Pod

- `kubectl replace -f pod-overview.yaml`

- 進入Pod的Container

- `kubectl exec -it myapp-pod --container=myapp-container -- /bin/sh`

四、Kubernetes簡介

- 4-6 常用Commands

- Pod (Hands-on 1-PodOverview)

- 刪除Pod

- `kubectl delete -f pod-overview.yaml`

- 查詢Pod Log

- `kubectl logs myapp-pod --container=myapp-container`

- 查詢Pod資訊

- `kubectl get pod myapp-pod`

- 查詢Pod 詳細描述

- `kubectl describe myapp-pod`

- 強制建立pod

- `kubectl replace --force -f pod-overview.yaml`

四、Kubernetes簡介

- 4-6 常用Commands
 - Pod (Hands-on 2-Pod-Policy)
 - Image 下載策略 (ImagePullPolicy)
 - always : 每次建立Pod時，下載新的Image
 - Never : 只使用本機Image
 - IfNotPresent : 只有當本機沒有時才下載Image

四、Kubernetes簡介

- 4-6 常用Commands
 - Pod (Hands-on 2-Pod-Policy)
 - Pod 重新啟動策略 (restartPolicy)
 - Always：當容器停止時，總是重新啟動容器，此為預設策略
 - OnFailure：當容器異常停止時 (退出碼非0時)，才重新啟動容器
 - Never：當容器終止退出時，不啟動容器
 - **kubectl create -f pod-policy.yaml**

四、Kubernetes簡介

- 4-6 常用Commands
- Replication Controller VS Deployment

-

比較	REPLICATION CONTROLLER	DEPLOYMENT
確保POD 數量	V	V
確保POD 健康情況	V	V
彈性伸縮	V	V
自動伸縮	V	V
滾動升級	V	V
版本記錄	X	V

四、Kubernetes簡介

- 4-6 常用Commands
 - Deployment (Hands-On:3-DeploymentOverView)
 - `kubectl get deployment`
 - `kubectl describe deployment`
 - `kubectl create -f deploy.yaml`
 - `kubectl apply -f deploy.yaml`
 - `kubectl replace -f deploy.yaml`
 - `kubectl replace --force -f deploy.yaml`
 - `kubectl delete -f deploy.yaml`
 - `kubectl get pod --selector app=nginx`

四、Kubernetes簡介

- 4-6 常用Commands
 - Deployment (Hands-On:3-DeploymentOverView)
 - 彈性伸縮
 - `kubectl scale --replicas=1 -f deploy.yaml`
 - `kubectl scale deployment my-nginx --replicas=4`
 - `kubectl scale --current-replicas=2 --replicas=3 -f deploy.yaml`
 - `kubectl scale --current-replicas=2 --replicas=3 deployment my-nginx`
 - `kubectl scale --replicas=0 -f deploy.yaml`

四、Kubernetes簡介

- 4-6 常用Commands
 - Deployment (Hands-On:4-DeploymentAutoScale)
 - 自動伸縮
 - `kubectl create -f auto-scale.yaml`
 - `kubectl replace --force -f auto-scale.yaml`

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: nginx-auto-scale
  namespace: default
  labels:
    app: nginx
spec:
  scaleTargetRef: #連結指定的Deployment
    apiVersion: apps/v1
    kind: Deployment
    name: my-nginx
  minReplicas: 1 #當所有連結的POD的平均CPU使用量小於50%，將pod備份最多縮減至1個
  maxReplicas: 10 #當所有連結的POD的平均CPU使用量大於50%，將pod備份最多增加至10個
  targetCPUUtilizationPercentage: 50
```

四、Kubernetes簡介

- 4-6 常用Commands
 - Service
 - 開放外部服務
 - NodePort (Hands-On : 5-ServiceNodePort)
 - k8s分配一組內部虛擬ip給Service，並在每個Node加入開放外部連線的隨機數字port (範圍：30000-32767)
 - LoadBalancer
 - k8s分配一組內部虛擬ip給Service，並在雲端平台中建立一組具有實體ip的負載平衡器
 - Ingress
 - Http路由轉發機制，支援SSL加密協定

四、Azure Kubernetes Service

- Hands-On : 6-LoadBalanceService
 - 建立自己的NameSpace
 - kubectl create namespace [NamespaceName]
 - 設定預設的NameSpace
 - kubectl config set-context demo-aks --namespace= [NamespaceName]

四、Azure Kubernetes Service

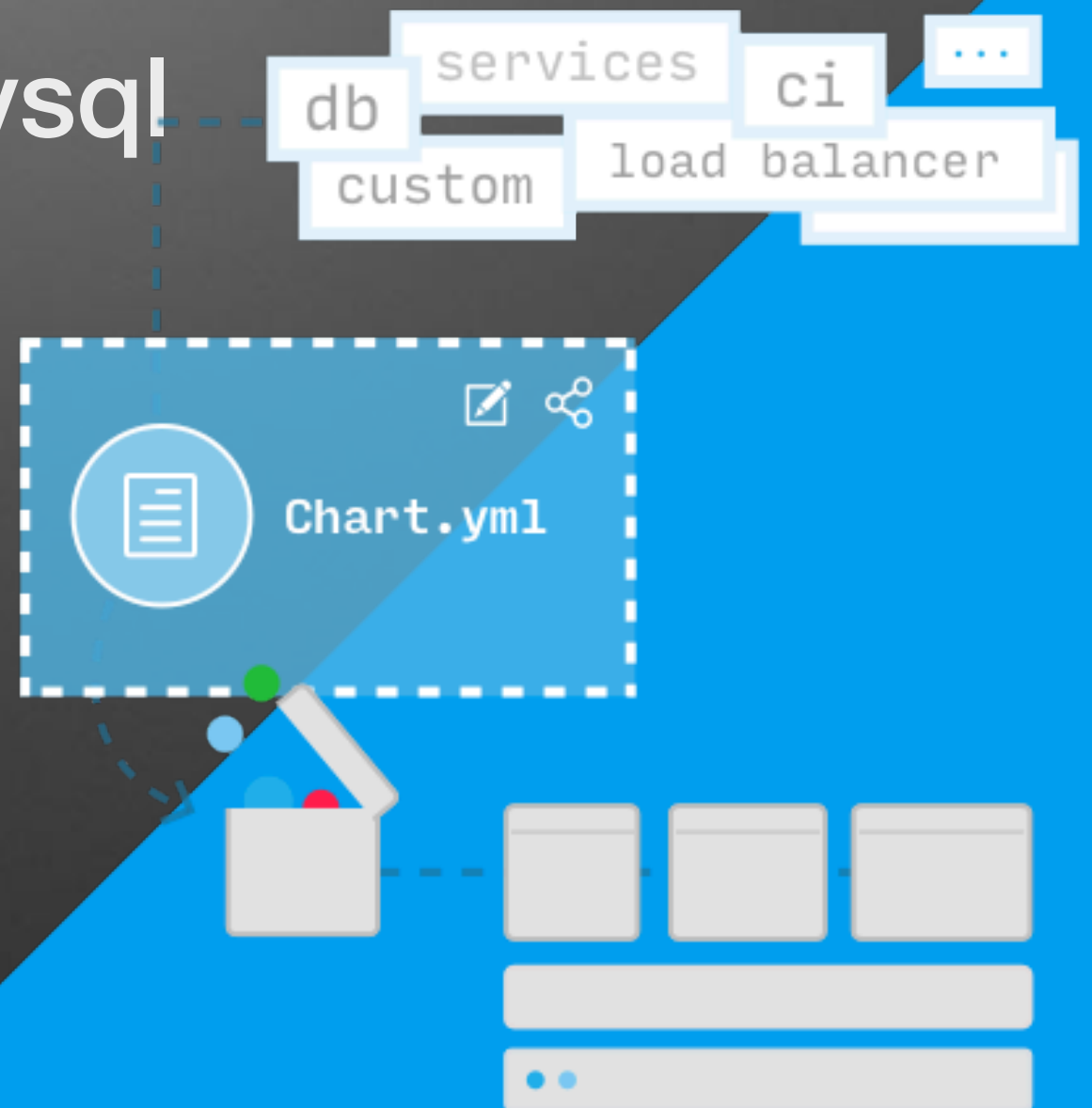
- Hands-On : 6-LoadBalancerService
- 建立LoadBalancer
 - az network public-ip create -g MC_AKSRG_demo-aks_japaneast -n [MyPublicIPName] --allocation-method static --dns-name [MyPublicIPName]
- 查詢LoadBalancer的ip和fqdn
 - az network public-ip show -g MC_AKSRG_demo-aks_japaneast -n [MyPublicIPName] --query "{fqdn: dnsSettings.fqdn, address: ipAddress}"

四、Azure Kubernetes Service

- Hands-On : 6-LoadBalancerService
 - 建立Azure Container Registry (ACR)連結
 - `az acr login --name [RegistryName]`
 - `az acr list -o table`
 - 建立和push Image
 - `docker build --no-cache --build-arg arg=production -f Dockerfile -t [ACR-LoginServer]/[ImageName]:[Tag] .`
 - `docker push ikgacr.azurecr.io/[ImageName]:[Tag]`
 - 建立k8s服務
 - `kubectl create -f deploy.yaml`

五、Helm簡介

- k8s的套件管理工具
- 套件所指的是在k8s所運行的物件
- 例如：wordpress+mysql



五、Helm簡介

- 組成要件
 - Chart
 - 記錄套件的Meta Data
 - Template
 - 用來建立架構相似，但使用目的不相同的物件

五、Helm簡介

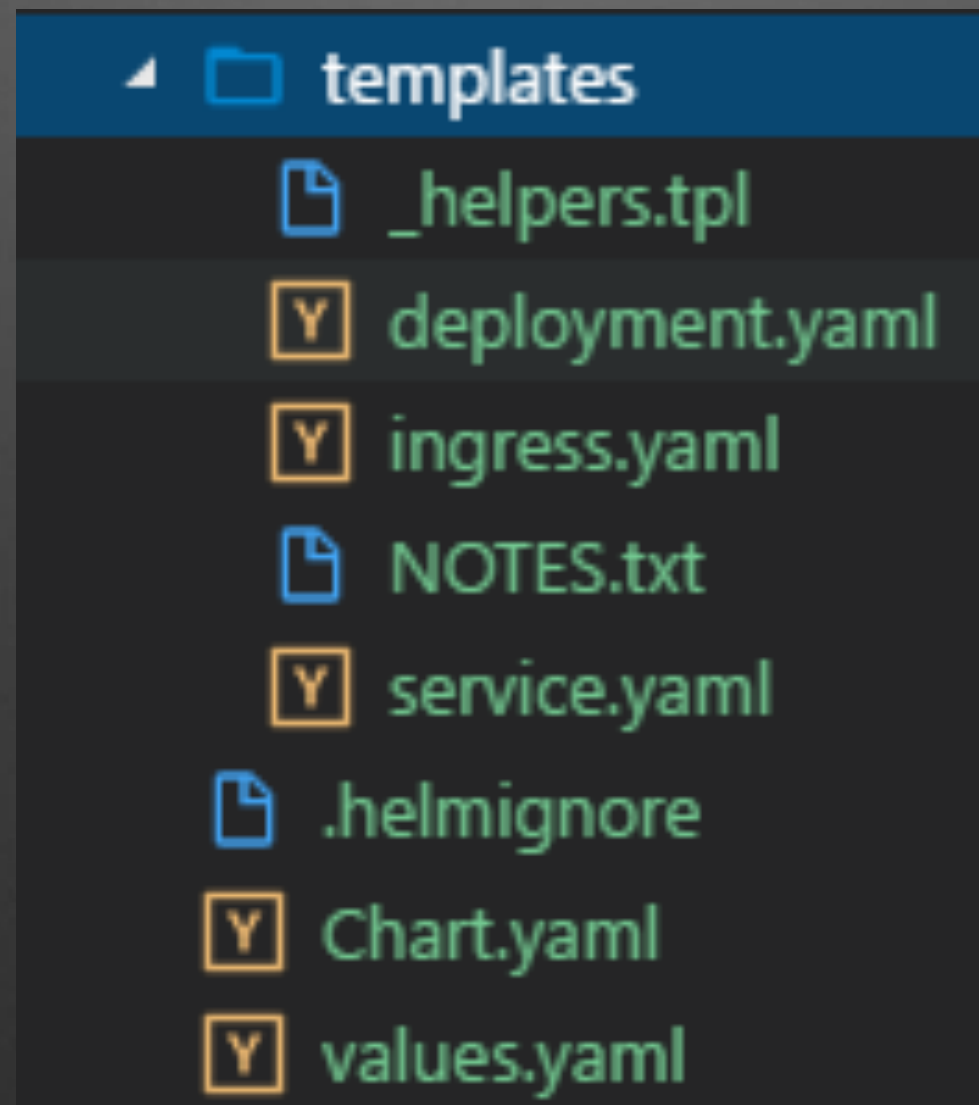
- 安裝方式
 - windows
 - choco install kubernetes-helm
 - mac
 - brew install kubernetes-helm
 - linux
 - wget <https://storage.googleapis.com/kubernetes-helm/helm-v2.9.1-linux-amd64.tar.gz>
 - tar -zxvf helm-v2.9.1-linux-amd64.tar.gz
 - sudo mv linux-amd64/helm /usr/local/bin

五、Helm簡介

- 常用Command
 - 初始化Helm
 - helm init
 - 建立chart
 - 安裝chart
 - 更新Chart
 - 查詢Chart的更新History
 - Rollback Chart
 - 刪除Chart的發佈

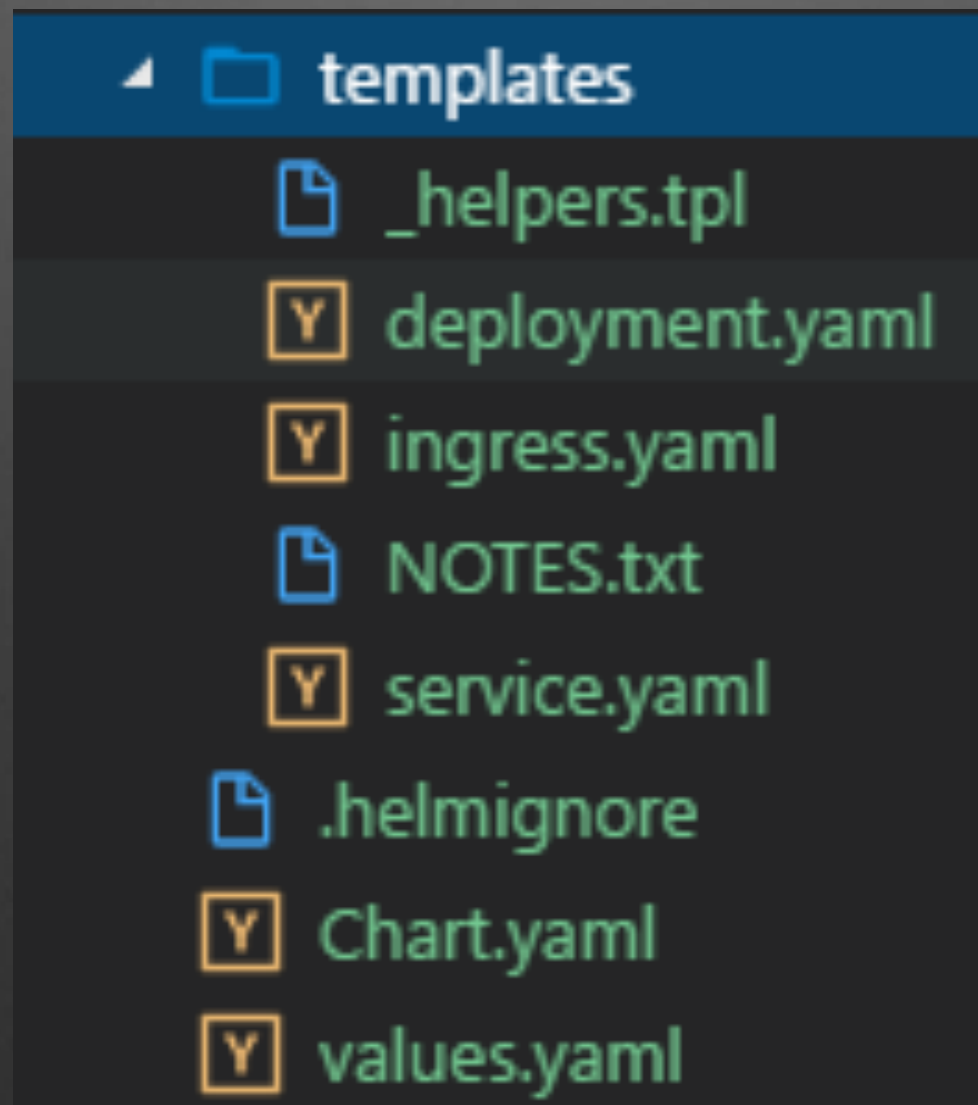
五、Helm簡介

- 常用Command
 - 建立chart
 - helm create [YourChartName]



五、Helm簡介

- 常用Command
 - 建立chart
 - helm create [YourChartName]



五、Helm簡介

- 常用Command
- 安裝chart (Hands-On:7-Helm)
 - `helm install --name [YourChartName] [ChartFolderPath] -f [values.yaml File Name]`
 - `helm install --name [YourChartName] ./docker-mvc-chart -f ./docker-mvc-chart/values.yaml`

五、Helm簡介

- 常用Command
- 更新chart (Hands-On:7-Helm)
 - `helm upgrade [ChartName] [ChartPath] -f [values.yaml file path] --set [options]`
- 其他常用參數
 - `--install` : 當chart不存在時，會先執行 `helm install` 指令，再執行 `helm upgrade` 指令

五、Helm簡介

- 常用Command
- 更新chart (Hands-On:7-Helm)
 - `helm upgrade [YourChartName] ./docker-mvc-chart -f ./docker-mvc-chart/values.yaml --set replicaCount=6`

五、Helm簡介

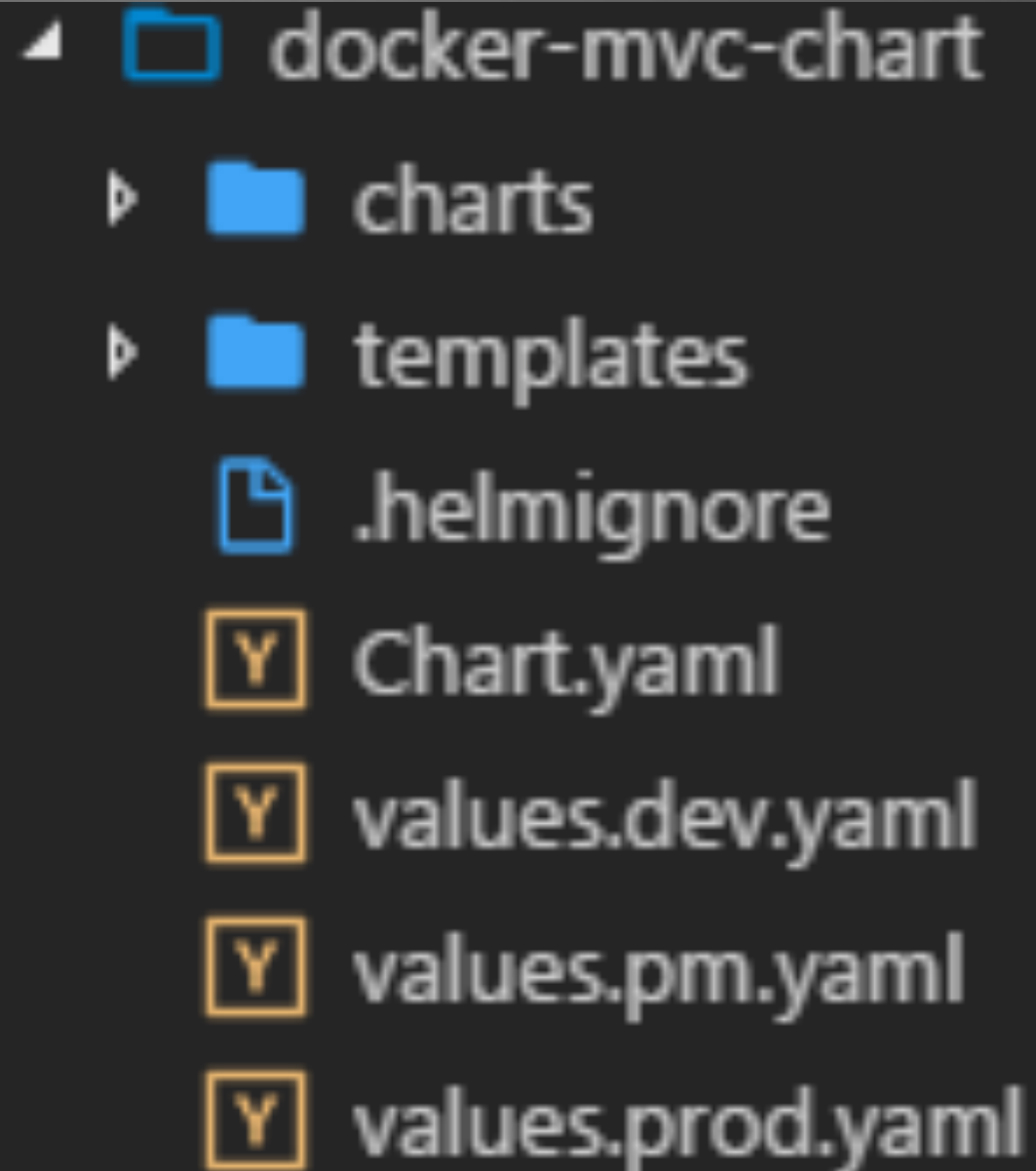
- 常用Command
 - 查看Chart歷史記錄 (Hands-On:7-Helm)
 - `helm history [ChartName]`
 - RollBack Chart (Hands-On:7-Helm)
 - `helm rollback [ChartName]`
`[VersionNumber]`

五、Helm簡介

- 常用Command
 - 刪除Chart的發佈
 - helm del [YourChartName]
 - 常用參數
 - --purge：刪除所有chart的版本

五、Helm簡介

- Hands-on:8-HelmMultiVersion
 - 先建立好你自己要使用的三個NameSpace



```
└─ docker-mvc-chart
  ├── charts
  ├── templates
  ├── .helmignore
  ├── Chart.yaml
  ├── values.dev.yaml
  ├── values.pm.yaml
  └── values.prod.yaml
```

A screenshot of a file explorer window showing the directory structure of a Helm chart named 'docker-mvc-chart'. The chart directory contains subdirectories 'charts' and 'templates', and files '.helmignore', 'Chart.yaml', 'values.dev.yaml', 'values.pm.yaml', and 'values.prod.yaml'. The 'values' files are highlighted with yellow selection boxes.

五、Helm簡介

- Hands-on:8-HelmMultiVersion
 - `helm install --name [YourChartName] ./docker-mvc-chart -f ./docker-mvc-chart/values.dev.yaml`
 - `helm install --name [YourChartName] ./docker-mvc-chart -f ./docker-mvc-chart/values.pm.yaml`
 - `helm install --name [YourChartName] ./docker-mvc-chart -f ./docker-mvc-chart/values.prod.yaml`

六、Reference

- <https://ithelp.ithome.com.tw/users/20103753/ironman/1590>
- <https://kubernetes.io/>
- <https://helm.sh/>