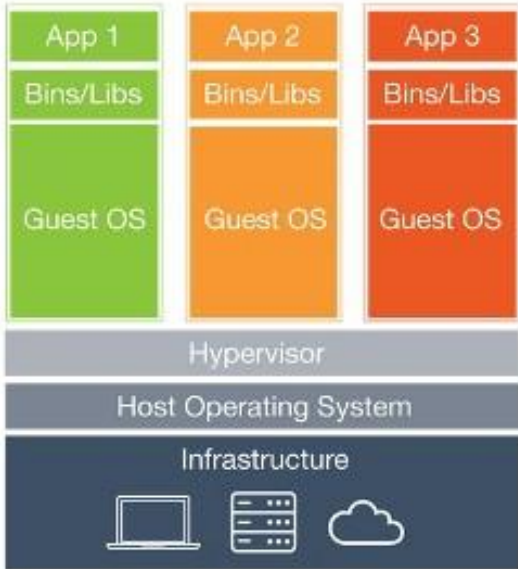


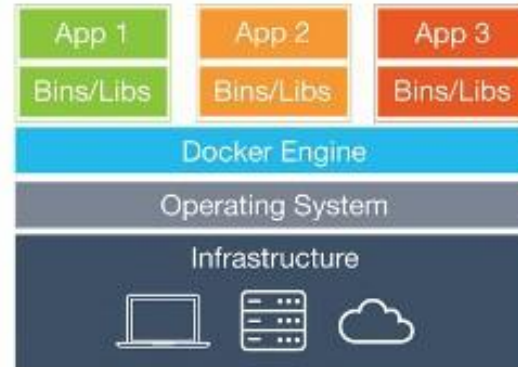
Introduction to MECAs

โดยทีมวิจัยนวัตกรรมอินเทอร์เน็ต
(Internet Innovation Research Team, INO)

Container & Docker Review



Virtual Machines



Containers

Container Benefit

- Isolation : สามารถแบ่งทรัพยากร สำหรับการทำงานแต่ละส่วน
- Standard : มีกลุ่มมาตรฐานสำหรับ Container Runtime และ Container Image Format
- Unify : สามารถนำไป Run บนที่ใดก็ได้ ไม่ว่าจะถูกสร้างที่ไหน ด้วย OS อะไร
- Portable : โปรแกรมจะถูกจัดเตรียมในรูปแบบของ Container Image
- Extensible : สามารถเริ่มต้นจาก Base Image ที่มีอยู่แล้ว
- Sharing : สามารถแบ่งปัน Container Image ไว้บน Repository



ทบทวน docker ก่อนเริ่มใช้งาน

Application

คือ สิ่งที่เราต้องการจะใช้งาน เช่น Apache, PHP, MySQL, WordPress, Laravel เป็นต้น

Docker image

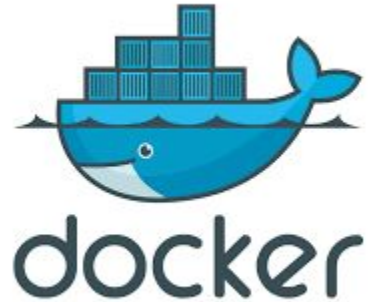
คือ ไฟล์ image ที่บรรจุ application ช่างต้นเอาไว้ โดยไฟล์นี้ได้มาจากการ build ไฟล์ **Dockerfile** ที่ใส่ command ของ Linux สำหรับติดตั้ง application ดังกล่าวเอาไว้

Docker container

คือ กล่องที่บรรจุ application ที่พร้อมทำงาน ซึ่งเป็นผลมาจากการรัน Docker image

Docker Hub

คือ ที่ๆ เราสามารถอัปโหลด image ที่ตัวเองสร้างขึ้นไปเพื่อแบ่งปันให้คนอื่นได้ใช้ด้วย หรือเราจะดาวน์โหลด image ที่คนอื่นทำไว้ดีแล้วมาใช้งานเลยก็ได้



Docker Cheat Sheet

ORCHESTRATE

Initialize swarm mode and listen on a specific interface
`docker swarm init --advertise-addr 10.1.0.2`

Join an existing swarm as a manager node
`docker swarm join --token <manager-token> 10.1.0.2:2377`

Join an existing swarm as a worker node
`docker swarm join --token <worker-token> 10.1.0.2:2377`

List the nodes participating in a swarm
`docker node ls`

Create a service from an image exposed on a specific port and deploy 3 instances
`docker service create --replicas 3 -p 80:80 --name web nginx`

List the services running in a swarm
`docker service ls`

Scale a service
`docker service scale web=5`

List the tasks of a service
`docker service ps web`



RUN

```
docker run
  --rm remove container automatically after it exits
  -it connect the container to terminal
  --name web name the container
  -p 5000:80 expose port 5000 externally and map to port 80
  -v ~/dev:/code create a host mapped volume inside the container
  alpine:3.4 the image from which the container is instantiated
  /bin/sh the command to run inside the container
```

Stop a running container through SIGTERM
`docker stop web`

Stop a running container through SIGKILL
`docker kill web`

Create an overlay network and specify a subnet
`docker network create --subnet 10.1.0.0/24 --gateway 10.1.0.1 -d overlay mynet`

List the networks
`docker network ls`

List the running containers
`docker ps`

Delete all running and stopped containers
`docker rm -f $(docker ps -aq)`

Create a new bash process inside the container and connect it to the terminal
`docker exec -it web bash`

Print the last 100 lines of a container's logs
`docker logs --tail 100 web`

BUILD

Build an image from the Dockerfile in the current directory and tag the image
`docker build -t myapp:1.0 .`

List all images that are locally stored with the Docker engine
`docker images`

Delete an image from the local image store
`docker rmi alpine:3.4`

SHIP

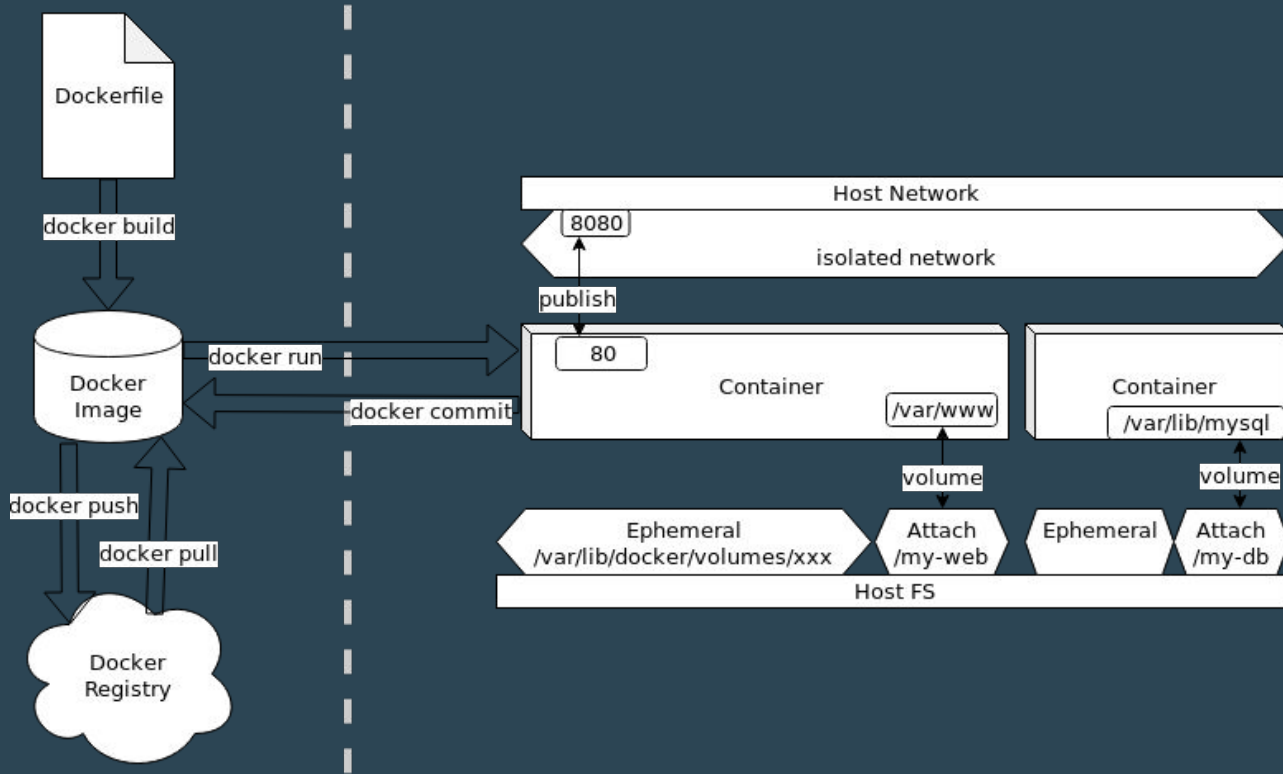
Pull an image from a registry
`docker pull alpine:3.4`

Retag a local image with a new image name and tag
`docker tag alpine:3.4 myrepo/myalpine:3.4`

Log in to a registry (the Docker Hub by default)
`docker login my.registry.com:8000`

Push an image to a registry
`docker push myrepo/myalpine:3.4`

Container Development Process



Public Cloud Services

- <https://aws.amazon.com/>
- <https://cloud.google.com/>
- <https://azure.microsoft.com/>

AWS Overview

- Begin on 2006 power by Amazon
- As of 2017, AWS owns a dominant 34% of all cloud.
- Over 140 AWS services are available including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications
- The AWS Cloud operates 60 Availability Zones within 20 geographic Regions around the world

Container on AWS

- Amazon Elastic Container Service <https://aws.amazon.com/ecs/>
- AWS Fargate <https://aws.amazon.com/fargate/>
- Amazon Elastic Container Service for Kubernetes
<https://aws.amazon.com/eks/>
- Amazon Elastic Container Registry <https://aws.amazon.com/ecr/>



Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only

Resources to be created:

Cluster
VPC (optional)
Subnets (optional)

Powered by AWS Fargate

EC2 Linux + Networking

Resources to be created:

Cluster
VPC
Subnets
Auto Scaling group with Linux AMI

EC2 Windows + Networking

Resources to be created:

Cluster
VPC
Subnets
Auto Scaling group with Windows AMI

*Required

Cancel

Next step

The screenshot shows the AWS console interface for creating a new task definition. The main window is titled "Create new Task Definition" and is divided into two steps: "Step 1: Select launch type compatibility" and "Step 2: Configure task and container definitions". The "Add container" dialog box is open, showing the "Standard" configuration. The "Container name*" field is empty. The "Image*" field contains the placeholder text "repository-uri/image:tag" and a help icon. Below the image field, the custom image format is defined as "[registry-uri]/[namespace]/[image]:[tag]". The "Private repository authentication*" checkbox is unchecked. The "Memory Limits (MiB)" section shows a "Soft limit" dropdown set to "128" and an "Add Hard limit" link. A detailed explanation of memory limits is provided below. At the bottom of the dialog, there is a "* Required" label and "Cancel" and "Add" buttons.

ECS Cluster Provisioning & Create Task Definition

Google Cloud Overview

- In April 2008, Google announced App Engine
- GCP has market shares around 5%.
- Google lists over 90 products under the Google Cloud brand.
- Google Cloud Platform is available in 19 regions and 58 zones.

Container on GCP

- Google Kubernetes Engine <https://cloud.google.com/kubernetes-engine/>
- Container Registry <https://cloud.google.com/container-registry/>



เทมเพลตคลัสเตอร์

เลือกเทมเพลตที่มีการตั้งค่าที่กำหนดค่าไว้ล่วงหน้า หรือ ปรับแต่งเทมเพลตให้เหมาะกับความต้องการของคุณ

โคลนคลัสเตอร์ที่มีอยู่
เลือกคลัสเตอร์ที่มีอยู่อย่างน้อย 1 รายการ เพื่อเพิ่มข้อมูลในฟิลด์

คลัสเตอร์มาตรฐาน
การรวมอย่างต่อเนื่อง, บริการเว็บ, แบ็กเอนด์ ซึ่งเป็นทางเลือกที่ดีที่สุดสำหรับการปรับแต่งเพิ่มเติม หรือหากไม่แน่ใจว่าจะเลือกอะไร

คลัสเตอร์แรก
การทดสอบกับ Kubernetes Engine กับ การทำให้แอปพลิเคชันแรกใช้งานได้ เป็นทางเลือกที่เหมาะสมสำหรับการเริ่มต้น

แอปพลิเคชันที่ใช้ CPU มาก
การรวบรวมข้อมูลเว็บหรือทุกสิ่งที่ต้องใช้ CPU มากขึ้น

แอปพลิเคชันที่ใช้หน่วยความจำ มาก
ฐานข้อมูล, การวิเคราะห์ หรืออย่างเช่น Hadoop, Spark, ETL หรือสิ่งอื่นๆ ที่ต้องการหน่วยความจำมากขึ้น

การประมวลผลแบบเร่งการทำงานของ GPU
แมชชีนเรียนรู้, การแปลงวิดีโอ, การคำนวณทางวิทยาศาสตร์หรือสิ่งอื่นๆ ที่ต้องใช้การประมวลผลอย่างหนักและอาจใช้ GPU

เทมเพลต "คลัสเตอร์มาตรฐาน"

การรวมอย่างต่อเนื่อง, บริการเว็บ, แบ็กเอนด์ ซึ่งเป็นทางเลือกที่ดีที่สุดสำหรับการปรับแต่งเพิ่มเติม หรือหากไม่แน่ใจว่าจะเลือกอะไร

ระบบจะเรียกข้อมูลเพิ่มเติม

หลังจากสร้างคลัสเตอร์แล้วจะทำการเปลี่ยนแปลงบางอย่างไม่ได้ ลองเลื่อนเมาส์ไปทางเหนือคอนโวลิวส์เพื่อดูข้อมูลเพิ่มเติม

ชื่อ

ประเภทตำแหน่งที่ตั้ง โอน ภูมิภาค

โซน

เวอร์ชันหลัก

Node Pool
Node Pool คือกลุ่มอินสแตนซ์หลายกลุ่มที่เรียกใช้ Kubernetes ในคลัสเตอร์ คุณอาจเพิ่ม Node Pool ในโซนต่างๆ เพื่อให้มีความพร้อมใช้งานสูง หรือเพิ่ม Node Pool ของเครื่องประเภทต่างๆ ได้ คุณจะไม่เพิ่ม Node Pool ได้โดยคลิกแก้ไขดูข้อมูลเพิ่มเติม

default-pool
จำนวนโหนด

ประเภทเครื่อง หน่วยความจำ 3.75 GB

การอัปเกรดอัตโนมัติ: เปิด

- คลัสเตอร์
- ภาระงาน**
- บริการ
- แอปพลิเคชัน
- การกำหนดค่า
- พื้นที่เก็บข้อมูล

1 คอนเทนเนอร์

แก้ไขคอนเทนเนอร์

- อิมเมจคอนเทนเนอร์ที่มีอยู่
- อิมเมจคอนเทนเนอร์ใหม่

เส้นทางอิมเมจ *

ป้อนเส้นทางของอิมเมจ หรือเลือกจาก Google Container Registry หรือคุณอาจลองทำให้ใช้งานได้ด้วยอิมเมจ nginx อย่างเป็นทางการ nginx:latest

ตัวแปรของสภาพแวดล้อม

คำสั่งเริ่มต้น

ปลั๊ก entypoint เริ่มต้นของอิมเมจคอนเทนเนอร์

2 การกำหนดค่า

Kubernetes Cluster Provision & Deploy Workload

Azure Overview

- Microsoft Azure launch on 2010
- Azure has 11% market shares
- Microsoft lists over 600 Azure services
- Azure has 54 regions worldwide

Container on Azure

- Azure Kubernetes Service (AKS)

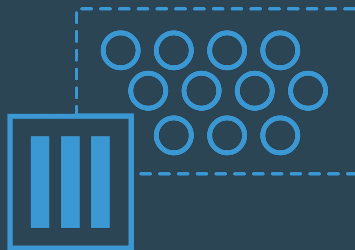
<https://azure.microsoft.com/en-us/services/kubernetes-service/>

- Container Instances

<https://azure.microsoft.com/en-us/services/container-instances/>

- Container Registry

<https://azure.microsoft.com/en-us/services/container-registry/>




Kubernetes services

ad (Default Directory)

+ Add Edit columns More

Filter by name...

NAME



No Kubernetes services to display

Use Azure Kubernetes Service to create and manage Kubernetes clusters. Azure will handle cluster operations, including creating, scaling, and upgrading, freeing up developers to focus on their application. To get started, create a cluster with Azure Kubernetes Service. [Learn more](#)

Create Kubernetes service

Create Kubernetes cluster

Basics Authentication Networking Monitoring Tags Review + create

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. [Learn more about Azure Kubernetes Service](#)

PROJECT DETAILS

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription Visual Studio Enterprise with MSDN

* Resource group Default-Storage-SoutheastAsia [Create new](#)

CLUSTER DETAILS

* Kubernetes cluster name

* Region Southeast Asia

* Kubernetes version 1.12.6

* DNS name prefix

SCALE

The number and size of nodes in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. You will not be able to change the node size after cluster creation, but you will be able to change the number of nodes in your cluster after creation. [Learn more about scaling in Azure Kubernetes Service](#)

* Node size **Standard DS2 v2**
2 vcpus, 7 GB memory
[Change size](#)

Review + create Previous Next: Authentication >

Create Container Instances

- 1 Basics
Configure basic settings >
- 2 Configuration
Specify container requirements >
- 3 Summary
Container Instances >

Basics

* Container name

Container image type
Public Private

* Container image

Subscription
Visual Studio Enterprise with MSDN

* Resource group
Select existing...
[Create new](#)

* Location
Southeast Asia

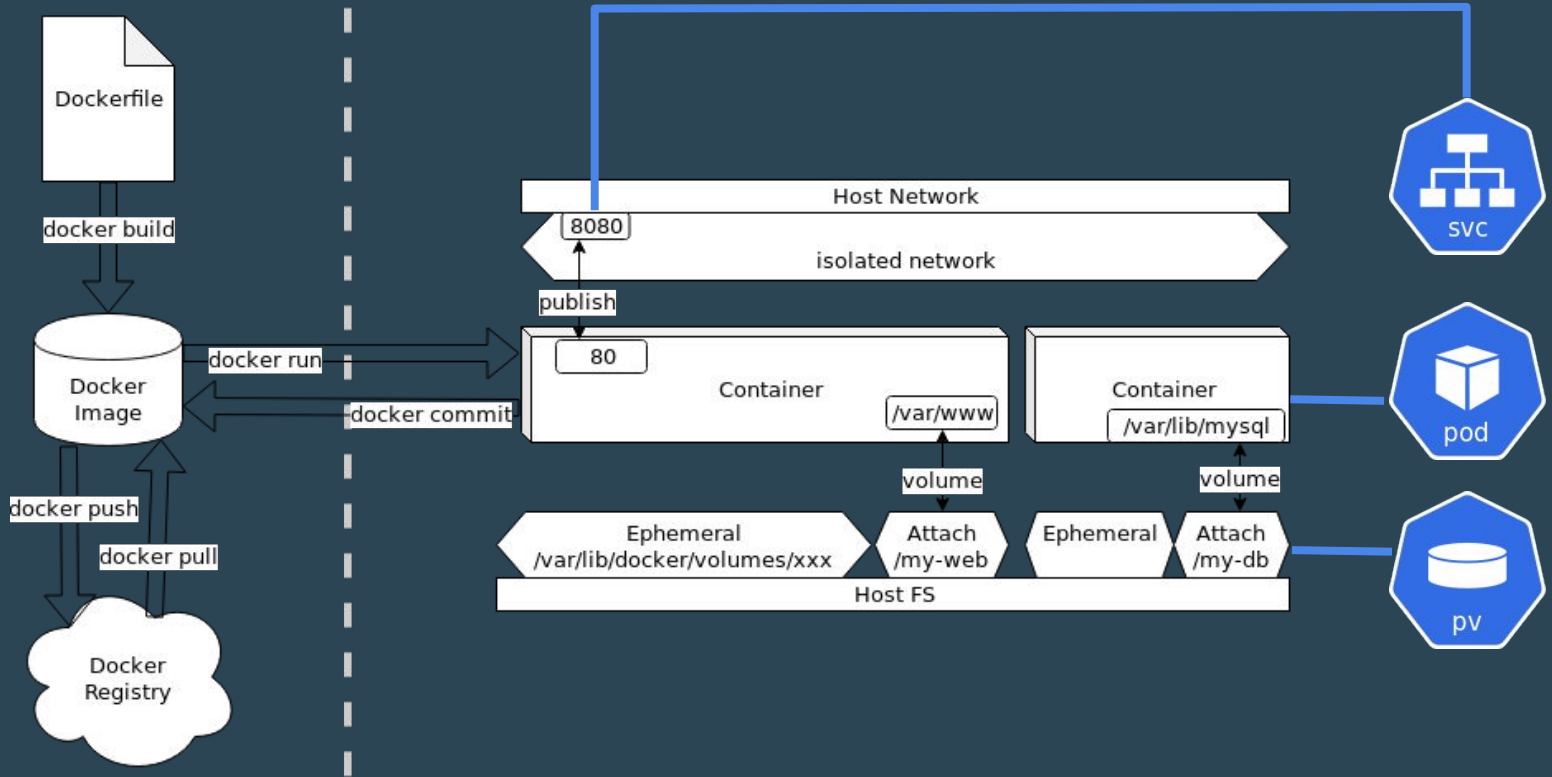
AKS Cluster Provisioning & Container Instances

Kubernetes

Google open-sourced the Kubernetes project in 2014. Kubernetes builds upon a decade and a half of experience that Google has with running production workloads at scale, combined with best-of-breed ideas and practices from the community.

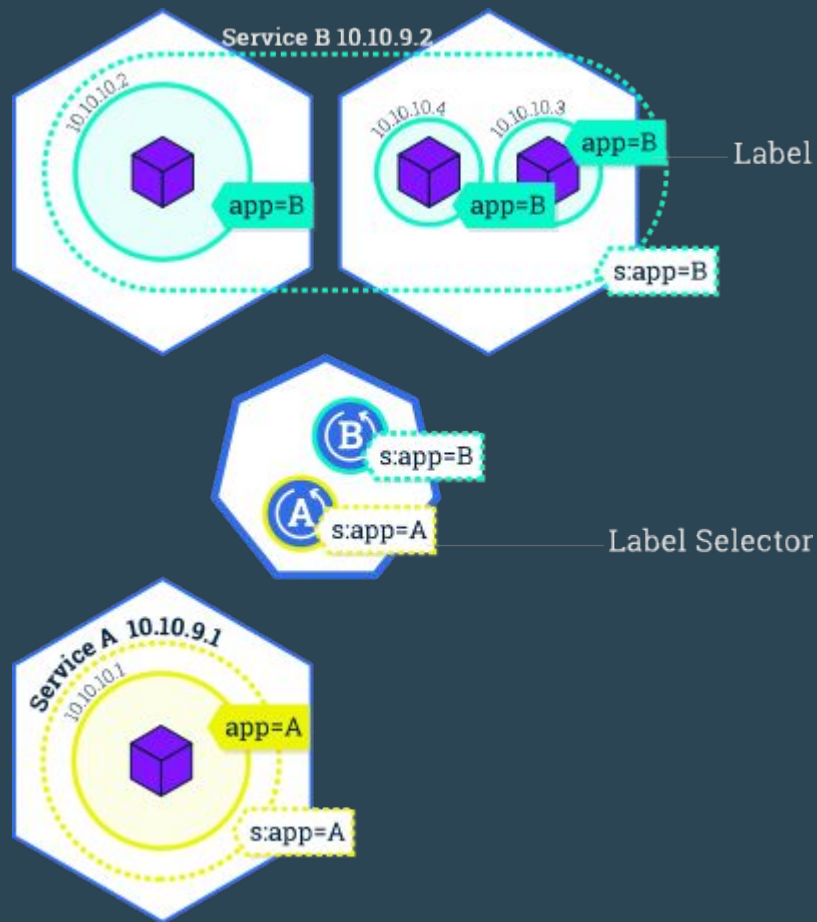


Container VS Kubernetes



Concept

- รูปหกเหลี่ยม หมายถึง node
- รูปลูกบาศก์ หมายถึง container
- รูป วงกลม แล้วมีลูกศรวนรอบ หมายถึง Deployment
- เส้นทึบ หมายถึง Pod
- เส้นประ หมายถึง Service



Resources

Compute



Pod

เป็นหน่วยที่เล็กที่สุดในการจัดการด้วย Kubernetes เปรียบเสมือนเครื่องคอมพิวเตอร์ 1 เครื่อง โดยจะมีทรัพยากรเป็นของตัวเอง ซึ่งประกอบด้วย Container ตั้งแต่ 1 container ขึ้นไปทำงานอยู่ภายใน



Deployment

เป็นส่วนที่ดูแล Life cycle ของ Pod โดย Kubernetes จะดูแลให้ การทำงานของ Pod เป็นไปตามที่ระบุไว้ใน Spec เช่น จำนวน Relicas, Health check, Restart

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

Network



Service

เป็นช่องทางในติดต่อผ่านเครือข่าย ซึ่ง Kubernetes จะสร้าง Cluster IP ที่สามารถติดต่อได้ทั้ง Cluster และสามารถทำการค้นพบได้ผ่าน DNS โดยการ ใช้ selector ผ่าน label



Network Policy

สามารถกำหนดการเข้าถึง Service ด้วยการกำหนดกฎของเครือข่าย

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: nginx
  externalIPs: ["203.185.64.1"]
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Storage



Persistent Volume

เป็นพื้นที่จัดเก็บข้อมูลถาวร ที่สร้างขึ้นโดยอาจจะใช้ storage backend รูปแบบต่างๆ ซึ่งสร้างโดยผู้ดูแลระบบ หรือสร้างโดยอัตโนมัติผ่าน Persistent Volume Claim และ Storage Class



Persistent Volume Claim

เป็นคำขอสร้าง พื้นที่จัดเก็บข้อมูล โดยระบบจะไปจับคู่กับ Persistent Volume ที่มีอยู่ หรือสร้างขึ้นมาใหม่



Storage Class

ชนิดของพื้นที่จัดเก็บข้อมูล ซึ่งถูกใช้ในการสร้างพื้นที่จัดเก็บข้อมูลตามที่ ร้องขออัตโนมัติ

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: standard
```

Config



ConfigMap

จัดเก็บข้อมูลแบบ Key - Value สำหรับข้อมูลที่
ไม่ใหญ่มากนัก และสามารถอ่านได้จากหลาย
Pod โดยสามารถนำไปใช้ผ่านการ Mount
เป็นไฟล์ หรือส่งเป็น Environment Variable
ใน Pod ได้



Secret

ใช้จัดเก็บข้อมูลเช่นเดียวกับ ConfigMap แต่
ภายใน Kubernetes จะมีการจัดการแบบพิเศษ
เพื่อเก็บรักษาข้อมูลที่เป็นความลับ

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: my-conf
data:
  my.ini: |
    [client]
    port = 3306
    socket = /var/run/mysqld/mysql.sock
```


Break

MECAs: Cloud Container Service Platform



Build



Launch

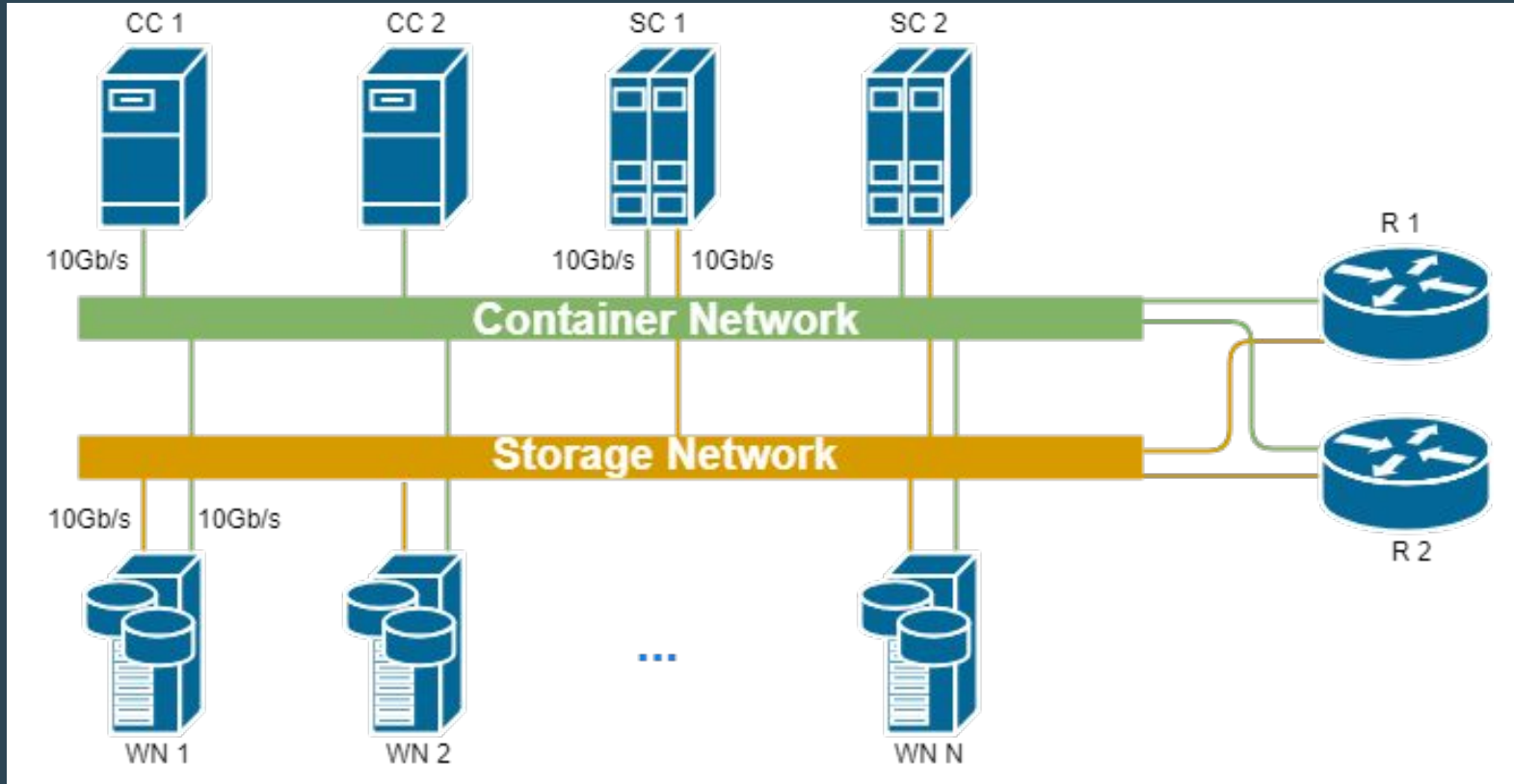


Serve

MECAs Features

- Containerized applications powered by Kubernetes
- Getting start with MECAs UI
- Storage volume with data replication over entire cluster
- HTTP/HTTPS service with dynamically assigned subdomain name
- Customized domain name for HTTP/HTTPS service
- Resource allocation for each namespace
- Built-in network load balancer

MECAs: Network Topology



MECAs: Components

- Container Controller (CC)
 - ควบคุมการสร้าง container, load balancer บน worker node
- Storage Controller (SC)
 - ควบคุมการจัดเก็บและสำเนาข้อมูลโดยใช้ disk บน worker node และควบคุมการเข้าถึงข้อมูล
- Worker Node (WN)
 - รัน container ตามคำสั่งของ container controller และจัดเก็บข้อมูลตามคำสั่งของ storage controller
- Router (R)
 - peer BGP routing กับ worker node (load balancer) เพื่อประกาศ Public IP ออกไป

MECAs Quota

- CPU: 4 cores (default: 200m core per container)
- Memory: 32 GB (default: 256 MB per container)
- Storage
 - Block Storage 2 replica: 400 GB
 - Block Storage 3 replica: 100 GB
 - File System (based on Block Storage)
- Portset (Public IP & Port)
 - Public IP address 1 address (สามารถขอเพิ่มเติมได้ในภายหลัง)

MECAs Portset

- IP และ Port numbers ที่ได้รับจัดสรร สามารถเปิดบริการได้หลากหลาย
- ตัวอย่างการใช้งาน Portset

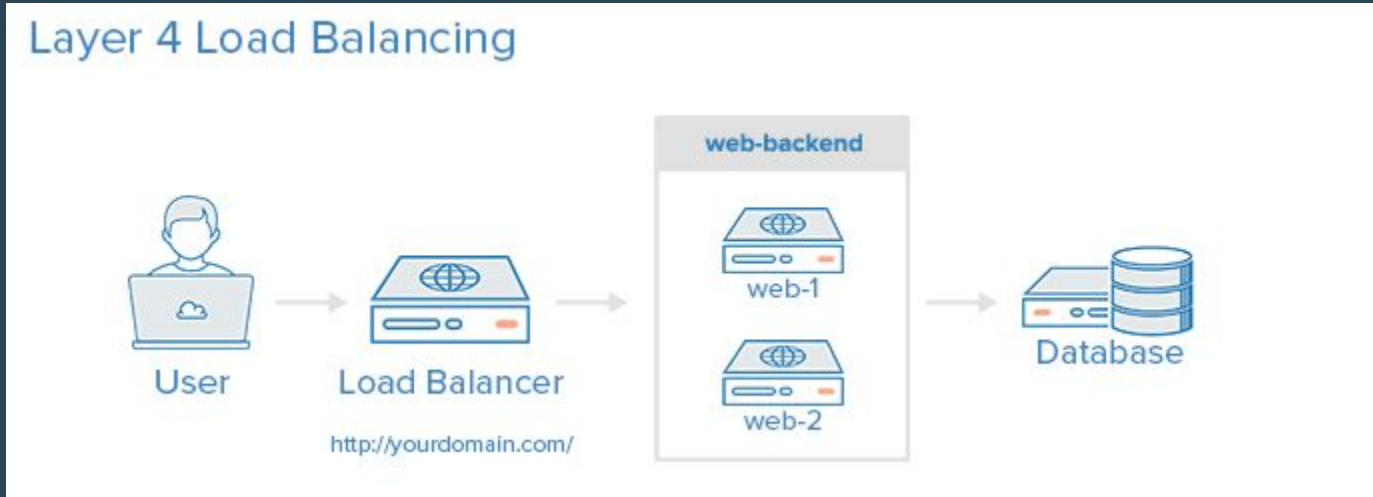
```
[root@dev ~]# kubectl get service -n wot
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT (S)	AGE
api	ClusterIP	10.14.226.14	<none>	80/TCP	2d
api-dns	ClusterIP	10.5.53.5	<u>203.185.64.254</u>	<u>53/UDP, 53/TCP</u>	7d
mariadb	ClusterIP	10.2.111.59	<none>	3306/TCP	10d
myadmin	ClusterIP	10.13.107.145	<none>	80/TCP	10d
relay-t1	ClusterIP	10.10.161.252	<u>203.185.64.254</u>	<u>443/TCP, 80/TCP, 8443/TCP</u>	7d
relay-t2	ClusterIP	10.15.109.36	203.185.64.253	443/TCP, 80/TCP, 8443/TCP	7d

- จะเห็นว่าหมายเลข 203.185.64.254 ถูกใช้งานหลาย Port
 - UDP 53, TCP 53 is used for DNS server
 - TCP 80, 443, 8443 is used for web server

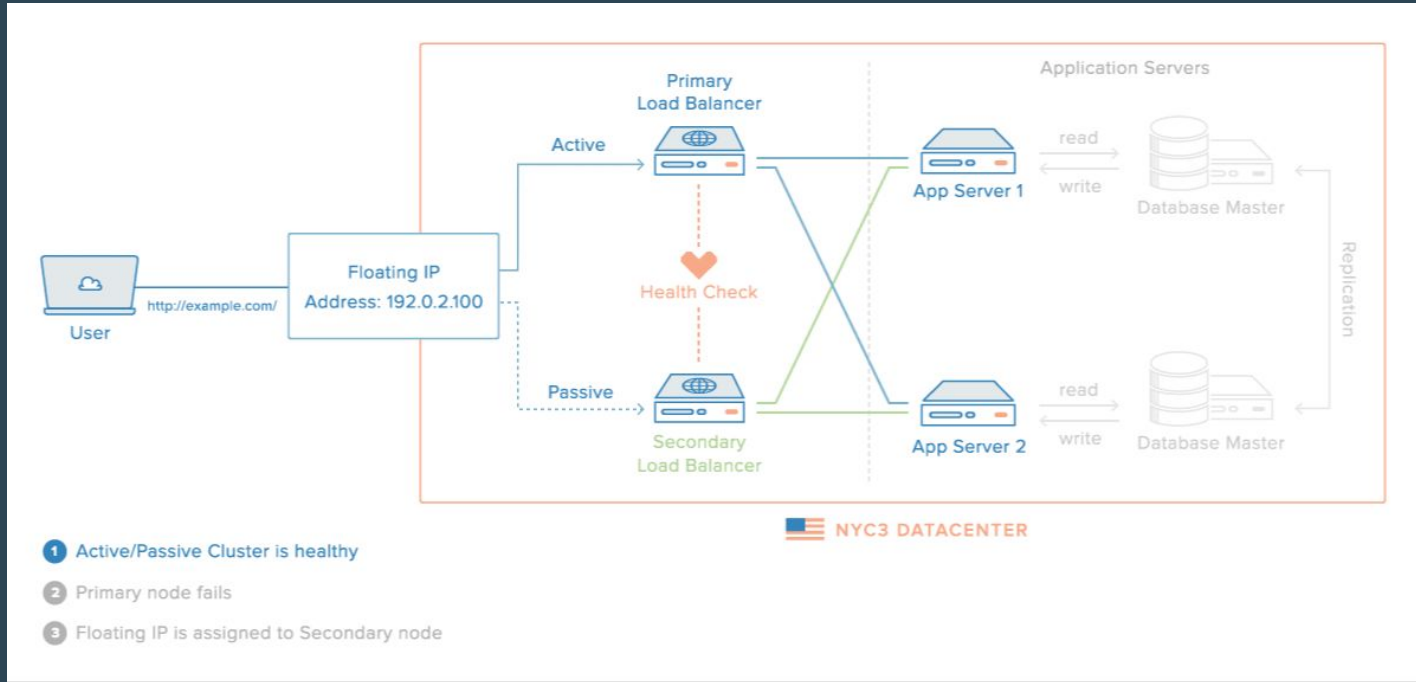
Load balancer

- Single load balancer



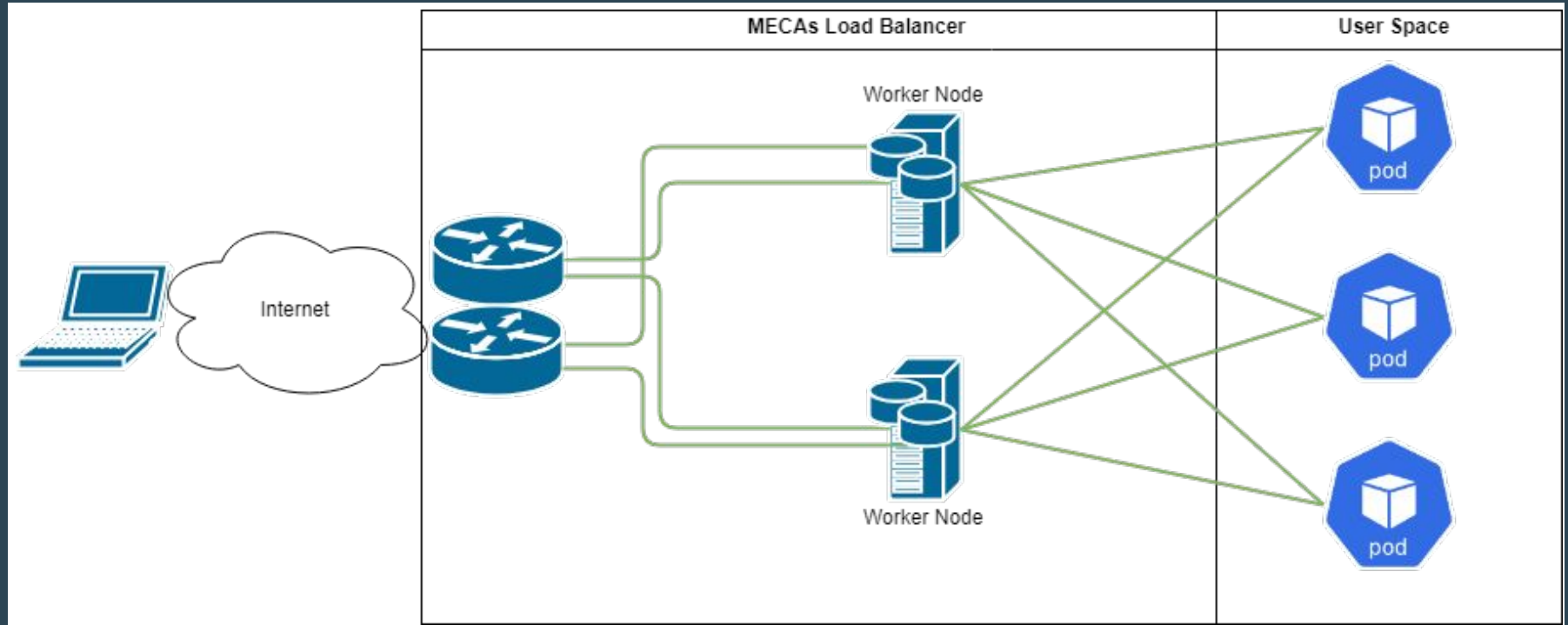
Source: <https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>

Load balancer with Active/Passive



MECAs Load balancer

Service ของ MECAs มาพร้อมกัับ Load balancer แบบ Active/Active



MECAs Workloads

Workload

Workload Name

Deployment

****UNNAMED DEPLOYMENT****

Add new Deployment

Service

Add new Service

Config Map

Persistent Volume Claim

Deployment

Deployment Name

Replicas

Image pull secrets

Containers

****UNNAMED CONTAINER****

General

Container Name

Image Name

MECAs Service Portal

You are invited to enjoy MECAs services.



Log In

Username or email

Password

[Forgot Password?](#)

Log In

New user? [Register](#)

mecas

<https://portal.meca.in.th>

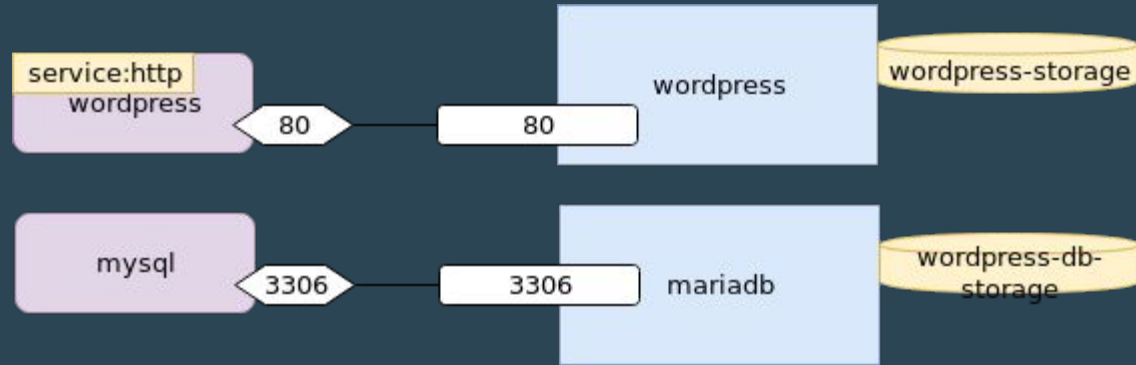
Lunch

Hand On

Example Application: Static web application



Example Application: Web application with database



Example Application: IoT Stack

