

K8s Workshop

葉信和 / Hsin-Ho Yeh
Software Engineer / CEO @ 信誠金融科技
hsinho.yeh@footprint-ai.com

About me

- 2020 - Present at 信誠金融科技
 - Tech solution provider for financial sectors
 - Deepselling: A deep analytics platform for ecommerce
 - Tintin: Everyone-can-use machine learning platform
- 2016 - 2020 at IglooInsure (16M+ in series A+ 2020)
 - Provide digital insurance for e-economic world
 - Funded in KUL, Headquartered in Singapore
 - First employee/ Engineering Lead / Regional Head/ Chief Engineer
- 2013 - 2016 at Studio Engineering @ hTC
 - Principal Engineer on Cloud Infrastructure Team
- 2009 - 2012 at IIS @ Academia Sinica
 - Computer vision, pattern recognition, and data mining
- CS@CCU, CS@NCKU alumni



Agenda

- Pre-requirement
- Why Kubernetes
- What is Container
- What is Kubernetes
- Hands-on session
- QA

Pre-requirement

- Be comfortable with UNIX command line
 - Navigating directories with ``cd`` or ``tree``
 - Editing files, like ``vim``, ``nano``
 - Bash scripting, like env or looping
- Be an expert with ``Google``
 - <https://letmegooglethat.com/?q=you+can+google+it>
- It is totally OK if you don't know what is Docker and Kubernetes

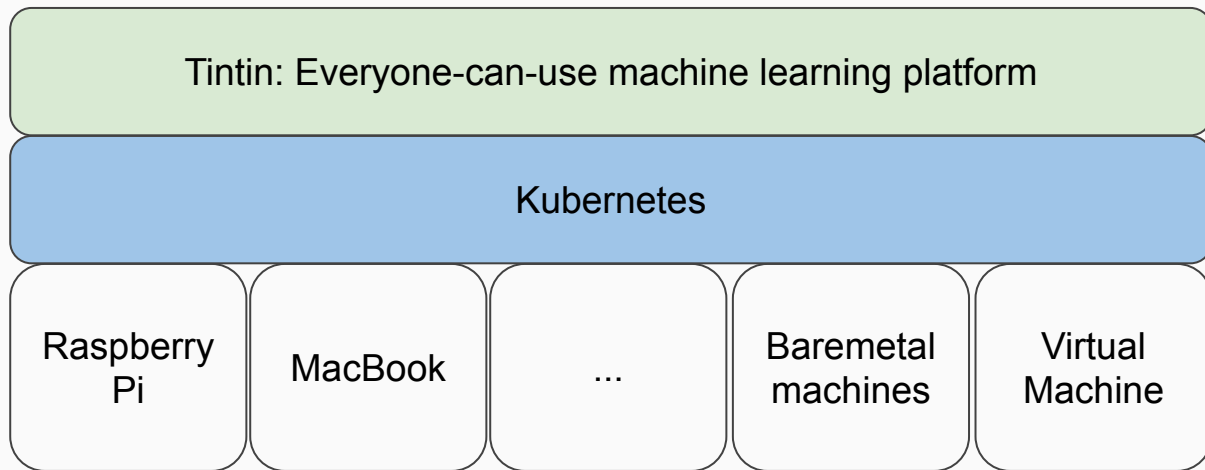
荀子《儒效篇》

「不聞不若聞之，聞之不若見之，見之不若知之，知之不若行之；學至于行之而止矣。」

Why Kubernetes?

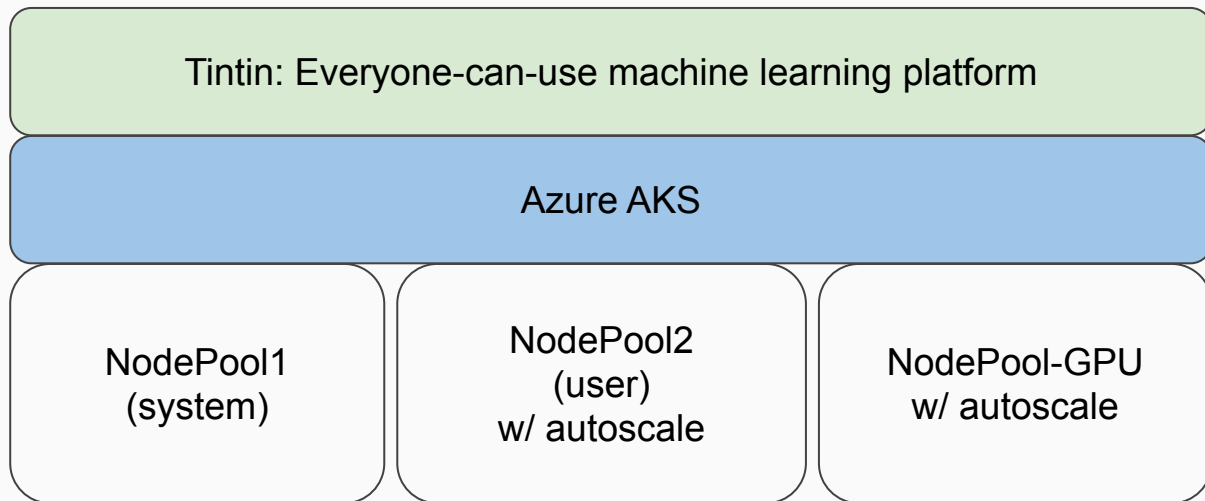
Why Kubernetes?

- Abstraction, abstraction, abstraction.



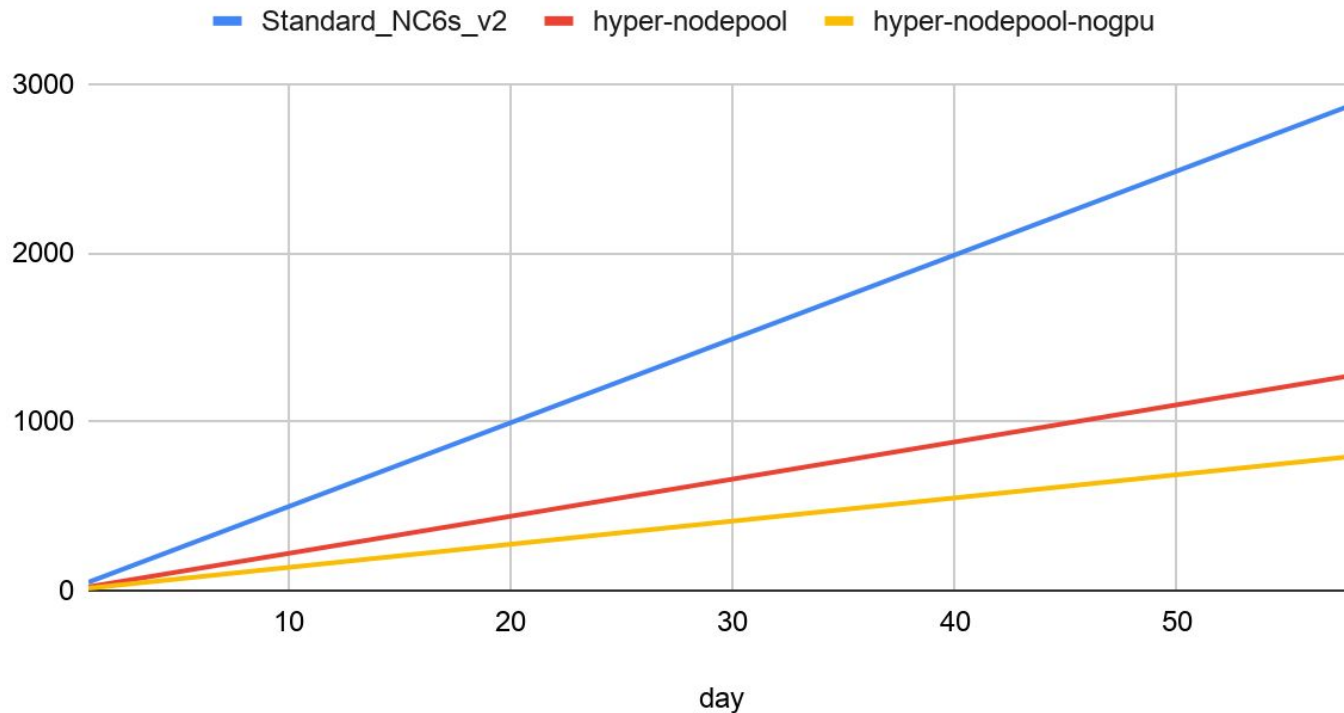
Case study: Hyper-nodepool deployment solution

- It also could be a nodepool w/o autoscale



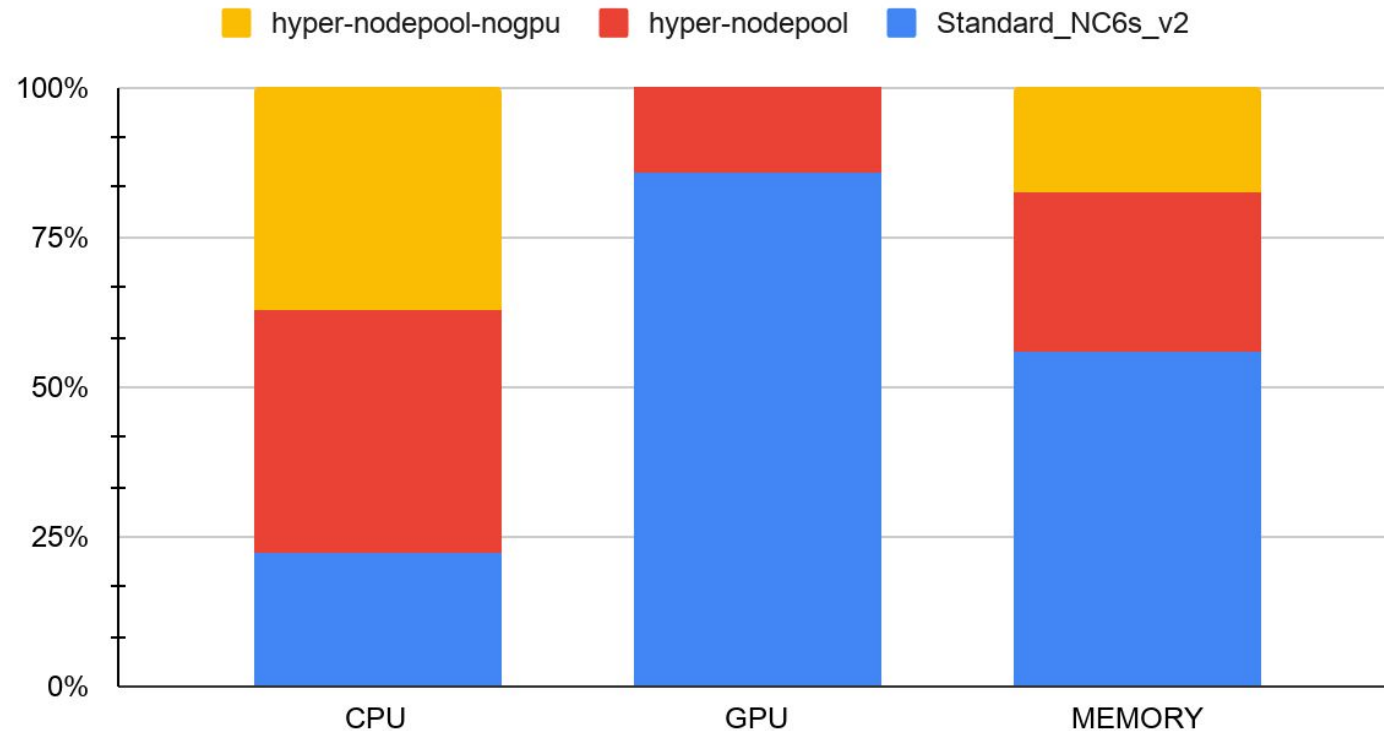
Case study: Hyper-nodepool deployment solution - Cost analysis

Cost comparison (\$:USD)



Case study: Hyper-nodepool deployment solution - Resource Comparison

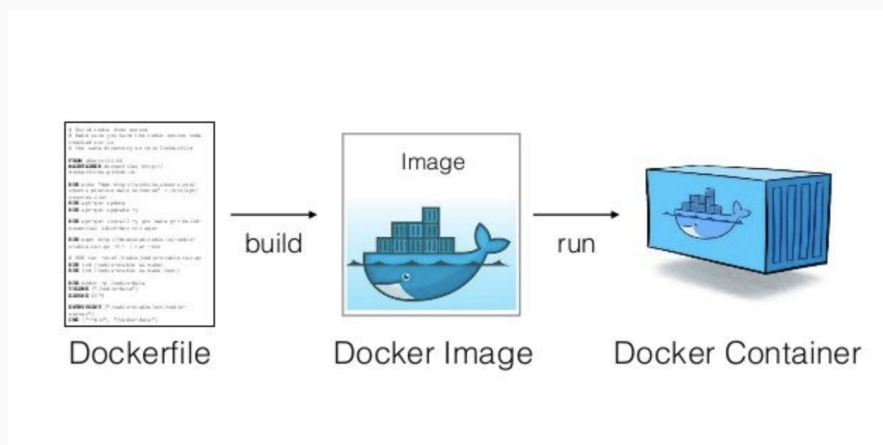
Resource Comparison



What is Container?

What is Container?

- Container
 - Container Image = Application code + dependencies
 - Runtime environment (cgroups, namespaces, env vars)
- Container Registry
 - Container repository



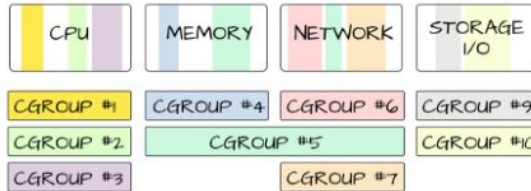
Ref: <https://medium.com/platformer-blog/practical-guide-on-writing-a-dockerfile-for-your-application-89376f88b3b5>

How container works?

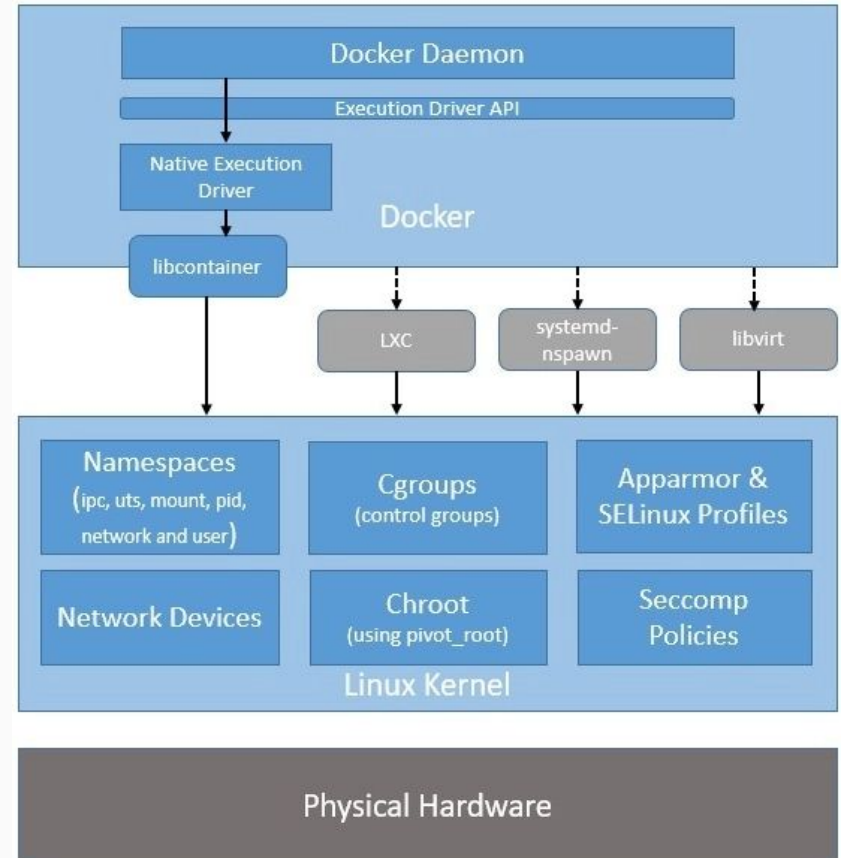
- Namespace for isolation
- Cgroups for resource limiting

Cgroups : Isolation and accounting

- cpu
- memory
- block i/o
- devices
- network
- numa
- freezer



Ref: <https://www.baeldung.com/linux/docker-containers-evolution>
<https://medium.com/@BeNitinAgarwal/understanding-the-docker-internals-7ccb052ce9fe>



What is Dockerfile?

- A dockerfile contains instructions needed to build a given image

```
FROM ubuntu:18.04
```

```
RUN apt-get update && apt-get install -y build-essential
```

```
COPY . /app
```

```
RUN make /app
```

```
CMD python /app/app.py
```

Install Docker runtime

- Install Docker on your host machine
 - <https://docs.docker.com/engine/install/ubuntu/>
- Or run the following command for quick installation

```
// installation ...  
curl -fsSL https://get.docker.com -o get-docker.sh  
  
sh get-docker.sh  
  
// check your installation by run  
sudo docker ps
```

```
docker run -it ubuntu:20.04
```

```
root> apt-get update
```

```
root> apt-get install -y lsb-release
```

```
root> lsb_release -a
```

```
Distributor ID: Ubuntu
```

```
Description:  Ubuntu 20.04.2 LTS
```

```
Release:      20.04
```

```
Codename:     focal
```


// port mapping

docker run -it -p 8080:8080 ubuntu:20.04

// mount host folder to container

docker run -it -v /host-folder:/container-folder ubuntu:20.04

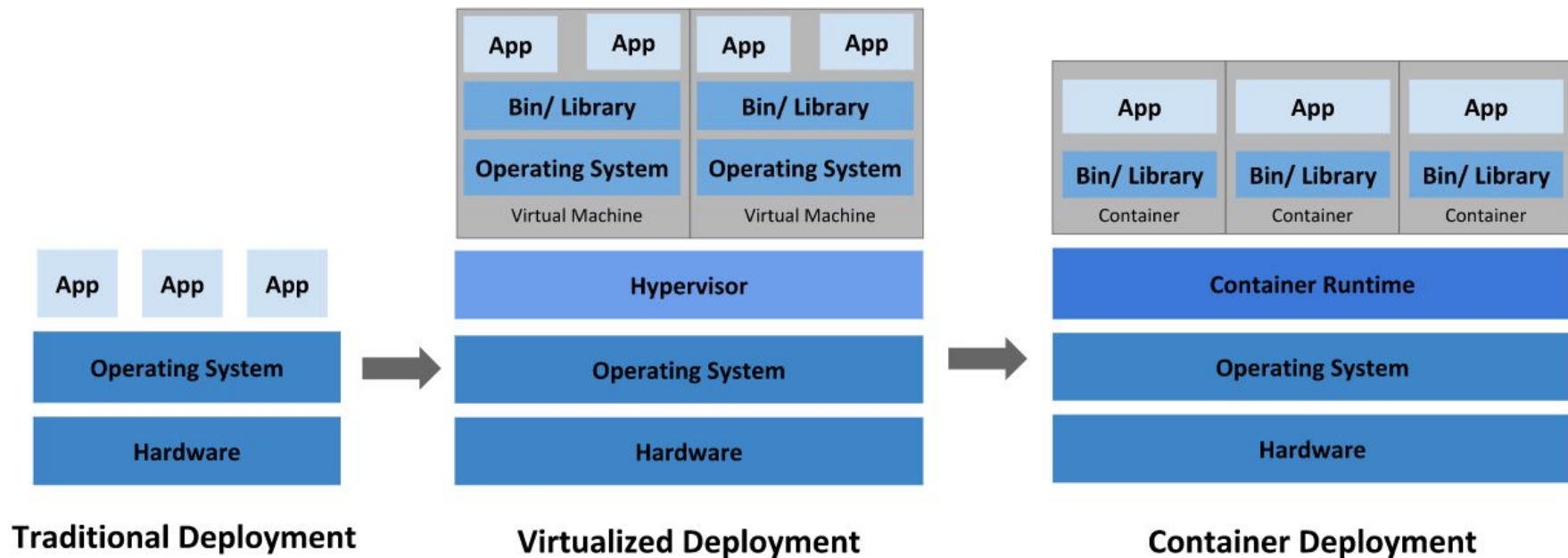
// build a container image

docker build -t footprintai/example:tag -f Dockerfile .

```
git clone https://github.com/FootprintAI/k8s-workshop.git
```

What is Kubernetes?

What is containerized deployment?

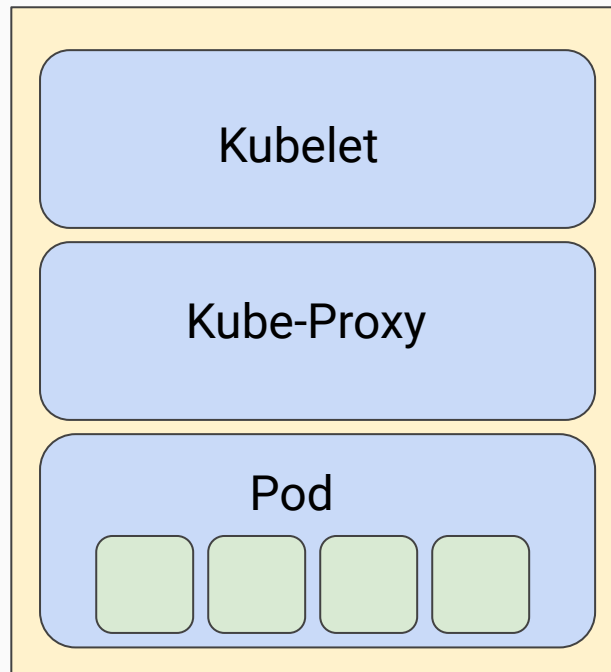


- A portable, extensible, open-source platform for managing **containerized** workloads and services
 - Service Discovery and load balancing
 - Storage Orchestration
 - Automated rollout and rollback
 - Automatic bin packing
 - Self-healing
 - Secret and config management

- High level concepts
 - **Node** are machine that run containerized applications.
 - **Pod** are unit for application workload.
 - **Scheduler** schedules pods to run on nodes.
 - **Replica Set** ensures that a specified number of pod replicas are running at any one time.
 - **Service** is an abstract way to expose an application running on a set of Pods as a network service.

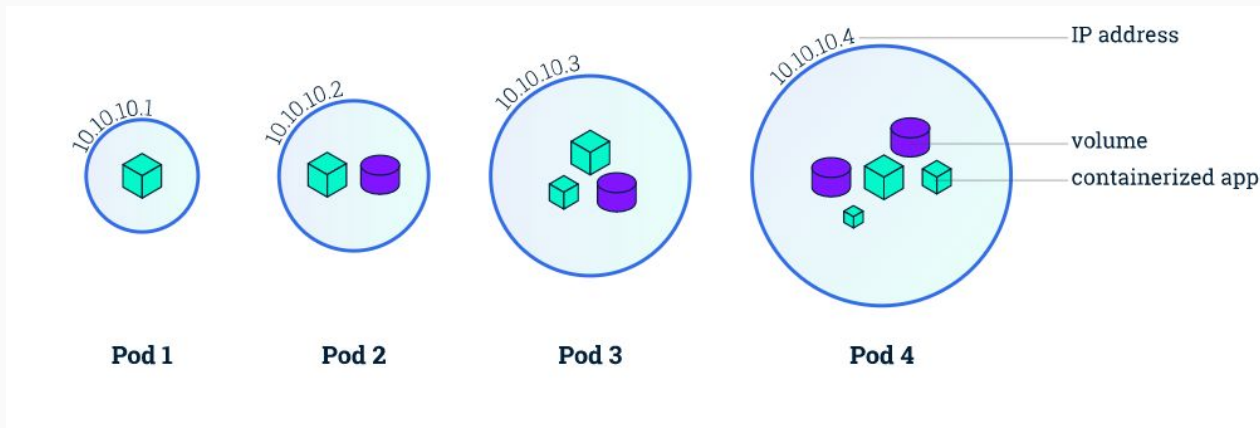
What is a Node?

- **Container runtime**
 - Docker
- **Kubelet**
 - Primary node agent running on each node
 - It register the node with api server and manage pods according to PodSpec.
- **Kube-Proxy**
 - Network proxy runs on each node. This reflects services as defined in the Kubernetes API on each node and can do simple TCP, UDP, and SCTP stream forwarding or round robin forwarding across a set of backends.



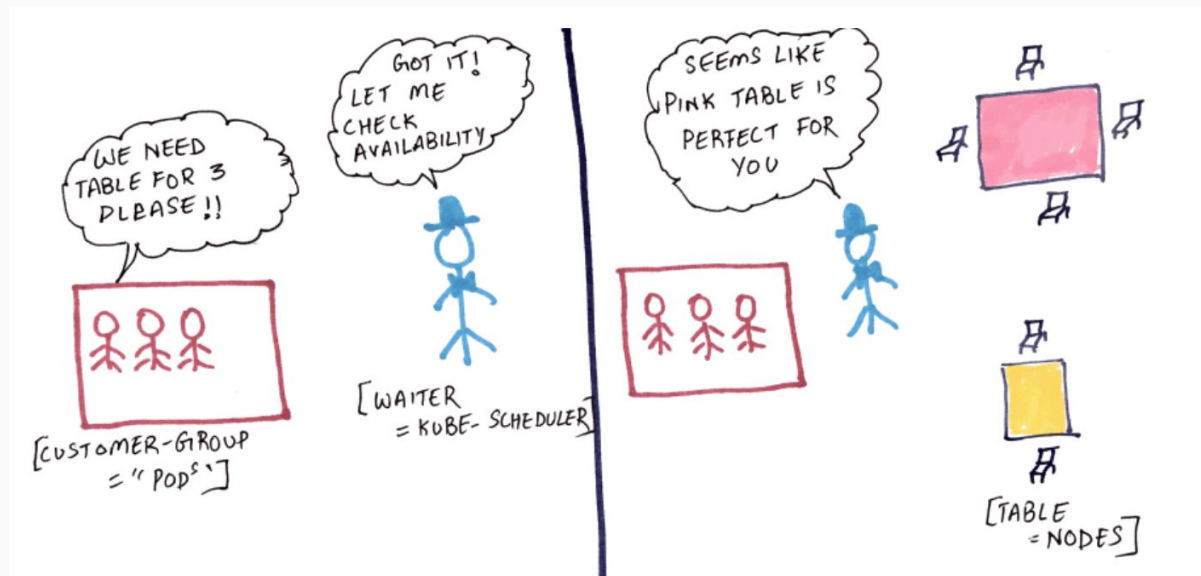
What is a Pod?

- A pod represents a logical application, it could contains a or multiple containers.
- A pod has unique IP address, persistent storage volume, and a configuration on how container should run
- Containers inside the same pod shares namespaces.
 - Containers inside the same pod can locate each other and communicate via localhost



What is a Scheduler?

- The scheduler determines (filtering & scoring) which Nodes are valid placements for each Pod in the scheduling queue according to constraints and available resources.

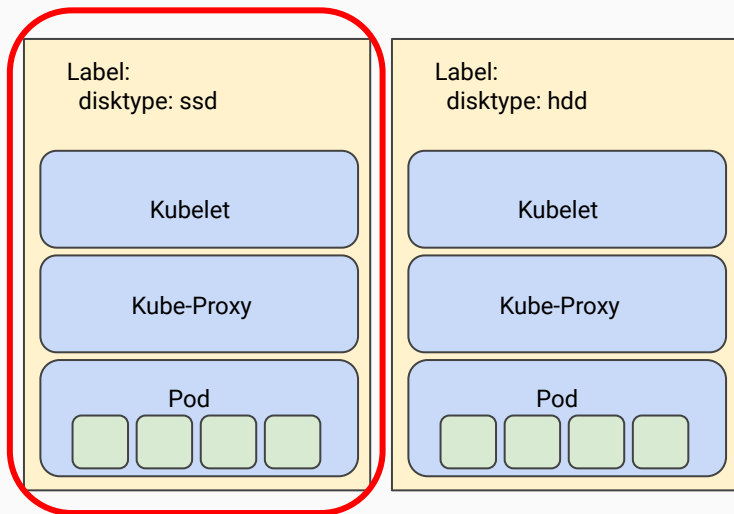


Ref:

<https://day10.com/learn12/kubernetes-scheduler-visualized-in-plain-english-with-a-story-1a4e>

What is a Scheduler?

- nodeSelector
 - Recommended way for node selection constraints
 - Filter by node labels



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
    - name: nginx
      image: nginx
      imagePullPolicy: IfNotPresent
  nodeSelector:
    disktype: ssd
```

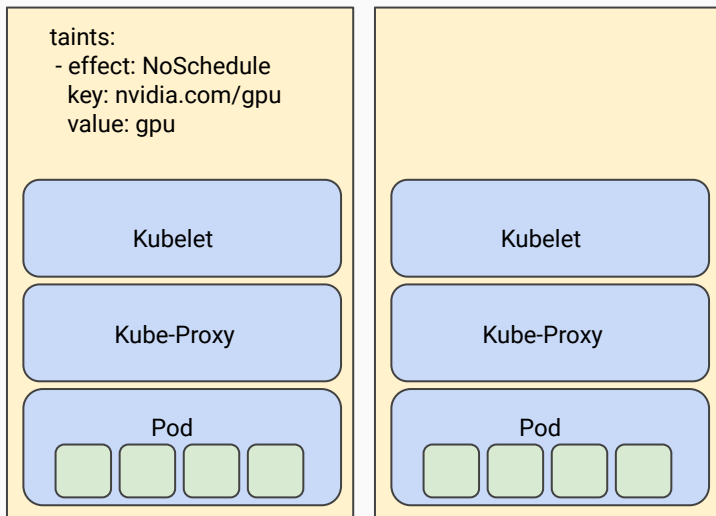
What is a Scheduler?

- Node affinity
 - similar to nodeSelector
 - But offer more fine-grant control

```
apiVersion: v1
kind: Pod
metadata:
  name: with-node-affinity
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: kubernetes.io/e2e-az-name
                operator: In
                values:
                  - e2e-az1
                  - e2e-az2
            preferredDuringSchedulingIgnoredDuringExecution:
              - weight: 1
                preference:
                  matchExpressions:
                    - key: another-node-label-key
                      operator: In
                      values:
                        - another-node-label-value
        containers:
          - name: with-node-affinity
            image: k8s.gcr.io/pause:2.0
```

What is a Scheduler? (1/3)

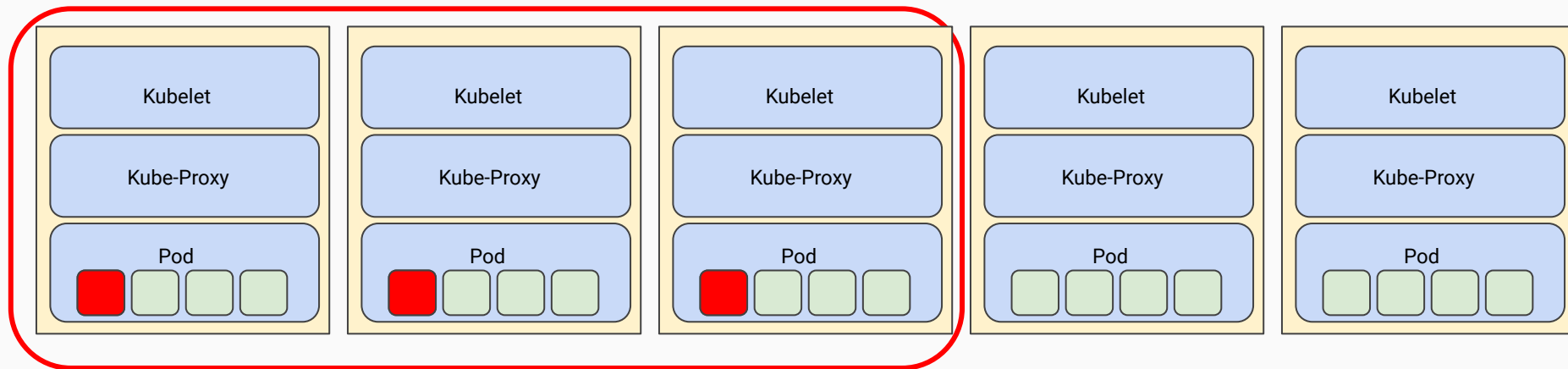
- Taints and tolerations
 - Taint allows a node to repel a set of pods
 - Tolerations allows (but do not require) the pods to schedule onto nodes with matching taints



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
    - name: nginx
      image: nginx
      imagePullPolicy: IfNotPresent
  tolerations:
    - key: "nvidia.com/gpu"
      operator: "Equal"
      value: "gpu"
      effect: "NoSchedule"
```

What is Replica Set?

- Manage a replicated set of pods
- Create pods from a template
- Ensure the desired number of pods running
- Online resizing and self-healing

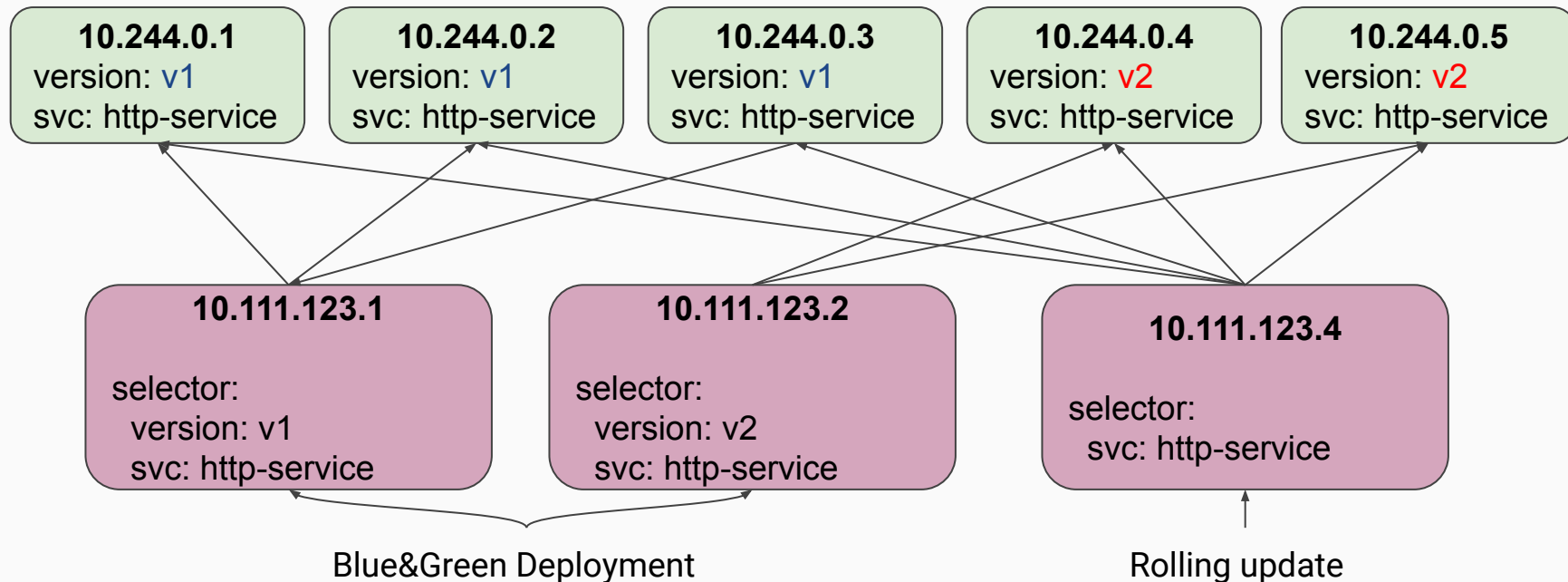


Replica = 3

What is Service?

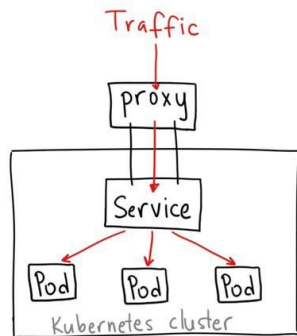
- As pods keeps creating/destroying in the cluster, service provides an abstraction way to expose applications to the world.
 - Service is a proxy runs on each Node,
 - Each service has its own virtual IP.
 - Service use dynamic backend based on label queries
- In-cluster domain name
- Different service type:
 - LoadBalancer
 - Nodeport
 - ClusterIP

What is Service?

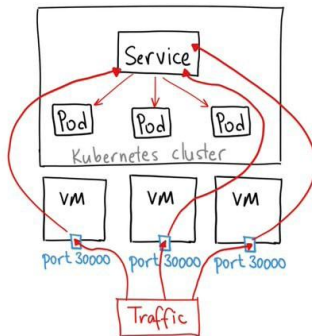
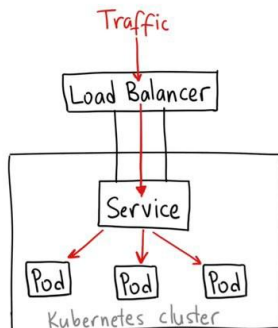


What is Service?

ClusterIP

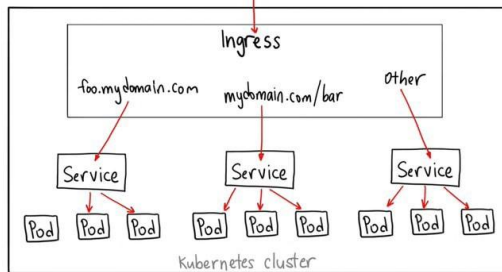


LoadBalancer



NodePort

Traffic



Ingress

Hands-on Session

Install Kubernetes

- Cloud-provider version:
 - GKE(Google kubernetes engine), AWS EKS(Amazon Elastic Kubernetes Service), AKS (Azure Kubernetes Service), ...
- Minikube: local kubernetes
 - <https://minikube.sigs.k8s.io/docs/start/>
- K3s: lightweight kubernetes
 - <https://k3s.io/>
- K3d: containerized k3s cluster
 - <https://github.com/rancher/k3d>

```
// install kubectl, a cli interface for kubernetes v1.20.0
// We use k3d as example, but you can also try other installations.

curl -LO https://dl.k8s.io/release/v1.20.0/bin/linux/amd64/kubectl
mv ./kubectl /usr/local/bin/kubectl

// install k3d
curl -s https://raw.githubusercontent.com/rancher/k3d/main/install.sh | bash

// create k3d cluster
k3d cluster create yuntech-workshop

// use k3d context
kubectl config use-context yuntech-workshop

// check
Kubectl get pods --all-namespaces

// create pseudo pod
kubectl run ubuntu --image=ubuntu:20.04
```

Q&A

One minute takeaway

- Dockerfile
 - Docker container and docker daemon
 - Pod/Service/ReplicaSet concepts
-
- And it is just a beginning ...

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city is densely packed with skyscrapers, many of which are illuminated with their interior lights. The Empire State Building is prominent in the center, with its top lit in red and green. The Hudson River is visible in the background, with the New York-New York Hotel & Casino's Big Apple Wheel visible on the right side.

Thank You
Questions?

Contact Us:
partnership@footprint-ai.com

- Documentations

- <https://kubernetes.io/docs/home/>
- <https://docs.docker.com/>

- Kubectl cheatsheet

- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

- Tutorials

- <https://qconuk2019.container.training>