The background of this section shows a photograph of Mahidol University buildings at sunset, with a clear sky and greenery in the foreground. Overlaid on the image is a white rectangular box containing text and the university's logo.

มหาวิทยาลัยมหิดล  
Mahidol University

ห้ามมิให้ผู้เรียนทำซ้ำ ดัดแปลง หรือใช้ประโยชน์จากการอันมีลิขสิทธิ์ปราบกฏอยู่ในระบบการศึกษาอิเล็กทรอนิกส์ (E-Learning) ของมหาวิทยาลัย ไม่ว่าจะทั้งหมดหรือบางส่วน โดยไม่ได้รับอนุญาตจากมหาวิทยาลัย นอกเหนือจากการศึกษาส่วนบุคคล ทั้งนี้ การทำซ้ำ ดัดแปลง หรือเผยแพร่ต่อสาธารณะชนซึ่งงานอันมีลิขสิทธิ์ จะมีโทษปรับตั้งแต่ 20,000 บาท ถึง 200,000 บาท และหากเป็นการกระทำเพื่อการค้า จะมีโทษจำคุกตั้งแต่ 6 เดือน ถึง 10 ปี หรือปรับตั้งแต่ 100,000 บาท ถึง 800,000 บาท หรือทั้งจำทั้งปรับ

Unless allowed by Mahidol University, the User shall not copy, modify, or exploit in part or in whole of the copyrighted materials on the Platform, other than for your own individual study. The copy, modify or communication to public of the copyrighted materials shall be inflicted with a fine from 20,000 Baht up to 200,000 Baht. If the offence is committed with the commercial purpose, the offender shall be inflicted with imprisonment for a term from 6 months up to 10 years or a fine from 100,000 Baht up to 800,000 Baht or both imprisonment and fine.



## Agenda

- Why do we need Kubernetes?
- What's Kubernetes?
- Kubernetes architecture
- Kubernetes key features
- Kubernetes basic concepts
- Kubernetes security

3



## Docker on a Server



A diagram showing a single blue rounded rectangle divided into two horizontal sections. The top section contains the word 'Container' in white. The bottom section contains the word 'Node' in white.

4

The diagram illustrates the concept of Docker on multiple servers. At the top, a blue rounded rectangle contains the 'mu' logo and 'Mahidol University' text. Below it, the title 'Docker on Multiple Servers' is centered. In the middle, three light blue rounded rectangles are labeled 'Node'. Above them, a dark blue rounded rectangle is labeled 'Container' with three question marks ('???' below it) indicating uncertainty or a question about containerization. At the bottom right, a small number '5' is positioned next to a decorative wavy line pattern.

The diagram illustrates the concept of Kubernetes (K8S). At the top, the 'mu' logo and 'Mahidol University' text are shown. Below it, the title 'What is Kubernetes (K8S)?' is centered. To the right, there is a blue hexagonal icon containing a white steering wheel. A bulleted list provides information about Kubernetes:

- Kubernetes, in Greek, means the Helmsman, or pilot of the ship.
- It is pilot of a ship of containers.
- Kubernetes is a software written in Go for automating deployment, scaling, and management of containerized applications.
- Focus on manage applications, not machines.
- Open source, open API container orchestrator
- Supports multiple cloud and bare-metal environments.

At the bottom right, a small number '6' is positioned next to a decorative wavy line pattern.



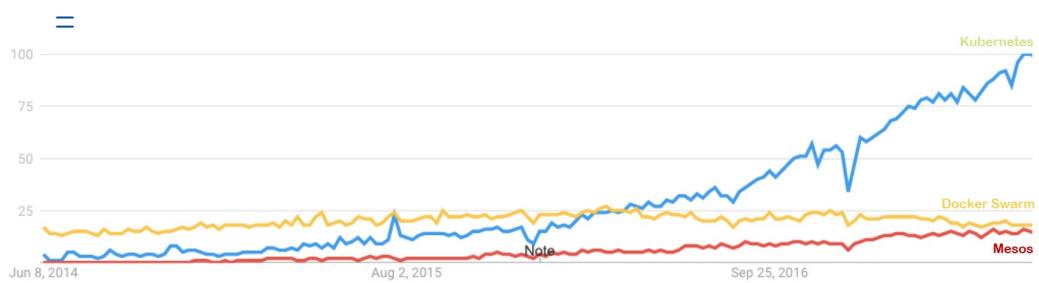
## What is Kubernetes (K8S)?

- Inspired and informed by 15 years of Google's experiences and internal systems.
- Kubernetes 1.0 was launched in OSCON 2015
- Kubernetes was based on Google Borg and Omega
- Other systems like Kubernetes are Apache Mesos, Docker Swarm, etc.

7



## Kubernetes is a Leader



8



- <https://www.cncf.io>
- Controls and manages projects under CNCF.
- Develops and supports related ecosystems.
- Organize events, seminar, training, certification
- In every 3 months approximately, there is a new release of Kubernetes.

9



## Kubernetes Key Features

- Automatic bin packing
- Self-healing
- Horizontal manual/auto-scaling
- Service discovery & load balancing
- Automated rollouts and rollbacks
- Secret and configuration management
- Storage orchestration
- Batch execution

10

**Kubernetes Platform**

**Certified Kubernetes - Distribution**

Source: <https://landscape.cncf.io>

11

**Kubernetes Platform**

**Certified Kubernetes - Installer**

**PaaS/Container Service**

Source: <https://landscape.cncf.io>

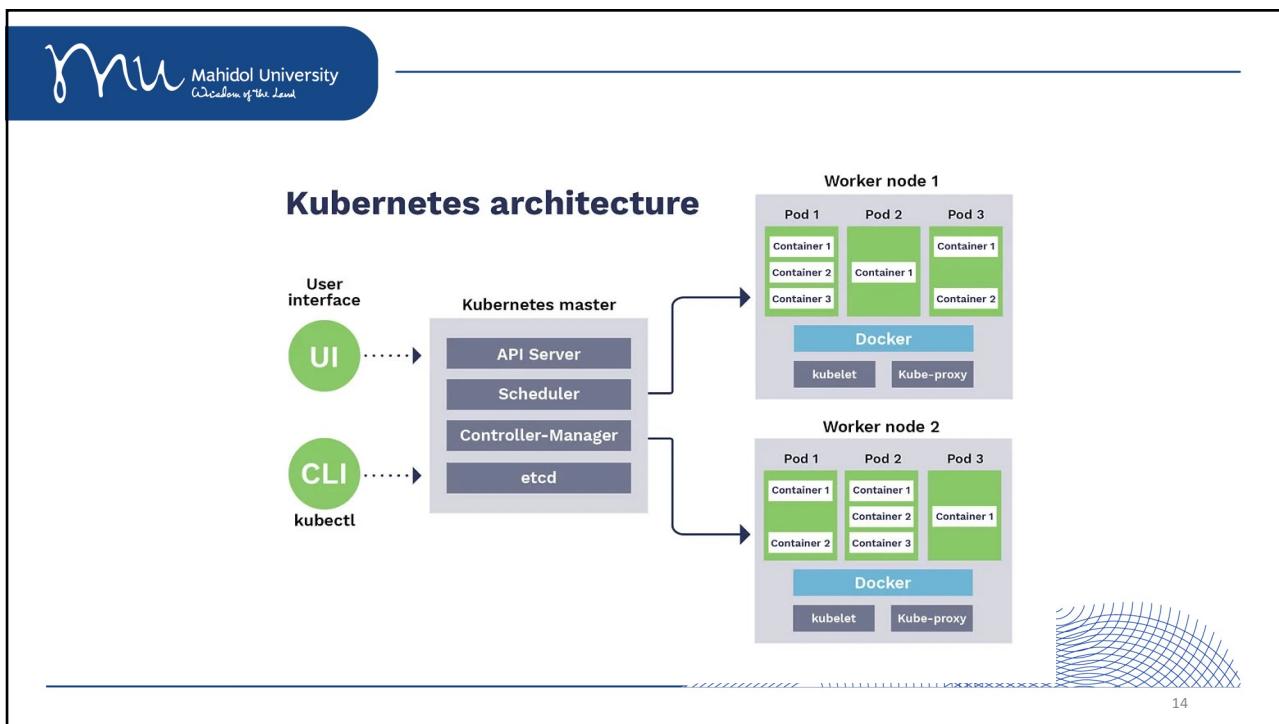
12

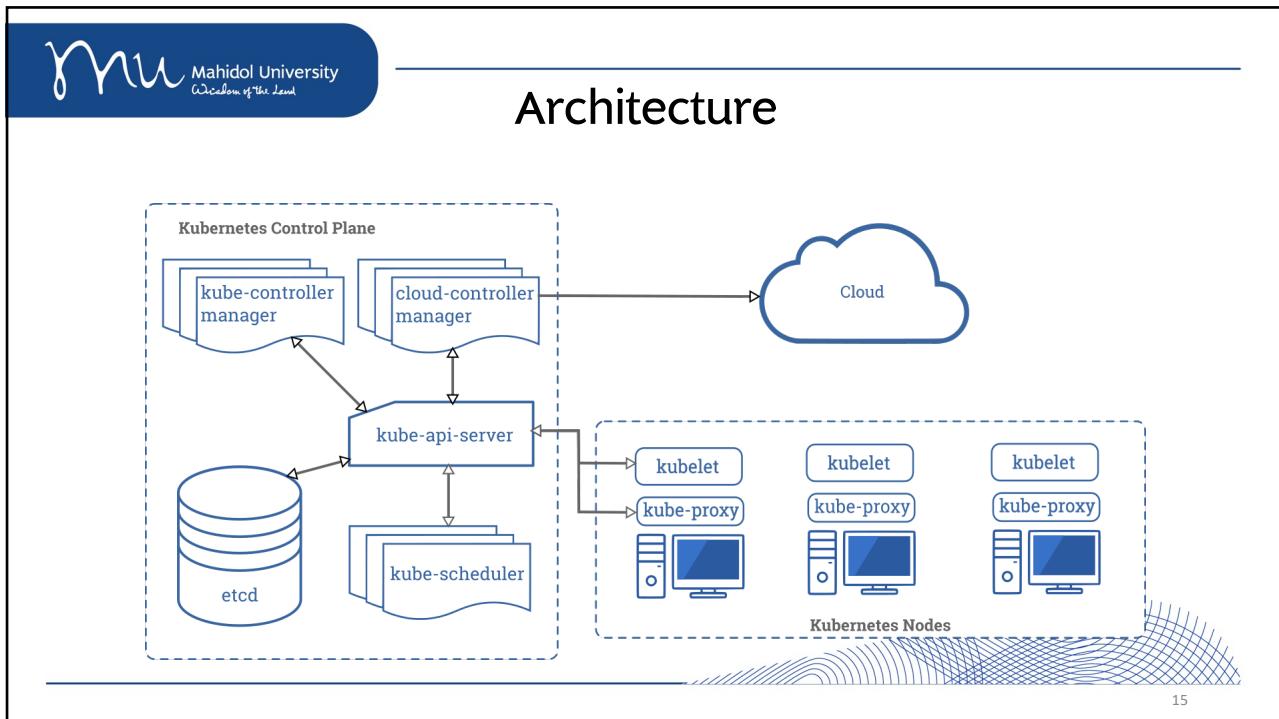
**Kubernetes Platform**

**Certified Kubernetes - Hosted**

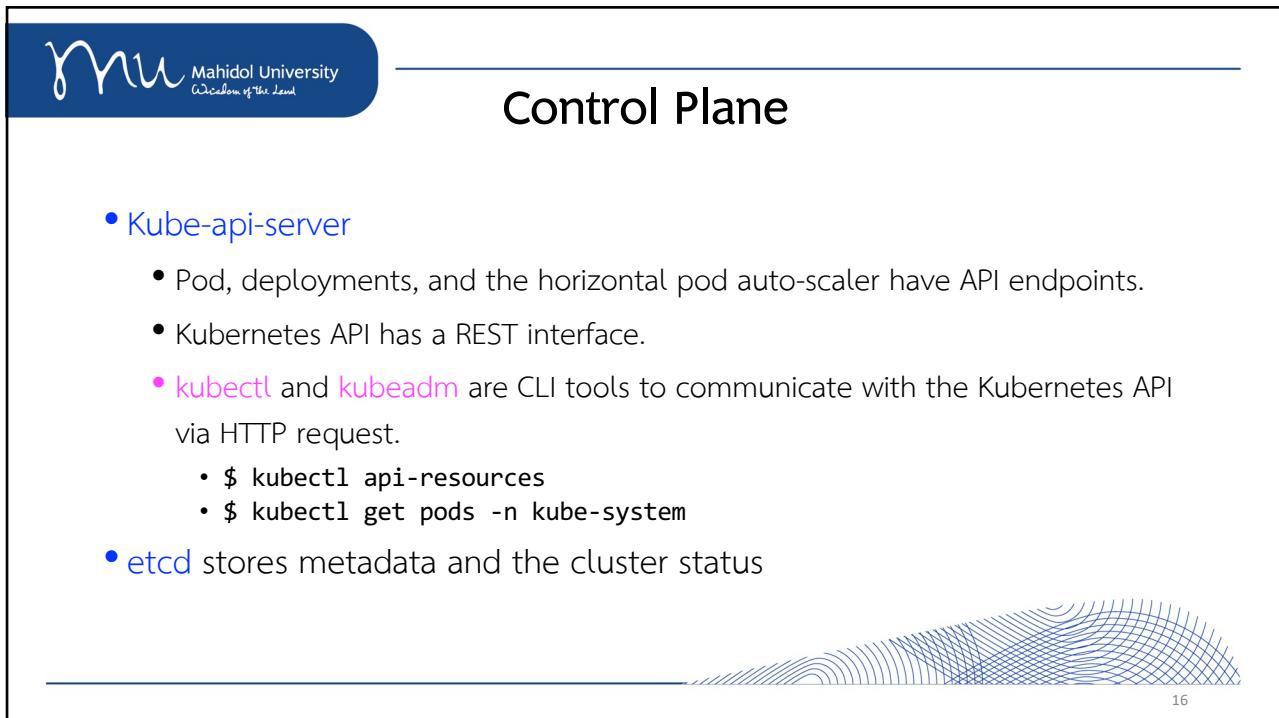
Source: <https://landscape.cncf.io>

13





15



16



## Control Plane

- **Kubelet**

- An agent that runs on every worker node
- Makes sure that containers in a pod are running and healthy.
- Communicates directly with the api-server in the control plane.

- **Container runtime**

- A kublet assigned to new pod starts a container using Container Runtime Interface (CRI).
- CRI enables that kublet to create containers with the engines: Containerd, CRI-O, Kata Containers, AWS Firecracker.

17



## Control Plane

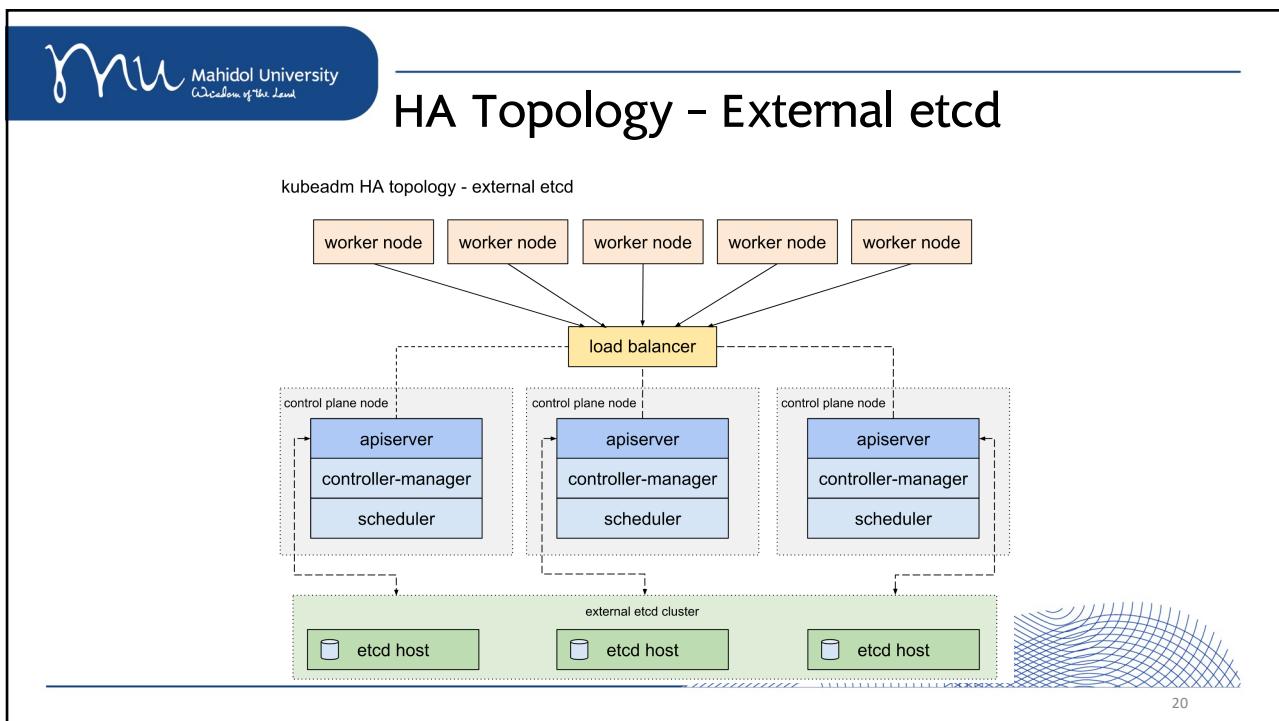
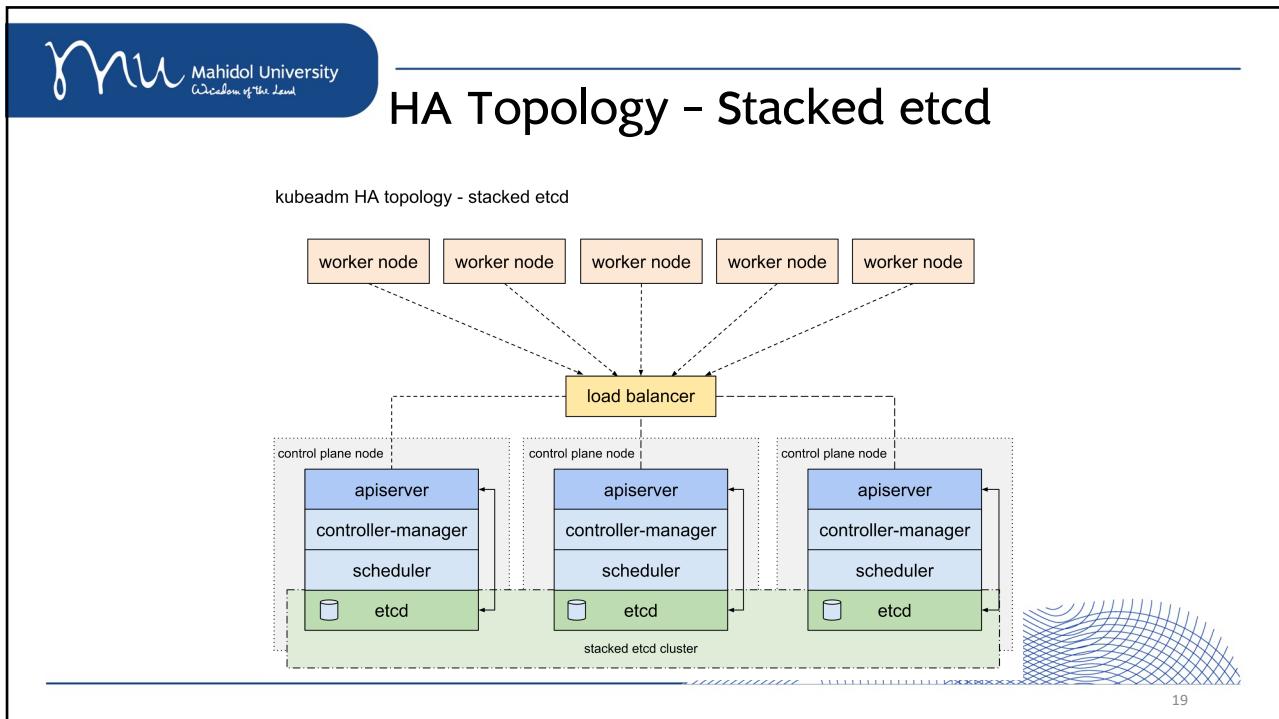
- **Kube-proxy**

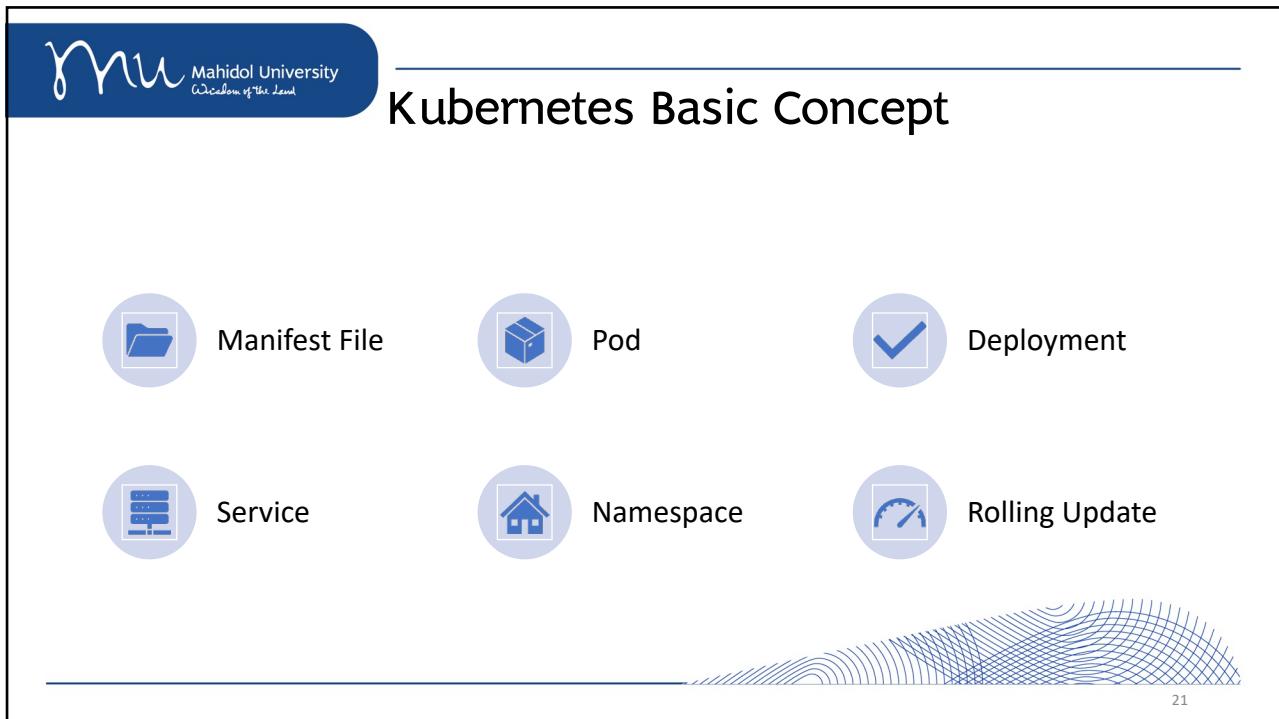
- Makes sure pods and services can communicate
- Each kube-proxy communicates directly with the kube-api-server.

- **Kube-scheduler**

- Pods scheduler in worker nodes based on resources availability, requests limit, etc.

18





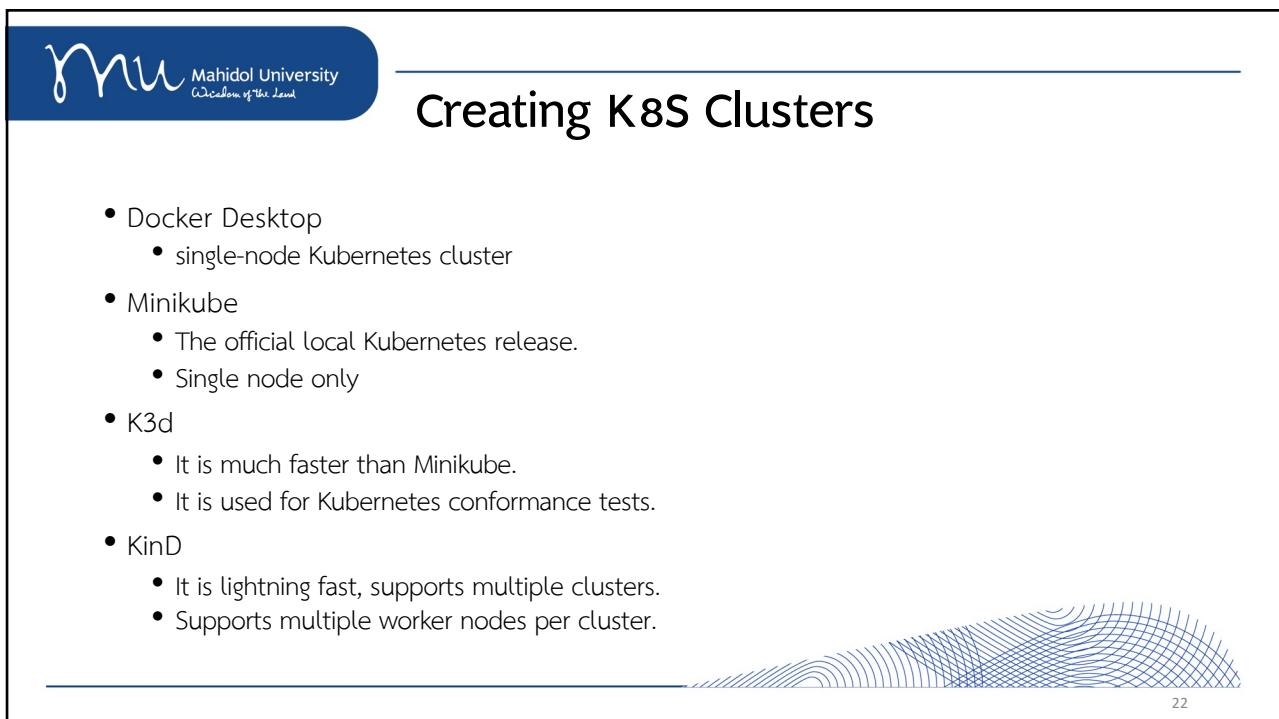
Mahidol University  
Wisdom of the Land

## Kubernetes Basic Concept

Manifest File      Pod      Deployment

Service      Namespace      Rolling Update

21



Mahidol University  
Wisdom of the Land

## Creating K8S Clusters

- Docker Desktop
  - single-node Kubernetes cluster
- Minikube
  - The official local Kubernetes release.
  - Single node only
- K3d
  - It is much faster than Minikube.
  - It is used for Kubernetes conformance tests.
- KinD
  - It is lightning fast, supports multiple clusters.
  - Supports multiple worker nodes per cluster.

22



## Lab1: Installation Options

- Docker Desktop: <https://www.docker.com/products/docker-desktop/>
- Install kubectl: <https://kubernetes.io/docs/tasks/tools/>
- Install minikube: <https://minikube.sigs.k8s.io/docs/start/>

23



## Kubernetes Sandbox



Play with Kubernetes

A simple, interactive and fun playground to learn Kubernetes

Login ▾

- <https://labs.play-with-k8s.com>
- It's free!
- Clean up every 4 hours

24



## YAML

- YAML Ain't Markup Language
- It's often used for configuration files.
- Like a suitcase, YAML lets you package and pick up your data and take it to another location and unload it without requiring anything extra from the developer.

25



## YAML

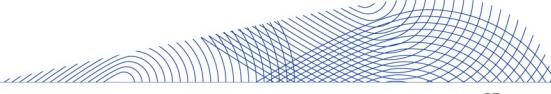
XML	JSON	YAML
<pre>&lt;Servers&gt;   &lt;Server&gt;     &lt;name&gt;Server1&lt;/name&gt;     &lt;owner&gt;John&lt;/owner&gt;     &lt;created&gt;05232023&lt;/created&gt;     &lt;status&gt;active&lt;/status&gt;   &lt;/Server&gt; &lt;/Servers&gt;</pre>	<pre>{   Servers: [     {       name: Server1,       owner: John,       created: 05232023,       status: active,     }   ] }</pre>	<pre>Servers:   - name: Server1     owner: John     created: 05232023     status: active</pre>

26

MU Mahidol University  
Wisdom of the Land

## YAML

Key Value Pair	Array/Lists	Dictionary/Map
Fruit: Apple Vegetable: Carrot Liquid: Water Meat: Chicken	Fruits: - Orange - Apple - Banana  Vegetables: - Carrot - Cauliflower - Tomato	Banana: Calories: 105 Fat: 0.4 g Carbs: 27 g  Grapes: Calories: 62 Fat: 0.3 g Carbs: 16 g



27

MU Mahidol University  
Wisdom of the Land

## YAML

Key value + dictionary + lists

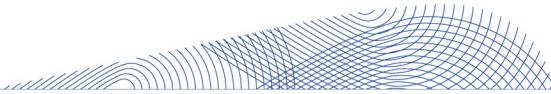
```

Fruits:
  - Banana:
    Calories: 105
    Fat: 0.4 g
    Carbs: 27 g
  - Grape:
    Calories: 105
    Fat: 0.3 g
    Carbs: 16 g

```

Notes:

- Ordering
- Dictionary = unordered
- List = ordered
- Comment = #



28

MU Mahidol University  
Wisdom of the Land

## Kubernetes Manifest File

```

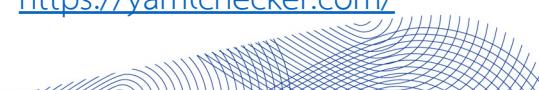
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
    type: front-end
spec:
  containers:
    - name: nginx-container
      image: nginx

```

**Infrastructure as code (IaC)**

Kind	Version
Pod	v1
Service	v1
ReplicaSet	apps/v1
Deployment	apps/v1

Syntax Checker:  
<https://yamlchecker.com/>

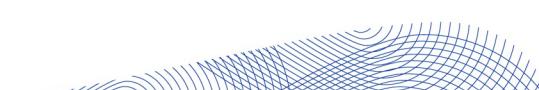


29

MU Mahidol University  
Wisdom of the Land

## Basic Commands

- kubectl get all
- kubectl get pods
- kubectl get services
- kubectl apply -f myfile.yml
- kubectl get services
- kubectl logs mypod
- kubectl delete pod mypod
- kubectl delete replicaset myapp\_replicaset
- kubectl delete service myservice
- kubectl delete pod --all
- kubectl delete service -all
- kubectl delete all --all



30

**MU** Mahidol University Wisdom of the Land

## Pod

- A pod is a smallest unit.
- Logical application
  - One or more containers and volumes
  - Shared namespaces
  - One IP per pod
- Resources are shared among containers in a pod.
- Each pod has unique ID (UID)

IP address  
volume  
containerized app

Pod 1      Pod 2      Pod 3      Pod 4

nginx  
application

GCE    iSCSI    NFS

Pod

31

**MU** Mahidol University Wisdom of the Land

## Pod

- We normally create a pod by Deployment, Job, or StatefulSet.
- Sidecar: a container provides a service, and other container is responsible for updating data in the first container
- We can do horizontal scaling by increasing a number of pods
- When a node fail, scheduler will create a pod for that.

32



## Lab 2: Creating a Pod

- Create a pod containing Nginx container using command

```
kubectl run nginx --image=nginx
kubectl get pods
kubectl get pods -o wide
```

- Delete the pod that we just created

```
kubectl delete pod podname
```

- Create a pod contained Nginx container using nginx.yaml file

```
kubectl create -f nginx.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
    type: front-end
spec:
  containers:
    - name: nginx-container
      image: nginx
```

33



## Pod

```
> kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
myapp-pod  1/1     Running   0          20s

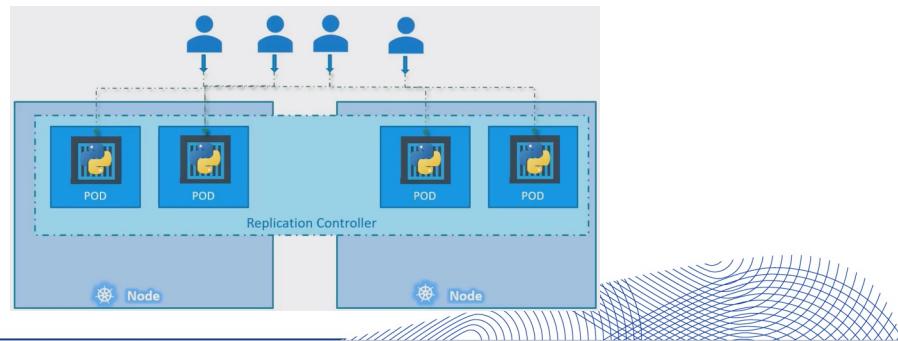
> kubectl describe pod myapp-pod
Name:           myapp-pod
Namespace:      default
Node:          minikube/192.168.99.100
Start Time:    Sat, 03 Mar 2018 14:26:14 +0800
Labels:        app=myapp
Annotations:   none
Status:        Running
IP:          172.17.0.24
Containers:
  nginx:
    Container ID: docker://830bb56c8c42a86b4bb70e9c148fae1bc38663e4918b6c2f5a783e7688b8c9d
    Image:          nginx
    Image ID:      docker-pullable://nginx@sha256:4771d0957c7c6a65299e110b3ee1c0a2592f5ea2618d23e4ffe7a4cab1ce5de
    Port:          <none>
    State:         Running
      Started:   Sat, 03 Mar 2018 14:26:21 +0800
    Ready:        True
    Restart Count: 0
    Environment:  none
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-x95w7 (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready       True
  PodScheduled  True
Events:
  Type  Reason  Age From      Message
  ----  -----  --  ----      -----
  Normal  Scheduled  34s  default-scheduler  Successfully assigned myapp-pod to minikube
  Normal  SuccessfulMountVolume  33s  kubelet, minikube  MountVolume.SetUp succeeded for volume "default-token-x95w7"
  Normal  Pulling   27s  kubelet, minikube  pull "nginx" for container "nginx"
  Normal  Pulled   27s  kubelet, minikube  successfully pulled image "nginx"
  Normal  Created   27s  kubelet, minikube  Created container
  Normal  Started   27s  kubelet, minikube  Started container
```

34



## Replica Controller & Replica Set

- Replication Controller (the old one) is replaced by Replica Set
- High availability, load balancing, and scalability



35



## Deployment

```
app: hello
replicas: 1
```



36

The diagram illustrates the deployment of three replicas of the 'hello' application across four nodes. In the first state, each node has one pod labeled 'hello'. In the second state, Node3 is shown in red, indicating it is the active node. In the third state, Node2 has two pods labeled 'hello', while Node1 and Node4 have one pod each. A blue wavy line at the bottom represents a cloud.

app: hello  
replicas: 3

Node1 Node2 Node3 Node4

Node1 Node2 Node3 Node4

Node1 Node2 Node3 Node4

37

The diagram illustrates the creation and management of a ReplicaSet named 'myapp-replicaset'. It shows the YAML definition of the ReplicaSet and the terminal output of running 'kubectl create -f replicaset-definition.yml' and 'kubectl get replicaset' followed by 'kubectl get pods'.

**replicaset-definition.yml**

```

replicaset-definition.yml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      type: front-end

```

**Terminal Output:**

```

> kubectl create -f replicaset-definition.yml
replicaset "myapp-replicaset" deleted

> kubectl get replicaset
NAME           DESIRED   CURRENT   READY   AGE
myapp-replicaset   3         3         3      19s

> kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
myapp-replicaset-9dd19  1/1    Running   0          45s
myapp-replicaset-9jtpx  1/1    Running   0          45s
myapp-replicaset-hq84m  1/1    Running   0          45s

```

38

MU Mahidol University  
Wisdom of the Land

## Scaling

```
replicaset-definition.yml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 6
  selector:
    matchLabels:
      type: front-end
```

```
> kubectl replace -f replicaset-definition.yml
```

```
> kubectl scale --replicas=6 -f replicaset-definition.yml
```

```
> kubectl scale --replicas=6 replicaset myapp-replicaset
```

TYPE                    NAME

39

MU Mahidol University  
Wisdom of the Land

## Lab 3: Replica Sets

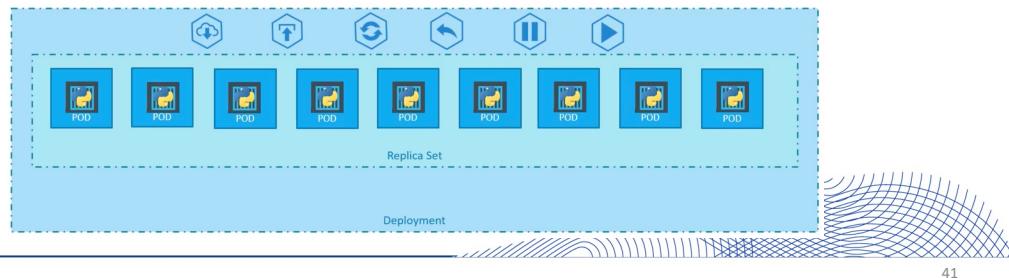
- Create a replica set as 3 using a yaml file (replica-definition.yaml) based Nginx container in Lab 2 as a template
- Try to delete one pod and check what happen
- Create a new pod with nginx.yaml file (in Lab 2) and with the same existing label of replicaset
- Change replicaset to 4 then check a number of pods you have by:  
`kubectl edit replicaset myapp-replicaset`
- Change replicaset to 2 then check a number of pods you have  
`kubectl scale replicaset myapp-replicaset --replicas=2`

40



## Deployment

- Deployment is used to define how many duplicated pods we want in our application, how upgrading/updating, how rolling back
- Services enables other applications from outside to connect to our application.



41



## Deployment Definition

```
deployment-definition.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp-deployment
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      type: front-end
```

```
> kubectl create -f deployment-definition.yml
deployment "myapp-deployment" created
```

```
> kubectl get deployments
NAME           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
myapp-deployment   3         3         3           3          21s
```

```
> kubectl get replicaset
NAME           DESIRED   CURRENT   READY   AGE
myapp-deployment-6795844b58   3         3         3      2m
```

```
> kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
myapp-deployment-6795844b58-5rbjl   1/1     Running   0          2m
myapp-deployment-6795844b58-h4w55   1/1     Running   0          2m
myapp-deployment-6795844b58-1fjhv   1/1     Running   0          2m
```

42



## Deployments, Replica Sets, and Pods

```
> kubectl get all
NAME           DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
deploy/myapp-deployment  3        3        3           3          9h
                           DESIRED  CURRENT  READY       AGE
rs/myapp-deployment-6795844b58  3        3        3           9h
                               READY    STATUS   RESTARTS  AGE
po/myapp-deployment-6795844b58-5rbjl  1/1     Running  0          9h
po/myapp-deployment-6795844b58-h4w55  1/1     Running  0          9h
po/myapp-deployment-6795844b58-lfjhv  1/1     Running  0          9h
```

43



## Lab 4: Creating a Deployment

- Create deployment.yaml file
- `kubectl create -f deployment.yaml`
- `kubectl get deployments`
- `kubectl get pods`
- `kubectl describe deployment myapp-deployment`
- `kubectl get all`

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: myapp-deployment
5    labels:
6      tier: frontend
7      app: nginx
8  spec:
9    selector:
10   matchLabels:
11     app: myapp
12   replicas: 3
13   template:
14     metadata:
15       name: nginx-2
16       labels:
17         app: myapp
18     spec:
19       containers:
20         - name: nginx
21           image: nginx
```

44

Mahidol University  
Wisdom of the Land

## Rollout and Update

Revision 1

nginx:1.7.0 nginx:1.7.0 nginx:1.7.0 nginx:1.7.0 nginx:1.7.0 nginx:1.7.0 nginx:1.7.0 nginx:1.7.0 nginx:1.7.0

Revision 2

nginx:1.7.1 nginx:1.7.1 nginx:1.7.1 nginx:1.7.1 nginx:1.7.1 nginx:1.7.1 nginx:1.7.1 nginx:1.7.1 nginx:1.7.1

45

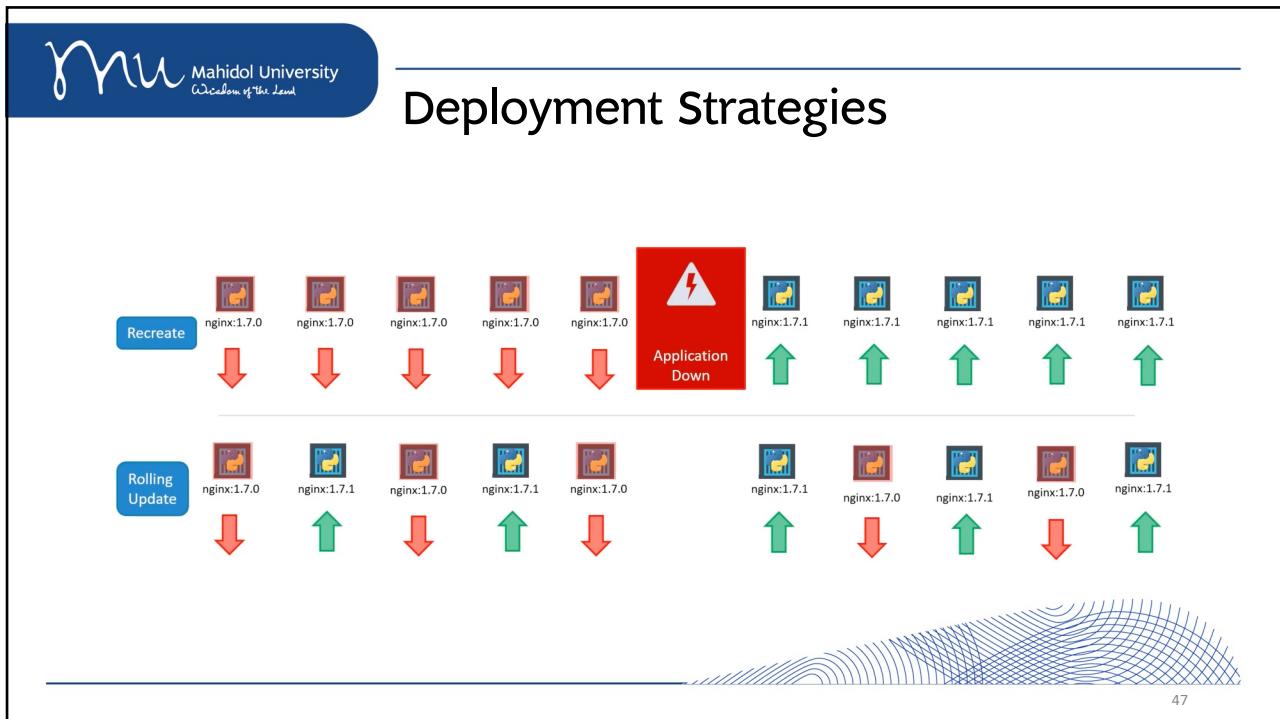
Mahidol University  
Wisdom of the Land

## Rollout Commands

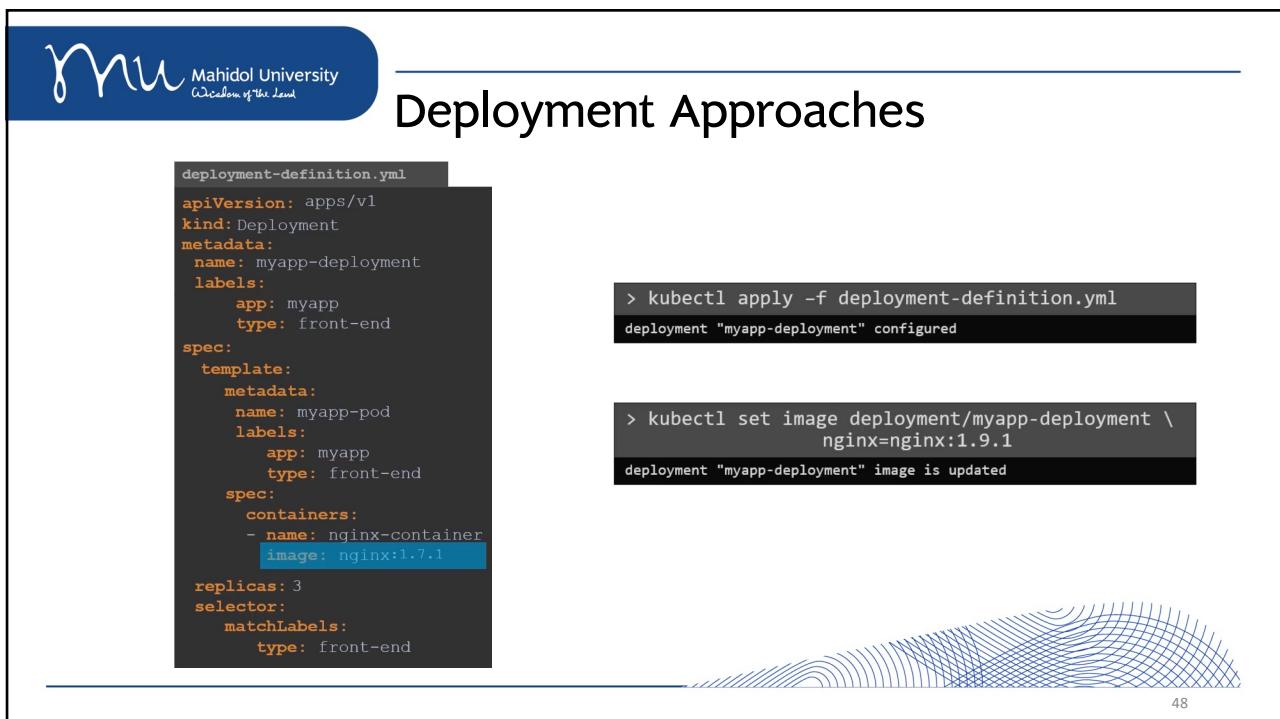
```
> kubectl rollout status deployment/myapp-deployment
Waiting for rollout to finish: 0 of 10 updated replicas are available...
Waiting for rollout to finish: 1 of 10 updated replicas are available...
Waiting for rollout to finish: 2 of 10 updated replicas are available...
Waiting for rollout to finish: 3 of 10 updated replicas are available...
Waiting for rollout to finish: 4 of 10 updated replicas are available...
Waiting for rollout to finish: 5 of 10 updated replicas are available...
Waiting for rollout to finish: 6 of 10 updated replicas are available...
Waiting for rollout to finish: 7 of 10 updated replicas are available...
Waiting for rollout to finish: 8 of 10 updated replicas are available...
Waiting for rollout to finish: 9 of 10 updated replicas are available...
deployment "myapp-deployment" successfully rolled out
```

```
> kubectl rollout history deployment/myapp-deployment
deployments "myapp-deployment"
REVISION  CHANGE-CAUSE
1        <none>
2        kubectl apply --filename=deployment-definition.yml --record=true
```

46



47



48

Mahidol University  
Wisdom of the Land

```
.\Kubernetes>kubectl describe deployment myapp-deployment
Name:           myapp-deployment
Namespace:      default
CreationTimestamp: Sat, 03 Mar 2018 17:01:55 +0800
Labels:         app=myapp
Annotations:    deployment.kubernetes.io/revision=2
                kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"Deployment","metadata":{"name":"myapp-deployment","namespace":"default"},"spec":{"replicas":5,"selector":{"matchLabels":{"app":"myapp","type":"front-end"}}, "template":{"metadata":{"labels":{"app":"myapp","type":"front-end"}}, "spec":{"containers":[{"name":"nginx", "image":"nginx:1.7.1", "ports":[{"port":80}], "resources":{}}, {"name:"nginx", "image:"nginx", "ports":[{"port":80}], "resources":{}}, {"name:"nginx", "image:"nginx", "ports":[{"port":80}], "resources":{}}, {"name:"nginx", "image:"nginx", "ports":[{"port":80}], "resources":{}}, {"name:"nginx", "image:"nginx", "ports":[{"port":80}], "resources":{}}]}}, "status":{"replicas":5, "readyReplicas":5, "availableReplicas":5, "unavailableReplicas":0, "conditions":[{"type": "Available", "status": "True", "reason": "MinimumReplicasAvailable", "lastTransitionTime": "2018-03-03T17:01:55Z"}, {"type": "Progressing", "status": "True", "reason": "NewReplicaSetAvailable", "lastTransitionTime": "2018-03-03T17:01:55Z"}, {"type": "Normal", "status": "True", "reason": "ReplicaSetUpdated", "lastTransitionTime": "2018-03-03T17:01:55Z"}, {"type": "Normal", "status": "True", "reason": "ReplicaSetCreated", "lastTransitionTime": "2018-03-03T17:01:55Z"}]}, "events": [{"type": "Normal", "reason": "ScalingReplicaSet", "age": "1m", "from": "deployment-controller", "message": "Scaled up replica set myapp-deployment-579584d58 to 5"}, {"type": "Normal", "reason": "ScalingReplicaSet", "age": "56s", "from": "deployment-controller", "message": "Scaled up replica set myapp-deployment-54c7d6ccc to 5"}]}
```

Recreate

```
.\Kubernetes>kubectl apply --filename=d:\Vmshared Files\Google Drive\Udemy\Kubernetes\Deployment\myapp-deployment-recreate.yaml
deployment.apps "myapp-deployment" recreated
```

RollingUpdate

49

Mahidol University  
Wisdom of the Land

## Upgrading

```
> kubectl get replicaset
NAME          DESIRED   CURRENT   READY   AGE
myapp-deployment-67c749c58c  0         0         0       22m
myapp-deployment-7d57dbdb8d  5         5         5       20m
```

50

Mahidol University  
Wisdom of the Land

## Rollback

Deployment

```
> kubectl rollout undo deployment/myapp-deployment
```

51

Mahidol University  
Wisdom of the Land

## Rollback

```
> kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-deployment-67c749c58c	0	0	0	22m
myapp-deployment-7d57dbdb8d	5	5	5	20m

```
> kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-deployment-67c749c58c	5	5	5	22m
myapp-deployment-7d57dbdb8d	0	0	0	20m

```
> kubectl rollout undo deployment/myapp-deployment
```

deployment "myapp-deployment" rolled back

52



## Command Summary

- `kubectl run nginx --image=nginx`
- `kubectl create -f deployment-definition.yml`
- `kubectl get deployments`
- `kubectl apply -f deployment-definition.yml`
- `kubectl set image deployment/myapp-deployment nginx=nginx:1.9.1`
- `kubectl rollout status deployment/myapp-deployment`
- `kubectl rollout history deployment/myall-deployment`
- `kubectl rollout undo deployment/myapp-deploy`

53



## Lab 5: Update & Rollback

- Create rollout deployment from deployment.yaml (in Lab 4) and check rollout status.
- Downgrade Nginx image to a lower version (e.g., 1.18)
- Downgrade Nginx image to a lower version (e.g., 1.18-perl) by using different approach from the previous
- Undo Nginx to the previous version (e.g., 1.18)
- Try to change Nginx image to any not existing name and see what happen

54

MU Mahidol University  
Wisdom of the Land

## Networking

The diagram illustrates a single Kubernetes node (Node) represented by a blue rounded rectangle. Inside the node, there is a cloud icon labeled '192.168.1.2' at the top. Below the cloud are three smaller clouds, each labeled '10.244.0.0'. Each of these clouds contains a yellow pod icon labeled 'POD' below it. A small circular icon with a gear and the word 'Node' is located at the bottom of the node.

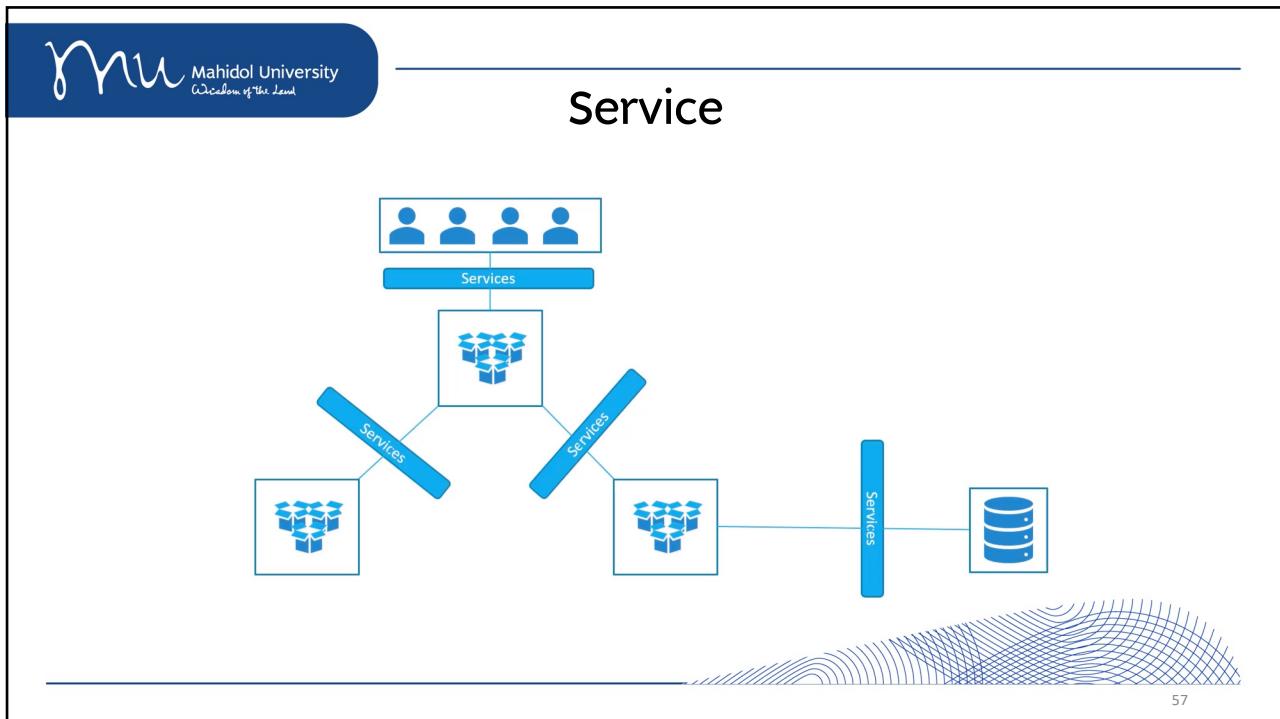
55

MU Mahidol University  
Wisdom of the Land

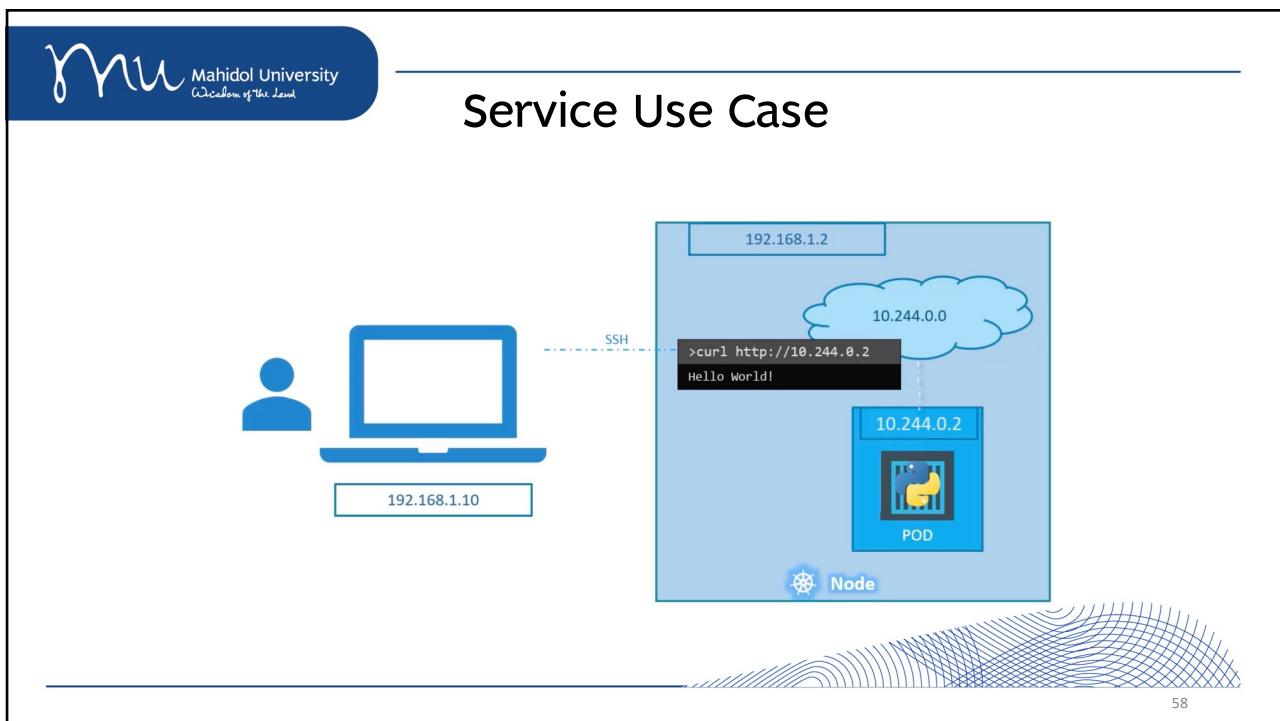
## Cluster Networking

The diagram shows two separate Kubernetes clusters, each enclosed in a blue rounded rectangle. The left cluster is labeled 'Kubernetes Cluster' at the bottom. It contains a cloud icon labeled '192.168.1.2' with a dashed line pointing to a cloud icon labeled '10.244.0.0'. Below this is a yellow pod icon labeled '10.244.0.2' with the word 'POD' underneath. A small circular icon with a gear and the word 'Node' is at the bottom. A red 'X' is drawn over the connection line between the two clusters, indicating that communication is blocked or disabled between them.

56



57



58

MU Mahidol University  
Wisdom of the Land

## Service Use Case

curl http://192.168.1.2:30008  
Hello World!

192.168.1.10

192.168.1.2

30008

Service

10.244.0.0

10.244.0.2

POD

Node

59

MU Mahidol University  
Wisdom of the Land

## Service Types

NodePort

ClusterIP

LoadBalancer

60

**NodePort**

```

service-definition.yml
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:

```

Range: 30000 - 32767

61

**NodePort**

```

service-definition.yml
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:

```

```

pod-definition.yml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
    type: front-end
spec:
  containers:
    - name: nginx-container
      image: nginx

```

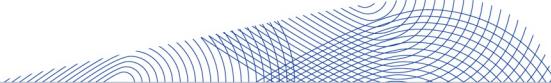
62

MU Mahidol University  
Wisdom of the Land

## NodePort

```
service-definition.yml
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: myapp
    type: front-end
```

```
pod-definition.yml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
spec:
  containers:
    - name: nginx-container
      image: nginx
```



63

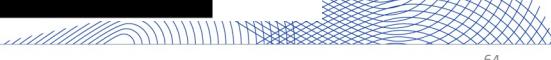
MU Mahidol University  
Wisdom of the Land

## NodePort

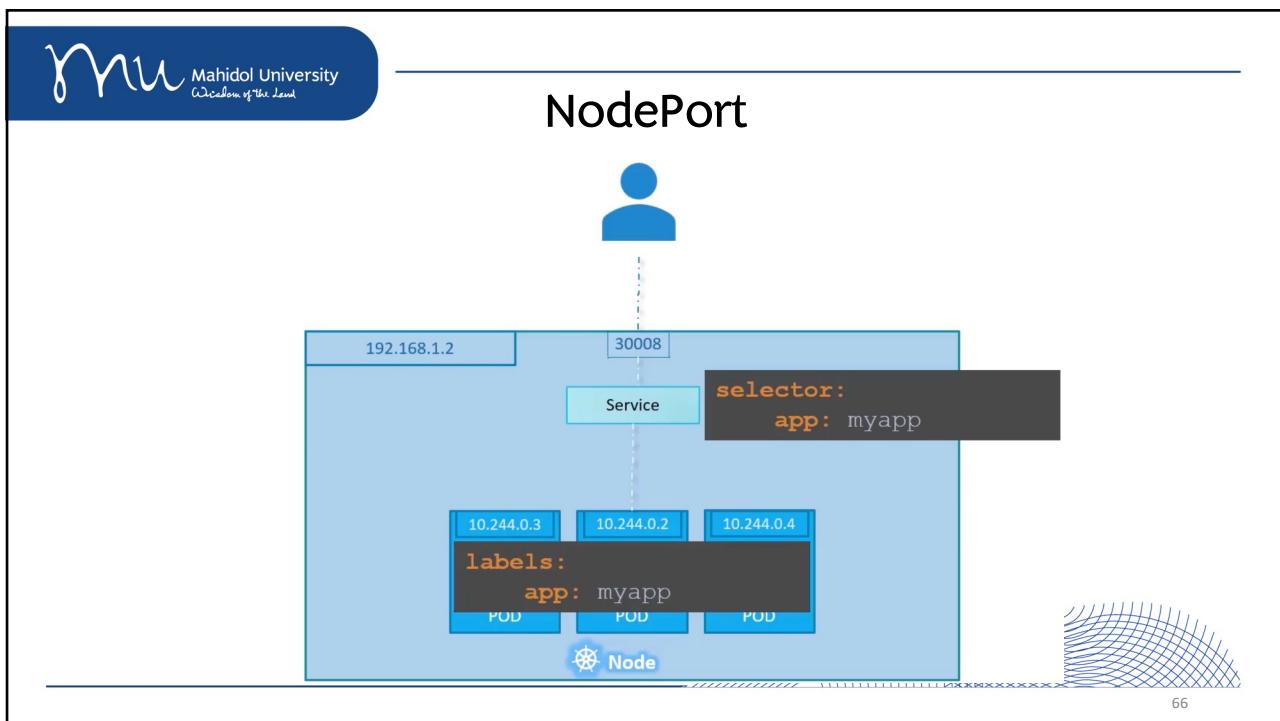
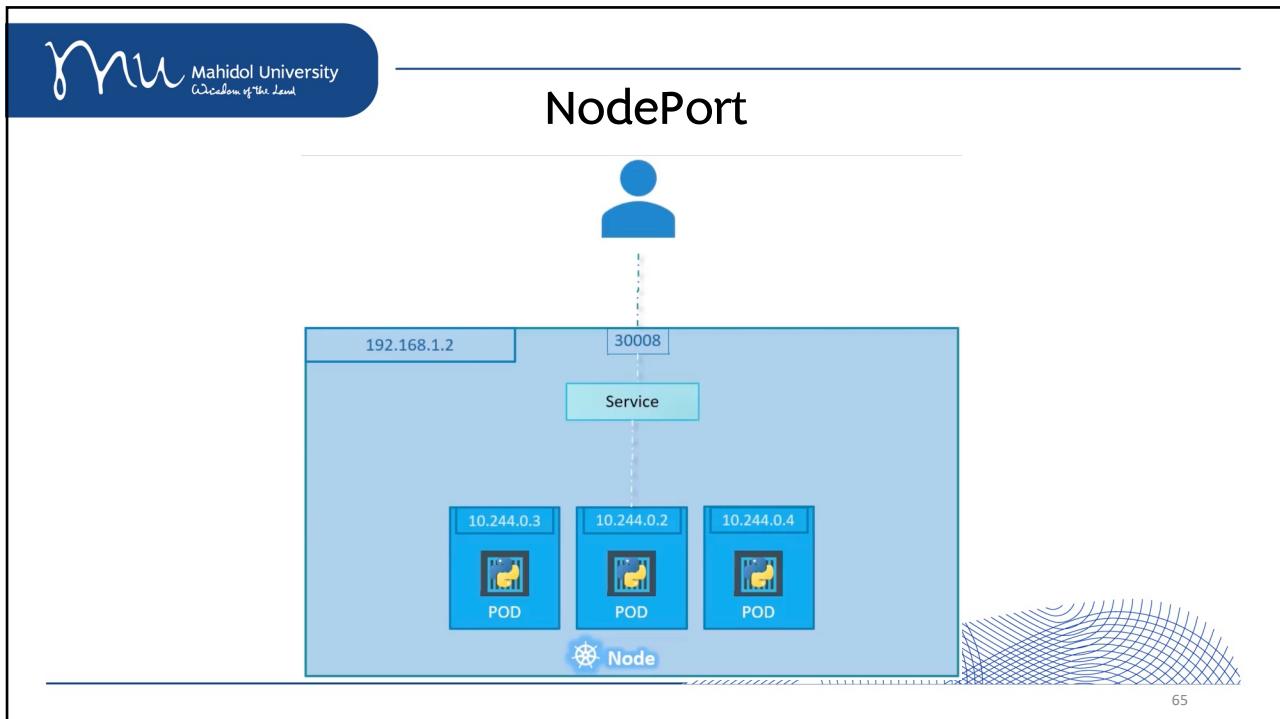
```
> kubectl create -f service-definition.yml
service "myapp-service" created
```

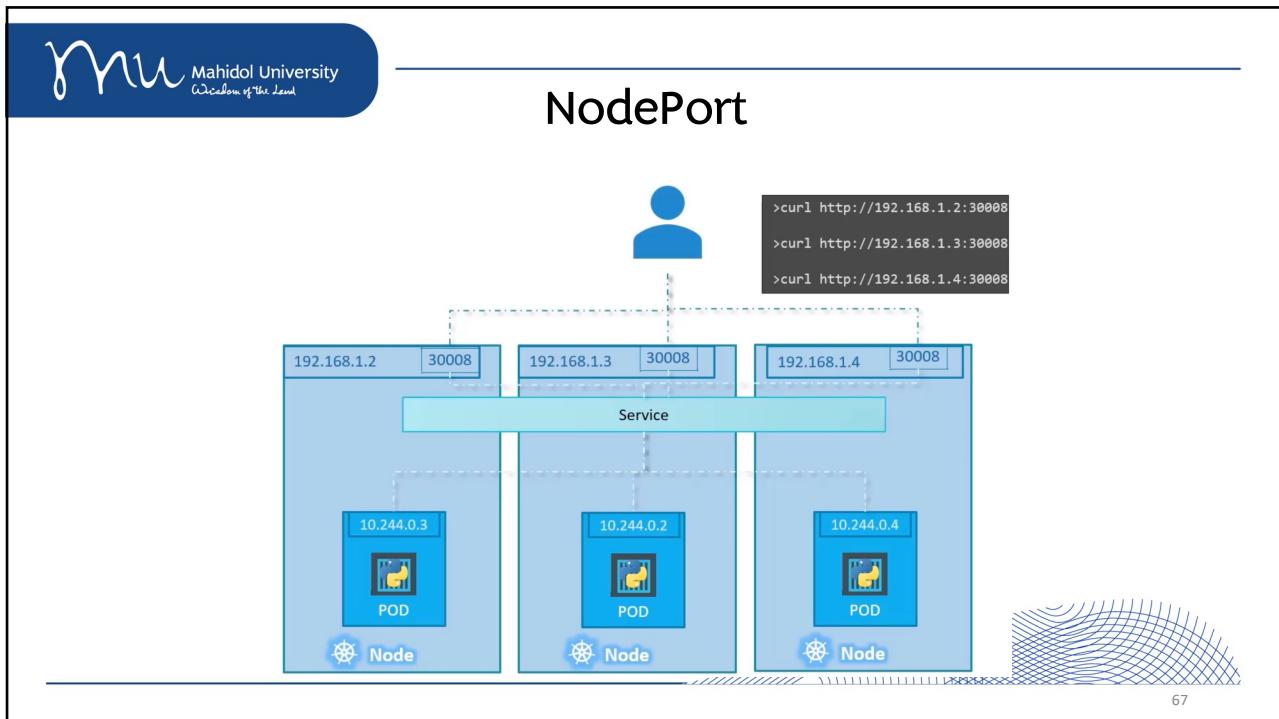
```
> kubectl get services
NAME         TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
kubernetes   ClusterIP  10.96.0.1   <none>       443/TCP       16d
myapp-service  NodePort   10.106.127.123  <none>       80:30008/TCP  5m
```

```
> curl http://192.168.1.2:30008
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
```

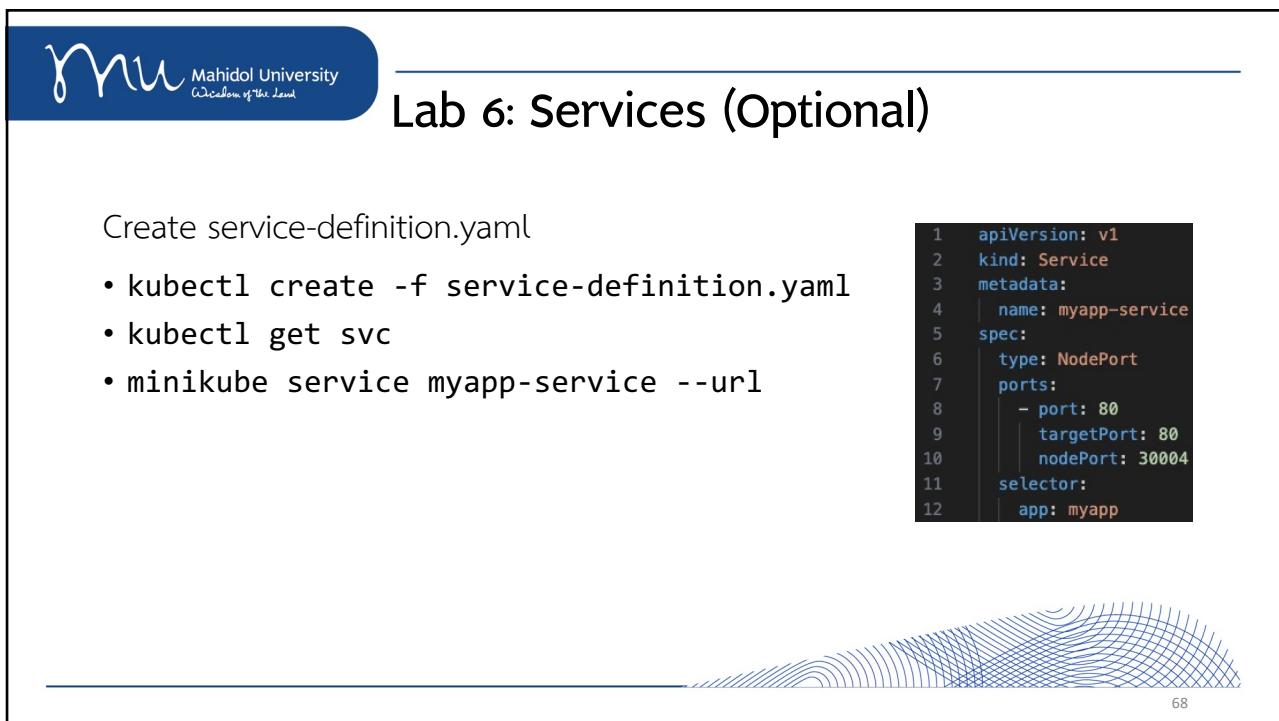


64

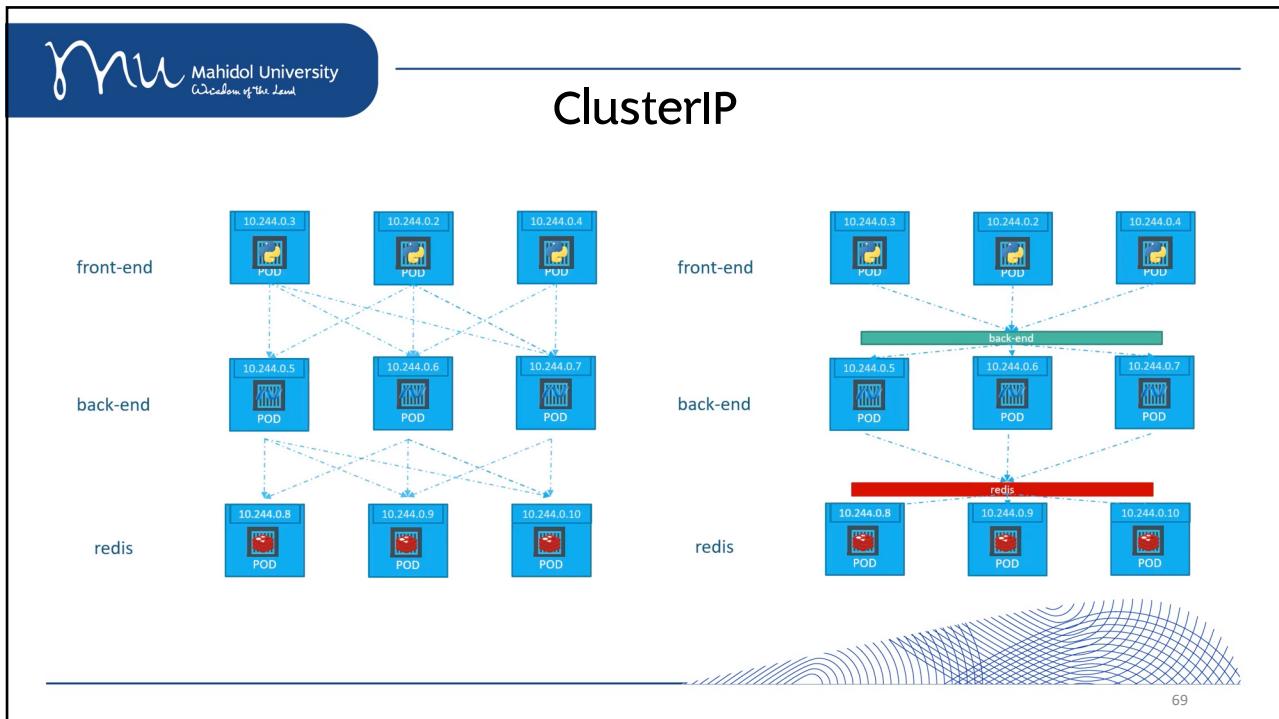




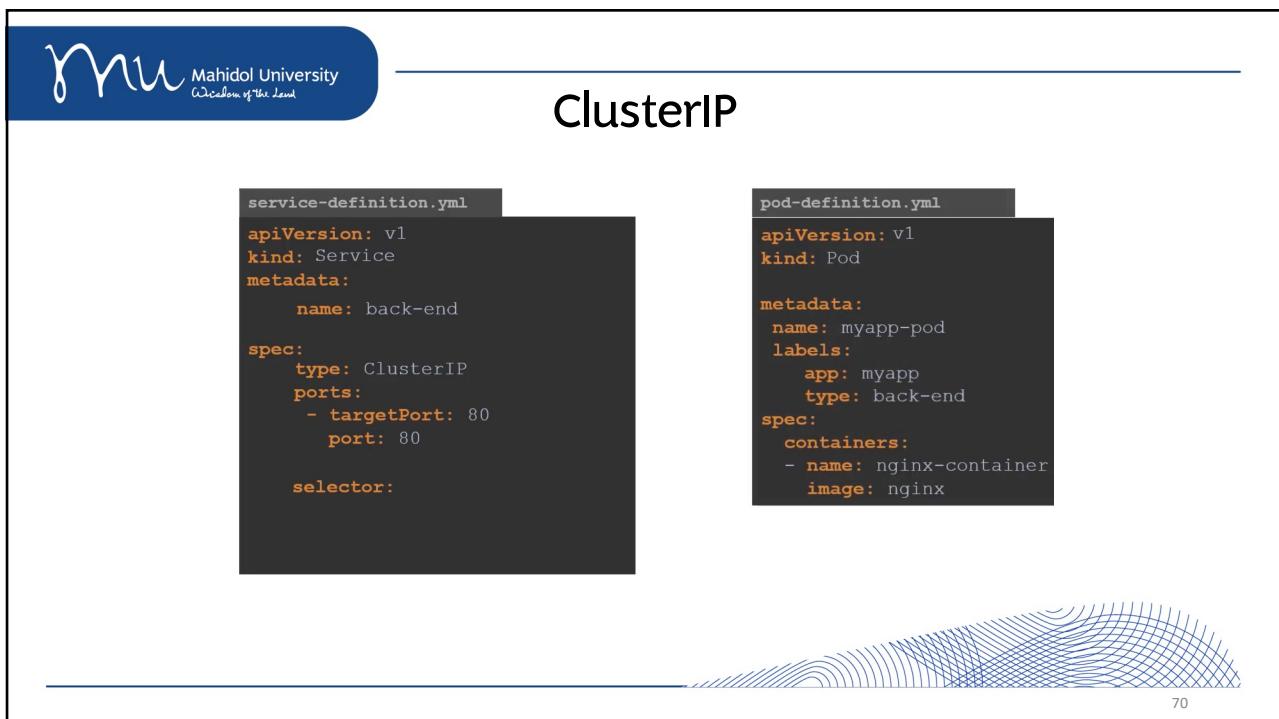
67



68



69



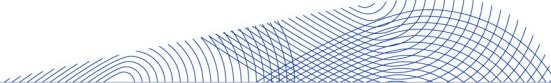
70

MU Mahidol University  
Wisdom of the Land

## ClusterIP

```
service-definition.yml
apiVersion: v1
kind: Service
metadata:
  name: back-end
spec:
  type: ClusterIP
  ports:
    - targetPort: 80
      port: 80
  selector:
    app: myapp
    type: back-end
```

```
pod-definition.yml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
spec:
  containers:
    - name: nginx-container
      image: nginx
```



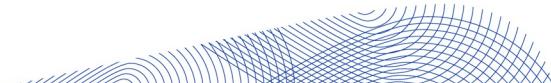
71

MU Mahidol University  
Wisdom of the Land

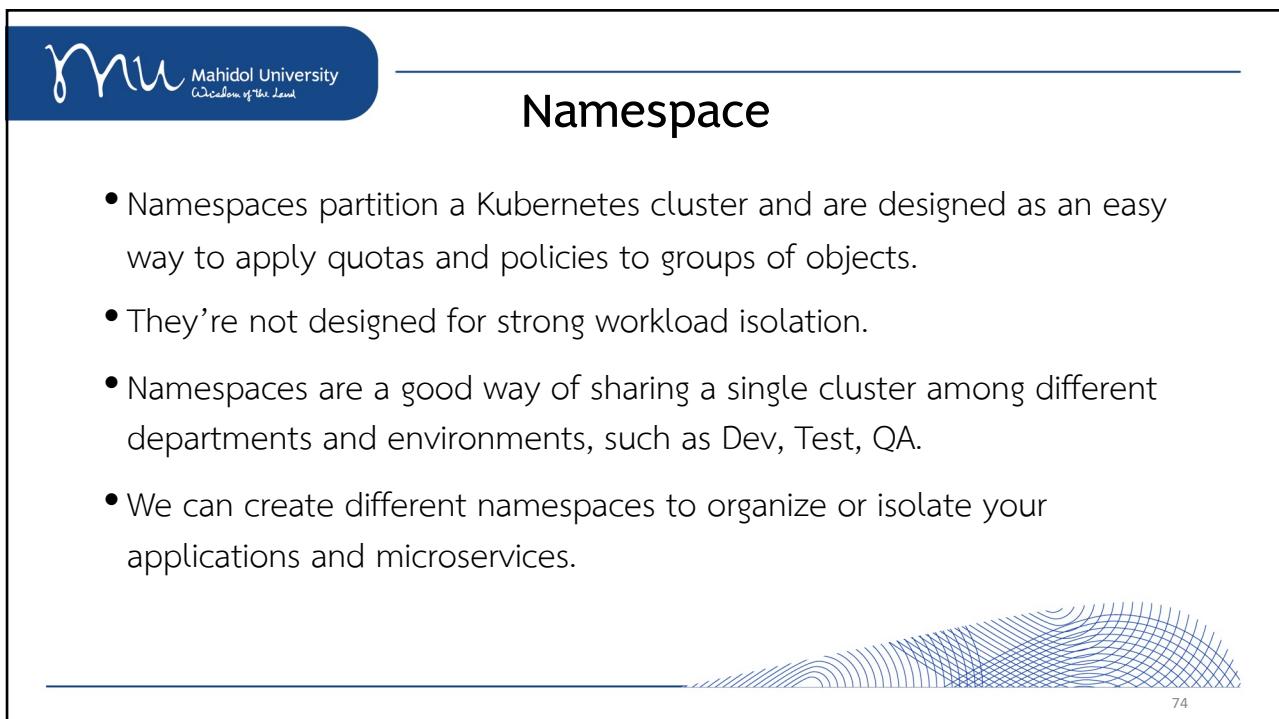
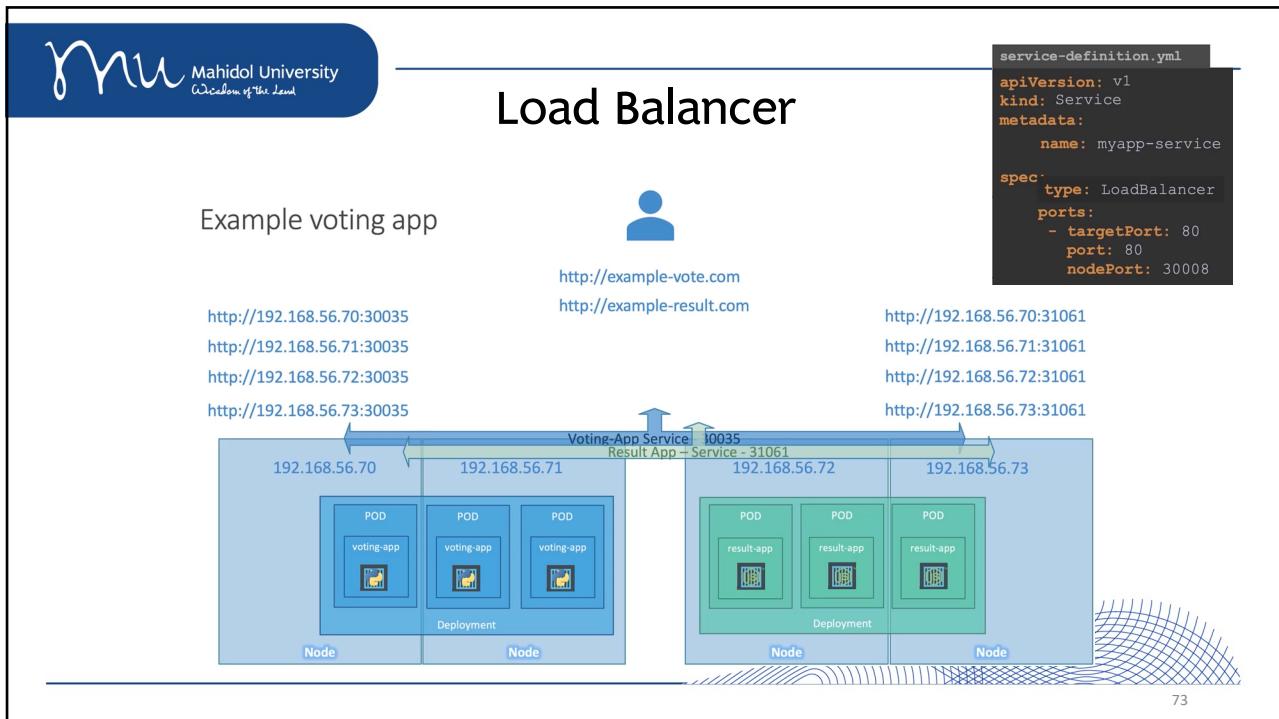
## ClusterIP

```
> kubectl create -f service-definition.yml
service "back-end" created
```

```
> kubectl get services
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP       16d
back-end     ClusterIP   10.106.127.123  <none>        80/TCP        2m
```



72



Namespace A

Namespace B

Service

Pod hello

Pod hello

Pod hello

Deployment

75

Namespace

- kubectl api-resources
- kubectl get namespaces
- kubectl apply -f *file.yml*
- kubectl delete -f *file.yml*
- kubectl create ns *mynamespace*
- kubectl delete ns *mynamespace*

```
1  ---
2  apiVersion: v1
3  kind: Namespace
4  metadata:
5  | name: development
6  ---
7  apiVersion: v1
8  kind: Namespace
9  metadata:
10 | name: production
```

76



## Deploying Namespace

```
apiVersion: v1
kind: ServiceAccount
metadata:
  namespace: shield      <===== Namespace
  name: default
---
apiVersion: v1
kind: Service
metadata:
  namespace: shield      <===== Namespace
  name: the-bus
spec:
  ports:
    - nodePort: 31112
      port: 8080
      targetPort: 8080
  selector:
    env: marvel
---
apiVersion: v1
kind: Pod
metadata:
  namespace: shield      <===== Namespace
  name: triskelion
<Snip>
```

```
$ kubectl apply -f shield-app.yml
serviceaccount/default configured
service/the-bus configured
pod/triskelion created

$ kubectl get pods -n shield
NAME        READY   STATUS    RESTARTS   AGE
triskelion  1/1     Running   0          48s

$ kubectl get svc -n shield
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
the-bus   NodePort  10.43.30.174   <none>       8080:31112/TCP  52s
```

77



## Logs

```
$ kubectl get pod -n namespace
$ kubectl logs pod_name -n namespace
```

78

Mahidol University  
Wisdom of the Land

## Resource Management

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
    - name: app
      image: images.my-company.example/app:v4
      resources:
        requests:
          memory: "64Mi"
          cpu: "250m"
        limits:
          memory: "128Mi"
          cpu: "500m"
    - name: log-aggregator
      image: images.my-company.example/log-aggregator:v6
      resources:
        requests:
          memory: "64Mi"
          cpu: "250m"
        limits:
          memory: "128Mi"
          cpu: "500m"
```

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

79

Mahidol University  
Wisdom of the Land

## Multi-node Cluster

The diagram illustrates a multi-node Kubernetes cluster. On the left, a blue box represents the 'Master' node, containing icons for kube-apiserver, etcd, controller, scheduler, and a gear icon labeled 'Master'. On the right, a larger blue box represents a 'Worker Node', containing icons for kubelet, Container Runtime, and a gear icon labeled 'Worker Node'. A dashed line connects the two nodes, representing the communication between them.

80

Multi-node Cluster Using kubeadm

1. Initialize  
2. kubeadm  
3. docker  
4. Master  
5. Join Node  
6. POD Network  
7. kubeadm  
8. docker  
9. Worker Node 1  
10. Join Node  
11. kubeadm  
12. docker  
13. Worker Node 2

81

## Kubernetes Security

- Any internet server is susceptible to attack.
- Threats are always changing.
- There are best practices you can apply right away.
- Steal data, cryptocurrency mining, DDoS attacks
- <https://docs.snyk.io/scan-cloud-deployment/snyk-infrastructure-as-code/snyk-cli-for-infrastructure-as-code/test-your-kubernetes-files-with-our-cli-tool>
- \$ snyk iac test file.yml
- A hardening guide:  
[https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR\\_KUBERNETES\\_HARDENING\\_GUIDANCE\\_1.2\\_20220829.PDF](https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR_KUBERNETES_HARDENING_GUIDANCE_1.2_20220829.PDF)

```

18 spec:
19   containers:
20     - name: pod-info-container
21       image: kimschles/pod-info-app:latest
22       resources:
23         requests:
24           memory: "64Mi"
25             cpu: "250m"
26         limits:
27           memory: "128Mi"
28             cpu: "500m"
29       securityContext:
30         allowPrivilegeEscalation: false
31         runAsNonRoot: true
32         capabilities:
33           drop:
34             - ALL
35             readOnlyRootFilesystem: true
36         ports:
37           - containerPort: 3000

```

82



## References

- KodeKloud.com
- Opsta
- Nigel Poulton, "The Kubernetes Book", 2023.