

LAB Guide

Module 2: Kubernetes

สำหรับ module 2 จะเป็นการแนะนำเกี่ยวกับ basic command ของ Kubernetes โดยใช้ kubectl ทดลอง deploy application, expose service, ตรวจสอบ log events และ enable Kubernetes dashboard

Step 1

เปลี่ยน directory บน master ไปที่ lab-module2 โดยใช้ command `cd ~/kic-training/lab-module2`

ตรวจสอบ status ของ cluster โดยใช้ command `kubectl get node` ตรง column status จะต้องขึ้นเป็น ready ทั้ง master และ worker01

```
vagrant@master:~/kic-training/lab-module1$ kubectl get node
NAME        STATUS    ROLES    AGE   VERSION
master      Ready    control-plane   3h16m   v1.28.2
worker01    Ready    <none>        3h11m   v1.28.2
vagrant@master:~/kic-training/lab-module1$
```

ตรวจสอบ status ของ pod ใน namespace kube-system ด้วย command

`kubectl get pod -n kube-system` ทุก pod จะต้องมี status เป็น running

```
vagrant@master:~/kic-training/lab-module1$ kubectl get pod -n kube-system
NAME                                READY    STATUS    RESTARTS   AGE
coredns-5dd5756b68-2gw9j           1/1     Running   0           3h17m
coredns-5dd5756b68-prp2l           1/1     Running   0           3h17m
etcd-master                         1/1     Running   0           3h18m
kube-apiserver-master               1/1     Running   0           3h18m
kube-controller-manager-master      1/1     Running   0           3h18m
kube-proxy-gwjmv                    1/1     Running   0           3h13m
kube-proxy-zc8qv                    1/1     Running   0           3h17m
kube-scheduler-master               1/1     Running   0           3h18m
weave-net-hkgzg                     2/2     Running   0           3h13m
weave-net-s9d9d                     2/2     Running   1 (3h14m ago) 3h14m
vagrant@master:~/kic-training/lab-module1$
```

* -n หมายถึง namespace หากไม่ระบุ namespace จะเป็นการ show object ใน namespace default

สามารถ show namespace ทั้งหมดใน cluster ด้วย command `kubectl get namespace`

```
vagrant@master:~/kic-training/lab-module1$ kubectl get namespace
NAME                STATUS    AGE
default             Active    3h21m
kube-node-lease     Active    3h21m
kube-public          Active    3h21m
kube-system          Active    3h21m
vagrant@master:~/kic-training/lab-module1$
```

Step 2

Deploy applications โดยใช้ manifest file ใน lab นี้จะมีทั้งหมด 2 app คือ coffee และ tea

ตัวอย่างไฟล์ coffee-deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: coffee
spec:
  replicas: 2
  selector:
    matchLabels:
      app: coffee
  template:
    metadata:
      labels:
        app: coffee
    spec:
      containers:
        - name: coffee
          image: nginxdemos/nginx-hello:plain-text
          ports:
            - containerPort: 8080
```

Version ของ api บน api server

ประเภทของ object ที่ต้องการสร้างจะเป็น deployment (pod + replicas)

กำหนดชื่อ deployment เท่ากับ coffee

แบ่งเป็น 2 ส่วน replicaSet กับ Pod

กำหนดจำนวนของ app ที่จะ deploy จำนวน 2 pod โดย match กับ pod ที่มี label เท่ากับ **app: coffee**

กำหนด spec ของ pod ให้ label เท่ากับ **app: coffee**

กำหนด container name เป็น coffee โดยใช้ images ของ nginx และ listen ที่ port 8080

ตัวอย่างไฟล์ coffee-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: coffee-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 8080
      nodePort: 30001
      protocol: TCP
      name: http
  selector:
    app: coffee
```

Version ของ api บน api server

ประเภทของ object ที่ต้องการสร้างจะเป็น service

กำหนดชื่อ service เท่ากับ coffee-svc

กำหนด service type เป็น NodePort เพื่อให้ Client ที่อยู่นอก Cluster สามารถ access app ได้

Port ที่ใช้ใน Cluster เท่ากับ 80

Port ของ Container ที่ listen ใน Pod ที่ได้จากการสร้าง Deployment

Port ที่ map กับ host เพื่อให้ client สามารถ access app ได้ โดยจะเริ่มต้นที่ port 30000

จะ map service เข้ากับ Pod ที่มี label เป็น **app: coffee**

Deploy application โดยใช้ command

```
kubectl apply -f coffee-deploy.yaml
```

```
kubectl apply -f coffee-svc.yaml
```

```
kubectl apply -f tea-deploy.yaml
```

```
kubectl apply -f tea-svc.yaml
```

ตรวจสอบ deployment และ service ที่ apply ไปโดยใช้ command `kubectl get deploy` เพื่อตรวจสอบ deployment จะมี 2 deployment คือ coffee และ tea สังเกต column READY จะมีตัวเลข 2/2 และ 3/3

```
vagrant@master:~/kic-training/lab-module2$ kubectl get deploy
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
coffee	2/2	2	2	104s
tea	3/3	3	3	103s

```
vagrant@master:~/kic-training/lab-module2$
```

ตัวเลขด้านหน้าหมายถึงจำนวน Pod ที่ run อยู่ทั้งหมด ส่วนตัวเลขด้านหลังจะเป็นจำนวน replicas ที่ set ไว้ใน manifest file ตัวเลข 2 ตัวนี้จะต้องเท่ากัน

ใช้ command `kubectl get svc` เพื่อแสดง service ทั้งหมดใน namespace จะมี coffee-svc และ tea-svc เพิ่มขึ้นมา จะมี Type = NodePort และ Port ที่ใช้ access app จะเป็น 30001 และ 30002

```
vagrant@master:~/kic-training/lab-module2$ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
coffee-svc	NodePort	10.108.26.227	<none>	80:30001/TCP	5m32s
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	5h49m
tea-svc	NodePort	10.104.70.226	<none>	80:30002/TCP	5m31s

ใช้ command `kubectl describe svc coffee-svc` เพื่อดูรายละเอียดของ service ส่วนของ Endpoints จะต้องเท่ากับจำนวน Pod ที่กำหนดใน replicas จากตัวอย่าง app coffee จะมี replicas = 2

```
vagrant@master:~/kic-training/lab-module2$ kubectl describe svc coffee-svc
Name: coffee-svc
Namespace: default
Labels: <none>
Annotations: <none>
Selector: app=coffee
Type: NodePort
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.108.26.227
IPs: 10.108.26.227
Port: http 80/TCP
TargetPort: 8080/TCP
NodePort: http 30001/TCP
Endpoints: 10.244.192.1:8080,10.244.192.5:8080
Session Affinity: None
External Traffic Policy: Cluster
Events: <none>
```

ใช้ command `kubectl get pod` เพื่อแสดง Pod ที่มีการสร้างจาก Deployment

```
vagrant@master:~/kic-training/lab-module2$ kubectl get pod
NAME READY STATUS RESTARTS AGE
coffee-6b8b6d6486-5tkwq 1/1 Running 0 10m
coffee-6b8b6d6486-r8scc 1/1 Running 0 10m
tea-9d8868bb4-9lh2f 1/1 Running 0 10m
tea-9d8868bb4-kfvdf 1/1 Running 0 10m
tea-9d8868bb4-qk1xn 1/1 Running 0 10m
```

สามารถใช้ option `-o wide` เพื่อดูข้อมูลเพิ่มเติมได้ โดยใช้ command `kubectl get pod -o wide` จะแสดงข้อมูล IP และ Worker Node ที่ Pod นั้นๆ run อยู่

```
vagrant@master:~/kic-training/lab-module2$ kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
coffee-6b8b6d6486-5tkwq 1/1 Running 0 12m 10.244.192.1 worker01 <none> <none>
coffee-6b8b6d6486-r8scc 1/1 Running 0 12m 10.244.192.5 worker01 <none> <none>
tea-9d8868bb4-9lh2f 1/1 Running 0 12m 10.244.192.4 worker01 <none> <none>
tea-9d8868bb4-kfvdf 1/1 Running 0 12m 10.244.192.2 worker01 <none> <none>
tea-9d8868bb4-qk1xn 1/1 Running 0 12m 10.244.192.3 worker01 <none> <none>
```

ทดสอบ access app coffee โดยใช้ curl http://192.168.100.2:30001 จะแสดง content ของ app coffee โดยดูจากชื่อของ server จะขึ้นต้นด้วย coffee

```
vagrant@master:~/kic-training/lab-module2$ curl http://192.168.100.2:30001
Server address: 10.244.192.5:8080
Server name: coffee-6b8b6d6486-r8scc
Date: 11/Dec/2023:11:07:20 +0000
URI: /
Request ID: 74716c6a9c1ecb7d4af3f5955e2efc5e
vagrant@master:~/kic-training/lab-module2$ curl http://192.168.100.2:30001
Server address: 10.244.192.1:8080
Server name: coffee-6b8b6d6486-5tkwq
Date: 11/Dec/2023:11:07:37 +0000
URI: /
Request ID: 0e6a23e94d96a68140d7a8b053330eac
```

ทดสอบ access app tea โดยใช้ curl http://192.168.100.2:30002 จะแสดง content ของ app tea โดยดูจากชื่อของ server จะขึ้นต้นด้วย tea

```
vagrant@master:~/kic-training/lab-module2$ curl http://192.168.100.2:30002
Server address: 10.244.192.3:8080
Server name: tea-9d8868bb4-qklxn
Date: 11/Dec/2023:11:09:42 +0000
URI: /
Request ID: 02a3e3dce9c9d3531c4c9deb95f526de
vagrant@master:~/kic-training/lab-module2$ curl http://192.168.100.2:30002
Server address: 10.244.192.3:8080
Server name: tea-9d8868bb4-qklxn
Date: 11/Dec/2023:11:09:43 +0000
URI: /
Request ID: 6b1ec91b1eaae6d7709c713d050011da
```


Step 3

ทดสอบ Scale application ด้วย command `kubectl scale deploy coffee --replicas=4` จากนั้นใช้ `kubectl get pod` เพื่อแสดง pod อีกครั้ง จะมี pod ที่ถูกสร้างมาใหม่อีก 2 pod เพื่อให้เท่ากับจำนวน replicas ที่กำหนด

```
vagrant@master:~/kic-training/lab-module2$ kubectl scale deploy coffee --replicas=4
deployment.apps/coffee scaled
vagrant@master:~/kic-training/lab-module2$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
coffee-6b8b6d6486-5tkwq	1/1	Running	0	32m
coffee-6b8b6d6486-7jwbq	1/1	Running	0	7s
coffee-6b8b6d6486-8cts8	1/1	Running	0	7s
coffee-6b8b6d6486-r8scc	1/1	Running	0	32m
tea-9d8868bb4-9lh2f	1/1	Running	0	32m
tea-9d8868bb4-kfvdf	1/1	Running	0	32m
tea-9d8868bb4-qklxn	1/1	Running	0	32m

```
vagrant@master:~/kic-training/lab-module2$
```

แก้ไข replicas กลับไปเป็น 2 เพื่อประหยัด resource ด้วย command `kubectl scale deploy coffee --replicas=2`

ทดสอบ delete Pod ด้วย command `kubectl delete pod {ชื่อ pod}`

เช่น `kubectl delete pod tea-9d8868bb4-qklxn` และลอง get pod ดูอีกครั้ง จะมี pod ใหม่ถูกสร้างขึ้นมาแทน pod ที่ถูกลบไปอัตโนมัติ เพื่อให้จำนวนของ Pod เท่ากับที่ set ไว้ใน replicas

```
vagrant@master:~/kic-training/lab-module2$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
coffee-6b8b6d6486-5tkwq	1/1	Running	0	38m
coffee-6b8b6d6486-r8scc	1/1	Running	0	38m
tea-9d8868bb4-9lh2f	1/1	Running	0	38m
tea-9d8868bb4-kfvdf	1/1	Running	0	38m
tea-9d8868bb4-zf6ct	1/1	Running	0	2s

Step 4

ตรวจสอบ Events และ Logs

สามารถตรวจสอบ events ใน namespace โดยใช้ command `kubectl get events` จะแสดง events ที่เกิดขึ้น เช่นการสร้าง deployment การสร้าง container รวมถึง error ด้วย

```
vagrant@master:~/kic-training/lab-module2$ kubectl get events
```

LAST SEEN	TYPE	REASON	OBJECT	MESSAGE
23m	Normal	Scheduled	pod/coffee-6b8b6d6486-5tkwq	Successfully assigned default/coffee-6b8b6d6486-5tkwq to worker01
23m	Normal	Pulled	pod/coffee-6b8b6d6486-5tkwq	Container image "nginxdemos/nginx-hello:plain-text" already present on machine
23m	Normal	Created	pod/coffee-6b8b6d6486-5tkwq	Created container coffee
23m	Normal	Started	pod/coffee-6b8b6d6486-5tkwq	Started container coffee
23m	Normal	Scheduled	pod/coffee-6b8b6d6486-r8scc	Successfully assigned default/coffee-6b8b6d6486-r8scc to worker01
23m	Normal	Pulled	pod/coffee-6b8b6d6486-r8scc	Container image "nginxdemos/nginx-hello:plain-text" already present on machine
23m	Normal	Created	pod/coffee-6b8b6d6486-r8scc	Created container coffee
23m	Normal	Started	pod/coffee-6b8b6d6486-r8scc	Started container coffee
23m	Normal	SuccessfulCreate	replicaset/coffee-6b8b6d6486	Created pod: coffee-6b8b6d6486-5tkwq
23m	Normal	SuccessfulCreate	replicaset/coffee-6b8b6d6486	Created pod: coffee-6b8b6d6486-r8scc
23m	Normal	ScalingReplicaSet	deployment/coffee	Scaled up replica set coffee-6b8b6d6486 to 2
23m	Normal	Scheduled	pod/tea-9d8868bb4-9lh2f	Successfully assigned default/tea-9d8868bb4-9lh2f to worker01
23m	Normal	Pulled	pod/tea-9d8868bb4-9lh2f	Container image "nginxdemos/nginx-hello:plain-text" already present on machine
23m	Normal	Created	pod/tea-9d8868bb4-9lh2f	Created container tea
23m	Normal	Started	pod/tea-9d8868bb4-9lh2f	Started container tea
23m	Normal	Scheduled	pod/tea-9d8868bb4-kfvdf	Successfully assigned default/tea-9d8868bb4-kfvdf to worker01
23m	Normal	Pulled	pod/tea-9d8868bb4-kfvdf	Container image "nginxdemos/nginx-hello:plain-text" already present on machine

หากต้องการดู logs ของแต่ละ pod สามารถใช้ command `kubectl logs {ชื่อ pod}`

เช่น `kubectl logs coffee-6b8b6d6486-5tkwq` จะเป็นการแสดง logs ของ container ที่อยู่ใน Pod

```
vagrant@master:~/kic-training/lab-module2$ kubectl logs coffee-6b8b6d6486-5tkwq
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf is not a file or does not exist
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/12/11 10:52:26 [warn] 1#1: the "user" directive makes sense only if the master process runs with super
nginx: [warn] the "user" directive makes sense only if the master process runs with super-user privileges,
2023/12/11 10:52:26 [notice] 1#1: using the "epoll" event method
```

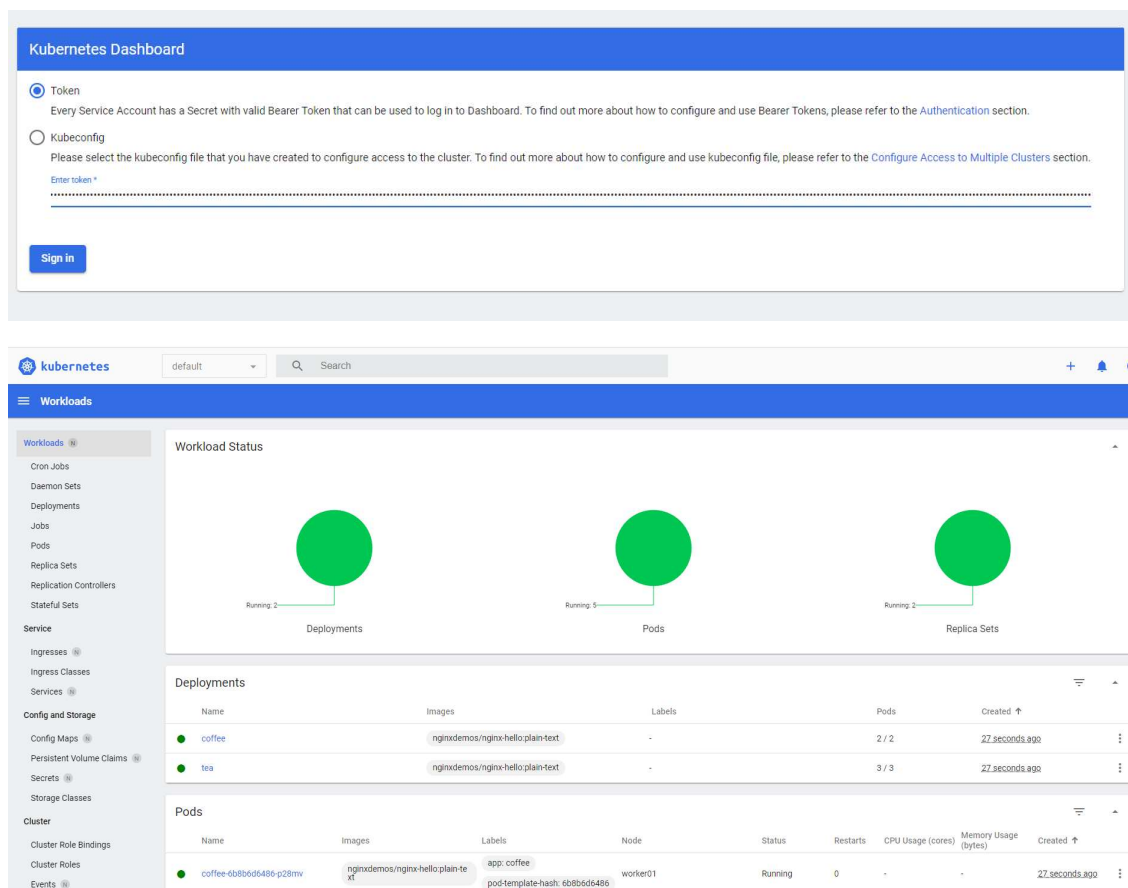

ใช้ Esc เพื่อออกจาก mode edit จากนั้นพิมพ์ :wq ตามด้วย Enter เพื่อ save config

ตรวจสอบ service อีกครั้งด้วย command `kubectl get svc -n Kubernetes-dashboard`

```
vagrant@master:~/kic-training/lab-module2$ kubectl get svc -n kubernetes-dashboard
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
dashboard-metrics-scraper	ClusterIP	10.101.132.243	<none>	8000/TCP	25m
kubernetes-dashboard	NodePort	10.102.165.236	<none>	443:30010/TCP	25m

เข้า Dashboard ด้วย web browser ใส่ URL `https://192.168.100.2:30010` ใส่ token ที่ได้จาก step ก่อนหน้านี้ จากนั้นกด Sign In



The screenshot shows the Kubernetes Dashboard interface. At the top, there's a 'Kubernetes Dashboard' header with two authentication options: 'Token' (selected) and 'Kubeconfig'. Below the 'Token' option, there's a 'Sign in' button. The main content area displays 'Workload Status' with three green circles representing 'Deployments', 'Pods', and 'Replica Sets'. Below this, there are two tables: 'Deployments' and 'Pods'. The 'Deployments' table shows two deployments: 'coffee' and 'tea', both using the 'nginxdemos/nginx-hello:plain-text' image. The 'Pods' table shows a single pod named 'coffee-6b8b6d6486-p28mrv' running on the 'worker01' node.

สามารถดู object ต่างๆได้ เช่น Deployment, Pod, Services และอื่นๆ

Step 6

Delete application ที่ทำการ Deploy ไปก่อนหน้านี้เพื่อเตรียมสำหรับ LAB Module 3 โดยใช้ command

```
kubectl delete -f coffee-deploy.yaml
```

```
kubectl delete -f coffee-svc.yaml
```

```
kubectl delete -f tea-deploy.yaml
```

```
kubectl delete -f tea-svc.yaml
```

ใช้ command `kubectl get all` เพื่อตรวจสอบอีกครั้ง จะต้องไม่มี deployment, service, pod ของ app coffee และ tea อยู่

```
vagrant@master:~/kic-training/lab-module2$ kubectl get all
NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                 ClusterIP     10.96.0.1     <none>         443/TCP    6h26m
vagrant@master:~/kic-training/lab-module2$
```