

LAB Guide

Module 3: NGINX Ingress Controller

ใน lab นี้จะเป็นการแนะนำขั้นตอนการ Deploy NGINX Ingress Controller และการ config Ingress Resource สำหรับทำ Routing ให้กับ Application ที่อยู่ด้านใน Kubernetes Cluster

Step 1

SSH ไปที่ Master ด้วย command `vagrant ssh master` จากนั้นเข้าไปที่ Directory `kic-training/lab-module3` ด้วย command `cd kic-training/lab-module3`

Set up role-based access control (RBAC)

```
kubectl apply -f common/ns-and-sa.yaml
```

```
kubectl apply -f rbac/rbac.yaml
```

Create common resources

```
kubectl apply -f common/nginx-config.yaml
```

```
kubectl apply -f common/ingress-class.yaml
```

Create custom resources

```
kubectl apply -f common/crds/k8s.nginx.org_virtualservers.yaml
```

```
kubectl apply -f common/crds/k8s.nginx.org_virtualserverroutes.yaml
```

```
kubectl apply -f common/crds/k8s.nginx.org_transportservers.yaml
```

```
kubectl apply -f common/crds/k8s.nginx.org_policies.yaml
```

Step 2

Deploy NGINX Ingress Controller จะมีการ Deploy 2 แบบคือ

1. Deployment: สามารถกำหนดจำนวนของ Ingress Controller ที่จะ Deploy ได้โดยใช้ ReplicasSet
2. DaemonSet: จะ Deploy Ingress Controller ไปทุกๆ Node ที่อยู่ใน Cluster หากมี Node เพิ่มขึ้นมา จะ Deploy Ingress Controller ให้เองอัตโนมัติ

ใน Lab นี้จะทำการ deploy แบบ deployment

```
kubectl apply -f deployment/nginx-ingress.yaml
```

ตรวจสอบ status ของ NGINX Ingress Controller โดยใช้ command

```
kubectl get pods --namespace=nginx-ingress
```

```
vagrant@master:~/kic-training/lab-module3$ kubectl get pod -n nginx-ingress
NAME                                READY   STATUS    RESTARTS   AGE
nginx-ingress-658bdcbb8-x88vp       1/1     Running   0           62m
vagrant@master:~/kic-training/lab-module3$
```

Expose NGINX Ingress Controller เพื่อให้สามารถ Access Application ที่อยู่ใน Cluster ได้ โดยใช้

```
command kubectl apply -f service/nodeport.yaml
```

ใช้ kubectl get svc -n nginx-ingress ตรวจสอบ port ที่จะใช้ในการ access application

```
vagrant@master:~/kic-training/lab-module3$ kubectl get svc -n nginx-ingress
NAME            TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
nginx-ingress   NodePort    10.111.91.182 <none>         80:30080/TCP,443:30443/TCP 60m
vagrant@master:~/kic-training/lab-module3$
```

ทดสอบ access ด้วย `curl http://192.168.100.3:30080` จะได้ผลลัพธ์เป็น 404 Not found เนื่องจากยังไม่
 ไม่มี config ของ ingress resource

```
vagrant@master:~/kic-training/lab-module3$ curl http://192.168.100.3:30080
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.25.3</center>
</body>
</html>
vagrant@master:~/kic-training/lab-module3$
```

Step 3

Deploy Cafe app ด้านในประกอบไปด้วย Service coffee และ tea โดยใช้ command

`kubectl apply -f cafe.yaml`

ตรวจสอบ application ที่ deploy ไปด้วย command `kubectl get all` จะ Pod และ Service ถูกสร้างขึ้น

โดย Service ที่สร้างมานั้นจะเป็น ClusterIP ไม่สามารถ access ได้จากภายนอก Cluster

```
vagrant@master:~/kic-training/lab-module3$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/coffee-6b8b6d6486-p28mv	1/1	Running	0	18h
pod/coffee-6b8b6d6486-q2lzn	1/1	Running	0	18h
pod/tea-9d8868bb4-7kdw	1/1	Running	0	18h
pod/tea-9d8868bb4-vf267	1/1	Running	0	18h
pod/tea-9d8868bb4-zntjj	1/1	Running	0	18h

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/coffee-svc	ClusterIP	10.103.160.206	<none>	80/TCP	18h
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	25h
service/tea-svc	ClusterIP	10.104.88.63	<none>	80/TCP	18h

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/coffee	2/2	2	2	18h
deployment.apps/tea	3/3	3	3	18h

Step 4

สร้าง Ingress Resource เพื่อ expose application café ด้วยใช้ command

```
kubectl apply -f cafe-ingress-http.yaml
```

```
apiVersion: networking.k8s.io/v1
kind: Ingress      → สร้าง Object Ingress
metadata:
  name: cafe-ingress → กำหนดชื่อของ ingress
spec:
  ingressClassName: nginx → กำหนดว่า ingress resource นี้จะผูกกับ controller ตัวไหน
  rules:
    - host: cafe.example.com → Match hostname เท่า cafe.example.com
      http:
        paths:
          - path: /tea → Match path เท่า /tea
            pathType: Prefix
            backend:
              service:
                name: tea-svc → ให้ route ไปที่ tea-svc port 80
                port:
                  number: 80
          - path: /coffee → Match path เท่า /coffee
            pathType: Prefix
            backend:
              service:
                name: coffee-svc → ให้ route ไปที่ tea-svc port 80
                port:
                  number: 80
```

ใช้ command `kubectl get ingress` เพื่อ show ingress resource ที่มีใน namespace

```
vagrant@master:~/kic-training/lab-module3$ kubectl get ingress
NAME          CLASS    HOSTS          ADDRESS    PORTS    AGE
cafe-ingress  nginx    cafe.example.com  80        17m
vagrant@master:~/kic-training/lab-module3$
```

ใช้ command `kubectl describe ingress cafe-ingress` เพื่อ show รายละเอียดของ ingress จะมี ip address ของ Pod ขึ้นมาในส่วนของ Backends

```
vagrant@master:~/kic-training/lab-module3$ kubectl describe ingress cafe-ingress
Name:          cafe-ingress
Labels:        <none>
Namespace:     default
Address:
Ingress Class: nginx
Default backend: <default>
Rules:
  Host      Path      Backends
  ----      -
  cafe.example.com
    /tea     tea-svc:80 (10.244.192.3:8080,10.244.192.4:8080,10.244.192.5:8080)
    /coffee coffee-svc:80 (10.244.192.6:8080,10.244.192.7:8080)
Annotations:  <none>
Events:
  Type      Reason      Age      From      Message
  ----      -
  Warning    Rejected     17m      nginx-ingress-controller spec.rules[0].host: Required value
  Normal    AddedOrUpdated 7m58s    nginx-ingress-controller Configuration for default/cafe-ingress was added or updated
```

ทดสอบ access application โดยใช้

`curl -H 'Host: cafe.example.com' http://192.168.100.3:30080/coffee`

`-H 'Host: cafe.example.com'` เป็นการกำหนดค่าของ headers เพื่อให้ตรงกับเงื่อนไขของ ingress resource ที่สร้างไว้ หากไม่ `-H` จะต้องไปแก้ไข file hosts บนเครื่องที่จะทดสอบ หรือเพิ่ม dns ก็ได้เช่นกัน

```
vagrant@master:~/kic-training/lab-module3$ curl -H 'Host: cafe.example.com' http://192.168.100.3:30080/coffee
Server address: 10.244.192.7:8080
Server name: coffee-6b8b6d6486-p28mv
Date: 12/Dec/2023:07:40:30 +0000
URI: /coffee
Request ID: 4c9dc58d6843746e64a7543e2c6f0028
```

server name จะขึ้นต้นด้วย coffee

ใช้ `curl -H 'Host: cafe.example.com' http://192.168.100.3:30080/tea`

```
vagrant@master:~/kic-training/lab-module3$ curl -H 'Host: cafe.example.com' http://192.168.100.3:30080/tea
Server address: 10.244.192.5:8080
Server name: tea-9d8868bb4-vf267
Date: 12/Dec/2023:07:46:06 +0000
URI: /tea
Request ID: d45994fdde3ce5f47a8c3646aa834753
```

server name จะเปลี่ยนเป็น tea ตาม routing ที่กำหนดไว้

Step 5

เพิ่ม ingress resource ให้รองรับกับ application ที่เพิ่มขึ้น โดยจะ deploy application ที่ชื่อว่า webapp และกำหนด hostname เป็น webapp.example.com

Deploy webapp โดยใช้ command `kubectl apply -f webapp.yaml`

```
vagrant@master:~/kic-training/lab-module3$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/coffee-6b8b6d6486-p28mv	1/1	Running	0	19h
pod/coffee-6b8b6d6486-q2lzn	1/1	Running	0	19h
pod/tea-9d8868bb4-7kdw	1/1	Running	0	19h
pod/tea-9d8868bb4-vf267	1/1	Running	0	19h
pod/tea-9d8868bb4-zntjj	1/1	Running	0	19h
pod/webapp-fd76d5df6-p74n4	1/1	Running	0	16s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/coffee-svc	ClusterIP	10.103.160.206	<none>	80/TCP	19h
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	26h
service/tea-svc	ClusterIP	10.104.88.63	<none>	80/TCP	19h
service/webapp-svc	ClusterIP	10.100.85.143	<none>	80/TCP	16s

ใช้ command `kubectl apply -f webapp-ingress-http.yaml` เพื่อเพิ่ม config ของ ingress resource

ใช้ command `kubectl describe ingress webapp-ingress` เพื่อ show รายละเอียดของ ingress

```
vagrant@master:~/kic-training/lab-module3$ kubectl describe ingress webapp-ingress
```

```
Name:          webapp-ingress
Labels:        <none>
Namespace:    default
Address:
Ingress Class: nginx
Default backend: <default>
Rules:
  Host            Path    Backends
  ----            -
  webapp.example.com
                  /      webapp-svc:80 (10.244.192.9:8080)
Annotations:    <none>
Events:
  Type    Reason      Age    From                      Message
  ----    -
  Warning  Rejected    64s    nginx-ingress-controller  All hosts are taken by other resources
  Normal  AddedOrUpdated  5s    nginx-ingress-controller  Configuration for default/webapp-ingress was added or updated
```

ทดสอบ access application โดยใช้

```
curl -H 'Host: webapp.example.com' http://192.168.100.3:30080
```

```
vagrant@master:~/kic-training/lab-module3$ curl -H 'Host: webapp.example.com' http://192.168.100.3:30080
Server address: 10.244.192.9:8080
Server name: webapp-fd76d5df6-p74n4
Date: 12/Dec/2023:07:57:32 +0000
URI: /
Request ID: 7489d94922e7ef1a4b4cce339c3324ea
vagrant@master:~/kic-training/lab-module3$
```

server name จะขึ้นต้นด้วย webapp ตาม config ใหม่ที่ update ไป

Step 6

SSL Terminate โดยสร้าง object ที่ชื่อว่า secret เพื่อเก็บ certificate และ private key ใน lab จะใช้เป็น self sign cert

สร้าง secret โดยใช้ command `kubectl apply -f cafe-secret.yaml`

ตัวอย่างไฟล์

```
apiVersion: v1
kind: Secret
metadata:
  name: cafe-secret
type: kubernetes.io/tls
data:
  tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0
  tls.key: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVk=
```

เมื่อได้ secret แล้วให้ทำการ update ingress resource ด้วย command

`kubectl apply -f cafe-ingress-https.yaml`

จากนั้นใช้ `kubectl describe ingress cafe-ingress` จะมี config ของ TLS เพิ่มขึ้นมา

```
vagrant@master:~/kic-training/lab-module3$ kubectl describe ingress cafe-ingress
Name:          cafe-ingress
Labels:        <none>
Namespace:     default
Address:
Ingress Class: nginx
Default backend: <default>
TLS:
  cafe-secret terminates cafe.example.com
Rules:
  Host            Path  Backends
  ----            -
  cafe.example.com /tea  tea-svc:80 (10.244.192.3:8080,10.244.192.4:8080,10.244.192.5:8080)
                  /coffee coffee-svc:80 (10.244.192.6:8080,10.244.192.7:8080)
Annotations:    <none>
Events:
  Type    Reason      Age    From                      Message
  ----    -
  Warning  Rejected    59m    nginx-ingress-controller  spec.rules[0].host: Required value
  Normal   AddedOrUpdated 104s (x4 over 50m)  nginx-ingress-controller  Configuration for default/cafe-ingress was added or updated
```


ทดสอบเข้าใช้งานแบบ HTTP ด้วย command

```
curl --resolve cafe.example.com:30080:192.168.100.3 http://cafe.example.com:30080
```

```
vagrant@master:~/kic-training/lab-module3$ curl --resolve cafe.example.com:30080:192.168.100.3 http://cafe.example.com:30080
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx/1.25.3</center>
</body>
</html>
```

จะ response เป็น http status 301 หากเข้าด้วย browser จะ Redirect ไปหา HTTPS

ทดสอบเข้าใช้งานแบบ HTTPS ด้วย command

```
curl --resolve cafe.example.com:30443:192.168.100.3 https://cafe.example.com:30443/coffee --insecure
```

*30080 = HTTPS, 30443=HTTPS

```
vagrant@master:~/kic-training/lab-module3$ curl --resolve cafe.examl
Server address: 10.244.192.6:8080
Server name: coffee-6b8b6d6486-q2lzn
Date: 12/Dec/2023:08:32:10 +0000
URI: /coffee
Request ID: 7db3df1ce2d7dc14ec9fba4a33521072
vagrant@master:~/kic-training/lab-module3$
```

Step 7

Config Load balance method เนื่องจาก default ของ Kubernetes การจัดการ request จะเป็นการ random ไปหา Pod ต่างๆ สามารถกำหนดรูปแบบได้ เมื่อนำ ingress controller มาใช้งานสามารถให้ ingress controller จัดการตรงส่วนนี้ได้ โดยการเพิ่ม config ของ ingress resource ในส่วนของ Annotations

ตัวอย่าง config จะใช้เป็น ip_hash สามารถปรับเป็นแบบอื่นได้ เช่น least_conn, round_robin

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: cafe-ingress
  annotations:
    nginx.org/lb-method: "ip_hash"
spec:
  ingressClassName: nginx
  tls:
  - hosts:
    - cafe.example.com
    secretName: cafe-secret
  rules:
```

Update ingress โดยใช้ command `kubectl apply -f cafe-ingress-https-lb.yaml`

ทดสอบ access application โดยใช้ `curl --resolve cafe.example.com:30443:192.168.100.3 https://cafe.example.com:30443/tea --insecure` สังเกตตรง server name จะไม่มีการเปลี่ยนแปลง ซึ่งเป็นไปตามเงื่อนไขของ Load Balance Method แบบ IP Hash

Step 8

กำหนด Log Format และตรวจสอบ logs ของ ingress controller โดยการกำหนด log format จะใช้ Object ที่ชื่อว่า configMap

ตัวอย่างไฟล์

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-config
  namespace: nginx-ingress
data:
  log-format: '$remote_addr - $remote_user [$time_local] "$request" $status'
```

Apply configMap เพื่อแก้ไข log format ของ ingress controller ด้วย command

```
kubectl apply -f log-format.yaml
```

ตรวจสอบ logs ของ ingress controller โดยการใช้ `kubectl get pod -n nginx-ingress` เพื่อดูชื่อของ Pod

ใช้ `kubectl logs {ชื่อของ Pod} -n nginx-ingress` เช่น `kubectl logs nginx-ingress-658bdcbb8-x88vp -n nginx-ingress`

ตัวอย่าง logs format ก่อนแก้ไข

```
10.244.192.0 - - [12/Dec/2023:08:41:18 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:21 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:22 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:22 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:23 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:24 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:25 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:25 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:26 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
10.244.192.0 - - [12/Dec/2023:08:41:27 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-"
```

หลังจากแก้ไข log format หน้าตาของ log จะเปลี่ยนไปตามที่ update

```
2023/12/12 08:57:09 [notice] 13#13: signal 29 (SIGIO) received
10.244.192.0 - - [12/Dec/2023:08:58:21 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-" "cafe-ingress" "ingress" "default" "tea-svc"
10.244.192.0 - - [12/Dec/2023:08:58:22 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-" "cafe-ingress" "ingress" "default" "tea-svc"
10.244.192.0 - - [12/Dec/2023:08:58:23 +0000] "GET /tea HTTP/1.1" 200 155 "-" "curl/7.68.0" "-" "cafe-ingress" "ingress" "default" "tea-svc"
vagrant@master:~/kic-training/lab-module3$
```