🚪 Lab 5.2 – Gateway API & Ingress Routing with MetalLB

🎯 Objectives

- Install **NGINX Gateway Fabric** and configure a `Gateway` with `HTTPRoutes`
- Install **NGINX Ingress Controller** and configure `Ingress` resources
- Use **MetalLB** to expose services of type `LoadBalancer`
- Route traffic to the same **demo app** using both Gateway API and Ingress

---

🔧 Prerequisites

- A running **Kind** cluster with **MetalLB** installed
- You can install MetalLB using:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.13.12/config/manifests/metallb-native.yaml
```

Then configure an address pool (replace with your Docker bridge range):

```yaml
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: default-pool
  namespace: metallb-system
spec:
  addresses:
  - 172.18.255.200-172.18.255.250
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: l2adv
  namespace: metallb-system
```

See previous lab for more details on MetalLB setup.

🧱 Step 1 – Deploy the Sample Applications (Coffee & Tea)

```yaml
# cafe.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: coffee
spec:
  replicas: 1
  selector:
    matchLabels:
      app: coffee
  template:
    metadata:
      labels:
        app: coffee
    spec:
      containers:
      - name: coffee
        image: nginxdemos/nginx-hello:plain-text
        ports:
        - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: coffee
spec:
  ports:
  - port: 80
    targetPort: 8080
    name: http
  selector:
    app: coffee
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tea
spec:
  replicas: 1
  selector:
    matchLabels:
      app: tea
  template:
    metadata:
      labels:
        app: tea
    spec:
      containers:
      - name: tea
        image: nginxdemos/nginx-hello:plain-text
        ports:
        - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: tea
spec:
  ports:
  - port: 80
    targetPort: 8080
    name: http
  selector:
    app: tea
```

```
kubectl apply -f module-5/manifests/coffee-tea.yaml
```

🌐 Step 2 – Install NGINX Gateway Fabric

a. Install Gateway API CRDs:

```
kubectl kustomize "https://github.com/nginx/nginx-gateway-fabric/config/crd/gateway-api/standard?ref=v2.0.1" | kubectl apply -f -
```

b. Install Gateway Fabric with MetalLB exposure:

```
helm install ngf oci://ghcr.io/nginx/charts/nginx-gateway-fabric \
  --create-namespace -n nginx-gateway \
  --set nginx.service.type=LoadBalancer
```

c. Create Gateway & HTTPRoutes

```
# gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: demo-gateway
  namespace: default
spec:
  gatewayClassName: nginx
  listeners:
  - name: http
    port: 80
    protocol: HTTP
    hostname: "*.gw.demo.k8s.local"
```

```yaml
# httproutes.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: coffee
spec:
  parentRefs:
  - name: demo-gateway
    sectionName: http
  hostnames:
  - "cafe.gw.demo.k8s.local"
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /coffee
    backendRefs:
    - name: coffee
      port: 80
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: tea
spec:
  parentRefs:
  - name: demo-gateway
    sectionName: http
  hostnames:
  - "cafe.gw.demo.k8s.local"
  rules:
  - matches:
    - path:
        type: Exact
        value: /tea
    backendRefs:
    - name: tea
      port: 80
```

```
kubectl apply -f module-5/manifests/gateway.yaml
kubectl apply -f module-5/manifests/httproutes.yaml
```

d. Update /etc/hosts

```
GW_IP=$(kubectl get svc demo-gateway-nginx -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo "$GW_IP cafe.gw.demo.k8s.local" | sudo tee -a /etc/hosts
```

🌍 Step 3 – Install NGINX Ingress Controller

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm install ingress-nginx ingress-nginx/ingress-nginx \
  --create-namespace -n ingress-nginx \
  --set controller.service.type=LoadBalancer
```

a. Update /etc/hosts

```
ING_IP=$(kubectl get svc -n ingress-nginx ingress-nginx-controller -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo "$ING_IP cafe.ing.demo.k8s.local" | sudo tee -a /etc/hosts
```

b. Create Ingress Resource

```yaml
# ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: demo-ingress
  namespace: default
spec:
  ingressClassName: nginx
  rules:
  - host: cafe.ing.demo.k8s.local
    http:
      paths:
      - path: /coffee
        pathType: Prefix
        backend:
          service:
            name: coffee
            port:
              number: 80
      - path: /tea
        pathType: Prefix
        backend:
          service:
            name: tea
            port:
              number: 80
```

```
kubectl apply -f module-5/manifests/ingress.yaml
```

✅ Test

```
curl --resolve cafe.gw.demo.k8s.local:80:$GW_IP http://cafe.gw.demo.k8s.local/coffee
curl --resolve cafe.gw.demo.k8s.local:80:$GW_IP http://cafe.gw.demo.k8s.local/tea

curl --resolve cafe.ing.demo.k8s.local:80:$ING_IP http://cafe.ing.demo.k8s.local/coffee
curl --resolve cafe.ing.demo.k8s.local:80:$ING_IP http://cafe.ing.demo.k8s.local/tea
```

🧼 Cleanup

```
sudo sed -i '' '/demo.k8s.local/d' /etc/hosts
kubectl delete -f module-5/manifests
helm uninstall ngf -n nginx-gateway
helm uninstall ingress-nginx -n ingress-nginx
```

✅ End of Lab – You’ve deployed routing with Gateway API and Ingress using MetalLB!