

## Lab 5.1 – Service Types & CoreDNS (Networking Fundamentals)

---

### Objectives

- Create and test different **Service types**: ClusterIP, NodePort, LoadBalancer
  - Validate **DNS resolution** using CoreDNS and `nslookup`
- 

### Prerequisites

► ► For MacOS Users only: make your docker network reachable

```
# Install via Homebrew
$ brew install chipmk/tap/docker-mac-net-connect

# Run the service and register it to launch at boot
$ sudo brew services start chipmk/tap/docker-mac-net-connect
```

This will allow your Docker containers to communicate with the host network, enabling services like MetalLB to function correctly.

Create a new kind cluster with MetalLB enabled:

```
# kind-config.yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
name: services-cluster
networking:
  kubeProxyMode: "iptables" # Use iptables for networking
nodes:
  - role: control-plane
    image: kindest/node:v1.32.2
  - role: worker
    image: kindest/node:v1.32.2
```

```
kind create cluster --config module-5/manifests/kind.yaml
```

## Step 1 – Deploy MetalLB (for LoadBalancer services)

Before deploying MetalLB, lets grab unused IP addresses from your network. For example, if your network is `172.18.0.0/16`, you might choose `172.18.255.200-172.18.255.250` for MetalLB.

```
docker inspect kind | jq ".[].IPAM.Config"
```

1. Install MetalLB manifests:

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.14.8/config/manifests/metallb-native.yaml
```

2. Create an IPAddressPool and L2Advertisement:

```
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  namespace: metallb-system
  name: first-pool
spec:
  addresses:
    - 172.18.255.200-172.18.255.250
---
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  namespace: metallb-system
  name: l2advertisement
```

Apply with:

```
kubectl apply -f module-5/manifests/metallb-config.yaml
```

---

## Step 2 – Create a Basic Deployment

```
kubectl create deployment webapp --image=nginx --port=80
```

---

## Step 3 – Expose with Different Service Types

ClusterIP (default)

```
kubectl expose deployment webapp --port=80 --target-port=80  
  --type=ClusterIP  
kubectl get svc webapp
```

## NodePort

```
kubectl create deployment webapp-nodeport --image=nginx --port=80  
kubectl expose deployment webapp-nodeport --port=80 --target-port=80 --type=NodePort  
kubectl get svc webapp-nodeport
```

## LoadBalancer (via MetalLB)

```
kubectl create deployment webapp-lb --image=nginx --port=80  
kubectl expose deployment webapp-lb --port=80 --target-port=80 --type=LoadBalancer  
kubectl get svc webapp-lb -o wide
```

---

## Step 4 – Test DNS Resolution with CoreDNS

Launch a debug pod:

```
kubectl run -it dnsutils --image=busybox:1.28 --restart=Never --rm -- nslookup webapp
```

Or use:

```
kubectl exec -it <any-pod> -- nslookup webapp
```

---

## Cleanup

```
kubectl delete deployment webapp webapp-nodeport webapp-lb  
kubectl delete svc webapp webapp-nodeport webapp-lb
```

---

We will use the same cluster for the next labs, so we won't delete the cluster itself.

👉 End of Lab 5.1 – You've explored service types and validated cluster DNS with CoreDNS and MetalLB