

# [ Network ] K8S Overlay Network ( IPIP -> VXLAN )

## K8S Overlay Network

### IPIP -> VXLAN

) POD가  
( pod 가 )

### Calico IP-IP Network VXLAN

Node : Controller / Worker01 / Worker02

```
## Controller
# Mode          IPIPMode
calicoctl get ippool -o wide
NAME            CIDR            NAT          IPIPMode
VXLANMode      Disabled      DisableBGPExport  Selector
default-ipv4-ippool  192.168.0.0/16  true         Always      Never
false          false
all()
```

```
# Manifest YAML
kubect delete -f calico.yml
```

```
## Controller / Worker
# 가          tunl0          가
sudo rm -rf /var/run/calico/
sudo rm -rf /var/lib/calico/
sudo rm -rf /etc/cni/net.d/
sudo rm -rf /var/lib/cni/
sudo reboot
```

```
## Controller
# Manifest. calico.yml          VXLAN
```

```
livenessProbe:
  exec:
    command:
      - /bin/calico-node
      - -felix-live
      # - -bird-live          // VXLAN    bird(BGP)

    periodSeconds: 10
    initialDelaySeconds: 10
    failureThreshold: 6
    timeoutSeconds: 10
  readinessProbe:
    exec:
      command:
        - /bin/calico-node
        - -felix-ready
        # - -bird-ready      //

# Enable IPIP
- name: CALICO_IPV4POOL_IPIP
  value: "Never"          // Always --> Never

# Enable or Disable VXLAN on the default IP pool.
- name: CALICO_IPV4POOL_VXLAN
  value: "Always"        // Never --> Always

kind: ConfigMap
apiVersion: v1
metadata:
  name: calico-config
  namespace: kube-system
data:
  # Typha is disabled.
  typha_service_name: "none"
  # Configure the backend to use.
  calico_backend: "vxlan"          // "bird" --> "vxlan"
  .

#
kubectly apply -f calico.yaml
```

```
# Calico Node Ready
kubectl get nodes -o wide -A
```

```
# Calico Pod kube-system PoD 가
kubectl get pod -o wide -A
```

```
# Calico Type BIRD
sudo calicoctl node status
Calico process is running.
The BGP backend process (BIRD) is not running.
```

```
# Network VXLANMODE 가
calicoctl get ippool -o wide
NAME CIDR NAT IPIPMODE
VXLANMODE DISABLED DISABLEBGPEXPORT SELECTOR
default-ipv4-ippool 192.168.0.0/16 true Never
Always false false
all()
```

```
# tunl0 가 vxlan 가
# vxlan 가
```

```
hostway@controller:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric
Ref Use Iface
0.0.0.0 10.10.10.1 0.0.0.0 UG 0 0
0 ens18
10.10.10.0 0.0.0.0 255.255.255.0 U 0 0
0 ens18 // External (SNAT)
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0
0 docker0 // Container Runtime Bridge
192.168.5.0 192.168.5.0 255.255.255.192 UG 0 0
0 vxlan.calico // Worker01
192.168.30.64 192.168.30.64 255.255.255.192 UG 0 0
0 vxlan.calico // Worker02
192.168.49.0 0.0.0.0 255.255.255.192 U 0 0
0 * // Controller vxlan
192.168.49.1 0.0.0.0 255.255.255.255 UH 0 0
0 cali09ae4a7064b // Node(Worker01)가 GW
192.168.49.2 0.0.0.0 255.255.255.255 UH 0 0
0 cali1fdac863dc5 // Node(Worker02)가 GW
```

# Worker

```
hostway@controller:~$ ip netns | grep vxlan
192.168.5.0 dev vxlan.calico lladdr 66:8c:33:86:44:ce
PERMANENT
192.168.30.64 dev vxlan.calico lladdr 66:fb:72:20:22:a1
PERMANENT
```

# VXLAN Traffic Port UDP

```
udp 0 0 0.0.0.0:4789 0.0.0.0:*
```

# PoD

```
hostway@controller:~$ kubectl create deployment sampleos --
image=gcr.io/google-samples/kubernetes-bootcamp:v1 --
replicas=3
```

deployment.apps/sampleos created

```
hostway@controller:~$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE
IP	NOMINATED	NODE	READINESS GATES	
sampleos-646dc9654b-8xjw9	1/1	Running	0	45s
192.168.5.11	<none>	worker01	<none>	
sampleos-646dc9654b-gxn75	1/1	Running	0	45s
192.168.5.10	<none>	worker01	<none>	
sampleos-646dc9654b-snkxg	1/1	Running	0	45s
192.168.30.75	<none>	worker02	<none>	

# VXLAN

// Controller

1) worker01 worker02 POD Ping

```
hostway@controller:~$ kubectl exec -it
```

```
sampleos-646dc9654b-8xjw9 -- ping 192.168.30.75
```

```
PING 192.168.30.75: 56 data bytes
```

```
64 bytes from 192.168.30.75: icmp_seq=0 ttl=115 time=92.124 ms
```

```
64 bytes from 192.168.30.75: icmp_seq=1 ttl=115 time=79.735 ms
```

```
64 bytes from 192.168.30.75: icmp_seq=2 ttl=115 time=79.233 ms
```

2) tcpdump

```
sudo tcpdump -i ens18 -w vxlan.pcap
```

3) Wireshark

. UDP

.

