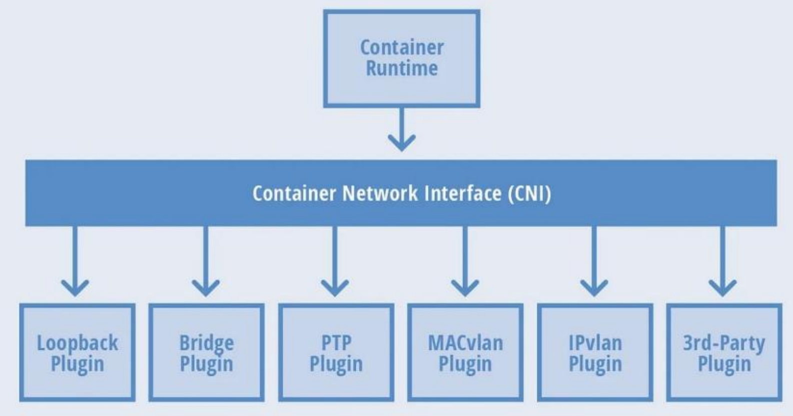
Container Network Interface (CNI) for K8s

- CNI Introduction
- CNI Types
  - ClusterIP
  - NodePort
  - Loadbalanced
- K8s Ingress
- CNI Plugins
  - Flannel
  - Calico
  - Weave
  - Cilium
  - Canal

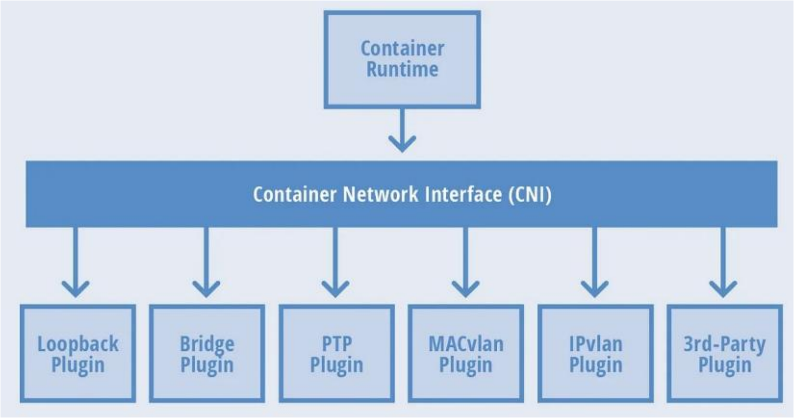
CNI

- What is CNI
  - Container Network Interface
  - k8s networking
  - consists of a specification and libs for writing plugins to configure container network interfaces
  - Focus on connectivity to containers and remove when they deleted



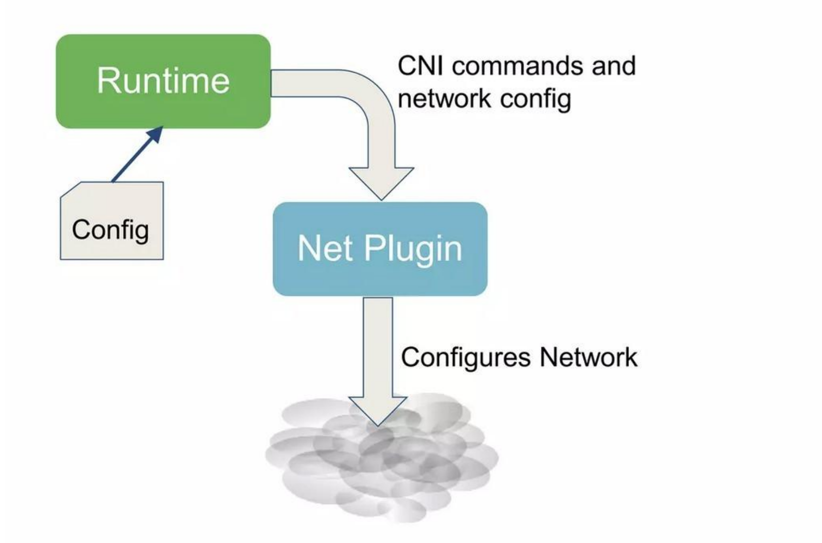
CNI

- What is CNI
  - CNI is called twice by k8's kubelet to set up loopback and eth0 interfaces for a pod
    - 1. when the K8s kubelet sets up loopback and eth0 interfaces for a pod
    - 2. when the K8s kubelet sets up loopback and eth0 interfaces for an external interface connectable to an external IP address



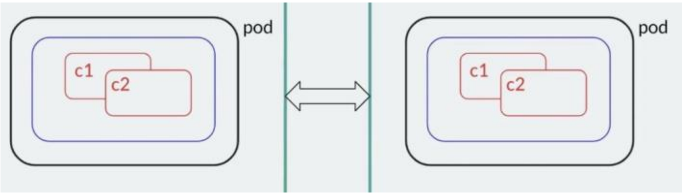
CNI

- What is CNI



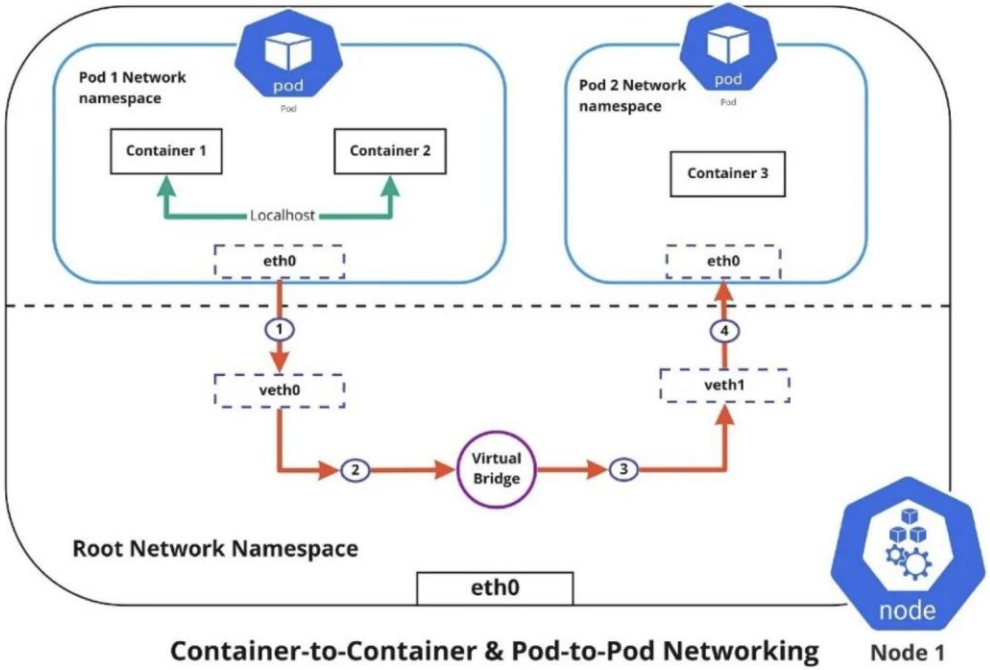
CNI

- CNI Provides
  - Cross node pod to pod communication
    - pod <==> pod without NAT
  - Service discovery
  - Services exposure for external access
  - Network security
  - High availability



CNI

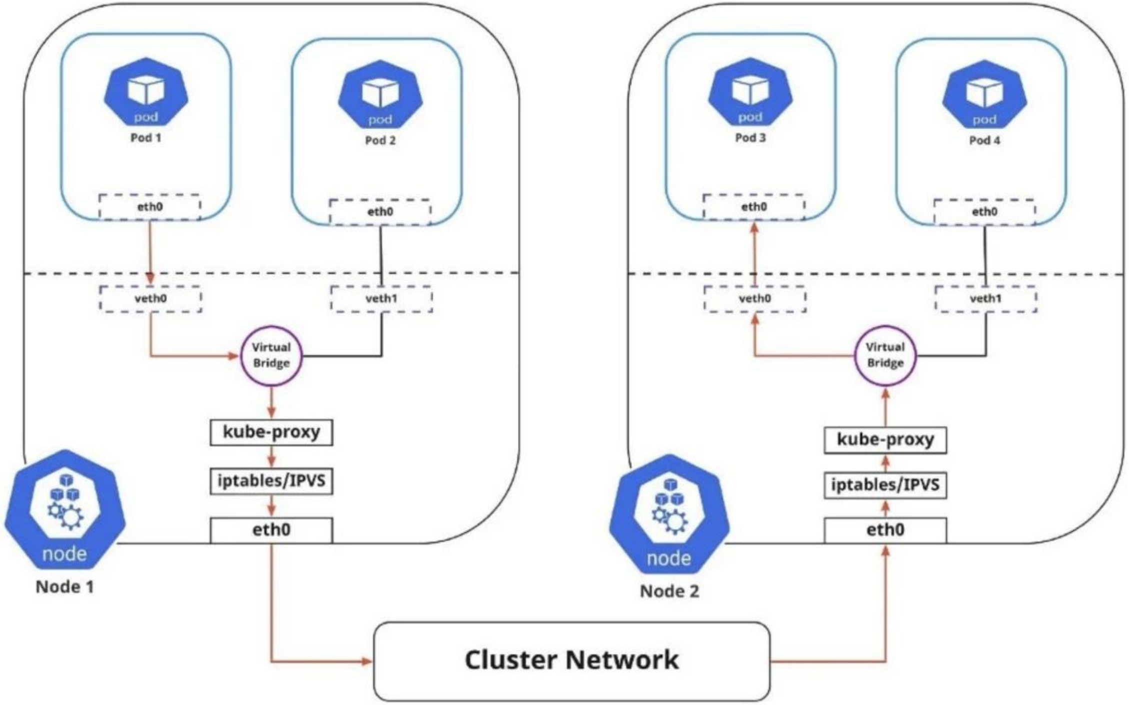
- Pod to pod



Container-to-Container & Pod-to-Pod Networking

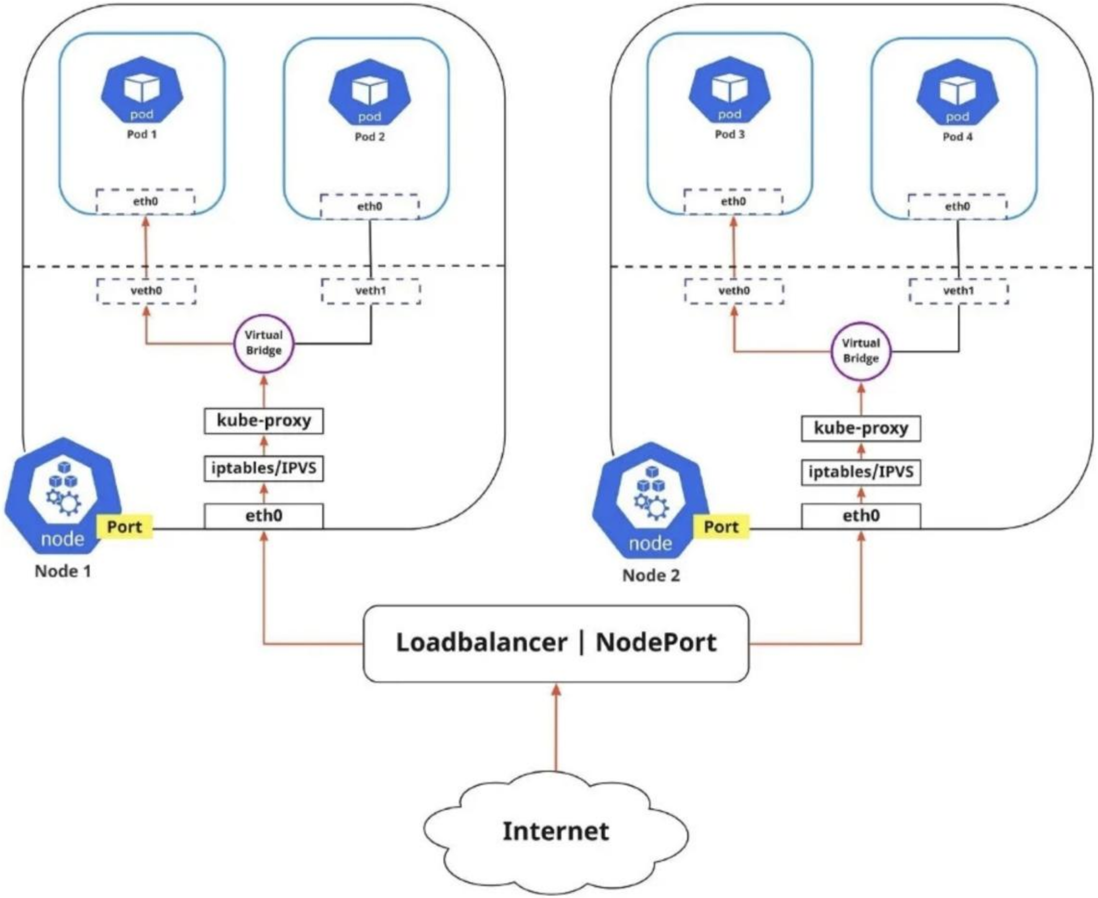
CNI

- Cluster network



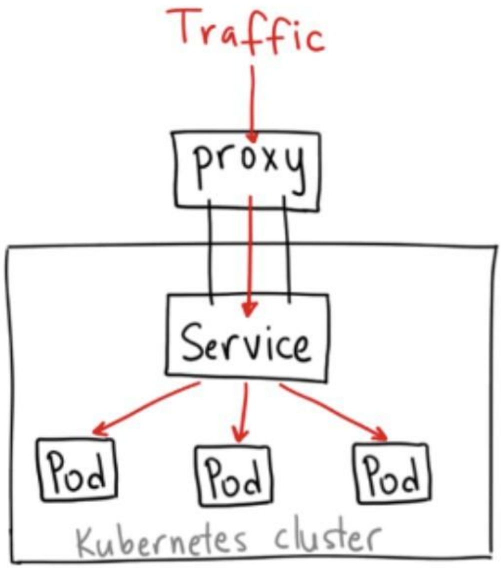
CNI

- Internet to Service network

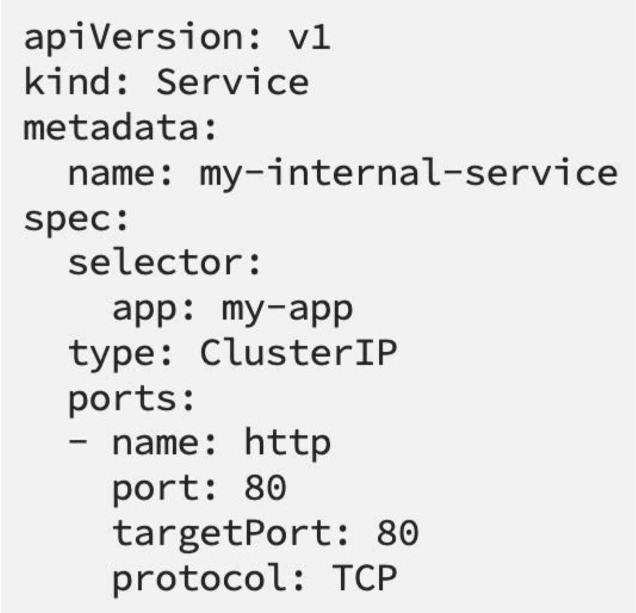


CNI

- K8s service types
  - ClusterIP (default service)
    - Service only reachable within the cluster
    - Apps within the cluster can communicate each other



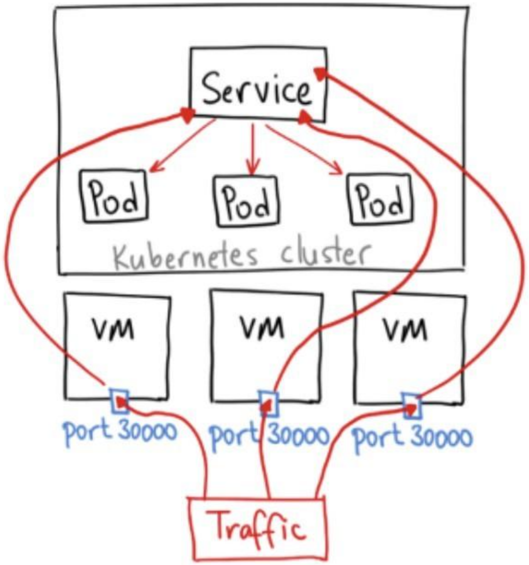
ClusterIP



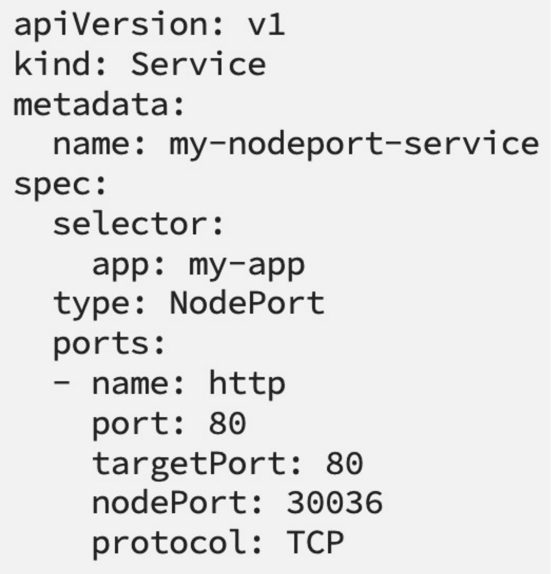
```
apiVersion: v1
kind: Service
metadata:
  name: my-internal-service
spec:
  selector:
    app: my-app
  type: ClusterIP
  ports:
  - name: http
    port: 80
    targetPort: 80
    protocol: TCP
```

CNI

- K8s service types
  - NodePort
    - allows the external traffic to access the Service by opening a specific port on all the nodes
    - The most primitive way
    - Many issues
      - one service per port
      - only use ports 30000–32767
      - Node/VM IP address change, you need to deal with that
    - Can not be use in production



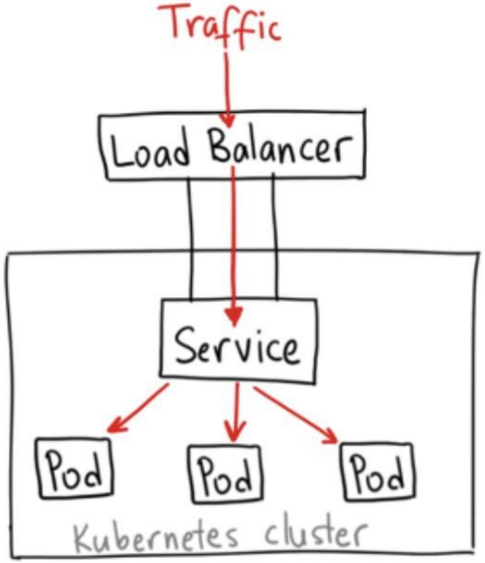
NodePort



```
apiVersion: v1
kind: Service
metadata:
  name: my-nodeport-service
spec:
  selector:
    app: my-app
  type: NodePort
  ports:
  - name: http
    port: 80
    targetPort: 80
    nodePort: 30036
    protocol: TCP
```

CNI

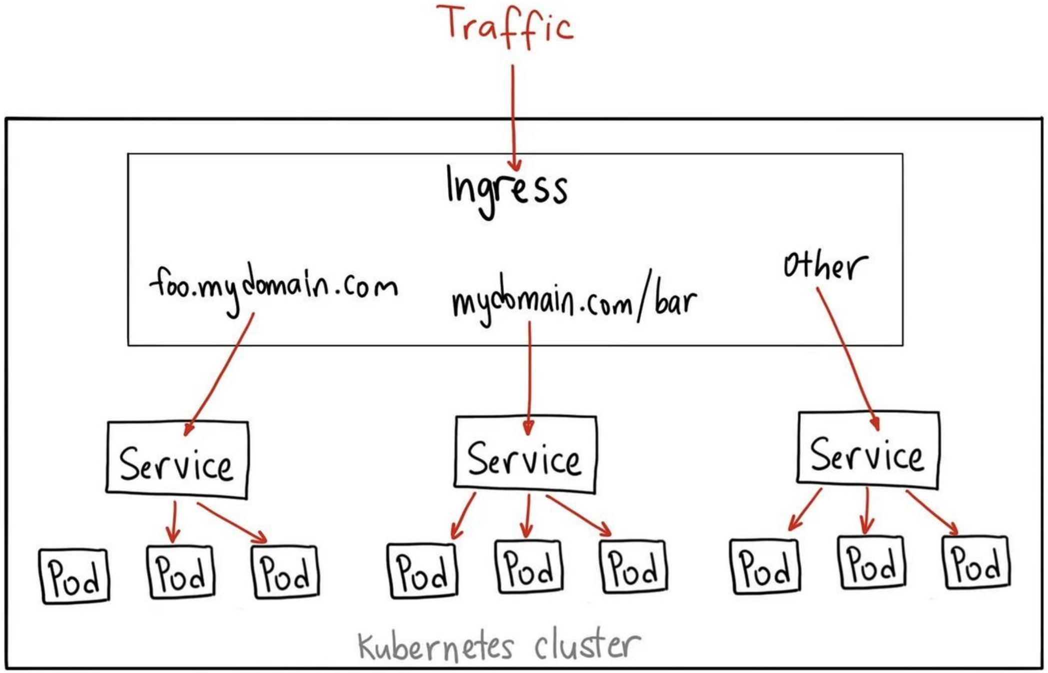
- K8s service types
  - LoadBalancer
    - standard way to expose a service to the Internet
    - need a load-balancer
      - External or Internal
    - need a separate IP for each service



Loadbalancer

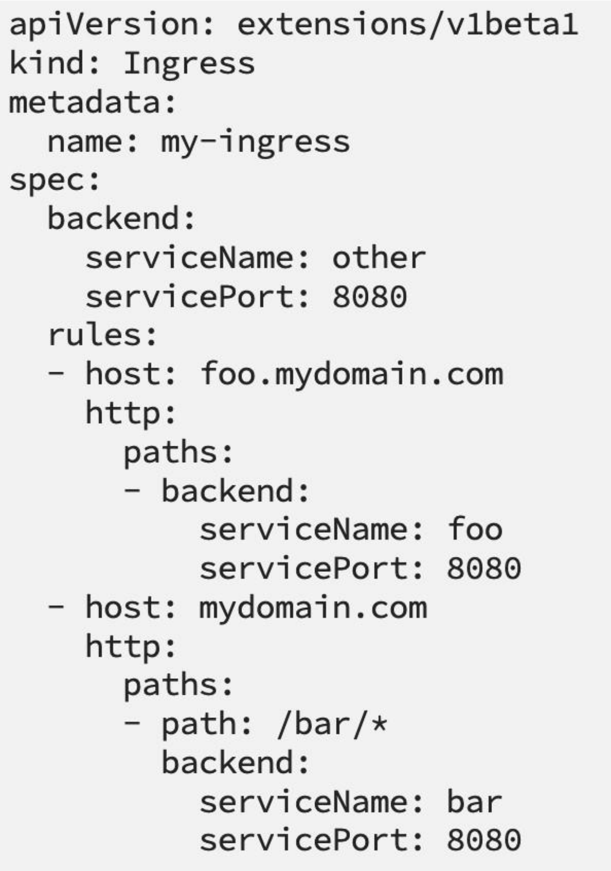
CNI

- K8s Ingress (not a type of service)
  - Sit in-front of multiple services
  - Act as smart router or entry point
  - Load balance based on domain name



CNI

- K8s Ingress



```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
spec:
  backend:
    serviceName: other
    servicePort: 8080
  rules:
  - host: foo.mydomain.com
    http:
      paths:
      - backend:
          serviceName: foo
          servicePort: 8080
  - host: mydomain.com
    http:
      paths:
      - path: /bar/*
        backend:
          serviceName: bar
          servicePort: 8080
```

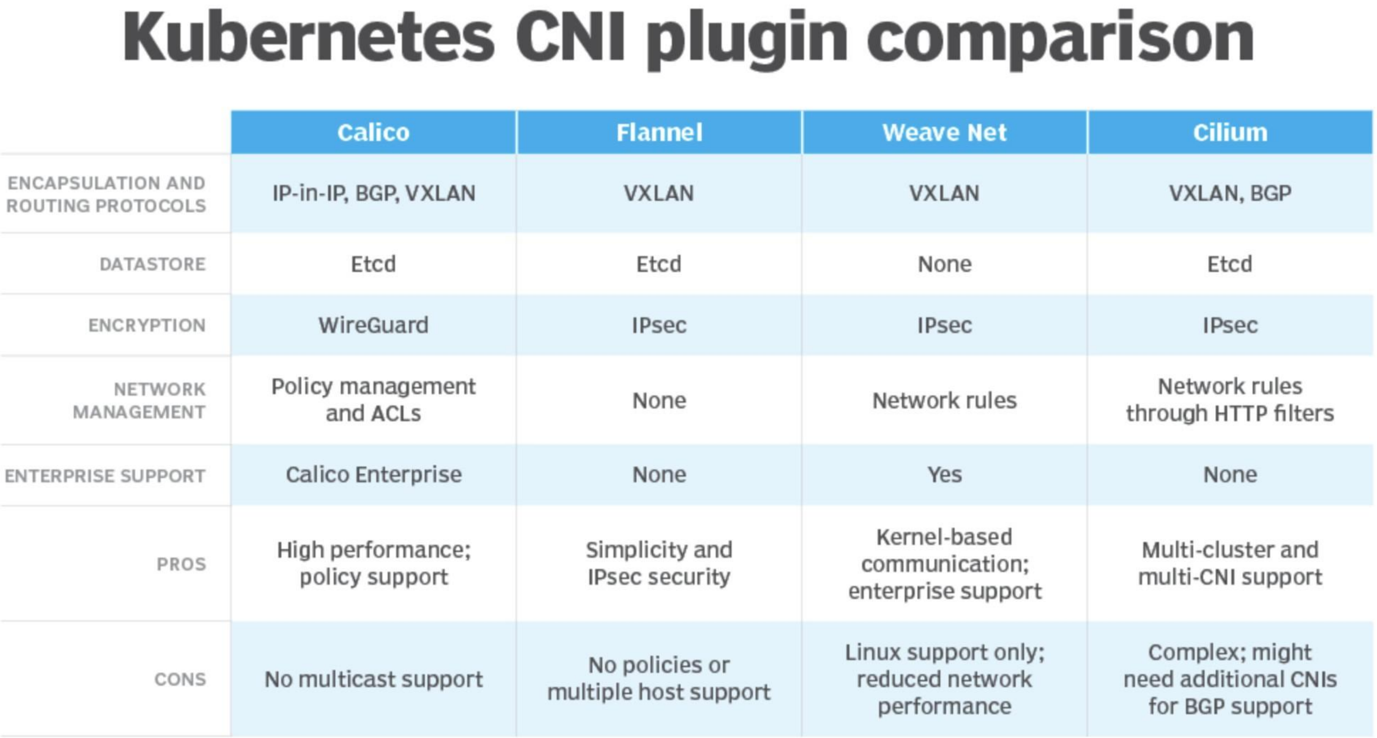
CNI

- CNI Plugins
  - Flannel
    - simple and easy way to configure a layer 3 network fabric
    - single binary agent called flanneld
    - runs on each node
    - responsible for allocating a subnet lease to each host out of a larger, preconfigured address space

  - Deploying Flannel with kubectl
    - kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
  - For custom podCIDR (not 10.244.0.0/16), download above to adjust

CNI

- CNI Plugins
  - Calico
    - Uses BGP
  - Weave
  - Cilium
  - Canal

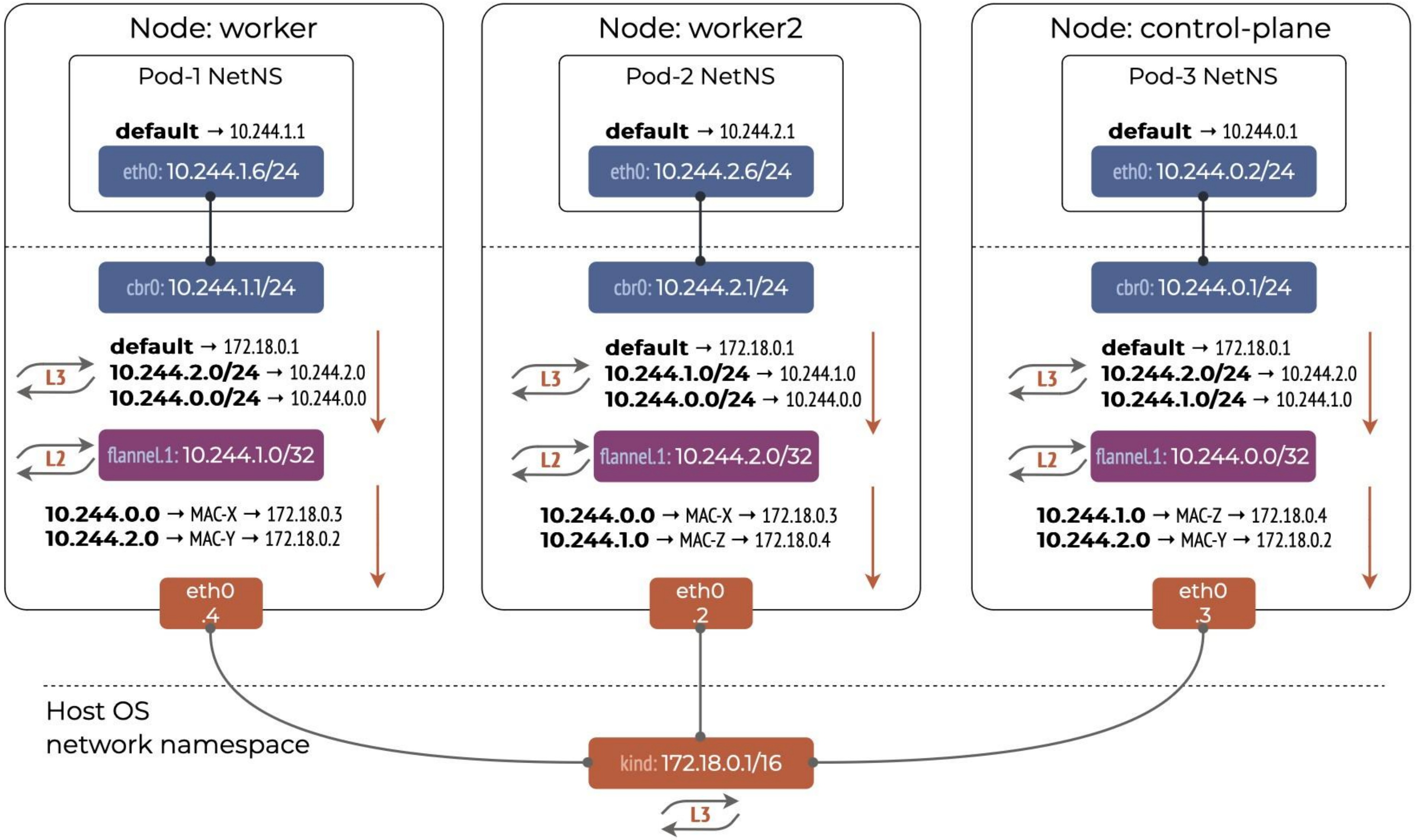


Kubernetes CNI plugin comparison

	Calico	Flannel	Weave Net	Cilium
ENCAPSULATION AND ROUTING PROTOCOLS	IP-in-IP, BGP, VXLAN	VXLAN	VXLAN	VXLAN, BGP
DATASTORE	Etcd	Etcd	None	Etcd
ENCRYPTION	WireGuard	IPsec	IPsec	IPsec
NETWORK MANAGEMENT	Policy management and ACLs	None	Network rules	Network rules through HTTP filters
ENTERPRISE SUPPORT	Calico Enterprise	None	Yes	None
PROS	High performance; policy support	Simplicity and IPsec security	Kernel-based communication; enterprise support	Multi-cluster and multi-CNI support
CONS	No multicast support	No policies or multiple host support	Linux support only; reduced network performance	Complex; might need additional CNIs for BGP support

CNI

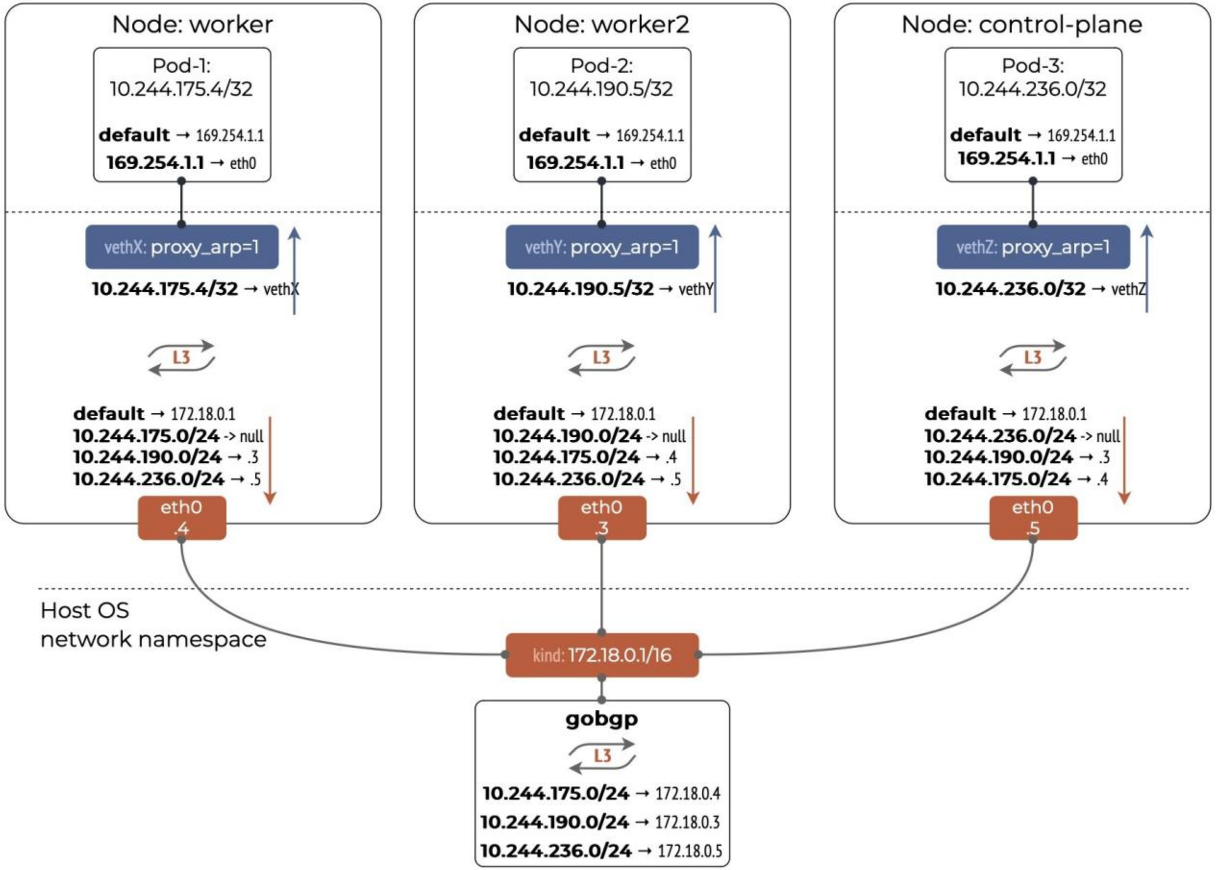
- Flannel



Node: worker
Pod-1 NetNS
default → 10.244.1.1
eth0: 10.244.1.6/24
cbr0: 10.244.1.1/24
L3
default → 172.18.0.1
10.244.2.0/24 → 10.244.2.0
10.244.0.0/24 → 10.244.0.0
L2
flannel.1: 10.244.1.0/32
10.244.0.0 → MAC-X → 172.18.0.3
10.244.2.0 → MAC-Y → 172.18.0.2
eth0 .4
Node: worker2
Pod-2 NetNS
default → 10.244.2.1
eth0: 10.244.2.6/24
cbr0: 10.244.2.1/24
L3
default → 172.18.0.1
10.244.1.0/24 → 10.244.1.0
10.244.0.0/24 → 10.244.0.0
L2
flannel.1: 10.244.2.0/32
10.244.0.0 → MAC-X → 172.18.0.3
10.244.1.0 → MAC-Z → 172.18.0.4
eth0 .2
Node: control-plane
Pod-3 NetNS
default → 10.244.0.1
eth0: 10.244.0.2/24
cbr0: 10.244.0.1/24
L3
default → 172.18.0.1
10.244.2.0/24 → 10.244.2.0
10.244.1.0/24 → 10.244.1.0
L2
flannel.1: 10.244.0.0/32
10.244.1.0 → MAC-Z → 172.18.0.4
10.244.2.0 → MAC-Y → 172.18.0.2
eth0 .3
Host OS
network namespace
kind: 172.18.0.1/16
L3

CNI

- Calico


Node: worker
Pod-1:
10.244.175.4/32
default → 169.254.1.1
169.254.1.1 → eth0
vethX: proxy_arp=1
10.244.175.4/32 → vethX
L3
default → 172.18.0.1
10.244.175.0/24 -> null
10.244.190.0/24 → .3
10.244.236.0/24 → .5
eth0
.4
Node: worker2
Pod-2:
10.244.190.5/32
default → 169.254.1.1
169.254.1.1 → eth0
vethY: proxy_arp=1
10.244.190.5/32 → vethY
L3
default → 172.18.0.1
10.244.190.0/24 -> null
10.244.175.0/24 → .4
10.244.236.0/24 → .5
eth0
.3
Node: control-plane
Pod-3:
10.244.236.0/32
default → 169.254.1.1
169.254.1.1 → eth0
vethZ: proxy_arp=1
10.244.236.0/32 → vethZ
L3
default → 172.18.0.1
10.244.236.0/24 -> null
10.244.190.0/24 → .3
10.244.175.0/24 → .4
eth0
.5
Host OS
network namespace
kind: 172.18.0.1/16
gobgp
L3
10.244.175.0/24 → 172.18.0.4
10.244.190.0/24 → 172.18.0.3
10.244.236.0/24 → 172.18.0.5