Kubernetes Networking Infra

Flannel, Calico & Weave CNI plugins

Arun Sriraman

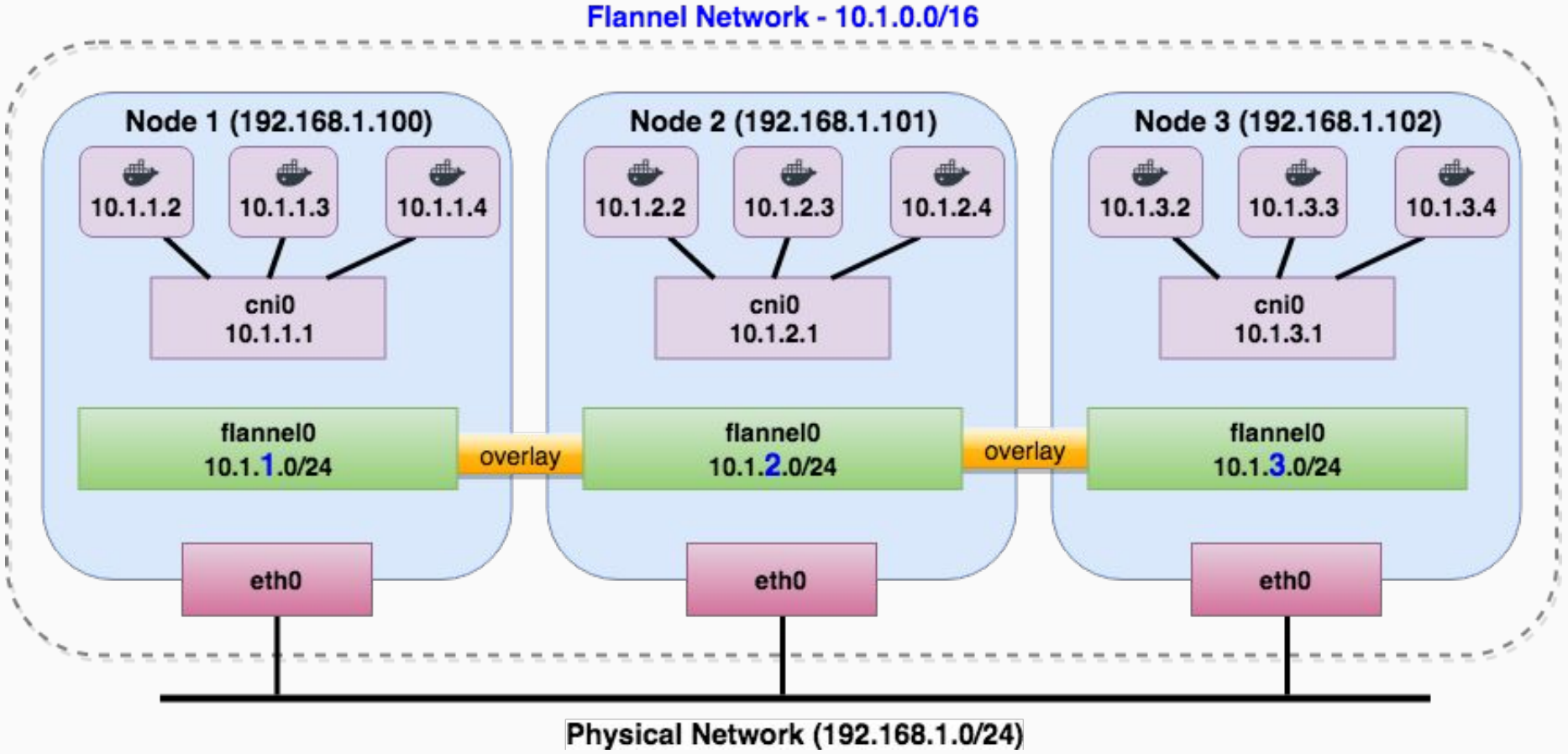
K8s network plugins - what do they do?

- Provide container network infrastructure
- Manage vif connections b/w Pod <-> Host
- May manage IPAM
- May provide mechanisms to connect Hosts together
- May provide NetworkPolicy enforcements
- Adhere to the CNI interface
- Eg: bridge, flannel, calico, weave, cilium, canal, ...

Flannel

- Provide Host <-> Host networking
- Uses overlay networks for connectivity:
  - VXLAN
  - UDP encap (UDP over IP)
- Experimental backends: IPIP, IPSec, AWS VPC, GCE
- Delegates Pod n/w on a host to the bridge driver
- “Was” default out-of-the-box choice for K8s n/w

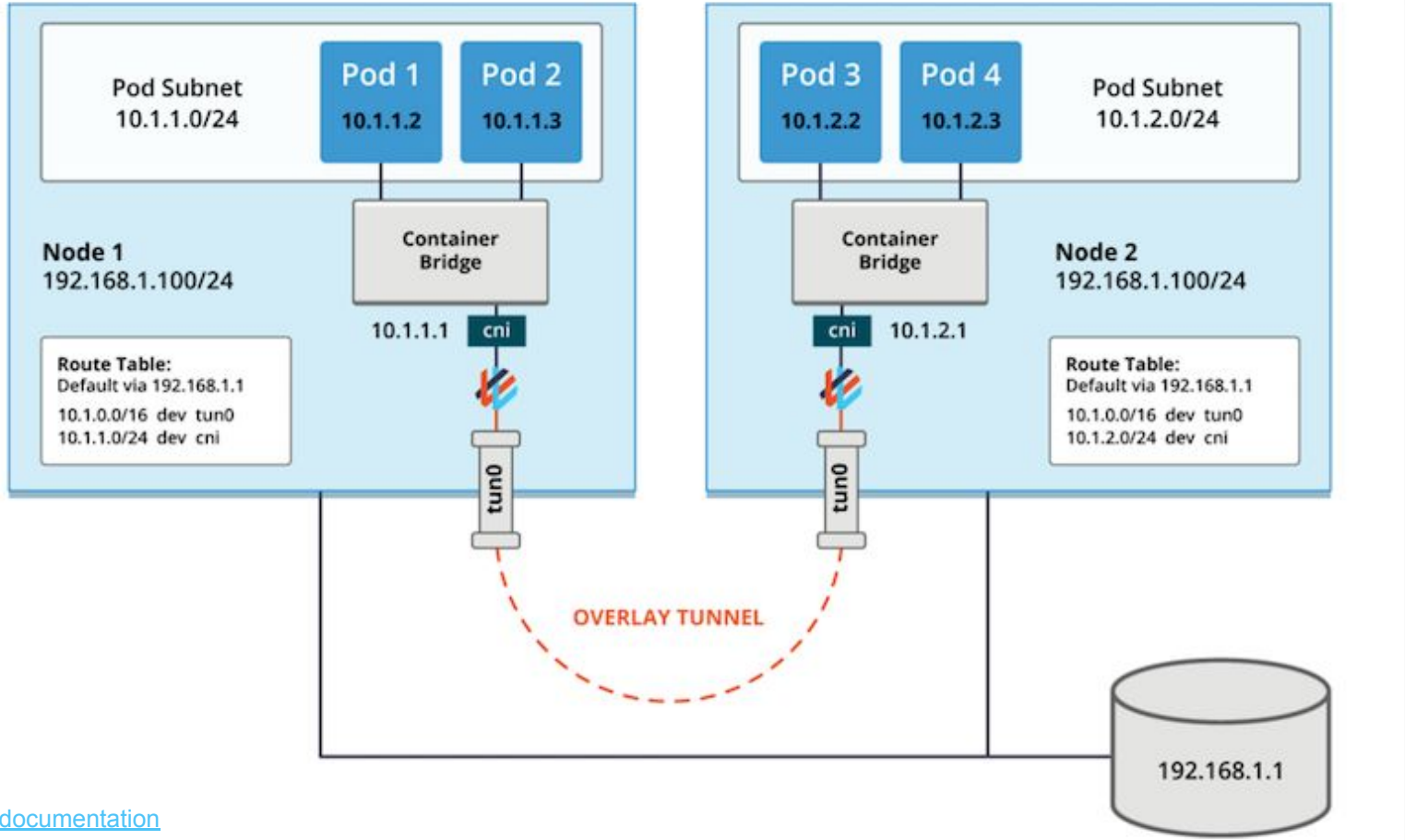
Flannel


Flannel Network - 10.1.0.0/16
Node 1 (192.168.1.100)
10.1.1.2
10.1.1.3
10.1.1.4
cni0
10.1.1.1
flannel0
10.1.1.0/24
eth0
overlay
Node 2 (192.168.1.101)
10.1.2.2
10.1.2.3
10.1.2.4
cni0
10.1.2.1
flannel0
10.1.2.0/24
eth0
overlay
Node 3 (192.168.1.102)
10.1.3.2
10.1.3.3
10.1.3.4
cni0
10.1.3.1
flannel0
10.1.3.0/24
eth0
Physical Network (192.168.1.0/24)

Weave

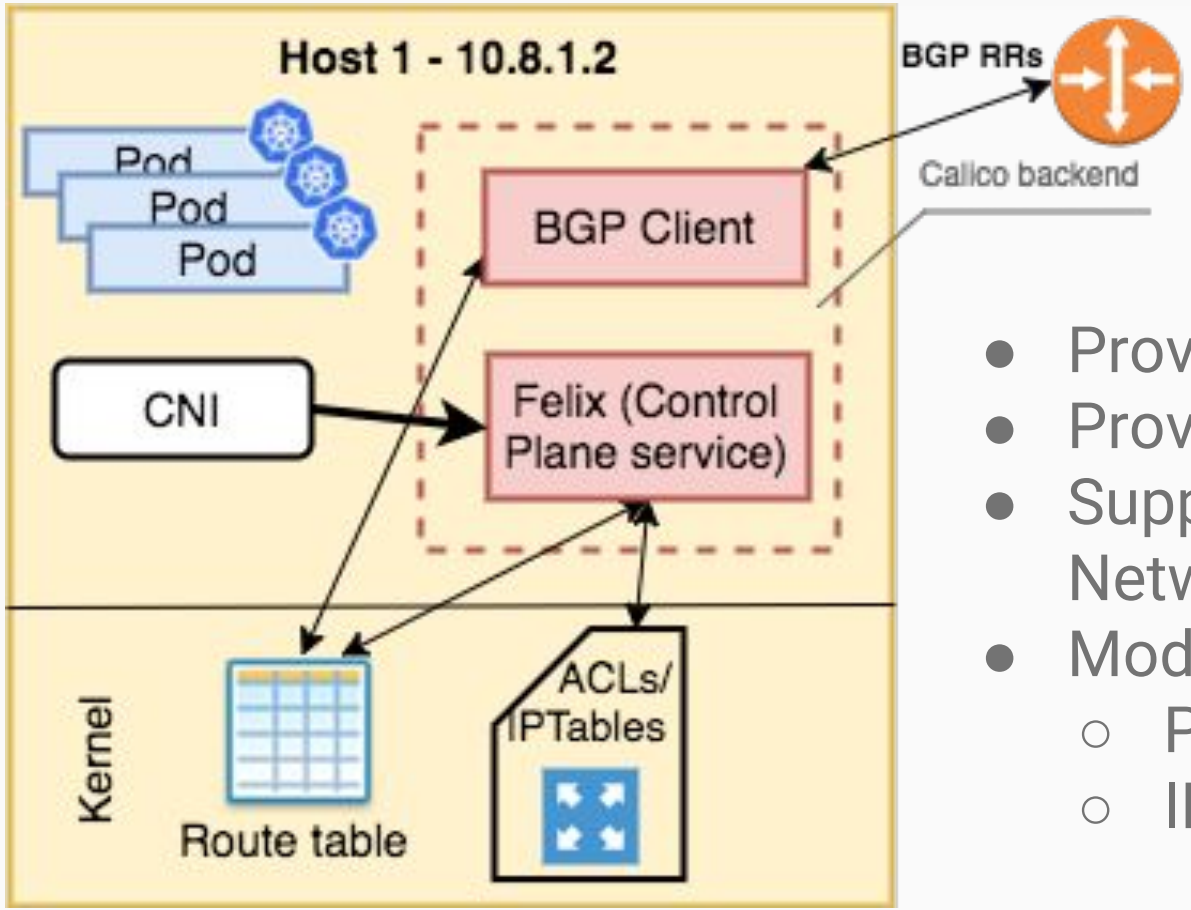
- Provide Host <-> Host networking & container networking. Uses weave router
- Provides IPAM and host networking
- Modes:
  - Sleeve (udp encap), uses tcp connections
  - **Fast-datapath (using openvswitch & VXLAN)**
- Provides Network Policy support
- Can be installed as a daemon set on K8s

Weave



Source: Weave documentation

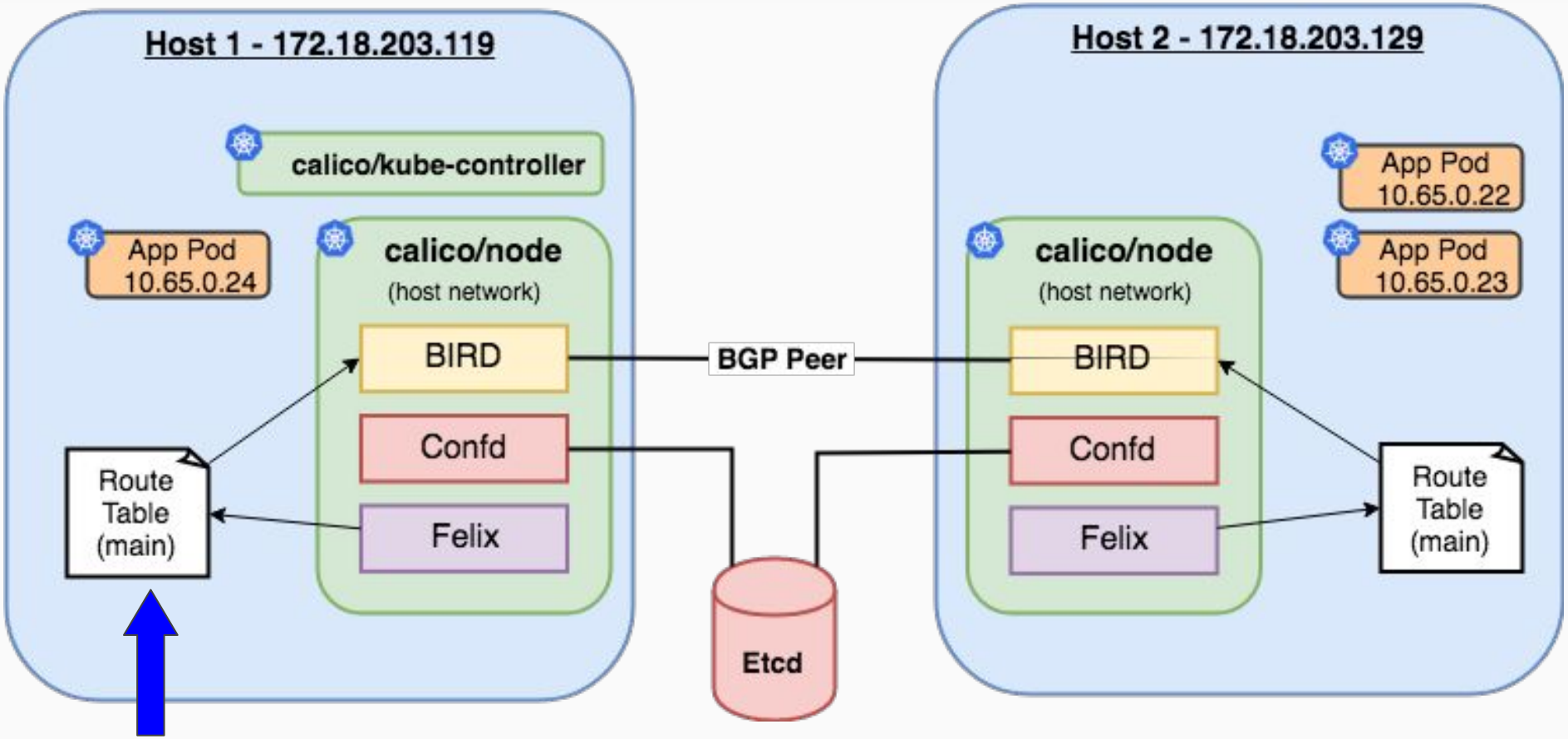
Calico



- Pure L3 based network solution
- Router per node

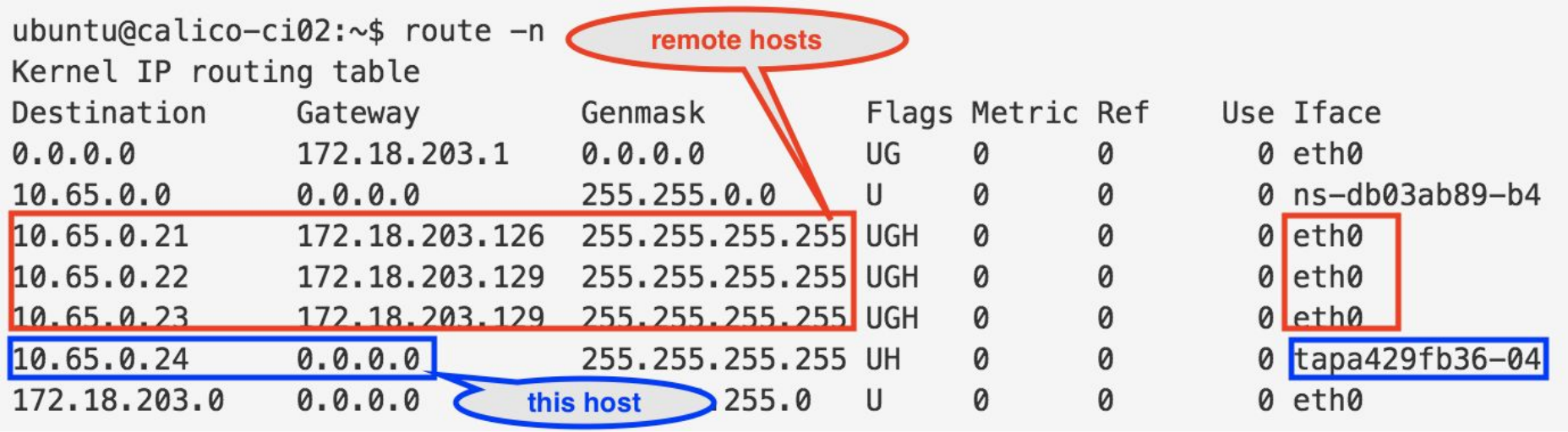
- Provides host-host n/w'ing
- Provides IPAM & pod n/w
- Supports Kubernetes NetworkPolicy constructs
- Modes:
  - Pure BGP
  - IP-IP encap

Calico

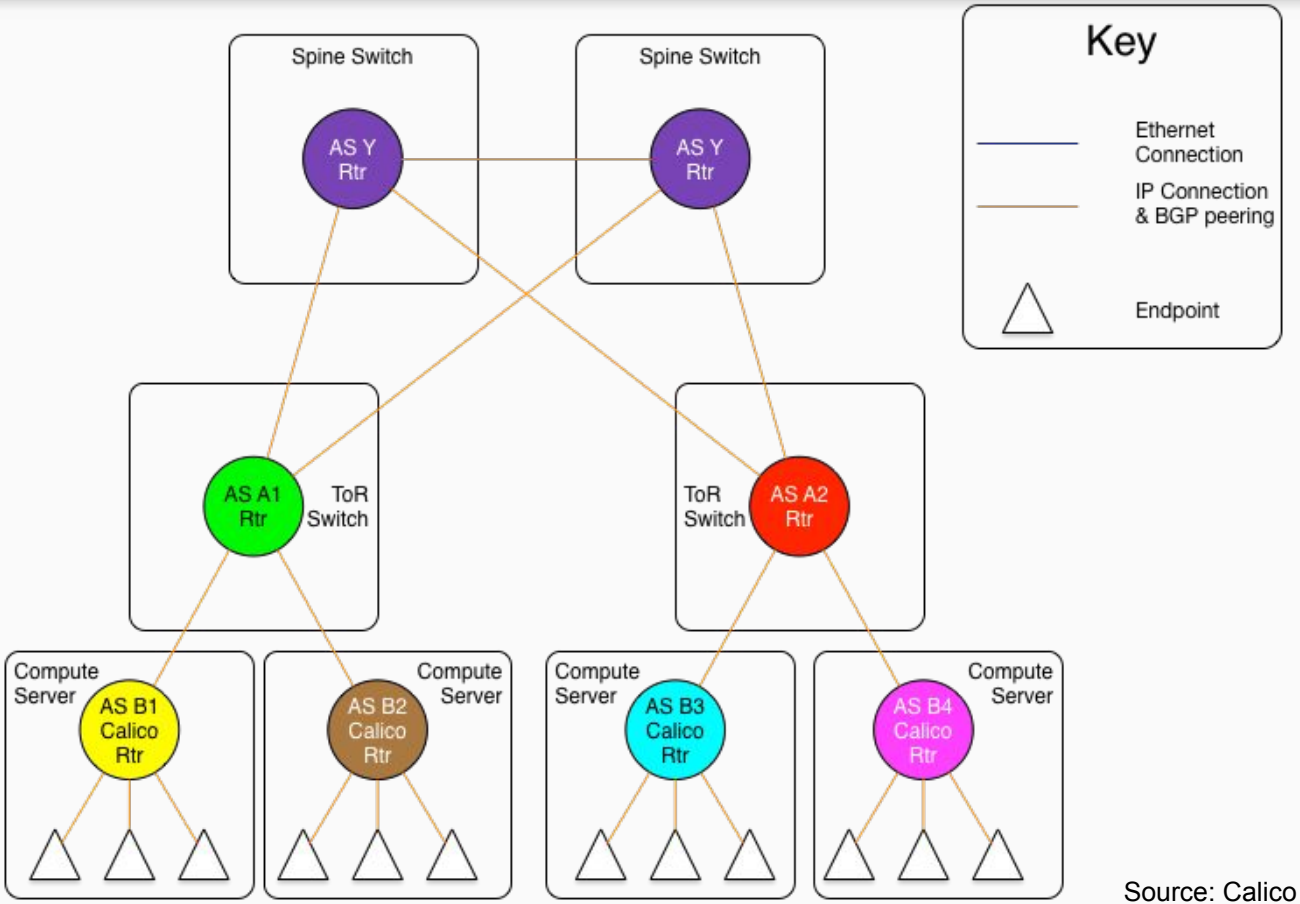

Host 1 - 172.18.203.119
calico/kube-controller
App Pod
10.65.0.24
calico/node
(host network)
BIRD
Confd
Felix
Route
Table
(main)
BGP Peer
Etcd
Host 2 - 172.18.203.129
App Pod
10.65.0.22
App Pod
10.65.0.23
calico/node
(host network)
BIRD
Confd
Felix
Route
Table
(main)

Calico

- calico/k8s-controller: policy, profile(ns), node, workloadendpoint (pod labels)
- calico/node: BIRD, confd, felix
- Each pod gets /32 route. #ofentries arbitrarily large!



Calico



Source: Calico Documentation

K8s NetworkPolicy

- Mechanism to provide “firewalling”/access security
- NetworkPolicies translate to L3-L7 network ACLs
- Works with src, destination filters (pod, ns, ip, …)
- “Generally” implemented as iptable rules
  - Counter example: Istio (sidecar model)
- Stable since k8s1.7 with supported n/w plugins
- Default - allow-all, this can however be changed

K8s NetworkPolicy

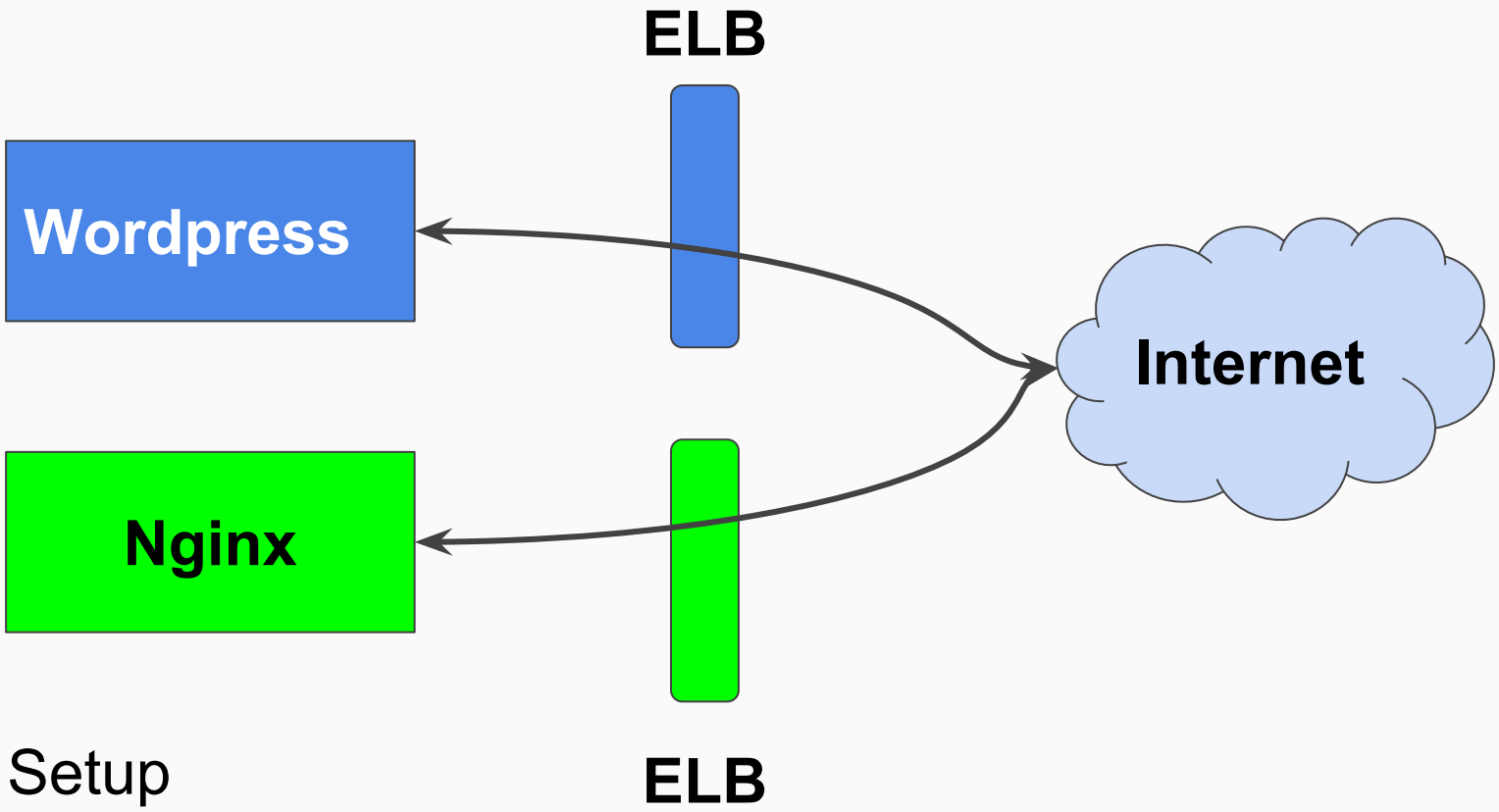
Default deny-all

```
apiVersion:
networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny
spec:
  podSelector: {}
  policyTypes:
  - Ingress
  - Egress
```

Default allow-all

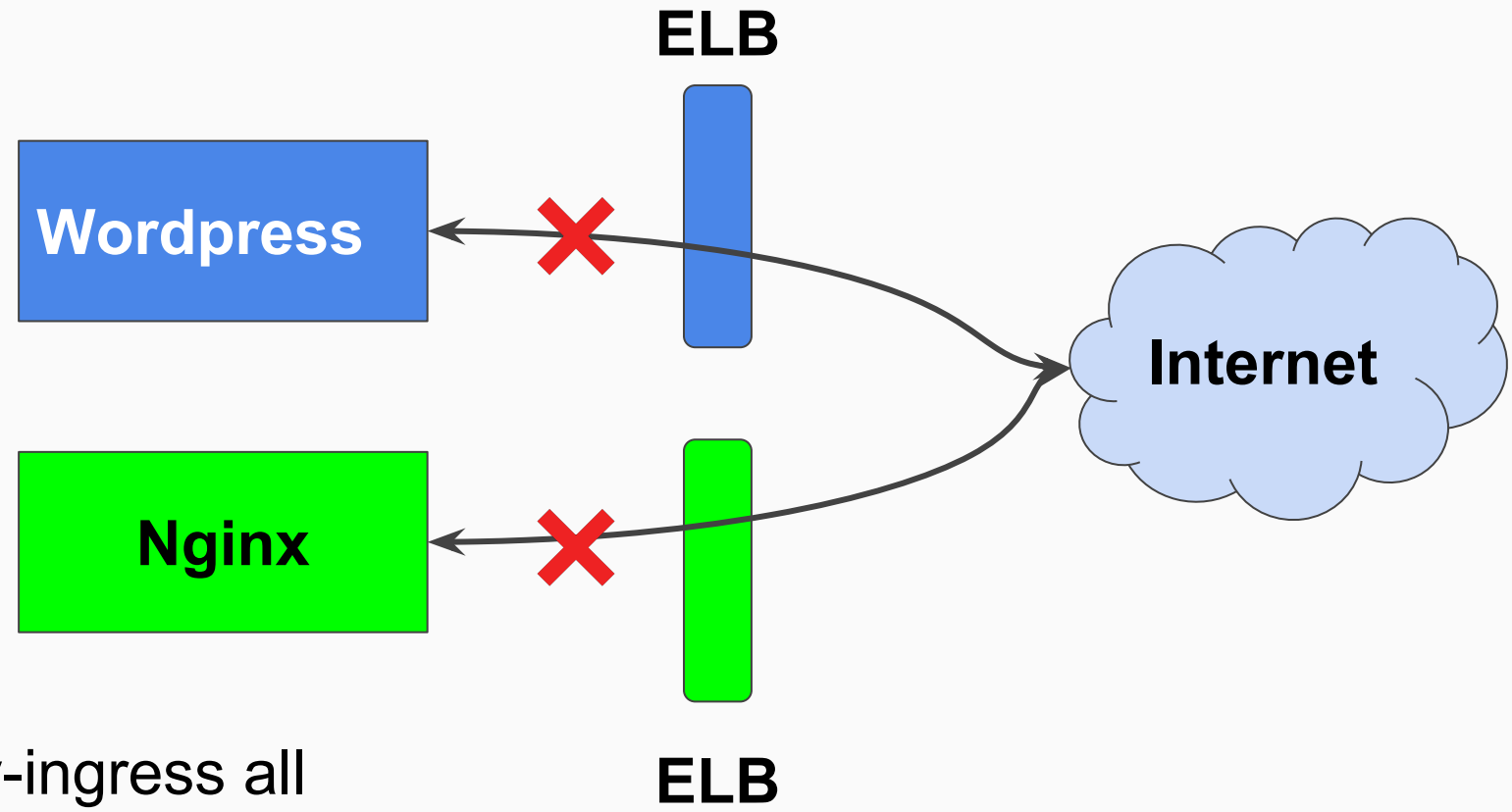
```
apiVersion:
networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-allow
spec:
  podSelector: {}
  egress:
  - {}
  Ingress:
  - {}
```

K8s NetworkPolicy - Demo



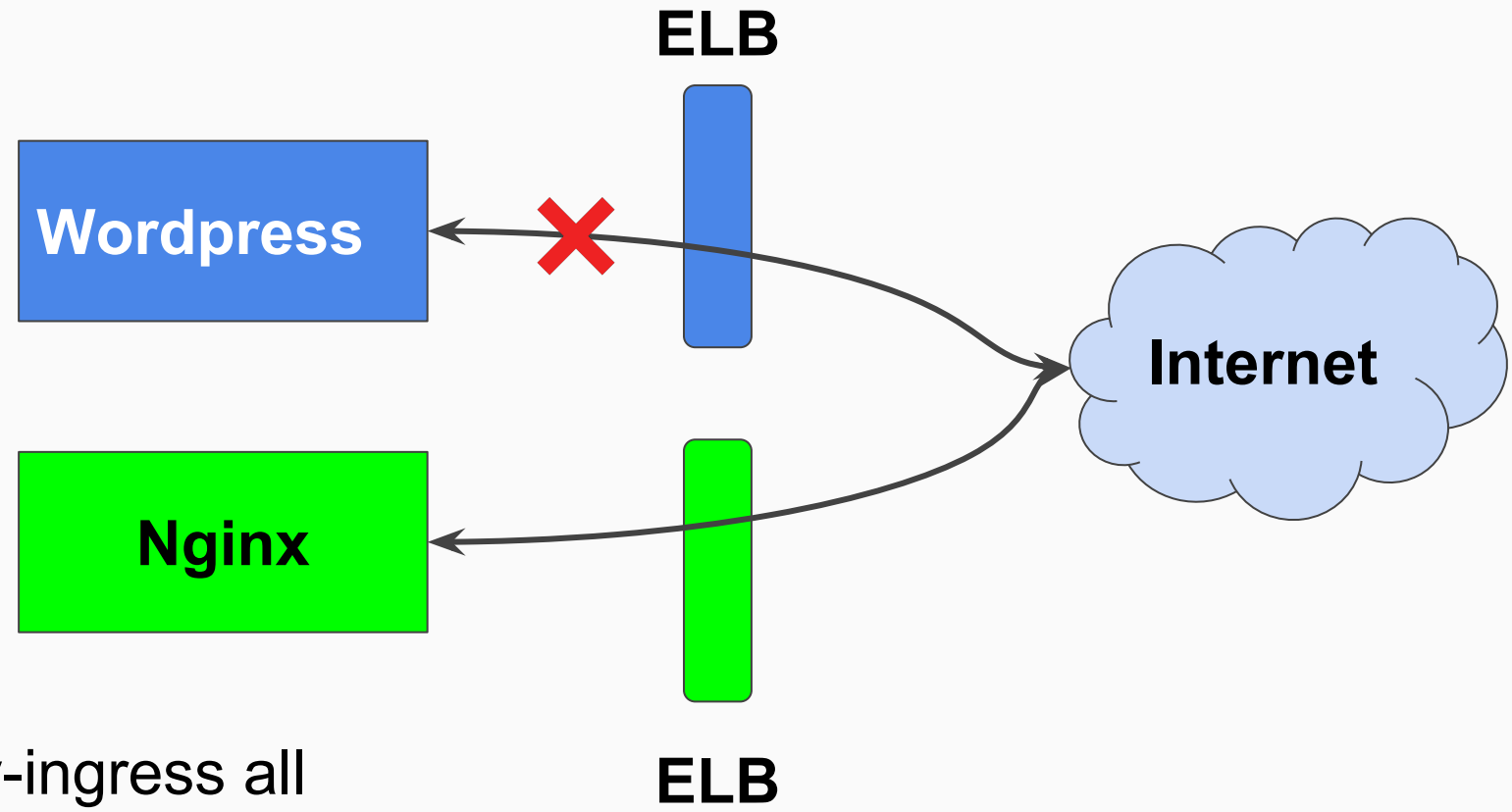
Initial Setup

K8s NetworkPolicy - Demo



Deny-ingress all

K8s NetworkPolicy - Demo



Deny-ingress all
Allow-nginx all

Questions?

NetworkPolicy resources