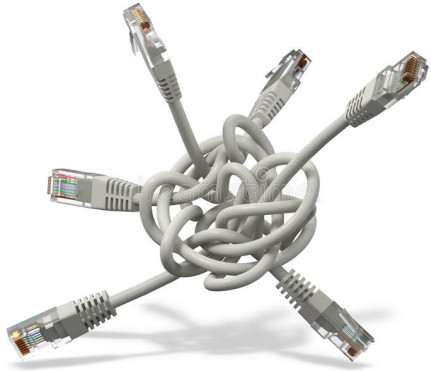
container networking from scratch



The aim

The network needs to satisfy the following (Kubenetes) requirements:

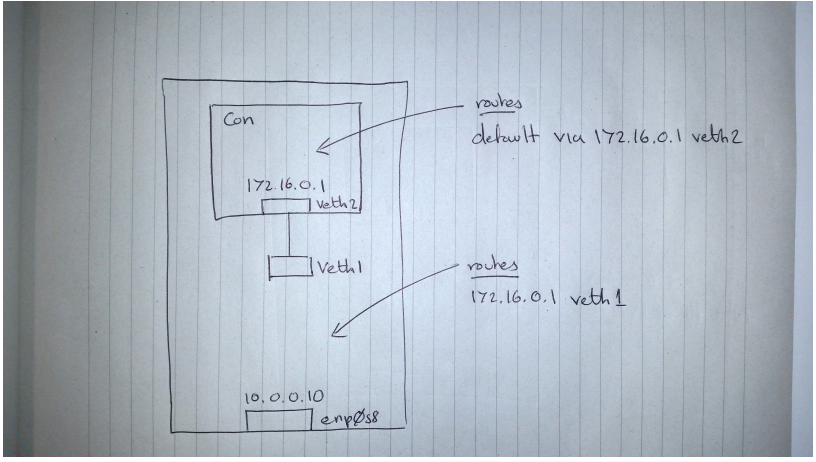
- All containers can communicate with all other containers without NAT
- All nodes can communicate with all containers (and vice-versa) without NAT
- The IP that a container sees itself as is the same IP that others see it as

The plan

To work our way from nothing, to a (flannel style) overlay network in 4 'easy' steps:

- Step 1: Single network namespace
- Step 2: Single node
- Step 3: Multi node
- Step 4: Overlay network

container networking from scratch

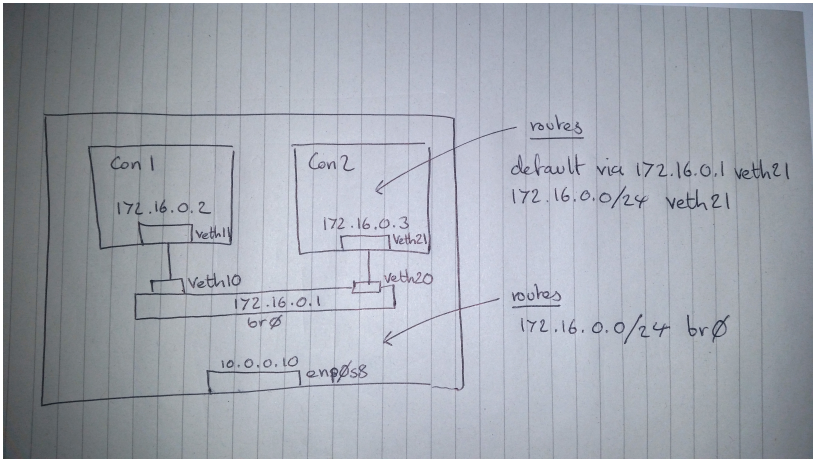


Routing rules 101

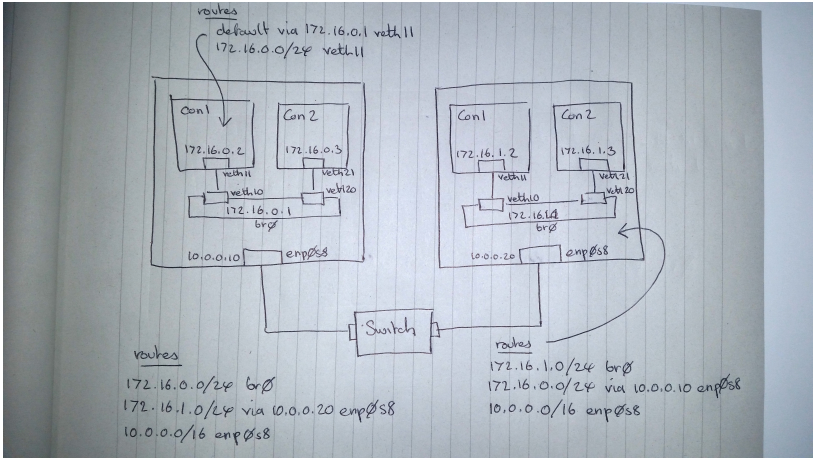
4 Types of routing rules (in order of preceedence):

1. Directly connected network, e.g. *10.0.0.0/24 eth1*
2. Static (manually added) routing rule, e.g. *10.0.0.0/24 via 10.0.0.1 eth0*
3. Dynamic (automatically added) routing rule, e.g. *10.0.0.0/24 via 10.0.0.1 eth0*
4. Default rule, e.g. *default via 10.0.0.1 eth0*

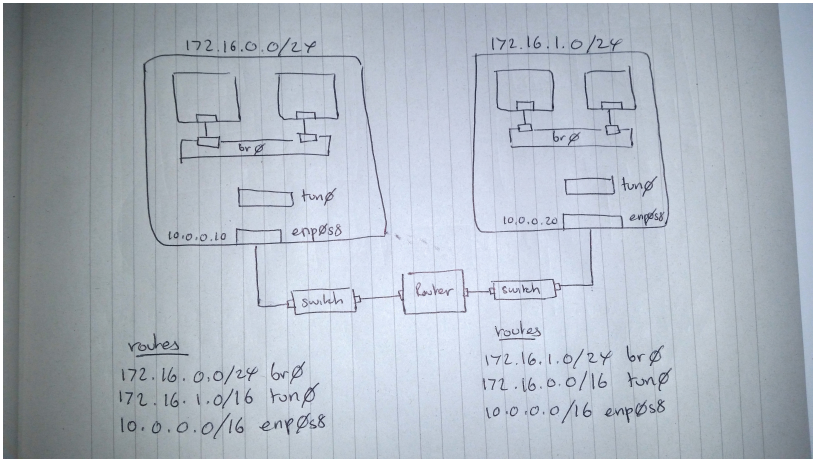
Within each of the above, the most specific CIDR range takes priority.


Con1
172.16.0.2
veth11
Con2
172.16.0.3
veth21
veth10
veth20
172.16.0.1
br0
10.0.0.10
enp0s8
routes
default via 172.16.0.1 veth21
172.16.0.0/24 veth21
routes
172.16.0.0/24 br0

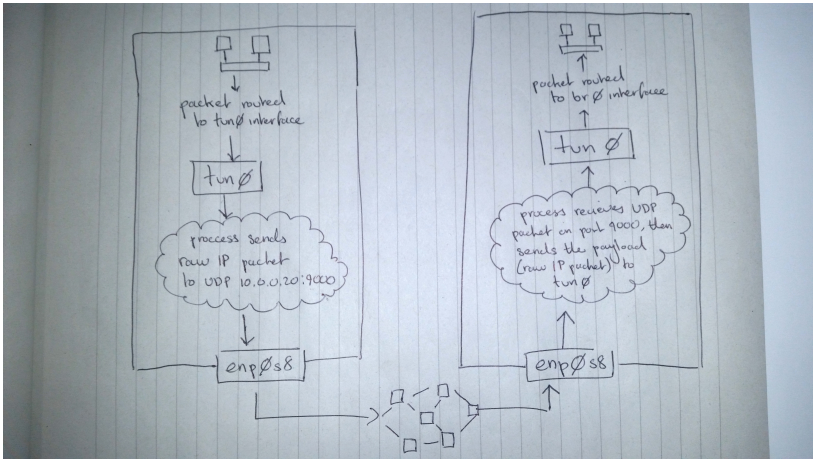
container networking from scratch



container networking from scratch



container networking from scratch



Putting it all together

1. ***Flannel***

    - *host-gw*: Step 3.
    - *udp*: Step 4.
    - *vxlan*: Step 4, but implemented in the kernel => more efficient!
    - *awsvpc*: Sets routes in AWS.
    - *gce*: Sets routes in GCE.
    - Node->pod-subnet mapping stored in *etcd*.

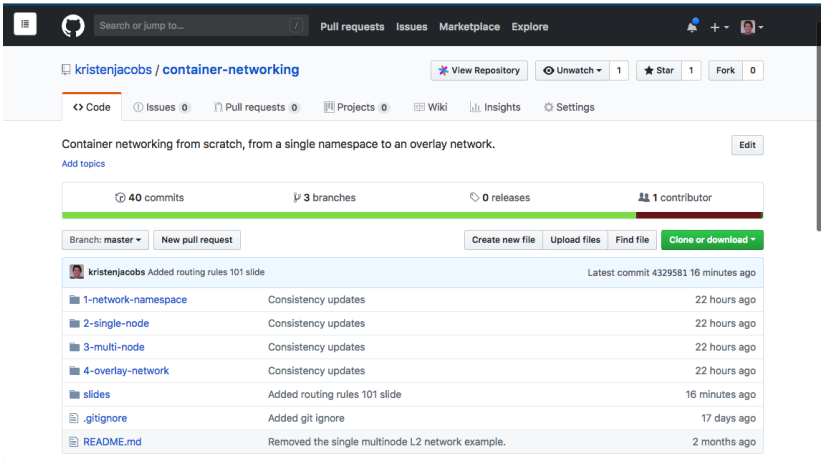
2. ***Calico***

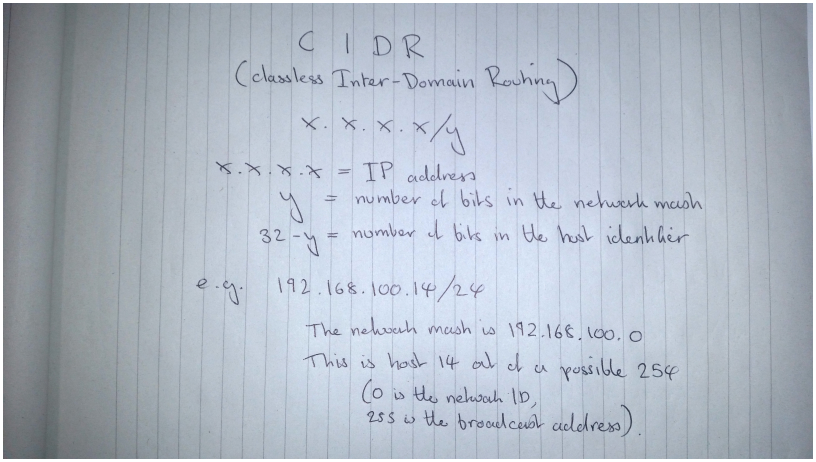
- No overlay for intra L2. Uses next-hop routing (step 3).
- For inter L2 node comminucation, uses IPIP overlay.
- Node->pod-subnet mappings distributed to nodes using BGP.

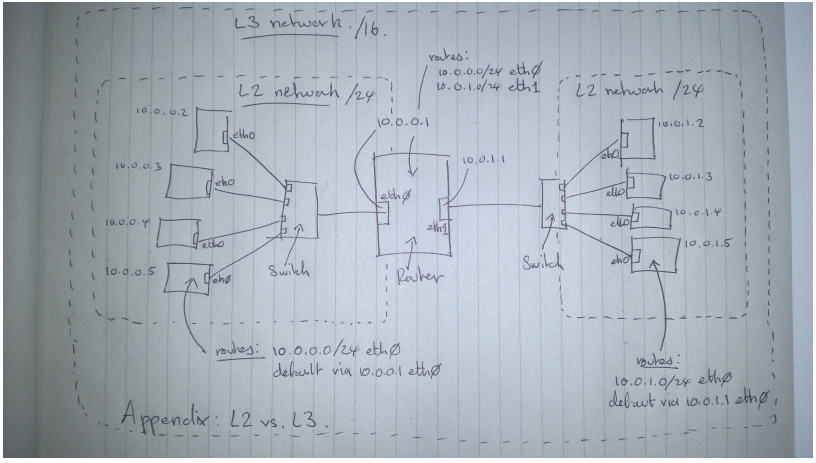
3. ***Weave***

- Similar to *Flannel*, i.e. uses *vxlan* overlay for connectivity.
- No need for *etcd*. Node->pod-subnet mapping distrubuted to each node peer to peer.

container networking from scratch




CIDR
(classless Inter-Domain Routing)
x.x.x.x/y
x.x.x.x = IP address
y = number of bits in the network mask
32-y = number of bits in the host identifier
e.g. 192.168.100.14/24
The network mask is 192.168.100.0
This is host 14 out of a possible 254
(0 is the network ID,
255 is the broadcast address).


L3 network /16.
routes:
10.0.0.0/24 eth0
10.0.1.0/24 eth1
L2 network /24
L2 network /24
10.0.0.2
eth0
10.0.0.3
eth0
10.0.0.4
eth0
10.0.0.5
eth0
Switch
10.0.0.1
eth0
eth1
10.0.1.1
Router
Switch
10.0.1.2
eth0
10.0.1.3
eth0
10.0.1.4
eth0
10.0.1.5
eth0
routes:
10.0.0.0/24 eth0
default via 10.0.0.1 eth0
routes:
10.0.1.0/24 eth0
default via 10.0.1.1 eth0
Appendix: L2 vs. L3.