Container Networking From Scratch

Kristen Jacobs

The Requirements

The network needs to satisfy the following (Kubernetes) 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

VOXXED DAYS BRISTOL

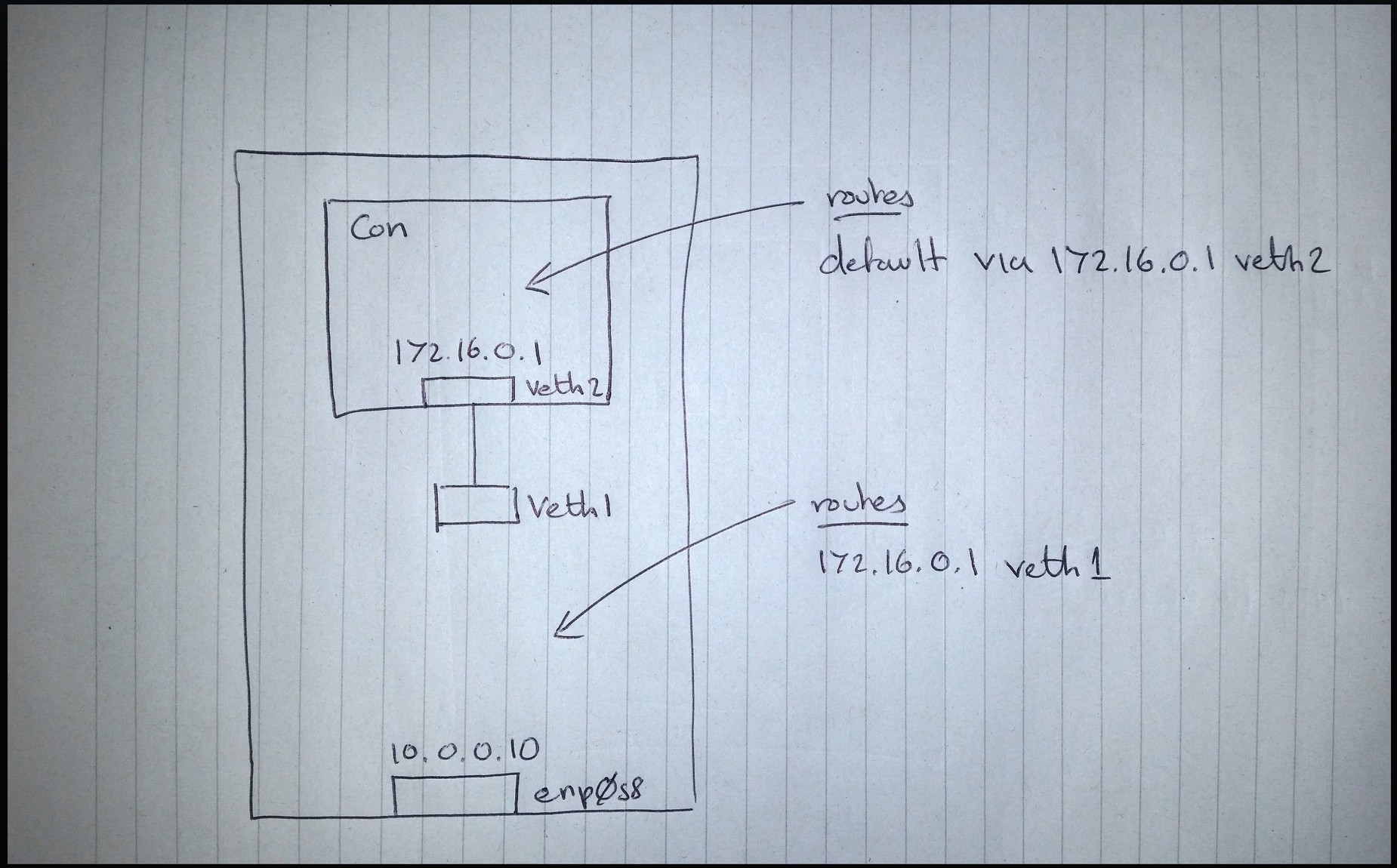
The Plan

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, 2 network namespaces.
- Step 3: Multiple nodes, same L2 network.
- Step 4: Multiple nodes, overlay network.

VOXXED DAYS BRISTOL

1. Single Network Namespace



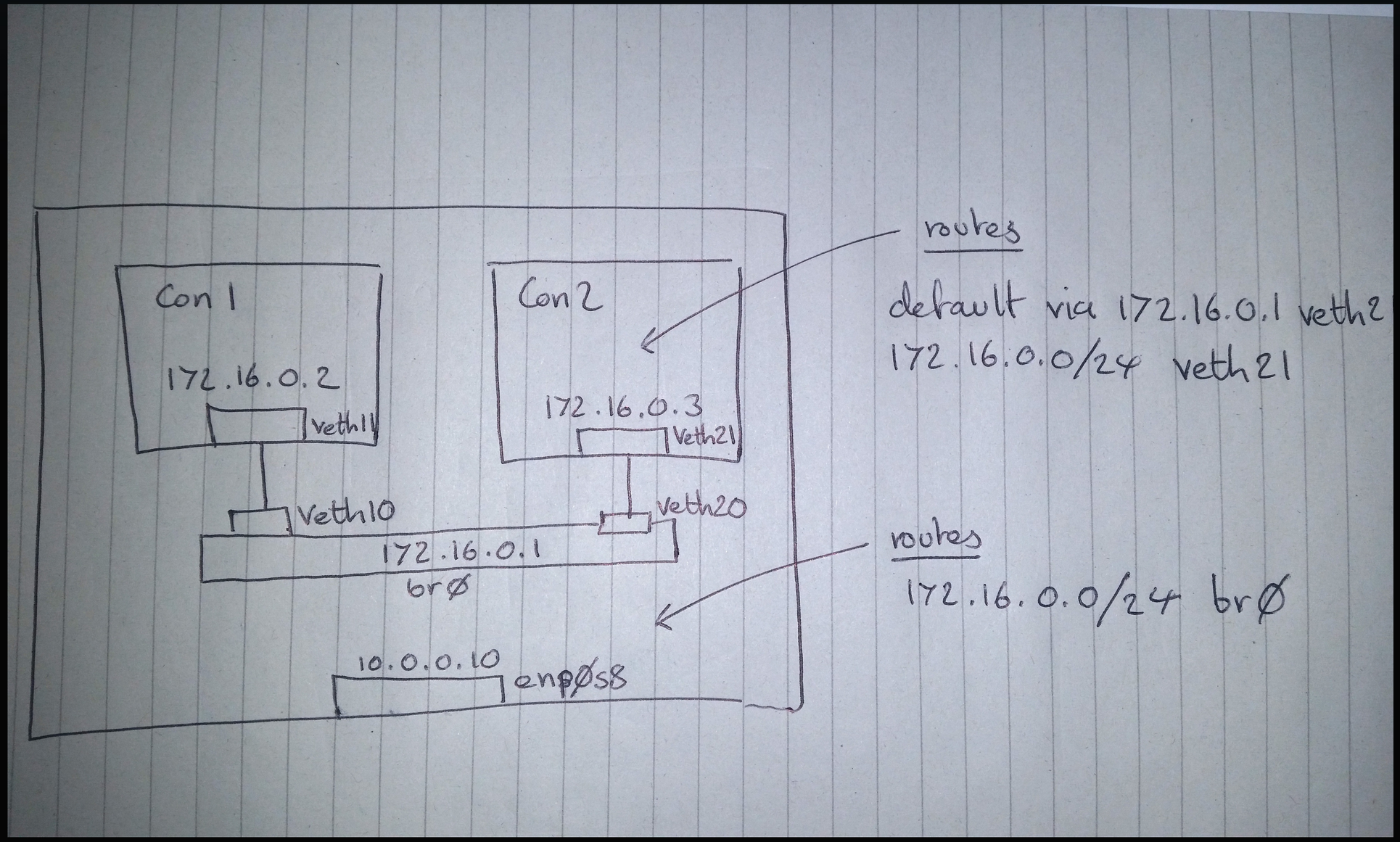
Routing Rules 101

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

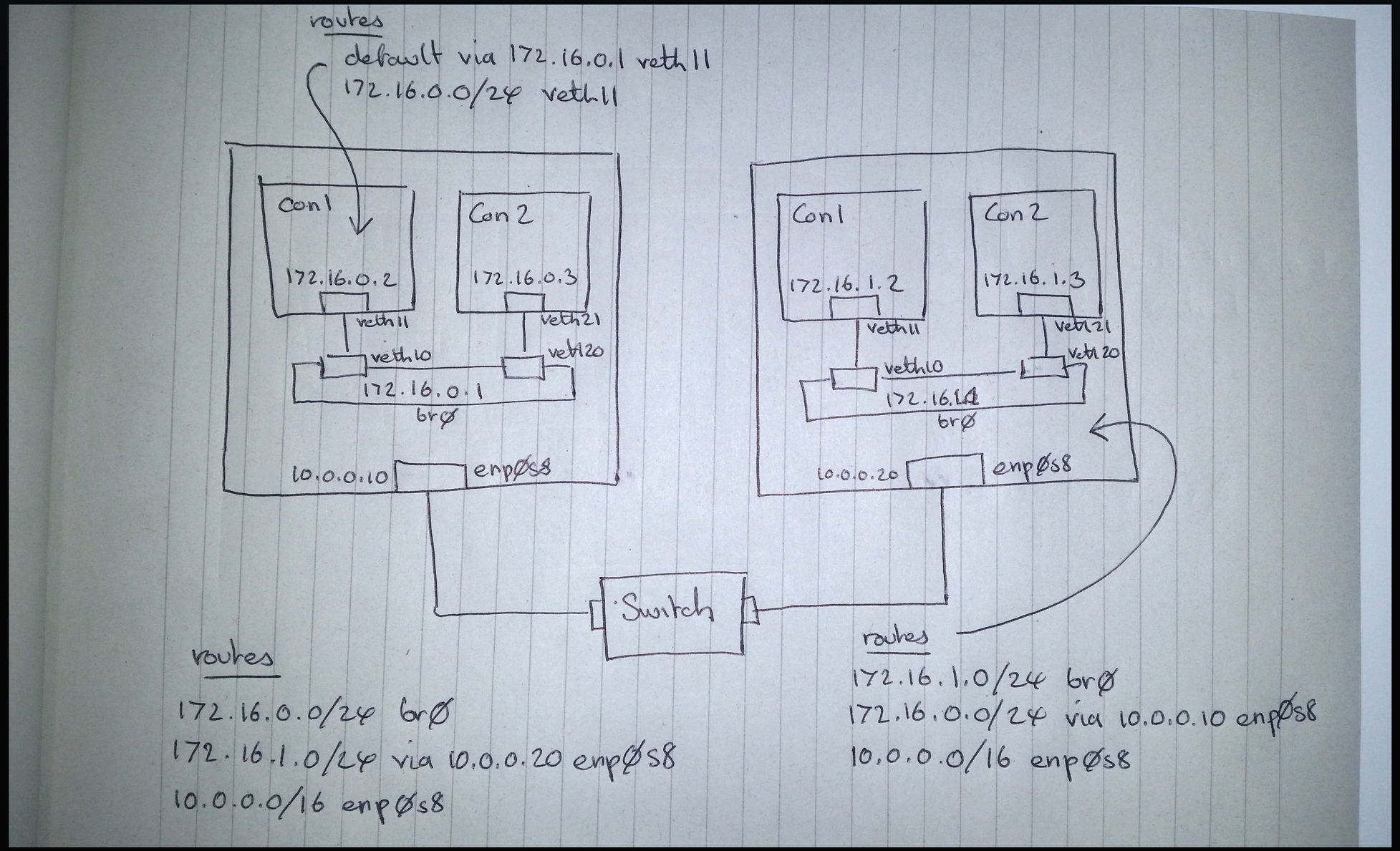
1. Directly connected network, e.g. *172.16.0.0/24 eth0*
2. Static (manually added) routing rule, e.g. *172.16.0.0/24 via 10.0.0.1 eth0*
3. Dynamic (automatically added) routing rule, e.g. *172.16.0.0/24 via 10.0.0.1 eth0*
4. Default rule, e.g. *default via 10.0.0.1 eth0*

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

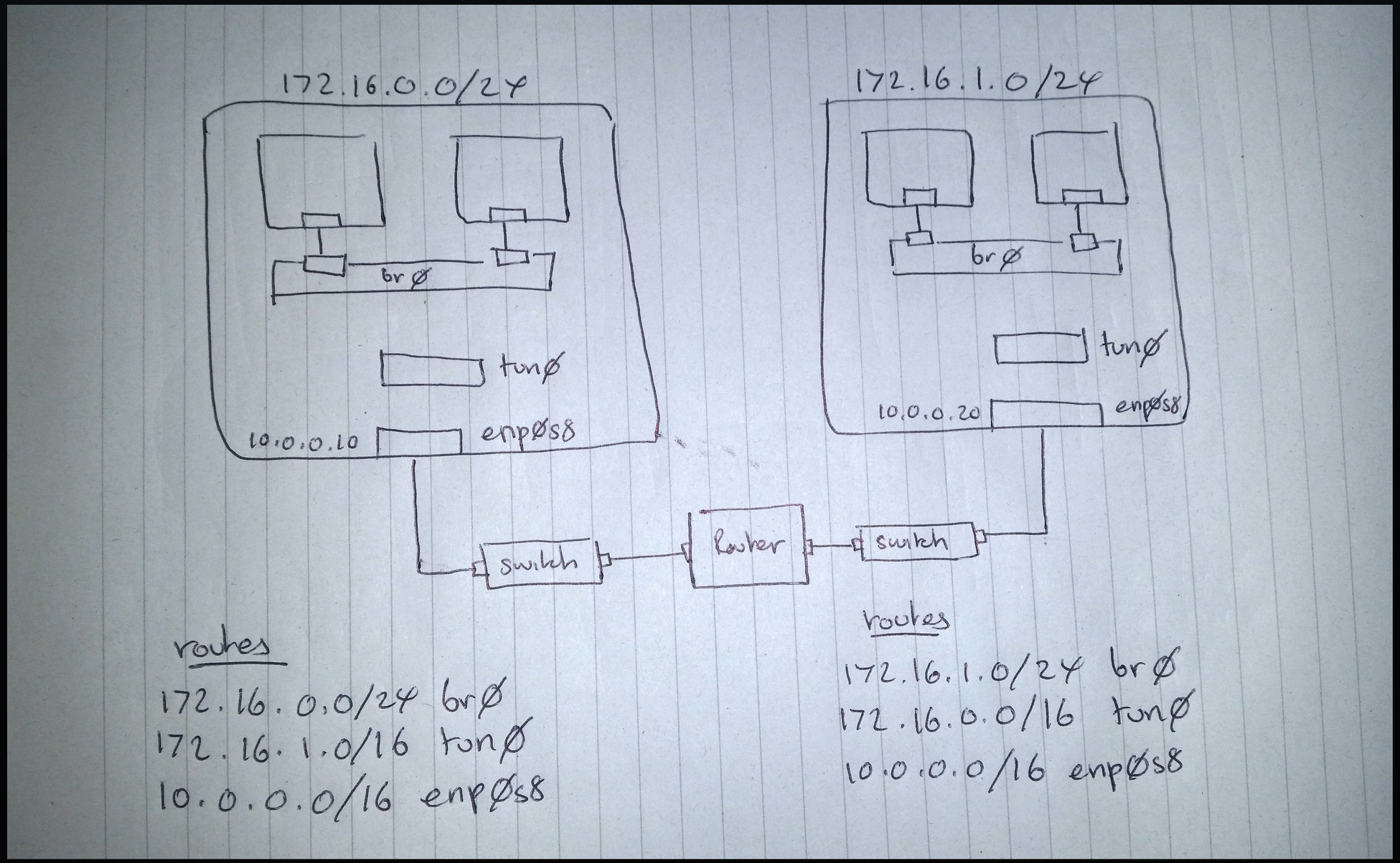
2. Single Node, 2 Namespaces

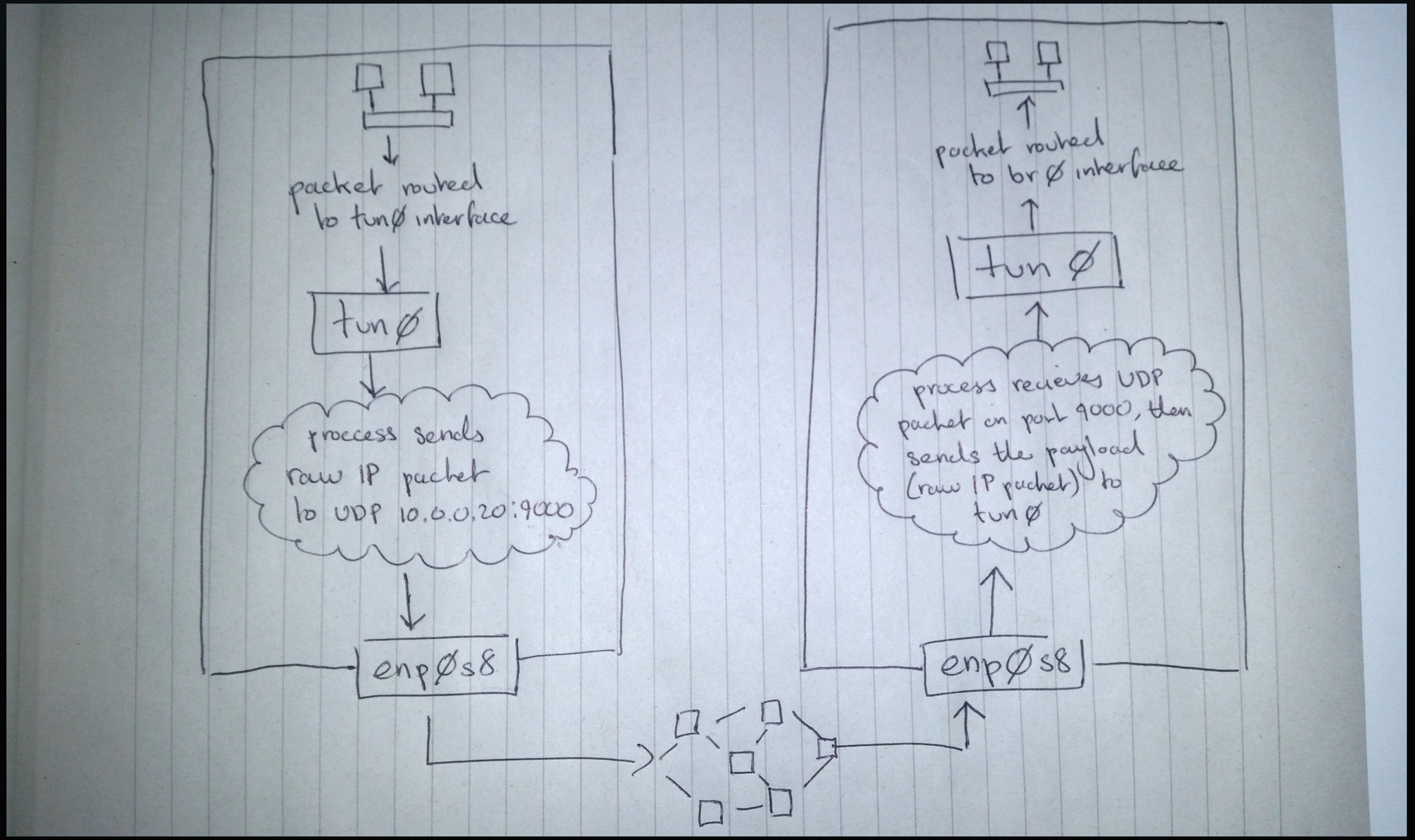


3. Multiple Nodes, Same L2 Network



4. Multiple Nodes, Overlay Network




packet routed to tun0 interface
tun0
process sends raw IP packet to UDP 10.0.0.20:9000
enp0s8
packet routed to br0 interface
tun0
process recieves UDP packet on port 9000, then sends the payload (raw IP packet) to tun0
enp0s8


VOXXEDDAYS
BRISTOL

Recap

- Step 1: Single network namespace => Veth pair connecting namespace to the node.
- Step 2: Single node, 2 network namespaces => Veth pairs + Linux bridge.
- Step 3: Multiple nodes, same L2 network => Routing rules on each node.
- Step 4: Multiple nodes, different L2 networks => Overlay network.

Key Takeaways (concepts)

- Routing rules => The key to understanding networks
- Tun/Tap devices => The key to understanding overlay networks

- Key Takeaways (tools)
  - ip => One-stop-shop for all networking operations
  - socat => Can connect anything to anything!

Putting it all together

1. ***Flannel***

Has multiple backends:

- *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 communication, uses IPIP overlay.
- Node->pod-subnet mappings distributed to nodes using BGP.

3. ***Weave***

- Similar to Flannel, i.e. Step 4. Uses vxlan overlay for connectivity.
- No need for etcd. Node->pod-subnet mapping distributed to each node peer to peer.

Search or jump to... / Pull requests Issues Marketplace Explore

kristenjacobs / container-networking

View Repository | Unwatch 1 | Star 1 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Container networking from scratch, from a single namespace to an overlay network. Edit

Add topics

40 commits | 3 branches | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

kristenjacobs Added routing rules 101 slide — Latest commit 4329581 16 minutes ago

1-network-namespace	Consistency updates	22 hours ago
2-single-node	Consistency updates	22 hours ago
3-multi-node	Consistency updates	22 hours ago
4-overlay-network	Consistency updates	22 hours ago
slides	Added routing rules 101 slide	16 minutes ago
.gitignore	Added git ignore	17 days ago
README.md	Removed the single multinode L2 network example.	2 months ago

VOXXED DAYS BRISTOL

Questions?

VOXXED DAYS BRISTOL