Kubernetes Networks

How kubernetes networks work

David Chang

Linkernetworks
BackEnd, DevOps,
Docker, Kubernetes

dchang@linkernetworks.com

Outline

- Docker containers networks
- Containers communication in a Pod
- Pods cross different nodes
- Pod to Service

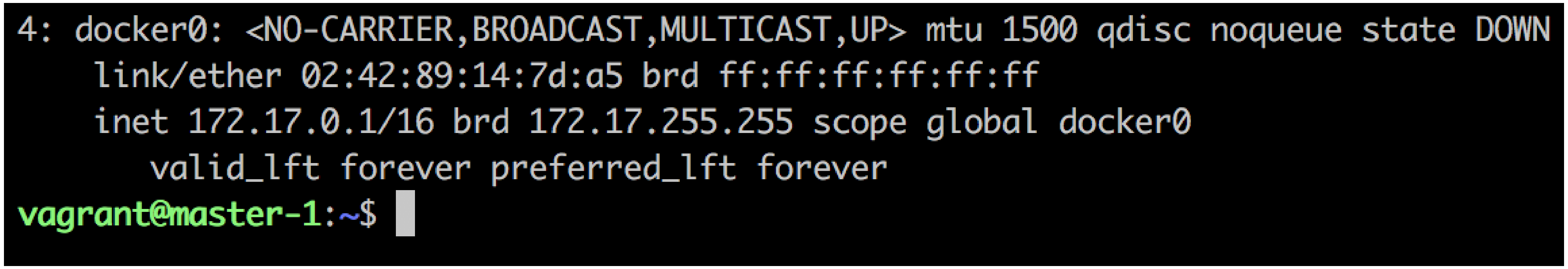
https://kubernetes.io/docs/concepts/cluster-administration/networking/

Docker Container Networks

- Bridge networks
  communicate namespaces through bridge
- Host networks
  use host’s port, ip…
- Container networks

https://docs.docker.com/network/

Docker Bridge Networks

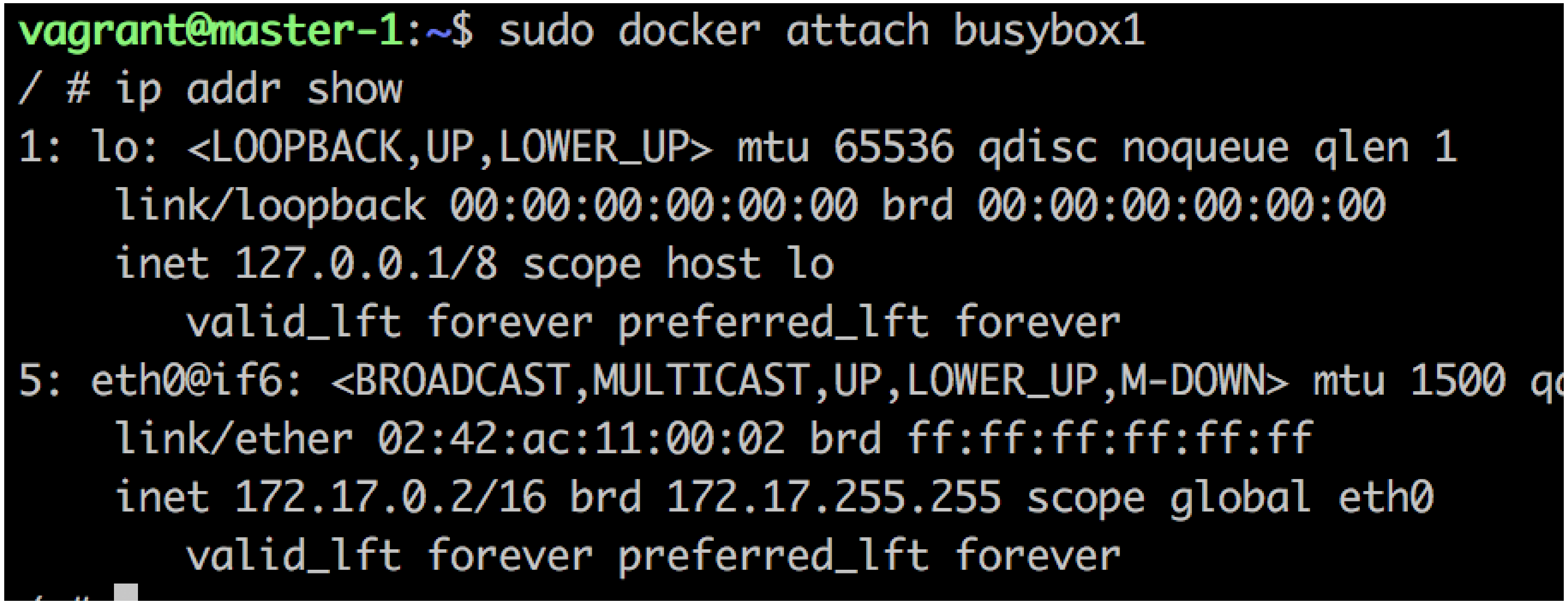


```
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:89:14:7d:a5 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
vagrant@master-1:~$
```

Docker daemon added a bridge, docker0 on host

https://docs.docker.com/v17.09/engine/userguide/networking/#default-networks

Docker Bridge Networks



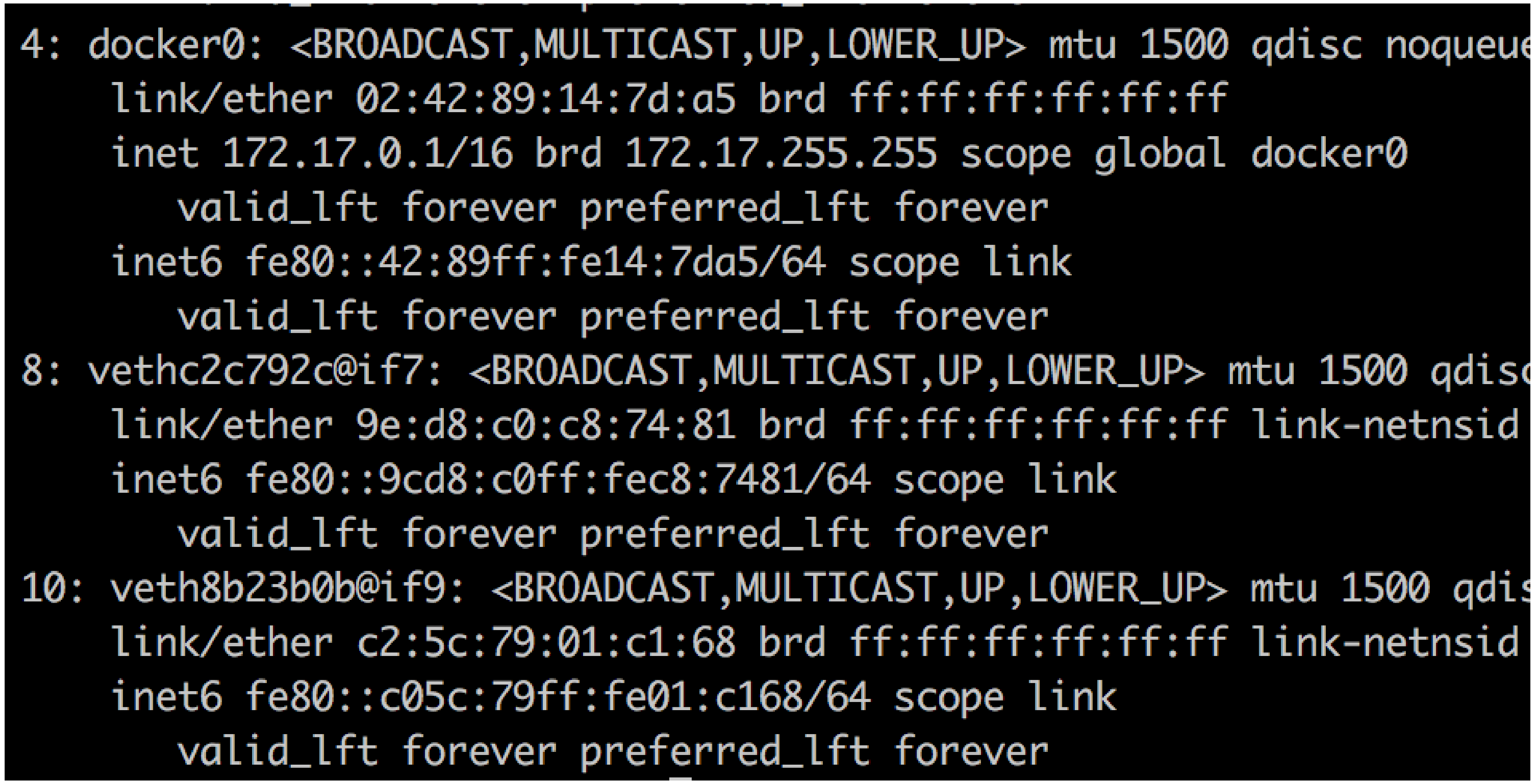
```
vagrant@master-1:~$ sudo docker attach busybox1
/ # ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qd
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

$ docker run -itd —name busybox1 busy box

A eth0 inside container

https://docs.docker.com/v17.09/engine/userguide/networking/#default-networks

Docker Bridge Networks



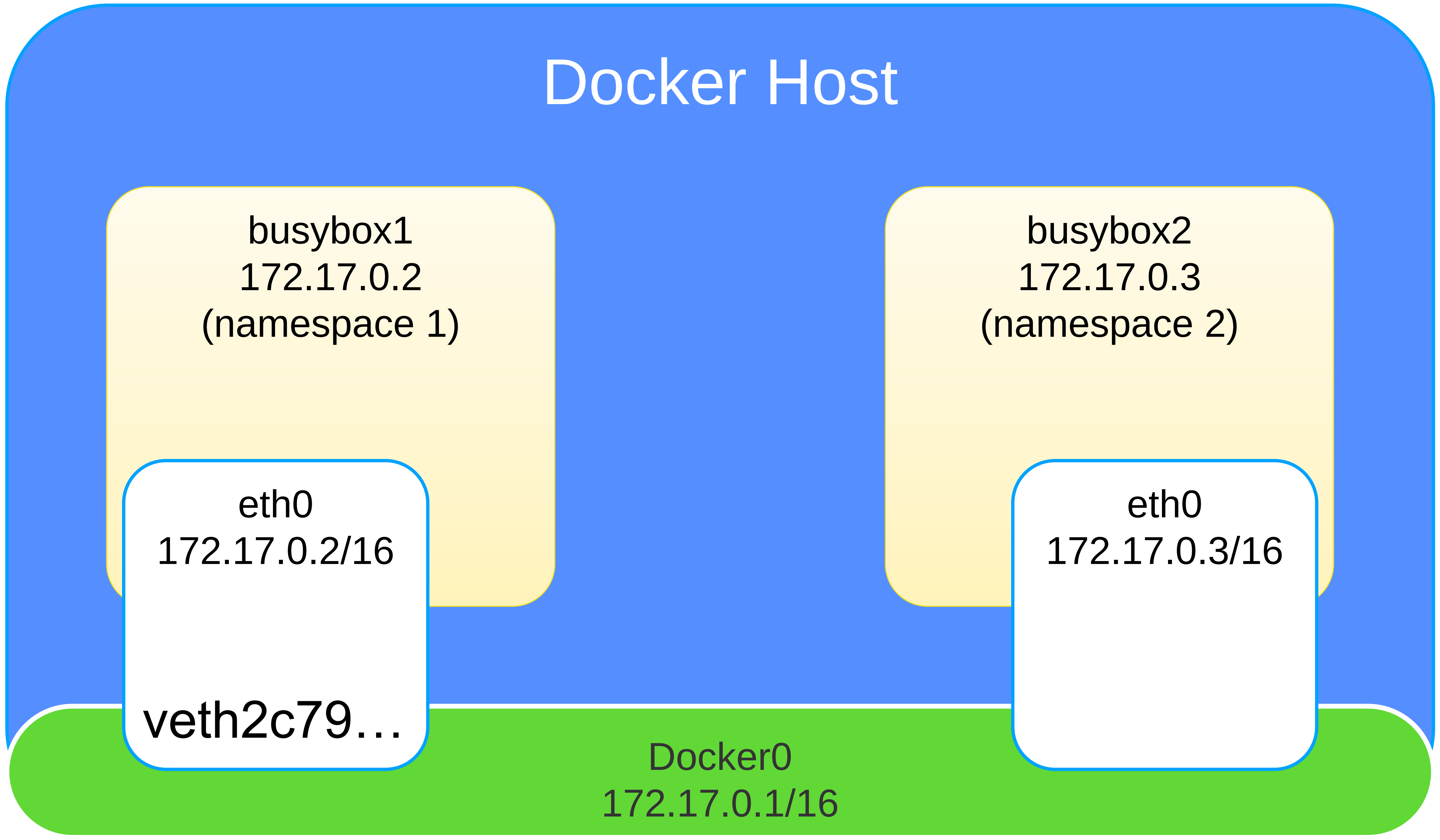
```
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    link/ether 02:42:89:14:7d:a5 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:89ff:fe14:7da5/64 scope link
       valid_lft forever preferred_lft forever
8: vethc2c792c@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    link/ether 9e:d8:c0:c8:74:81 brd ff:ff:ff:ff:ff:ff link-netnsid
    inet6 fe80::9cd8:c0ff:fec8:7481/64 scope link
       valid_lft forever preferred_lft forever
10: veth8b23b0b@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdis
    link/ether c2:5c:79:01:c1:68 brd ff:ff:ff:ff:ff:ff link-netnsid
    inet6 fe80::c05c:79ff:fe01:c168/64 scope link
       valid_lft forever preferred_lft forever
```

eth0@busybox1 - veth2c792c@if7 - docker0

Assign an ip from docker0 to eth@busybox1

https://docs.docker.com/v17.09/engine/userguide/networking/#default-networks

Docker Bridge Networks



Container has unique ip on a single docker host

Kubernetes Model

- Docker container networks
- Containers communication in a Pod
- Pods cross different nodes
- Pod to Service

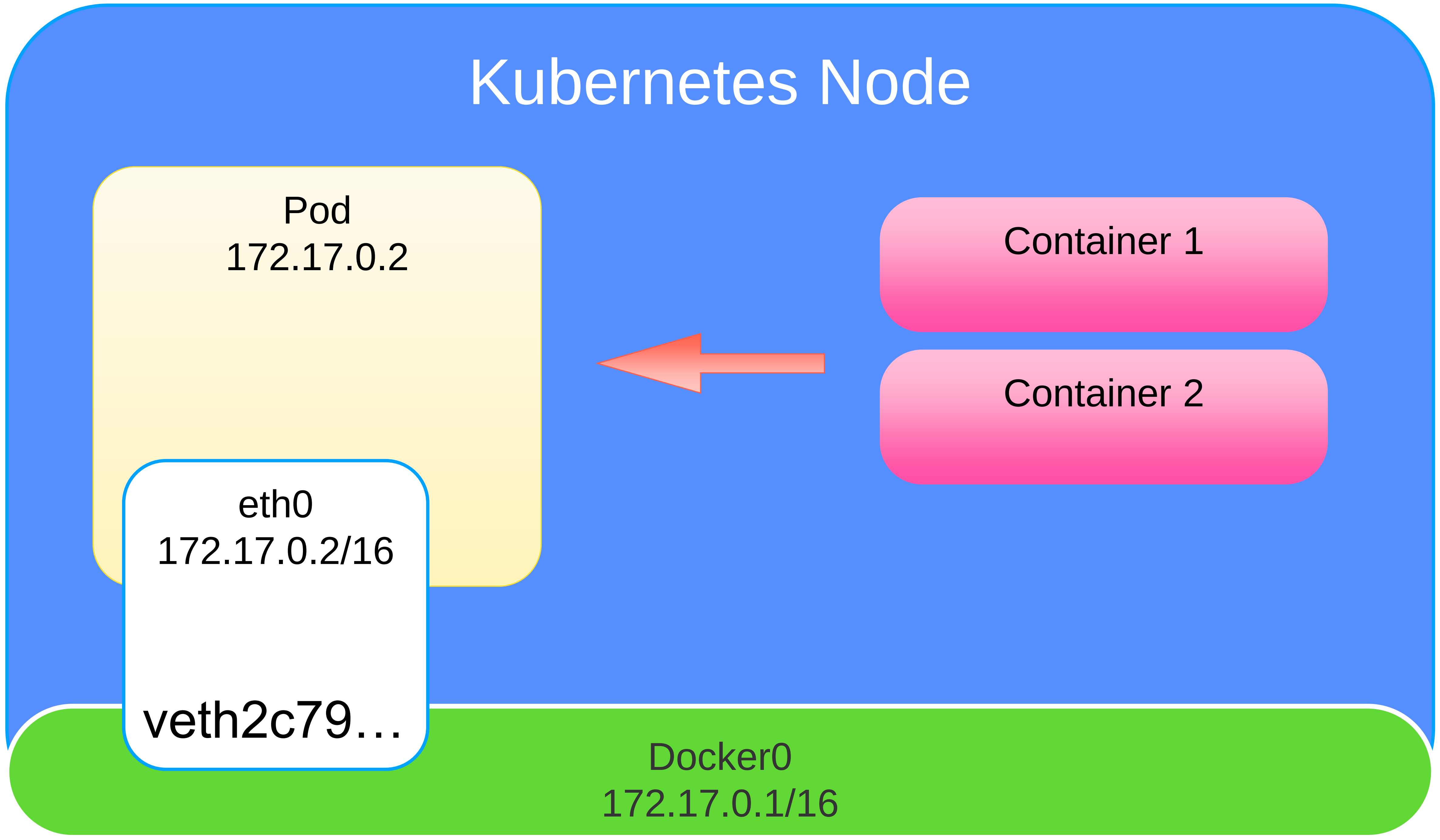
https://kubernetes.io/docs/concepts/cluster-administration/networking/

Kubernetes Networks

- Containers communication in a Pod
  - How to create a Pod
  - Assign a Pod a unique IP
- Pods on different nodes
- Pod to Service

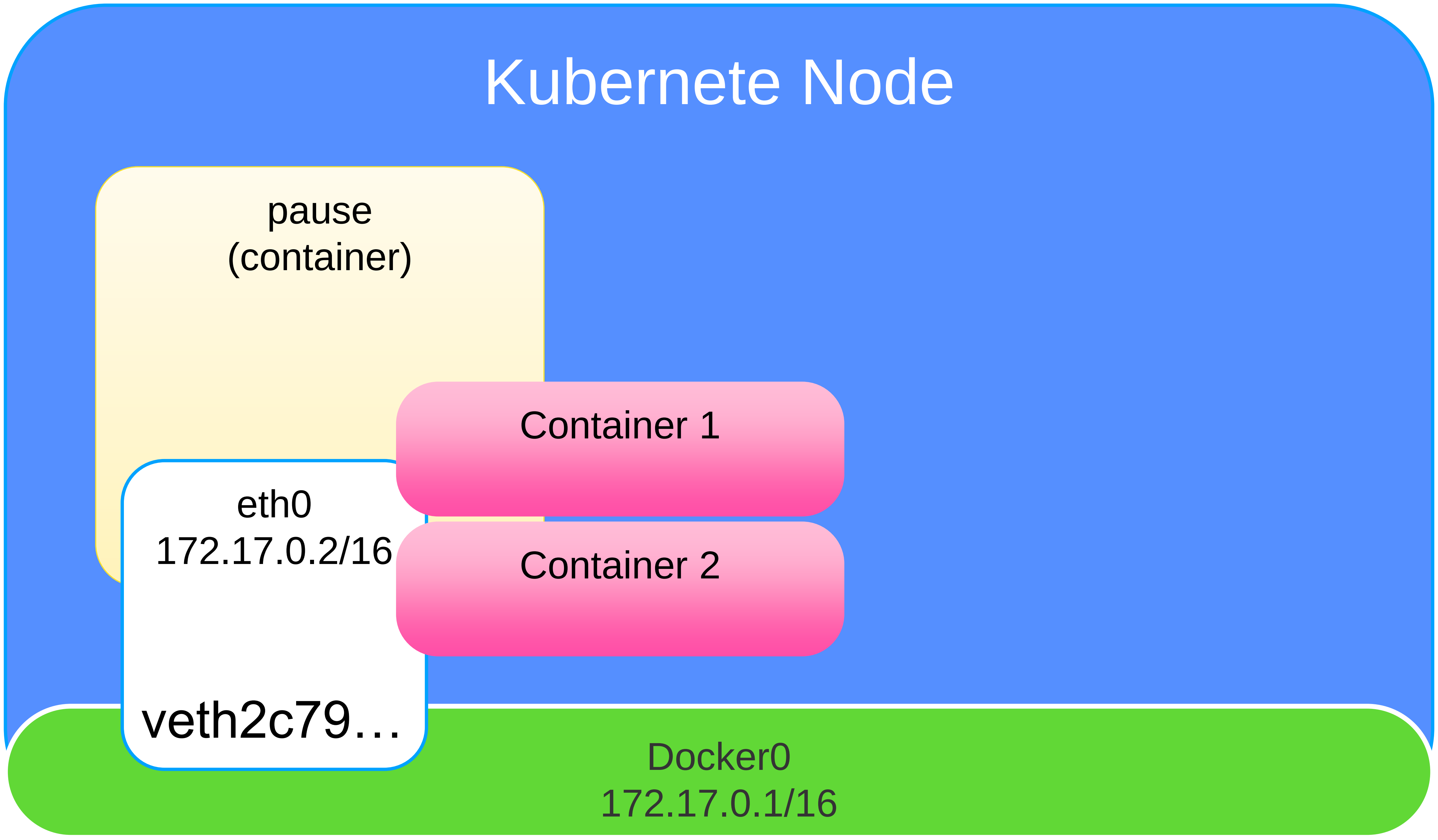
https://kubernetes.io/docs/concepts/cluster-administration/networking/

Containers in a Pod



How to have many containers into a Pod?

Pause



Create pause, and “attach” containers to its network

Pod Networking

- Container unique IP -> Pod unique IP
  - The pause container get its **IP** and then pause
  - Add containers to pause’s networks
  - Containers communicate with localhost
  - Containers share the same IP across cluster

Kubernetes Networks

- Containers communication in a Pod
  - How to create a Pod
  - Assign a Pod a unique IP
- Pods on different nodes
- Pod to Service

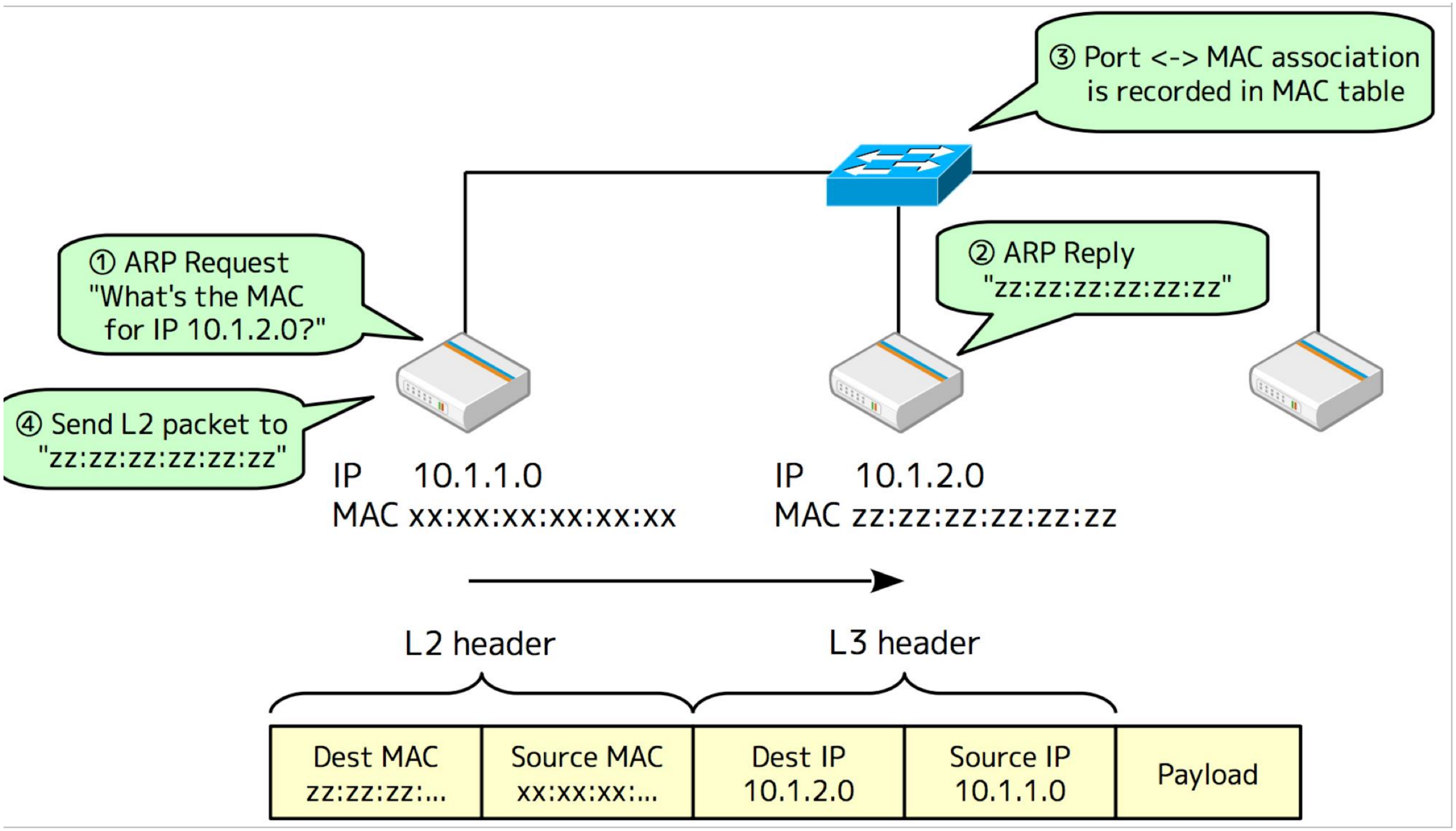
https://kubernetes.io/docs/concepts/cluster-administration/networking/

Flannel

- A flanneld on each host
- Flanneld creates a subnet for each host out of a larger address space
- Packet forward mechanism: VXLAN

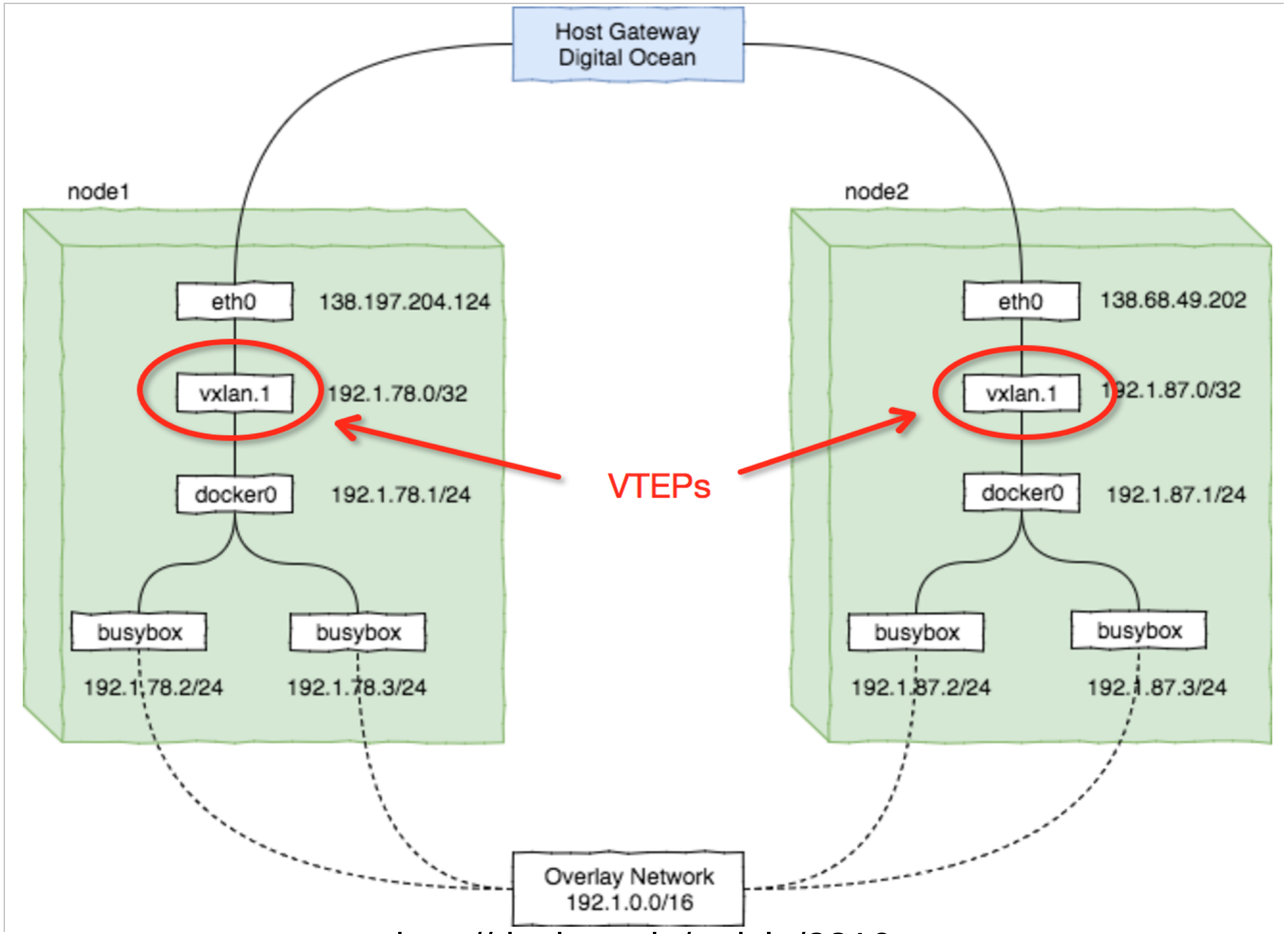
https://github.com/coreos/flannel

Virtual eXtensible Local Area Network


③ Port <-> MAC association is recorded in MAC table
① ARP Request "What's the MAC for IP 10.1.2.0?"
② ARP Reply "zz:zz:zz:zz:zz:zz"
④ Send L2 packet to "zz:zz:zz:zz:zz:zz"
IP 10.1.1.0
MAC xx:xx:xx:xx:xx:xx
IP 10.1.2.0
MAC zz:zz:zz:zz:zz:zz
L2 header
L3 header
Dest MAC zz:zz:zz:...
Source MAC xx:xx:xx:...
Dest IP 10.1.2.0
Source IP 10.1.1.0
Payload

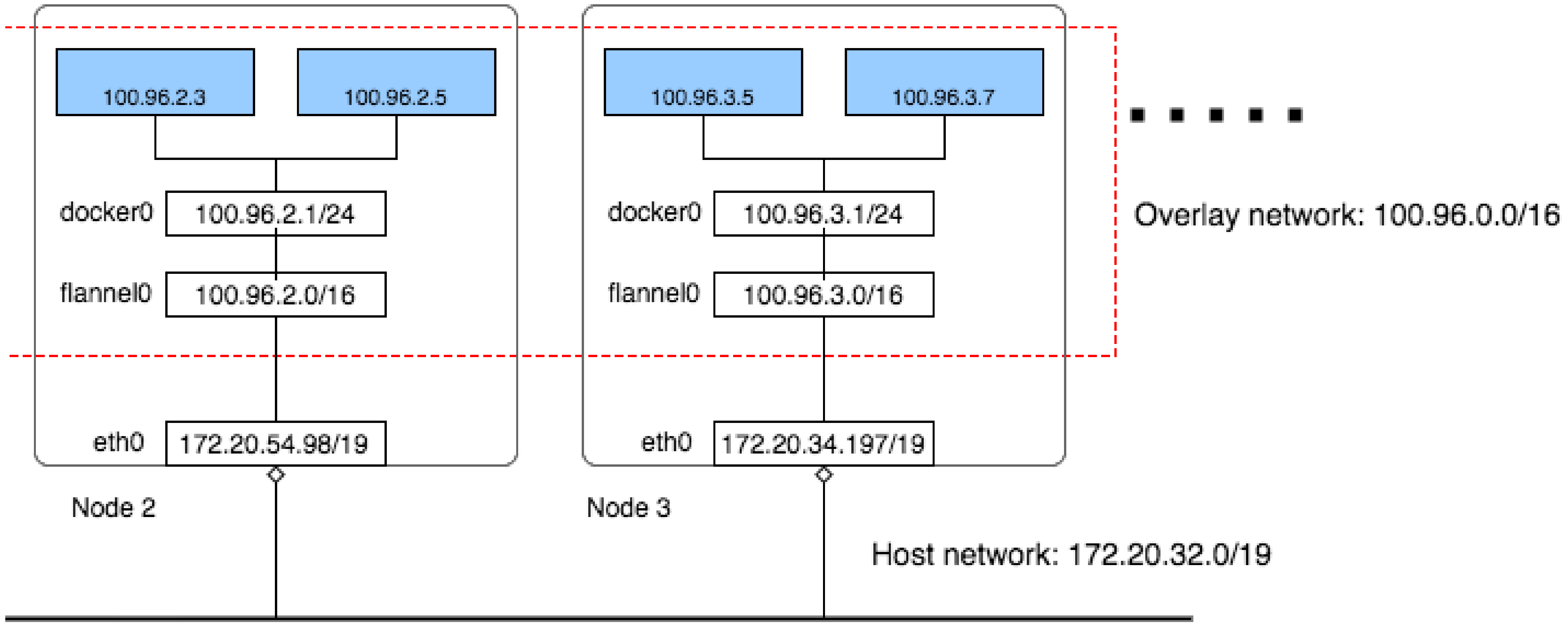
http://dockone.io/article/2216

VXLAN



http://dockone.io/article/2216

Overlay Network



https://blog.laputa.io/kubernetes-flannel-networking-6a1cb1f8ec7c

Kubernetes Networks

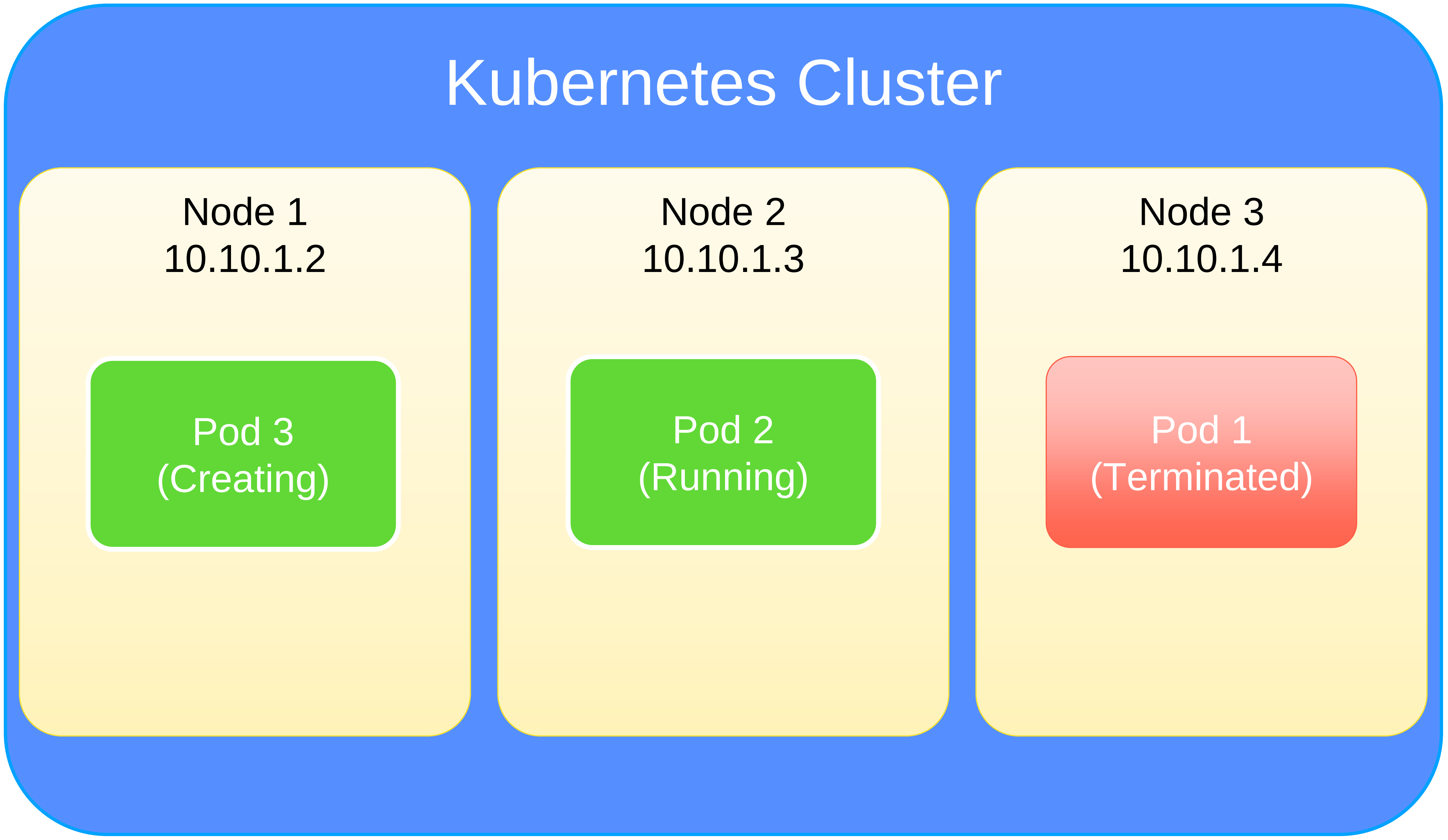
- Containers communication in a Pod -> localhost
  - How to create a Pod -> pause
  - Assign a Pod a unique IP -> flannel address space(CNI)
- Pods communicates across different nodes
  -> flannel (vxlan, overlay networks)
- Pod to Service

Kubernetes Networks

- Pod has unique IP but hard to access through pure IP
- Pod to kubernetes Service

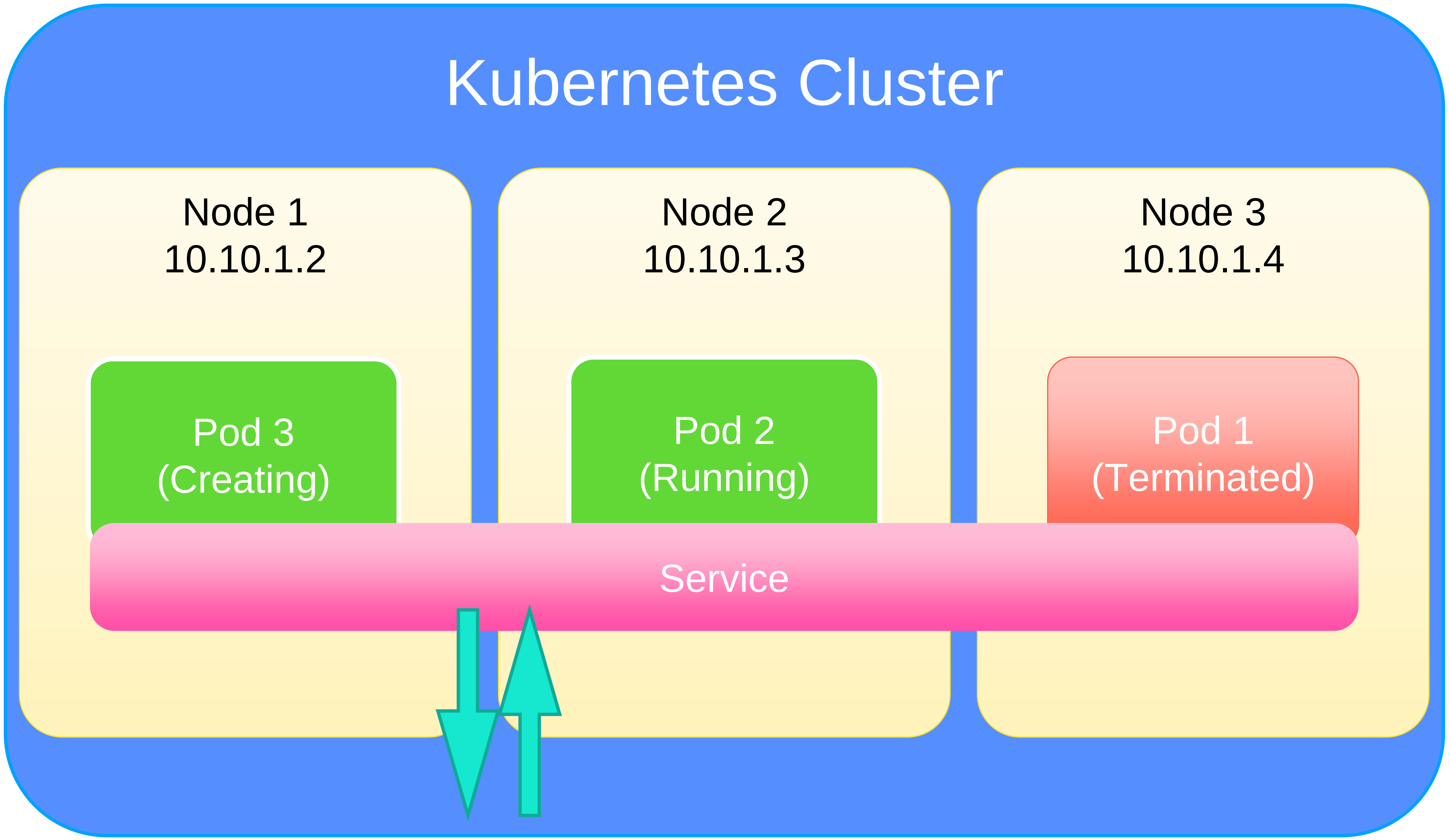
https://kubernetes.io/docs/concepts/services-networking/service/

Pod to Service



How to access app instead of access to a specific Pod IP?

Kubernetes Service



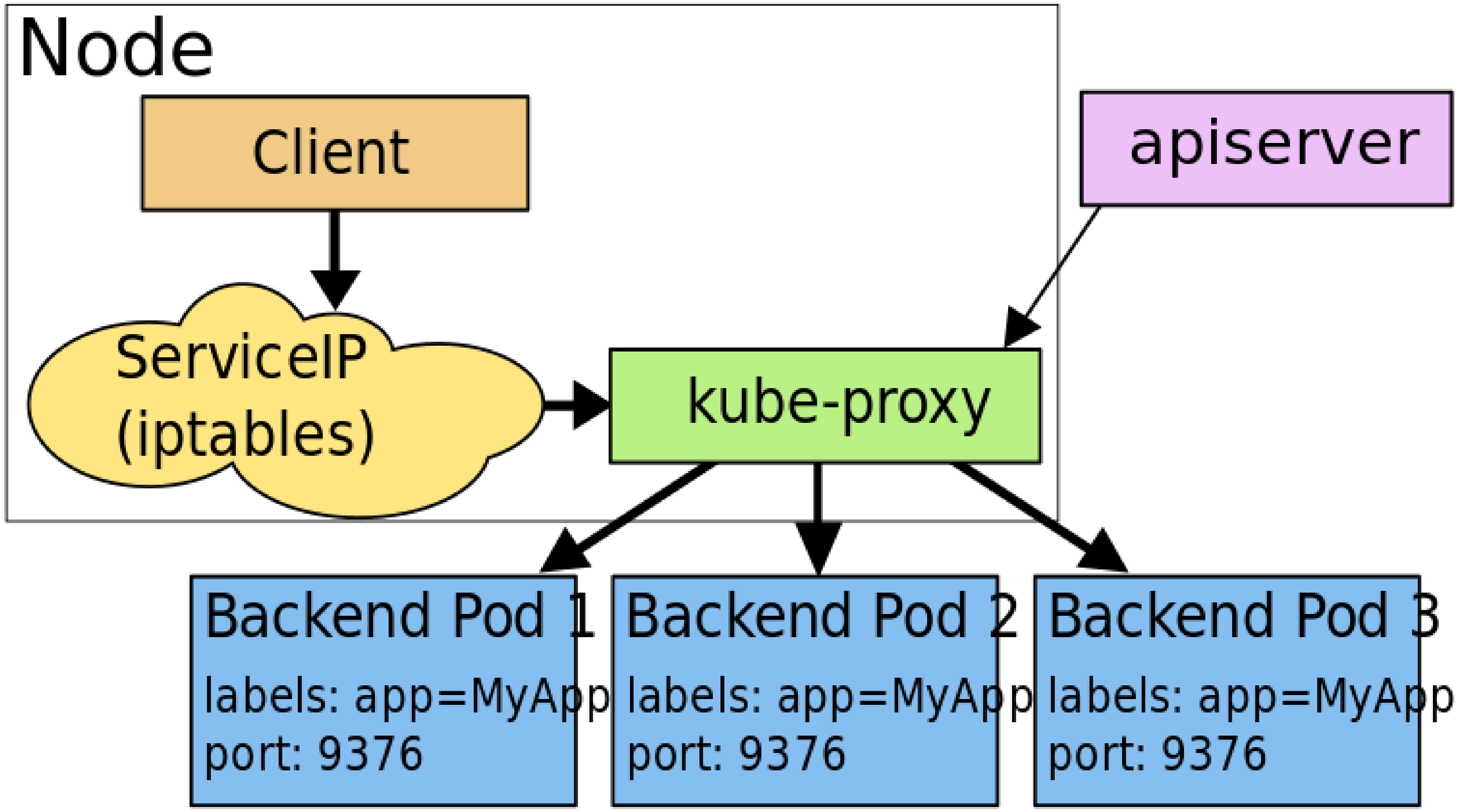
Access to a set of Pods through a single endpoint

Kubernetes Service

- A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them - sometimes called a micro-service.

https://kubernetes.io/docs/concepts/services-networking/service/

Kube-proxy



https://kubernetes.io/docs/concepts/services-networking/service/

Kube-proxy

- kube-proxy is responsible for implementing a form of virtual IP for Services of type other than ExternalName.
- In short, update iptable rules to nodes for each services

```
node$ sudo iptables-save -t nat

-A KUBE-SERVICES ! -s 10.233.64.0/18 -d 10.233.0.1/32 -p tcp -m comment --comment "defa

-A KUBE-SERVICES -d 10.233.0.1/32 -p tcp -m comment --comment "default/kubernetes:https
```

https://kubernetes.io/docs/concepts/services-networking/service/

Discovering Service

- Environment variables injection by kubelet
  REDIS_MASTER_SERVICE_HOST=10.0.0.11
  REDIS_MASTER_SERVICE_PORT=6379

- DNS (Recommended)
  The DNS server watches the Kubernetes API for new Service and creates a set of DNS records for each

https://kubernetes.io/docs/concepts/services-networking/service/

Kube-dns

- Kubernetes DNS schedules a DNS Pod and Service on the cluster, and configures the kubelets to tell individual containers to use the DNS Service’s IP to resolve DNS names.

Service
my-svc.my-namespace.svc.cluster.local -> nginx.default.svc.cluster.local

Pod
pod-ip-address.my-namespace.pod.cluster.local -> nginx-1.default.pod.cluster.local

https://kubernetes.io/docs/concepts/services-networking/service/

Kubernetes Networks

- Containers communication in a Pod
  -> localhost
- Assign a Pod a unique IP
  -> flannel address space
- Pods on different nodes
  -> flannel (VXLAN, overlay networks)
- Pod to Service
  -> service, proxy, dns

Thank you

Kubernetes Networks

Thank you