KubeCon CloudNativeCon Europe 2021

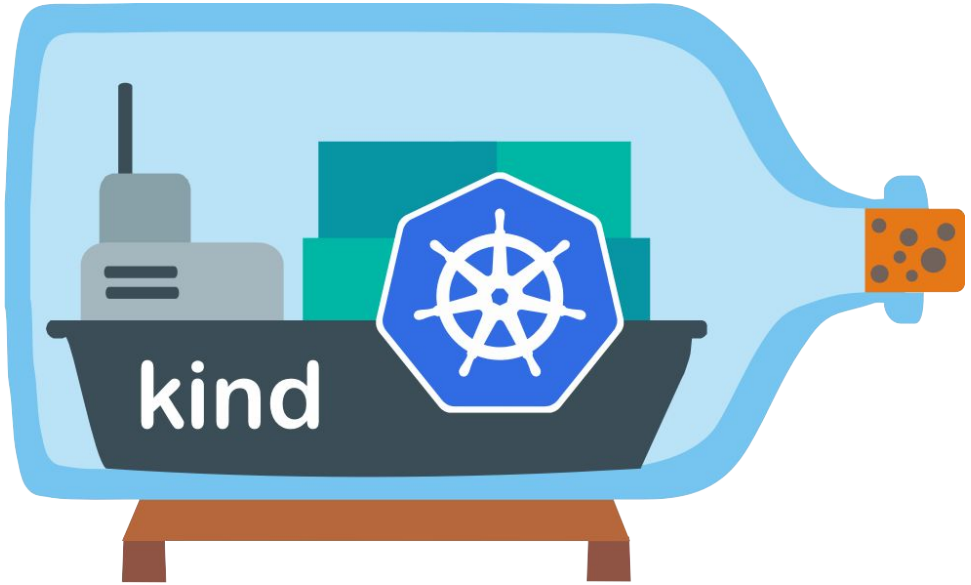
Virtual

Kubernetes Advanced Networking Testing with KIND

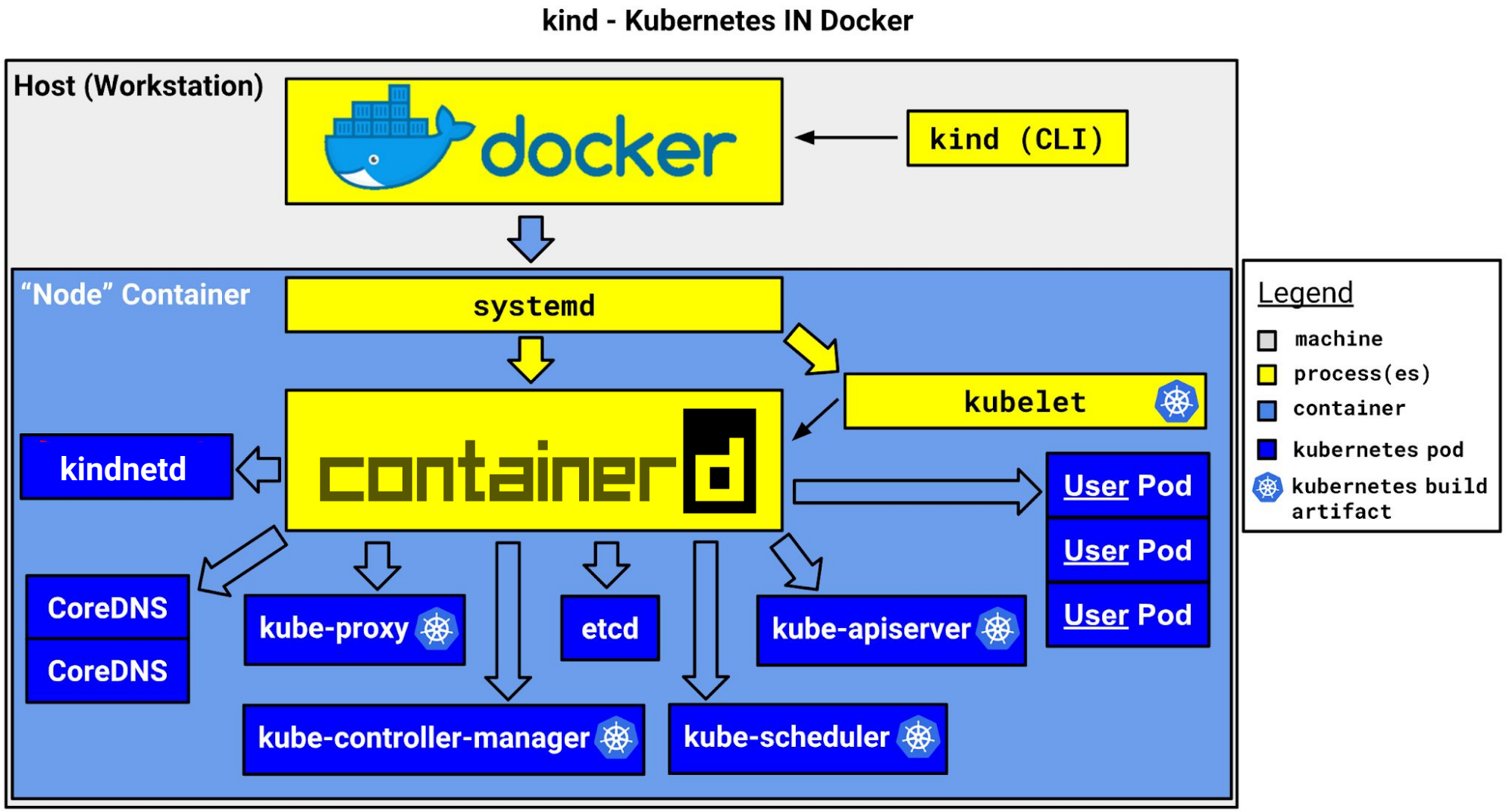
Antonio Ojea

What is KIND?

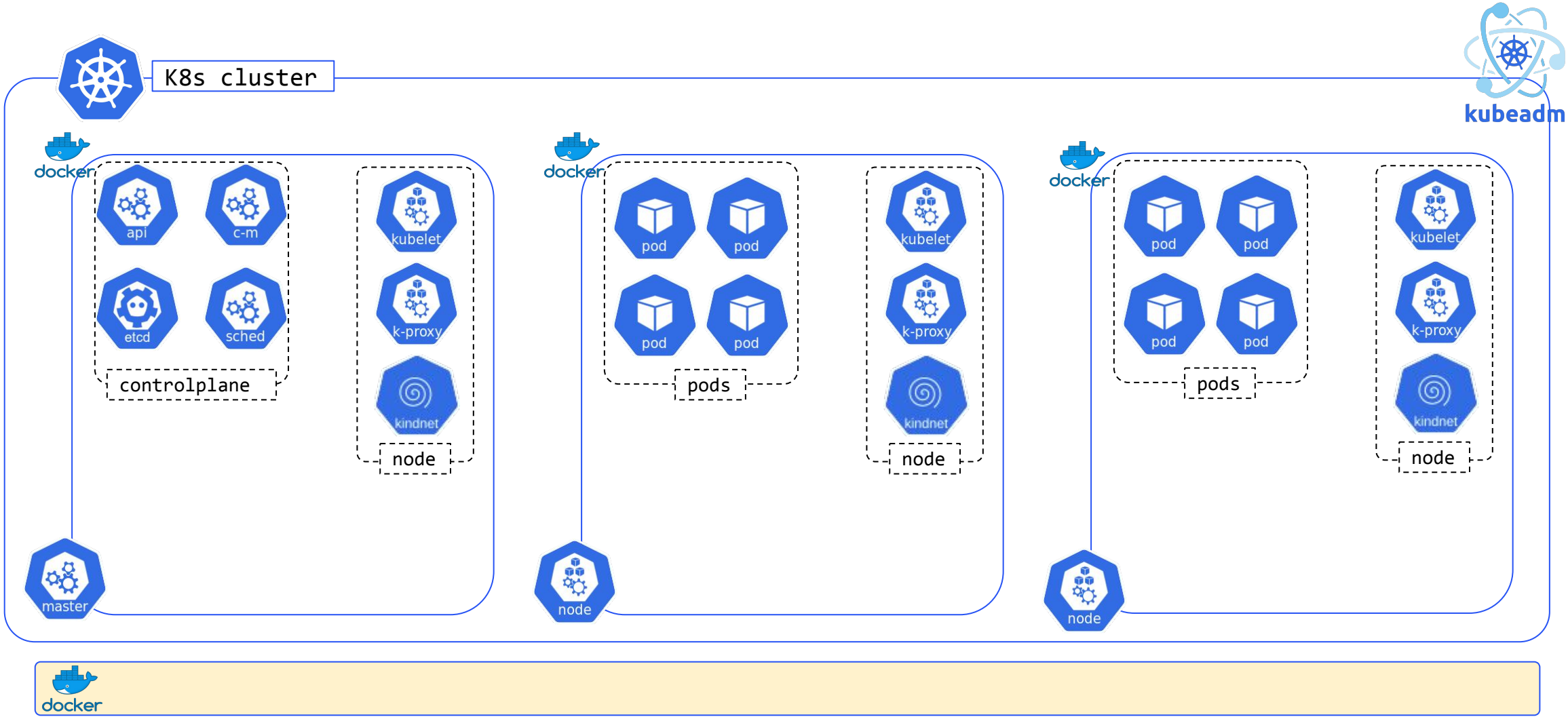
- **K**ubernetes-**in**-**D**ocker
- Uses Docker containers to simulate nodes
- **Multi-node**
- HA control plane
- **Build & run Kubernetes from source**
- Boots a cluster in ~30s 🎉



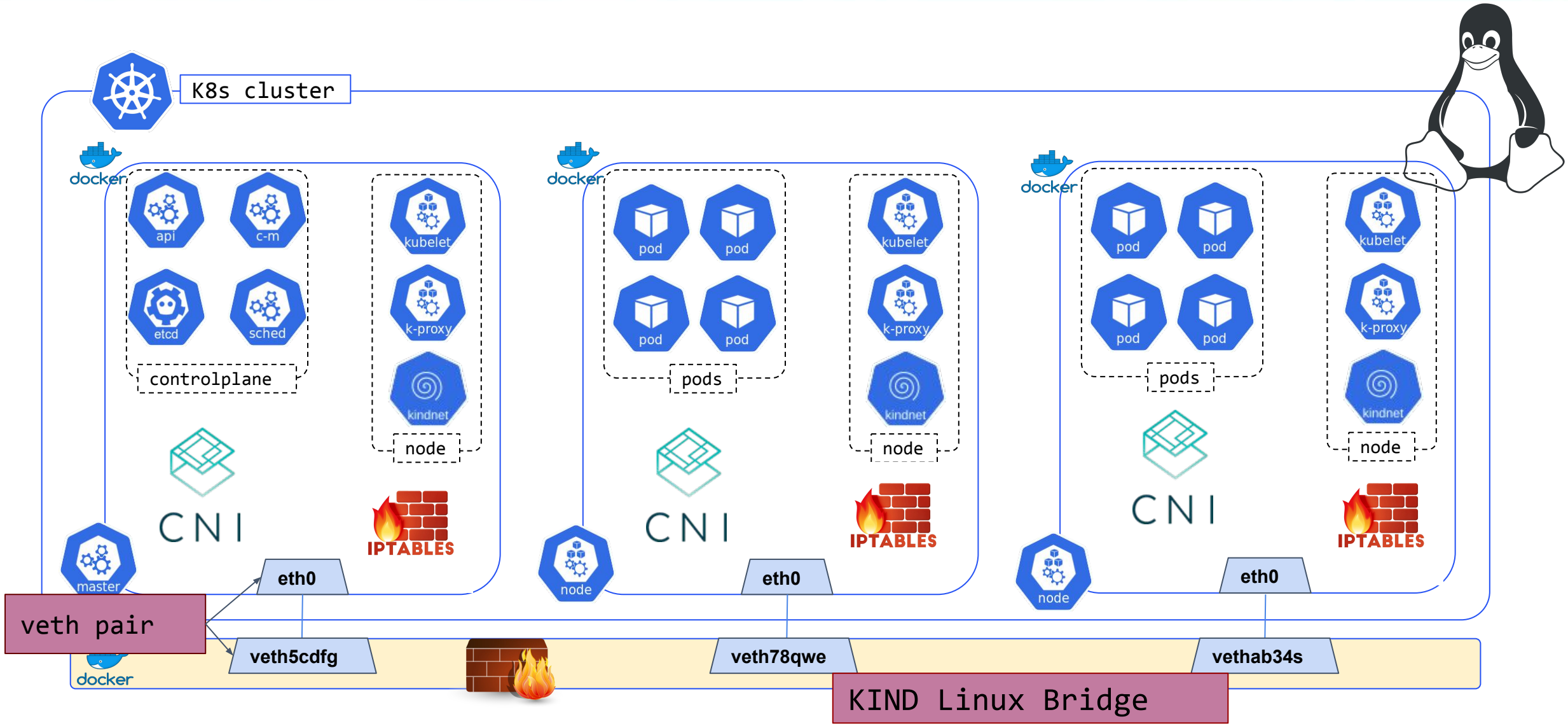
Kubernetes IN Docker



KIND Cluster



KIND Networking



KIND API = Plugins

pkg.go.dev/sigs.k8s.io/kind

Black Lives Matter

GO Search for a package Why Go Getting Started

Discover Packages > sigs.k8s.io/kind

kind command module

Version: v0.10.0 LATEST | Published: Jan 23, 2021 | License: Apache-2.0 | Imports: 1 | Imported by: 0

Jump to ...

- README
  - Please see for more in-dept…
  - Installation and usage
  - Community
  - Why kind?
- Documentation
- Source Files
- Directories

README

Please see Our Documentation for more in-depth installation etc.

Expand ▾

Documentation

Overview

github.com/aojea/kind-networking-plugins

README.md

kind-networking-plugins

Plugins to extend KIND networking capabilities with plugins using the KIND API

Overview

KIND is a tool for running local Kubernetes clusters using Docker container “nodes”.

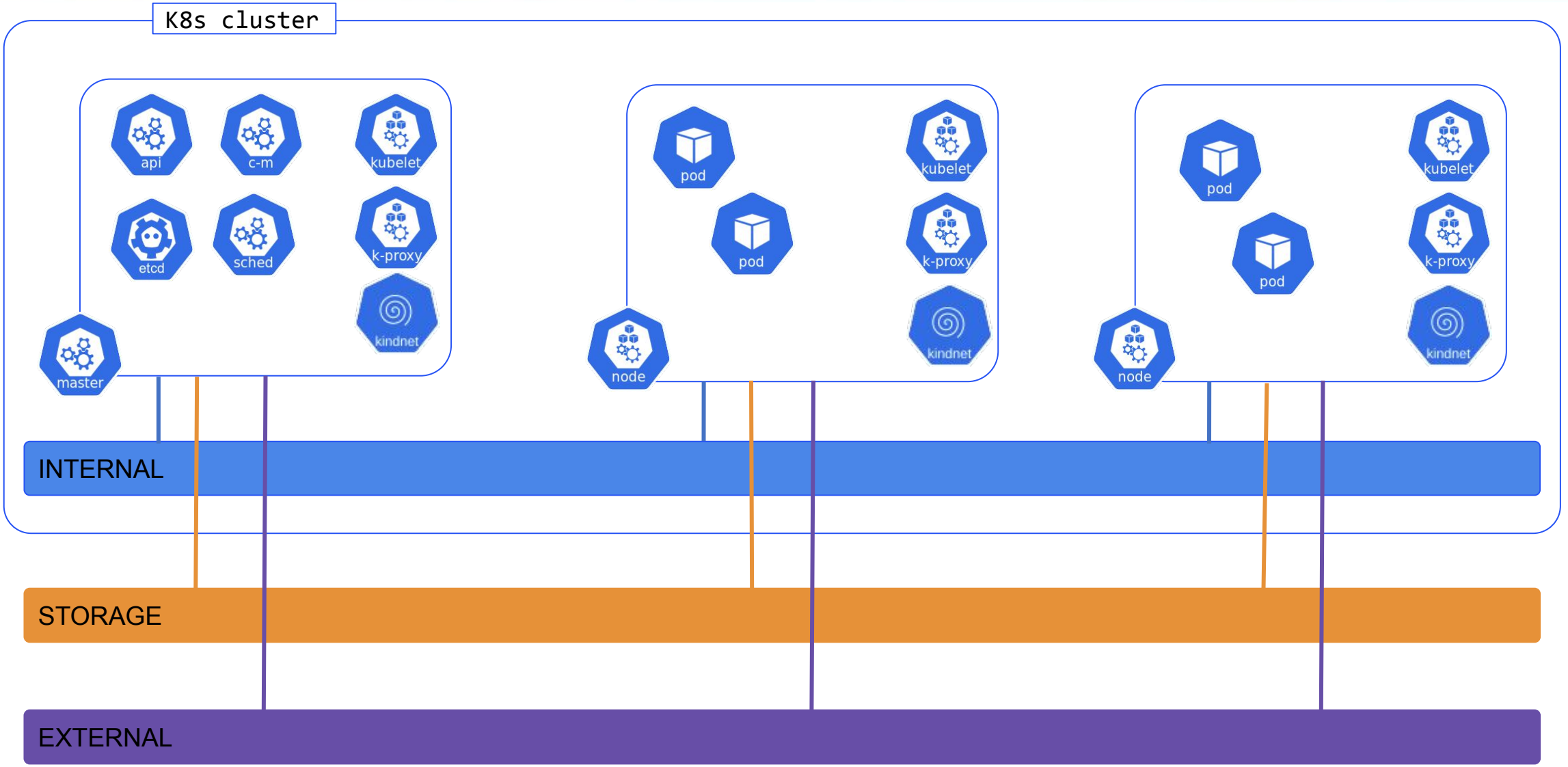
KIND was primarily designed for testing Kubernetes itself, but may be used for local dev
a strong focus in stability and resilience, thus adding new features is complicated. Howe
can be leveraged for automation.

In the other hand, testing networking is always complicated, because it requires more co
cover all the features. Traditionally, this was difficult to automate, but nowadays, current
containers and virual networks make it possible.

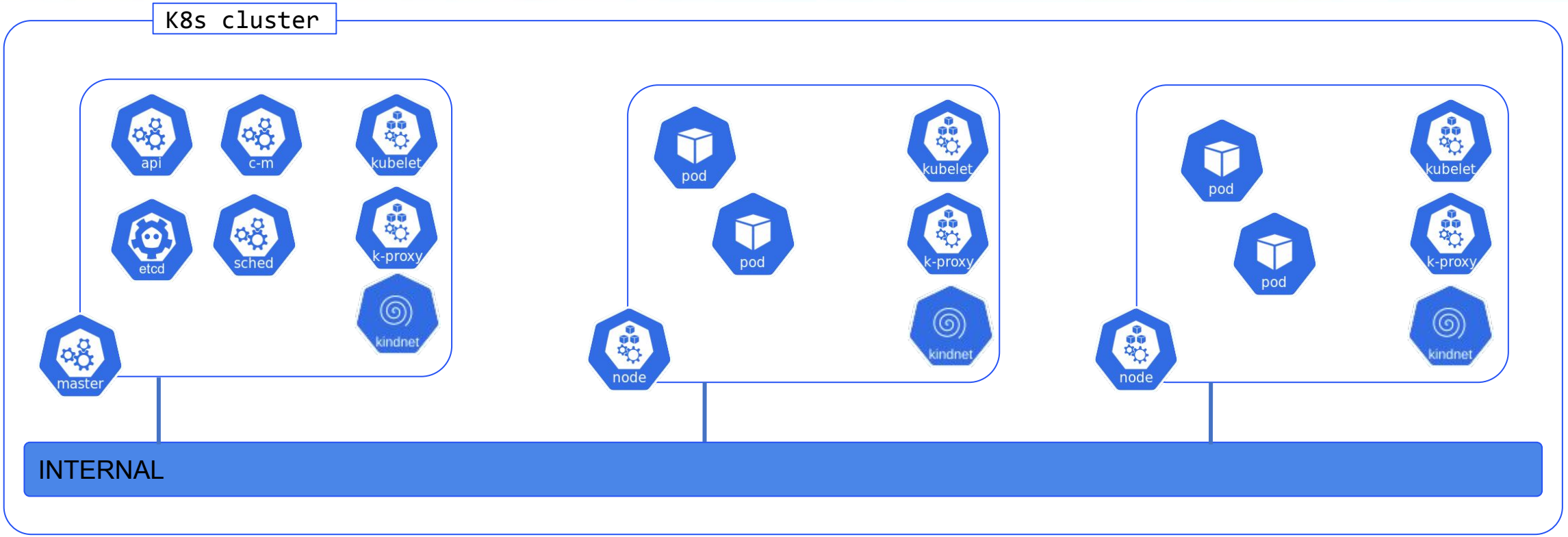
This repository contains some example plugins to demonstrate how to extend KIND and
Kubernetes clusters.

Multicluster

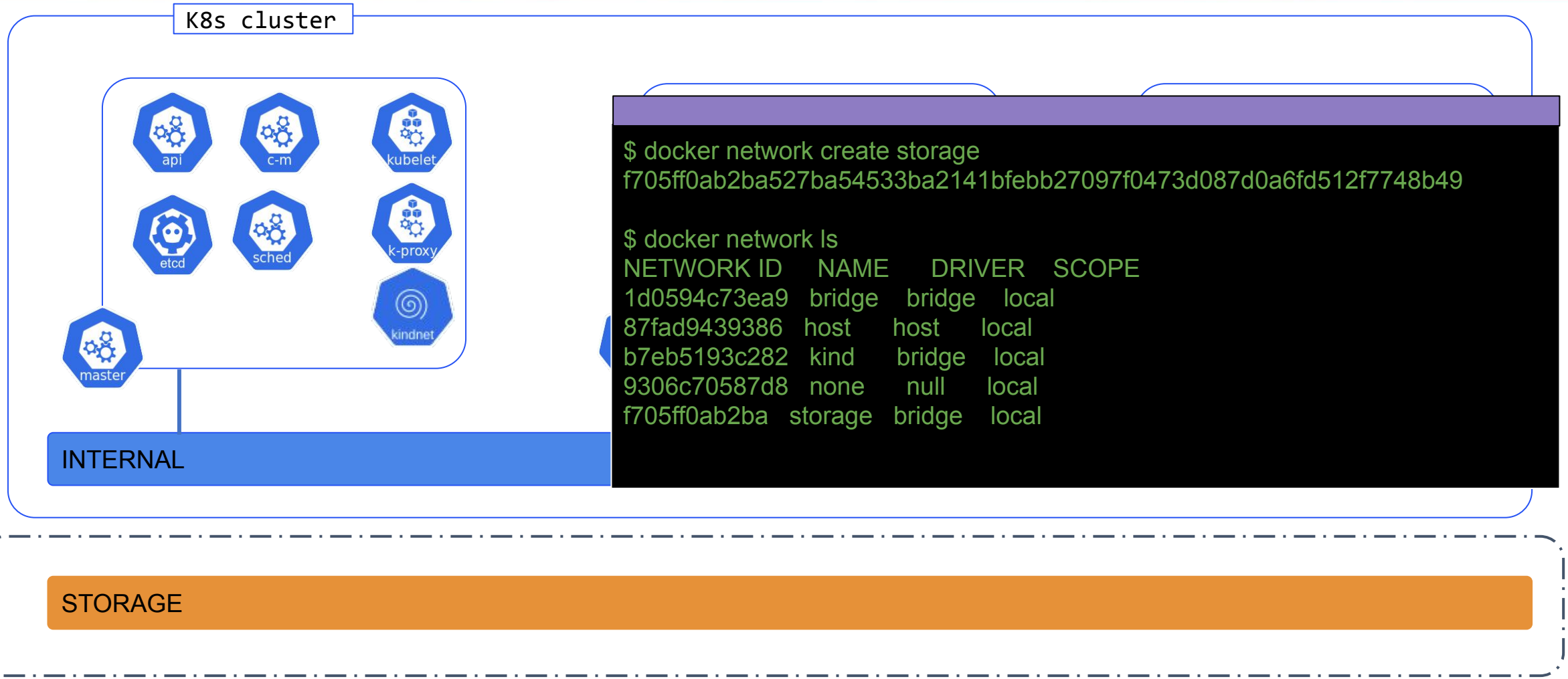
KIND Multiple Networks



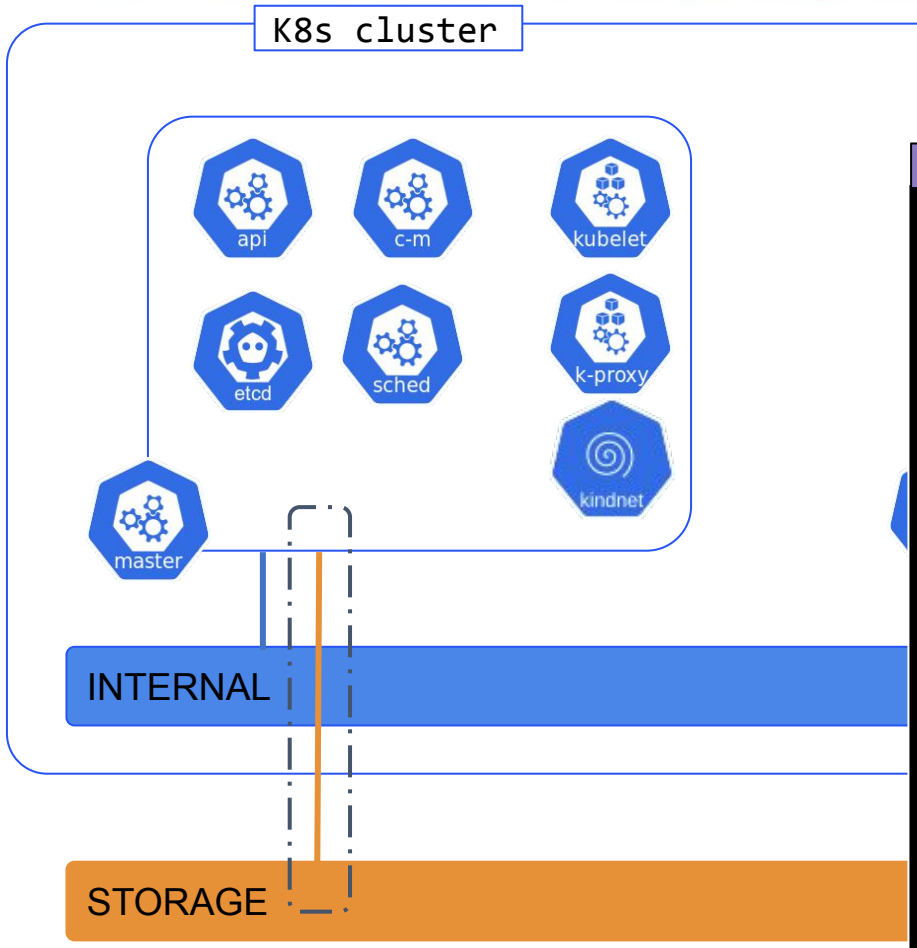
KIND Multiple Networks



KIND Multiple Networks

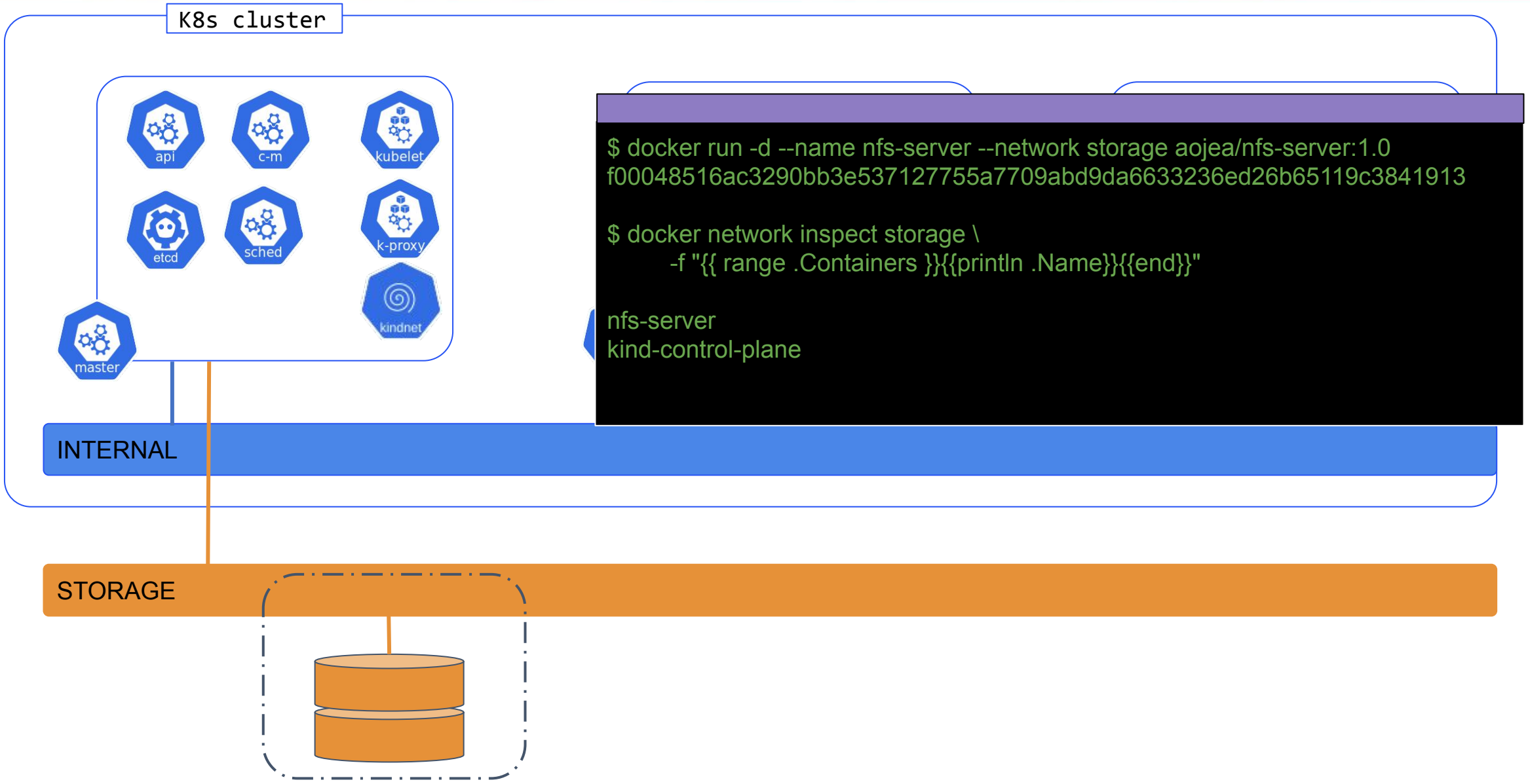


KIND Multiple Networks

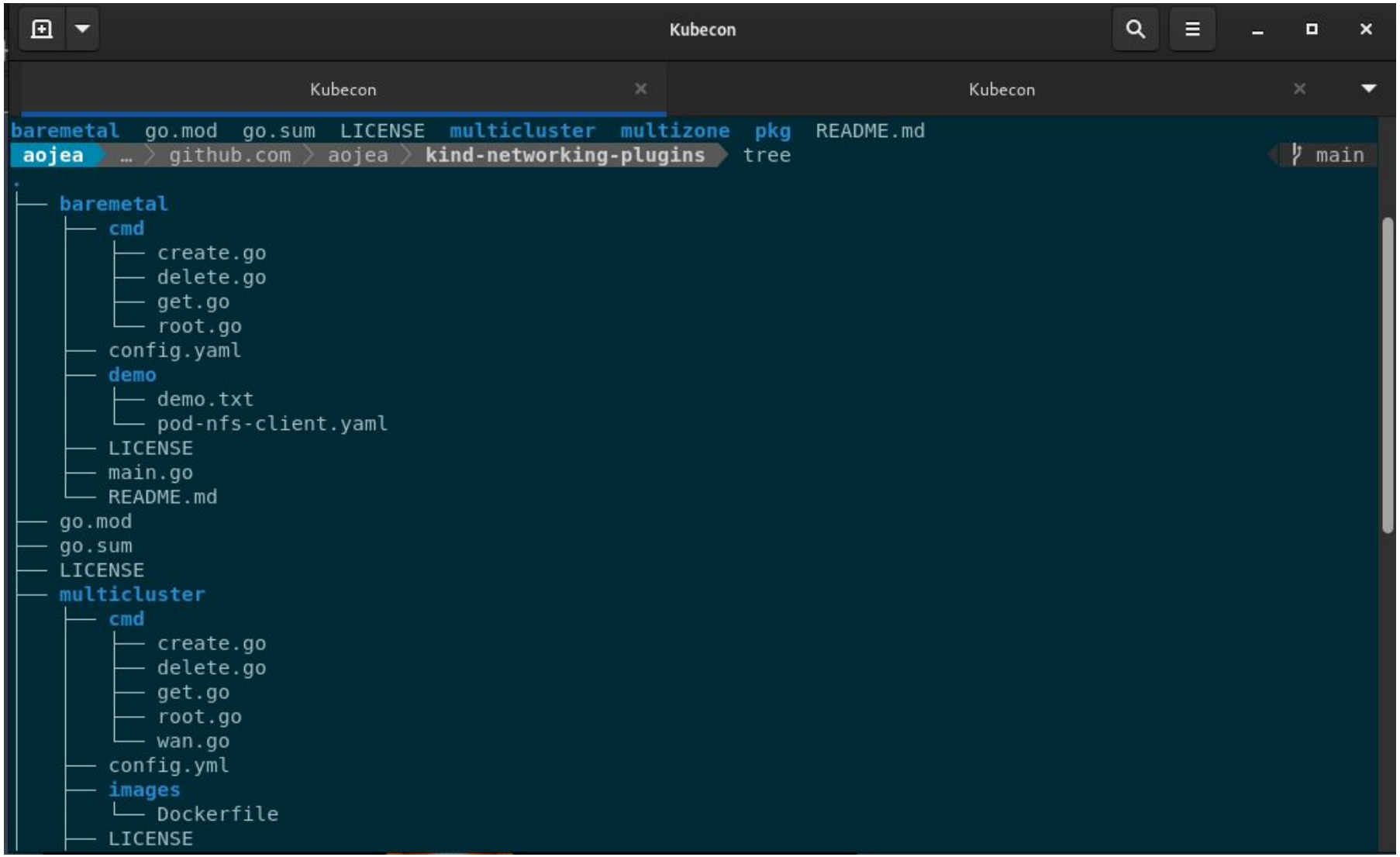



```
$ docker network connect storage kind-control-plane
f705ff0ab2ba527ba54533ba2141bfebb27097f0473d087d0a6fd512f7748b49

$ docker exec -it kind-control-plane ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    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
79: eth0@if80: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500
qdisc noqueue
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
81: eth1@if82: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500
qdisc noqueue
    link/ether 02:42:ac:19:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.25.0.3/16 brd 172.25.255.255 scope global eth1
       valid_lft forever preferred_lft forever
```

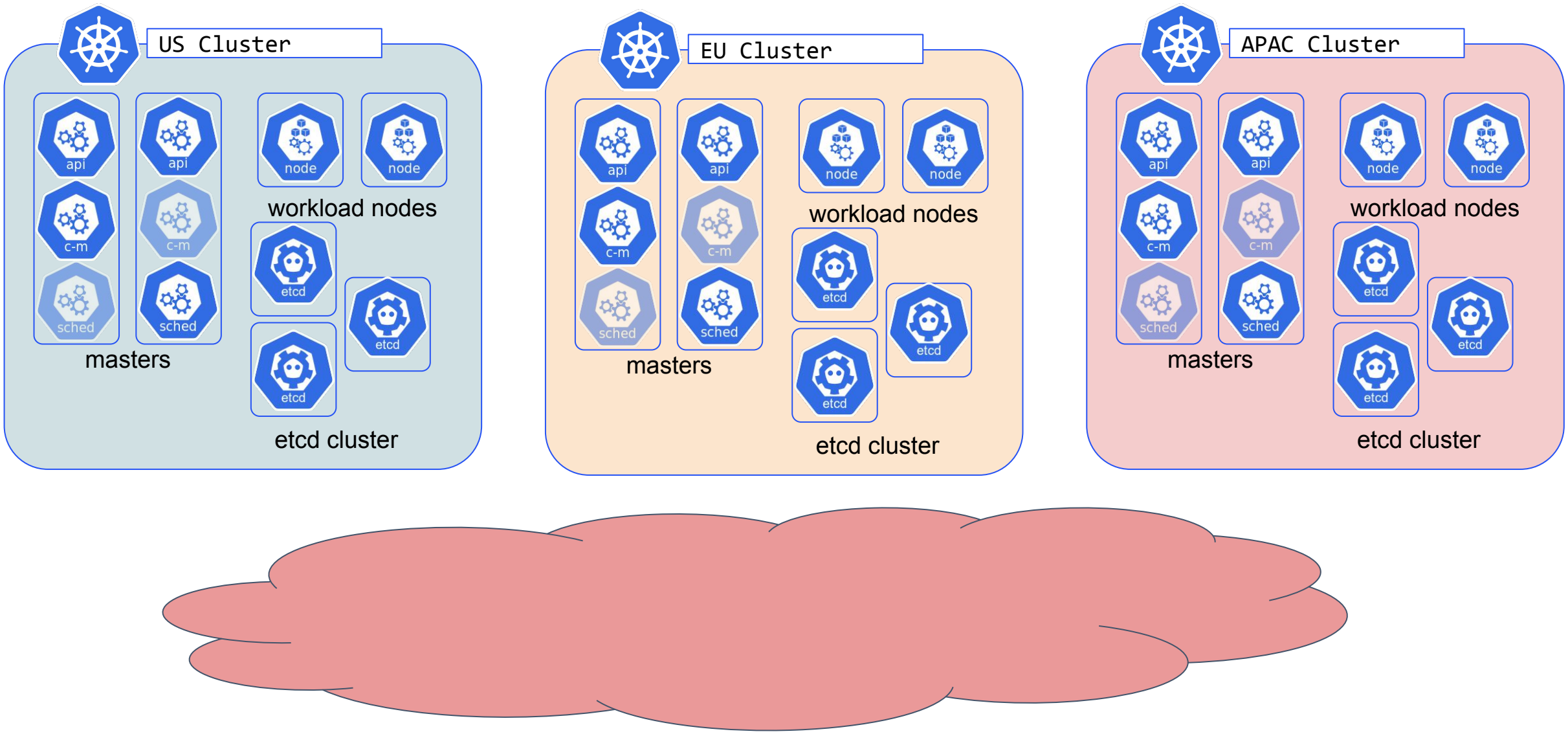
KIND Multiple Networks



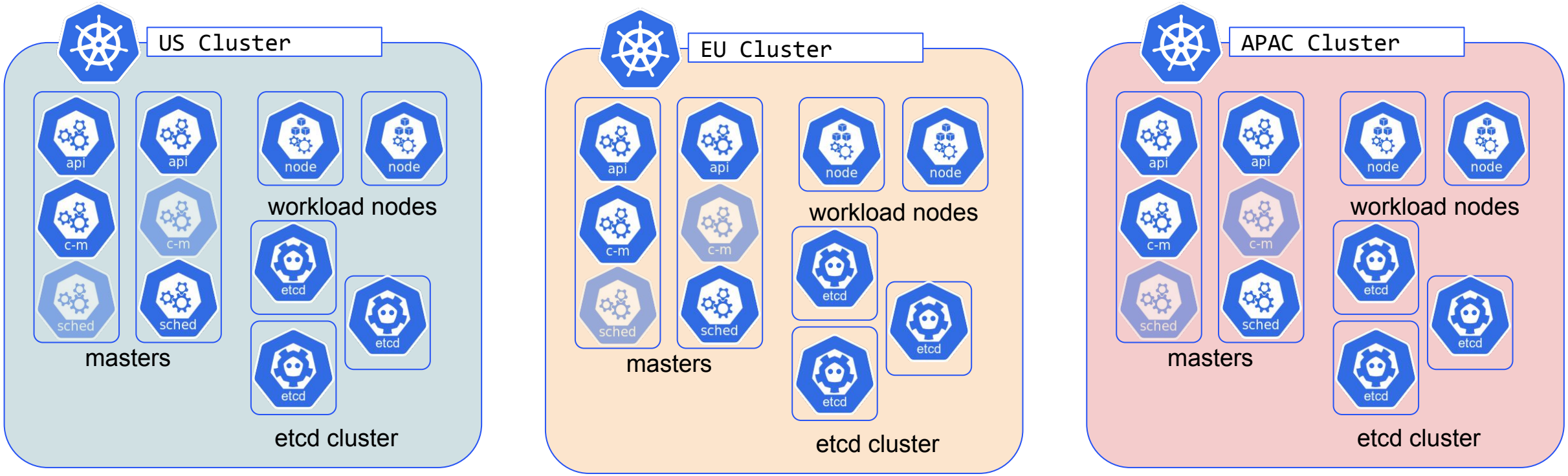
DEMO: Multiple Networks

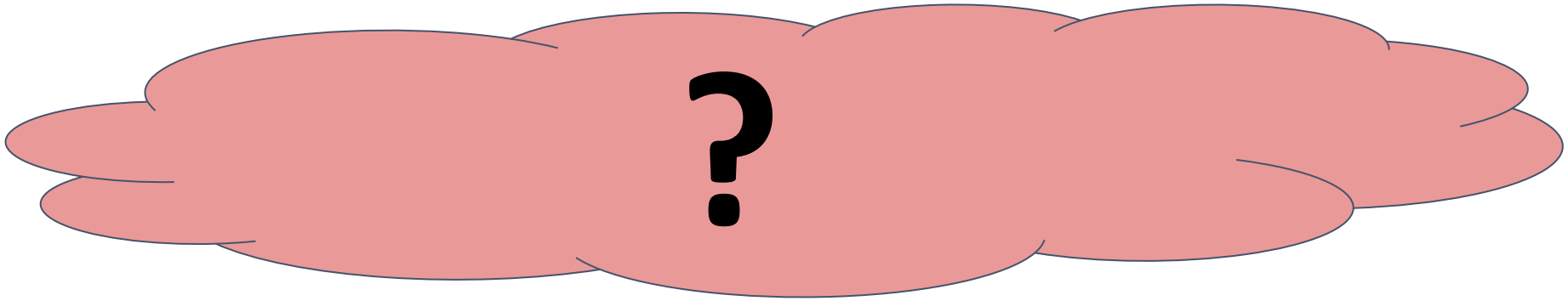


KIND MultiCluster

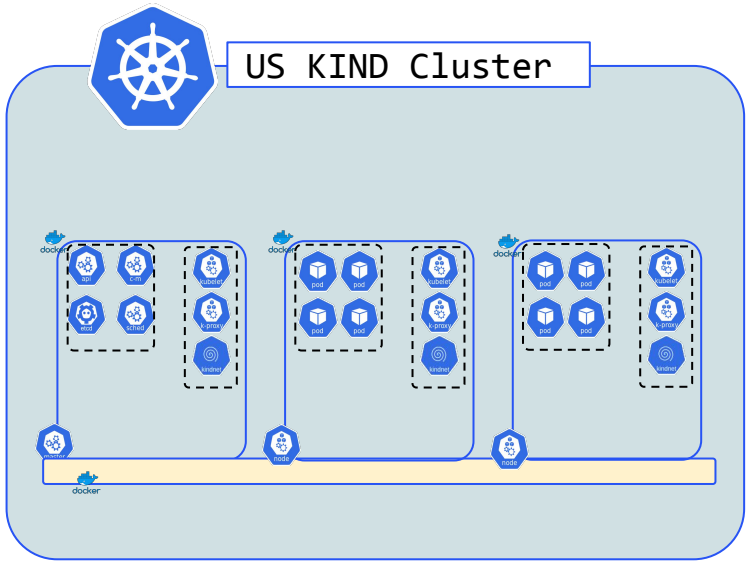


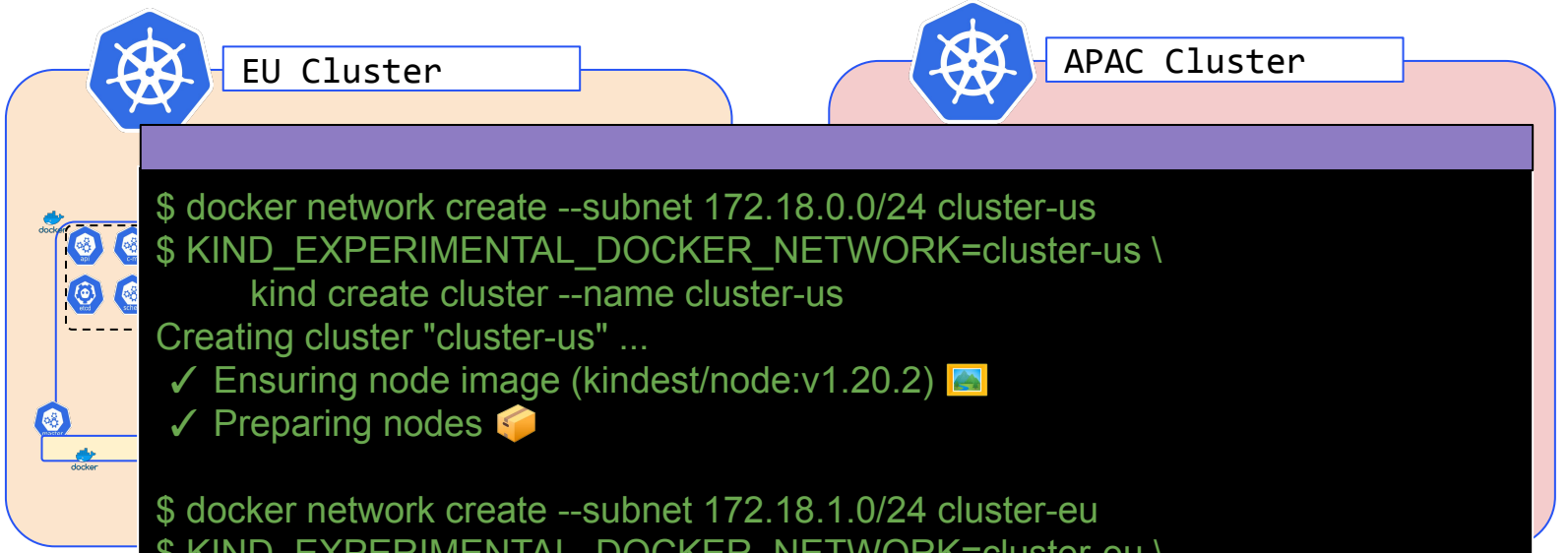
KIND MultiCluster





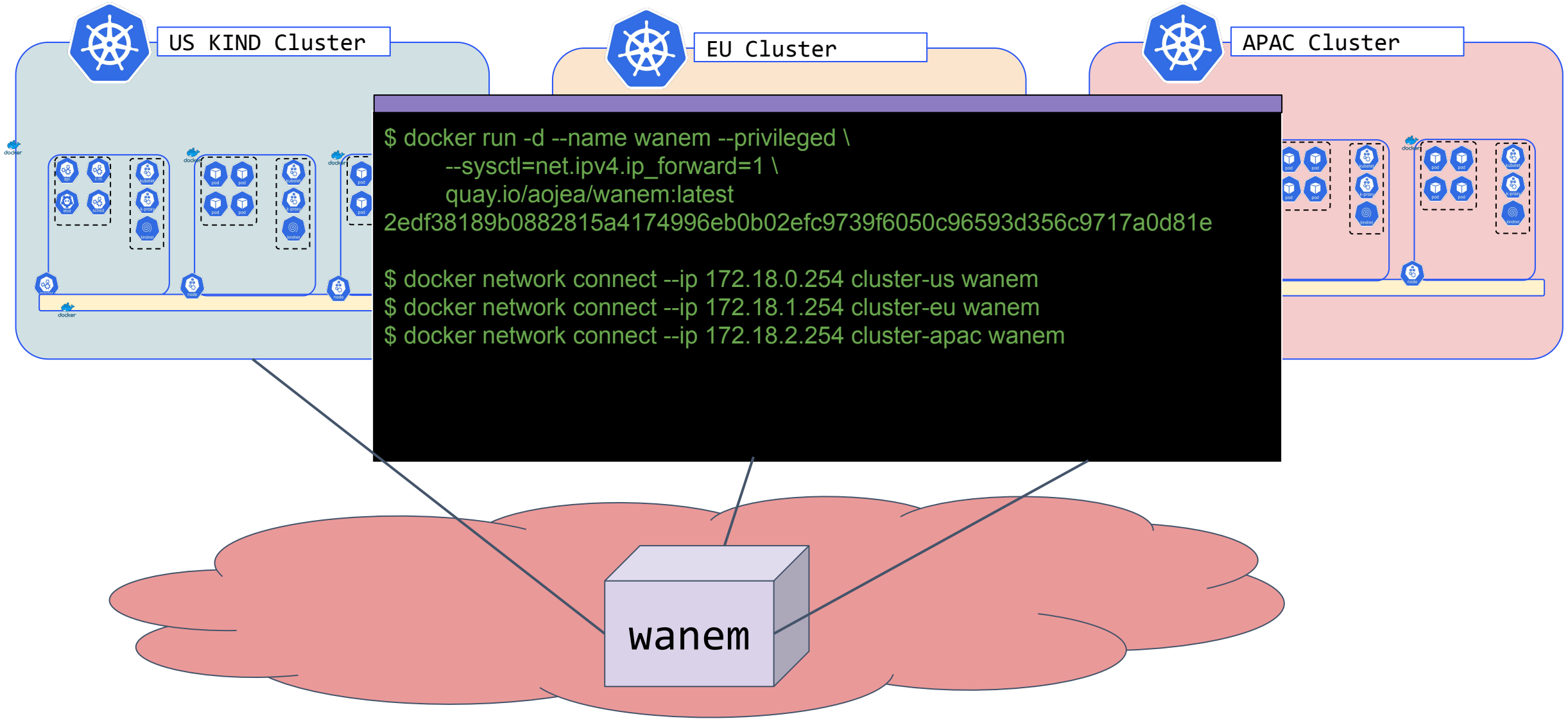
KIND MultiCluster



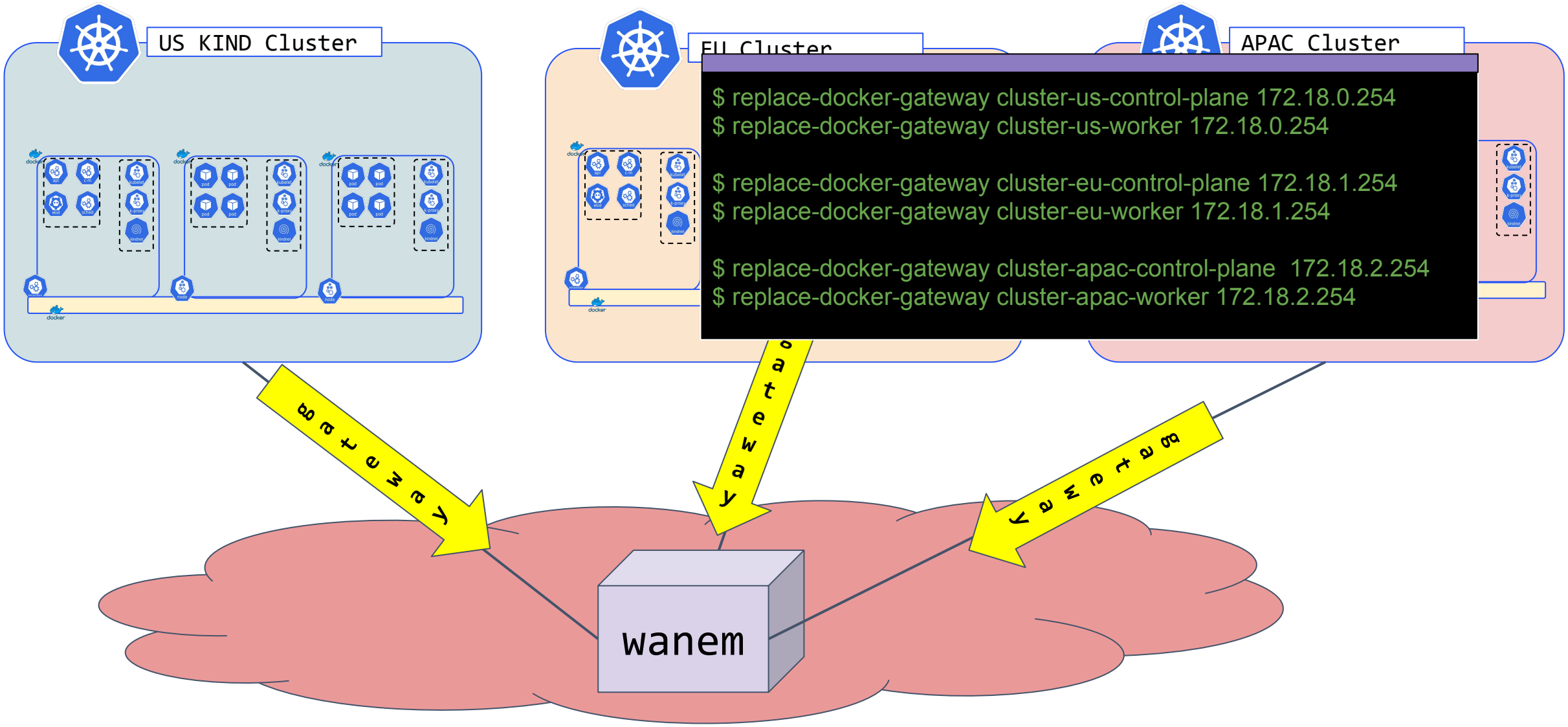



```
$ docker network create --subnet 172.18.0.0/24 cluster-us
$ KIND_EXPERIMENTAL_DOCKER_NETWORK=cluster-us \
        kind create cluster --name cluster-us
Creating cluster "cluster-us" ...
 ✓ Ensuring node image (kindest/node:v1.20.2) 🖼
 ✓ Preparing nodes 📦

$ docker network create --subnet 172.18.1.0/24 cluster-eu
$ KIND_EXPERIMENTAL_DOCKER_NETWORK=cluster-eu \
        kind create cluster --name cluster-eu
Creating cluster "cluster-eu" ...
 ✓ Ensuring node image (kindest/node:v1.20.2) 🖼
 ✓ Preparing nodes 📦

$ docker network create --subnet 172.18.2.0/24 cluster-apac
$ KIND_EXPERIMENTAL_DOCKER_NETWORK=cluster-apac \
        kind create cluster --name cluster-apac
Creating cluster "cluster-apac" ...
 ✓ Ensuring node image (kindest/node:v1.20.2) 🖼
 ✓ Preparing nodes 📦
```

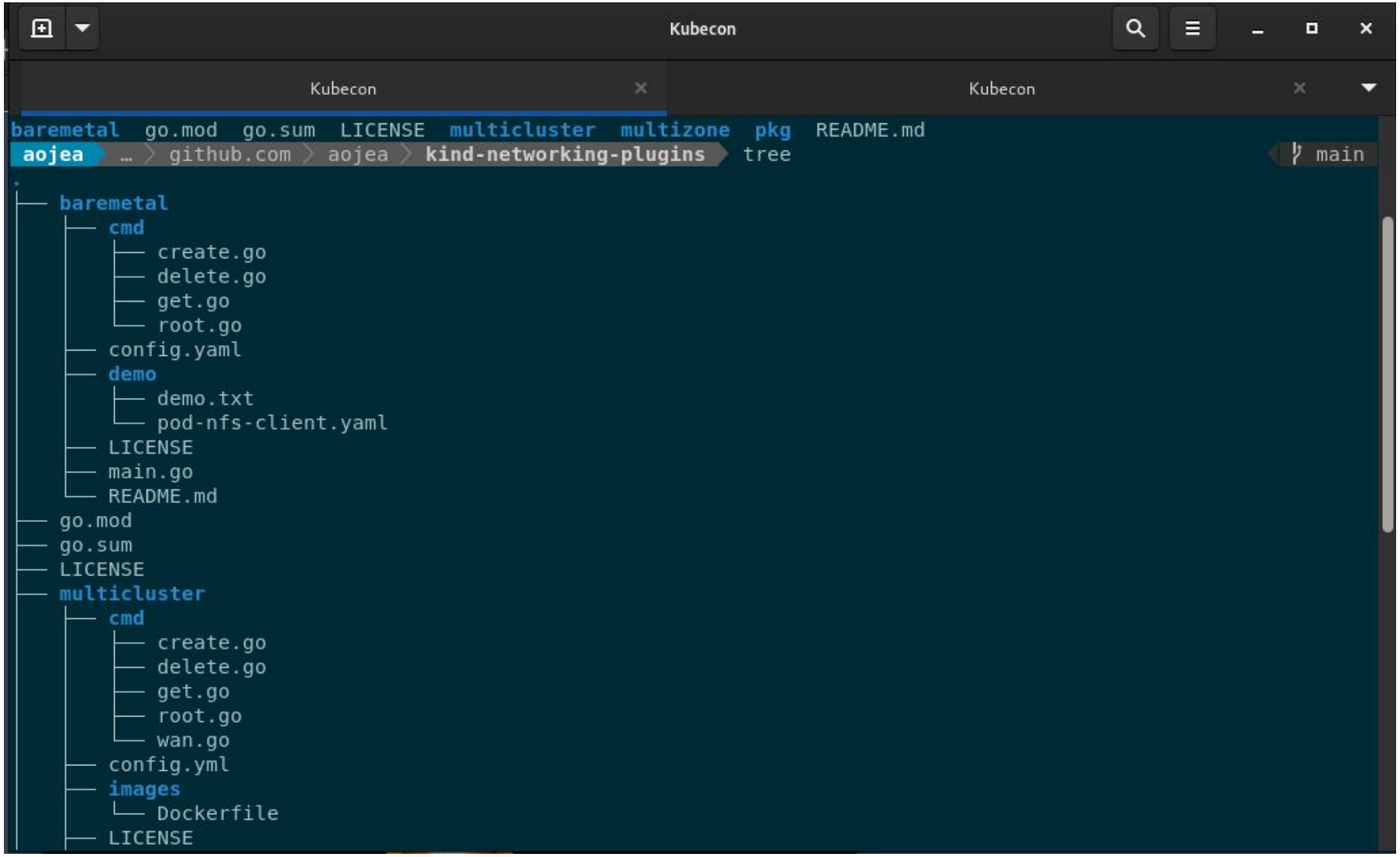
KIND MultiCluster



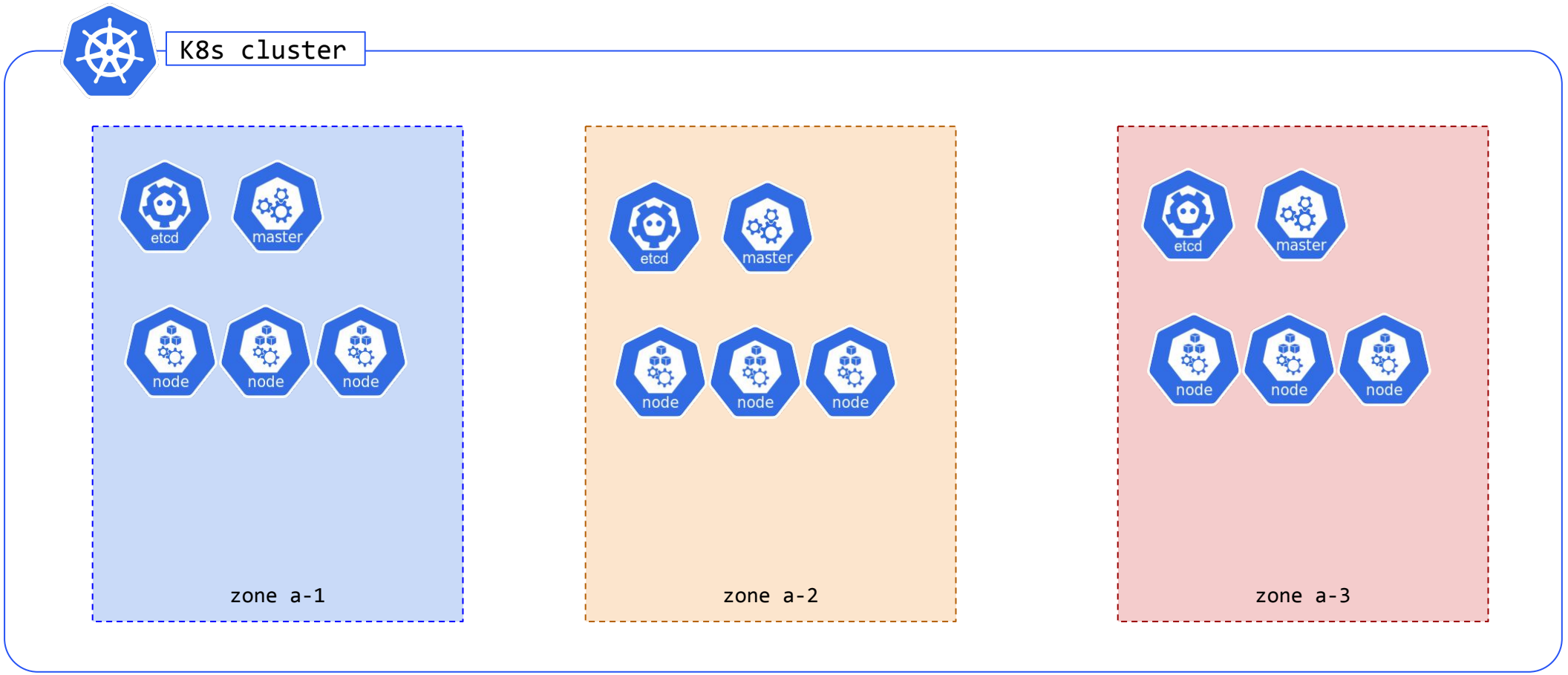
KIND MultiCluster



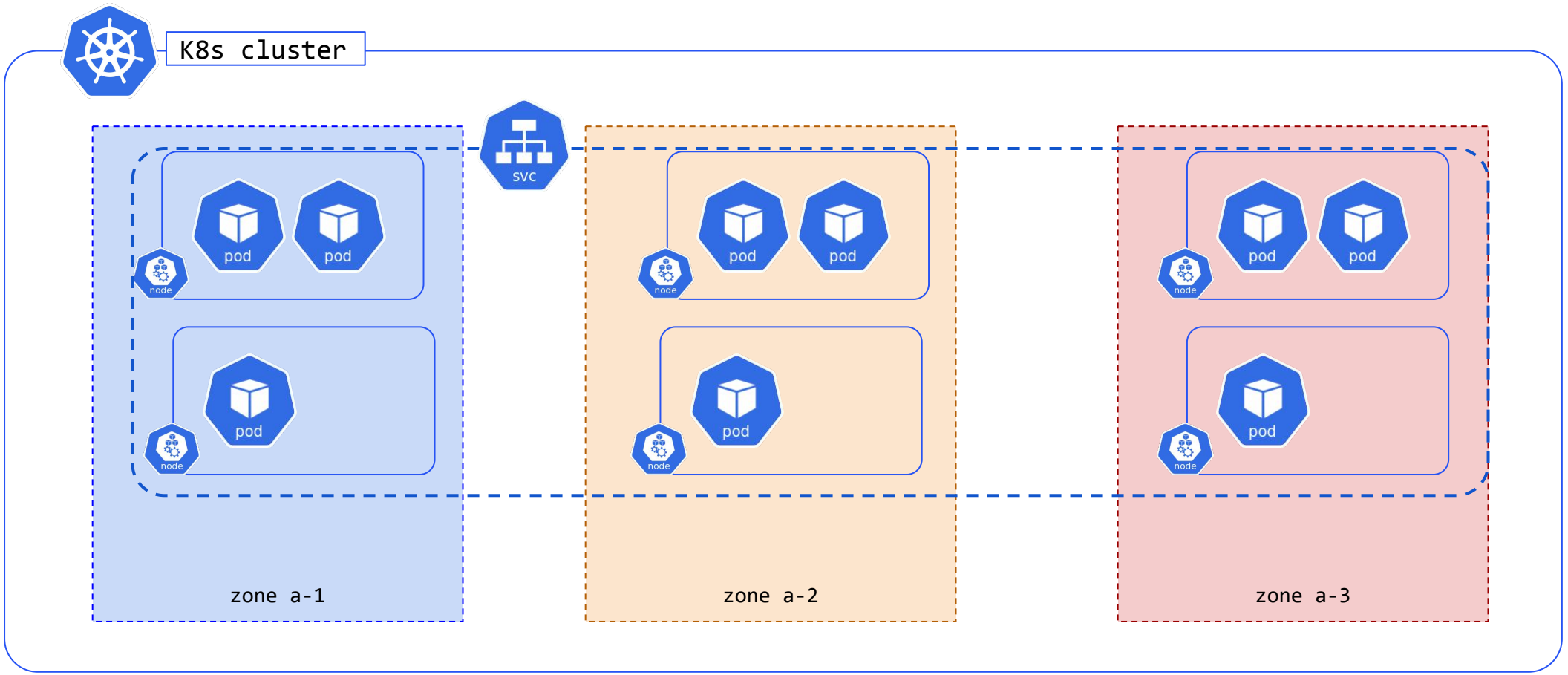
DEMO MultiCluster



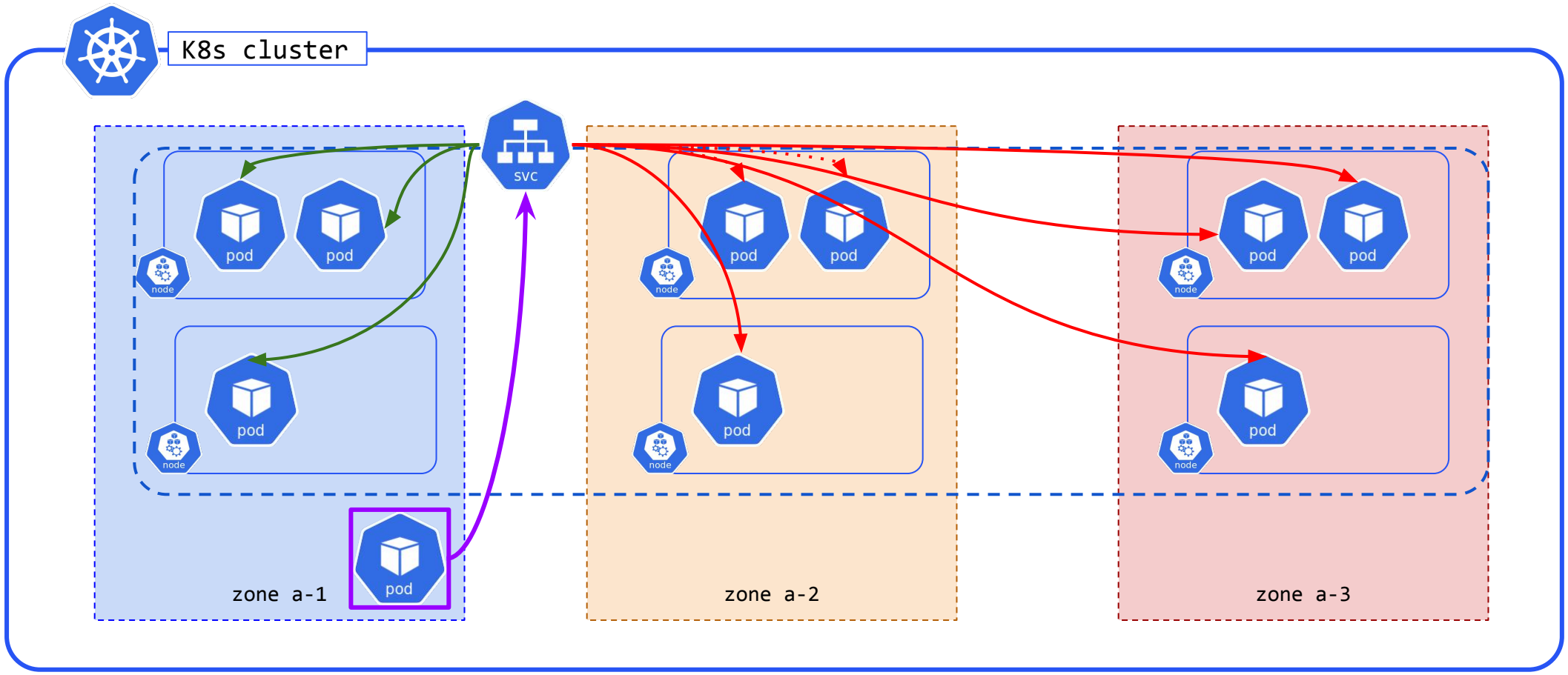
KIND MultiZone



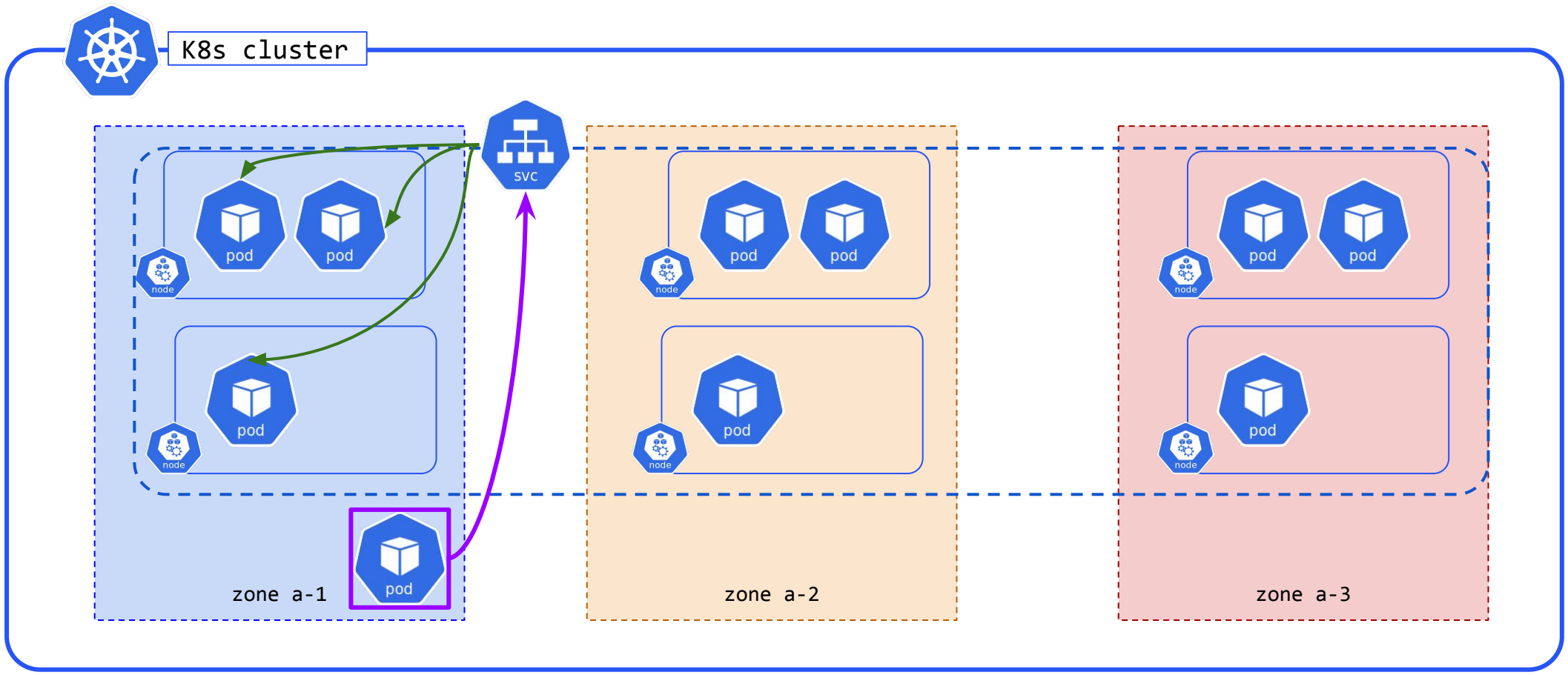
KIND MultiZone: Services



KIND MultiZone: Traffic

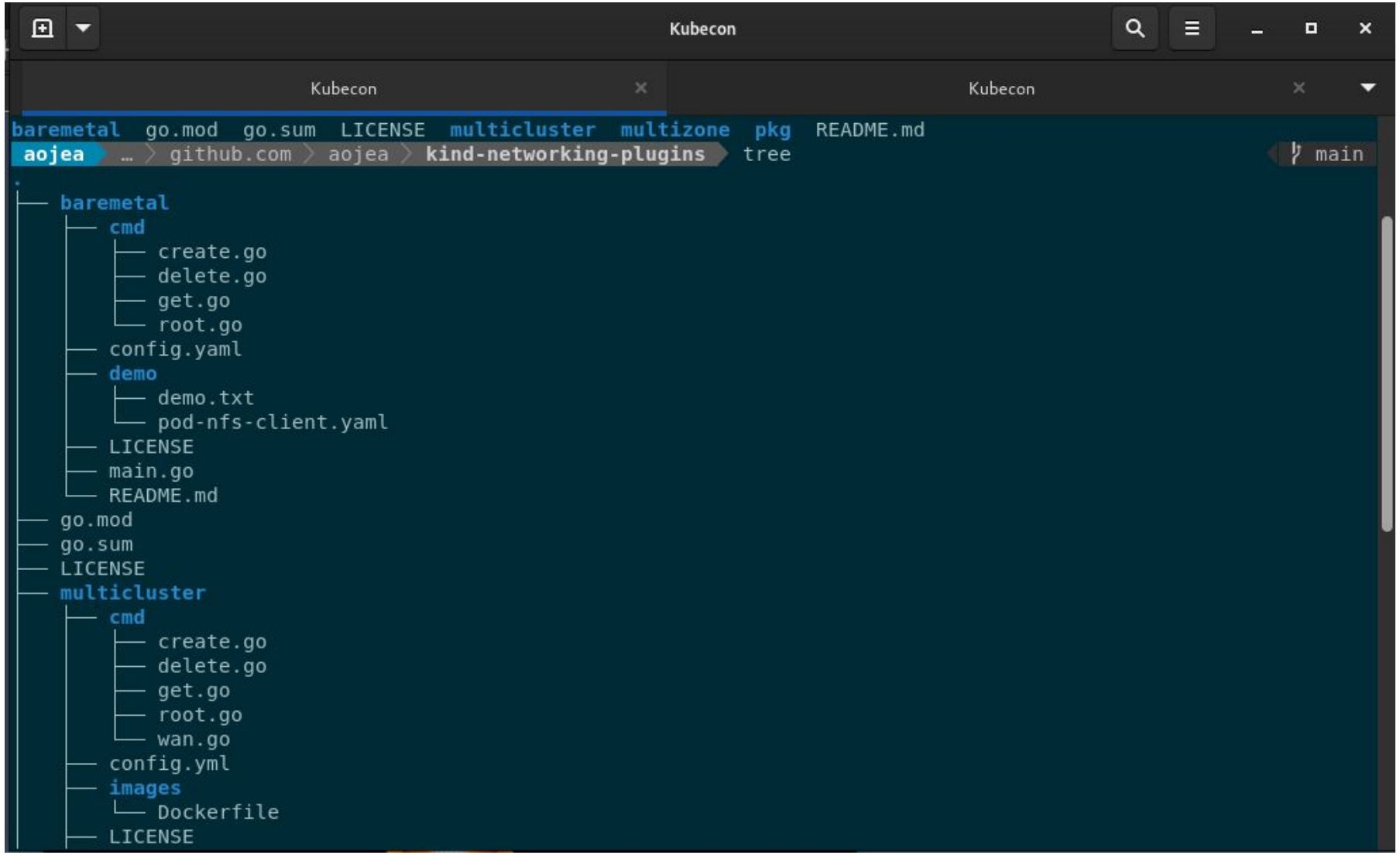


KIND MultiZone: Topology



DEMO: MultiZone

KubeCon CloudNativeCon Europe 2021 Virtual



Wrapping up

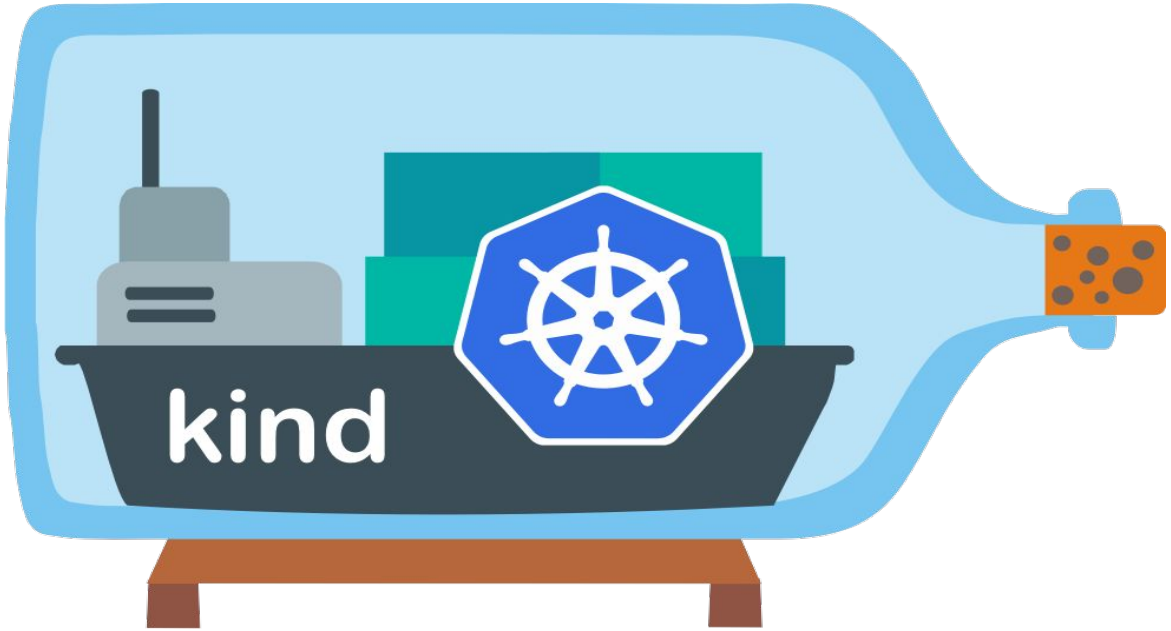
- KIND allows to emulate and test complex E2E scenarios
  - try to keep it as close as possible to the reality
- KIND is NOT a replacement for SYSTEM TESTING
  - “it works in my laptop”

Join us

- #kind, #sig-testing and #sig-network on the Kubernetes slack.k8s.io
- Repo: sigs.k8s.io/kind
- Docs: kind.sigs.k8s.io
- KIND networking plugins repo:

  https://github.com/aojea/kind-networking-plugins

Thanks



kind.sigs.k8s.io


KubeCon
CloudNativeCon
Europe 2021
Virtual
Forward Together »