bit.ly/mpuck8s

Kubernetes

An Introduction

Who am I?

michael palassis

- infrastructure architect
- ops background

mxnxpx at gmail * com

github.com/MxNxPx

Who am I?

michael palassis

- infrastructure architect
- ops background

mxnxpx at gmail * com

github.com/MxNxPx

venmo: @Michael-Palassis

Level setting

By show of hands...

- Who has heard of containers (Docker)?
- Who has heard of Kubernetes?
- Has anyone worked with Kubernetes before?

Pets vs Cattle

Pets = Legacy approach = VM or BM apps

- Require great attention
- Given actual names
- Unique and procured individually
- Sad if they die

Pets vs Cattle

Cattle = New Approach = Containers

- Look after themselves
- Don't have specific names - given numbers
- Can be simply replaced if they die
- Standardized

Containers are the atomic unit for cloud

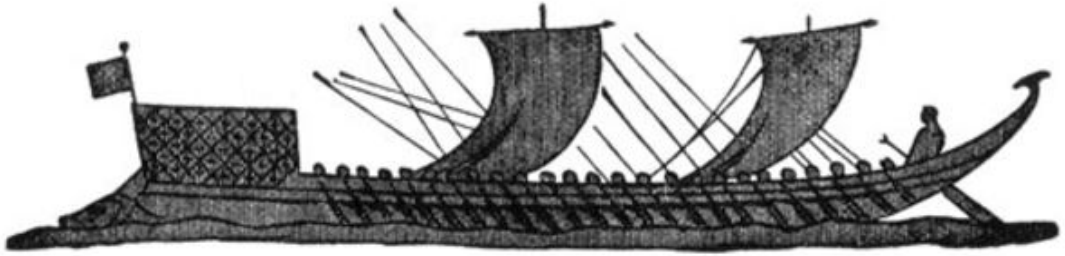
Moooooooooooo!

But what if I have a herd-load of cattle?

Kubernetes

What Does “Kubernetes” Mean?

Greek for “pilot” or
“Helmsman of a ship”



Shorthand (numeronym)

k8s

What’s the history of k8s?

- Google-grown, based on Borg and Omega, systems that run inside of Google right now and are proven to work at Google for over 10 years
- Created by three Google employees initially during the summer of 2014; grew exponentially and became the first project to get donated to the CNCF
- Hit the first production-grade version v1.0.1 in July 2015

Who uses Kubernetes?

Here are *some* of the companies using Kubernetes...

adform, adidas, AMADEUS, ancestry, 蚂蚁金服 ANT FINANCIAL, JD.COM, NAIC National Association of Insurance Commissioners, Nav, 網易 NETEASE www.163.com, The New York Times

AppDirect, babylon, BlaBlaCar, BLACKROCK, Booking.com, NOKIA, NORDSTROM, Northwestern Mutual, ocado, OpenAI

Booz | Allen | Hamilton, BOSE, box, buffer, Capital One, Pear Deck, Pearson, PingCAP, Pinterest, Nerdalize

CERN, China unicom 中国联通, Montréal, Crowdfire, DENSO, Prowise, ricardo.ch, SLAMTEC, sling TELEVISION, SOS INTERNATIONAL

GOLFNOW Go. Play., HAUFE.Group, HUAWEI, IBM, ING, Spotify, SQUARESPACE, THREDUP, VSCO, WIKIMEDIA

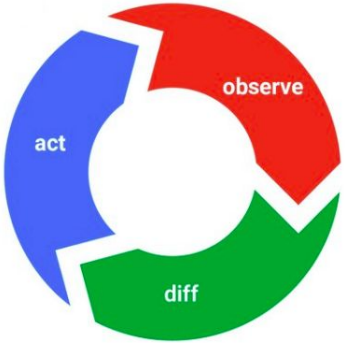
* https://kubernetes.io/case-studies/

What does Kubernetes do?

A **Container Orchestration System**

How does Kubernetes work?

- Kubernetes is a **declarative** way to **describe** your applications
- Kubernetes uses **control loops** & **reconciliation** to ensure desired state
- Everything in Kubernetes is a **resource**



What good is container orch?

- Immense and smooth scaling
  - Resource efficiency and density with isolation
  - Speed: start, create, replicate or destroy quickly
- Operational simplicity
  - Roll out updates
  - Consistency
  - Self-healing
- Improved developer productivity

Self Healing

Kubernetes will **ALWAYS** try and steer the cluster to its desired state

- **Me:** “I want 3 healthy instances of redis to always be running”
- **Kubernetes:** “Okay, I’ll ensure there are always 3 instances up and running”
- **(some time later...) Kubernetes:** “Oh look, one has died... I’m going to attempt to spin up a new one”

How to interface with k8s?

- API calls

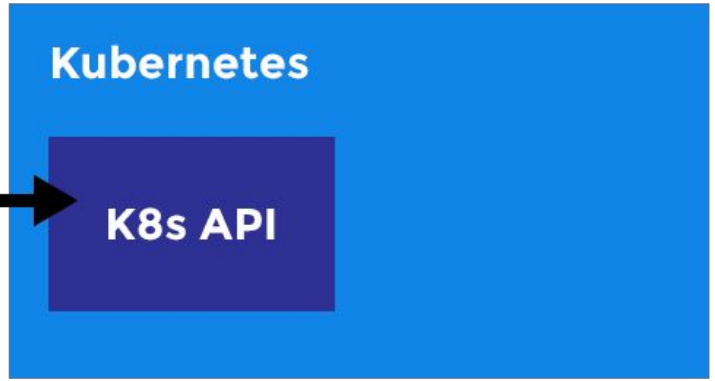
  Resources = endpoints in the Kubernetes API

- CLIs: kubectl or helm

  Makes the API calls to Kubernetes

- YAML

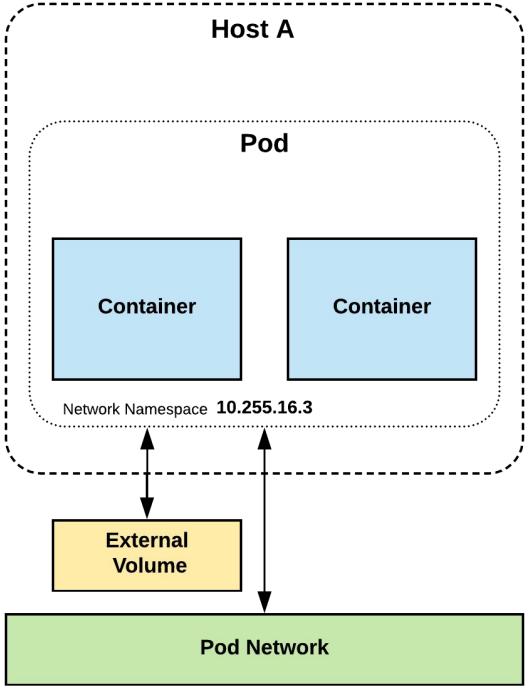
```
# What does YAML mean?
YAML:
    - Y: YAML
    - A: Ain't
    - M: Markup
    - L: Language
```



A Couple Key Concepts...

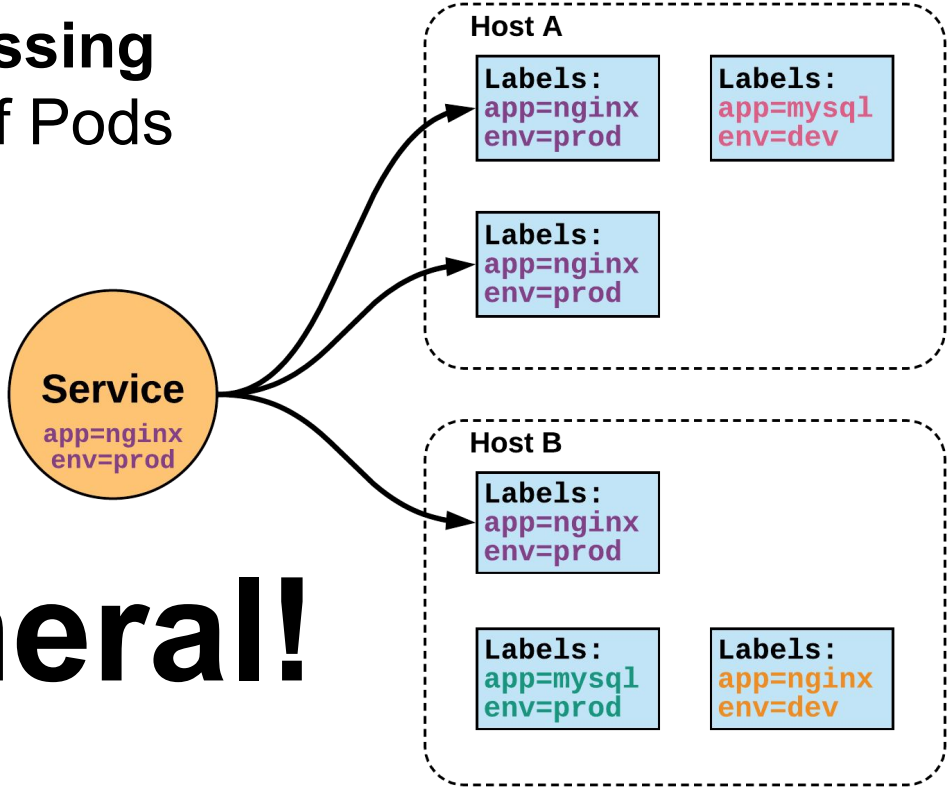
Pods

- **Atomic unit** or smallest "*unit of work*"of Kubernetes
- Pods are **one or MORE containers** that share volumes and namespace
- **They are ephemeral!**



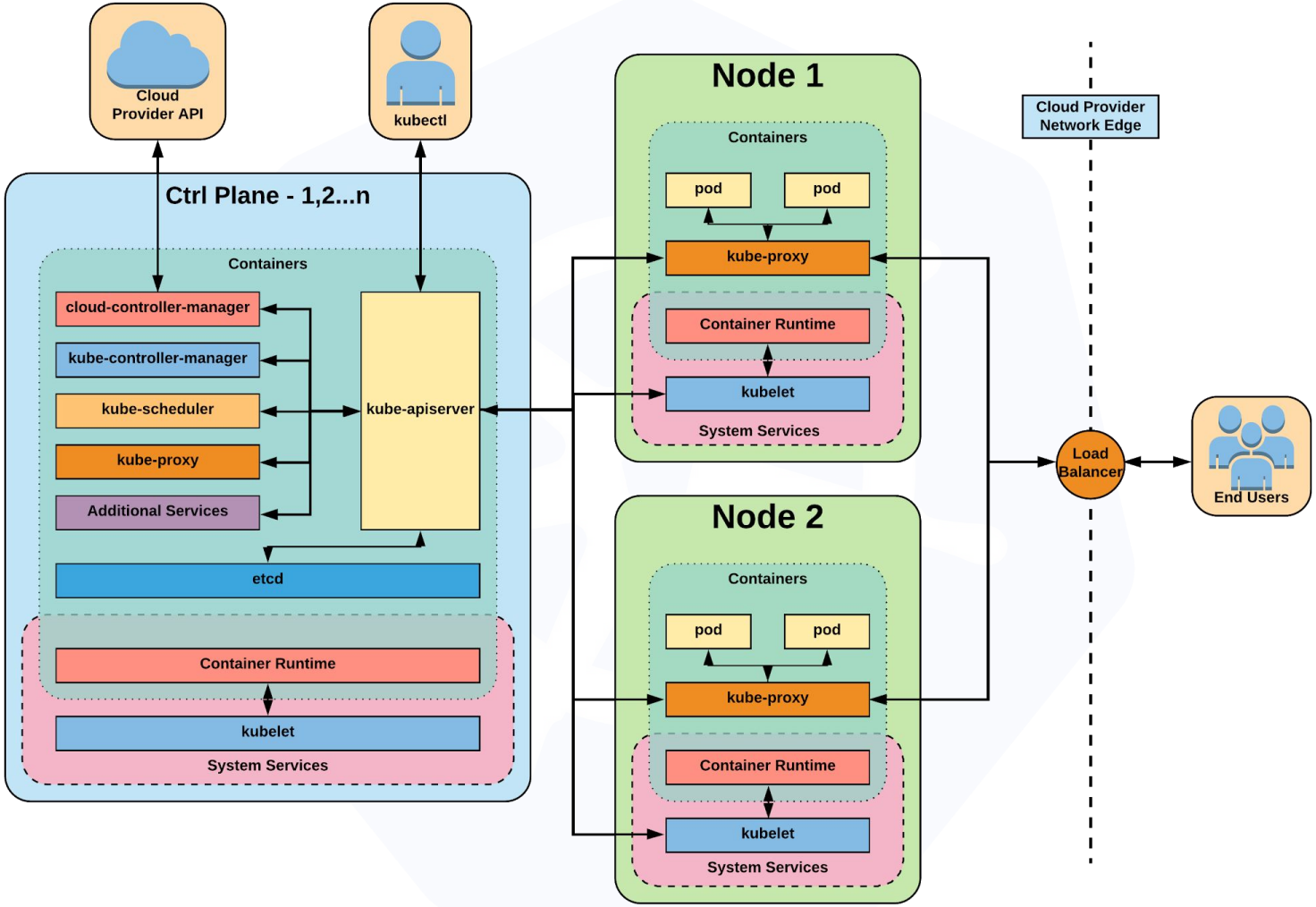
Services

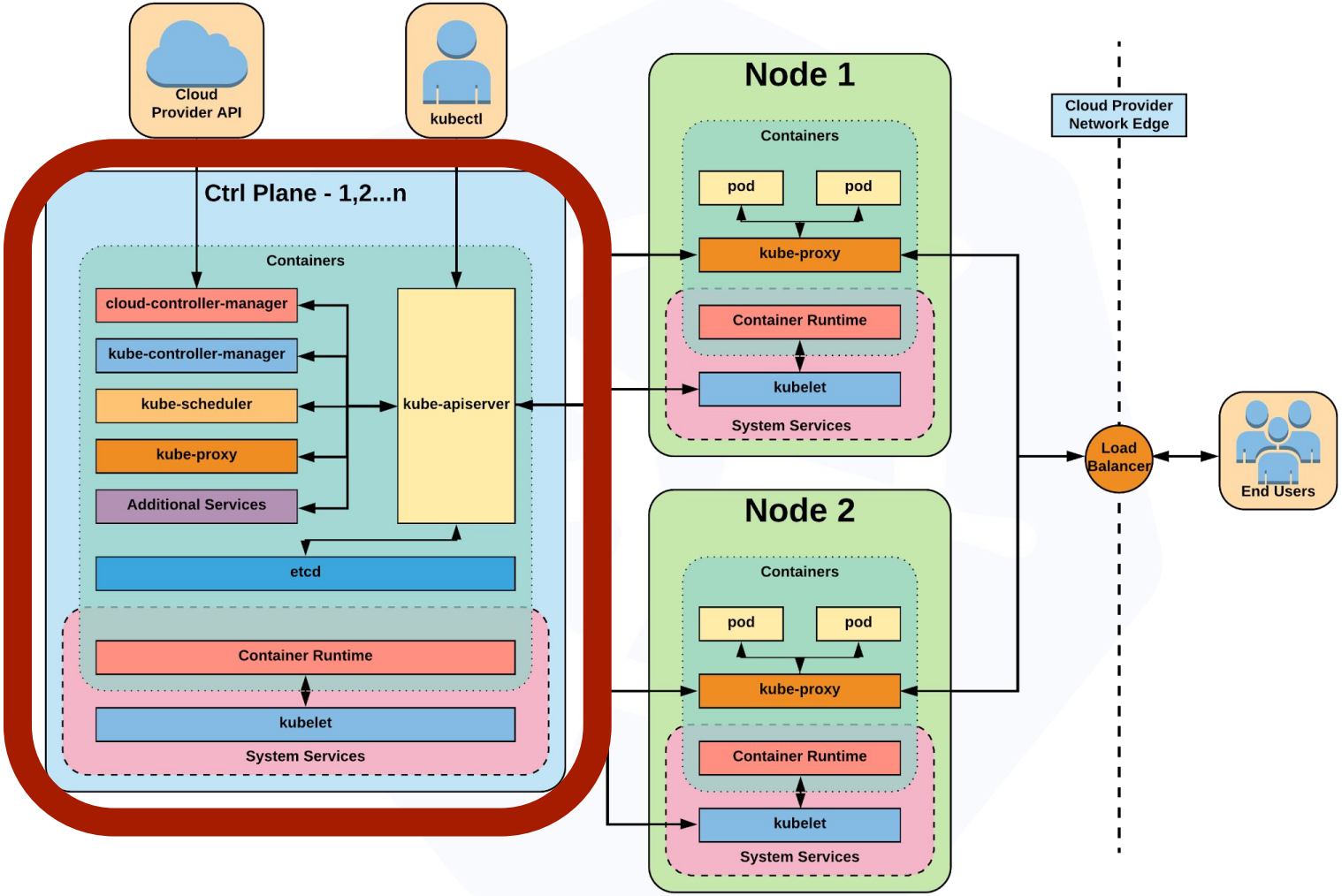
- **Unified method of accessing** the exposed workloads of Pods
- **Durable resource**
  - static cluster IP
  - static namespaced DNS name



NOT Ephemeral!

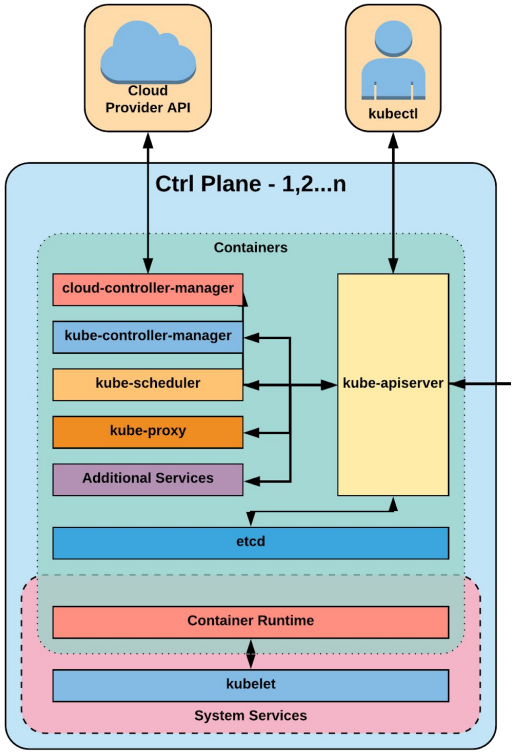
Kubernetes Cluster Architecture Overview

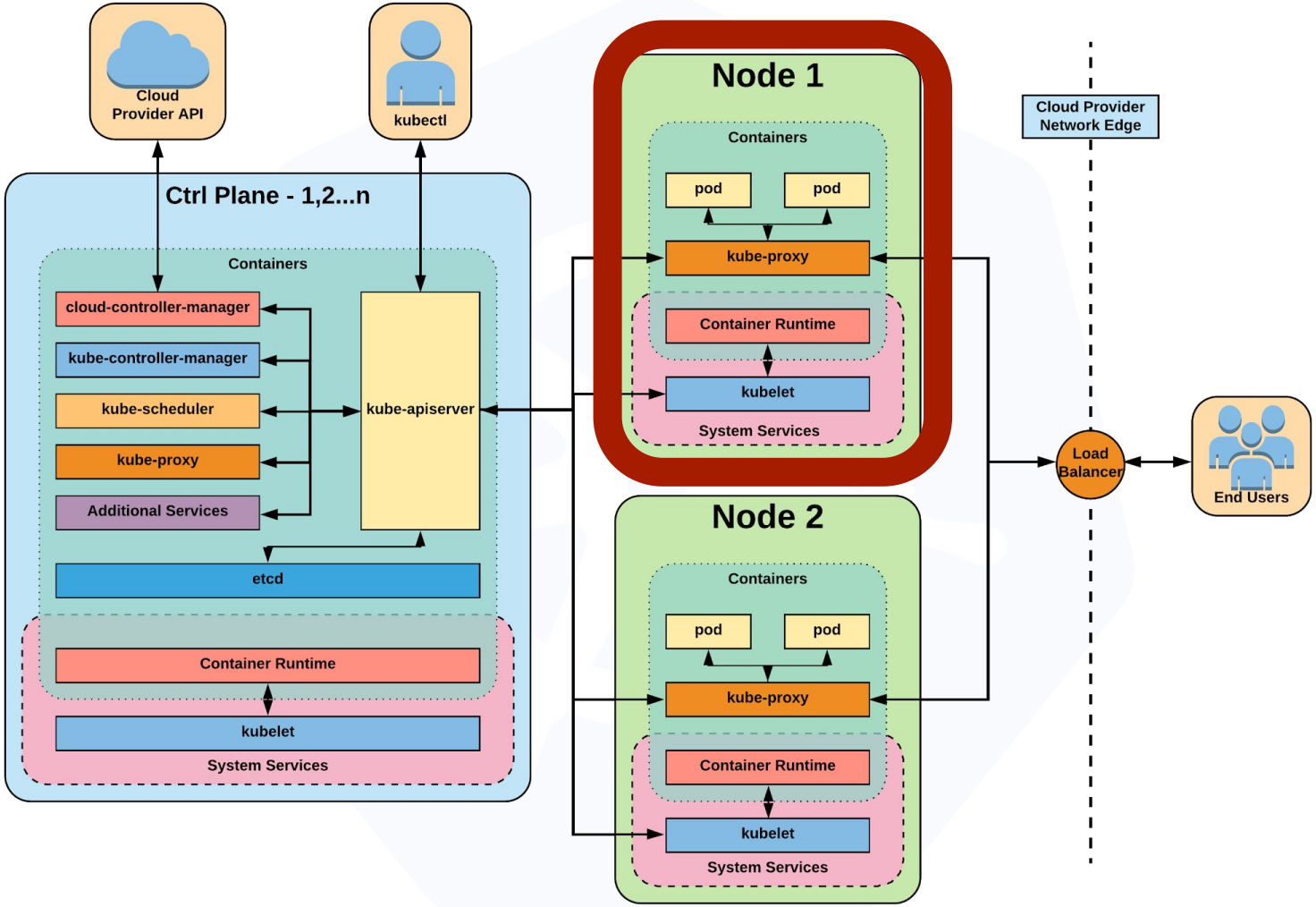

Cloud Provider API
kubectl
Ctrl Plane - 1,2...n
Containers
cloud-controller-manager
kube-controller-manager
kube-scheduler
kube-proxy
Additional Services
kube-apiserver
etcd
Container Runtime
kubelet
System Services
Node 1
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Node 2
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users


Cloud Provider API
kubectl
Ctrl Plane - 1,2...n
Containers
cloud-controller-manager
kube-controller-manager
kube-scheduler
kube-proxy
Additional Services
kube-apiserver
etcd
Container Runtime
kubelet
System Services
Node 1
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Node 2
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users

Control Plane Components

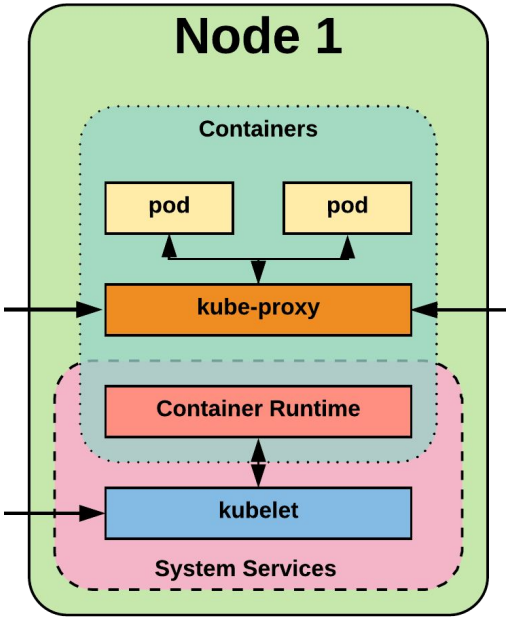
- kube-apiserver
- etcd
- kube-controller-manager
- kube-scheduler
- cloud-controller-manager

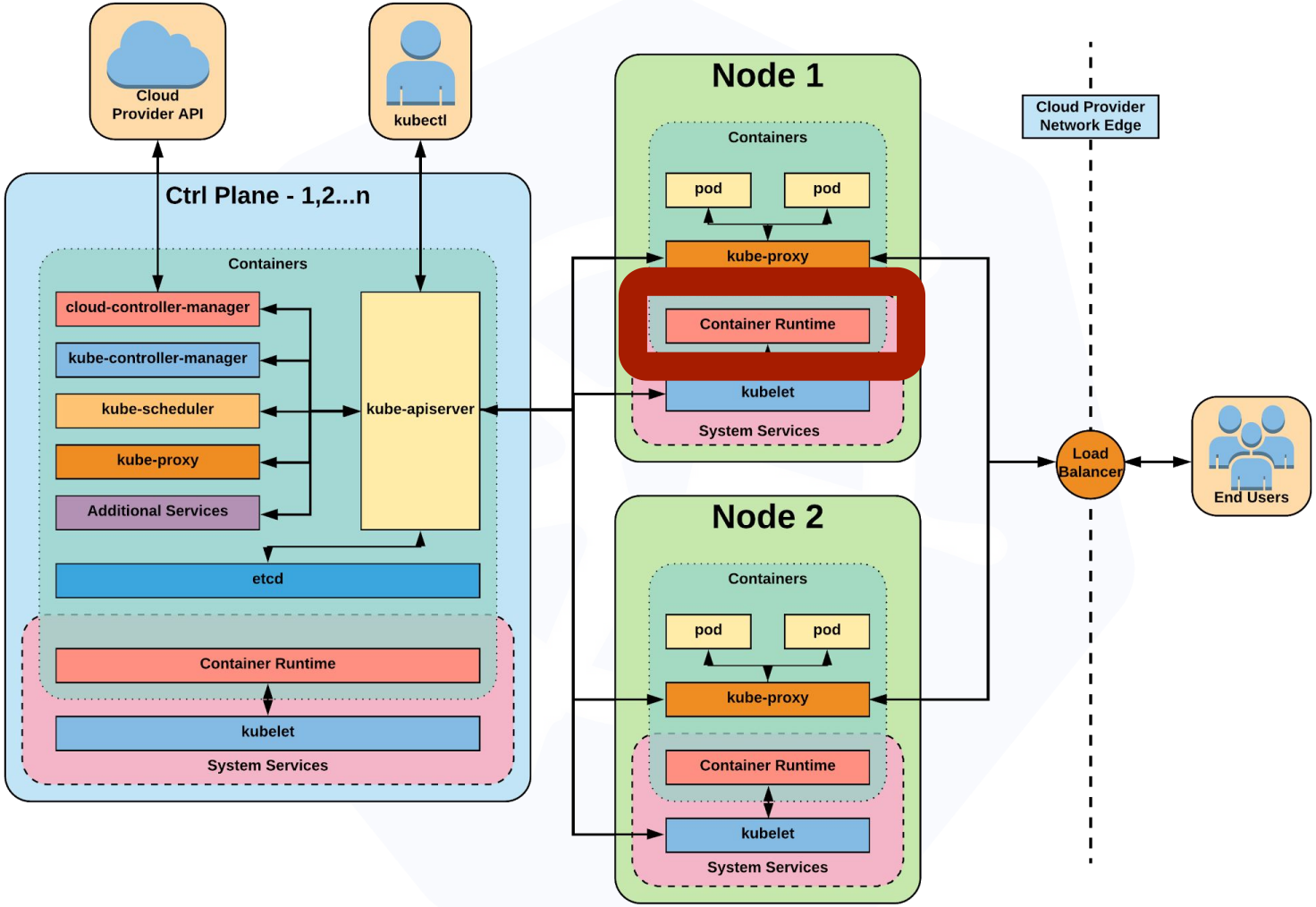



Cloud Provider API
kubectl
Ctrl Plane - 1,2...n
Containers
cloud-controller-manager
kube-controller-manager
kube-scheduler
kube-proxy
Additional Services
kube-apiserver
etcd
Container Runtime
kubelet
System Services
Node 1
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Node 2
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users

Node Components

- kubelet
- kube-proxy
- Container Runtime Engine




Cloud Provider API
kubectl
Ctrl Plane - 1,2...n
Containers
cloud-controller-manager
kube-controller-manager
kube-scheduler
kube-proxy
Additional Services
kube-apiserver
etcd
Container Runtime
kubelet
System Services
Node 1
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Node 2
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users

Container Runtime Engine

- A container runtime is a CRI (Container Runtime Interface) compatible application that executes and manages containers.
  - Containerd (docker)
  - Cri-o
  - Rkt
  - Kata (formerly clear and hyper)
  - Virtlet (VM CRI compatible runtime)

Core Resources

- Namespaces
- Pods
- Labels
- Deployments

Namespaces

Namespaces are a logical cluster or environment, and are the primary method of partitioning a cluster or scoping access

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: prod
  labels:
    app: MyBigWebApp
```

```
$ kubectl get ns --show-labels
NAME          STATUS    AGE       LABELS
default       Active    11h       <none>
kube-public   Active    11h       <none>
kube-system   Active    11h       <none>
prod          Active    6s        app=MyBigWebApp
```

Pod Example

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-example
spec:
  containers:
  - name: nginx
    image: docker.io/nginx:stable-alpine
    ports:
    - containerPort: 80
```

Key Pod Container Attributes

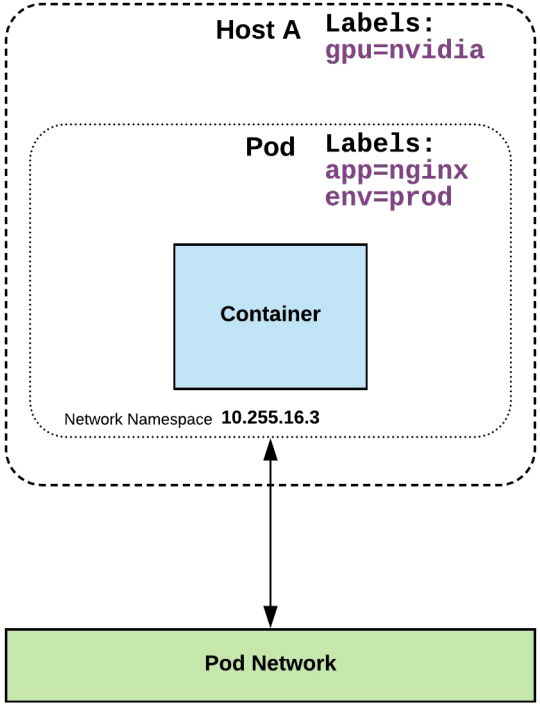
- `name` - The name of the container
- `image` - The container image
- `ports` - array of ports to expose -- can be granted a friendly name and protocol may be specified
- `env` - array of environment variables
- `command` – Entrypoint array (equiv to Docker `ENTRYPOINT`)
- `args` - Arguments to pass to the command (equiv to Docker `CMD`)

Container

```
name: nginx
image: nginx:stable-alpine
ports:
  - containerPort: 80
    name: http
    protocol: TCP
env:
  - name: MYVAR
    value: isAwesome
command: ["/bin/sh", "-c"]
args: ["echo ${MYVAR}"]
```

Labels

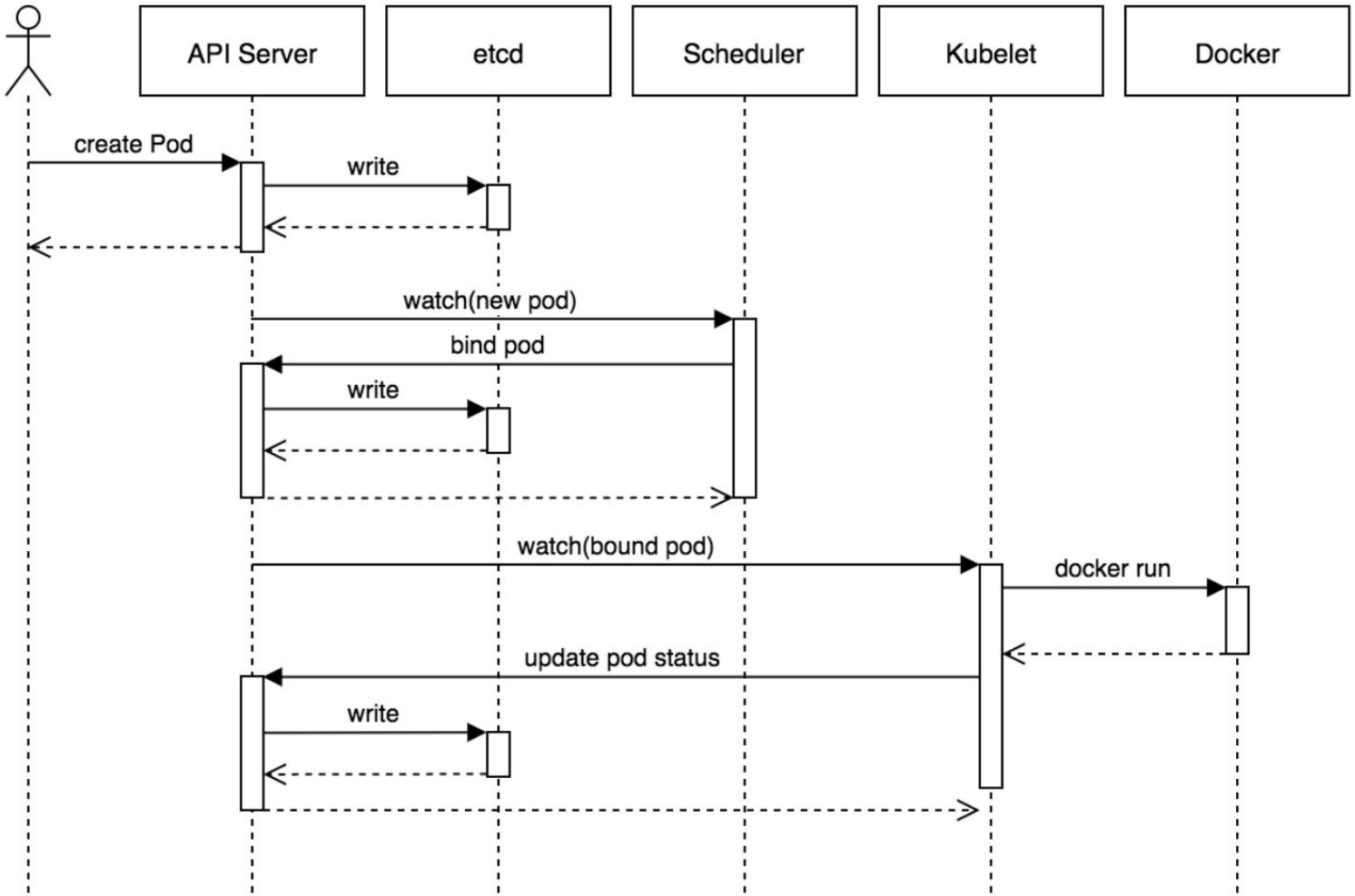
- key-value pairs that are used to identify, describe and group together related sets of objects or resources
- Have a strict syntax with a slightly limited character set*



* https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#syntax-and-character-set

Deployment

- Way of managing Pods via **ReplicaSets**
- Provide rollback functionality and update control
- Updates are managed through the **pod-template-hash** label
- Each iteration creates a unique label that is assigned to both the **ReplicaSet** and subsequent Pods

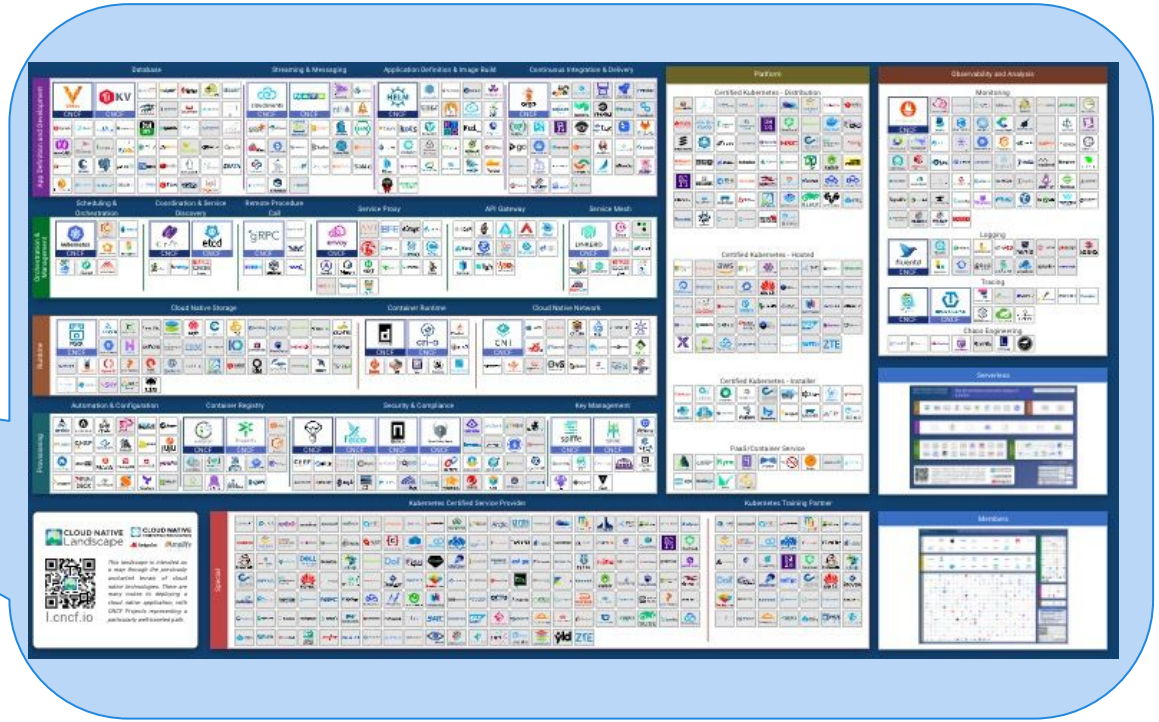

API Server
etcd
Scheduler
Kubelet
Docker
create Pod
write
watch(new pod)
bind pod
write
watch(bound pod)
docker run
update pod status
write

Live demo

Play along in cloud env: bit.ly/mpuck8sd (requires github account)
Code: github.com/MxNxPx/kind-kubethanos

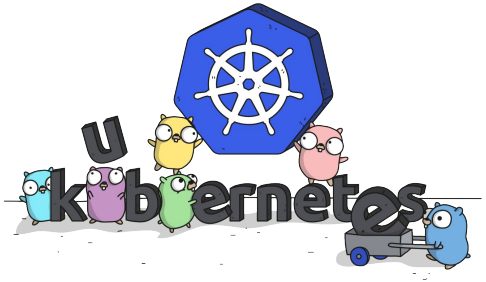
Final thoughts

- Kubernetes is great, but beware...



Links

- **Free Kubernetes Courses**
  https://kube.academy/ & https://kubernetes.io/training/
- **Interactive Kubernetes Tutorials**
  https://www.katacoda.com/courses/kubernetes
- **Learn Kubernetes the Hard Way**
  https://github.com/kelseyhightower/kubernetes-the-hard-way
- **Official Kubernetes Youtube Channel**
  https://www.youtube.com/c/KubernetesCommunity
- **Official CNCF Youtube Channel**
  https://www.youtube.com/c/cloudnativefdn
- **Track to becoming a CKA/CKAD (Certified Kubernetes Administrator/Application Developer)**
  https://www.cncf.io/certification/expert/
- **Awesome Kubernetes**
  https://ramitsurana.github.io/awesome-kubernetes/



Any Questions?