# DevOps - Provisioning - K8s - Lab

## Hands on practice with Kubernetes

Samir Teymurov

Mental Arts

22 February 2024

# Agenda

- Kubernetes Recap: 50 min
- Kubernetes Lab Part 1: 50 min
- Kubernetes Lab Part 2: 50 min

# TL; DR

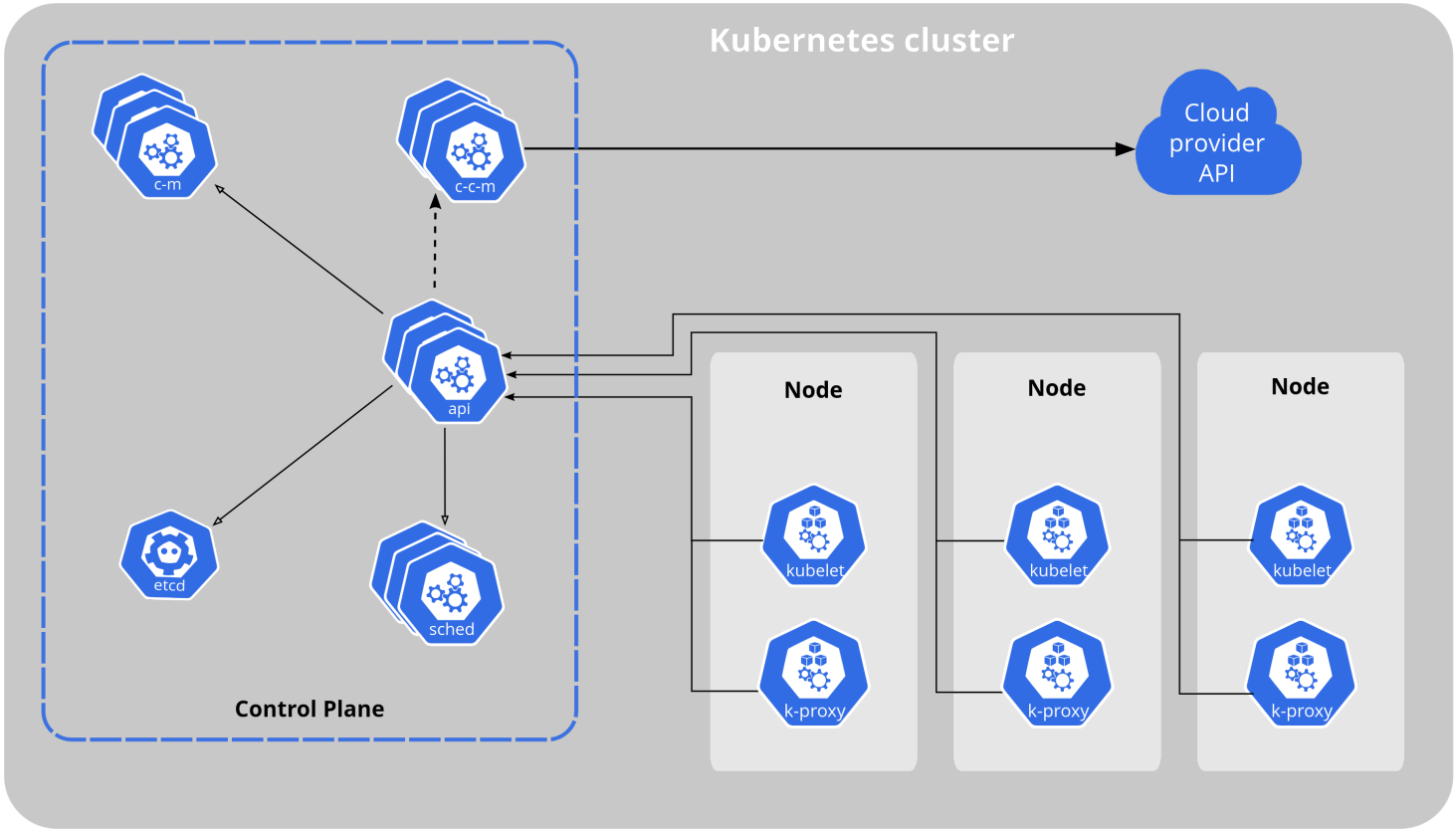
- Kubernetes Recap
- Kubernetes Lab
  - KinD Installation
  - Deploying FastAPI Application in Kubernetes
  - Homework Assignment
  - Q&A
- Homework Assignment
- Q&A
- Thank you!
- References

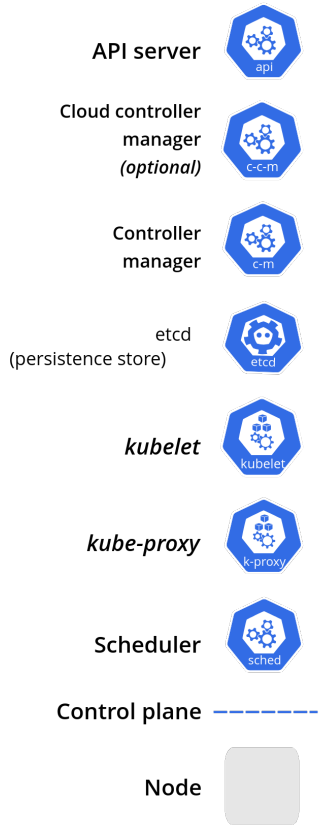
# Kubernetes Definition

At the highest level Kubernetes is two things:

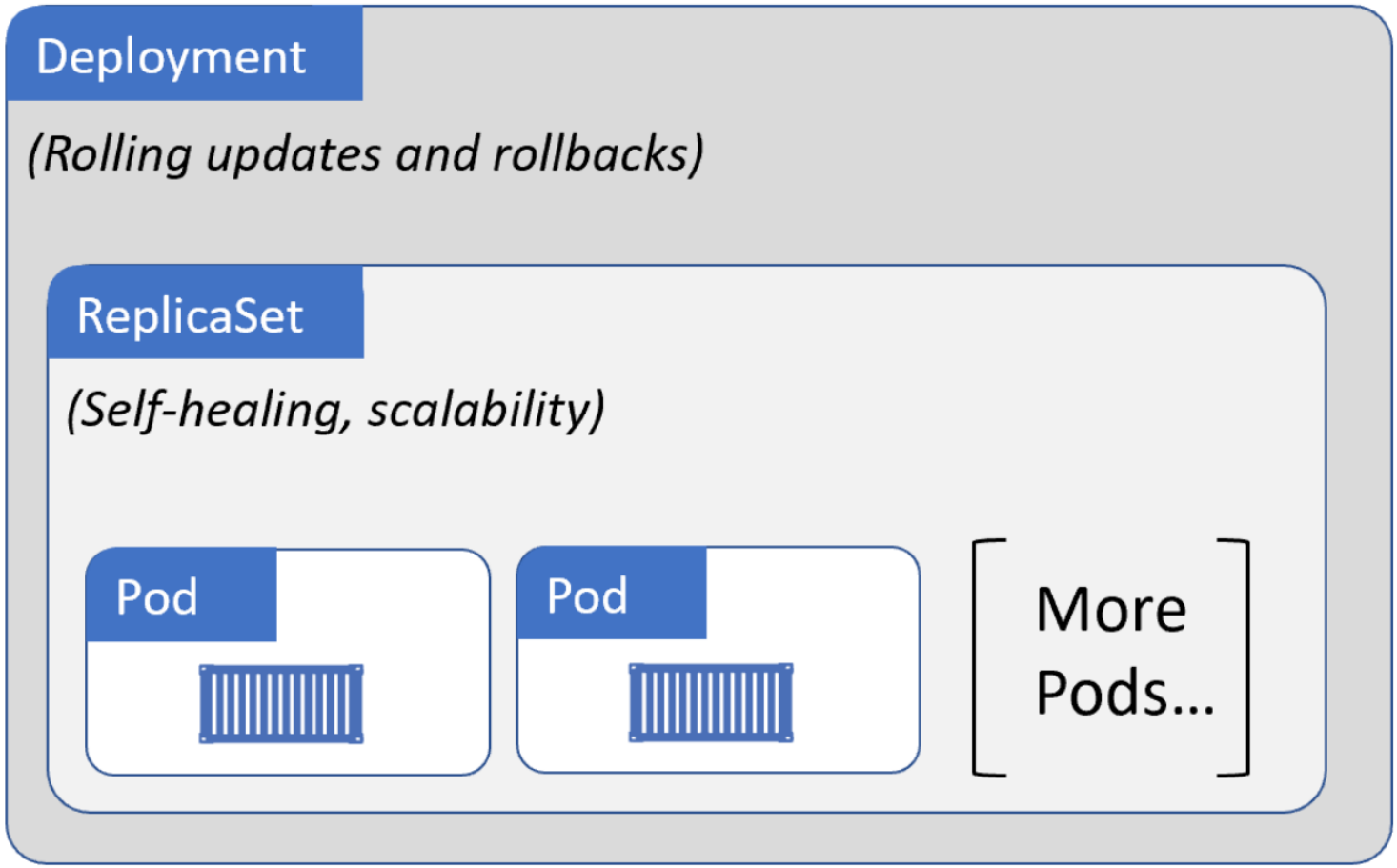
- A cluster to run applications on
- An orchestrator of cloud-native microservices apps

# Kubernetes Components



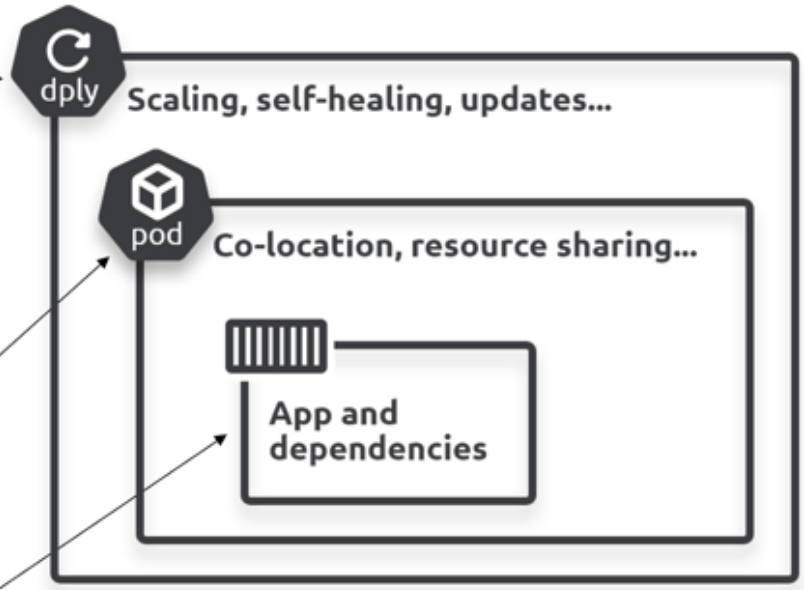



# Deployment Theory



# Deployment Theory

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-deploy
spec:
  replicas: 10
  selector:
    matchLabels:
      app: hello-world
  minReadySeconds: 10
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
      - name: hello-pod
        image: nigelpoulton/k8sbook:1.0
        ports:
        - containerPort: 8080
```



# Kubernetes Services

There are three primary types of Kubernetes services:

- ClusterIP
- NodePort
- LoadBalancer

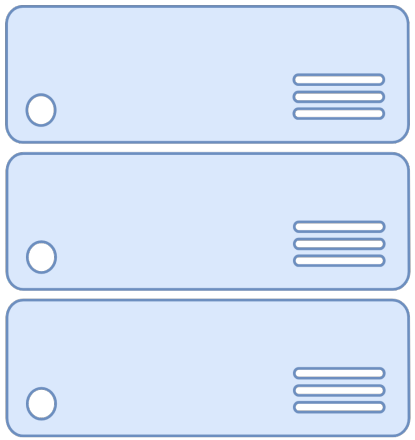
The Ingress resource is not a type of Service, although it serves a somewhat similar purpose. It will be described later.

# Kubernetes Services

| | ClusterIP Service | NodePort Service | LoadBalancer Service | Ingress + Service |
|---|---|---|---|---|
| Native K8s Resource | Yes | Yes | Yes, but needs cloud provider load balancer | Yes, but needs ingress controller deployed in cluster |
| Protocol (OSI Layer) | layer 4 | layer 4 | layer 4 and below* | layer 7 - http and https only |
| Allows multiple services per IP | No | No | Yes, but not same port** | Yes |
| Can expose outside the cluster | No | Yes | Yes (1 service) | Yes (multiple services) |

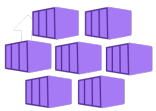
# Getting Kubernetes

- Bare MetaL: Anywhere you've got a server
- Cloud: AWS, GCP, Azure, DigitalOcean, etc.
- Local Development: Kind, Minikube, MicroK8s, K3D, etc.

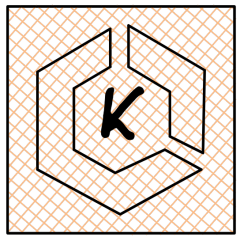


Server Rack

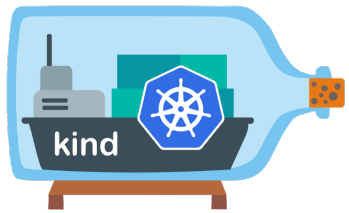
Google Kubernetes Engine



Azure Kubernetes Services



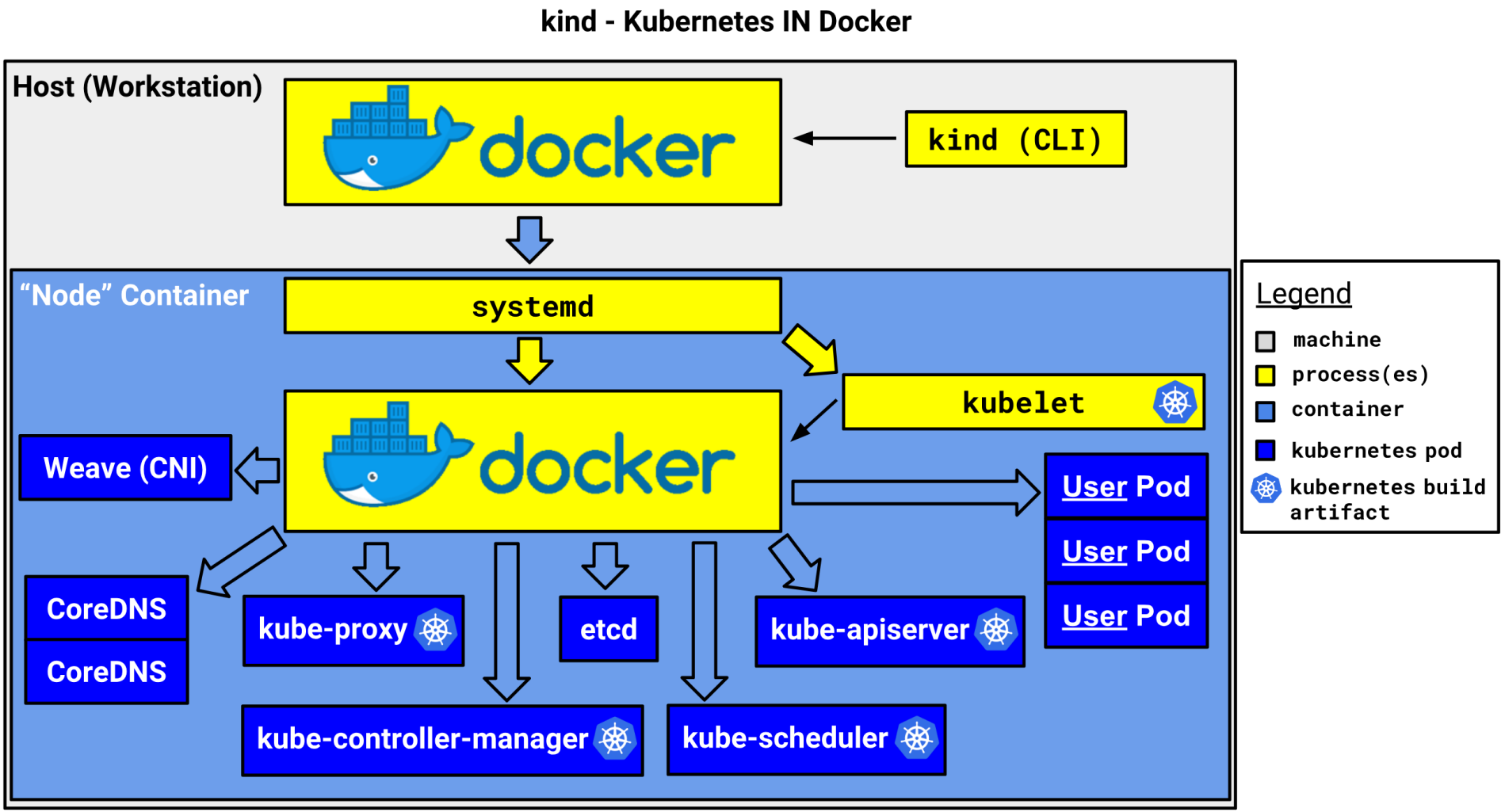
AWS EKS





minikube

# KinD Architecture

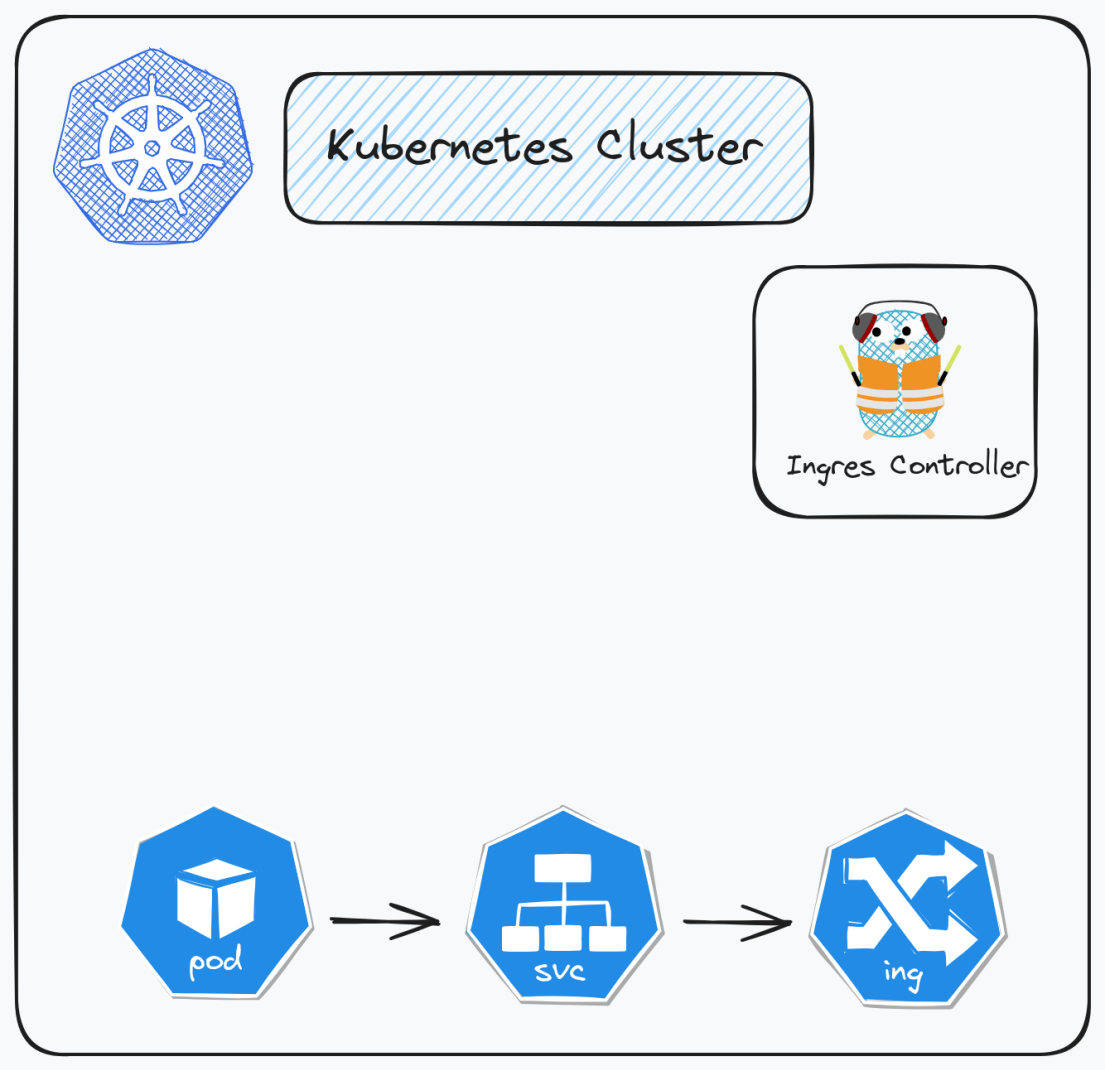


# KinD Installation

- KinD Documentation
- Installing with a Package Manager
- Installing from Release Binaries
- Installing from Source

Please install KinD on your local machine, and let me know once it's done.

# Deploying FastAPI Application in Kubernetes



# Service Example Yaml

```yaml
kind: Service
apiVersion: v1

metadata:
  name: hostname-service

spec:
  type: NodePort
  selector:
    app: echo-hostname
  ports:
    - nodePort: 30163
      port: 8080
      targetPort: 80
```

Make the service available to network requests from external clients

Forward requests to pods with label of this value

**nodePort**
access service via this external port number

**port**
port number exposed internally in cluster

**targetPort**
port that containers are listening on

# Homework Assignment

- Project Assignment
- Project Work

# Thank you!

- Thanks you for listening!
- You are now 1 step closer to becoming a Kubernetes expert!

# References


- Kubernetes Documentation
- KinD Documentation
- FastAPI Documentation
- Traefik Documentation
- Kubernetes Service vs LoadBalancer vs Ingress
- The Kubernetes Book - Nigel Poulton
- TechWorld with Nana
- Minikube vs K3D vs KinD vs GetDeck Beiboot