smartship

A data analytics, decision support and circular economy – based multi-layer optimization platform towards a holistic energy efficiency, fuel consumption and emissions management of vessels

Training meeting
Kubernetes for Developers
Y.X.2021

Presenter: Jakub Rola BLS

The SmartShip project has received funding from the European Union's
Horizon 2020 research and Innovation programme under the
Marie Skłodowska-Curie grant agreement No 823916

About me

- Software Developer
  - JAVA Spring Boot
  - Angular
  - Flutter
- Working in BlueSoft
  - R&D departmant
  - RISE and RIA projects group

Overview

- History of Kubernetes:
  - Monolithic approach
  - Virtual machines
  - Containerization
- Overview
  - Kubernetes architecture
  - Kubernetes components
- Networking
  - Internal network
  - Exposing to external networks
  - Work balancing
- Application deployment
  - YAML file format
  - 3 parts of deployment file

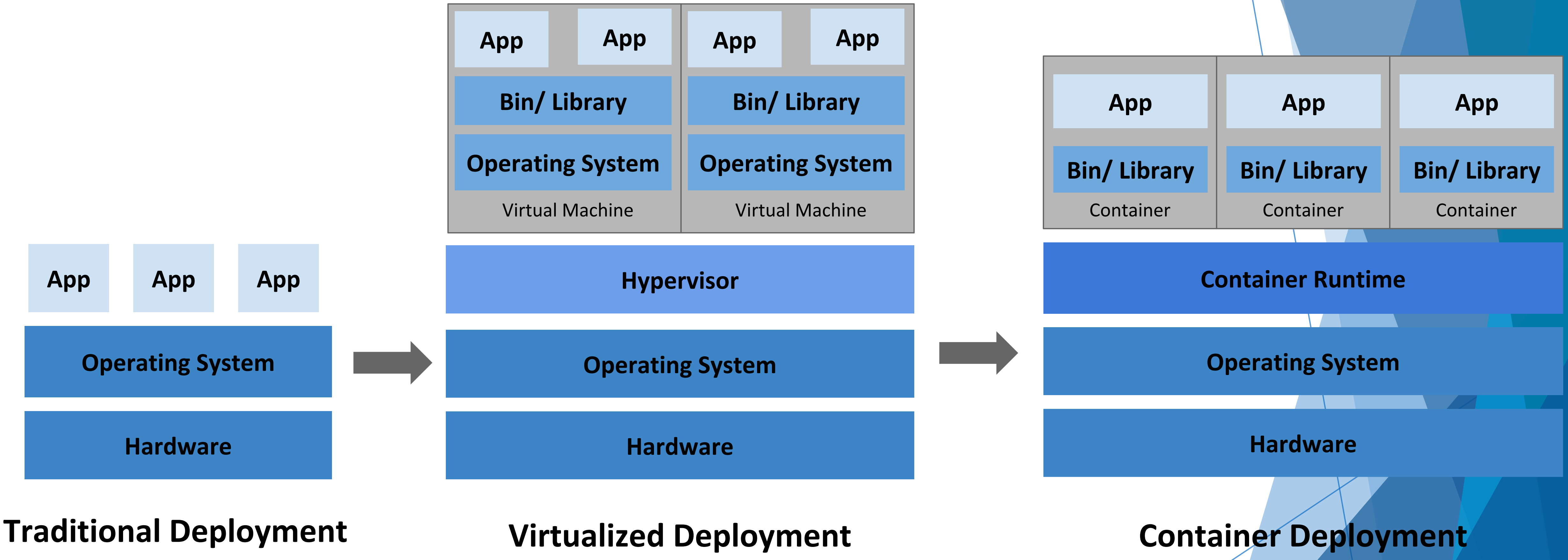
Introduction

Short animated film about Kubernetes :
https://www.youtube.com/watch?v=4ht22ReBjno

Agenda

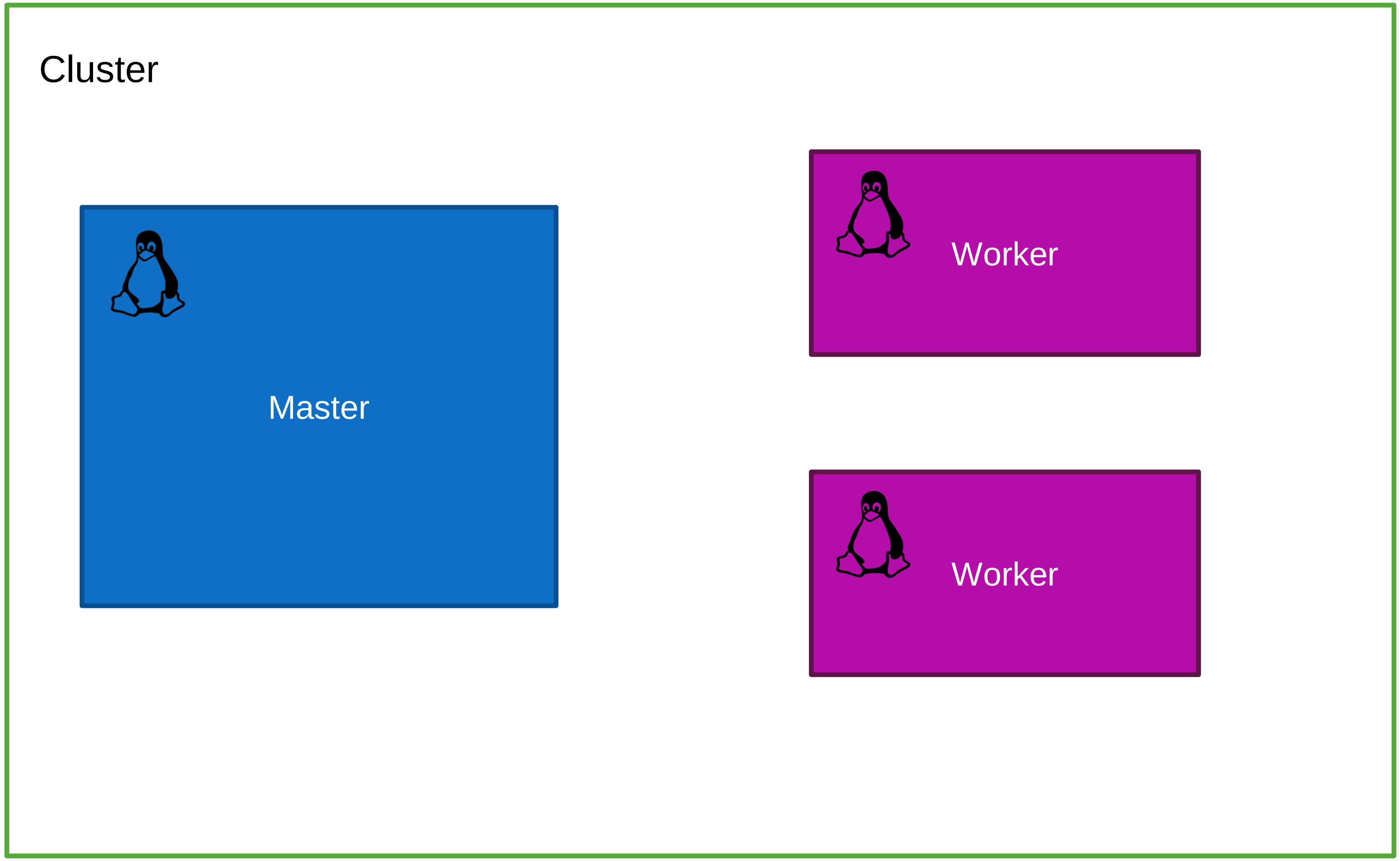
- History o Kubernetes and why it was created ?
- Architecture of k8s components and its stateless nature
- Networking
- YAML files as configuration standard
- Deployment of application

Origin of Kubernetes

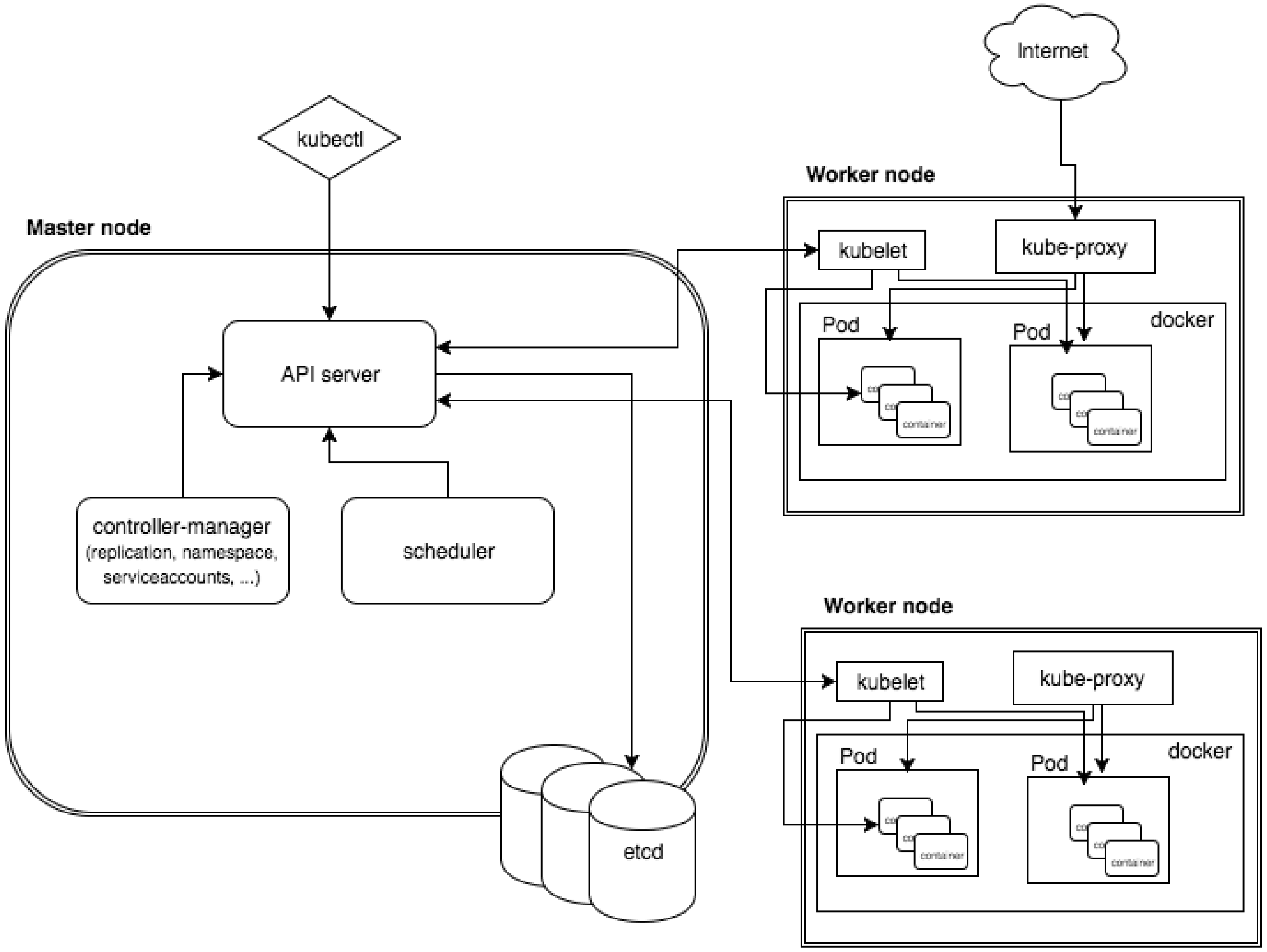


SmartShip - https://kubernetes.io/docs/home/

High-level architecture

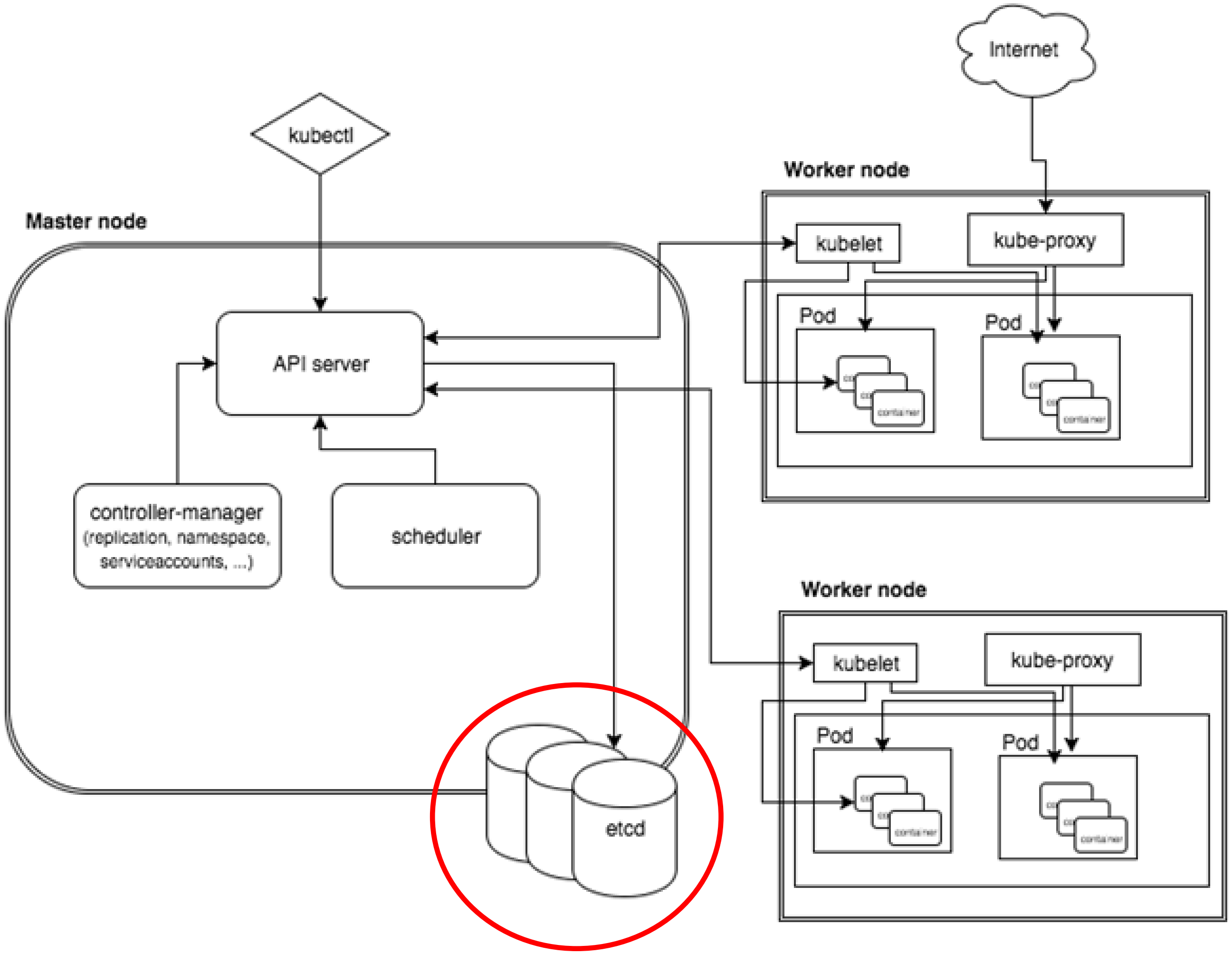


Architecture



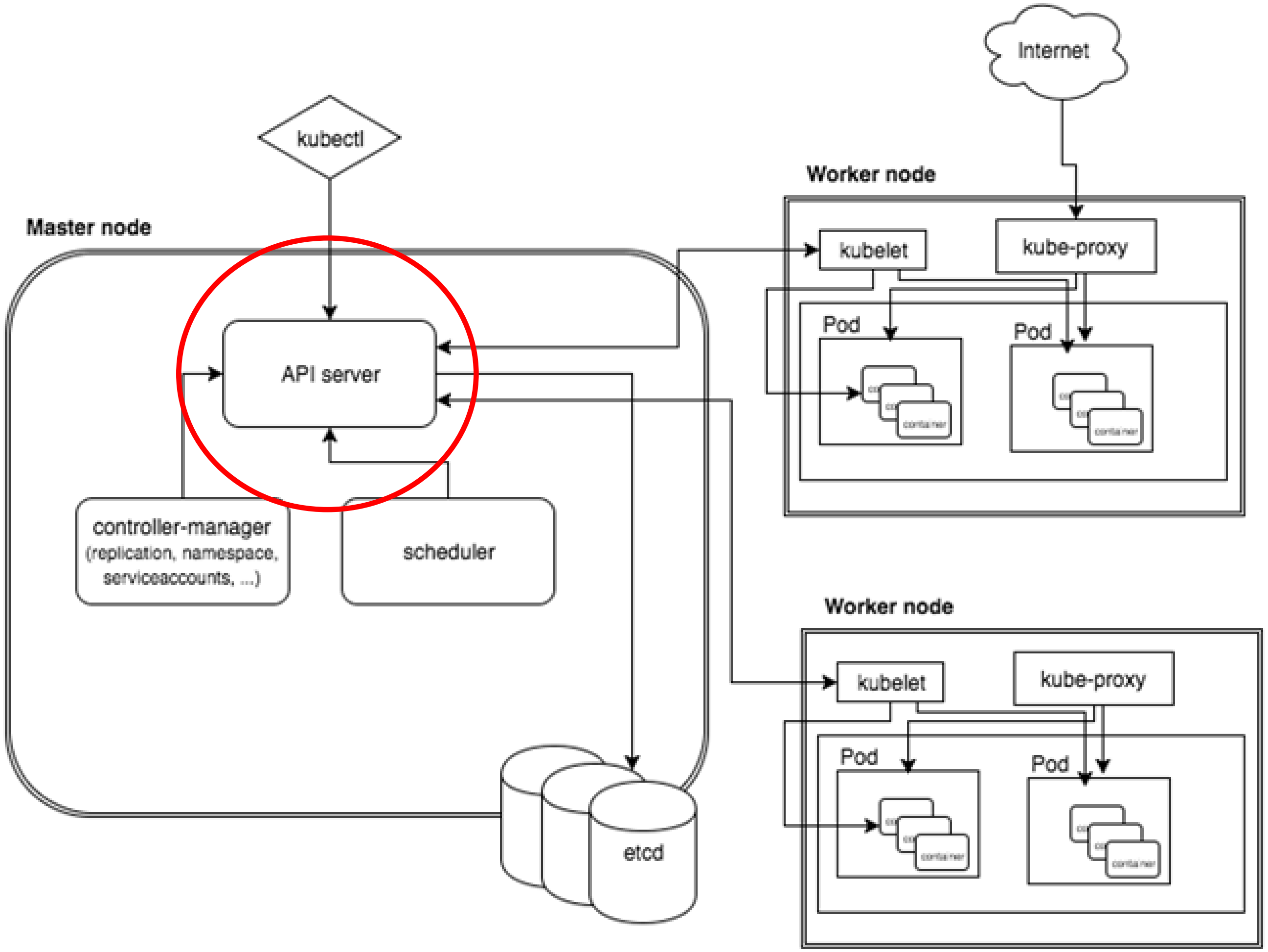
https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



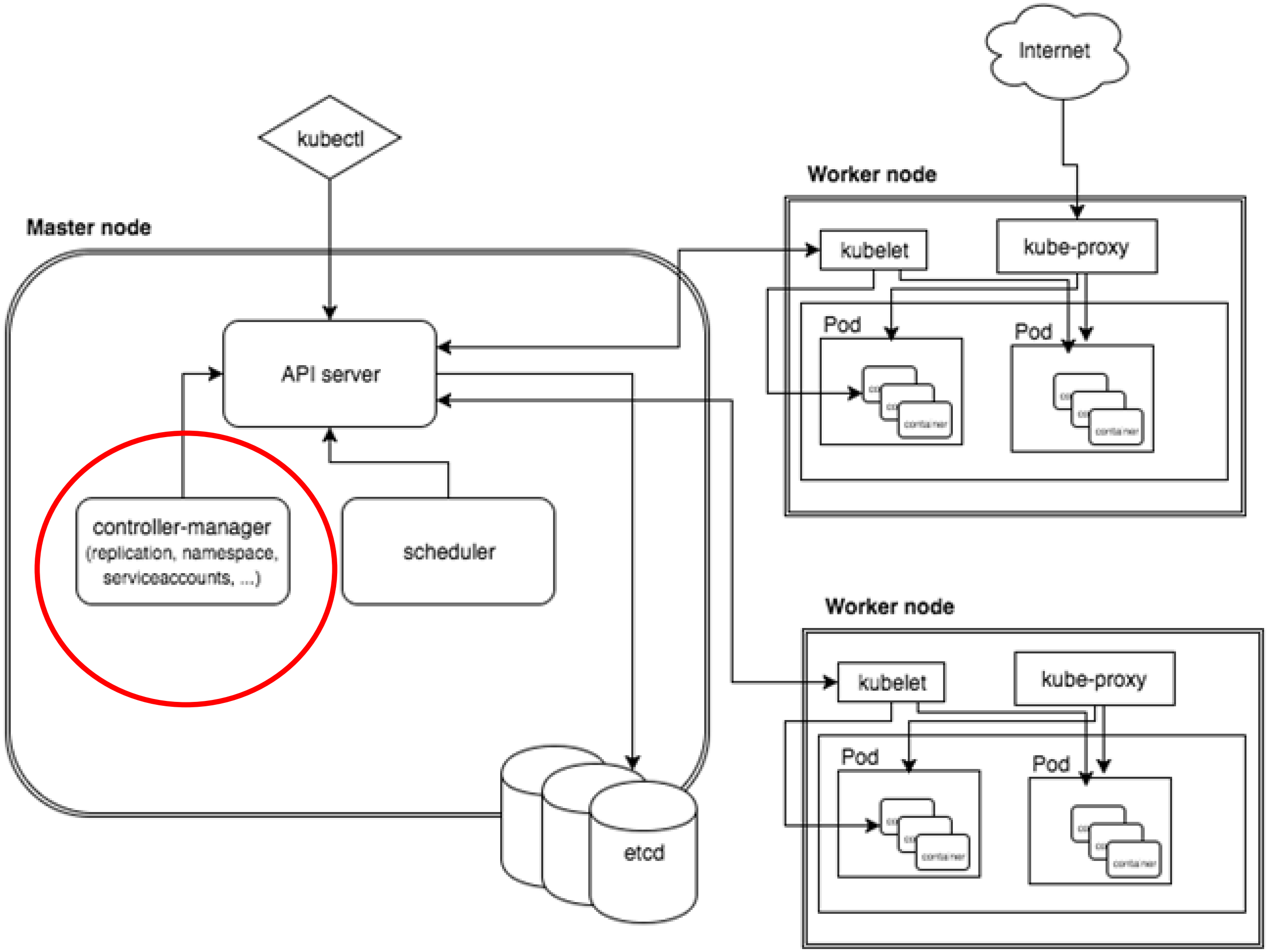
https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



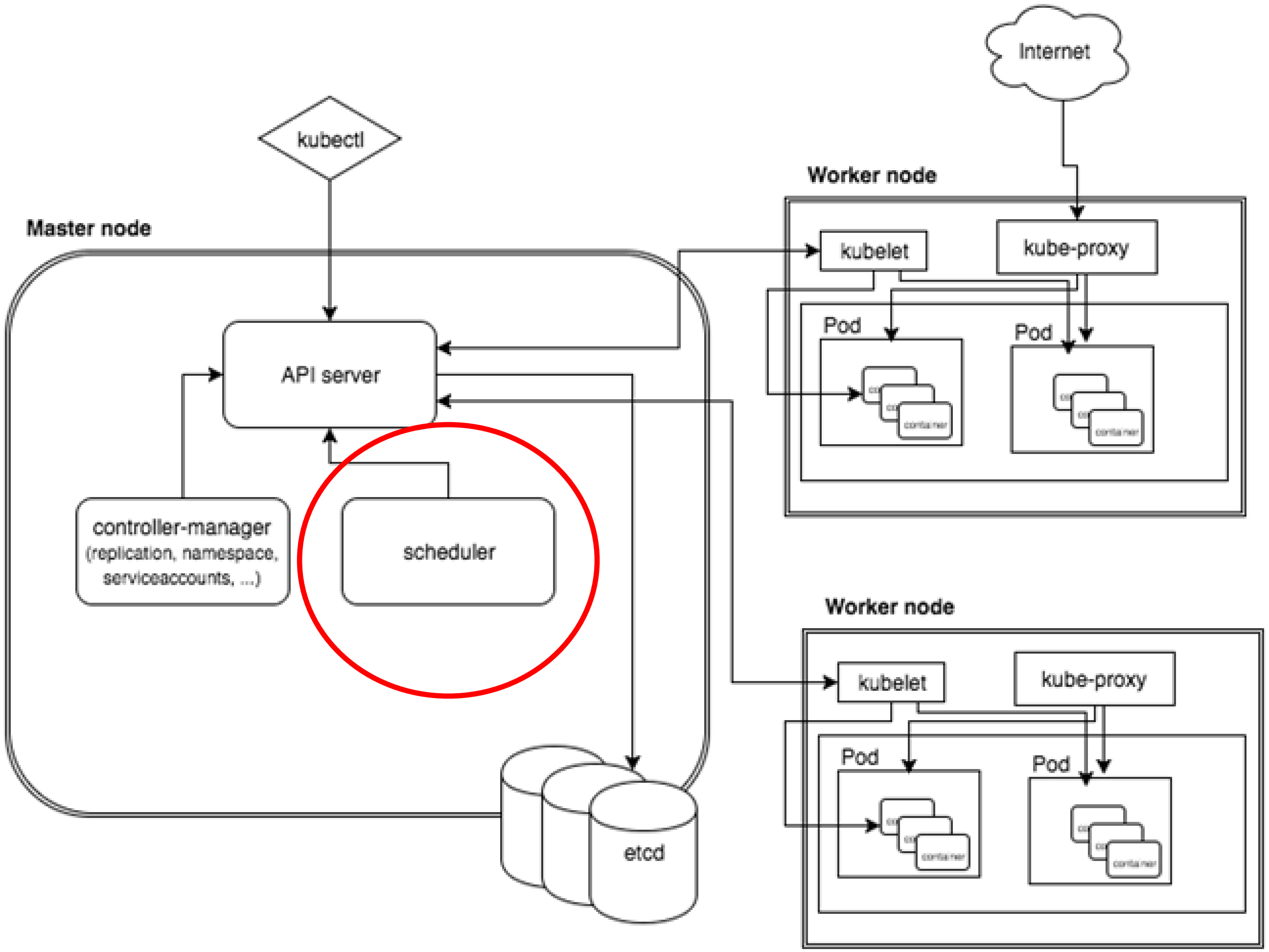
SmartShip – WP7 Workshop https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



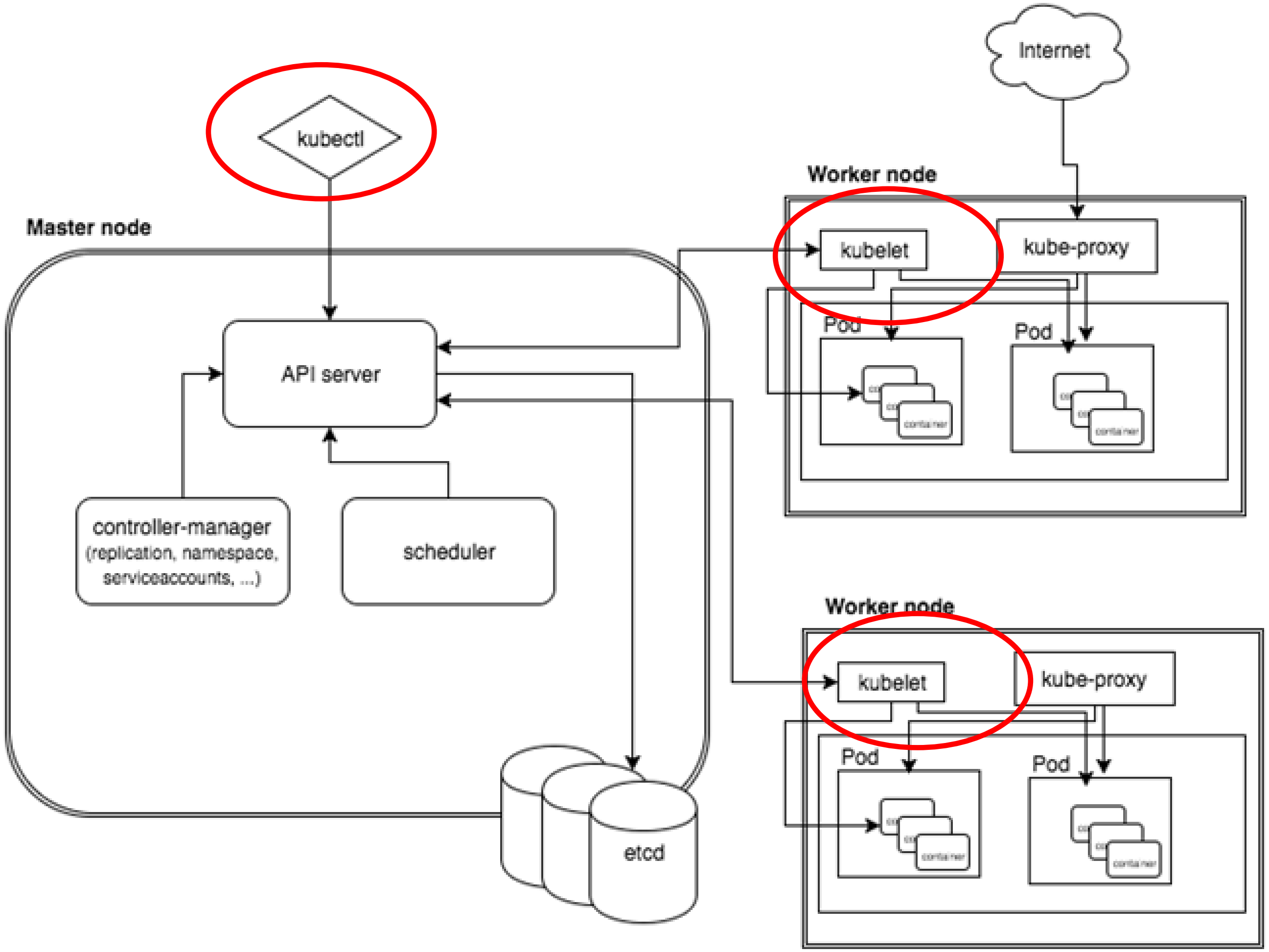
https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



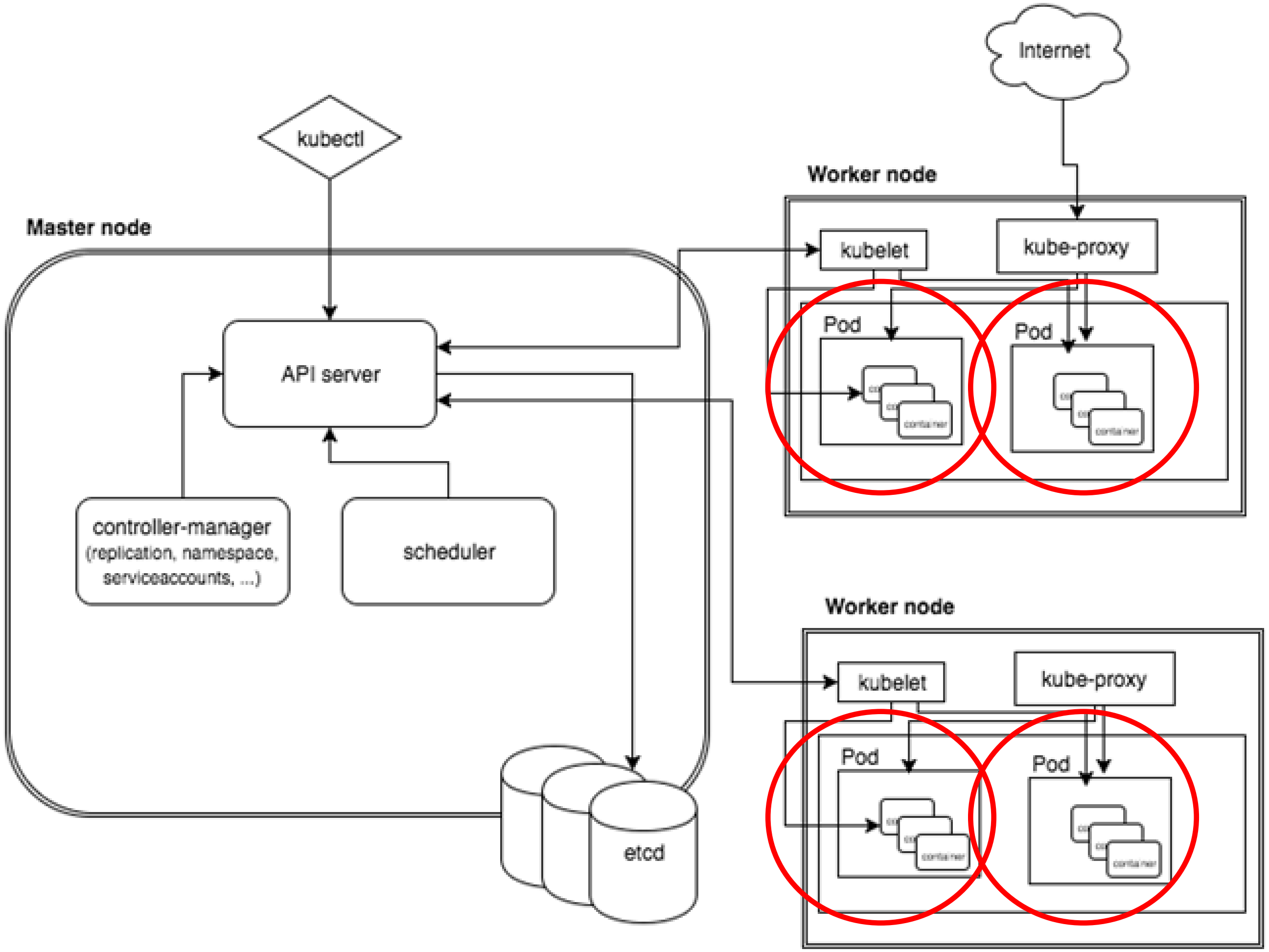
https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



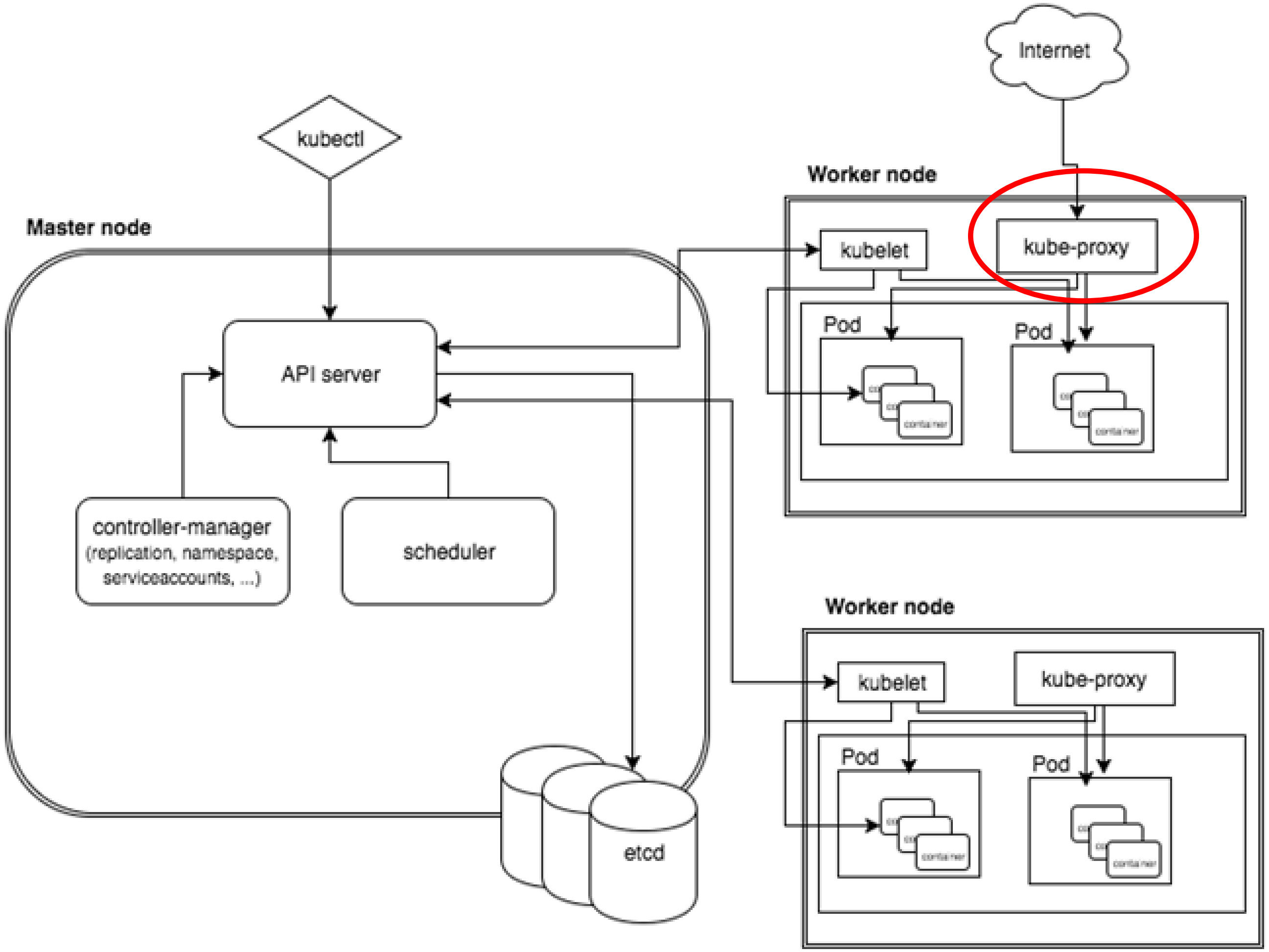
https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



https://x-team.com/blog/introduction-kubernetes-architecture/

Architecture



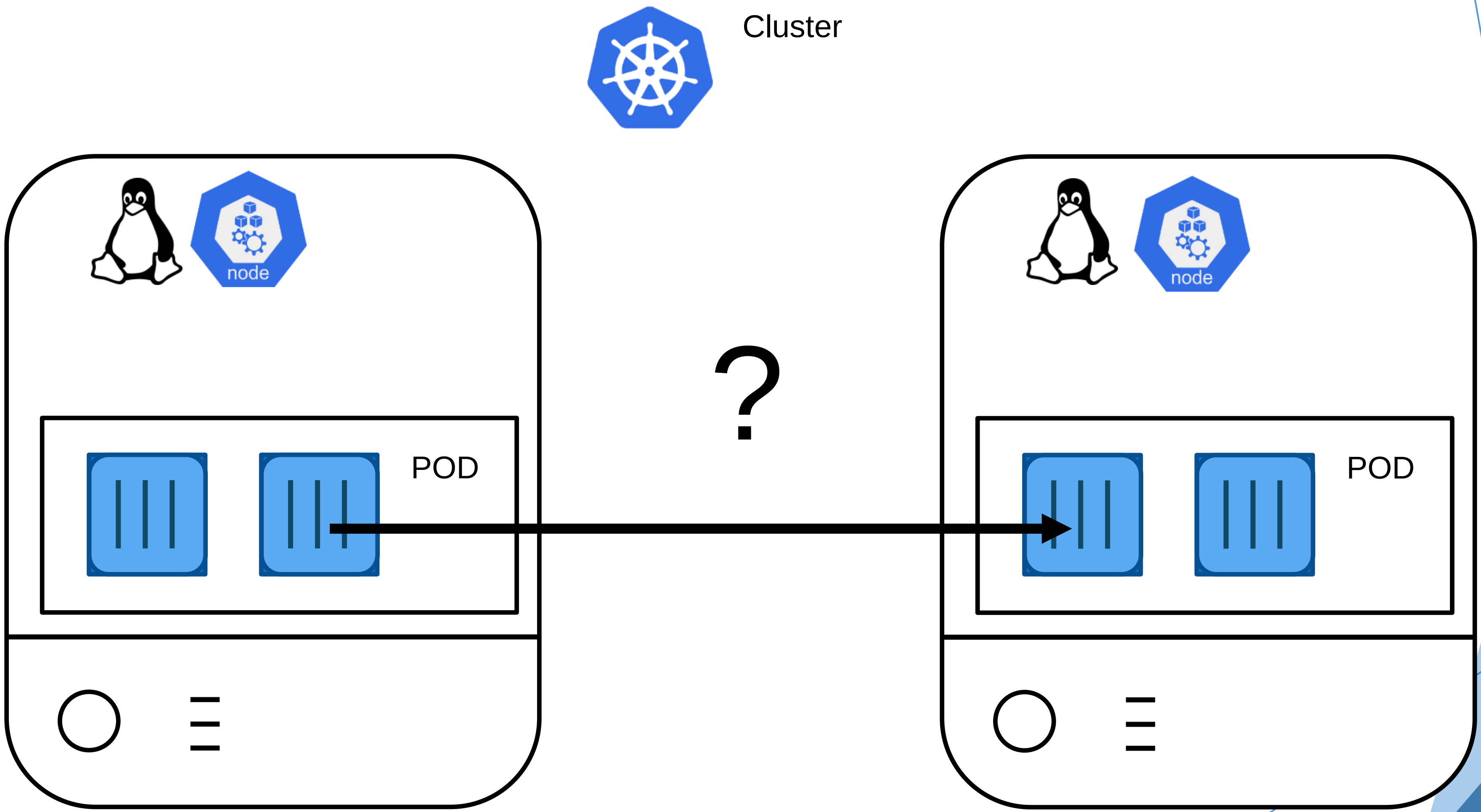
https://x-team.com/blog/introduction-kubernetes-architecture/

Network - Challenges

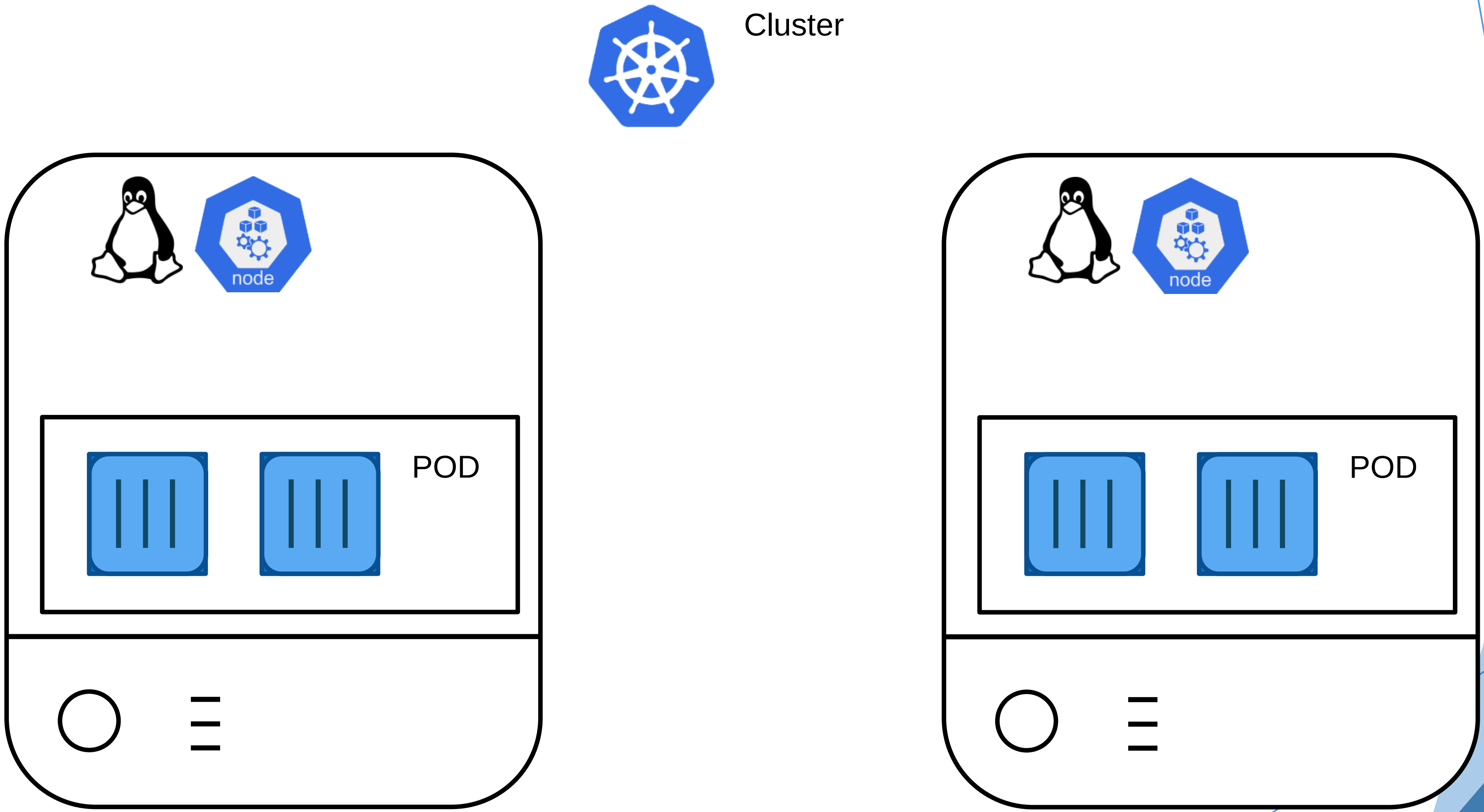
- Container to container communication inside Pods
- Pod to Pod communication on the same node and across cluster nodes
- Pod to Service communication within the same namespace and across cluster name spaces
- External to Service communication for clients to acces application in a cluster

https://x-team.com/blog/introduction-kubernetes-architecture/

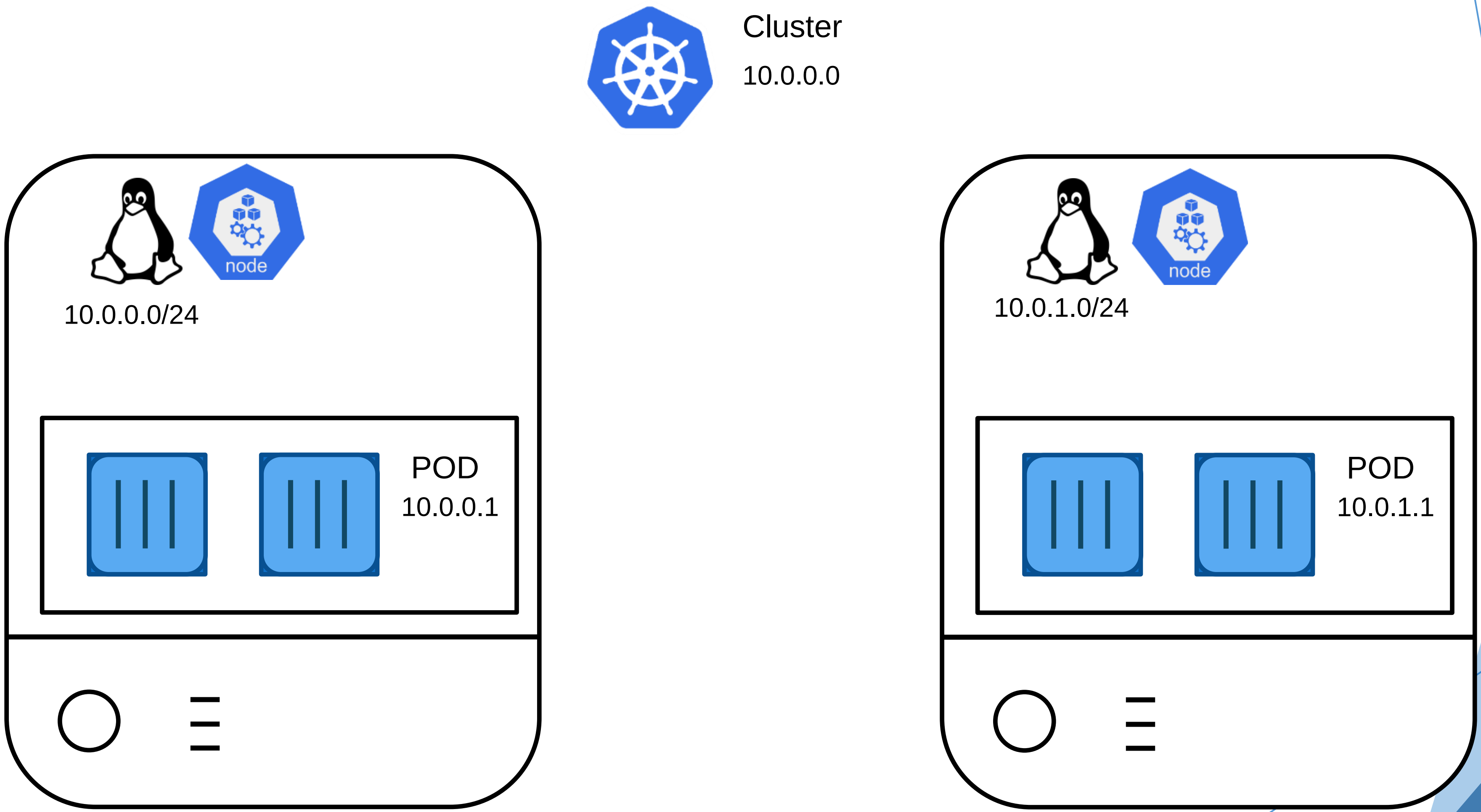
Communication overview



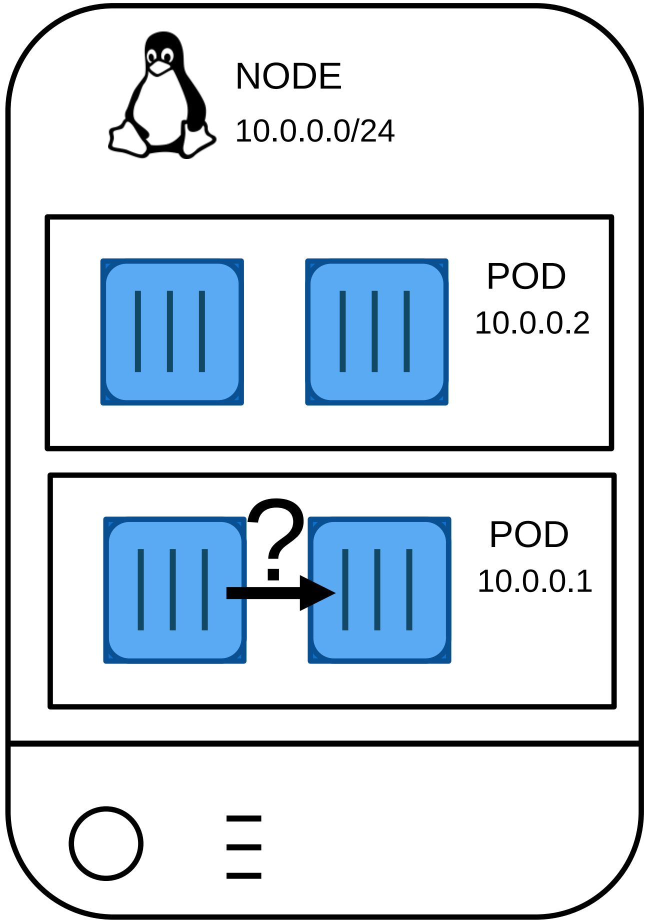
Communication overview



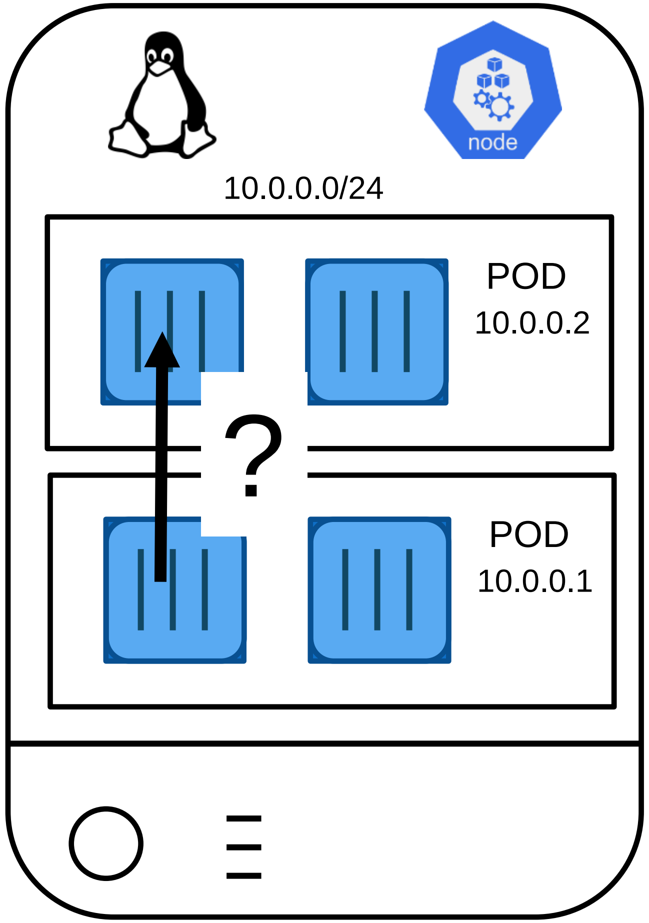
Communication overview



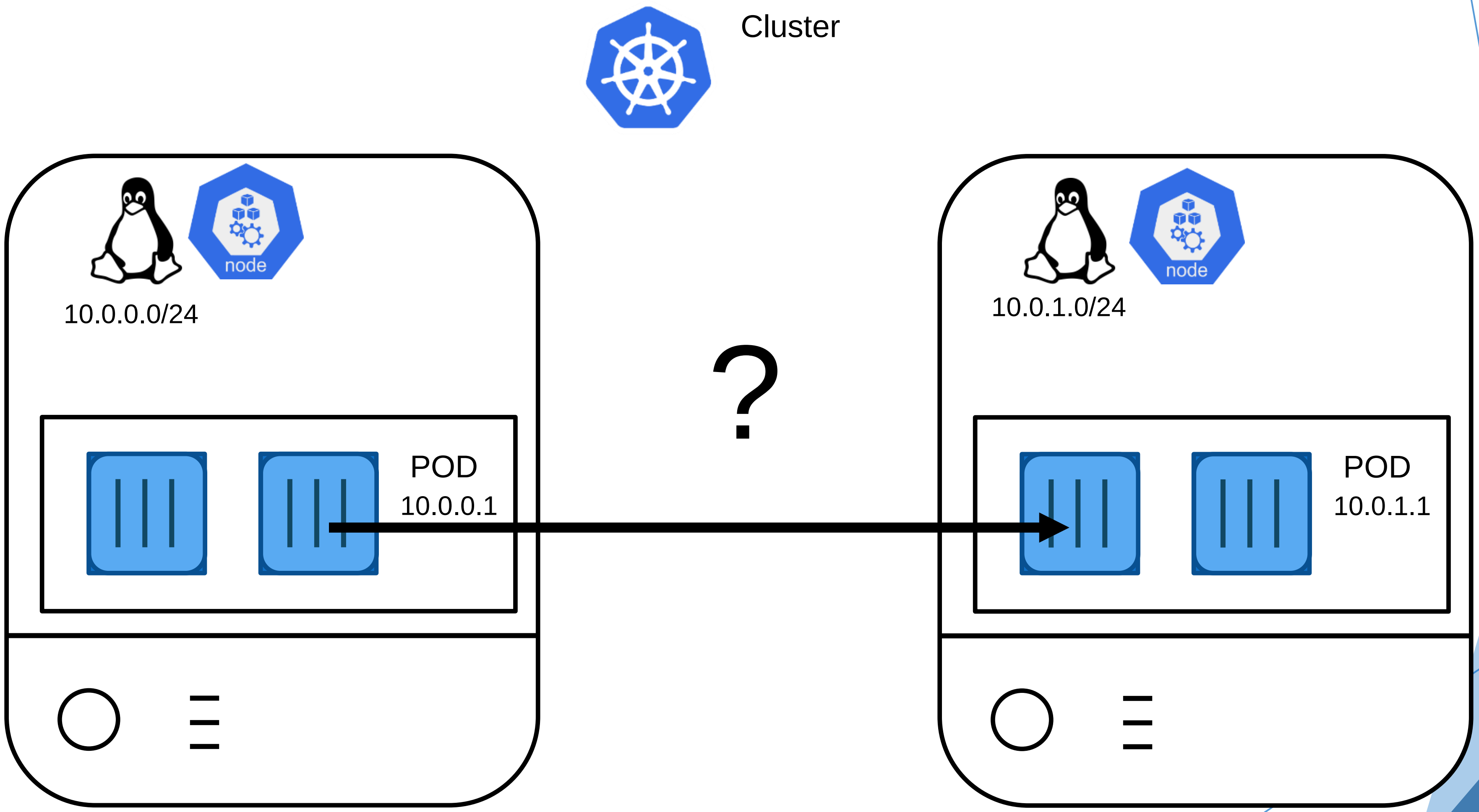
Container to container communication inside Pods



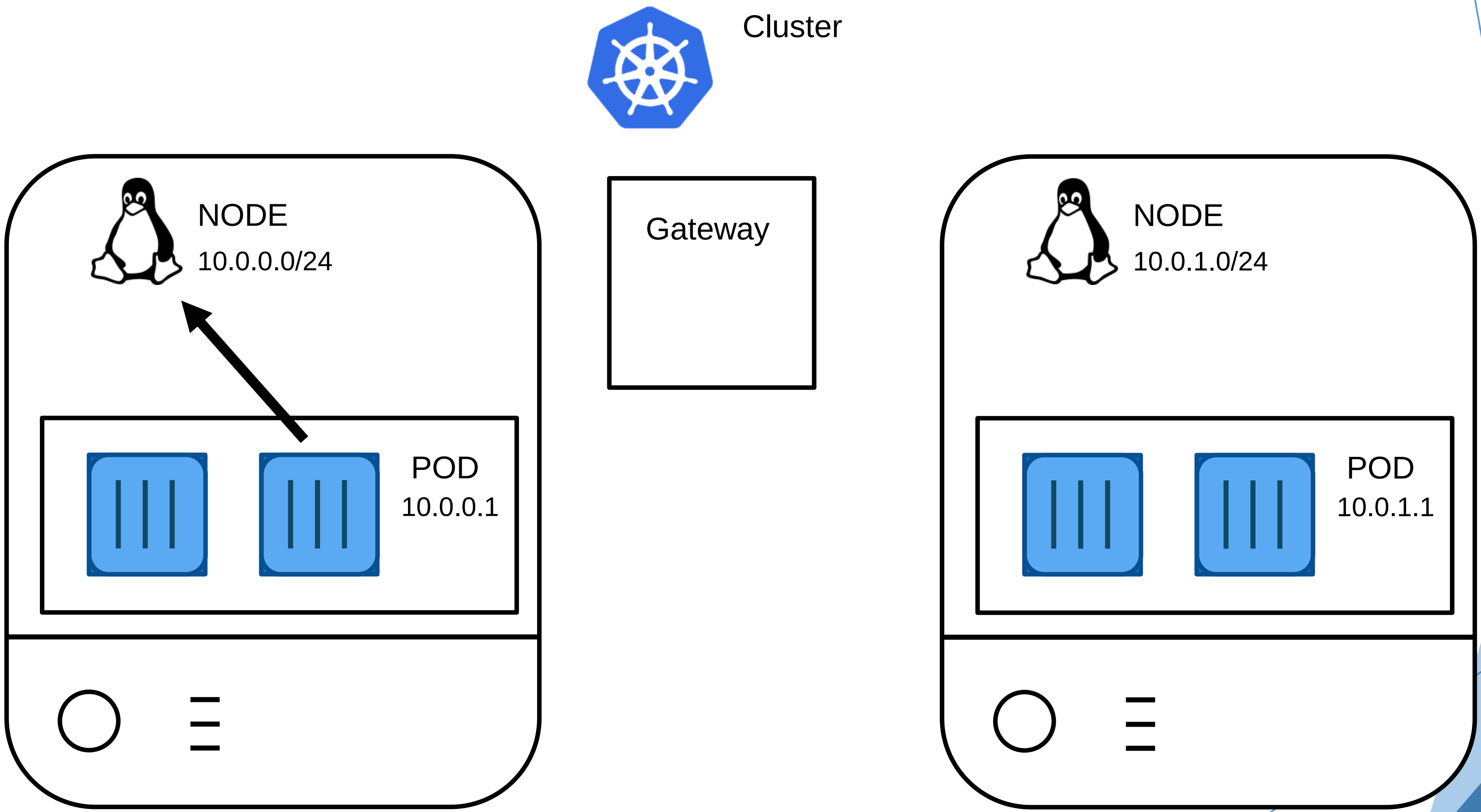
Pod to Pod communication on the same node



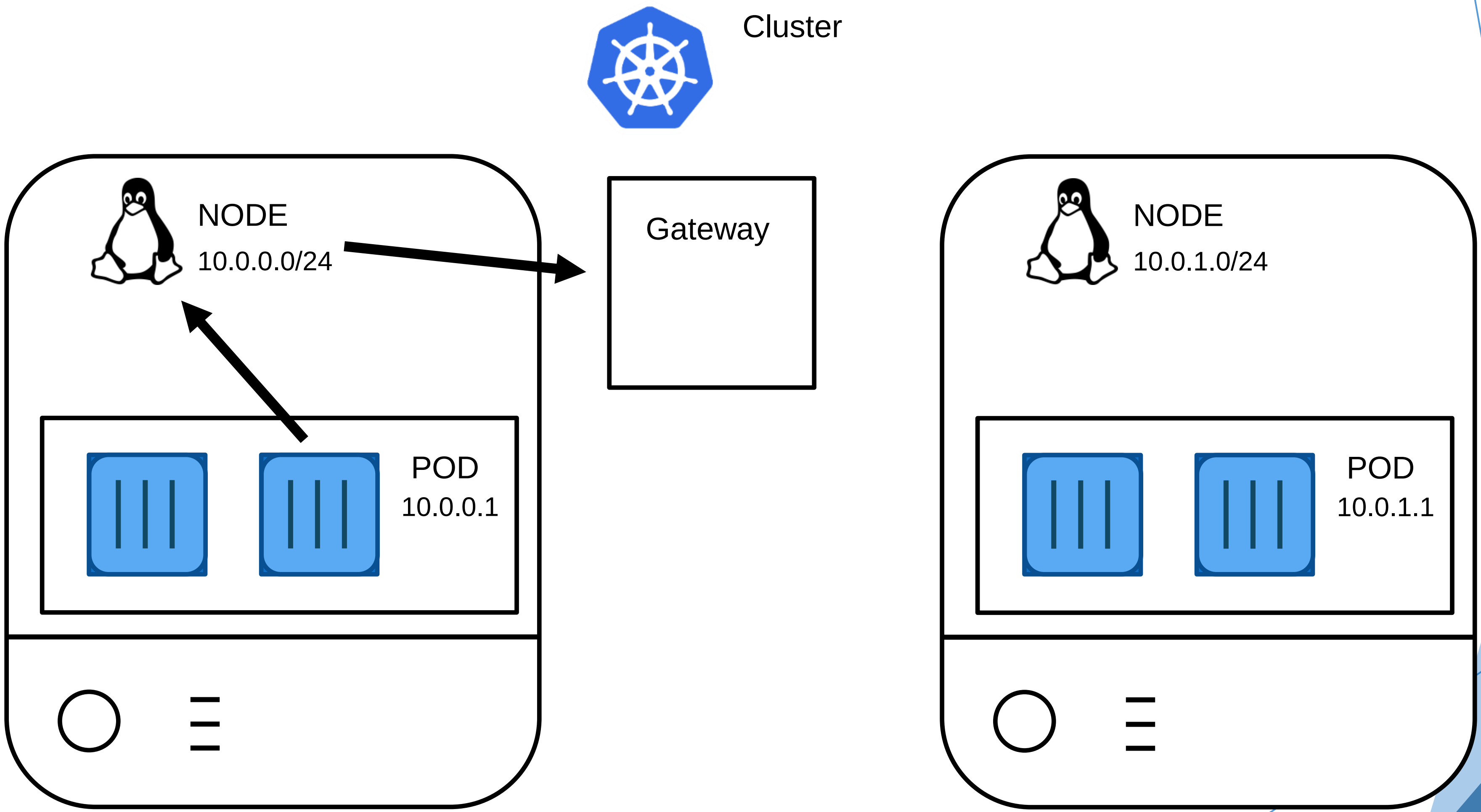
Pod to Pod communication cluster nodes



Pod to Pod communication between clusters nodes

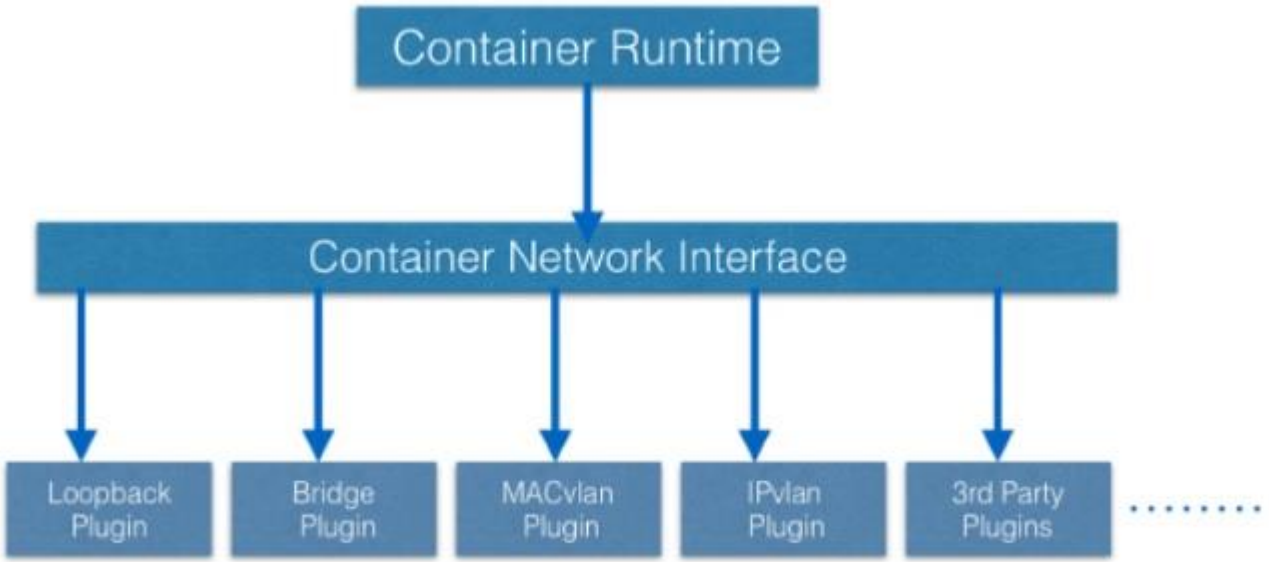


Pod to Pod communication between clusters nodes



CNI Container Network Interface

CNI



Container Runtime

Container Network Interface

Loopback Plugin

Bridge Plugin

MACvlan Plugin

IPvlan Plugin

3rd Party Plugins

....

Container Network Interface (CNI) Core Plugins

4 ways of implementation

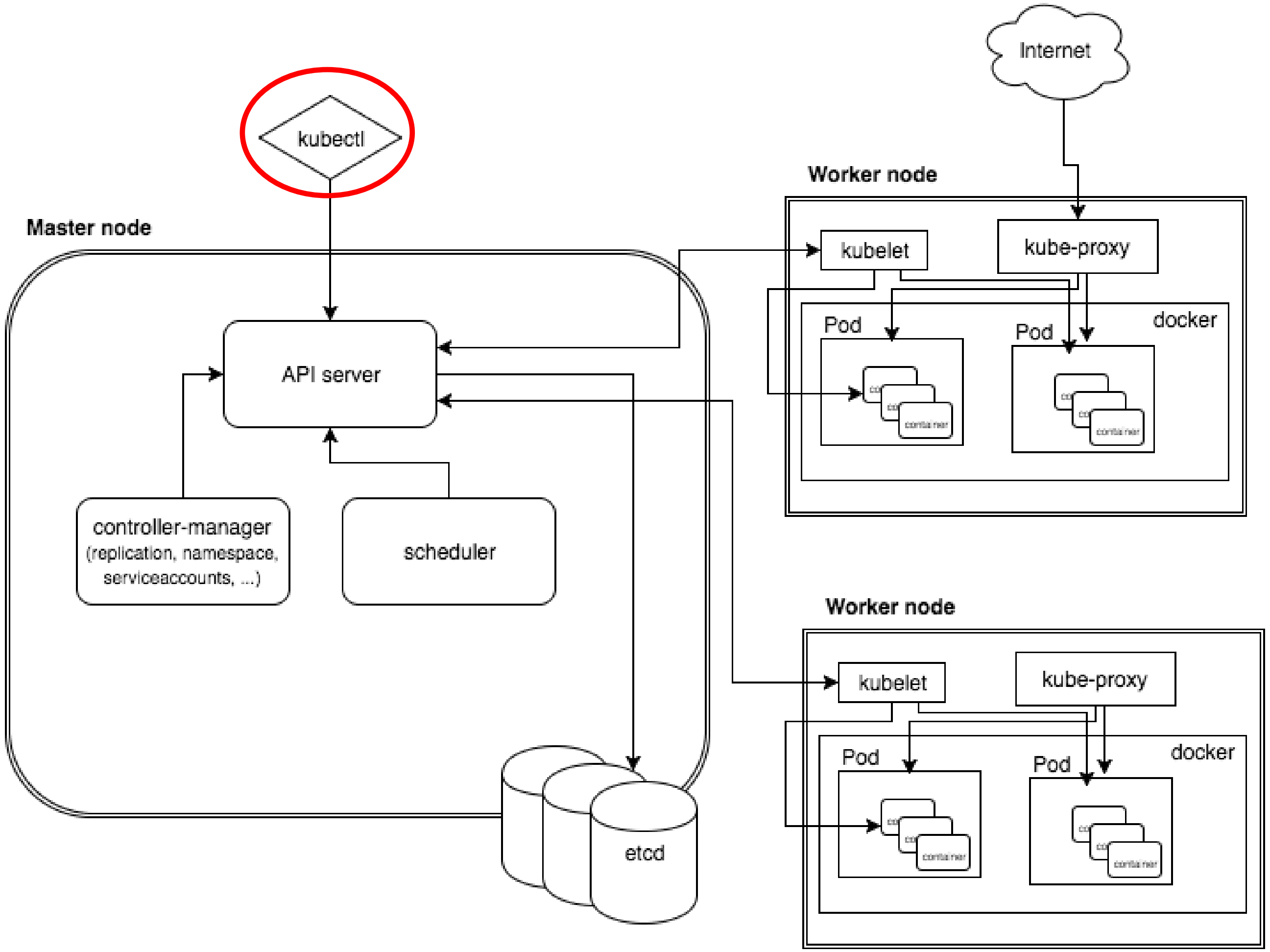
- Owning the infrastructure and connect the nodes with the switches on the layer 2
- Use the IP config tables
- Use the encapsulation of the packets -> IP/IP tunnel
- VXLAN

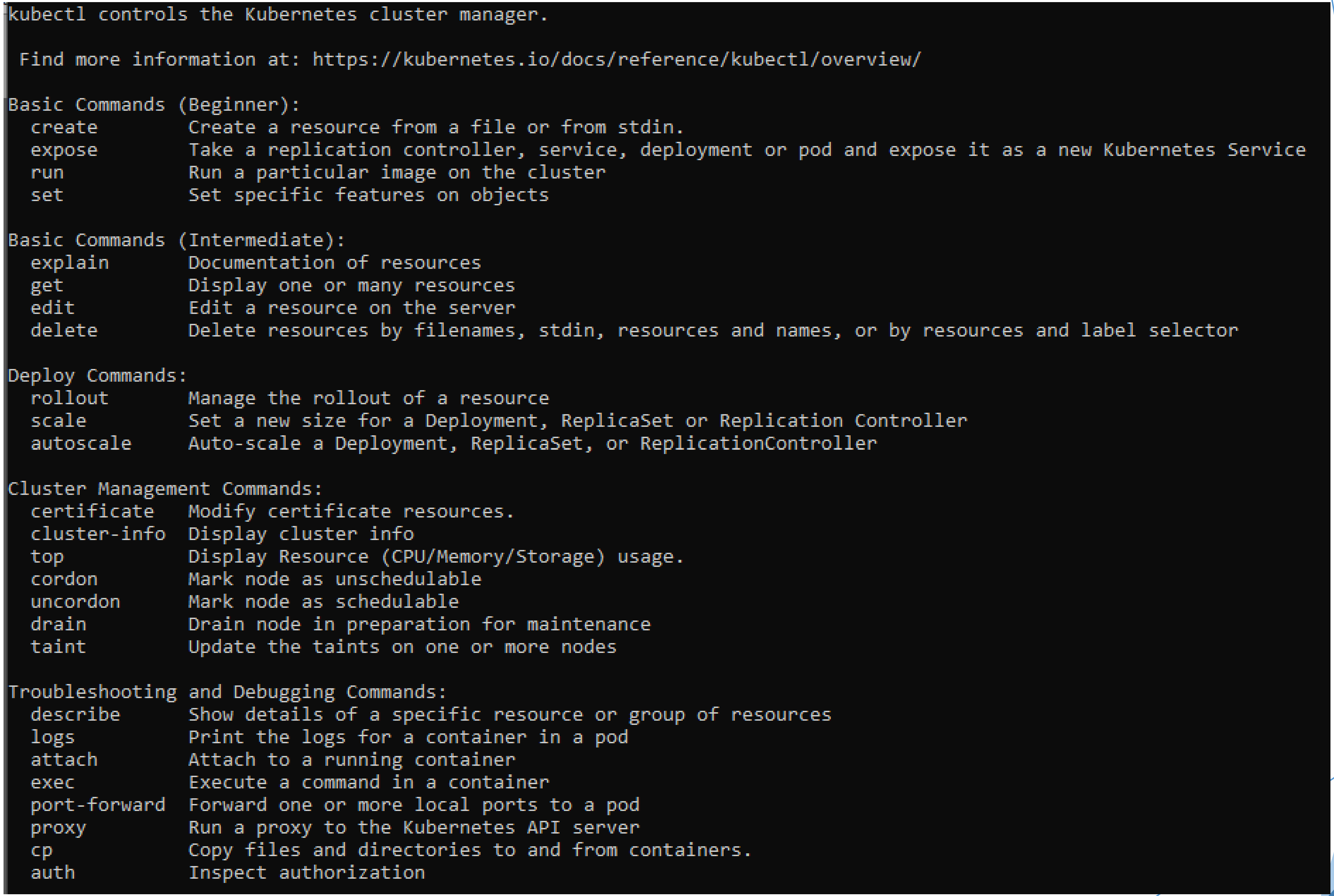
CNI Container Network Interface
Most popular implementations

flannel

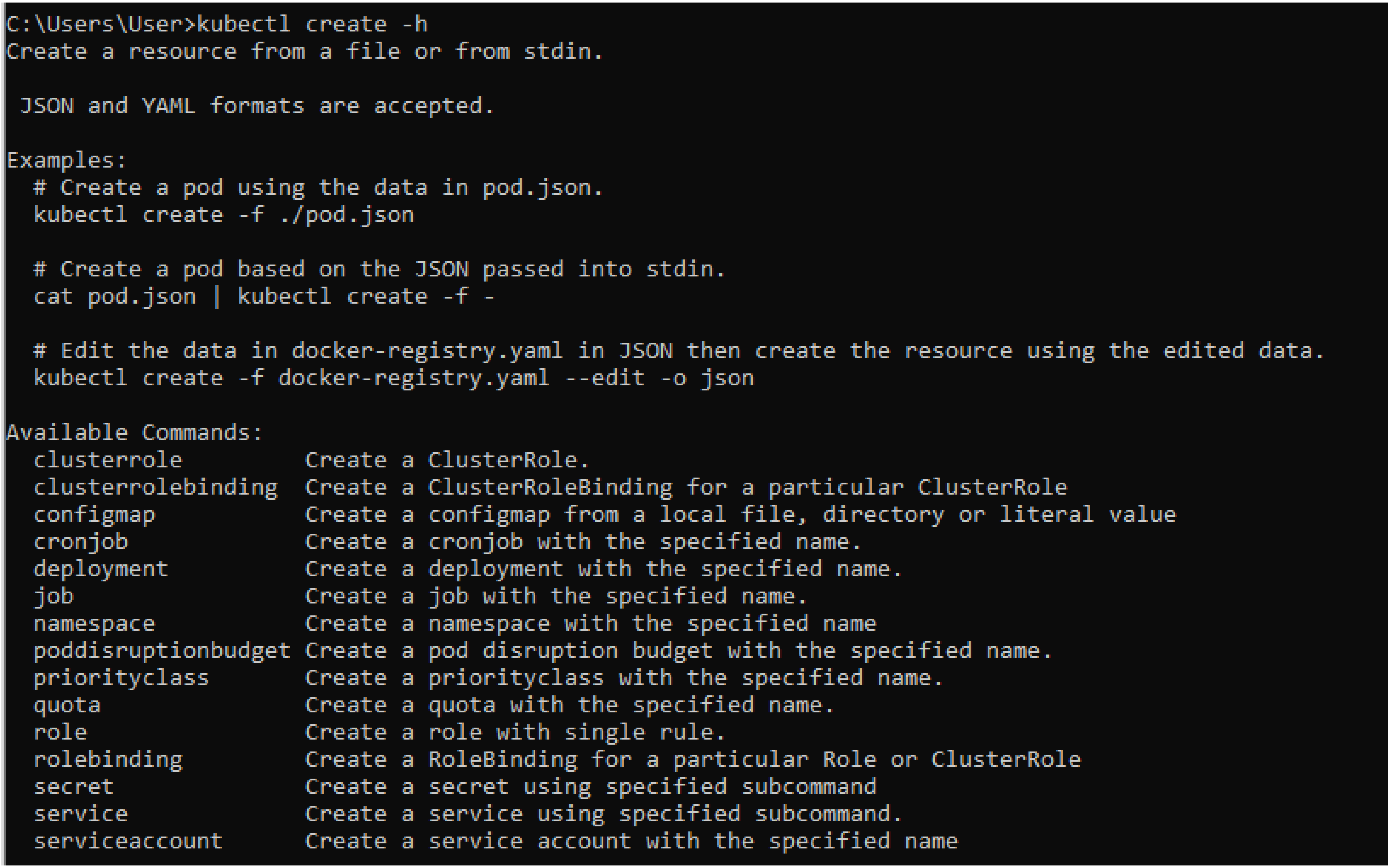
weavecloud

How to interact with cluster



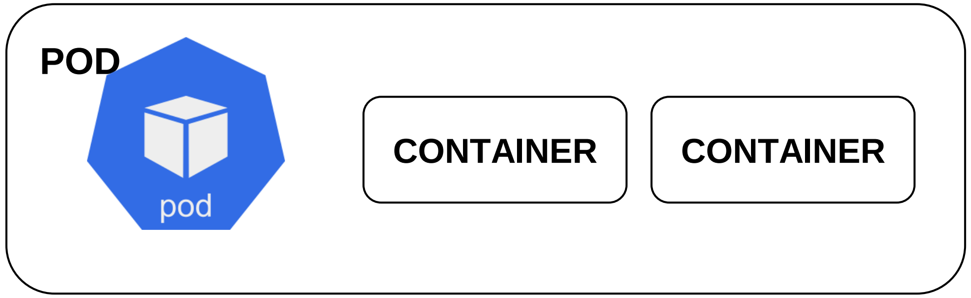


```
kubectl controls the Kubernetes cluster manager.

 Find more information at: https://kubernetes.io/docs/reference/kubectl/overview/

Basic Commands (Beginner):
  create        Create a resource from a file or from stdin.
  expose        Take a replication controller, service, deployment or pod and expose it as a new Kubernetes Service
  run           Run a particular image on the cluster
  set           Set specific features on objects

Basic Commands (Intermediate):
  explain       Documentation of resources
  get           Display one or many resources
  edit          Edit a resource on the server
  delete        Delete resources by filenames, stdin, resources and names, or by resources and label selector

Deploy Commands:
  rollout       Manage the rollout of a resource
  scale         Set a new size for a Deployment, ReplicaSet or Replication Controller
  autoscale     Auto-scale a Deployment, ReplicaSet, or ReplicationController

Cluster Management Commands:
  certificate   Modify certificate resources.
  cluster-info  Display cluster info
  top           Display Resource (CPU/Memory/Storage) usage.
  cordon        Mark node as unschedulable
  uncordon      Mark node as schedulable
  drain         Drain node in preparation for maintenance
  taint         Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe      Show details of a specific resource or group of resources
  logs          Print the logs for a container in a pod
  attach        Attach to a running container
  exec          Execute a command in a container
  port-forward  Forward one or more local ports to a pod
  proxy         Run a proxy to the Kubernetes API server
  cp            Copy files and directories to and from containers.
  auth          Inspect authorization
```



```
C:\Users\User>kubectl create -h
Create a resource from a file or from stdin.

 JSON and YAML formats are accepted.

Examples:
  # Create a pod using the data in pod.json.
  kubectl create -f ./pod.json

  # Create a pod based on the JSON passed into stdin.
  cat pod.json | kubectl create -f -

  # Edit the data in docker-registry.yaml in JSON then create the resource using the edited data.
  kubectl create -f docker-registry.yaml --edit -o json

Available Commands:
  clusterrole         Create a ClusterRole.
  clusterrolebinding  Create a ClusterRoleBinding for a particular ClusterRole
  configmap           Create a configmap from a local file, directory or literal value
  cronjob             Create a cronjob with the specified name.
  deployment          Create a deployment with the specified name.
  job                 Create a job with the specified name.
  namespace           Create a namespace with the specified name
  poddisruptionbudget Create a pod disruption budget with the specified name.
  priorityclass       Create a priorityclass with the specified name.
  quota               Create a quota with the specified name.
  role                Create a role with single rule.
  rolebinding         Create a RoleBinding for a particular Role or ClusterRole
  secret              Create a secret using specified subcommand
  service             Create a service using specified subcommand.
  serviceaccount      Create a service account with the specified name
```

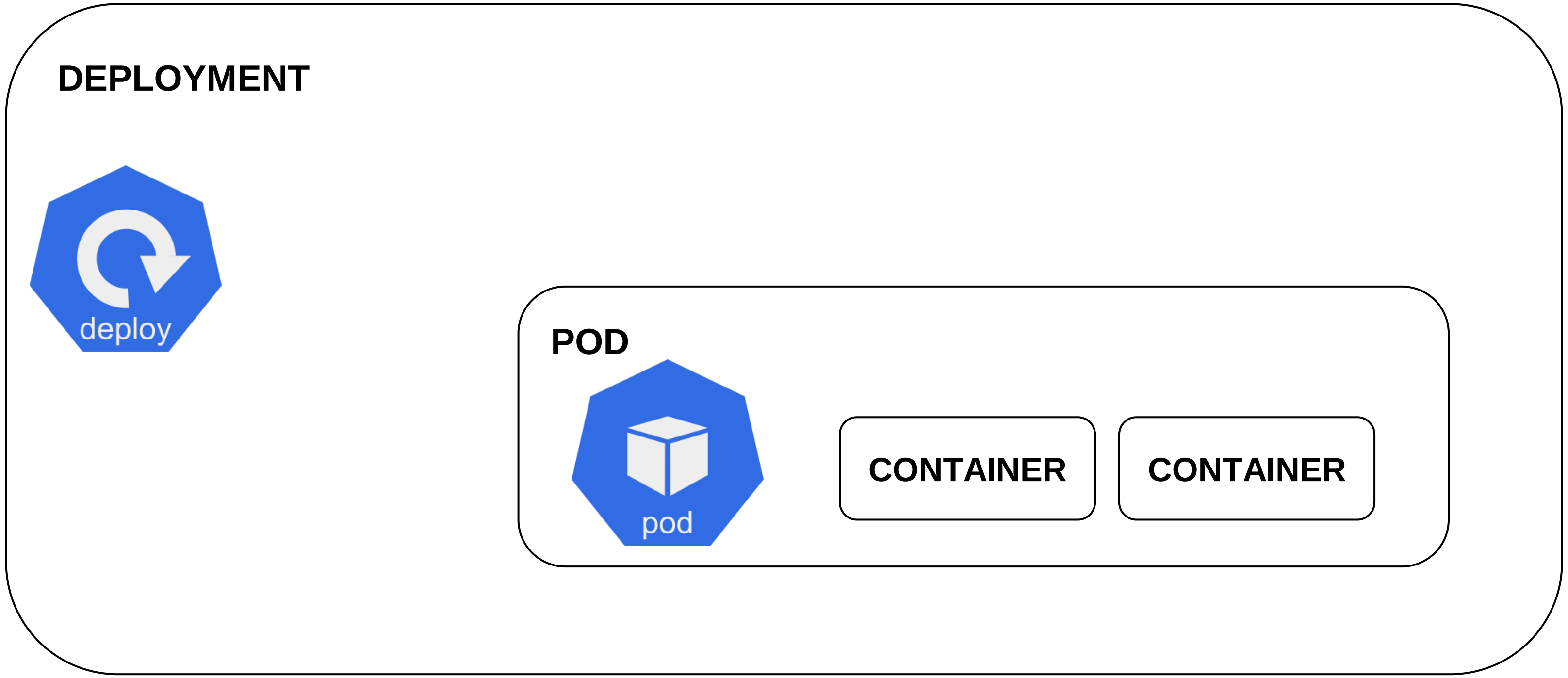
Deployment


POD

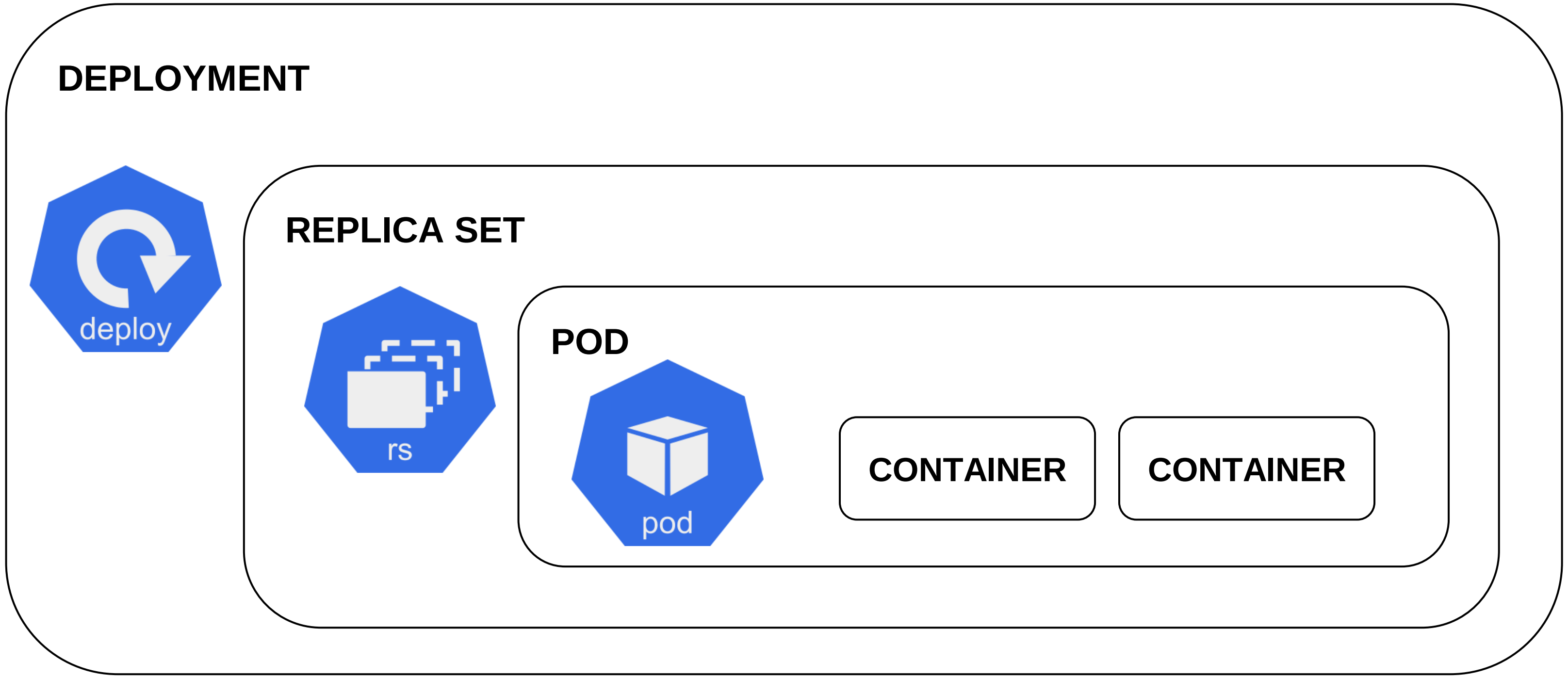
CONTAINER

CONTAINER

Deployment

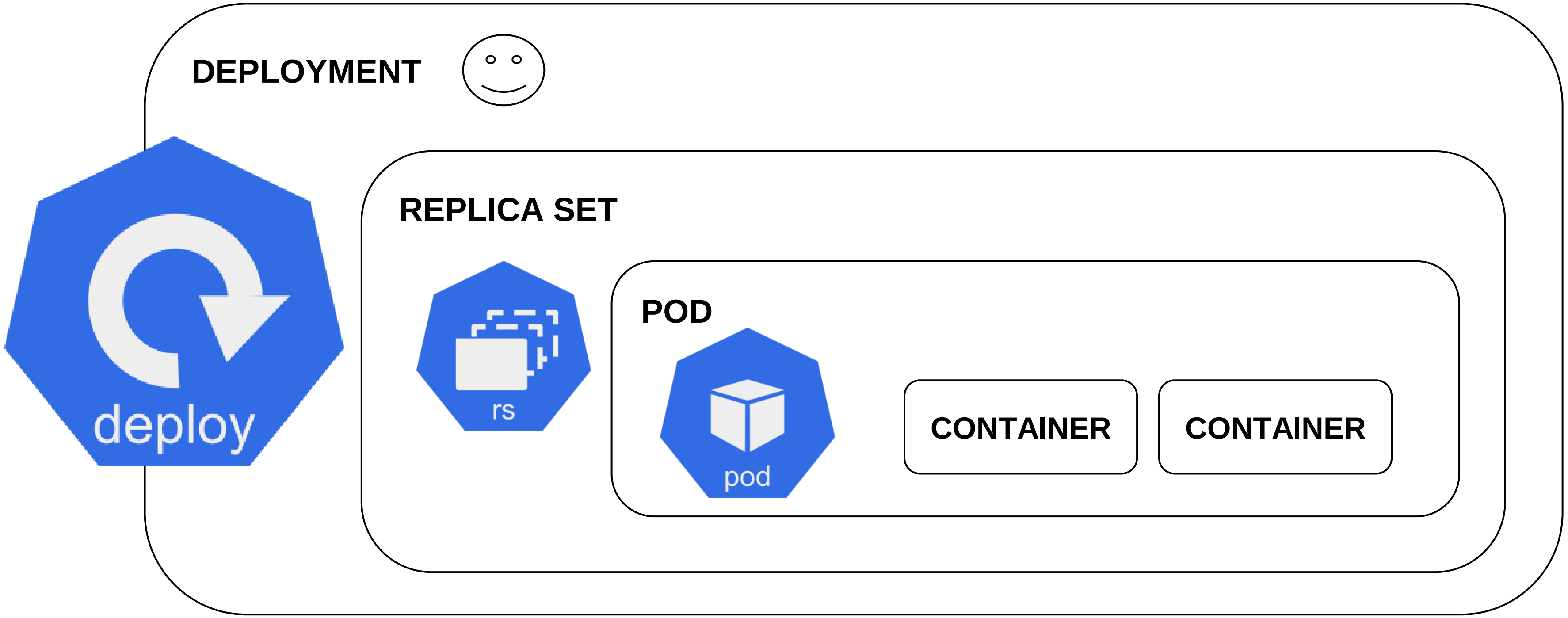


Deployment

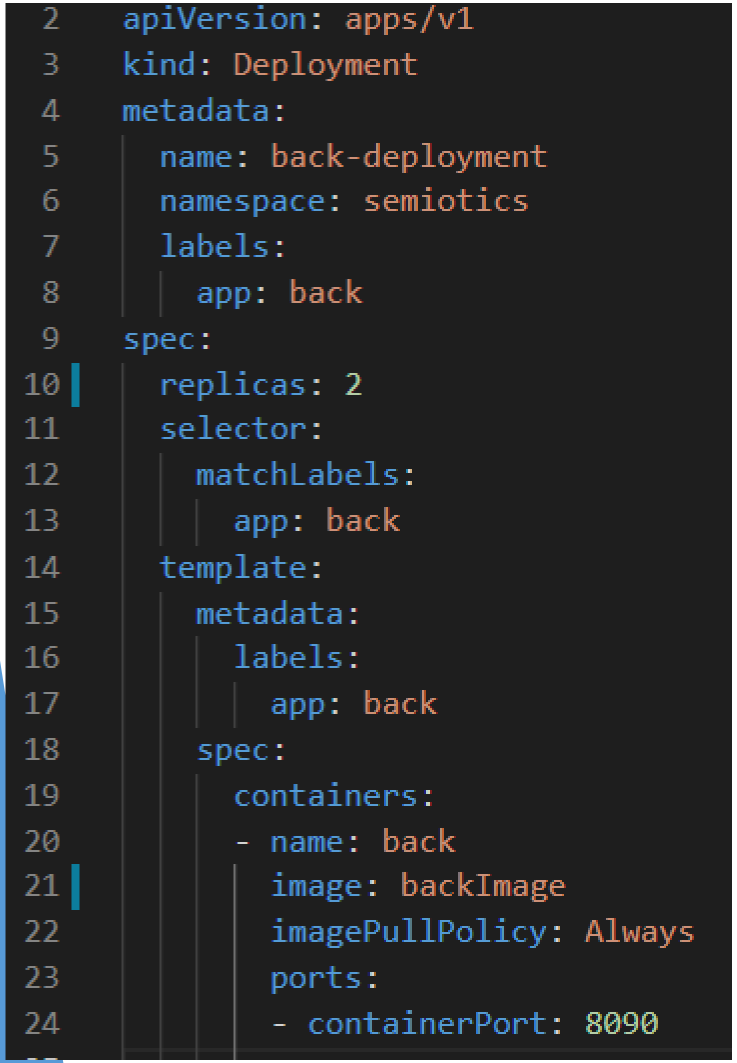


DEPLOYMENT
deploy
REPLICA SET
rs
POD
pod
CONTAINER
CONTAINER

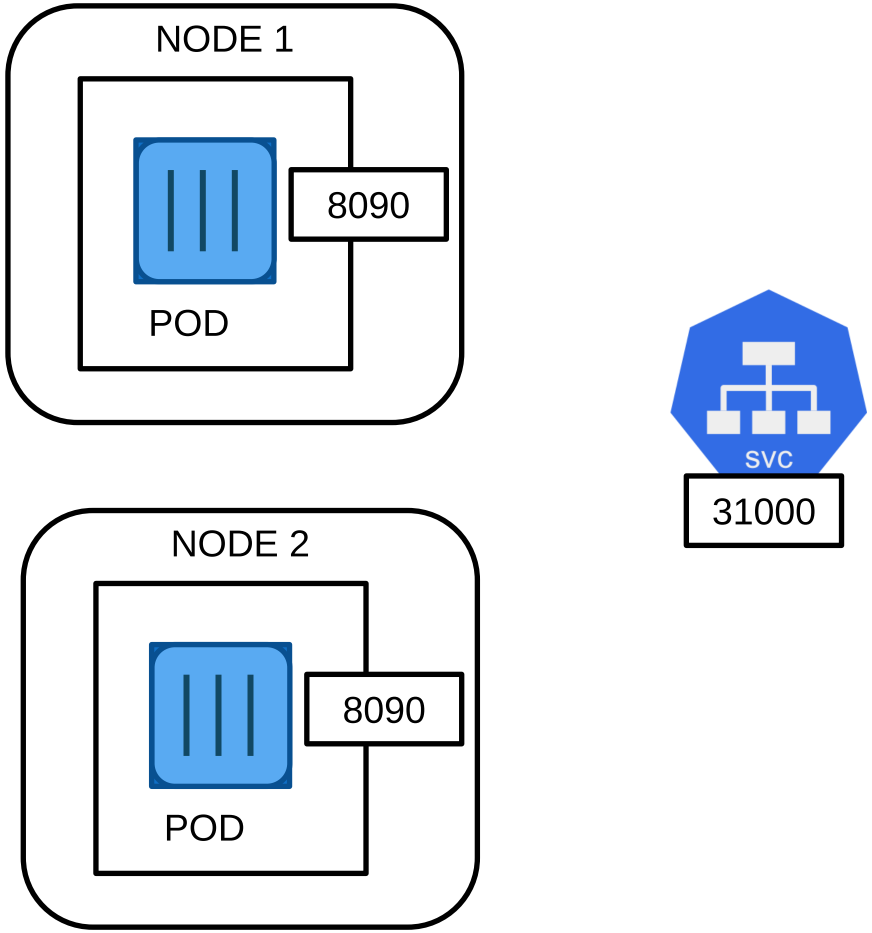
Deployment

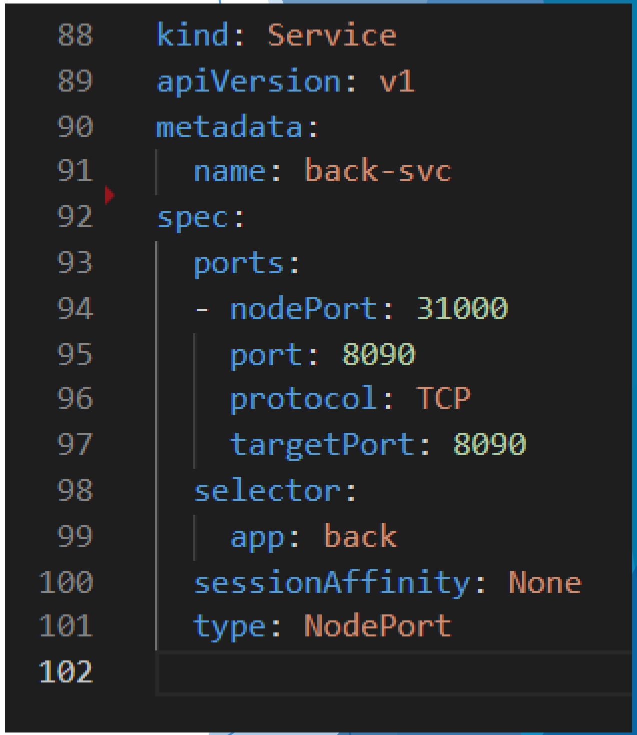

DEPLOYMENT
deploy
REPLICA SET
rs
POD
pod
CONTAINER
CONTAINER

Services – abstraction of the network



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: back-deployment
  namespace: semiotics
  labels:
    app: back
spec:
  replicas: 2
  selector:
    matchLabels:
      app: back
  template:
    metadata:
      labels:
        app: back
    spec:
      containers:
      - name: back
        image: backImage
        imagePullPolicy: Always
        ports:
        - containerPort: 8090
```





```yaml
kind: Service
apiVersion: v1
metadata:
  name: back-svc
spec:
  ports:
  - nodePort: 31000
    port: 8090
    protocol: TCP
    targetPort: 8090
  selector:
    app: back
  sessionAffinity: None
  type: NodePort

```

Tips

- Minikube
- DinD – Docker in Docker
- Cloud
  - EKS Elastic Kubernetes Service - Amazon
  - AKS Azure Kubernetes Service – MS
  - CKS Cloud Kubernetes Service – IBM
  - GKE Google Kubernetes Engine – Google
- Secrets
- Persistens Volumes

Thanks for your attention

Questions

Reference

1. https://kubernetes.io/docs/home/
2. https://x-team.com/blog/introduction-kubernetes-architecture/
3. https://enterprisersproject.com/article/2020/6/kubernetes-statistics-2020
4. https://www.youtube.com/watch?v=ysk_2w_diKY
5. https://www.youtube.com/watch?v=X48VuDVv0do
6. https://www.cncf.io/
7. https://github.com/coreos/flannel/
8. https://www.weave.works/
9. https://github.com/kubernetes-retired/kubeadm-dind-cluster