KUBERNETES

AN INTRODUCTION

MUST USE CONTAINERS

KUBERNETES:

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

EX. KUBERNETES DOES NOT HAVE THE CONCEPT OF NETWORKING. FOR IT IS A PLUG IN TO MOUNT ON TOP. AS STORAGE AND OTHER THINGS

WHAT DOES "KUBERNETES" MEAN?

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

THE IDEA IS THAT IT DRIVES YOUR INFRASTRUCTURE

EX: IT'S AN ORCHESTRA WHERE EVERY MUSICIAN IS A COMPONENT OR A RESOURCE, AND KUBERNETS IS THE DIRECTOR

WHAT IS KUBERNETS?

- Project that was spun out of Google as an open source container orchestration platform.
- Built from the lessons learned in the experiences of developing and running Google's Borg.
- Designed from the ground-up as a loosely coupled collection of components centered around deploying, maintaining and scaling workloads.

IT WAS A DIFFERENT ENVIRONMENT BUT NOT OPEN SOURCE

CAN I JUST INSTALL KUBERNETES?

yes, and no, you need to choose a distribution before

Open source

LIGHT WEIGHT
NOT SCALABLE

- K0S (Mirantis)
- K3S (Rancher)
- K8S (google)

THE SUGGEST ONE

Commercial/Managed

- EKS-D (Amazon)
- OpenShift (Redhat)
- AKS (Microsoft)
- GKE (Google)

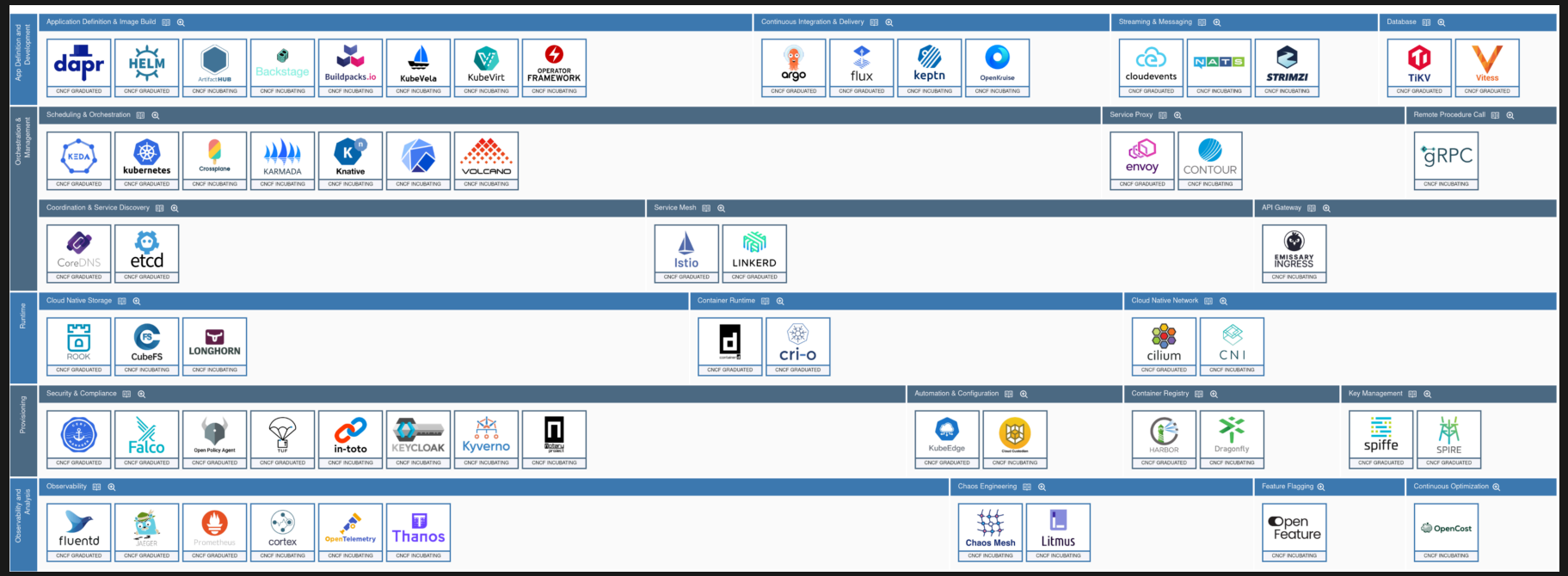
WHO MANAGES KUBERNETES?

CLOUD NATIVE
COMPUTING FOUNDATION

The CNCF is a child entity of the Linux Foundation and operates as a vendor neutral governance group.

CNCF PORJECTS

(GRADUATED & INCUBATING)



WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery and load balancing Ex: DNS
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

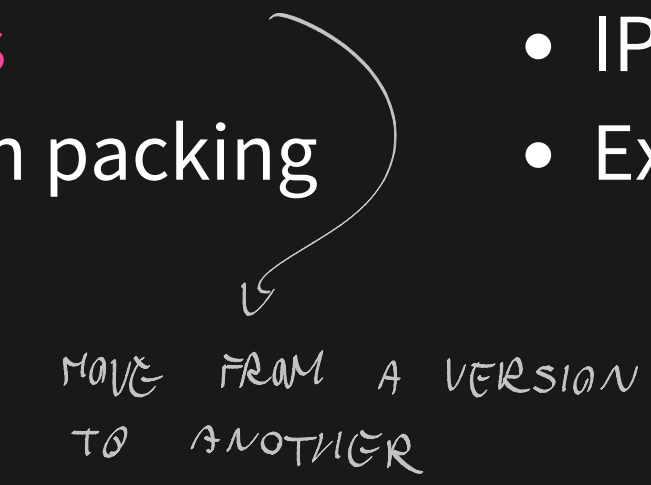
WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

EX. AUTOMATING MOUNTING FILE SYSTEM (EX: NFS) THAT CONTAINS ALL CONTAINERS ON DIFFERENT MACHINE

WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts and rollbacks
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility



WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic process packing (PROCESS MENAGER)
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

IF THE SYSTEM KEEP CRASHING AND NEED TO UNDERSTAND WHAT IS HAPPENING

WHAT DOES KUBERNETES PROVIDES YOU WITH?

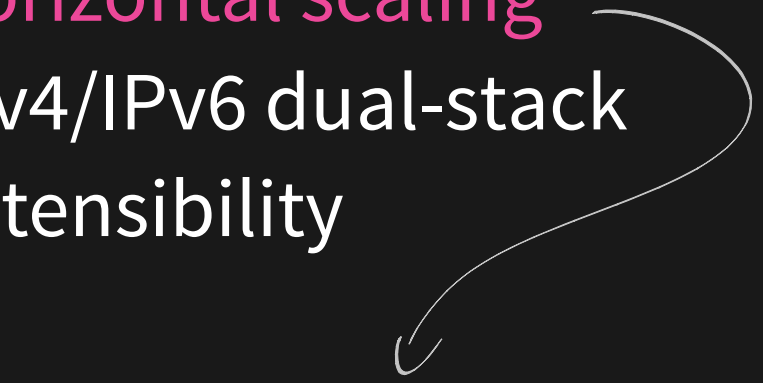
- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Secret and configuration management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution → MENAGE WORKLOADS
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility



WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Extensibility

WHAT DOES KUBERNETES PROVIDES YOU WITH?

- Service discovery
- Storage orchestration
- Automated rollouts
- Automatic bin packing
- Self-healing
- Config management
- Batch execution
- Horizontal scaling
- IPv4/IPv6 dual-stack
- Designed for extensibility

WHAT KUBERNETES DOES NOT

- limit the types of applications supported
- deploy source code/build your application
- provide application-level services
- dictate logging, monitoring, or alerting solutions.
- provide/mandate a configuration language
- comprehensive machine configuration, maintenance, management, or self-healing systems.

YOU CAN HAVE 2 DIFFERENT KINDS OF APPLICATION

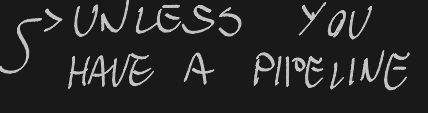
STATELESS VS STATEFUL

stateless: no past data nor state is stored or needs to be persistent when a new container is created (eg. CDN)

stateful: prior request history impacts the current state; the server must access and hold onto state information generated during the processing of the earlier request. (eg. DB)

WHAT KUBERNETES DOES NOT

- limit the types of applications supported
- deploy source code/build your application
- provide application-level services
- dictate logging, monitoring, or alerting solutions.
- provide/mandate a configuration language
- comprehensive machine configuration, maintenance, management, or self-healing systems.



WHAT KUBERNETES DOES NOT

- limit the types of applications supported
- deploy source code/build your application
- provide application-level services
- dictate logging, monitoring, or alerting solutions.
- provide/mandate a configuration language
- comprehensive machine configuration, maintenance, management, or self-healing systems.

IT DOES NOT GIVE MEMORY, BUT NEEDS A SOFTWARE

WHAT KUBERNETES DOES NOT

- limit the types of applications supported
- deploy source code/build your application
- provide application-level services
- dictate logging, monitoring, or alerting solutions.
- provide/mandate a configuration language
- comprehensive machine configuration, maintenance, management, or self-healing systems.

NEED A PLUG IN FOR LOG AND MONITORING

WHAT KUBERNETES DOES NOT

- limit the types of applications supported
- deploy source code/build your application
- provide application-level services
- dictate logging, monitoring, or alerting solutions.
- provide/mandate a configuration language
- comprehensive machine configuration, maintenance, management, or self-healing systems.

-> HIS OWN, BUT HAVE MANY DEGREES OF FREEDOM

WHAT KUBERNETES DOES NOT

- limit the types of applications supported
- deploy source code/build your application
- provide application-level services
- dictate logging, monitoring, or alerting solutions.
- provide/mandate a configuration language
- comprehensive machine configuration, maintenance, management, or self-healing systems.

↳ IT DOESN'T CARE ABOUT THE MACHINE

WHAT DO YOU WANT WHEN YOU EMBRACE KUBERNETES?

A DECLARATIVE INFRASTRUCTURE

WHAT IS A DECLARATIVE INFRASTRUCTURE?

Def: method of defining infrastructure in code by describing the desired state of resources.

Example Statement: "Deploy 3 instances of an application with load balancing enabled."

HOW IS THIS DIFFERENT FROM STANDARD MODE?
YOU JUST ASK AND DON'T NEED TO WRITE ALL NECESSARY CODE

Example:

- 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.”

... something happens ...

- Kubernetes: “Oh look, one has died. I’m going to attempt to spin up a new one.”

Declarative	**Imperative**
Specifies desired state	Specifies sequence of steps
Eg.: YAML, JSON configurations	Eg.: Shell scripts, Python
Idempotent	May cause issues on repeated runs
Less error-prone	More complex troubleshooting

WHY IAC?

- Consistency: Enforces desired state consistently across environments.
- Scalability: Simplifies scaling
- Error Reduction: Eliminates complex scripting and minimizes human error.
- Auditability: Clear record of infrastructure states in version control.

A POSSIBLE FULL IAC STACK

- Terraform: Provisions the underlying infrastructure
  VMs, networks, etc.
- Ansible: Configures the provisioned infrastructure
  installing software, setting up configurations
- Kubernetes: Deploys/Manages containerized app.
  on the provisioned & configured infrastructure.

OBVIOUSLY THE COMPLEXITY
RAISE A LOT

FROM DOCKER TO KUBERNETES

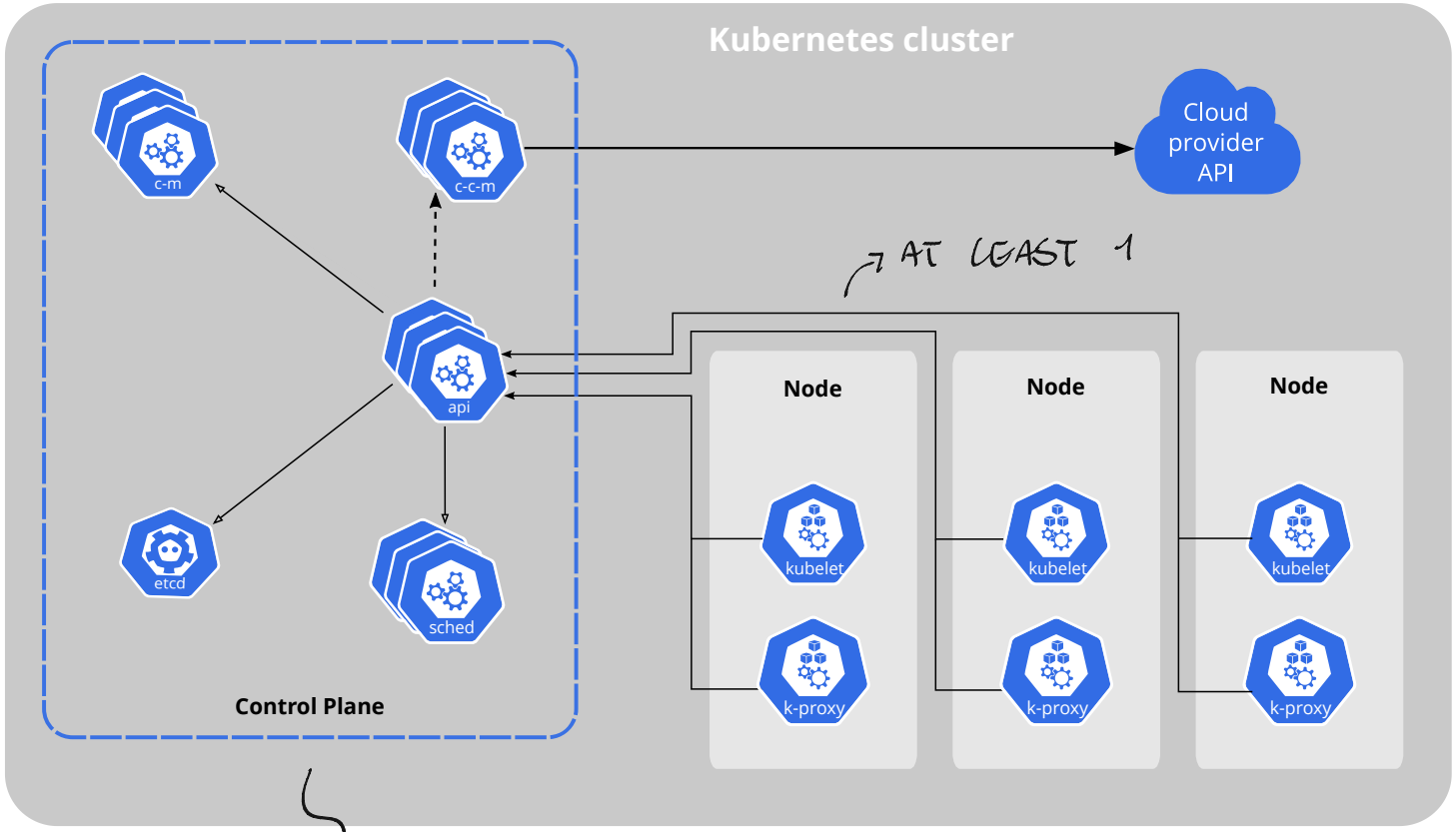
- Docker (single node)
- Docker compose (single node)
- Portainer, docker swarm (multi node)
  Gray area, small medium environment
- Kubernetes (multi node orchestration)

SHOULD I STILL USE DOCKER?

yes! Development and testing of your app can be done in docker.

KUBERNETES IS NOT MADE FOR TESTING, DOCKER YES

COMPONENTS OVERVIEW



SPECIAL NODE WITH ALL INTERNAL COMPONENTS

USUALLY YOU TALK WITH THE API AND GET THE WORK DONE

API server

Cloud controller manager *(optional)* → SOFTWARE THAT SERVE API

Controller manager

etcd (persistence store) → DATABASE THAT KEEP TRACK OF THE SYSTEM

kubelet → APPLICATION THAT KUBERNETES TALK TO IN EACH NODE

kube-proxy

Scheduler

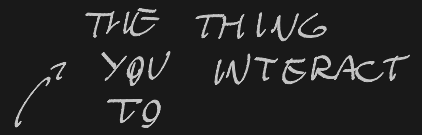
Control plane

Node

CONTROL PLANE COMPONENTS

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

exposes the Kubernetes API.



The API server is the front end for the Kubernetes control plane. The main implementation of a Kubernetes API server is kube-apiserver. kube-apiserver (main implementation) scale horizontally.

CONTROL PLANE COMPONENTS

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

key value store used as backing store for all cluster data.

CONTROL PLANE COMPONENTS

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

watches for newly created Pods with no assigned node, and selects a node for them to run on.

CONTROL PLANE COMPONENTS

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

scheduling decisions based on:

- individual and collective resource
- hardware/software/policy constraints (ex: deploy a container that needs GPU on a machine that have one)
- affinity and anti-affinity
- data locality
- etc.

CONTROL PLANE COMPONENTS

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

runs controller processes:

- Node controller: Responsible for noticing and responding when nodes go down.
- Job controller: Watches for Job objects that represent one-off tasks, then creates Pods to run those tasks to completion.
- EndpointSlice controller: Populates EndpointSlice objects
- etc.

CONTROL PLANE COMPONENTS

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

cloud-specific control logic. only runs controllers that are specific to your cloud provider. If you are running Kubernetes on your own premises, there is no cloud controller manager.

NODE COMPONENTS

- kubelet
- kube-proxy (optional)
- Container runtime

An agent. The kubelet takes a set of PodSpecs and ensures that the containers described are running and healthy.

NODE COMPONENTS

- kubelet
- kube-proxy (optional)
- Container runtime

kube-proxy maintains network rules on nodes allowing network communication to your Pods from network sessions inside or outside of your cluster. CNI can have other proxy meccanisms.

NODE COMPONENTS

- kubelet
- kube-proxy (optional)
- Container runtime

It is responsible for managing the execution and lifecycle of containers within the Kubernetes environment. (container-d/CRIO)

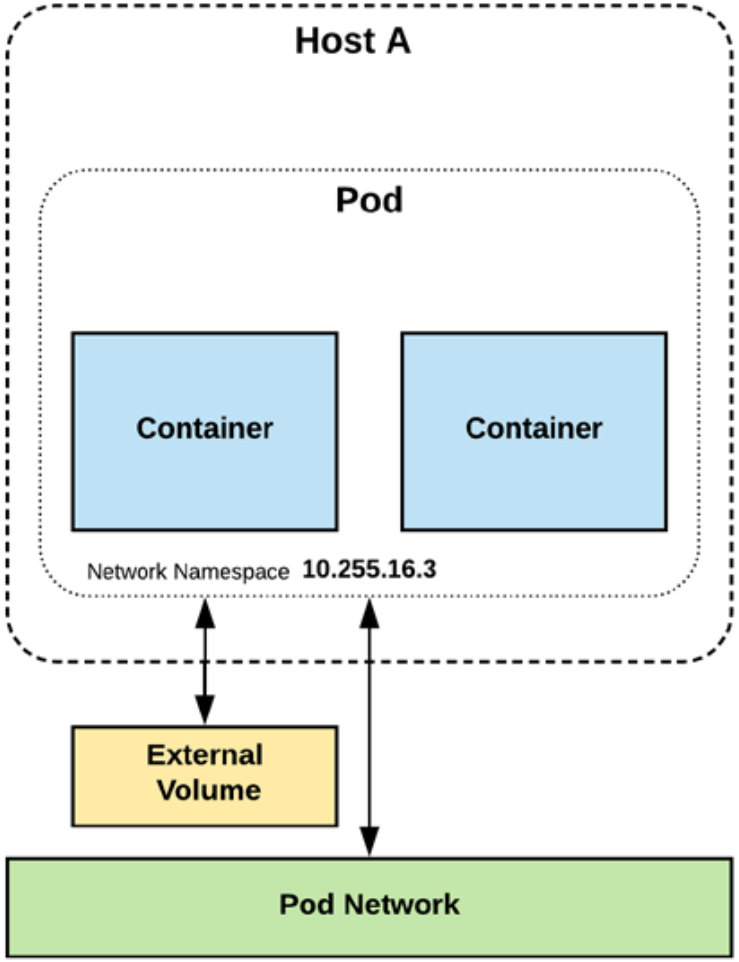
KUBE

SELECTED

OBJECTS

POD

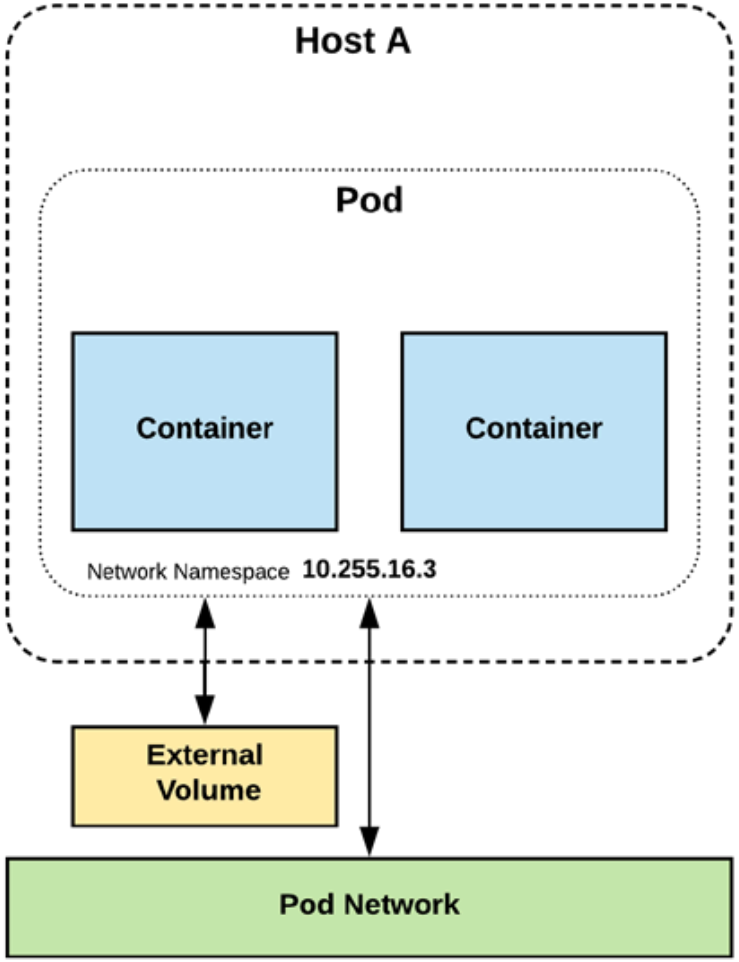
- smallest deployable units of computing
- Pods are one or MORE containers that share volumes, a network namespace, and are a part of a single context.div.
- pods are ephemeral



POD

You can use workload resources to create and manage multiple Pods: for example:

- ReplicaSet -> Deployment
- StatefulSet
- DaemonSet



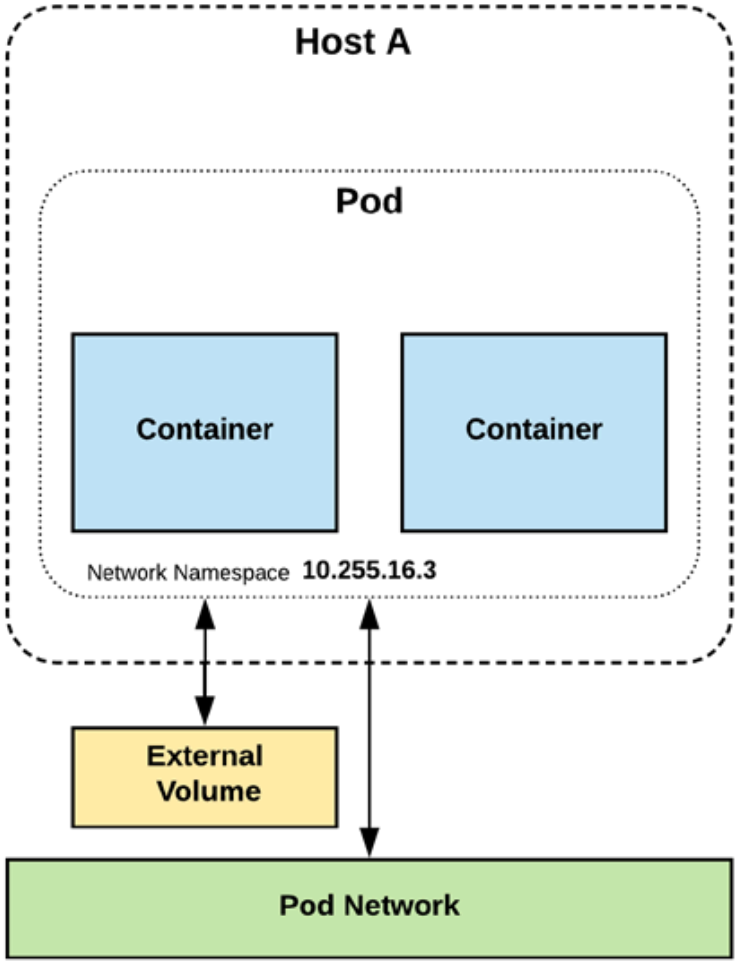
POD

- storage

  A Pod can specify a set of shared storage volumes. All containers in the Pod can access the shared volumes

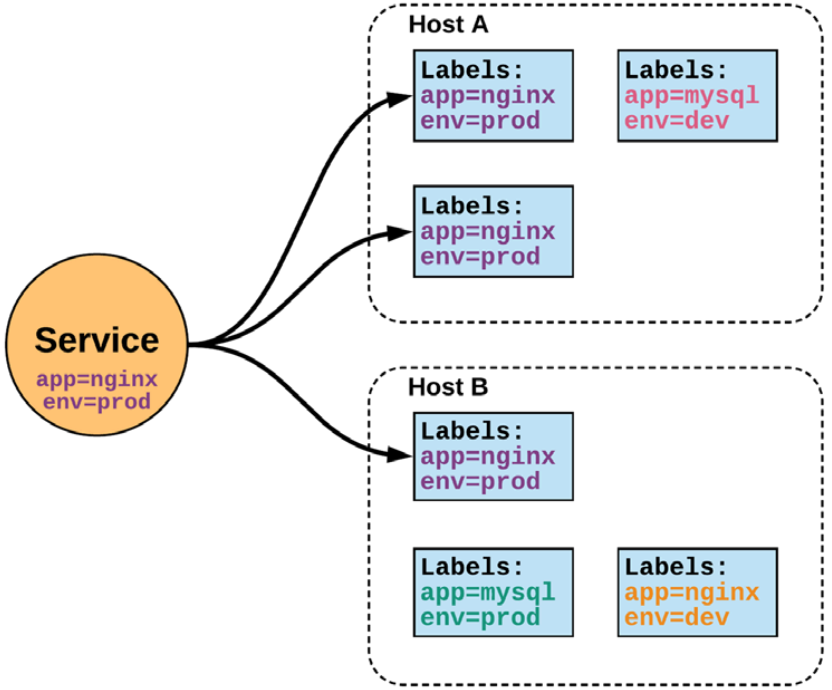
- network

  Each Pod is assigned a unique IP address for each address family. Every container in a Pod shares the network namespace, including the IP address and network ports.



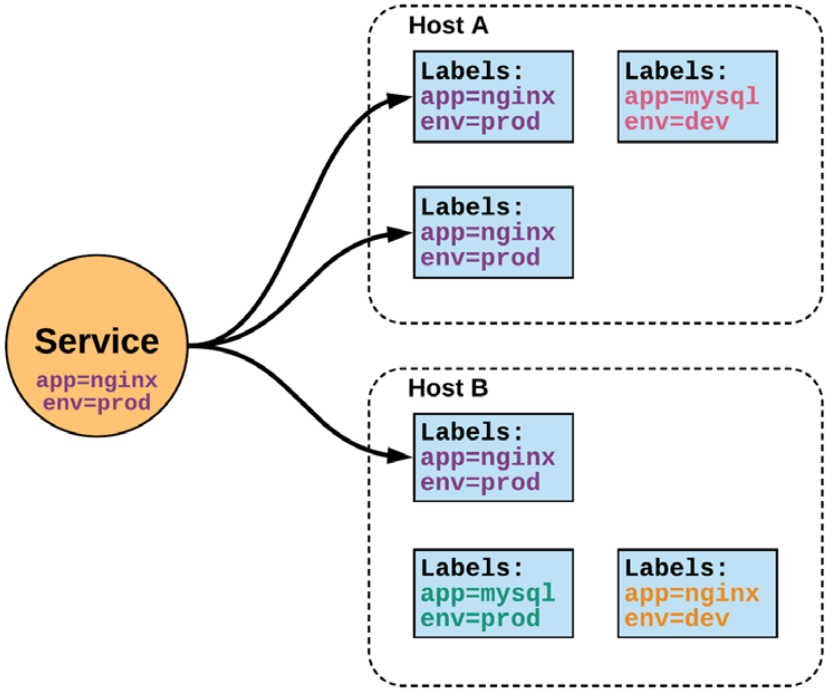
SERVICE

Expose an application running in your cluster behind a single outward-facing endpoint, even when the workload is split across multiple backends.



SERVICE

- Unified method of accessing the exposed workloads of Pods.
- Not ephemeral!



SERVICE

- Cluster IP
- Node port
- Load Balancer
- External Name

the Service on a cluster-internal IP.

SERVICE

- Cluster IP
- Node port
- Load Balancer
- External Name

the control plane allocates a port from a range. Each node proxies that port into your Service.

SERVICE

- Cluster IP
- Node port
- Load Balancer
- External Name

L2 loadblanacer, not available without a plugin (like a metallb)

SERVICE

- Cluster IP
- Node port
- Load Balancer
- External Name

Maps the Service to the contents of the externalName field.

STORAGE

Data in the pod is not persistent (pods are ephemeral)

to ensure data persistency: persistent volume (PV)

PV? Represent physical storage resources, such as a block storage volume or a network share

STORAGE CLASSES:

- storage should be dinamically provisioned
- you need an external service for the storage Volume Provisioning

CONFIGMAP/SECRET

Are the correct way to store static config files/ secrets

INGRESS: THE GATEWAY TO SERVICES

L3 LOAD BALANCER

exposes HTTP/S (usually)services to the outside world

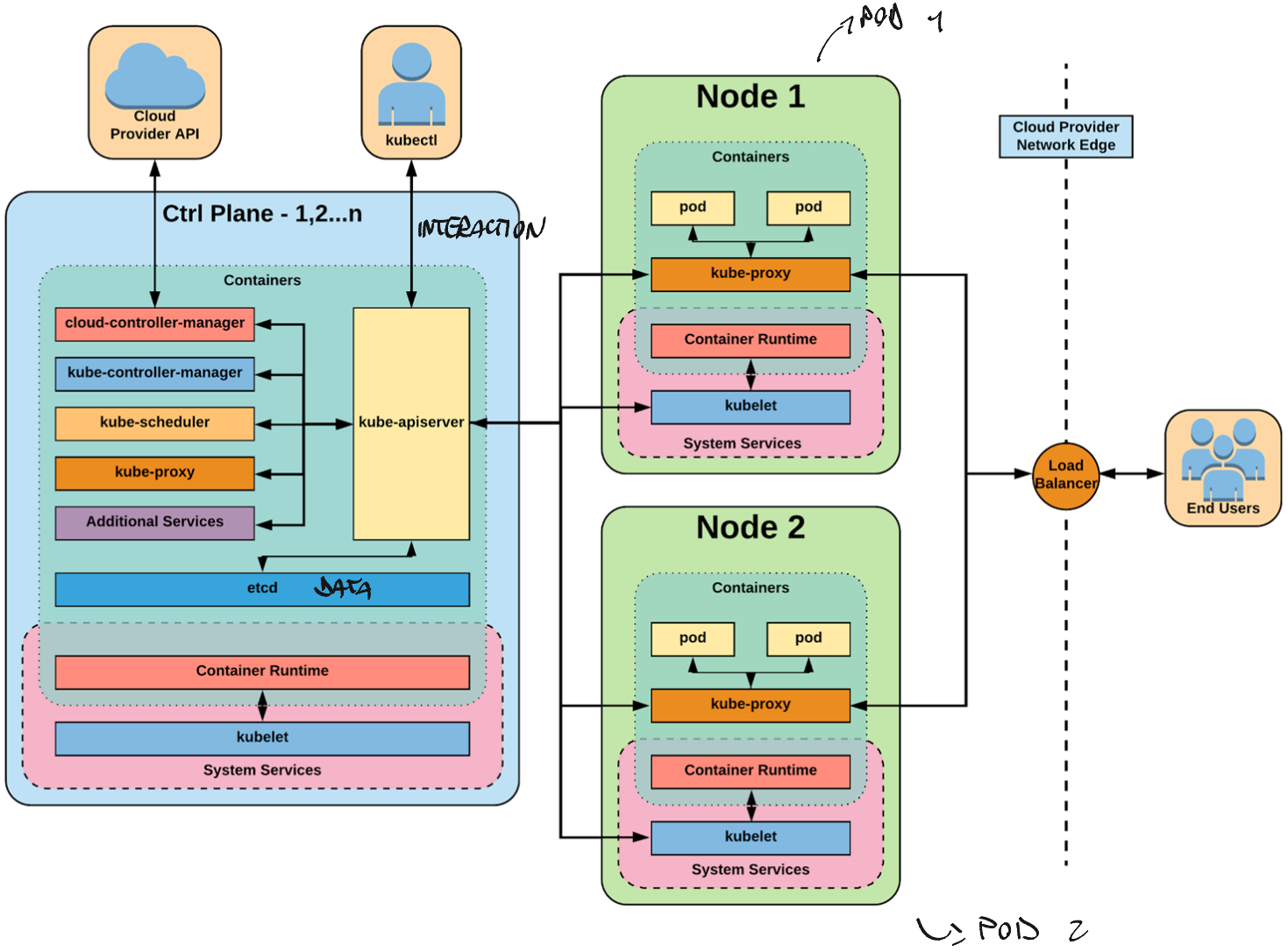
Acts as a reverse proxy, routing incoming traffic to different services based on URL paths, hostnames, or other criteria.

OBJECT EXAMPLE

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
```

containers:
- NAME : NGINX
  IMAGE: NGINX : 1.14.2
  PORTS:
  - CONTAINERPORTS : 72

APPLICATION ARCHITECTURE



TAKEAWAY MESSAGES

WHAT IS KUBERNETES?

A powerful open-source platform for automating deployment, scaling, and management of containerized applications.

TAKEAWAY MESSAGES

KEY BENEFITS:

- Simplified Deployment
- Automated Scaling
- Self-Healing
- Load Balancing
- Container Orchestration

TAKEAWAY MESSAGES

CORE CONCEPTS (AT LEAST TWO):

- Pods
- Services