# Kubernetes Installation and Configuration Fundamentals

## INTRODUCTION AND EXPLORING KUBERNETES ARCHITECTURE

**Kien Bui**
DevOps & Platform Engineer

# Course Overview

Introduction

Exploring Kubernetes Architecture

Installing and Configuring Kubernetes

Working with Your Kubernetes Cluster

# Overview

What is Kubernetes?

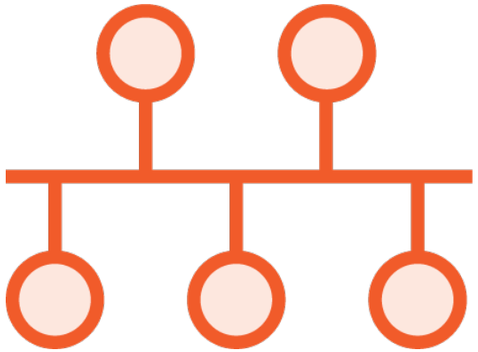
Exploring Kubernetes Architecture

- Cluster Components
- Networking Fundamentals

# What Is Kubernetes?

Container Orchestrator

Workload Placement



Infrastructure Abstraction

Desired State

# Benefits of Using Kubernetes

Speed of deployment

Ability to absorb change quickly

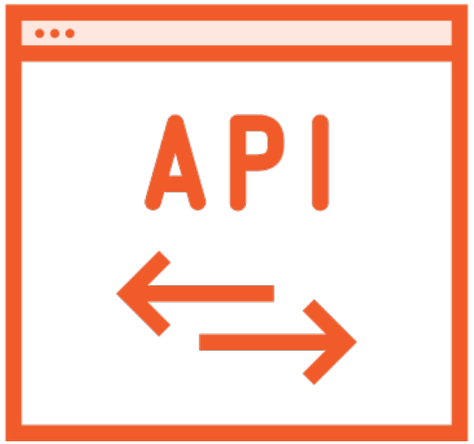
Ability to recover quickly

Hide complexity in the cluster

# Kubernetes Principles

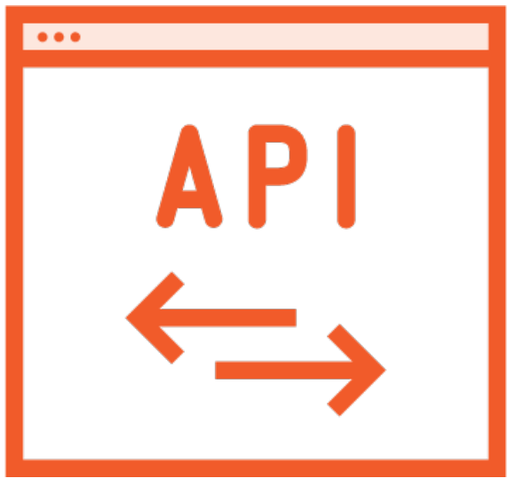
**Desired State/**
Declarative
Configuration

**Controllers/**
Control Loops




Kubernetes API/
The API Server

# Kubernetes API



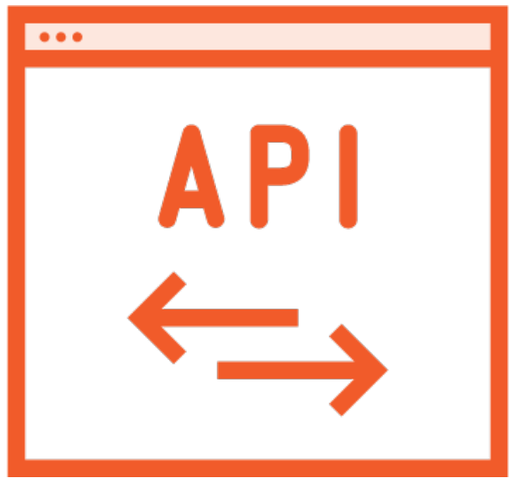

API Objects

- Collection of primitives to represent your system's state

Enables configuration of state

- Declaratively
- Imperatively

# Kubernetes API Server




RESTful API over HTTP using JSON

The sole way to interact with your cluster

The sole way Kubernetes interacts with your cluster

Serialized and persisted

# Kubernetes API Objects

Pods Controllers Services Storage

Not an exhaustive list, but these are the key players

# Pods

One or more containers

It’s your application or service

The most basic unit of work

Unit of scheduling

Ephemeral - no Pod is ever “redeployed”

Atomicity - they’re there or NOT

# Pods - Continued

Kubernetes' job is keeping your Pods running

More specifically keeping the desired state

State - is the Pod up and running

Health - is the application in the Pod running

Probes

# So how does Kubernetes manage my Pods' state?

# Controllers

Defines your desired state

Create and manage Pods for you

Respond to Pod state and health

`ReplicaSet`

- Number of replicas

`Deployment`

- Manage rollout of `ReplicaSets`

Many more…and not just Pods

# So how does Kubernetes add persistency to all this ephemerality?

# Services

Adds persistency to our ephemeral world

Networking abstraction for Pod access

IP and DNS name for the Service

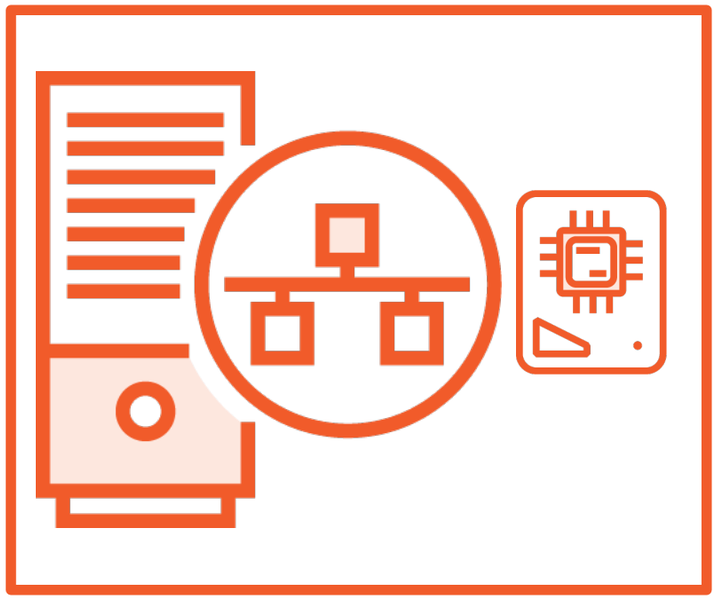
Dynamically updated based on Pod lifecycle

Scaled by adding/removing Pods

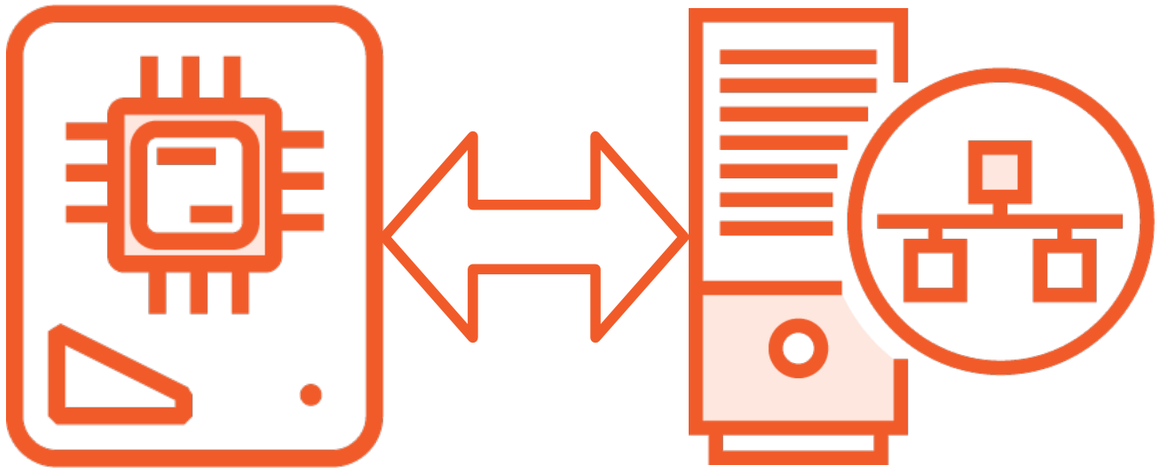
Load balancing

# What about my data?
# Where's that stored in Kubernetes?

# Storage in Kubernetes



Volumes



Persistent Volume

Persistent Volume Claim

# Exploring Kubernetes Architecture

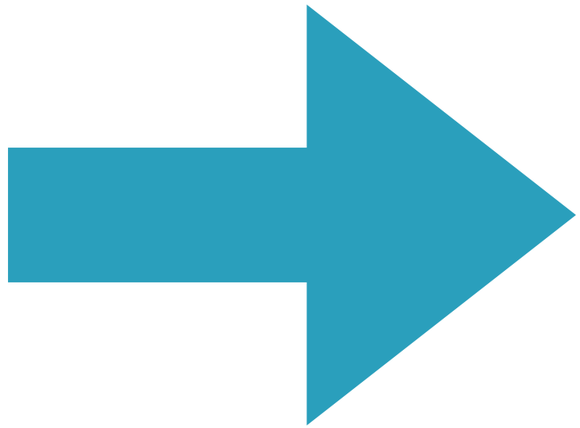
# Cluster Components

Control Plane Node

Node

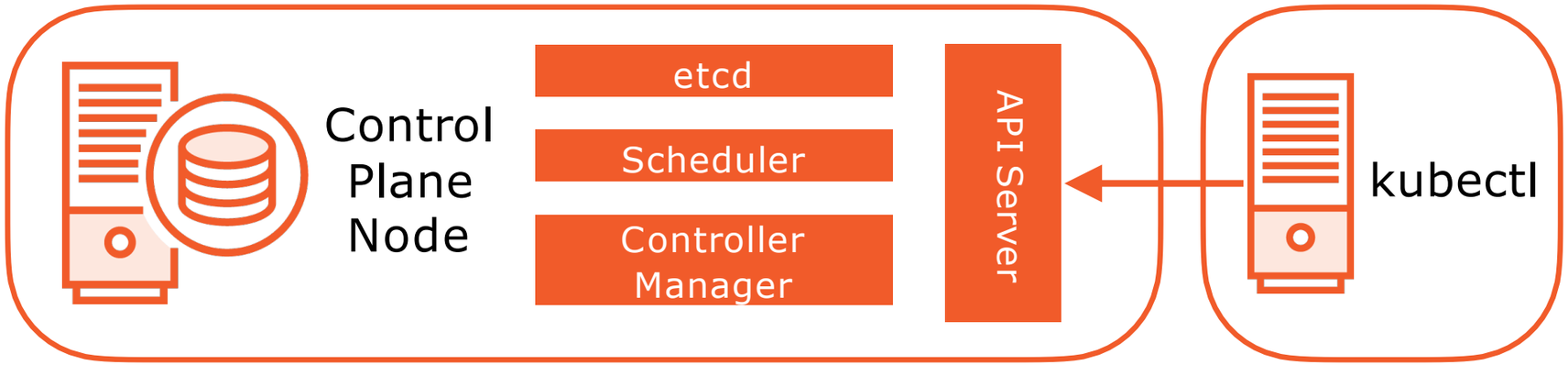
# Control Plane Node

Master Node



Control Plane Node

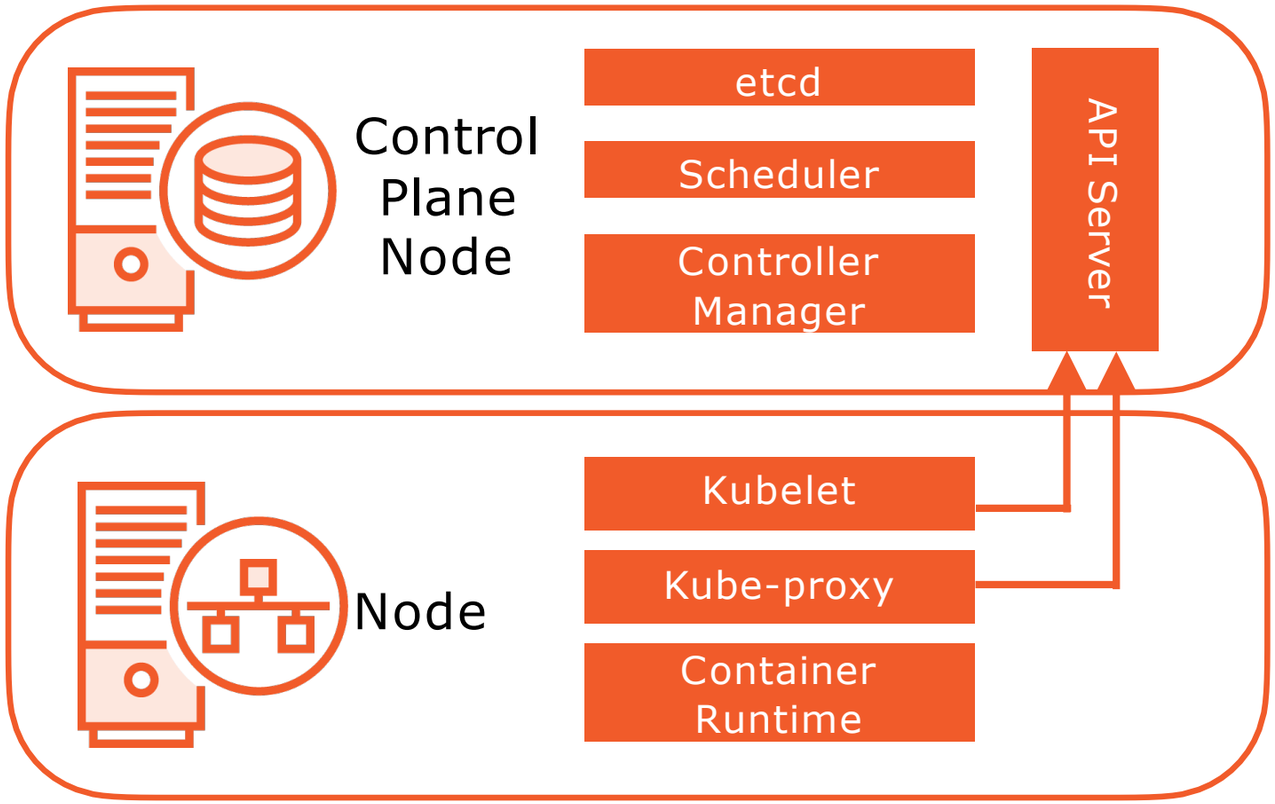
# Control Plane Node



# Control Plane Components

| API Server | etcd | Scheduler | Controller Manager |
| --- | --- | --- | --- |
| Central | Persists State | Watches API Server | Controller Loops |
| Simple | API Objects | Schedules Pods | Lifecycle functions and desired state |
| RESTful | Key-value | Resources | Watch and update the API Server |
| Updates etcd | | Respects contraints | ReplicaSet |

# Nodes


Control
Plane
Node
etcd
Scheduler
Controller
Manager
API Server
Node
Kubelet
Kube-proxy
Container
Runtime

On All Nodes!

# Nodes

## Kubelet

- Monitors API Server for changes
- Responsible for Pod Lifecycle
- Reports Node & Pod state
- Pod probes

## kube-proxy

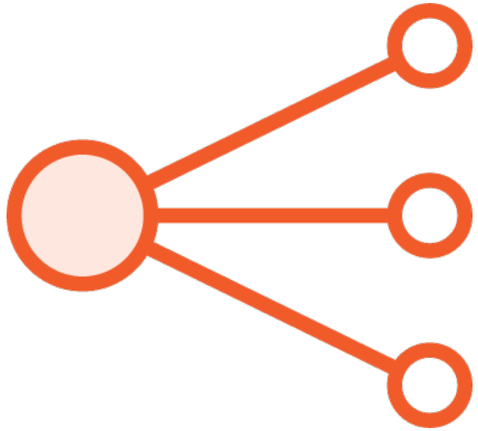
- iptables
- Implements Services
- Routing traffic to Pods
- Load Balancing

## Container Runtime

- Downloads images & runs containers
- Container Runtime Interface (CRI)
- containerd
- Many others…

# Cluster Add-on Pods

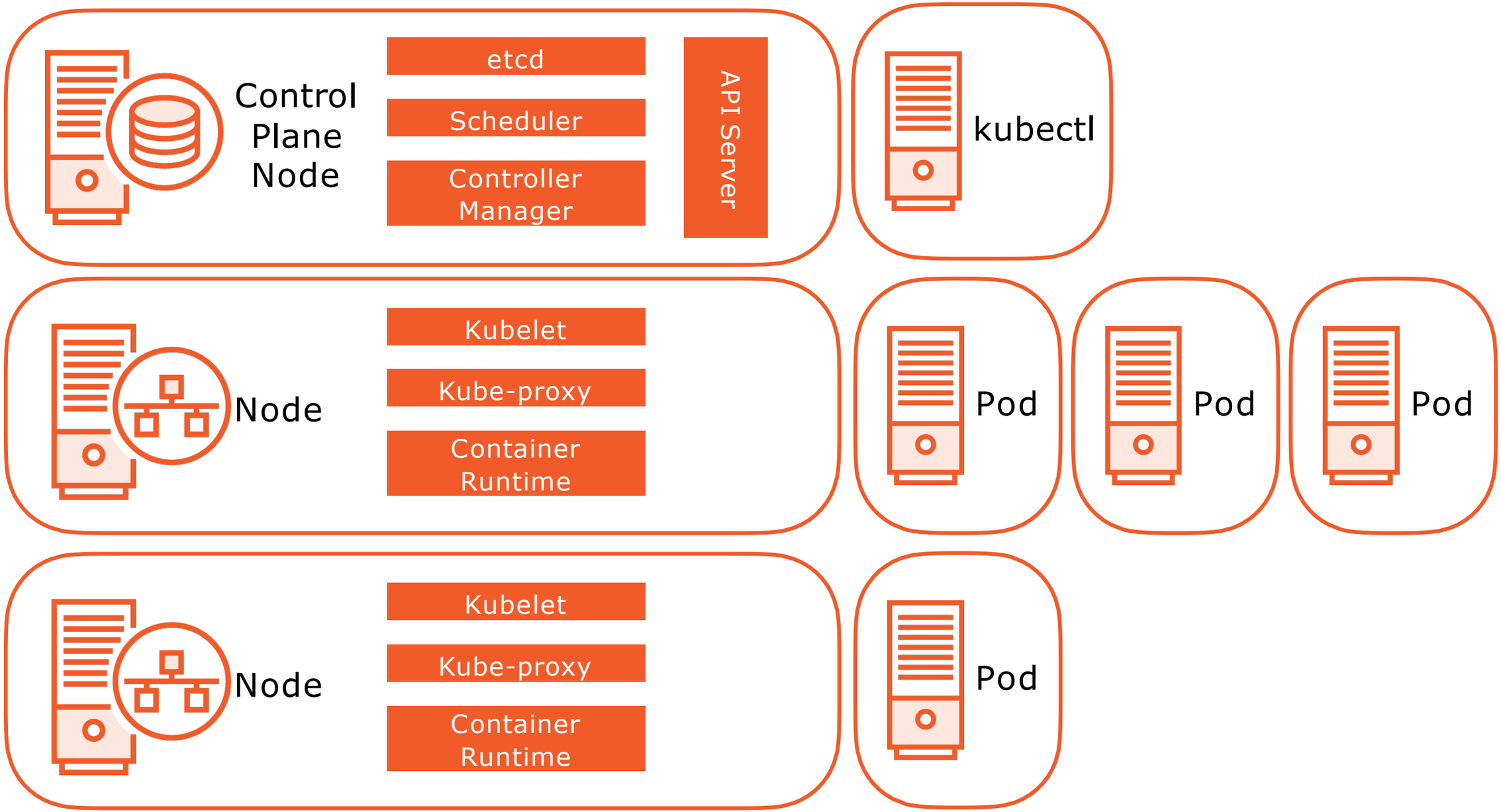

DNS



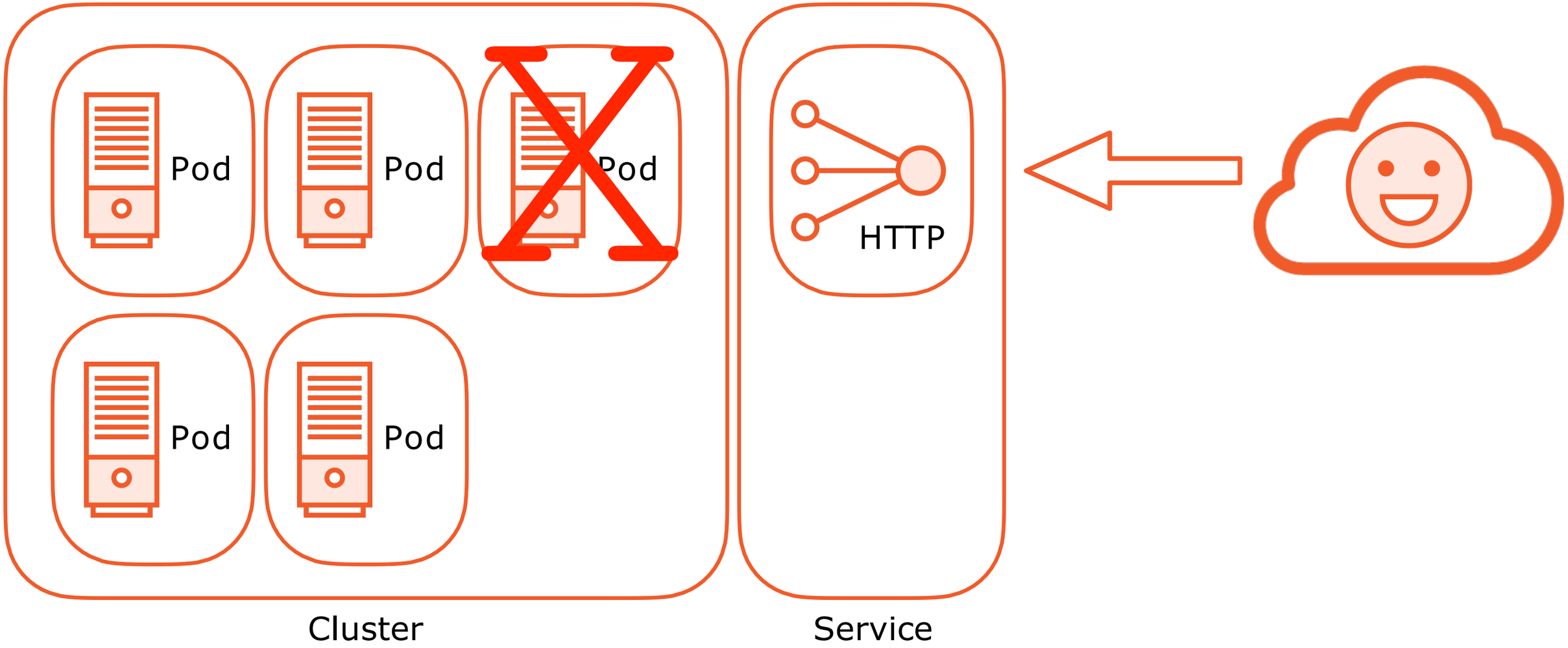
Ingress

Dashboard

# Pod Operations



# Services



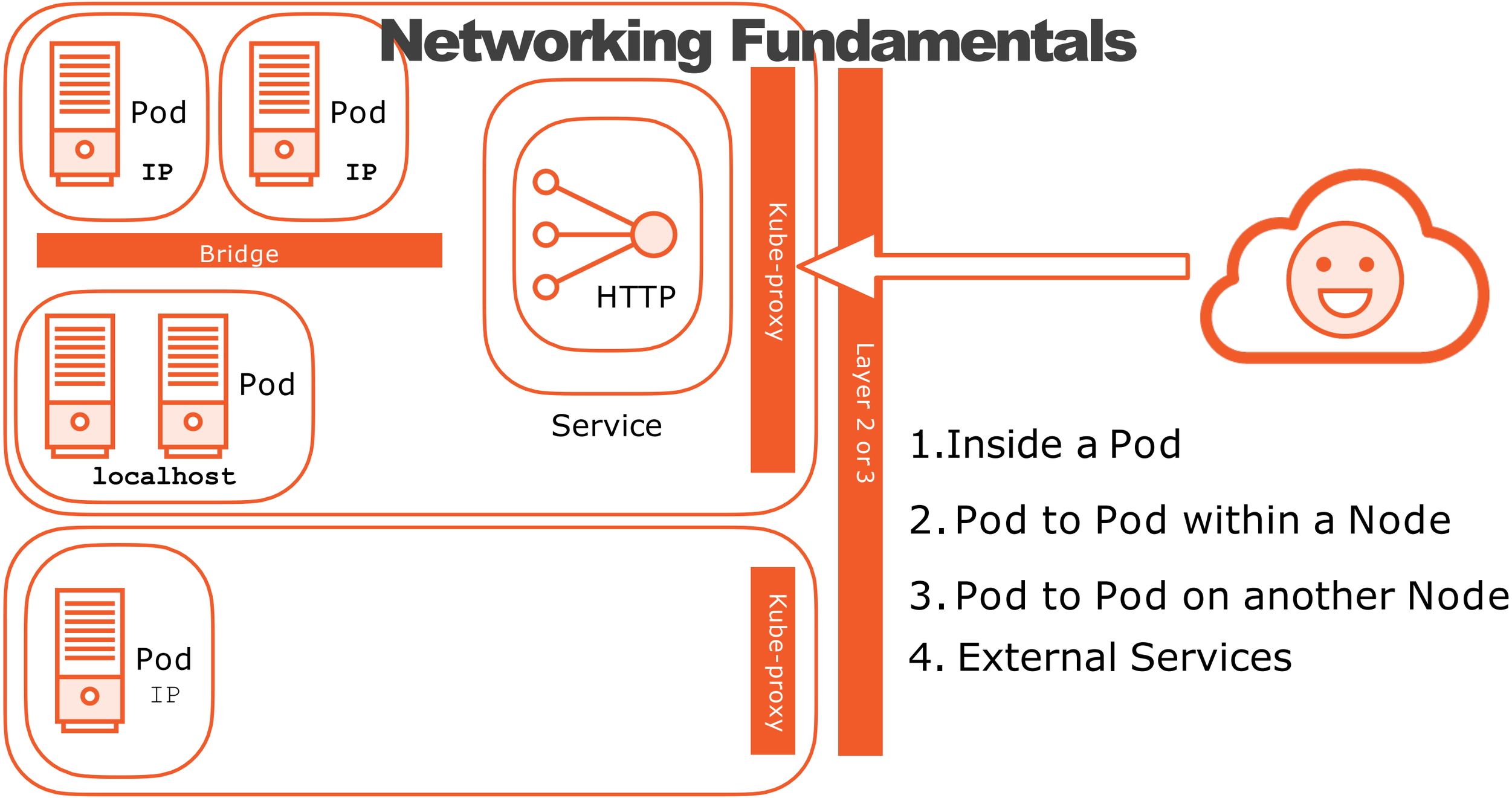
# Kubernetes Networking Fundamentals

# Kubernetes Networking Requirements

Pods on a Node can communicate with all Pods on all Nodes without Network Address Translation (NAT)

Agents on a Node can communicate with all Pods on that Node

# Networking Fundamentals




1. Inside a Pod
2. Pod to Pod within a Node
3. Pod to Pod on another Node
4. External Services

# Summary

What is Kubernetes?

Exploring Kubernetes Architecture

- Cluster Components
- Networking Fundamentals

# What's Next!

Installing and Configuring Kubernetes