Kubernetes Installation and Configuration Fundamentals

INTRODUCTION AND EXPLORING KUBERNETES ARCHITECTURE

Anthony E. Nocentino
ENTERPRISE ARCHITECT @ CENTINO SYSTEMS

@nocentino www.centinosystems.com

Course Overview

Introduction

Exploring Kubernetes Architecture

Installing and Configuring Kubernetes

Working with Your Kubernetes Cluster

Overview

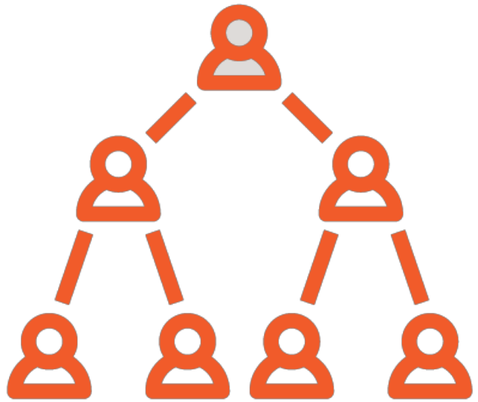
What is Kubernetes?

Exploring Kubernetes Architecture

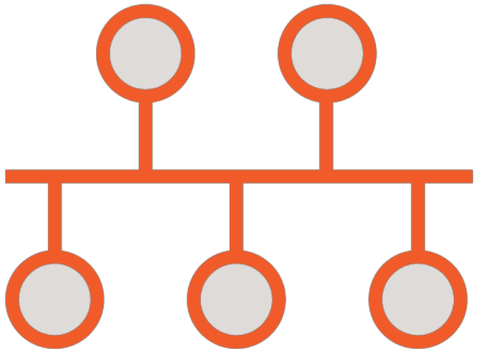
- Cluster Components
- Networking Fundamentals

What Is Kubernetes?

Container Orchestrator



Workload Placement



Infrastructure Abstraction

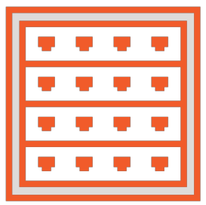
Desired State

Benefits of Using Kubernetes

Speed of deployment

Ability to absorb change quickly

Ability to recover quickly

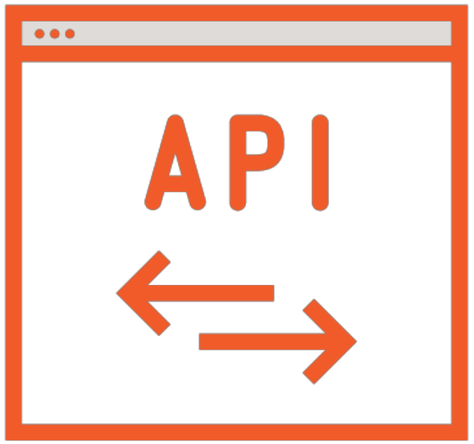


Hide complexity in the cluster

Kubernetes Principles

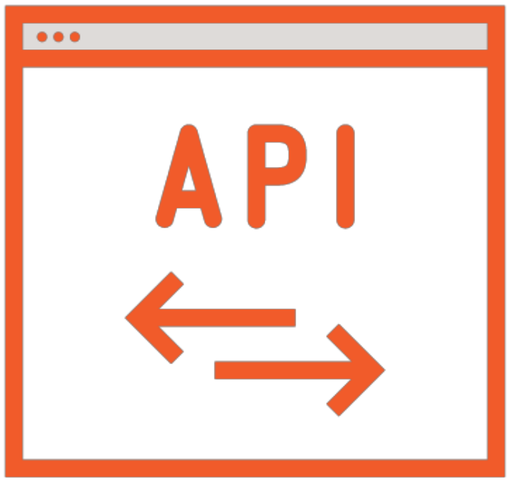
Desired State
Declarative
Configuration

Controllers
Control Loops



One Master
The API Server

Kubernetes API



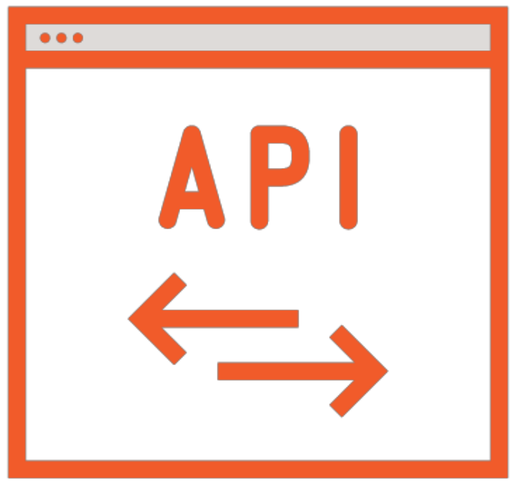
API Objects

- **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

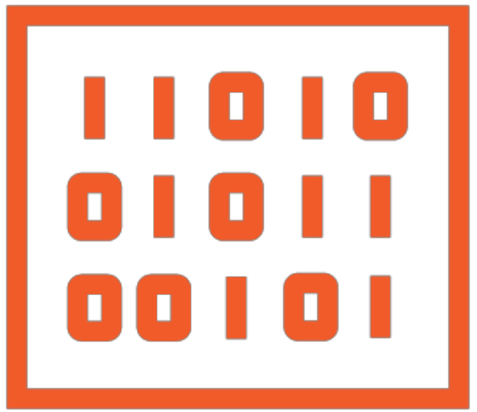
Serialized and persisted

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
- Liveness probes

So how does Kubernetes manage my Pods' state?

Controllers

Create and manage Pods for you

Define your desired state

Respond to Pod state and health

`ReplicaSet`

- **Number of replicas**

`Deployment`

- **Manage rollout of `ReplicaSet`**

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

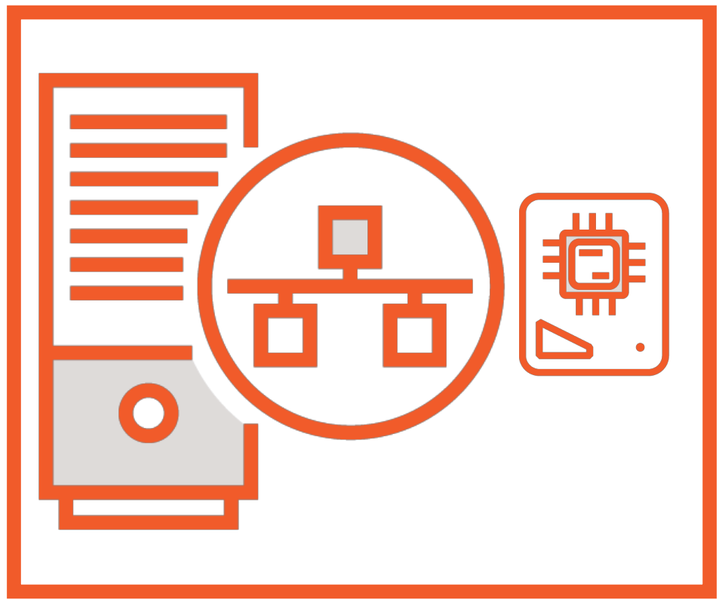
Redeployed Pods automatically updated

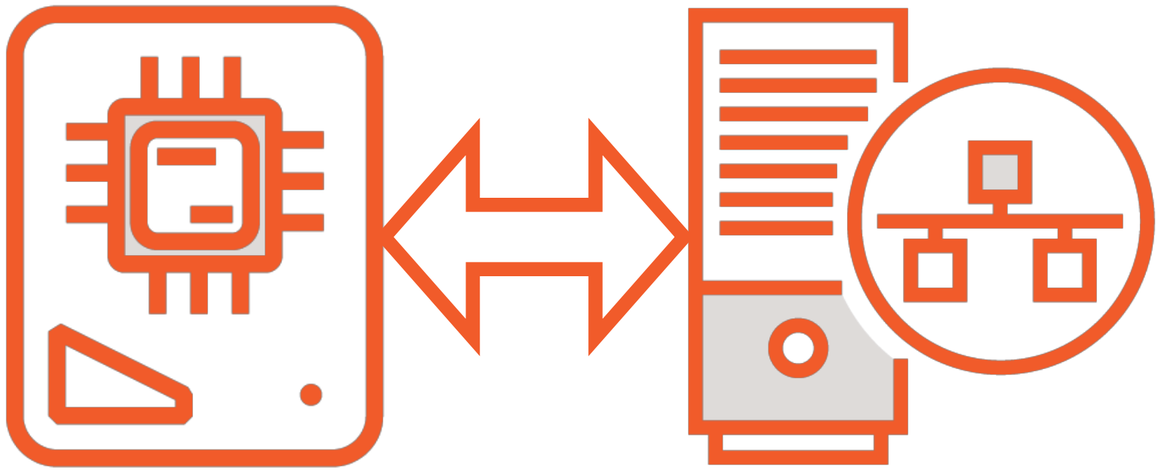
Scaled by adding/removing Pods

Load balancing

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

Storage in Kubernetes





Volumes

Persistent Volume

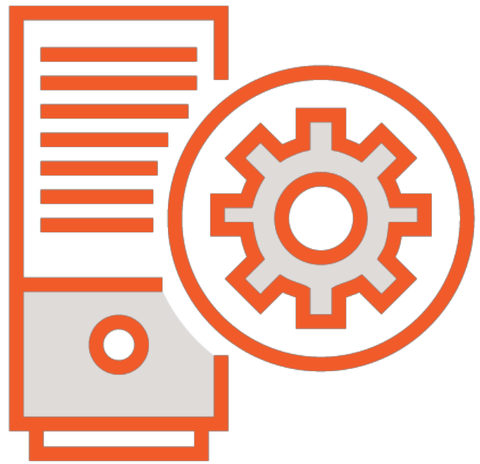
Persistent Volume Claim

Exploring Kubernetes Architecture

Cluster Components

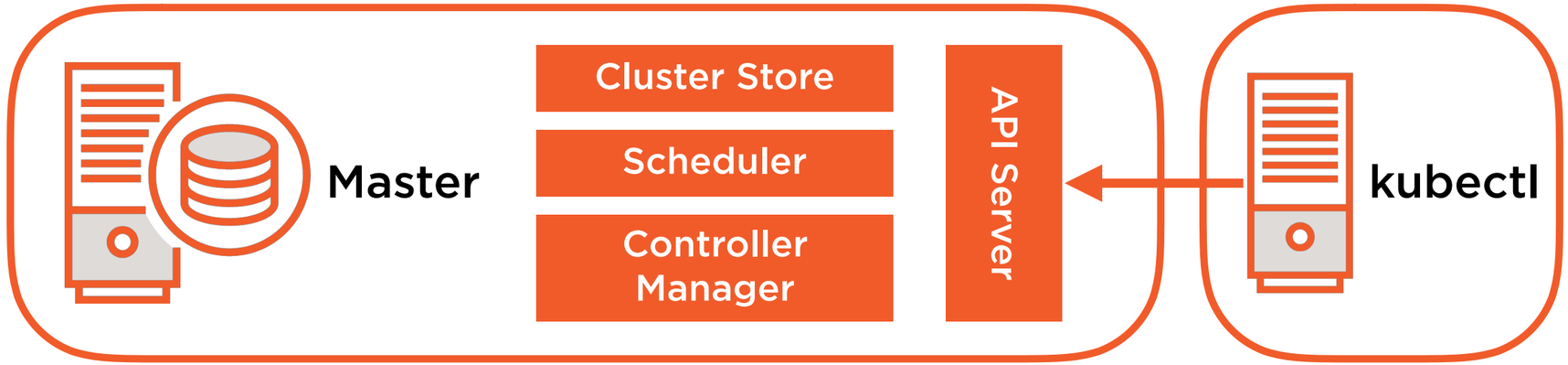
Master

Node



Scheduled/Add Ons

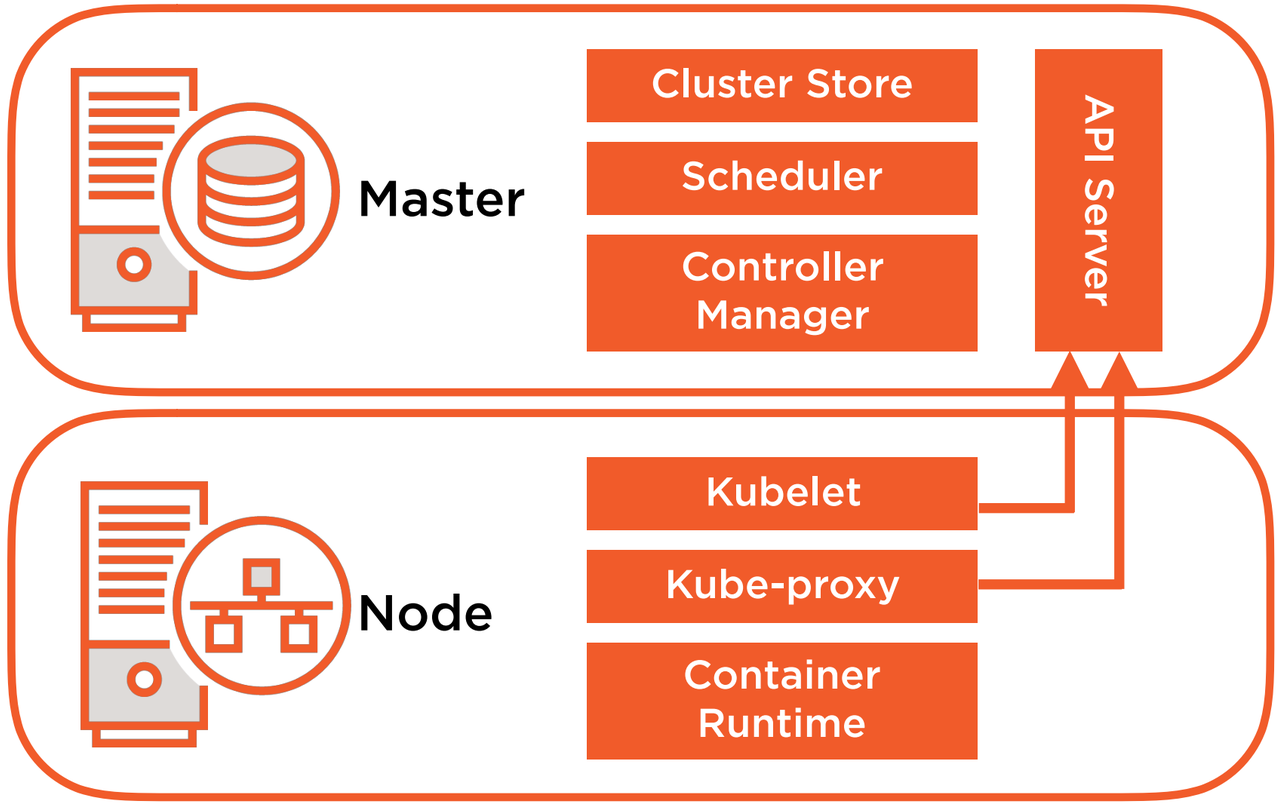
Master - Control Plane Components



Master - Control Plane Components

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

Nodes


Master
Cluster Store
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 liveness probes

kube-proxy

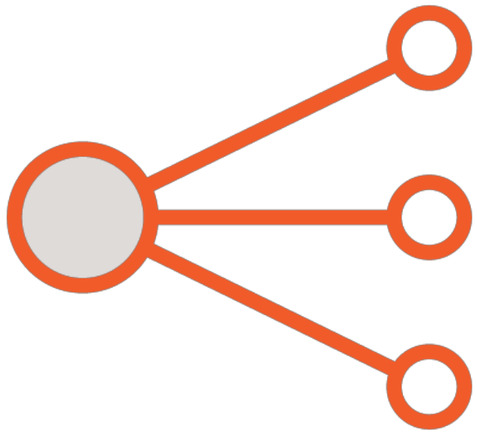
- Network proxy iptables
- Implements Services
- Routing traffic to Pods
- Load Balancing

Container Runtime

- Downloads images & runs containers
- Container Runtime Interface
- Docker
- Many others...

Scheduled/Add-On Pods

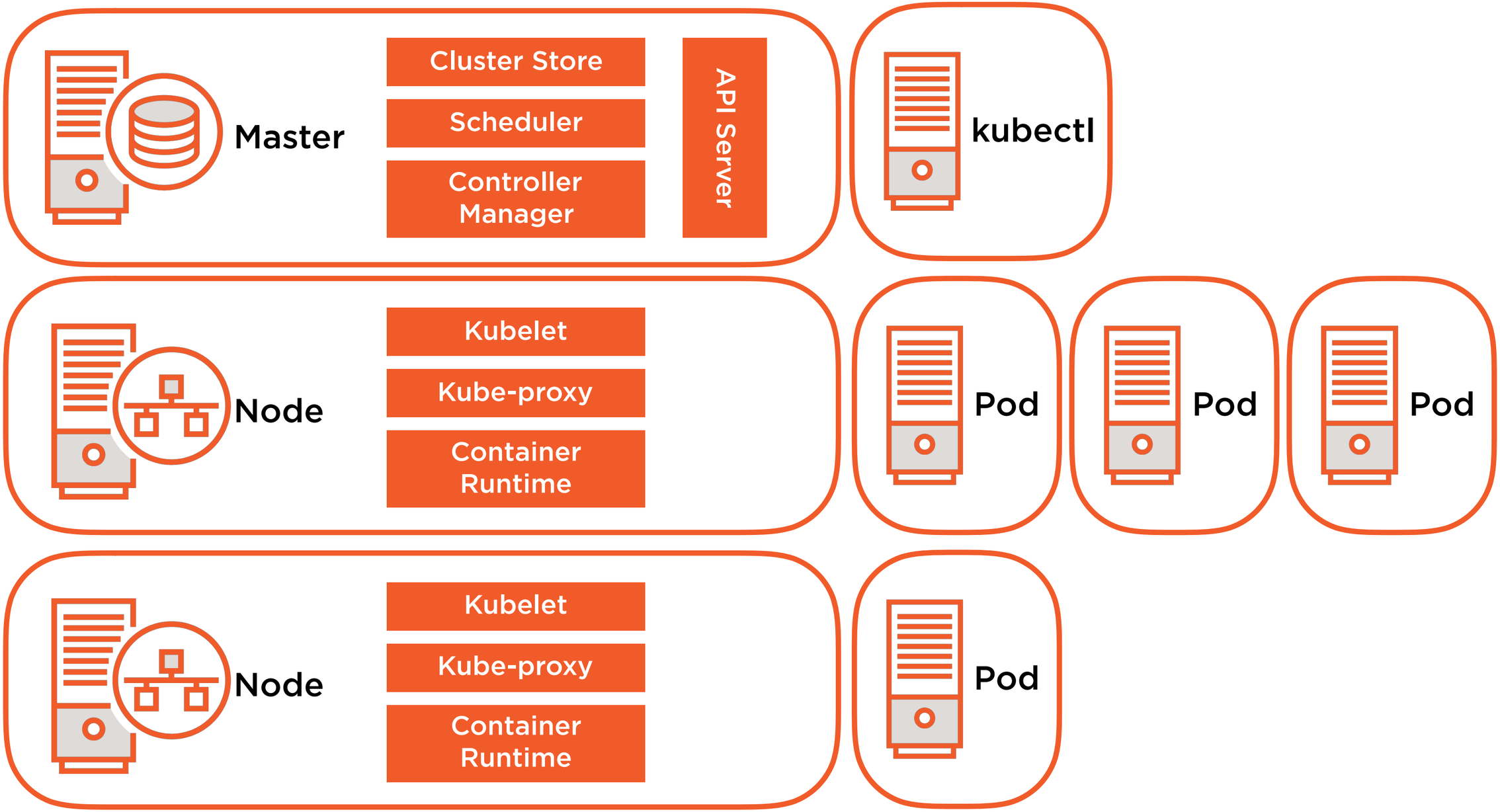
DNS



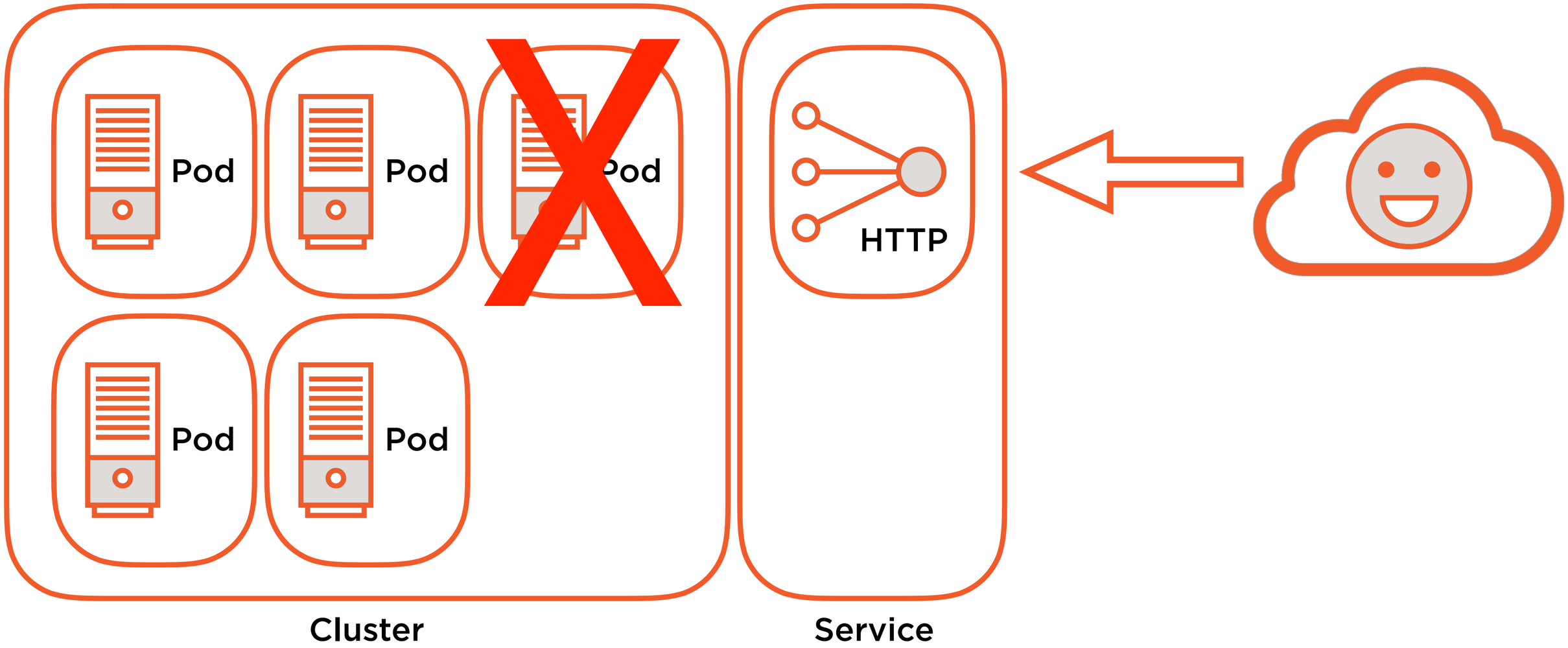
Ingress

Dashboard

Pod Operations



Services



Kubernetes Networking Fundamentals

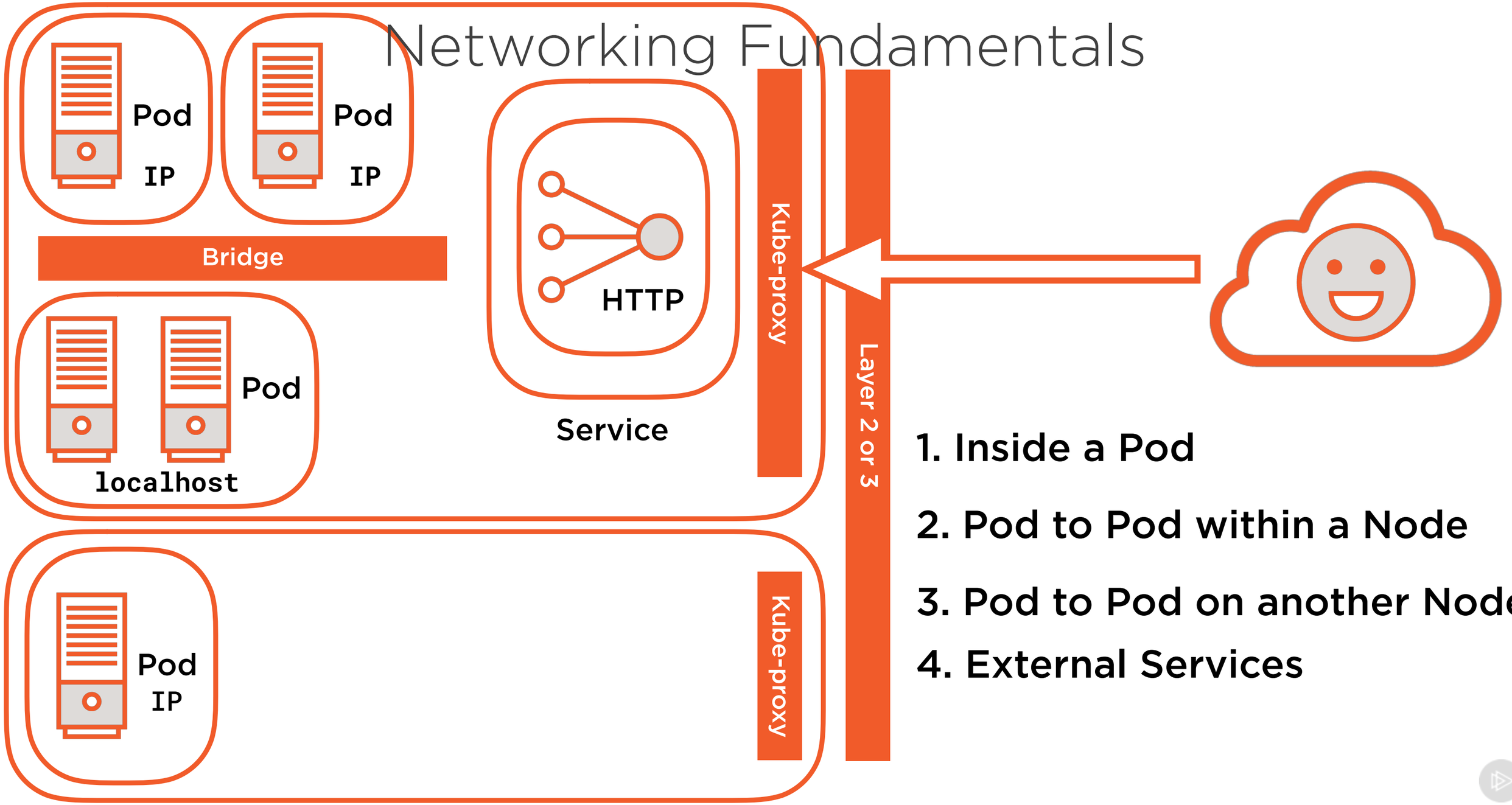
Kubernetes Networking Requirements

All Pods can communicate with each other on all Nodes

All Nodes can communicate with all Pods

No Network Address Translation (NAT)

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