nic

Future Edition

Kubernetes 101

Kubernetes 101

Damian Flynn

MVP Cloud & Datacenter + Cisco Champion

Lumagate

@Damian_Flynn

www.DamianFlynn.Com

nic

Introductions

- Where did it come from
- What does the name mean
- What does it do
- Why do we have it

Introductions

- Born in Google
- 2014, dontanted to the CNCF
- Composed in Go/Golang
- Online Community
  - Git: ghttps://github.com/kubernetes/kubnernetes
  - Twitter: @kubernetesio
  - Slack: slack.k8s.io

Introductions

Borg
Proprietary

Omega
Proprietary

Kubernetes
Open Source

Introductions

Kubernetes

- Greek for “Helmsman”
  - The person responsible for steering the ship

Introductions

~~Kubernetes~~

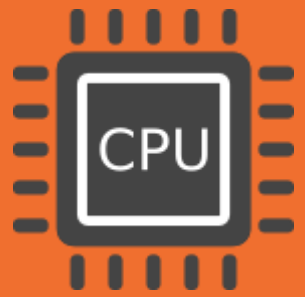
K8s

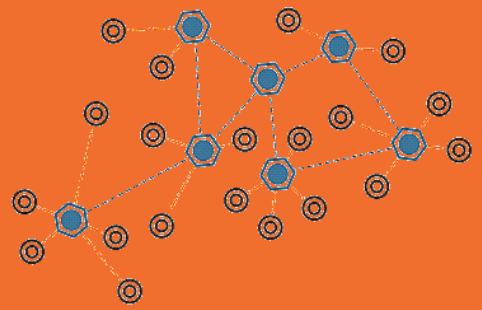
Introductions

- Opensource (Apache 2.0 License)
- Shortened to ‘K8s’
- Based on Googles Internal tools, Borg and Omega
- Donated to CNCF in 2014
- V1.0 release in July 2015
- Current release V1.8

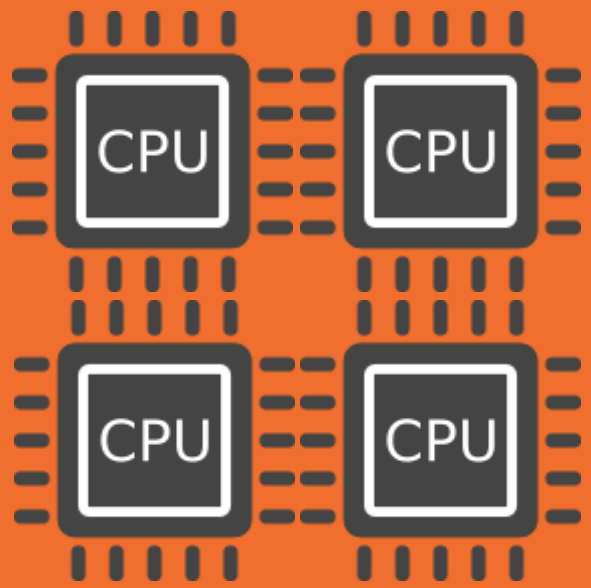
Containers

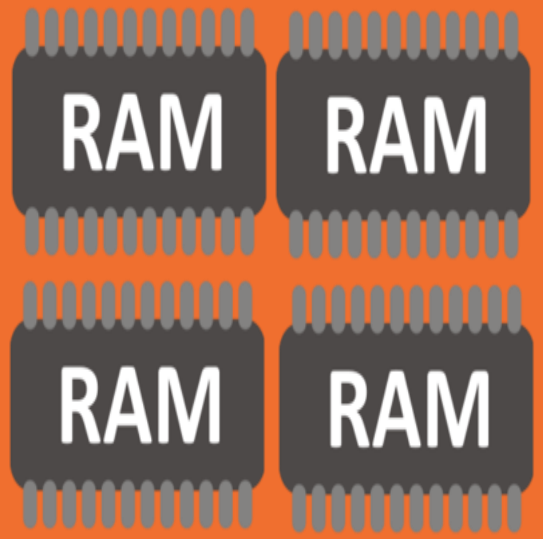
Scalability Challanges

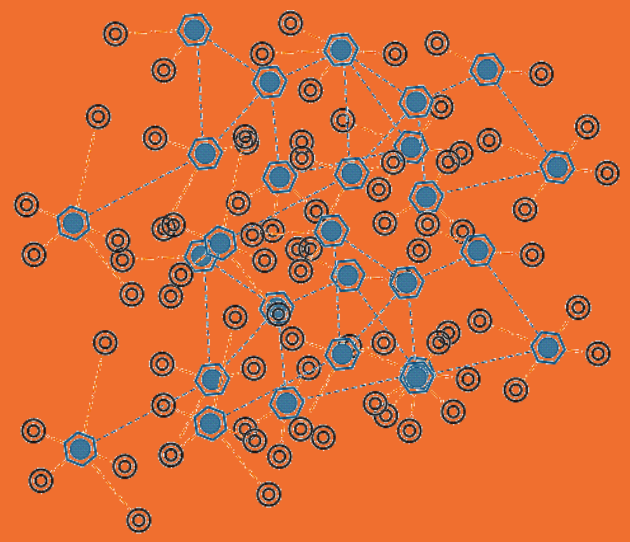




Viewing the Data Center as a Computer







Viewing the Data Center as a Computer

Kubernetes can manage it

What is it?

- Standard Package format
- Packaging Manifest
- Submit for Delivery

What is it?

- Evolving technology still considered Early days
- Strongly positioned
- Extremely platform agnostic
- Targeted Deployments
- Worth the learning curve

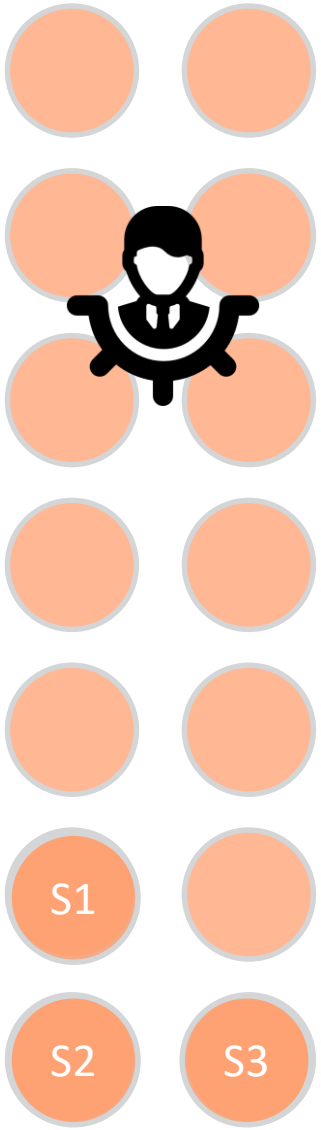
What is it?

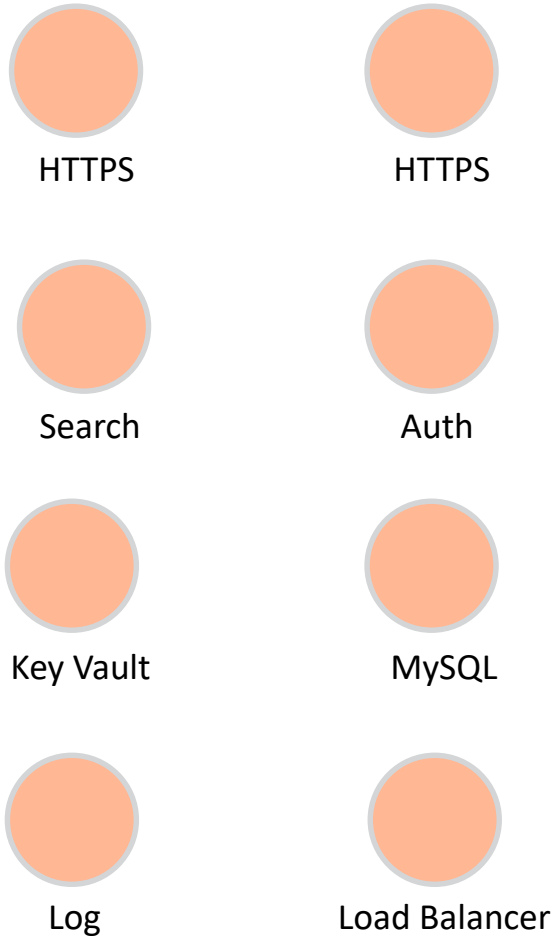
Kubernetes is Evolving FAST!

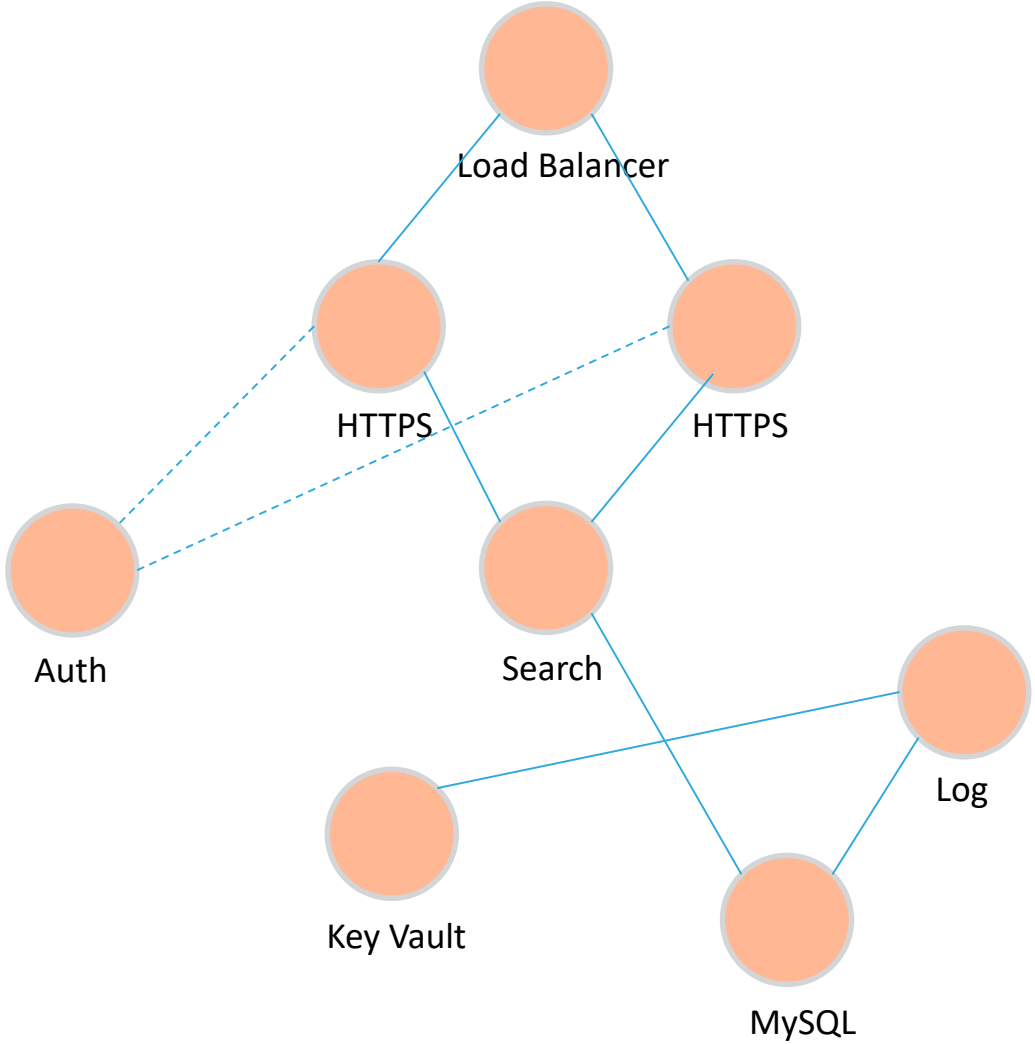
Kubernetes

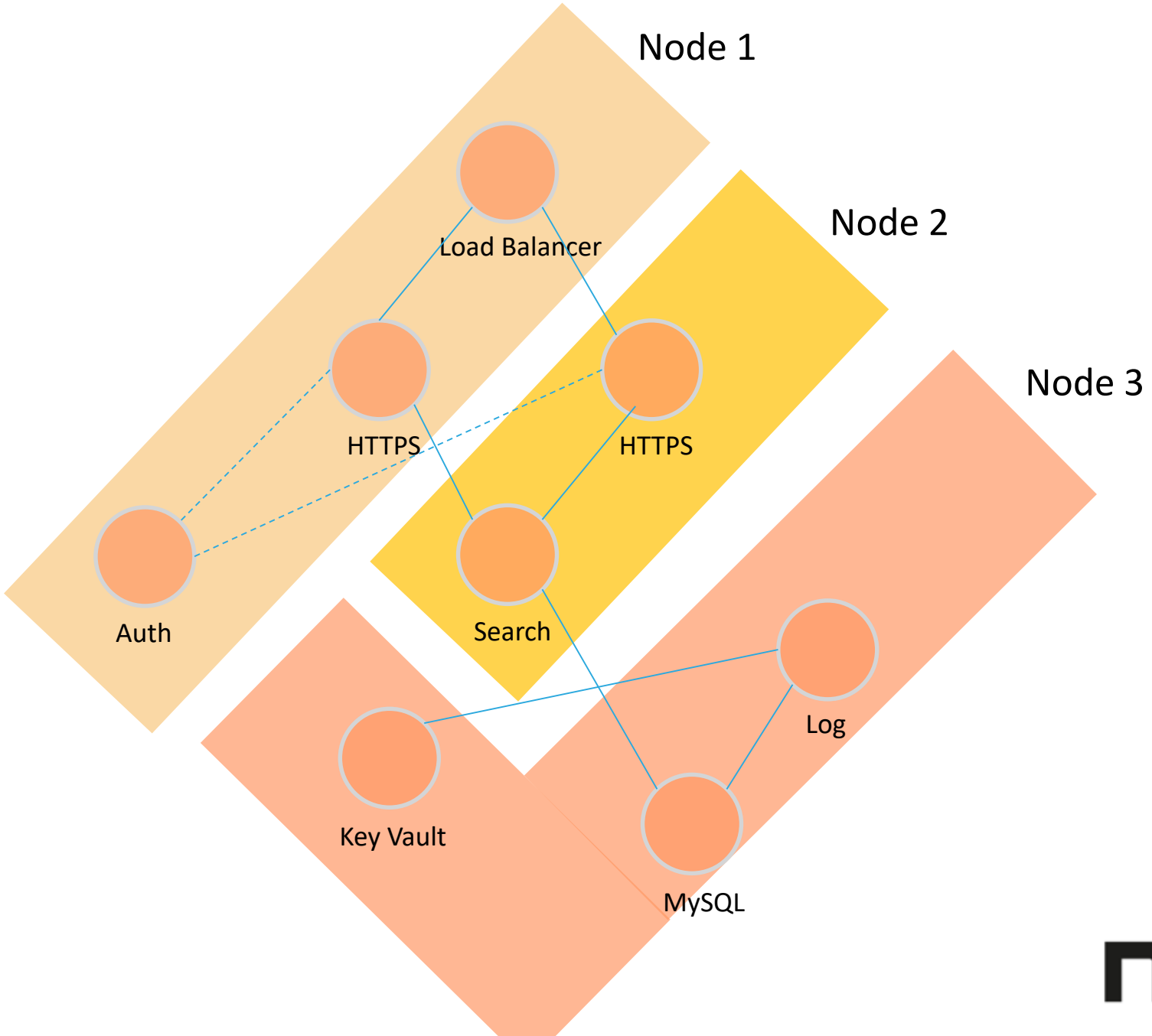
Micro Service Apps Orchestrator

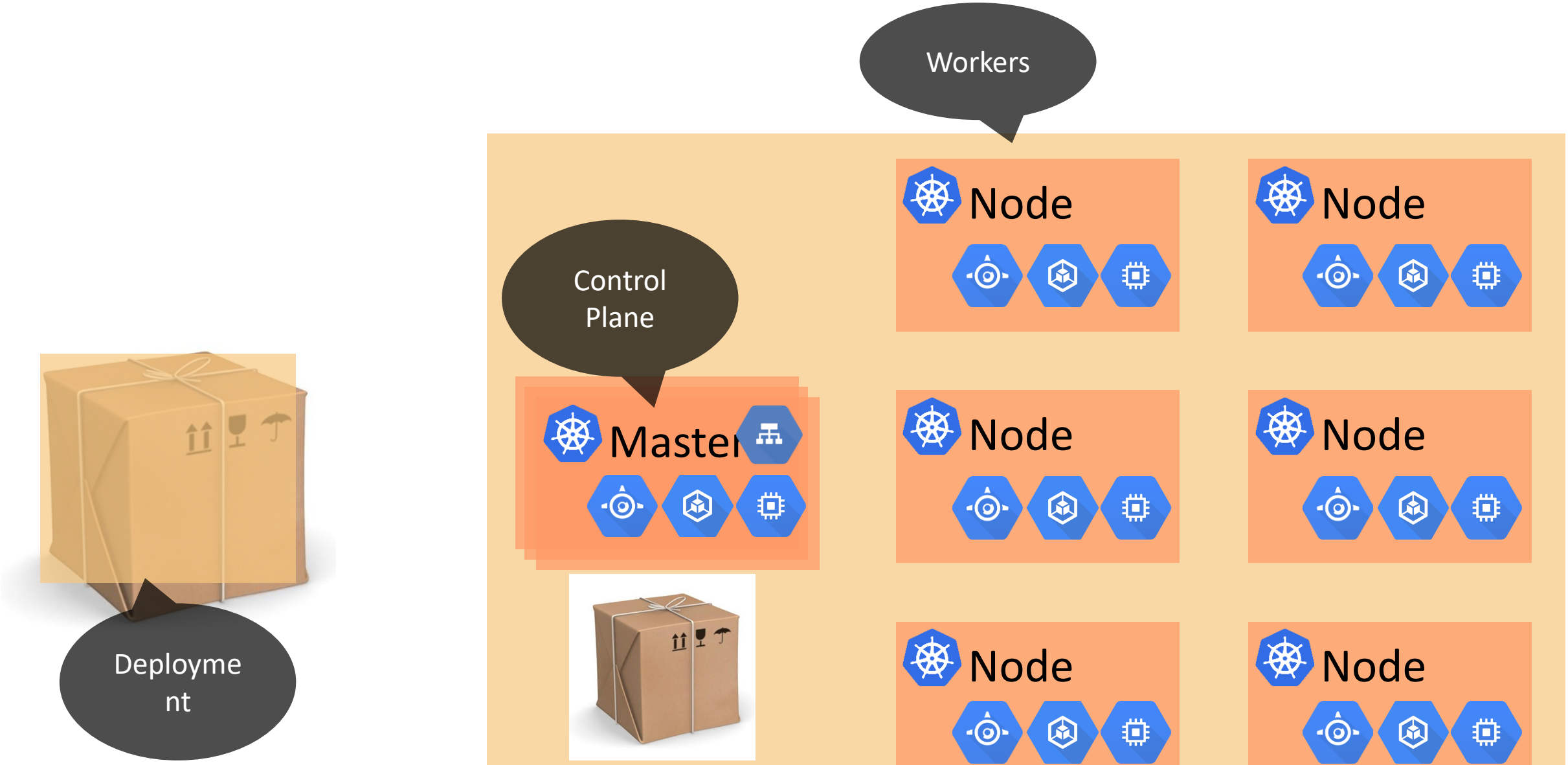
Team




HTTPS
HTTPS
Search
Auth
Key Vault
MySQL
Log
Load Balancer

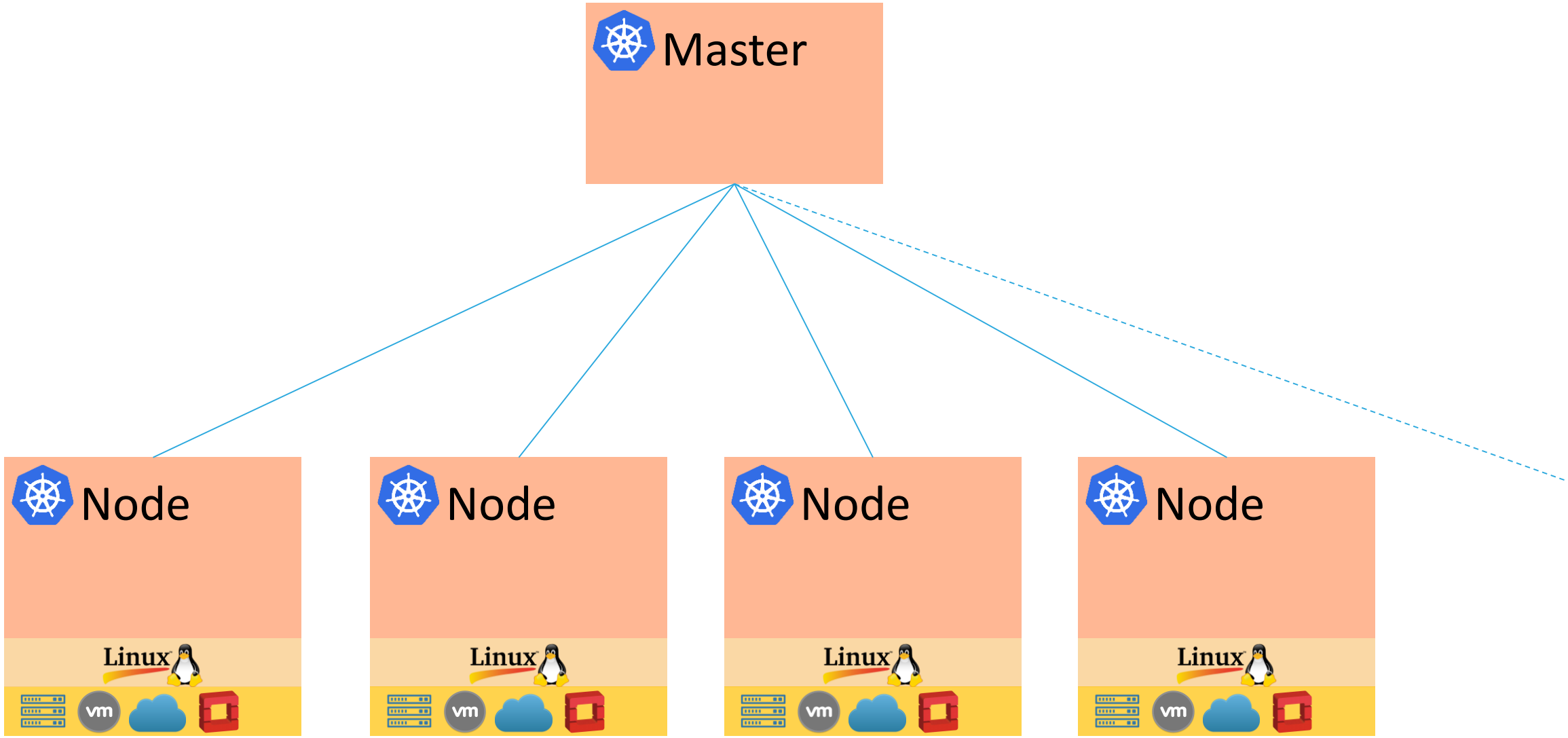

Load Balancer
HTTPS
HTTPS
Auth
Search
Log
Key Vault
MySQL

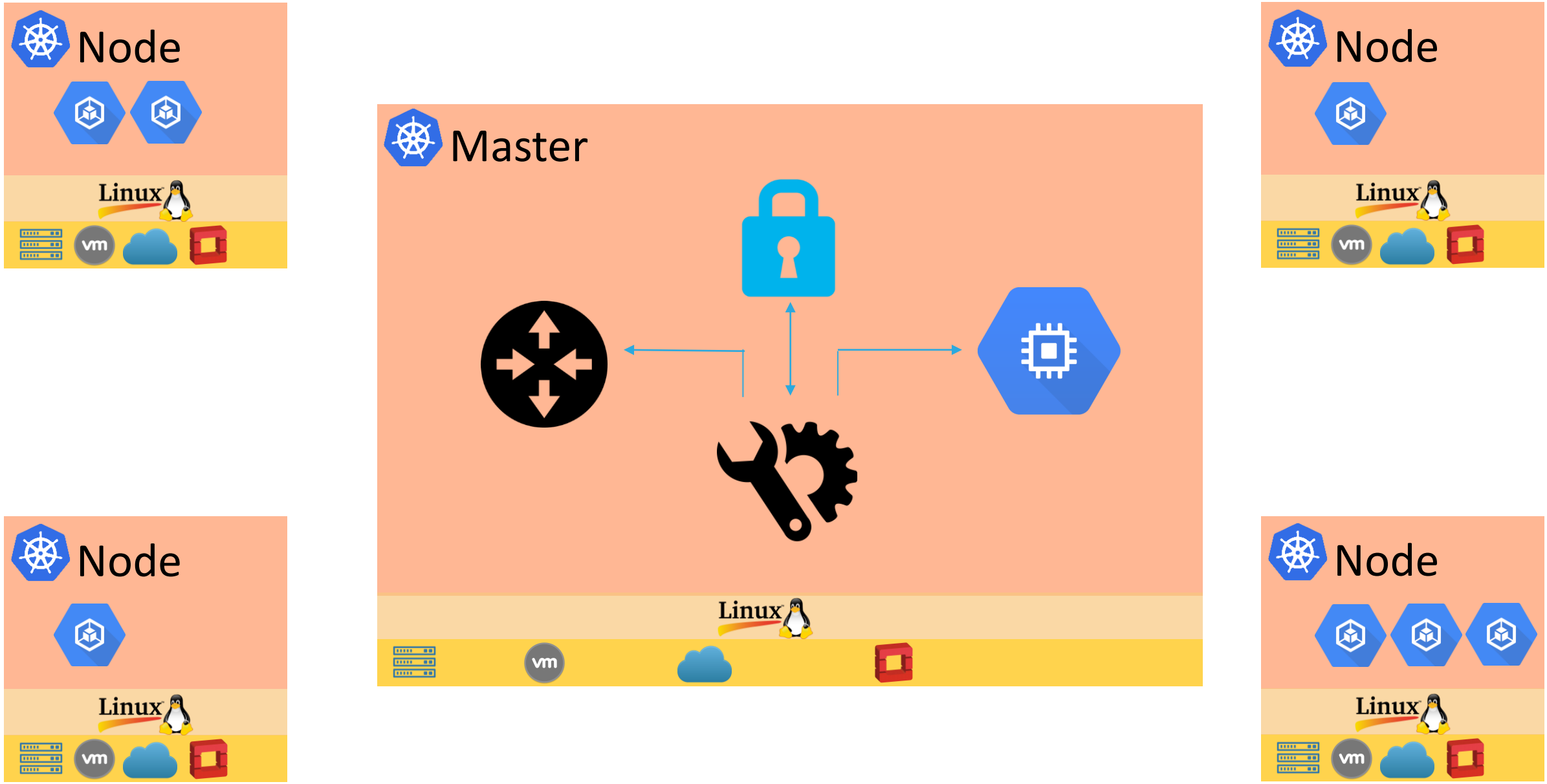

Node 1
Node 2
Node 3
Load Balancer
HTTPS
HTTPS
Auth
Search
Log
Key Vault
MySQL

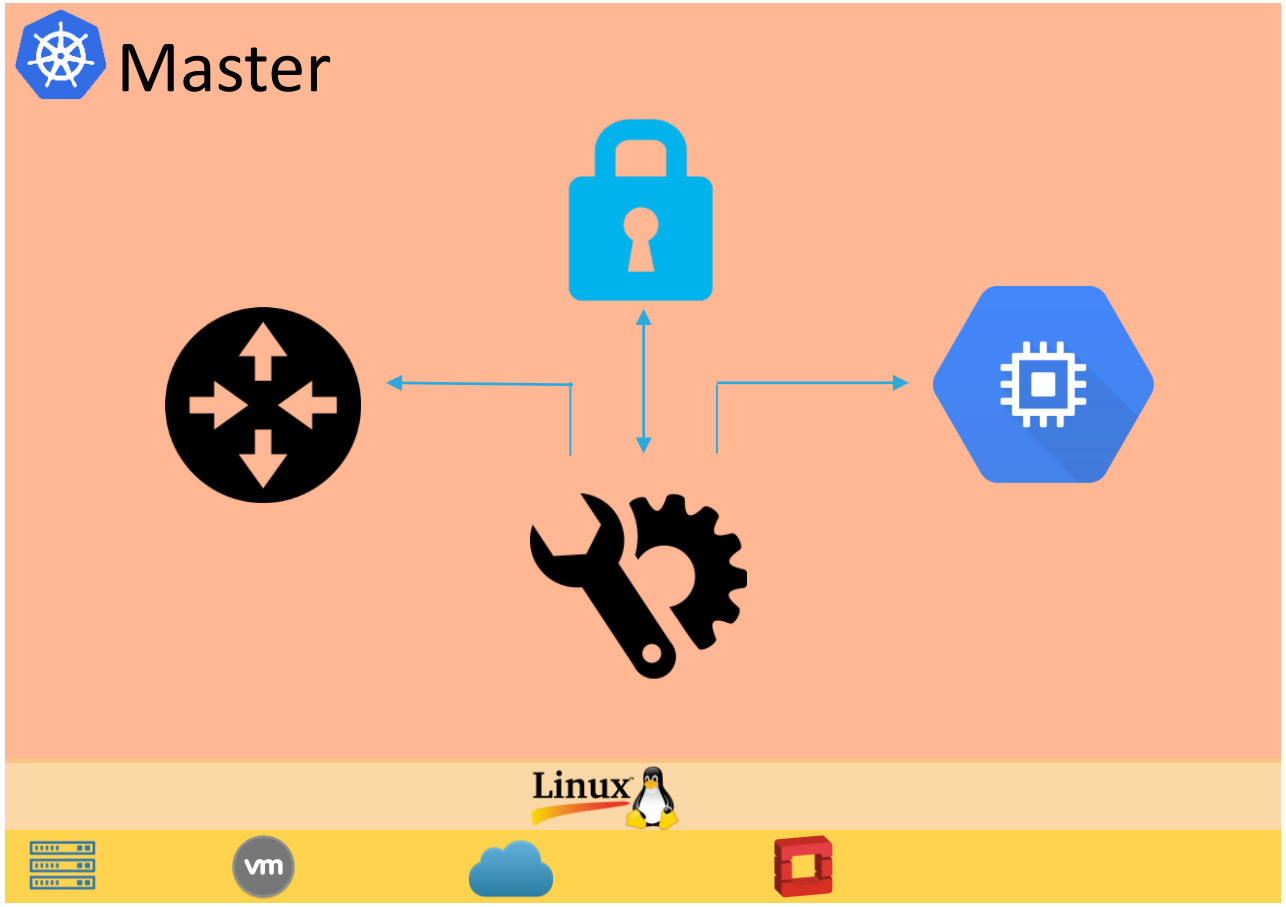

Workers
Control Plane
Master
Node
Node
Node
Node
Node
Node
Deployment

Masters

The Kubernetes Control Plane

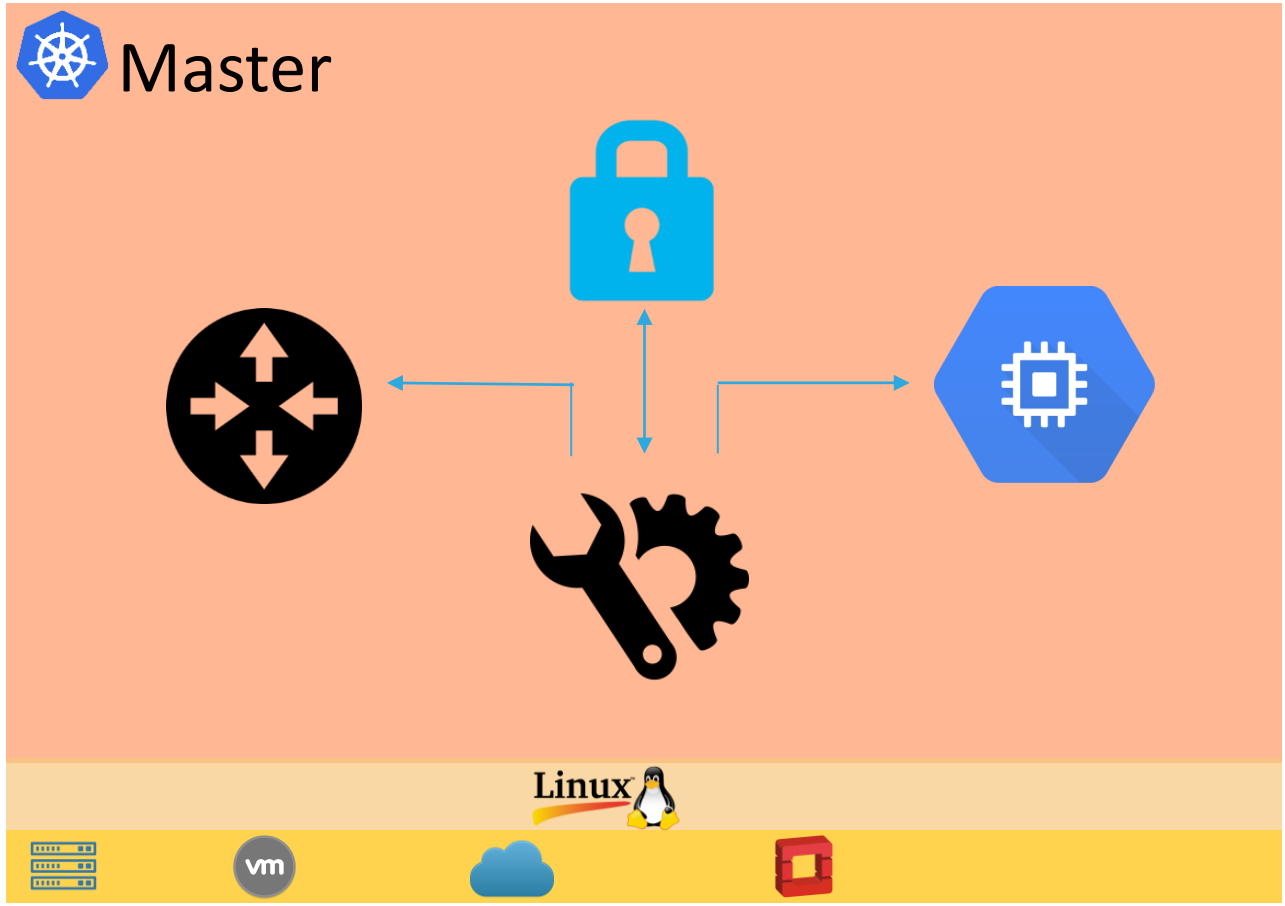

Master
Node
Linux
vm
Node
Linux
vm
Node
Linux
vm
Node
Linux
vm


Node
Linux
vm
Node
Linux
vm
Master
Linux
vm
Node
Linux
vm
Node
Linux
vm


Master
Linux
vm

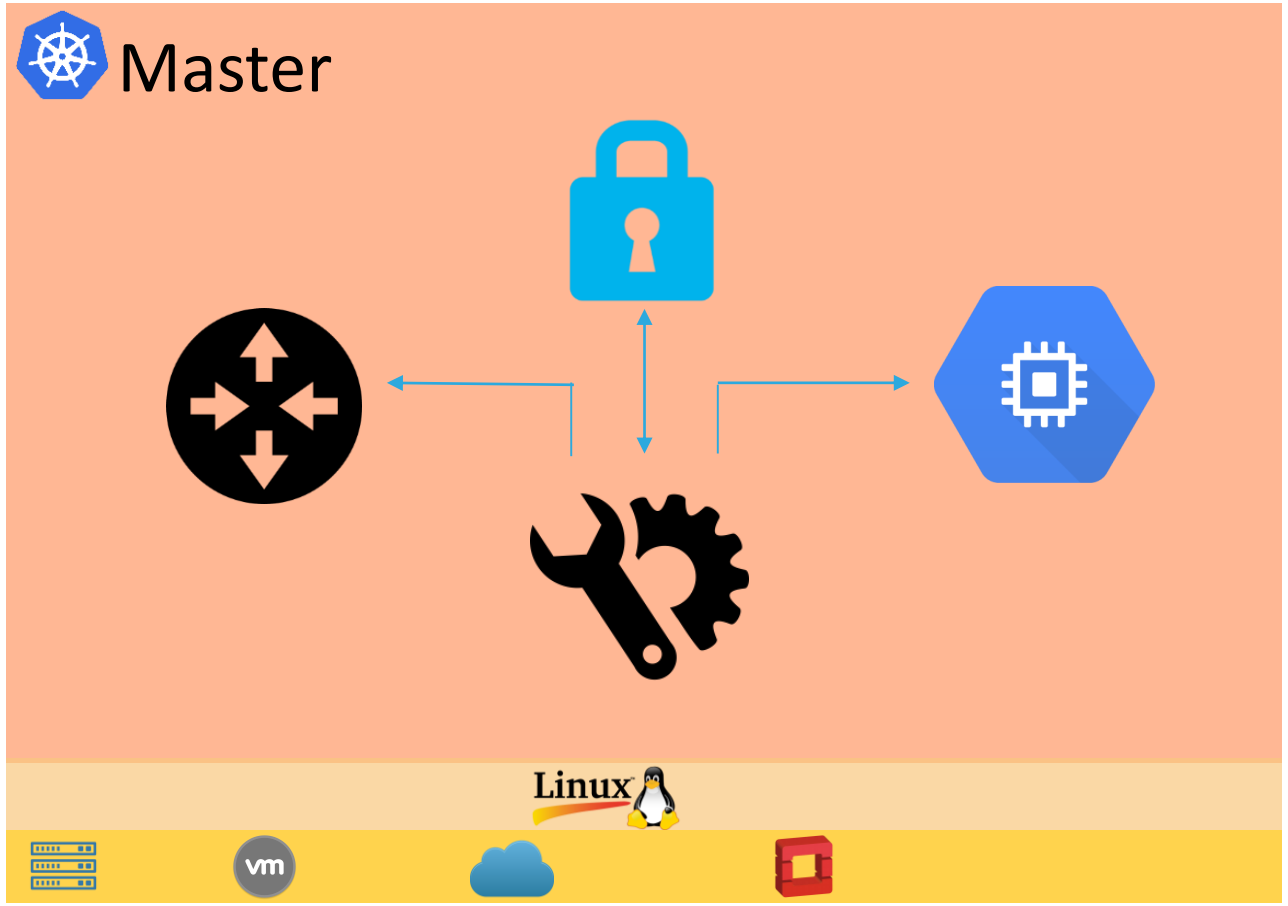
Kube-apiserver

- Control Plane Front End
- Exposes ReST API
- Processes JSON Manifest Files



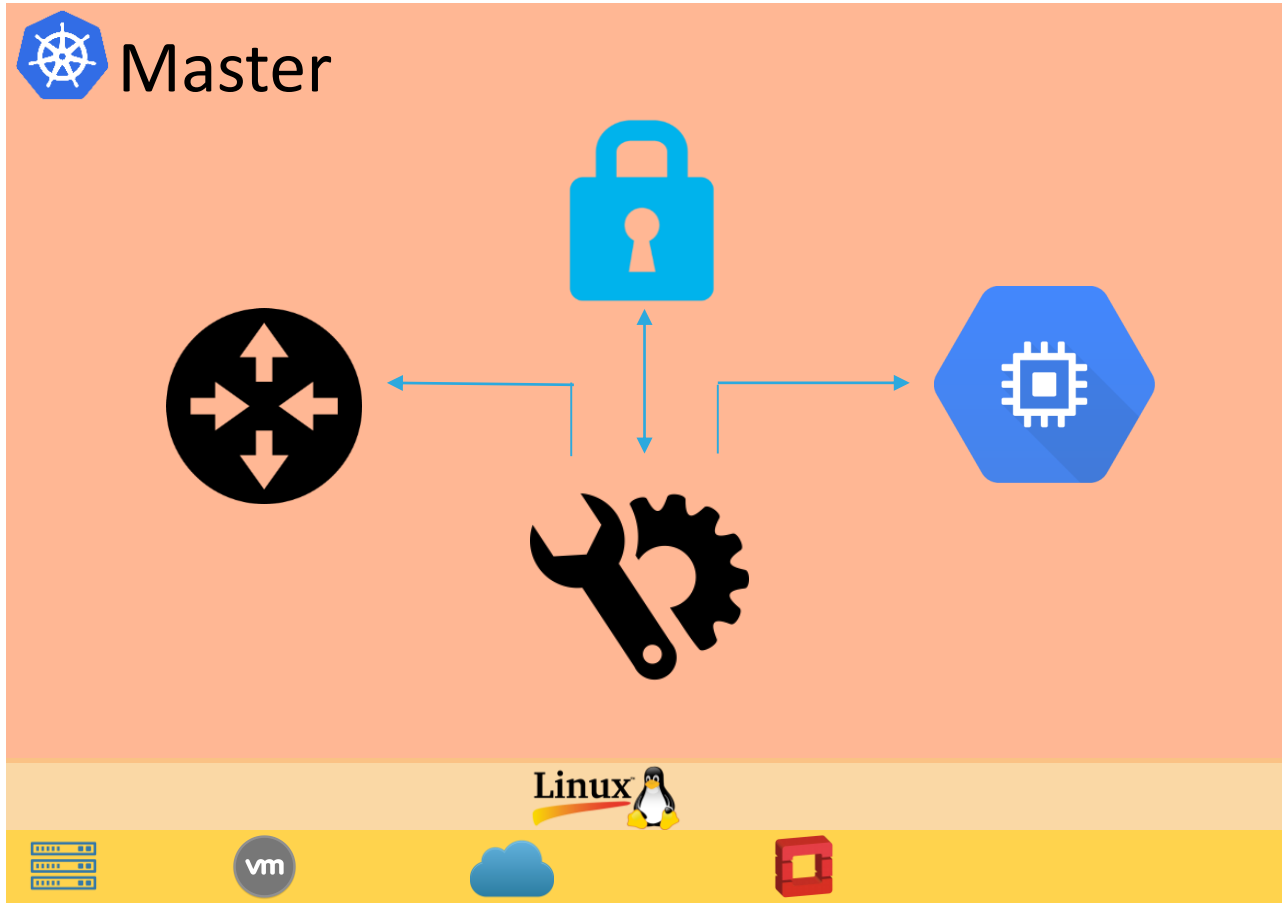
Cluster Store

- Persistent Storage
- Cluster state and config
- Uses **etcd**
  - No SQL Database
  - Key Value Store
- Distributed, Consistent, Watchable
- Source of ‘Truth’ – Back It Up!



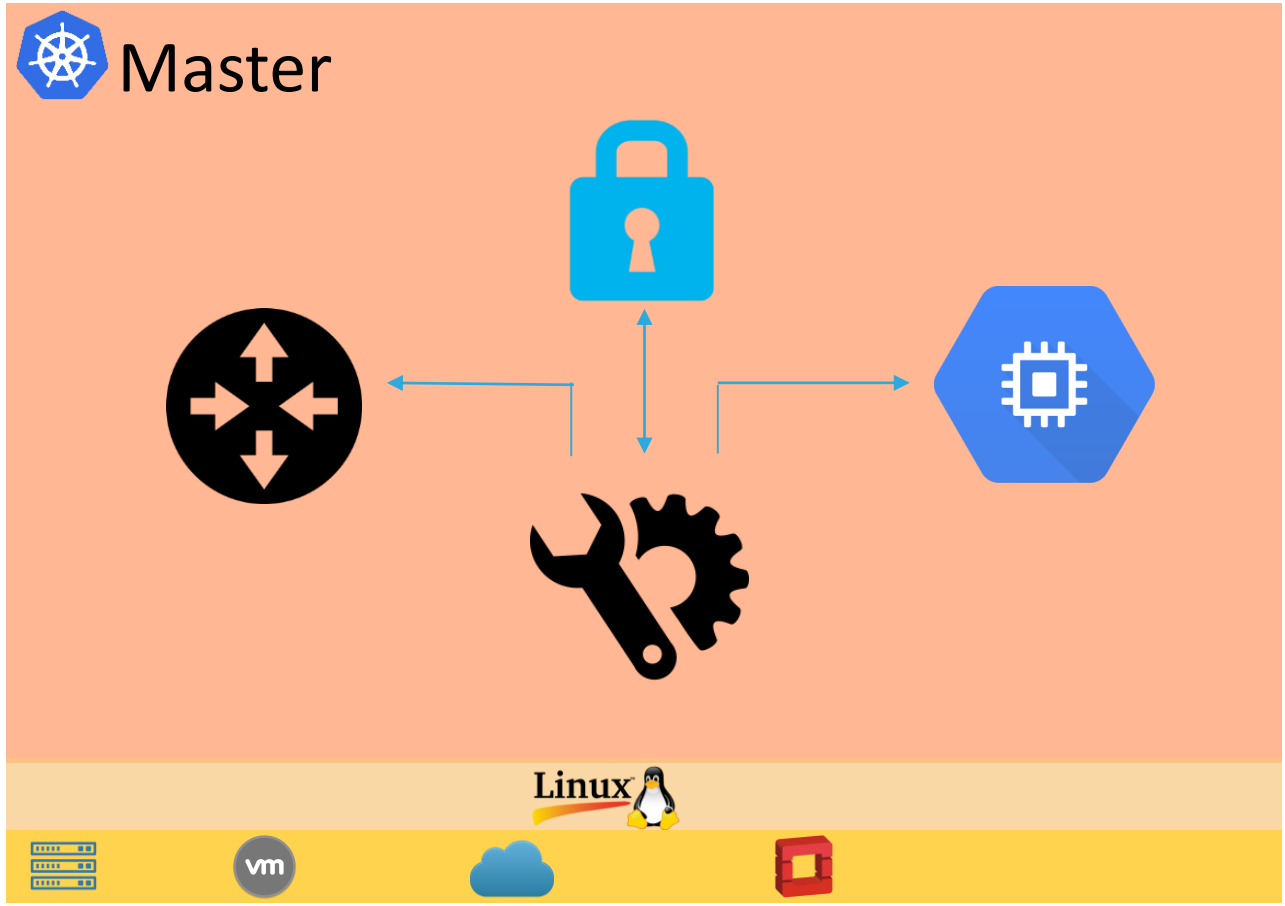
Kube-controller-manager

- Controller of controllers
  - Node Controller
  - Endpoint Controller
  - Namespace Controller
- Loops watching for changes
- Maintains desired state

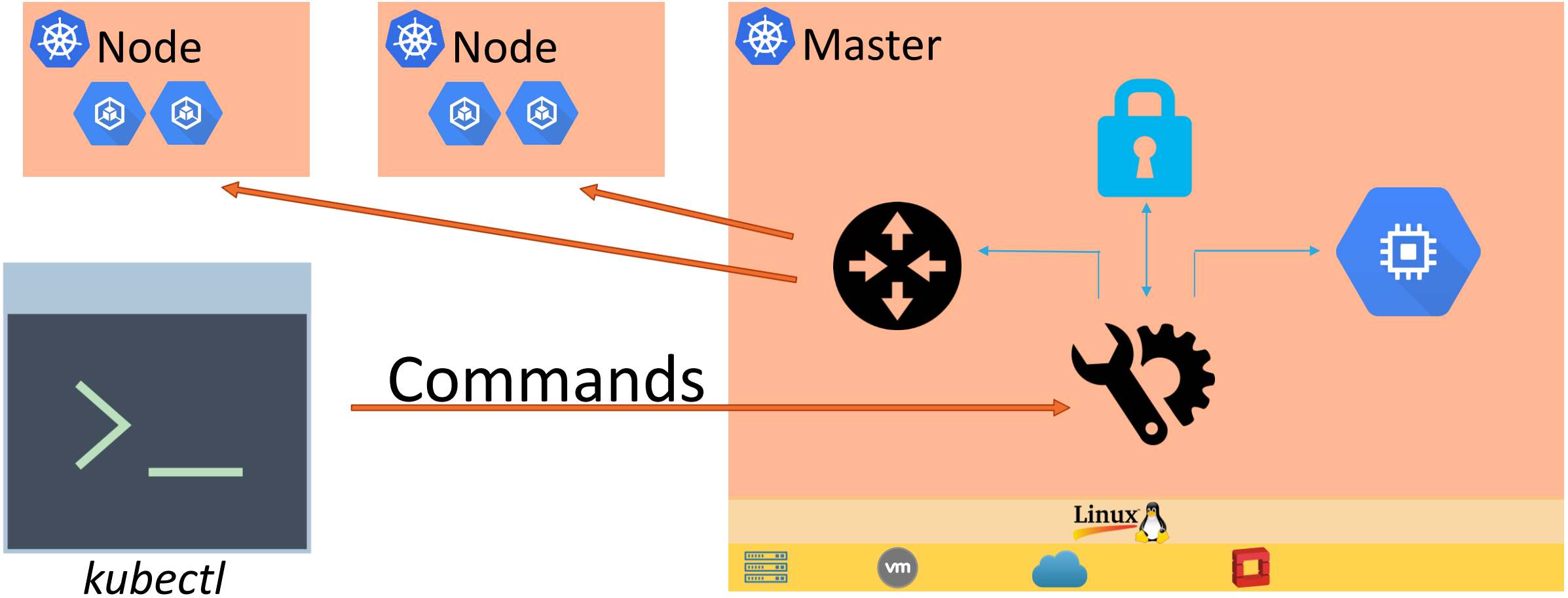


Kube-scheduler

- Watches **apiserver** for new Pods
- Assigns work to Nodes
  - affinity/anti-affinity
  - constraints
  - resources

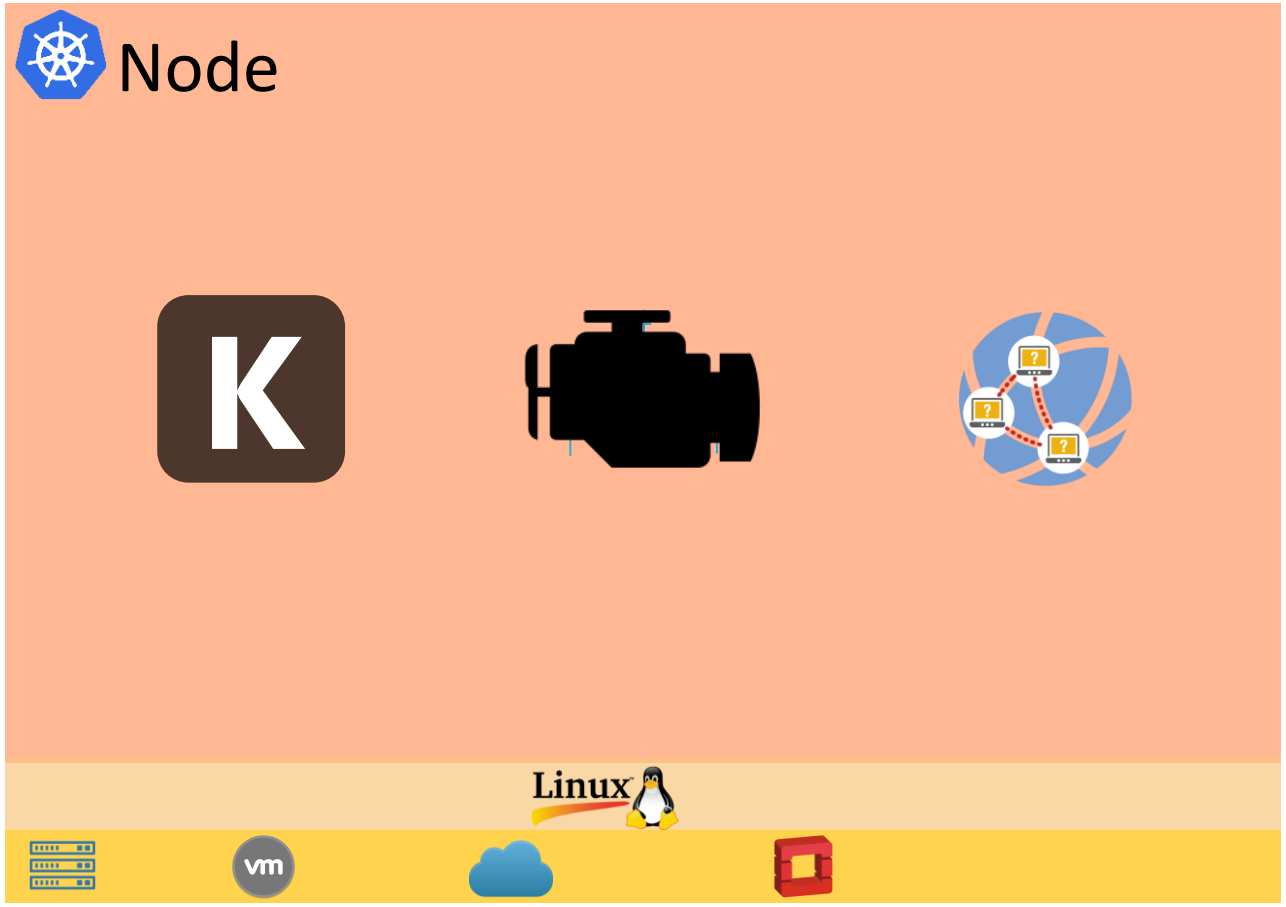


Master



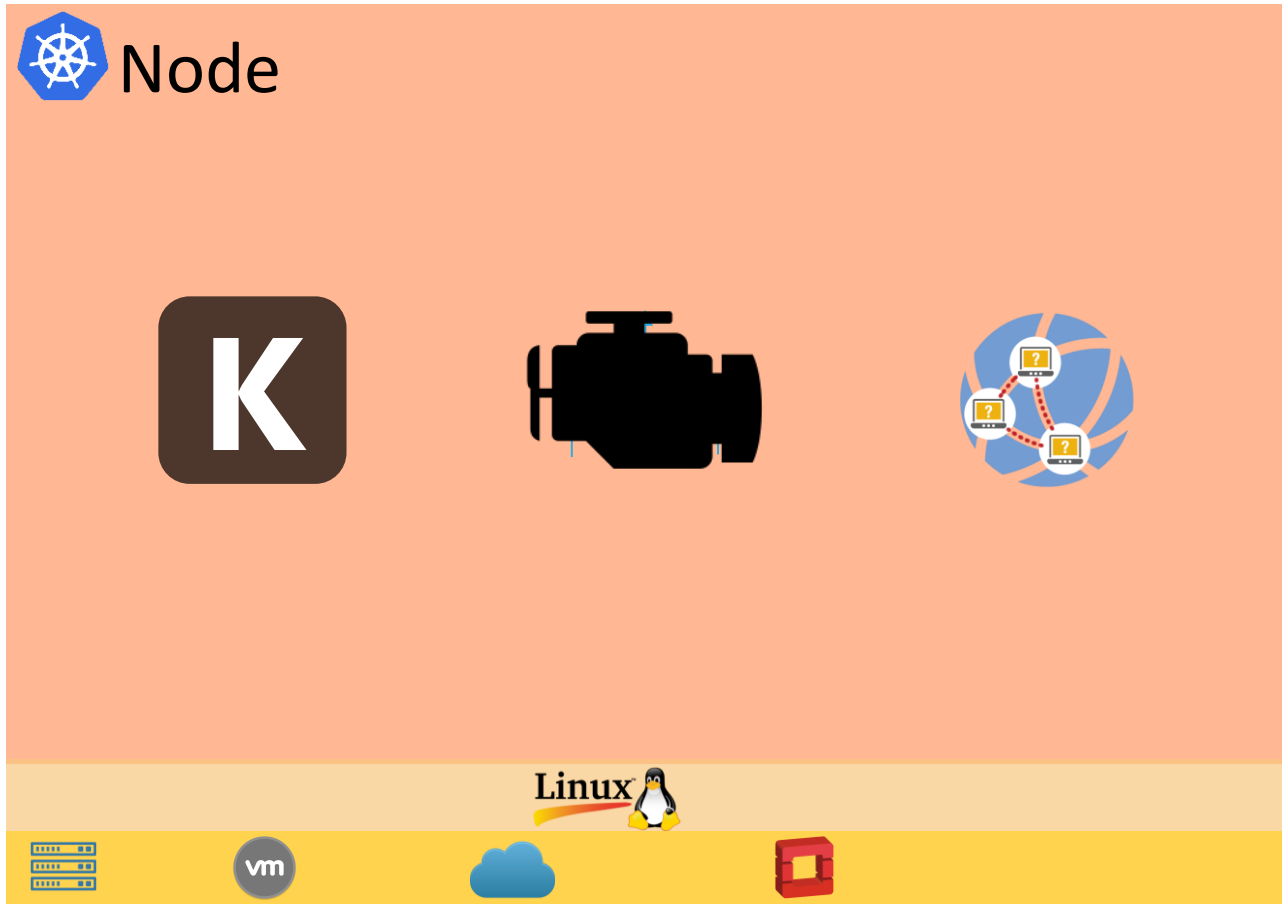
Nodes

The Kubernetes Workers

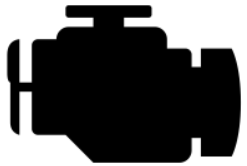

Node
K
Linux
vm

Kubelet

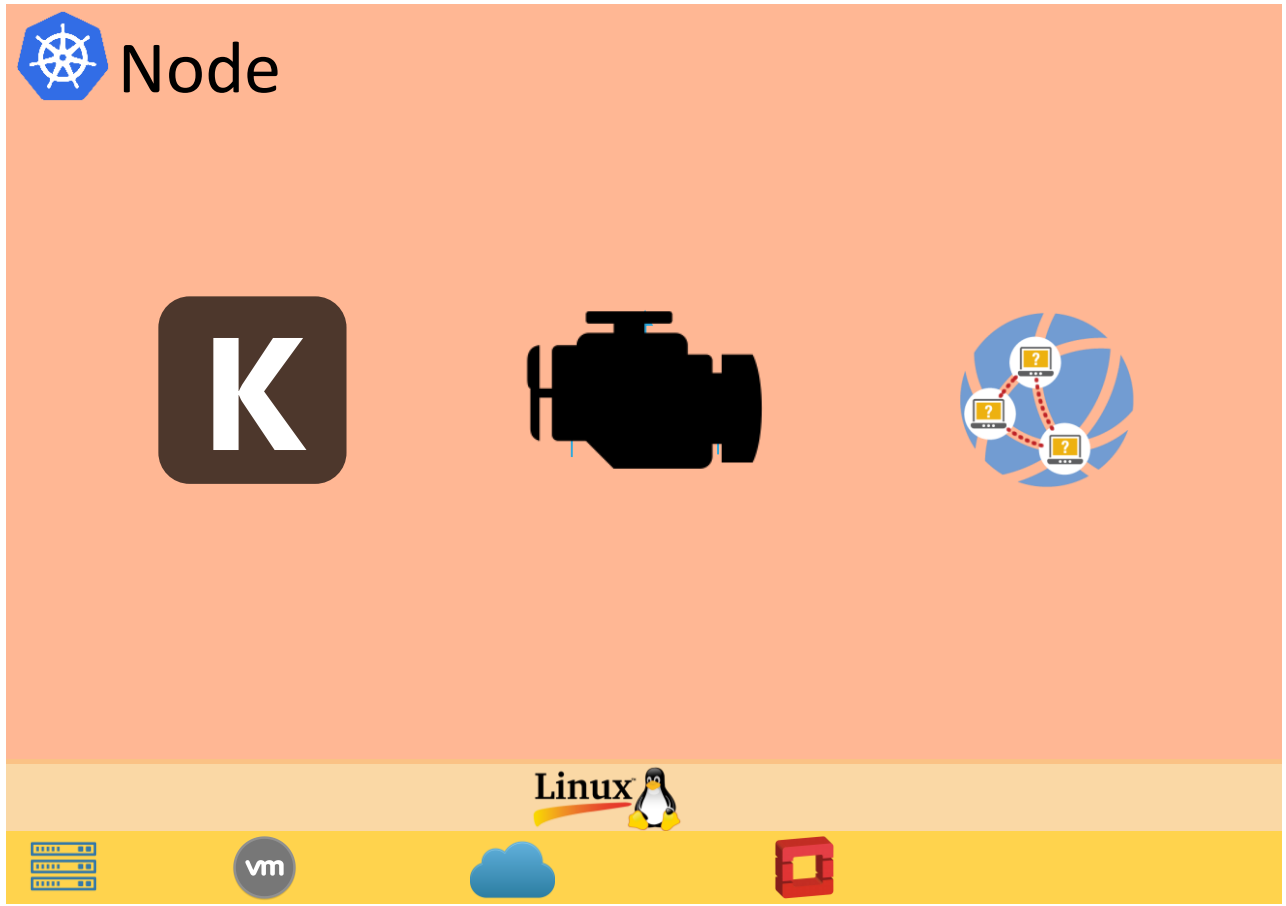
- Main Kubernetes Agent
- Registers node with cluster
- Watches **apiserver**
- Instantiates **Pods**
- Reports back to master
- Exposes endpoint TCP 10255
  - /spec
  - /healthz
  - /pods



Container Engine

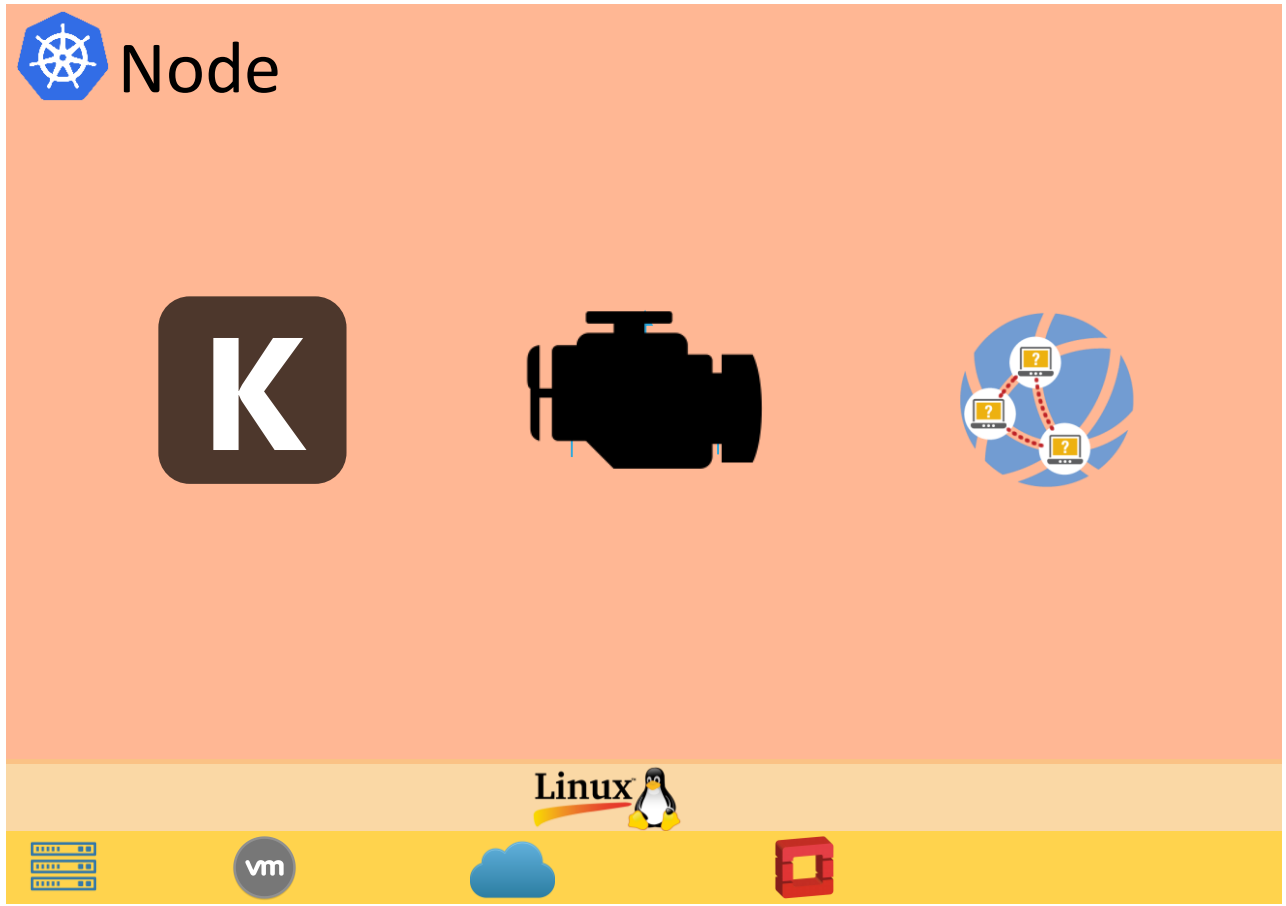


- Container Management
  - Retrieving Images
  - Starting & Stopping Containers
  - …
- Pluggable
  - **Docker**
  - **CoreOS rkt**



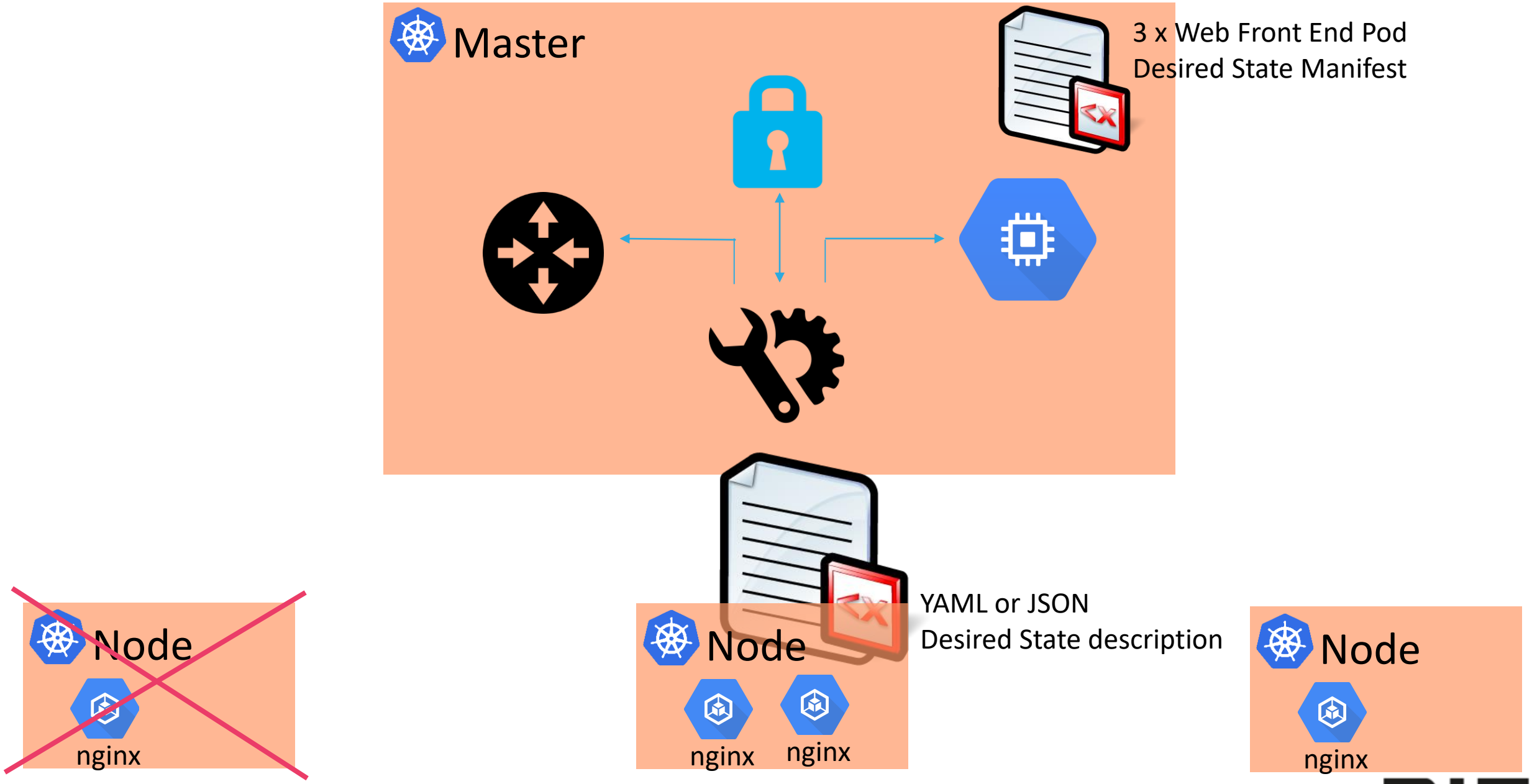
Kube-proxy

- Networking
  - Pod IP address
    - All containers in a pod share a single IP
  - Load balances across all pods in a service



Declarative Model

Desired State


Master
3 x Web Front End Pod
Desired State Manifest
YAML or JSON
Desired State description
Node
nginx
Node
nginx
nginx
Node
nginx

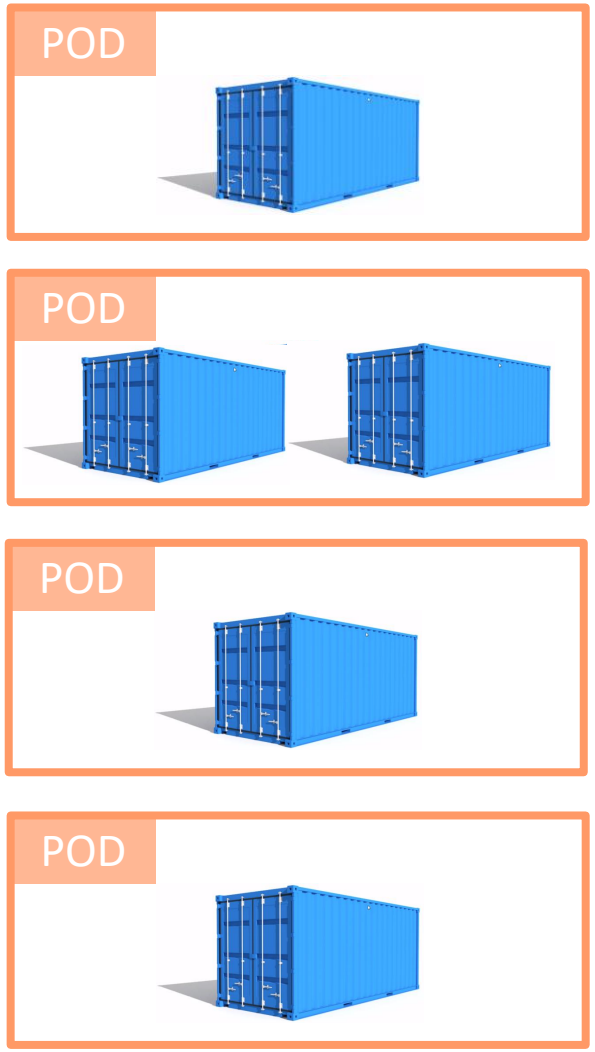
Pods

Atomic Units

Atomic Unit Of Scheduling

Containers

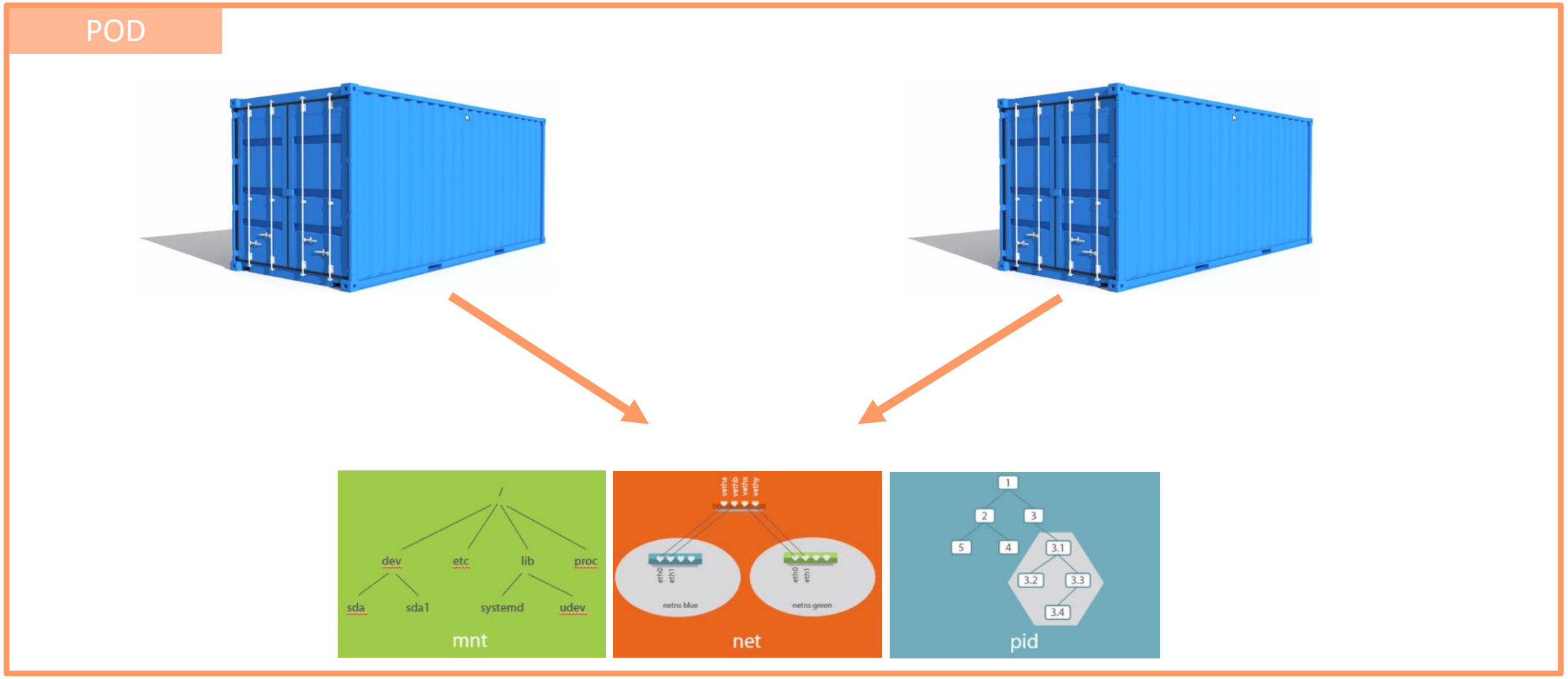
- Kubernetes runs container
- Only Inside Pods
- Pods may contain multiple containers



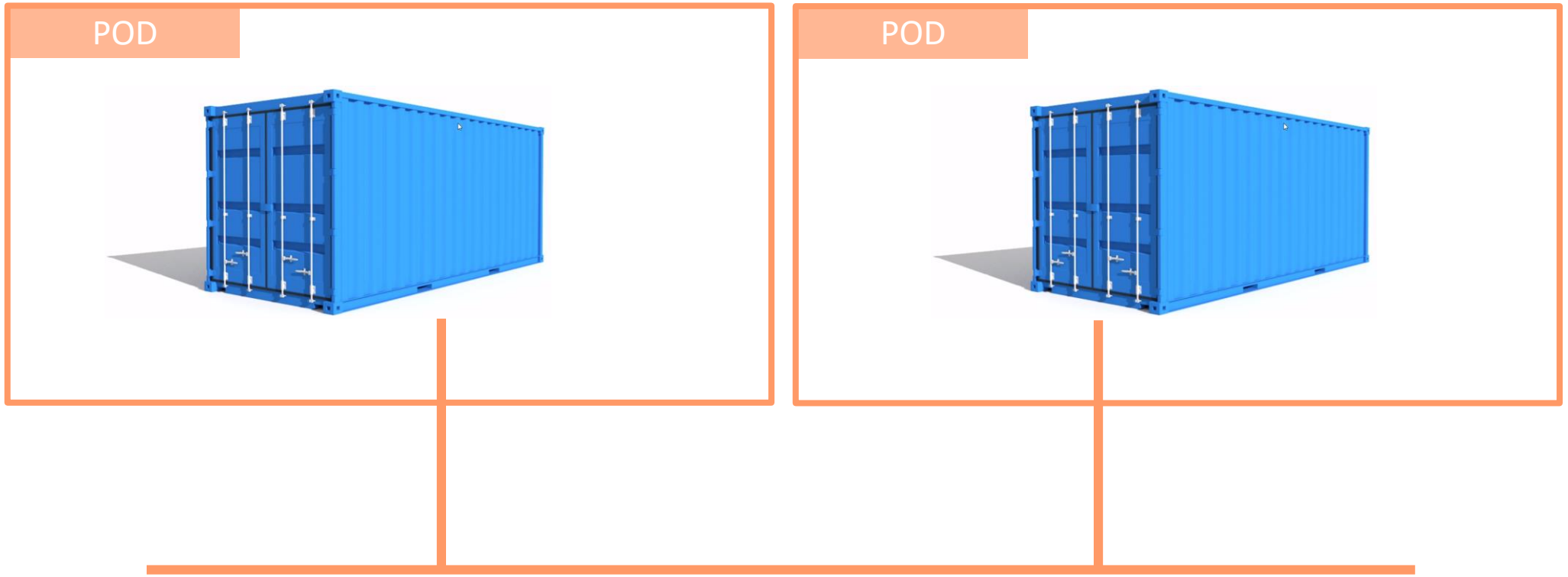
Pod

- Ring fenced environment
  - Network Stack
  - File System
  - Kernel Namespaces
  - …
- Hosts Container(s)
- All Containers SHARE the Pod environment

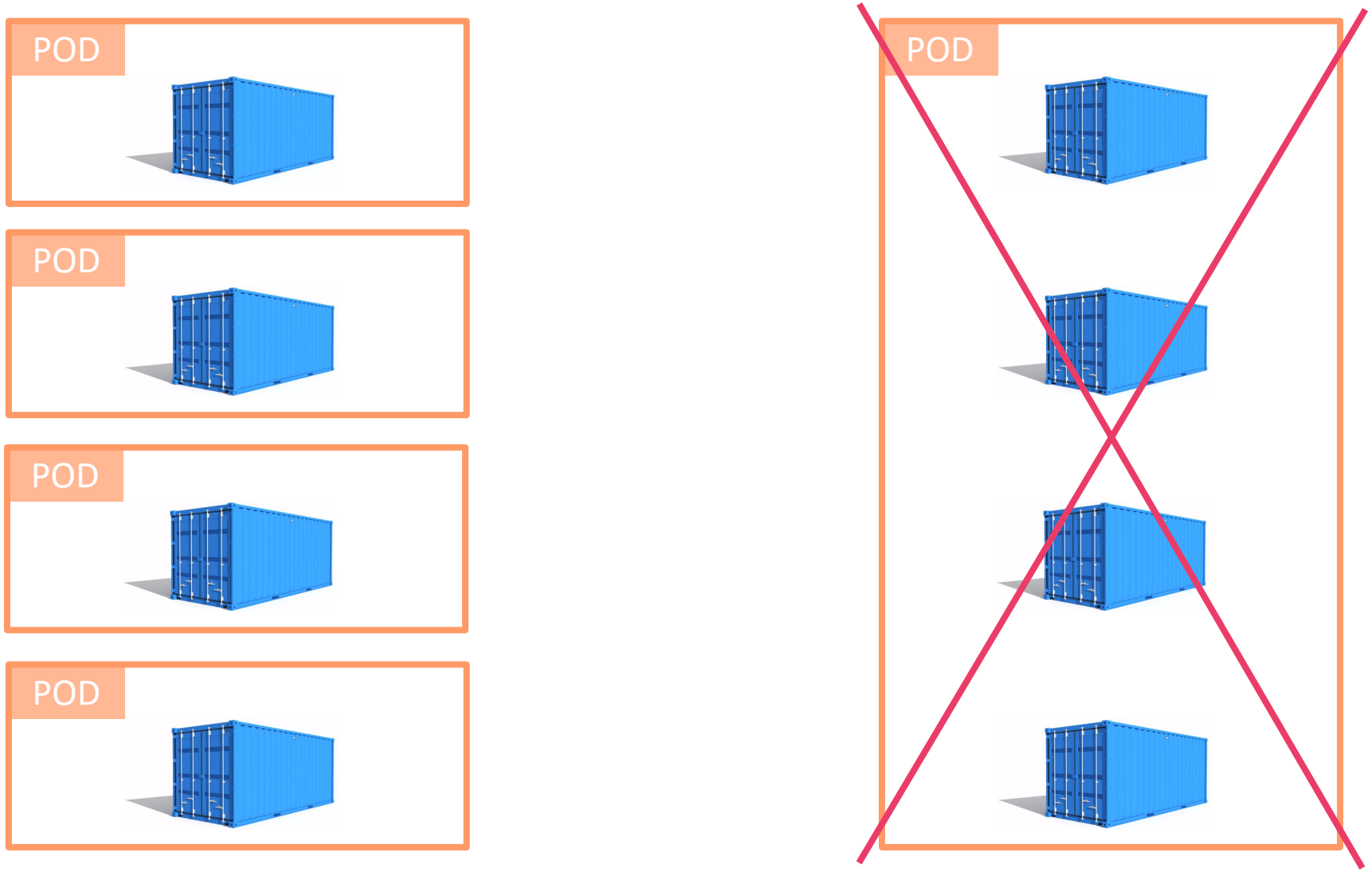
Tight Coupling



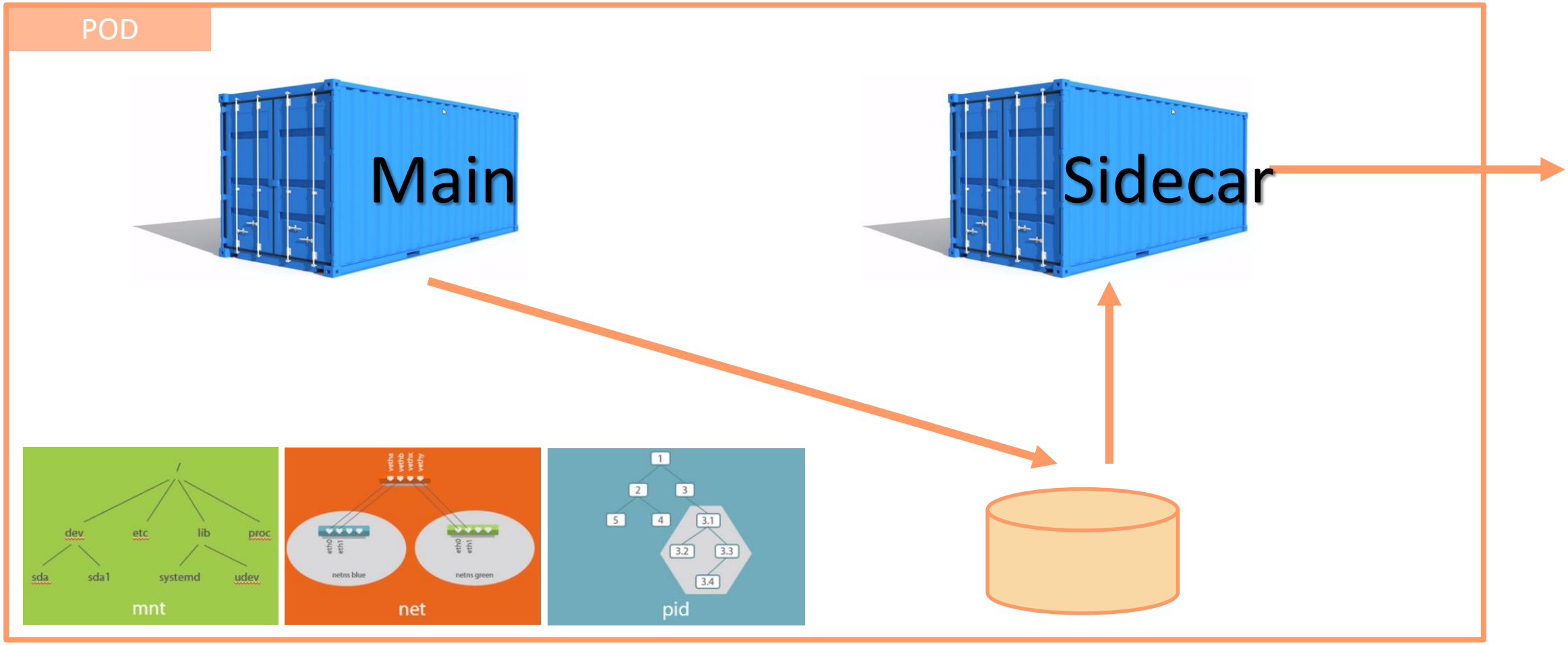
Loose Coupling



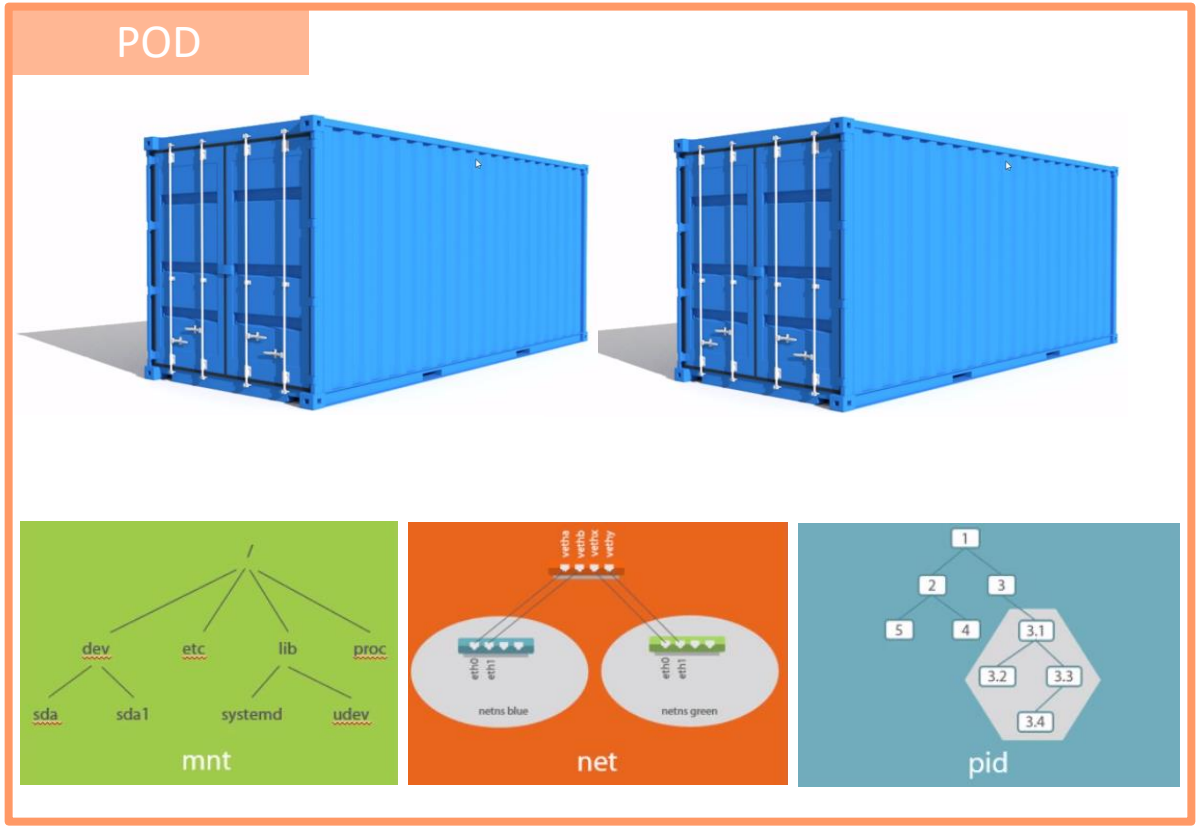
Scaling



Mutiple Containers



Atomic



Status: Ready

Status: Pending

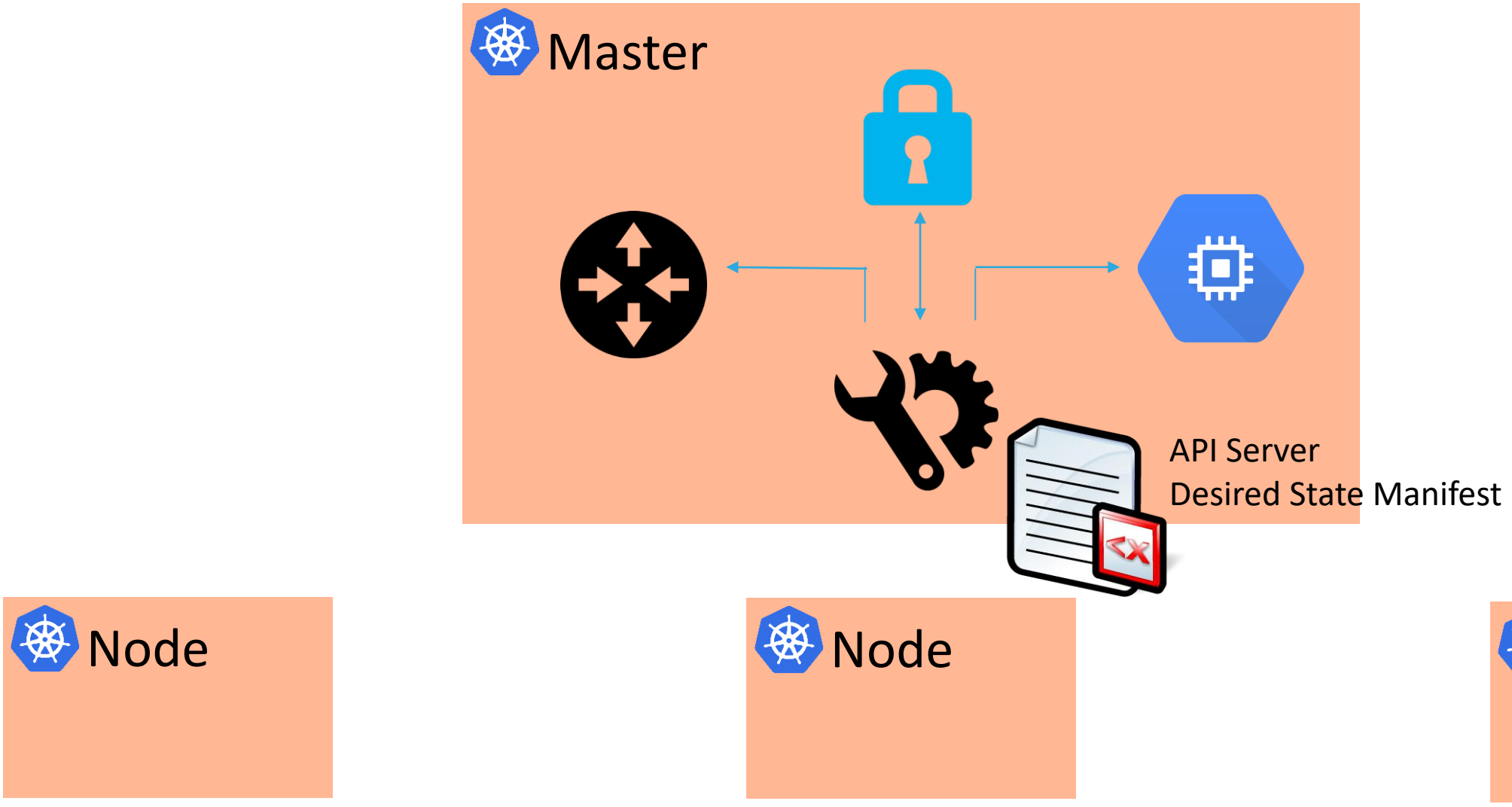
Pods are Mortal!

Status: Pending

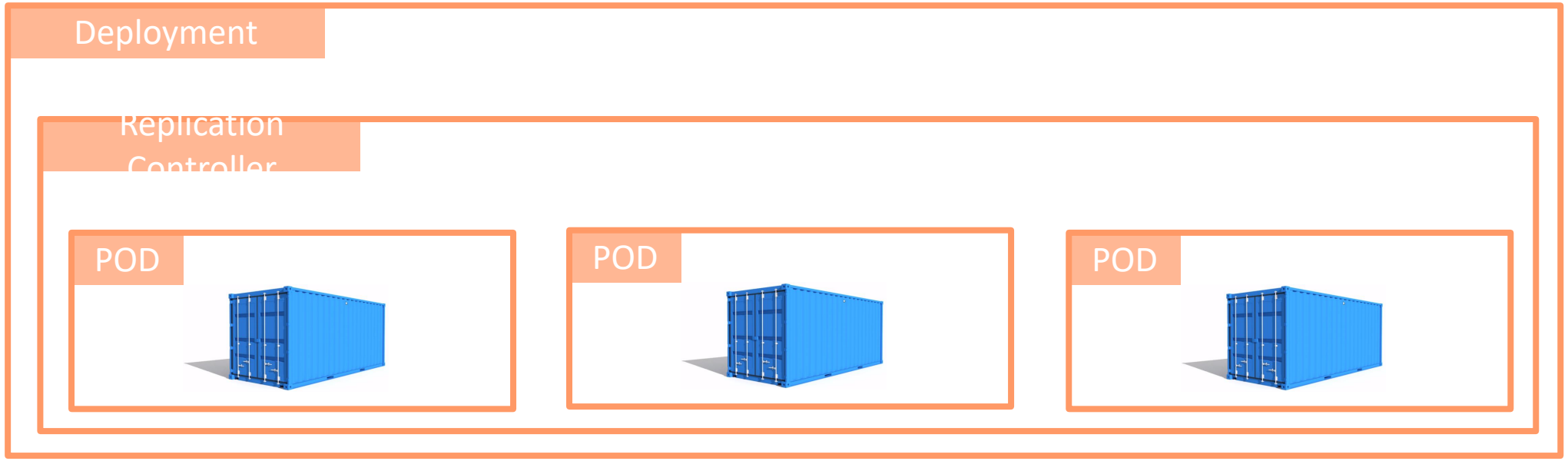
Status: Running

Status: Succeeded/Failed

Deployment


Master
API Server
Desired State Manifest
Node
Node
Node
nginx

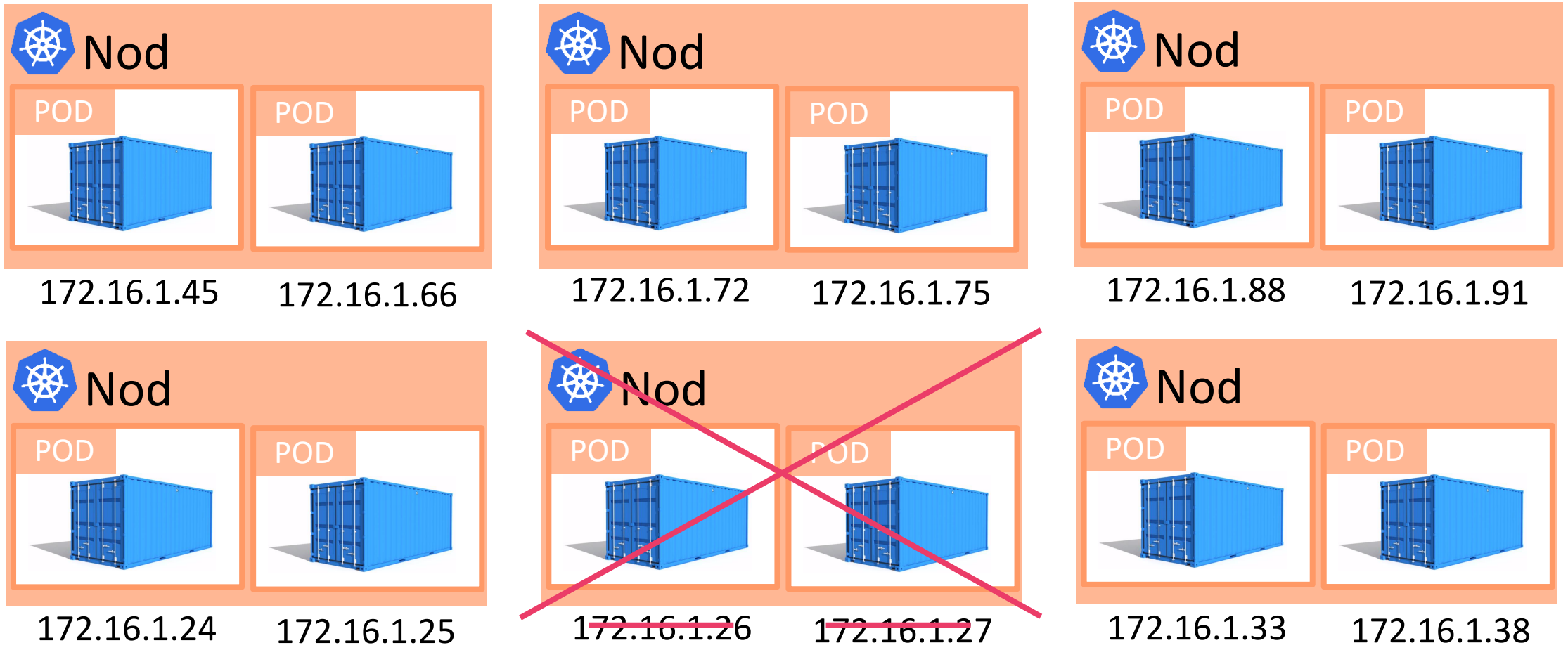
Deployment Abstraction



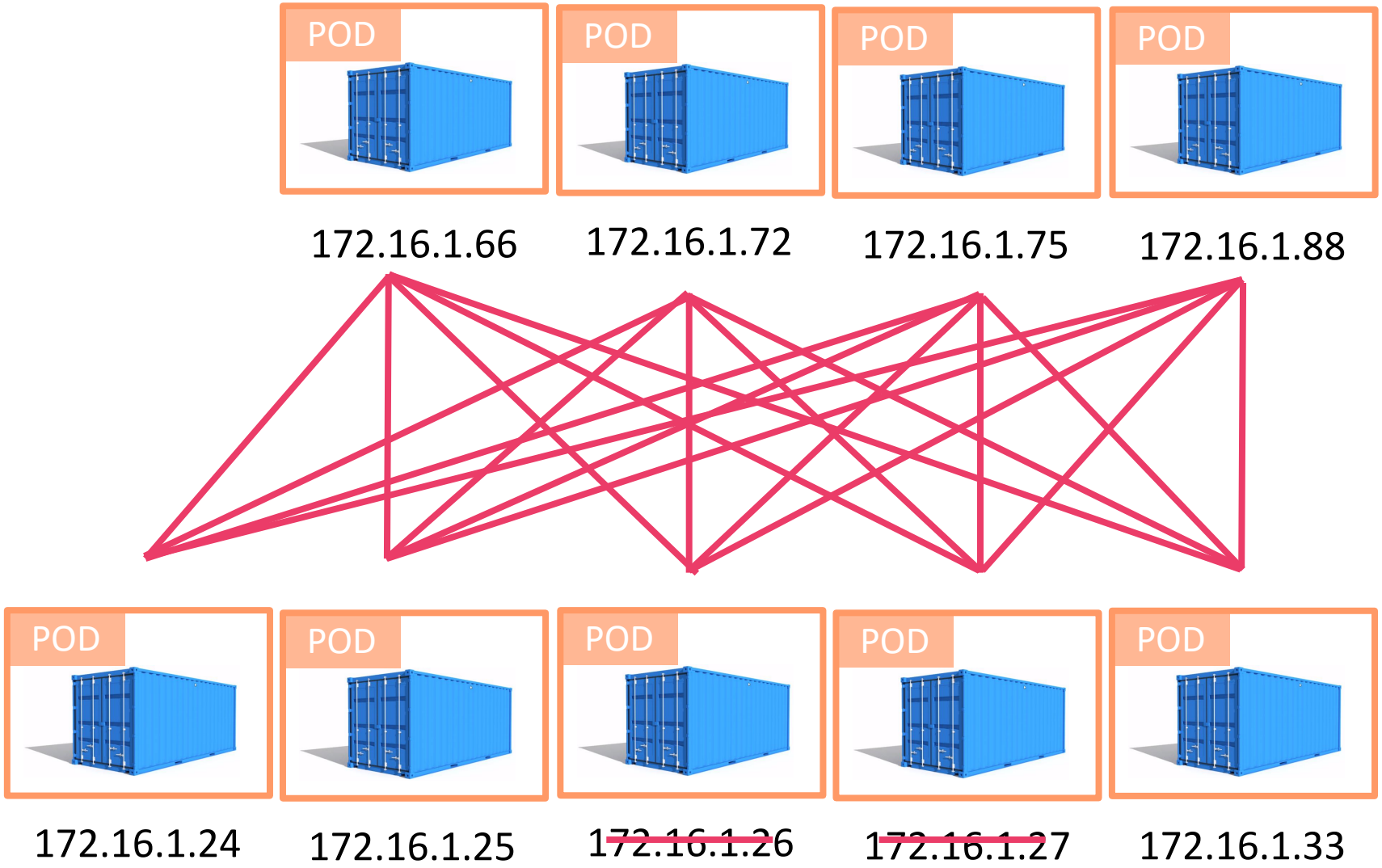
Services

Atomic Units

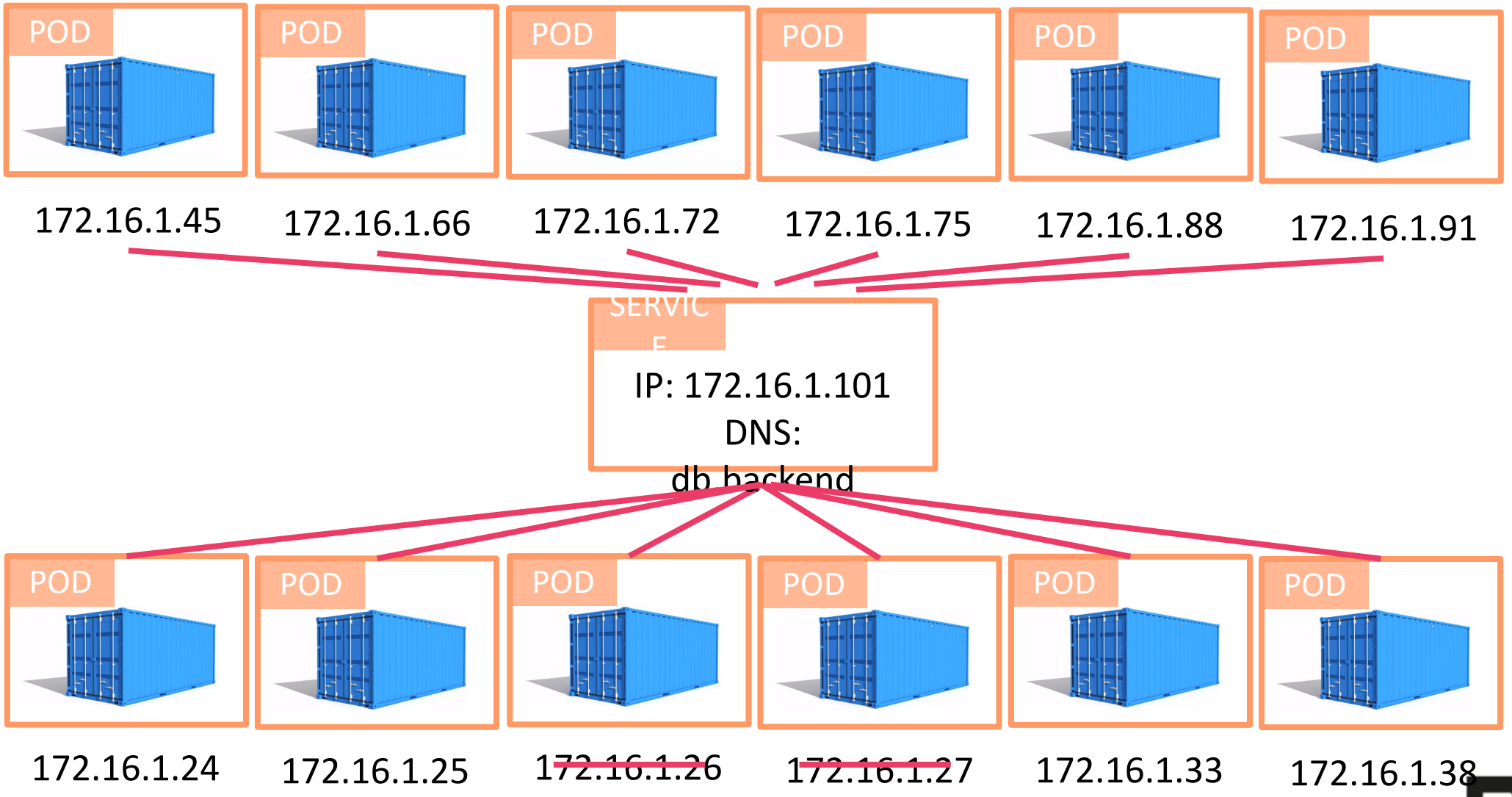
IP Churn


Nod
POD
POD
172.16.1.45
172.16.1.66
Nod
POD
POD
172.16.1.72
172.16.1.75
Nod
POD
POD
172.16.1.88
172.16.1.91
Nod
POD
POD
172.16.1.24
172.16.1.25
Nod
POD
POD
172.16.1.26
172.16.1.27
Nod
POD
POD
172.16.1.33
172.16.1.38

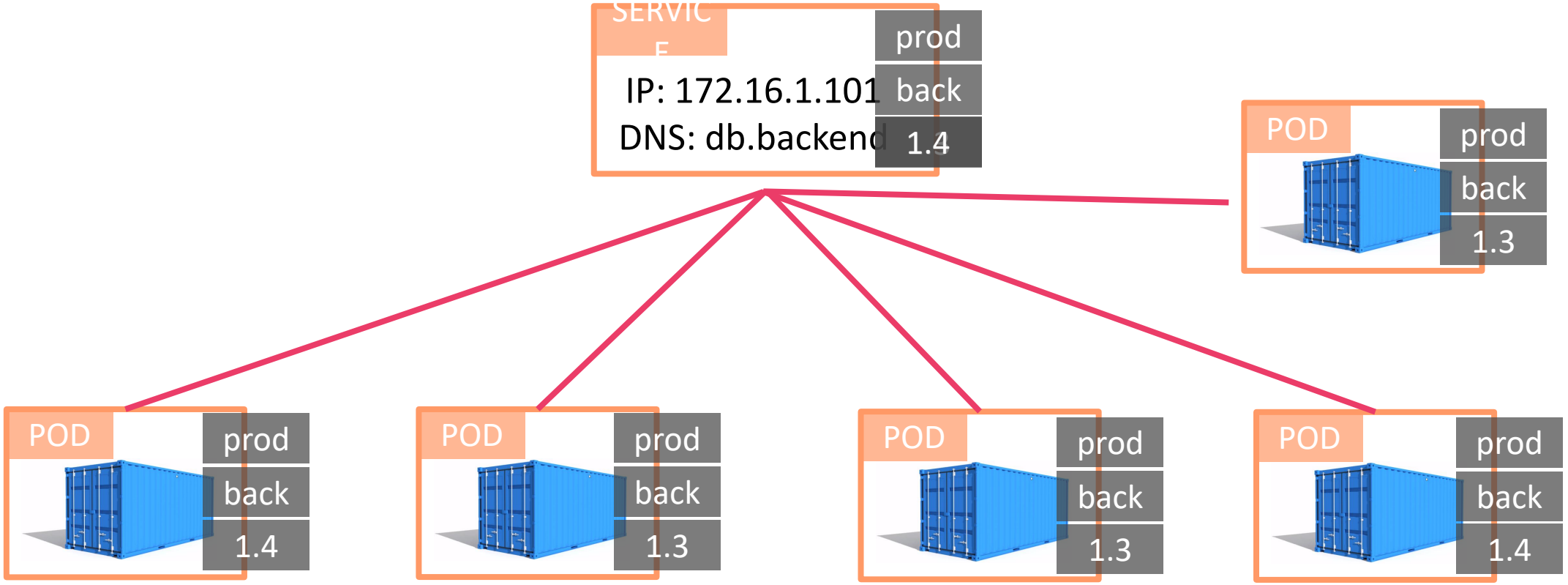
IP Churn


POD
POD
POD
POD
172.16.1.66
172.16.1.72
172.16.1.75
172.16.1.88
POD
POD
POD
POD
POD
172.16.1.24
172.16.1.25
172.16.1.26
172.16.1.27
172.16.1.33

Services

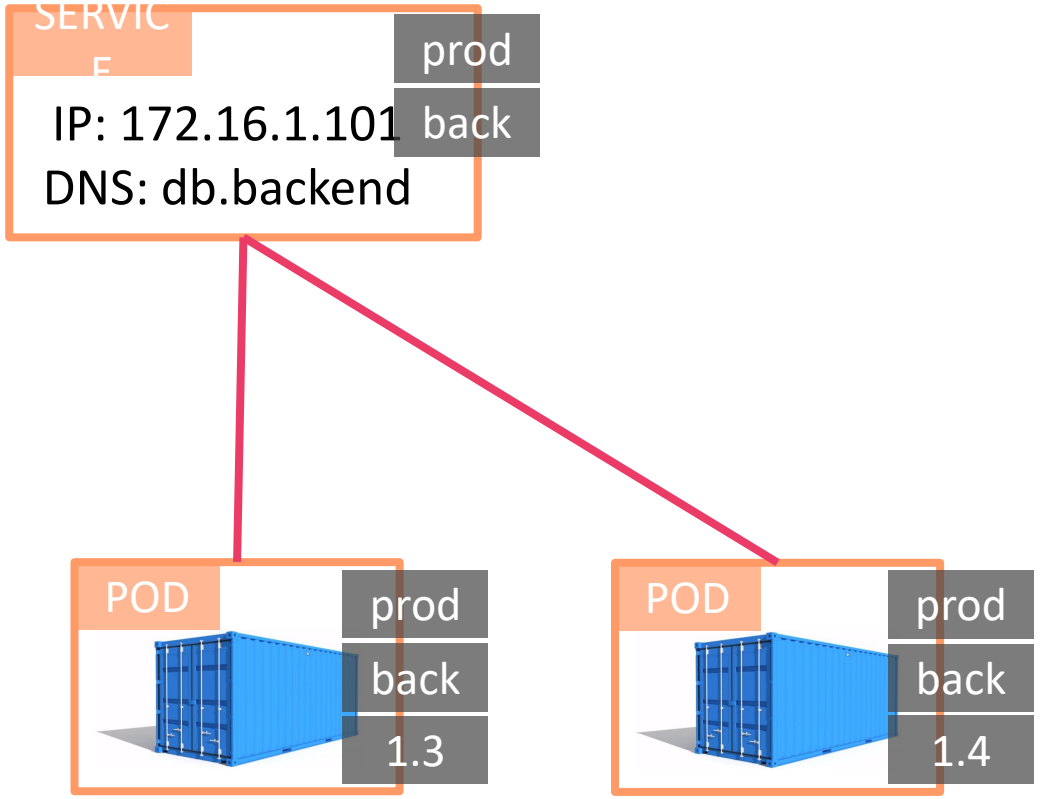

POD
POD
POD
POD
POD
POD
172.16.1.45
172.16.1.66
172.16.1.72
172.16.1.75
172.16.1.88
172.16.1.91
SERVIC
E
IP: 172.16.1.101
DNS:
db.backend
POD
POD
POD
POD
POD
POD
172.16.1.24
172.16.1.25
172.16.1.26
172.16.1.27
172.16.1.33
172.16.1.38

Labels


SERVIC
E
IP: 172.16.1.101
DNS: db.backend
prod
back
1.4
POD
prod
back
1.3
POD
prod
back
1.4
POD
prod
back
1.3
POD
prod
back
1.3
POD
prod
back
1.4

Services

- Only communicates to Healthy Pods
- Can be configured for Session Affinity
- Can point to resources outside the service
- Random Load Balancing
- Uses TCP by Default



Resources

Slides and demos from the conference will be available at
github.com/nordicinfrastructureconference/2018 (bit.ly/2y7JhA3)