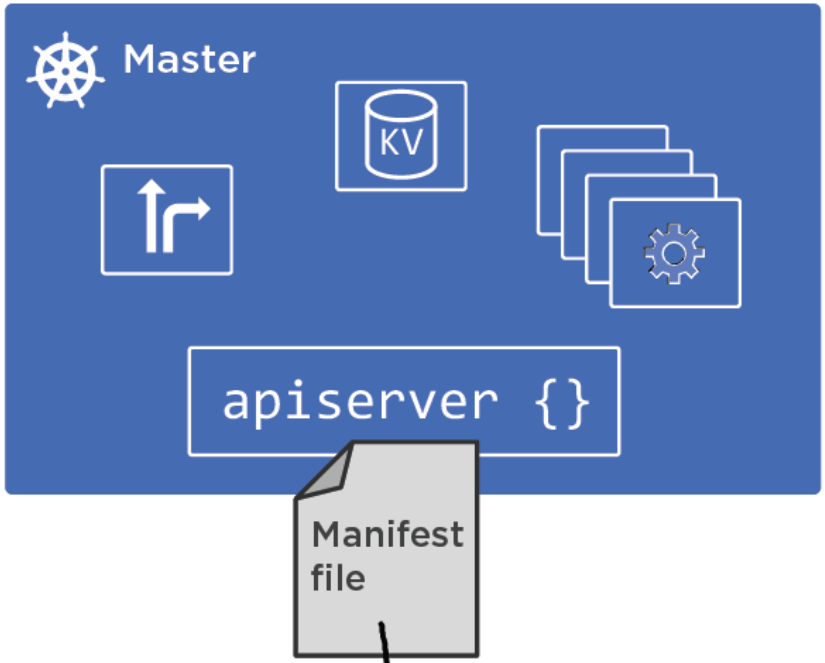
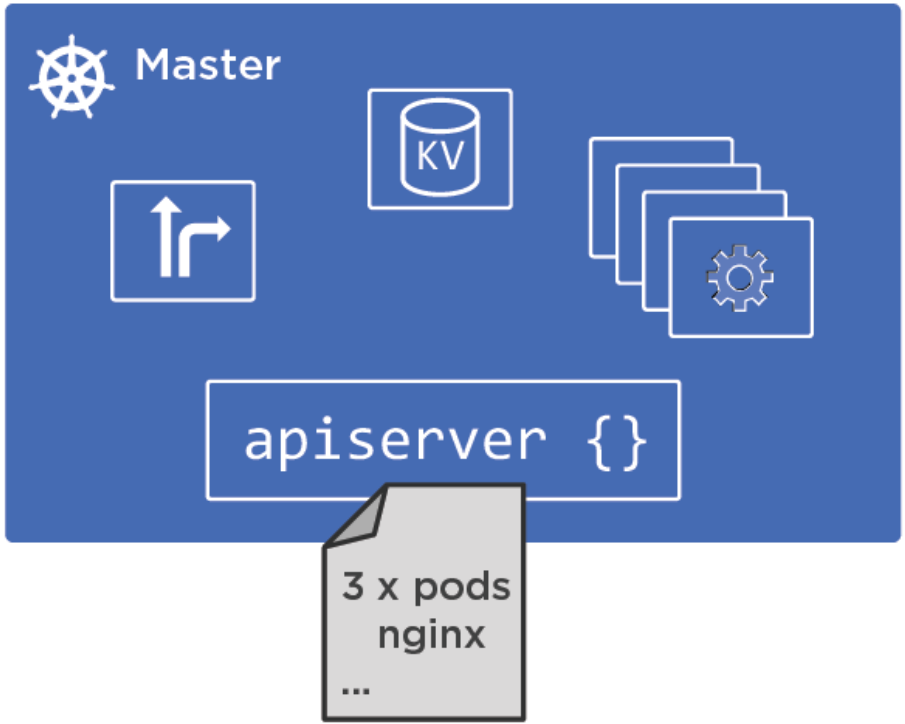
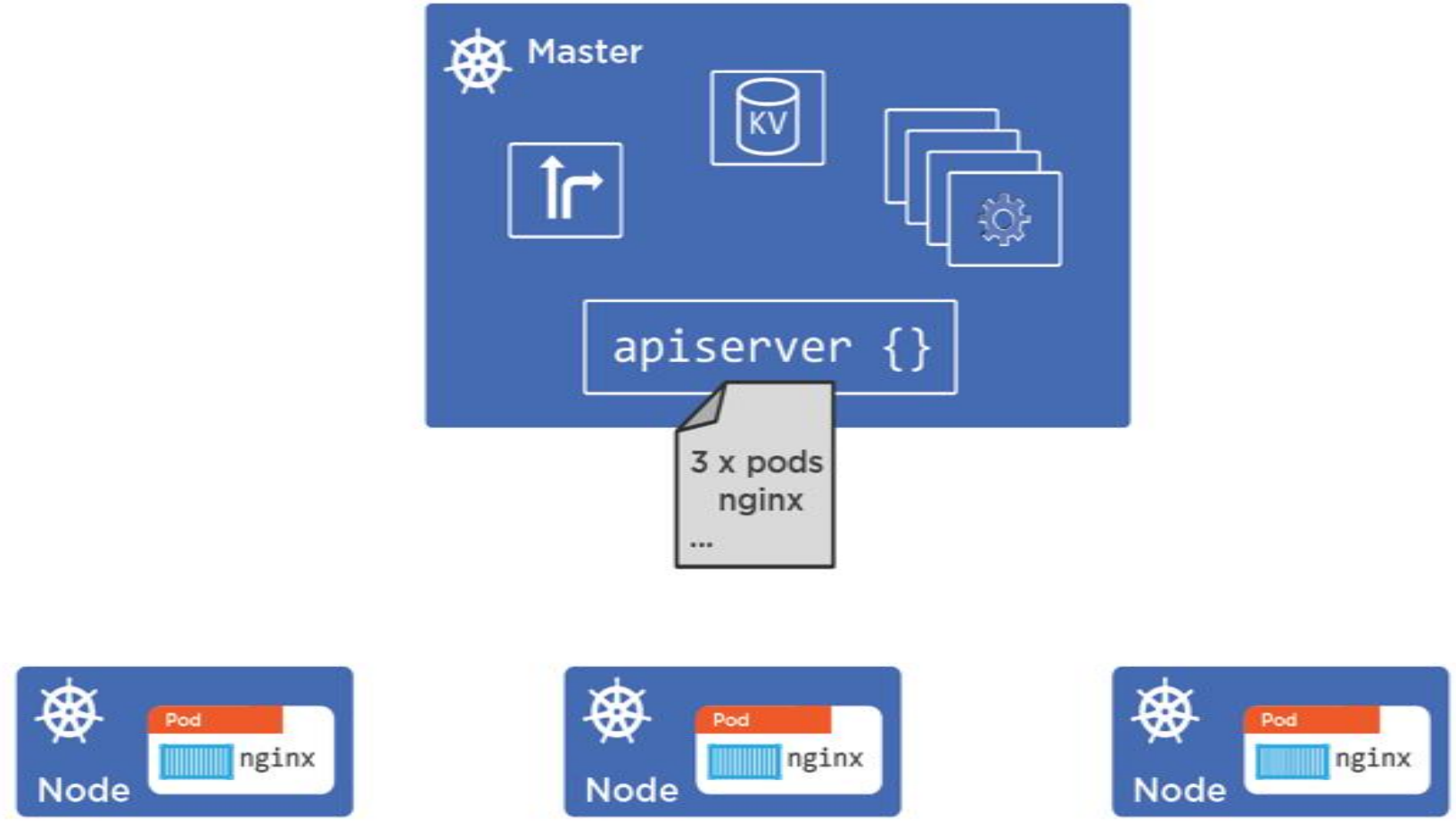
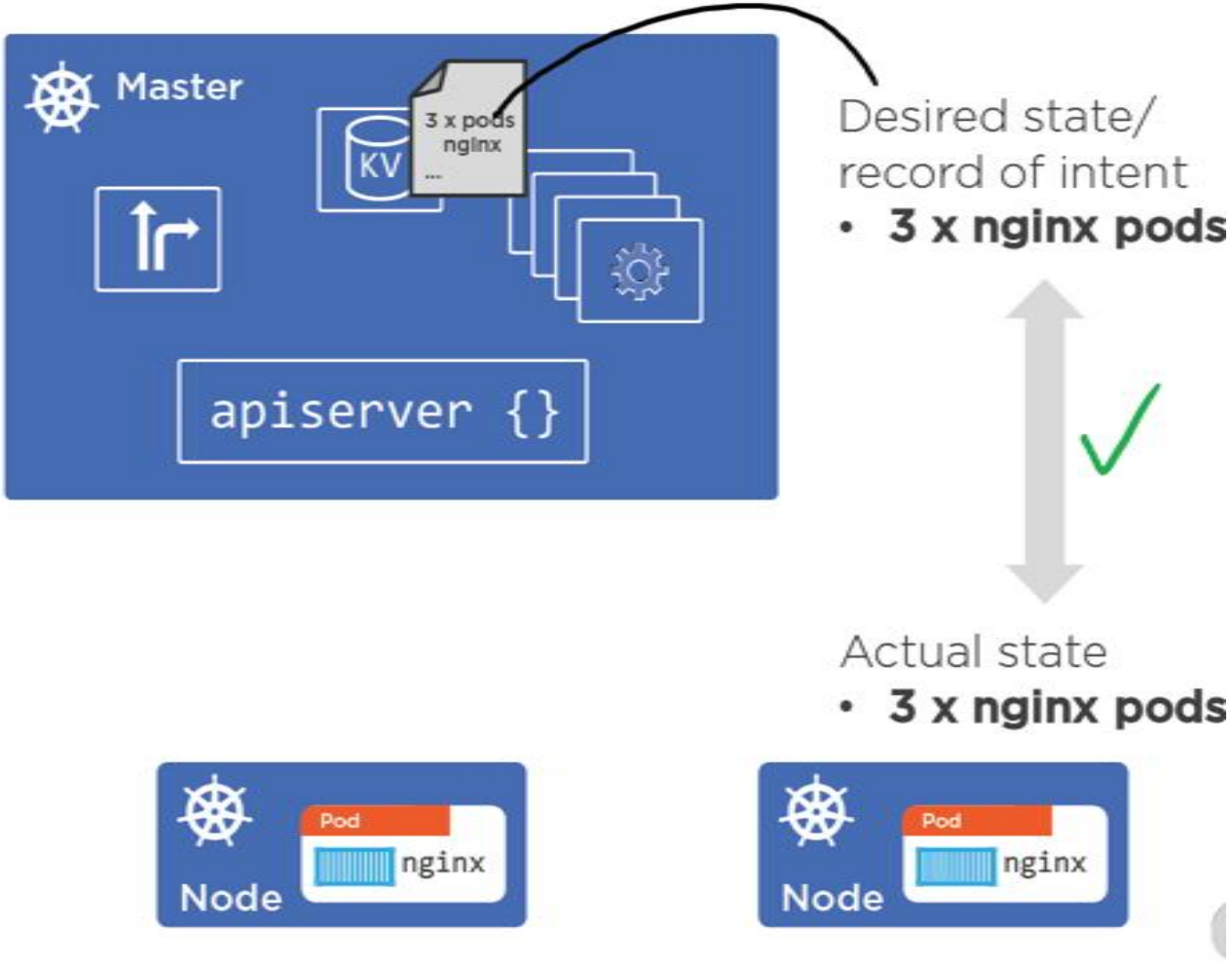


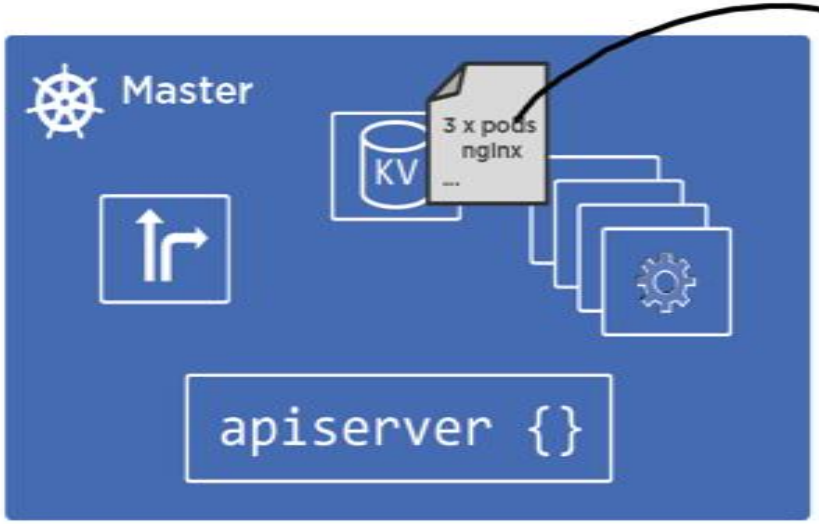
YAML or JSON

Describe **desired** state




Master
KV
apiserver {}
3 x pods
nginx
...
Node
Pod
nginx
Node
Pod
nginx
Node
Pod
nginx


Master
KV
3 x pods
nginx
...
apiserver {}
Desired state/
record of intent
• 3 x nginx pods
Actual state
• 3 x nginx pods
Node
Pod
nginx
Node
Pod
nginx
Node
Pod
nginx

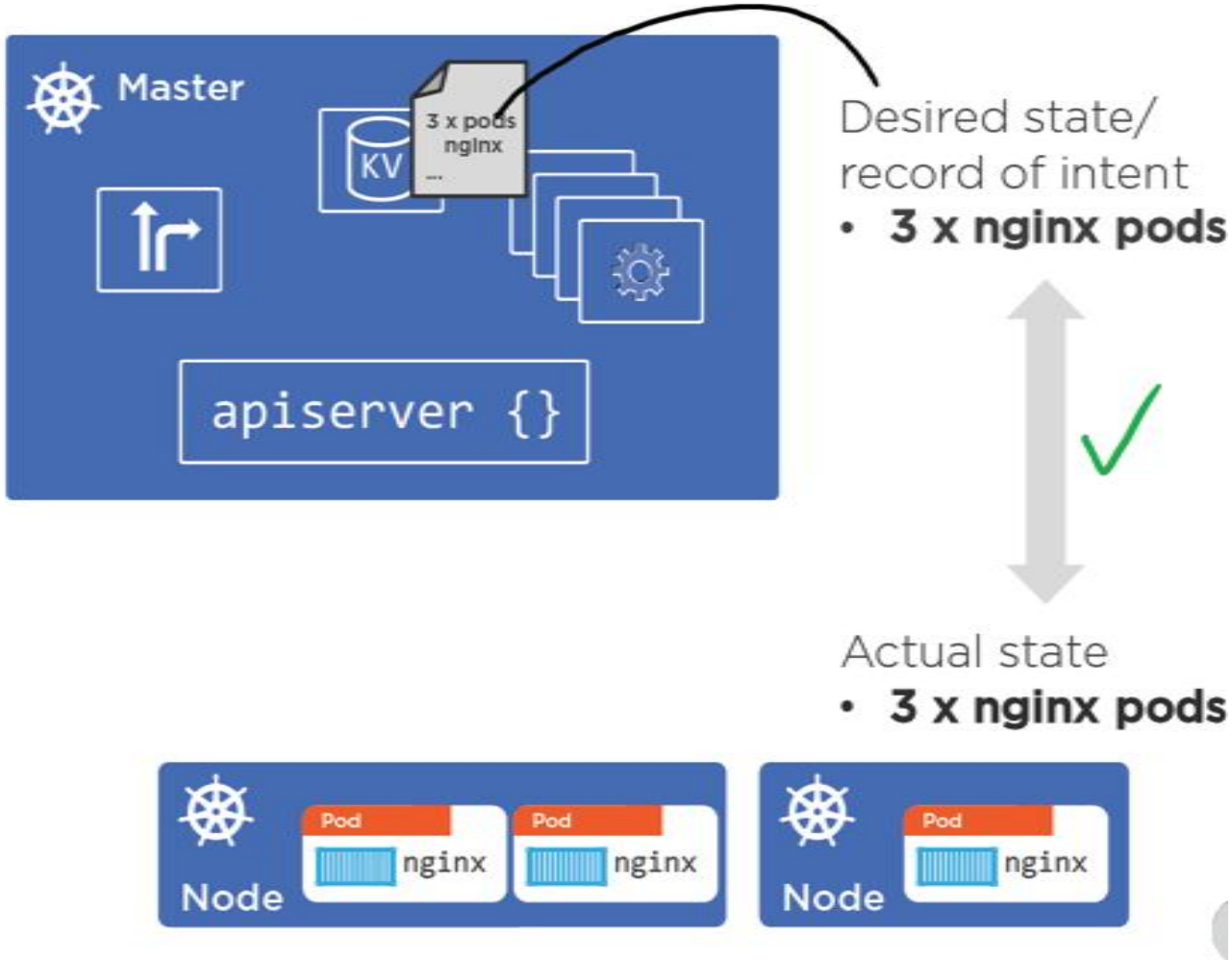


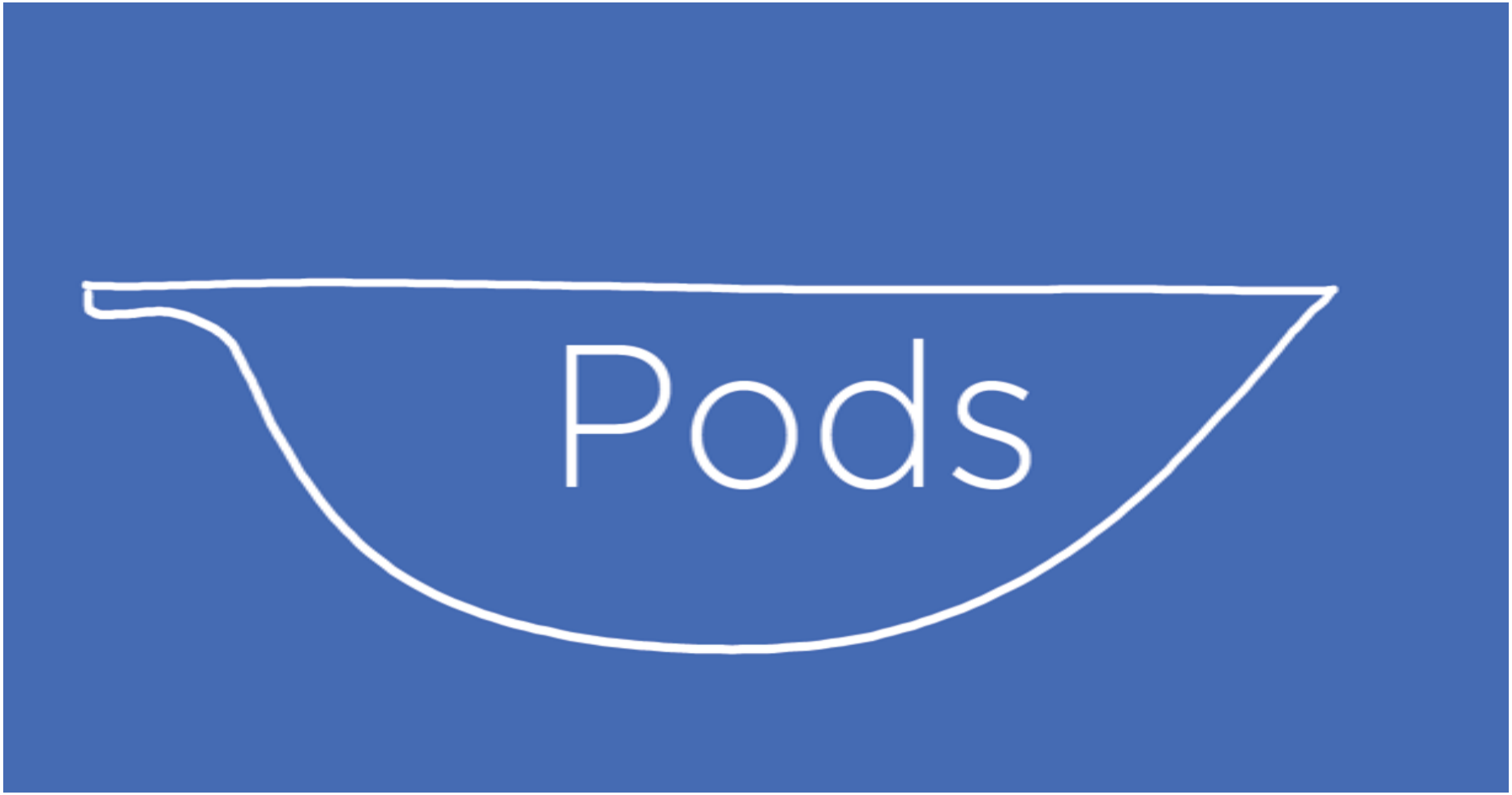
Desired state/
record of intent

- **3 x nginx pods**

Actual state

- **2 x nginx pods**


Master
3 x pods
nginx
...
KV
apiserver {}
Desired state/
record of intent
• 3 x nginx pods
Actual state
• 3 x nginx pods
Pod
nginx
Node
Pod
nginx
Node
Pod
nginx
Node


Pods

VM

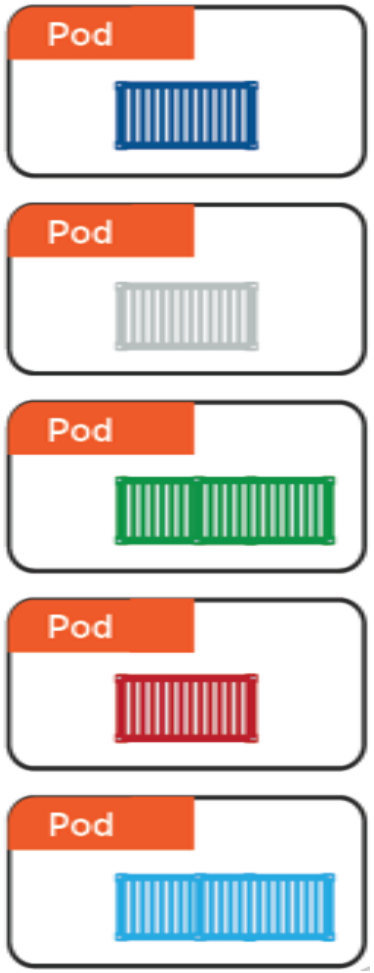
Container

Pod

Atomic units of scheduling

Containers always run inside of pods

Pods can have multiple containers
(advanced use-case)

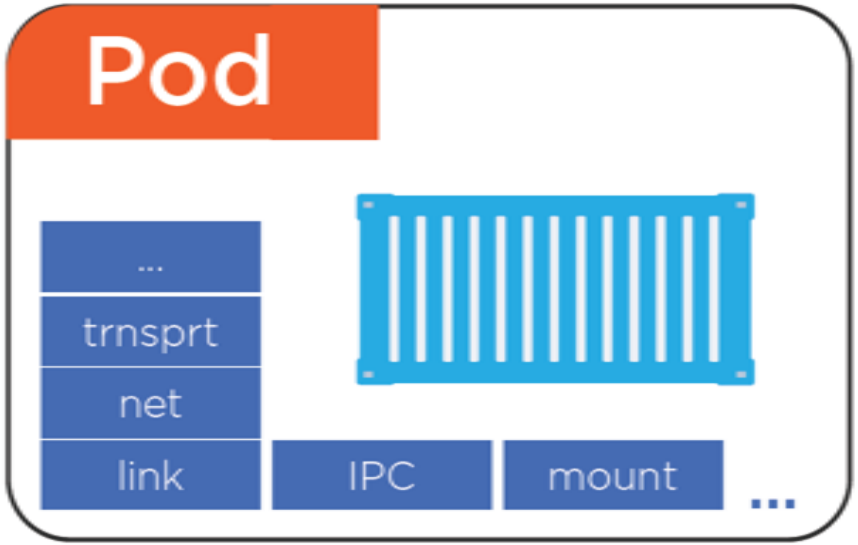


Ring-fenced environment

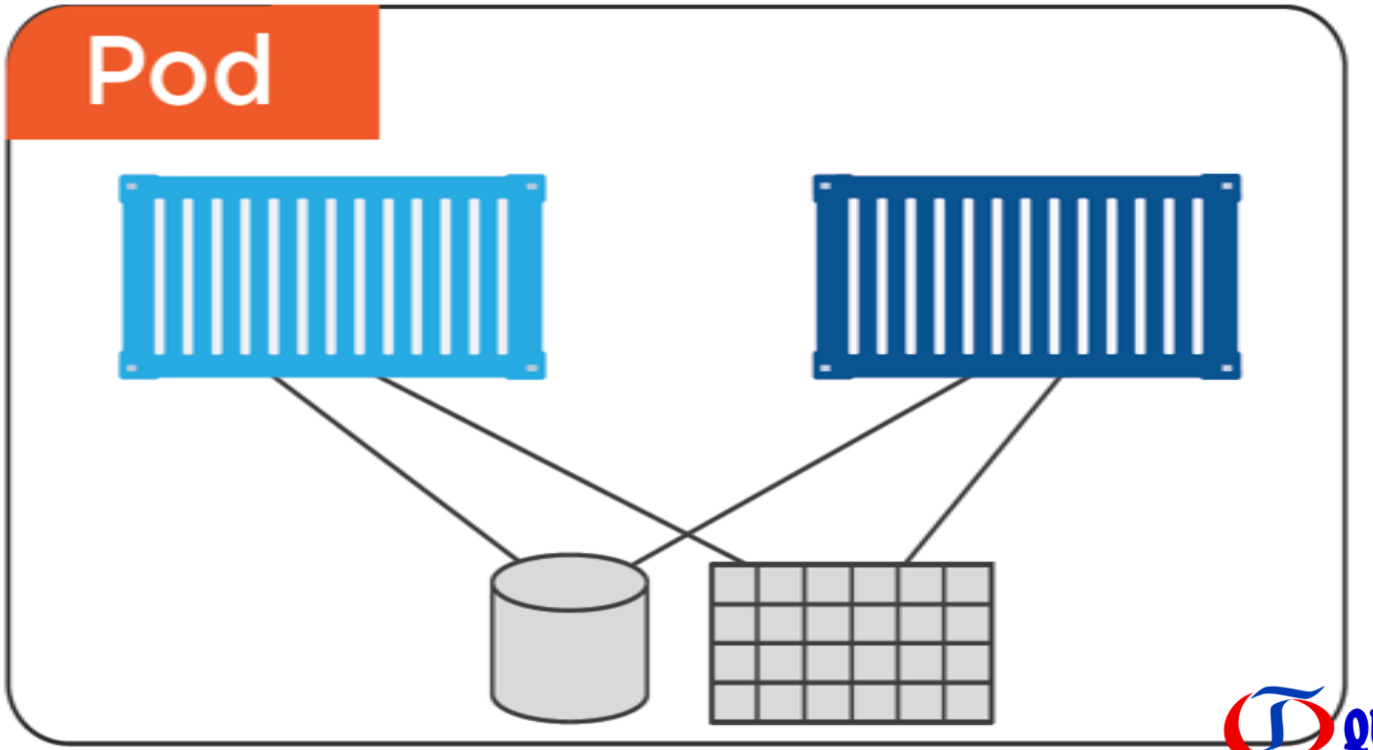
- Network stack
- Kernel namespaces
- ...

n containers

<u>All containers in pod share the pod environment</u>

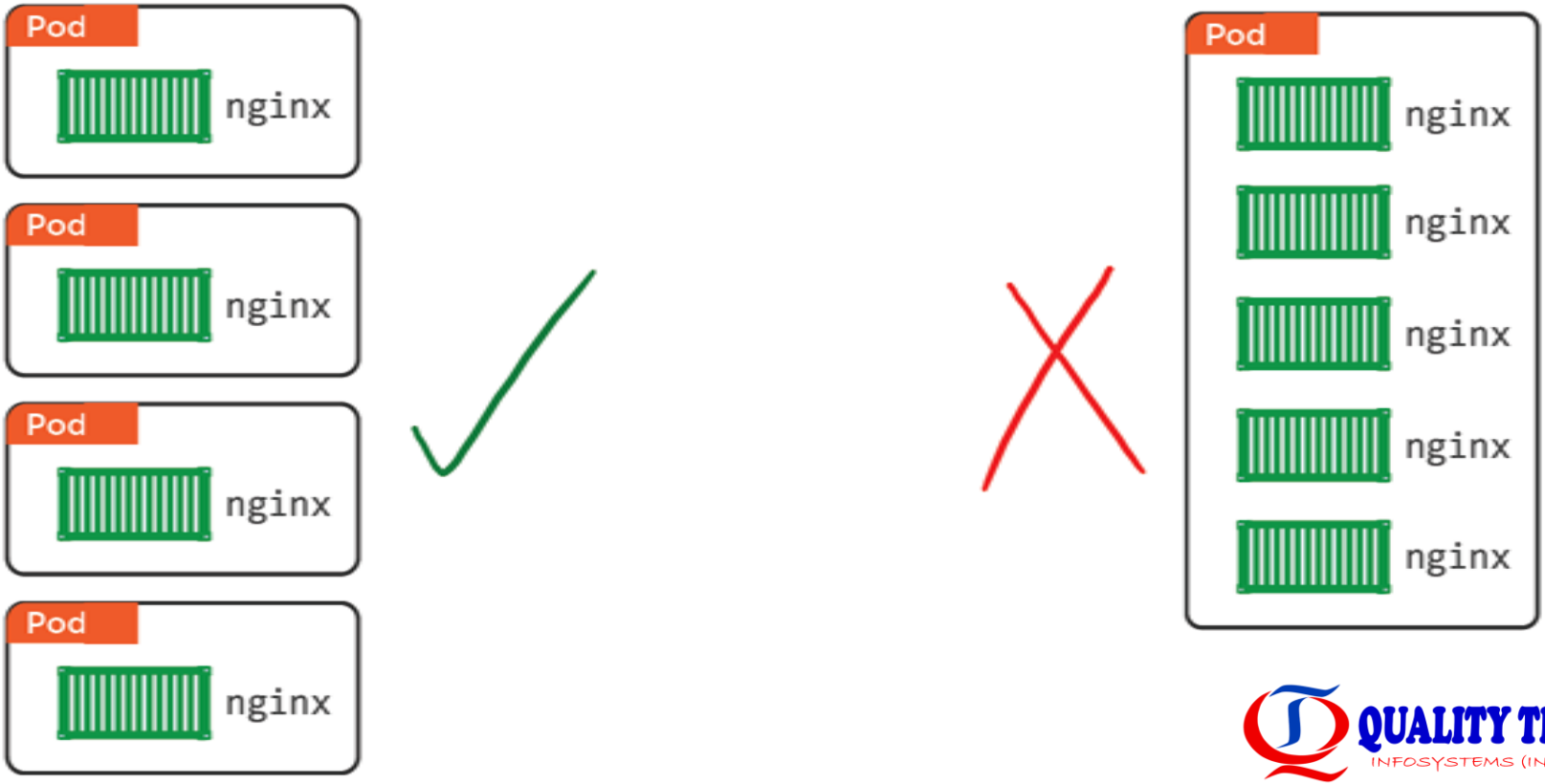


Tight Coupling

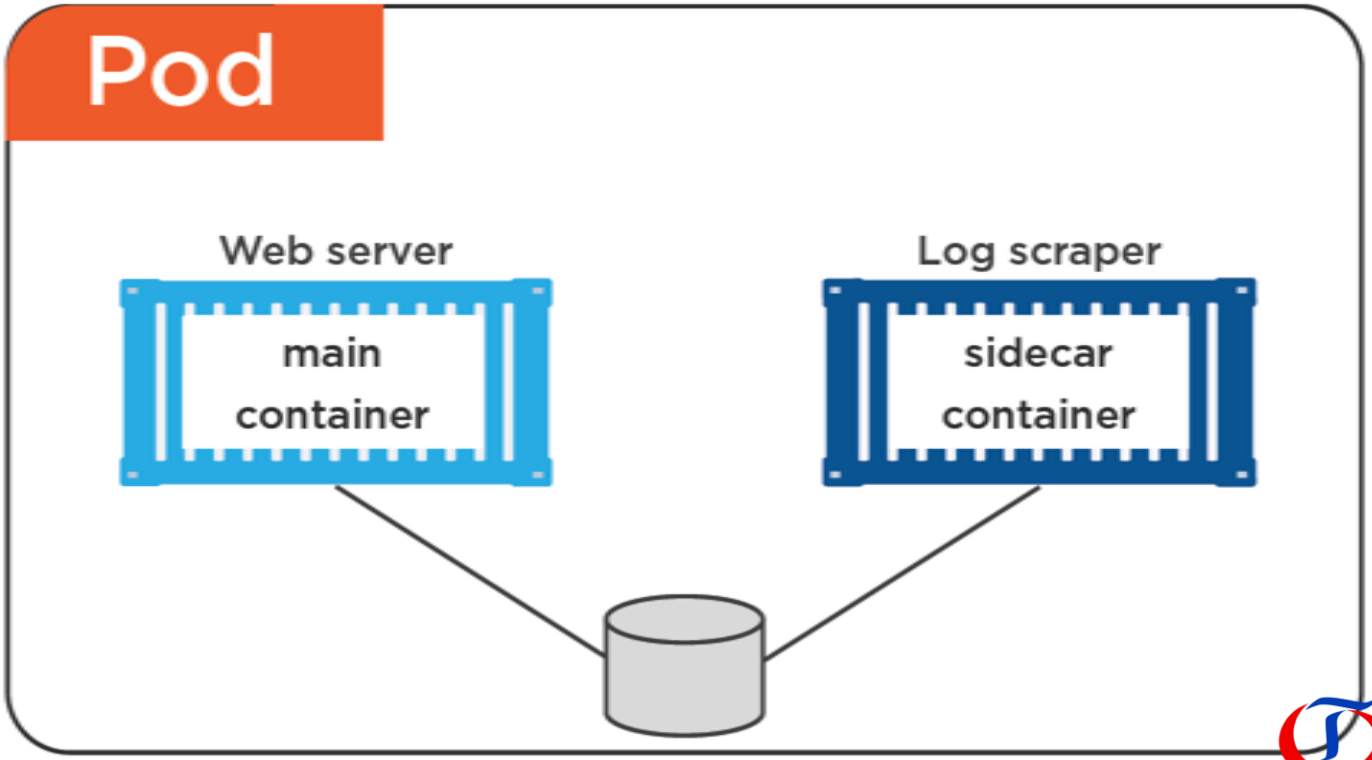


Loose Coupling

Pods and Scaling



Multi-container Pods



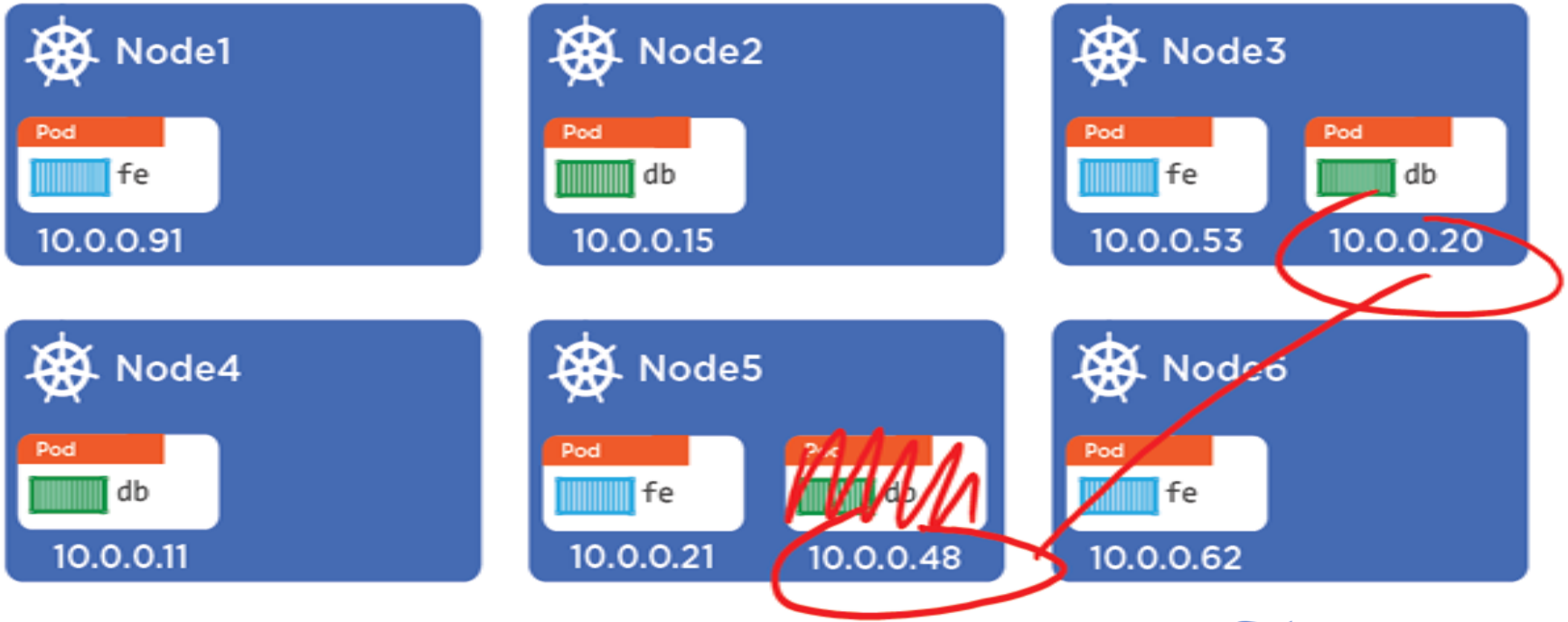
Pod Lifecycle

Phase: pending

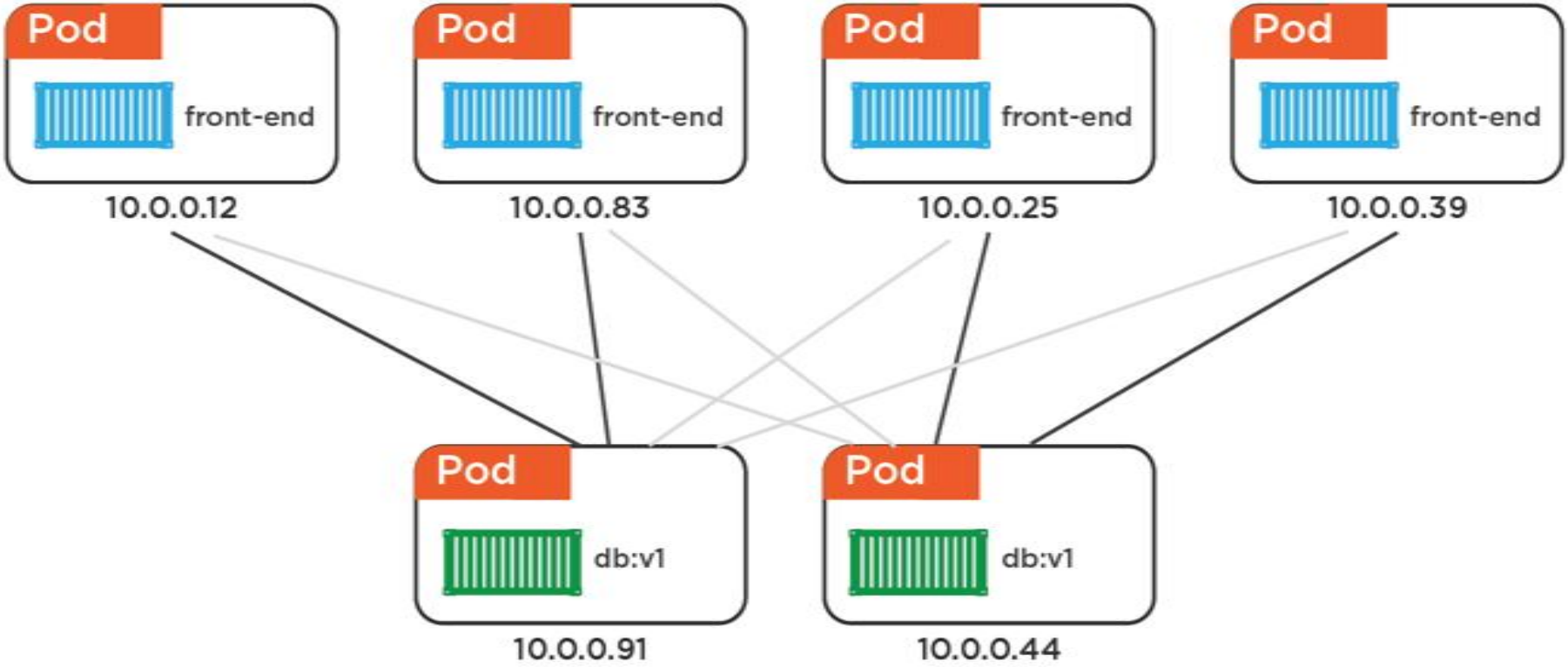
Phase: running

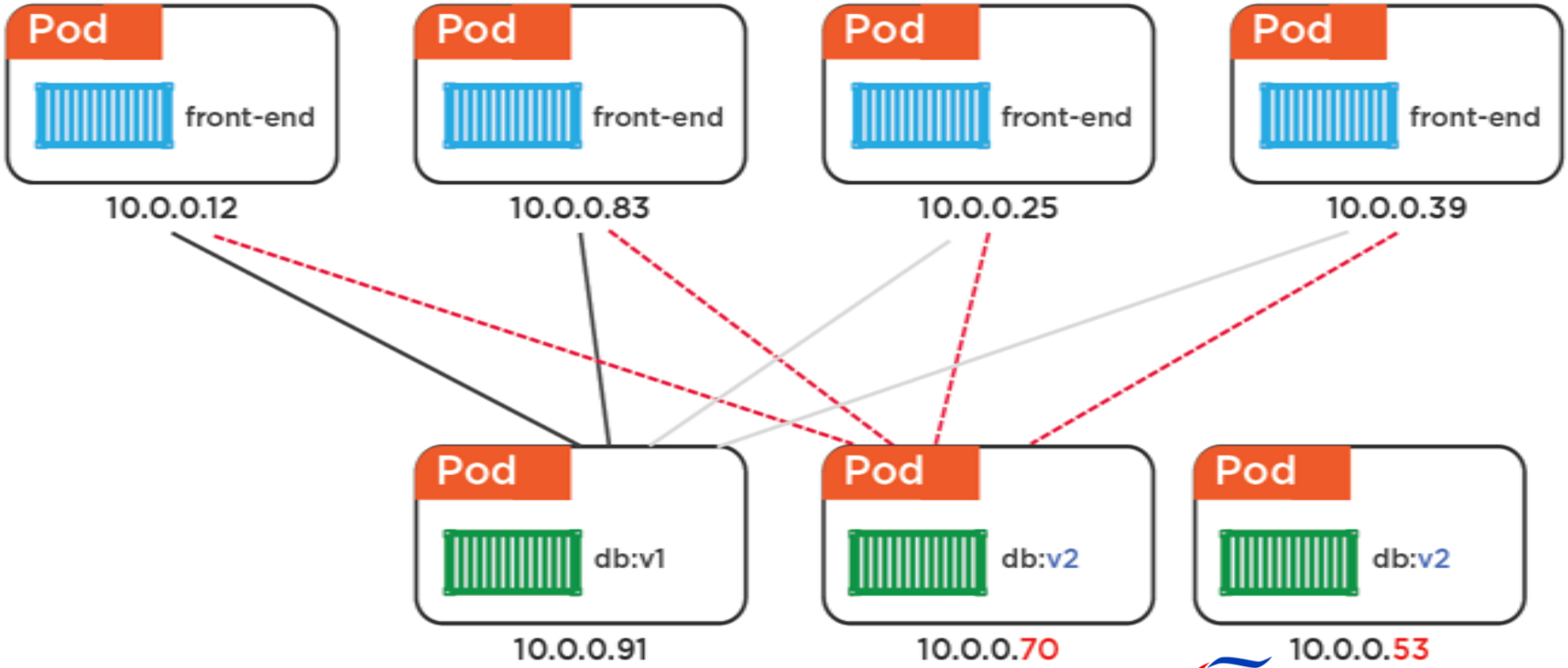
Phase: succeeded/failed

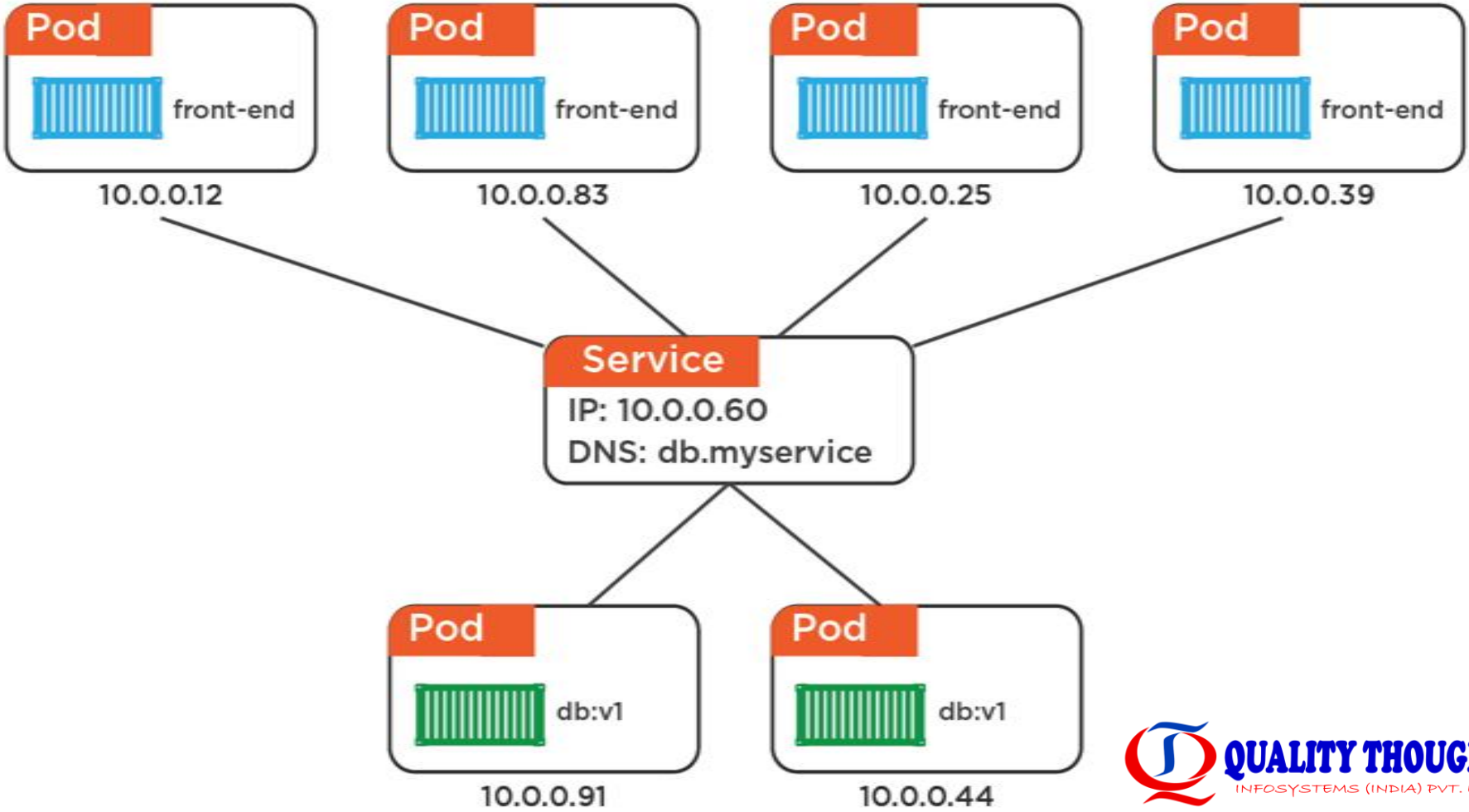
Services

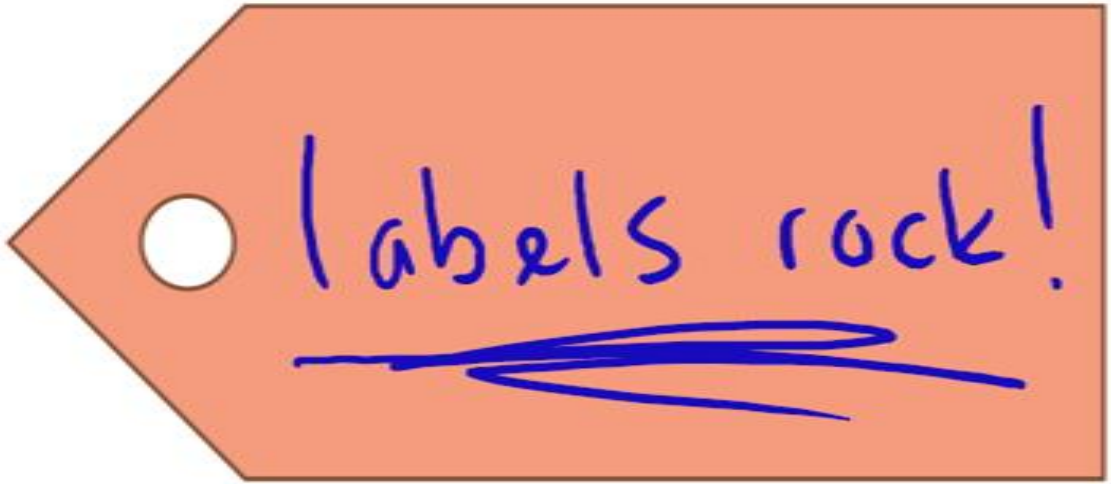


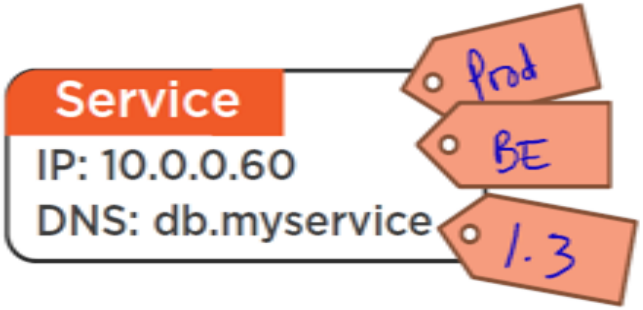
Every **new** pod gets a new IP = IP churn!

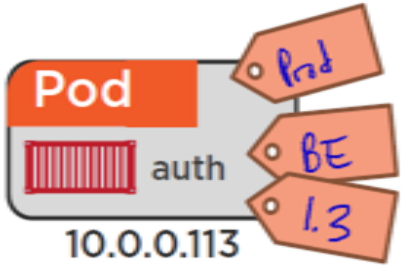


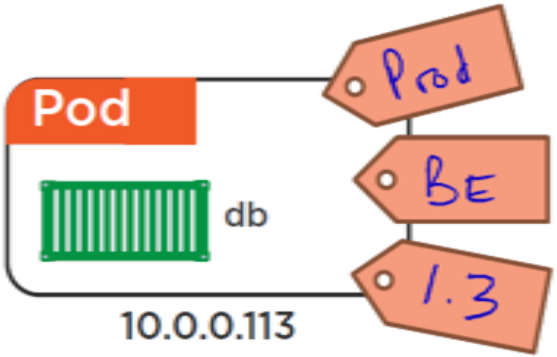


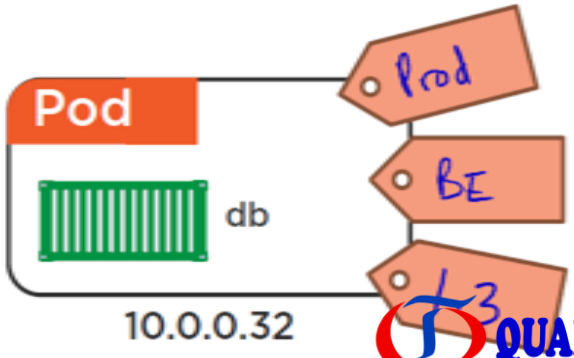


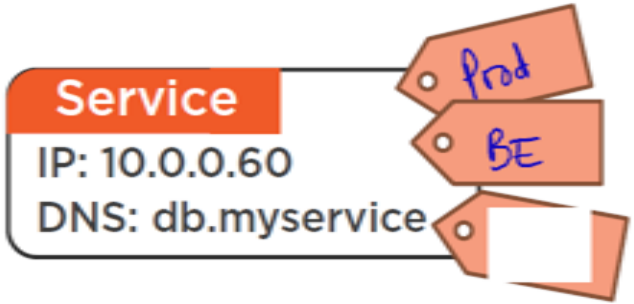


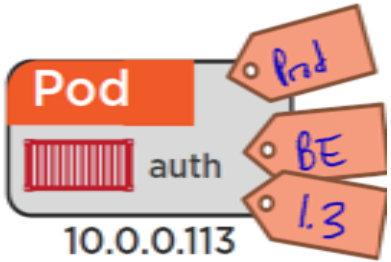


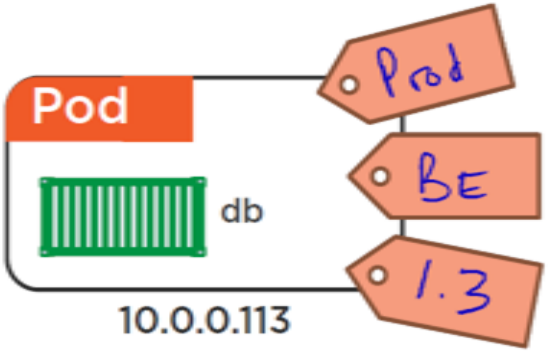


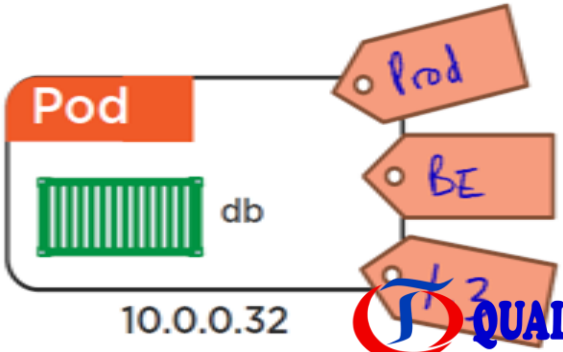


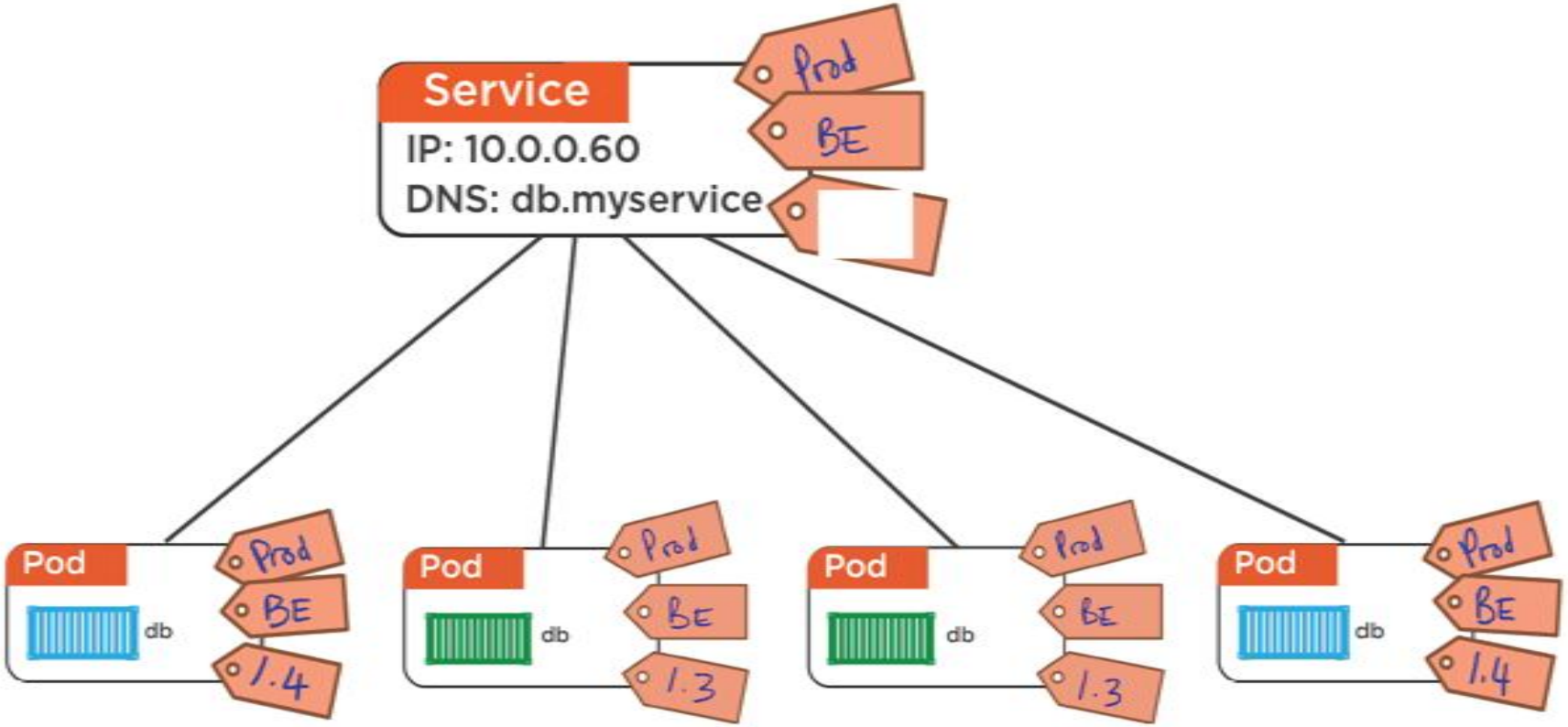


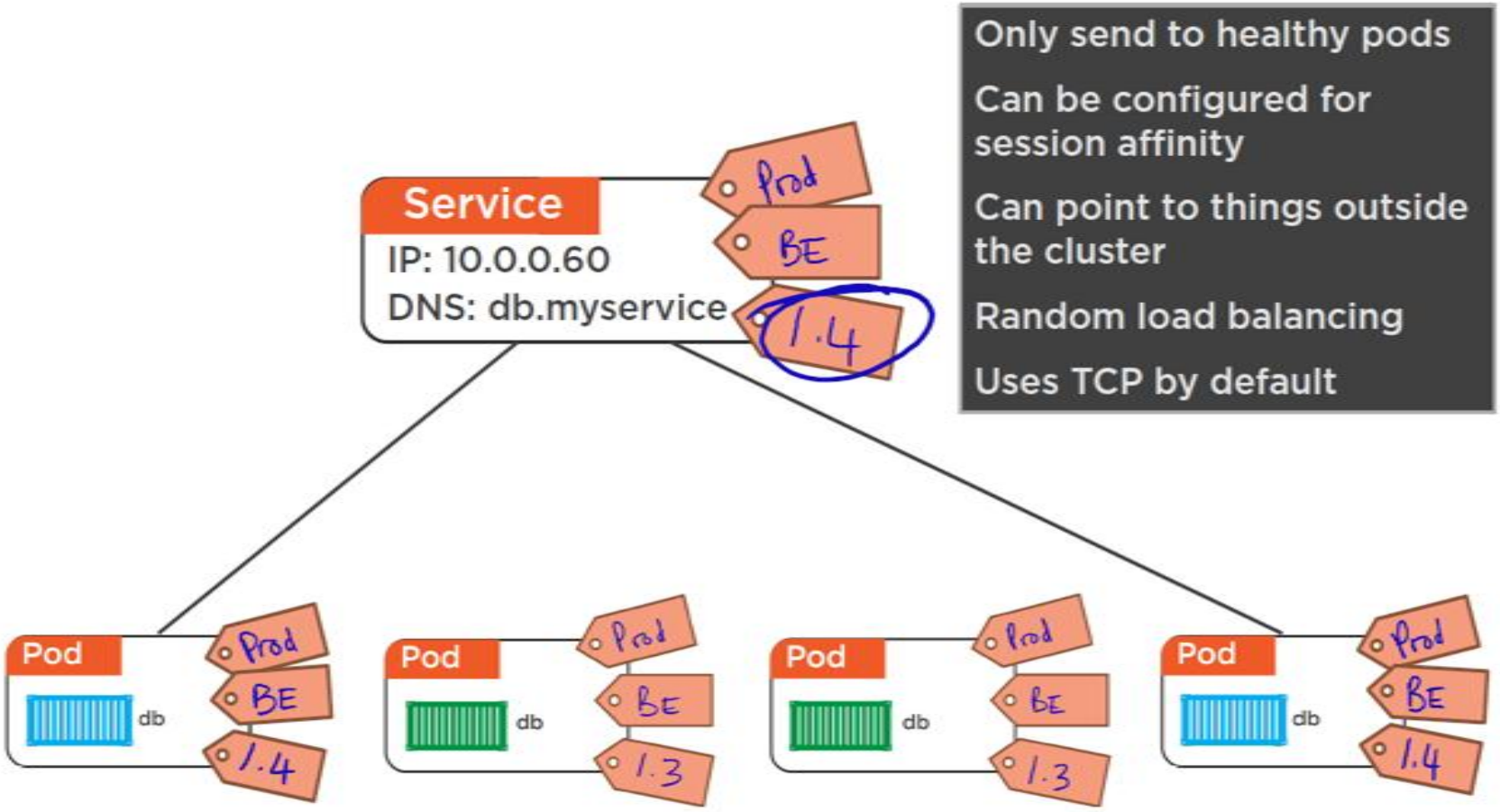




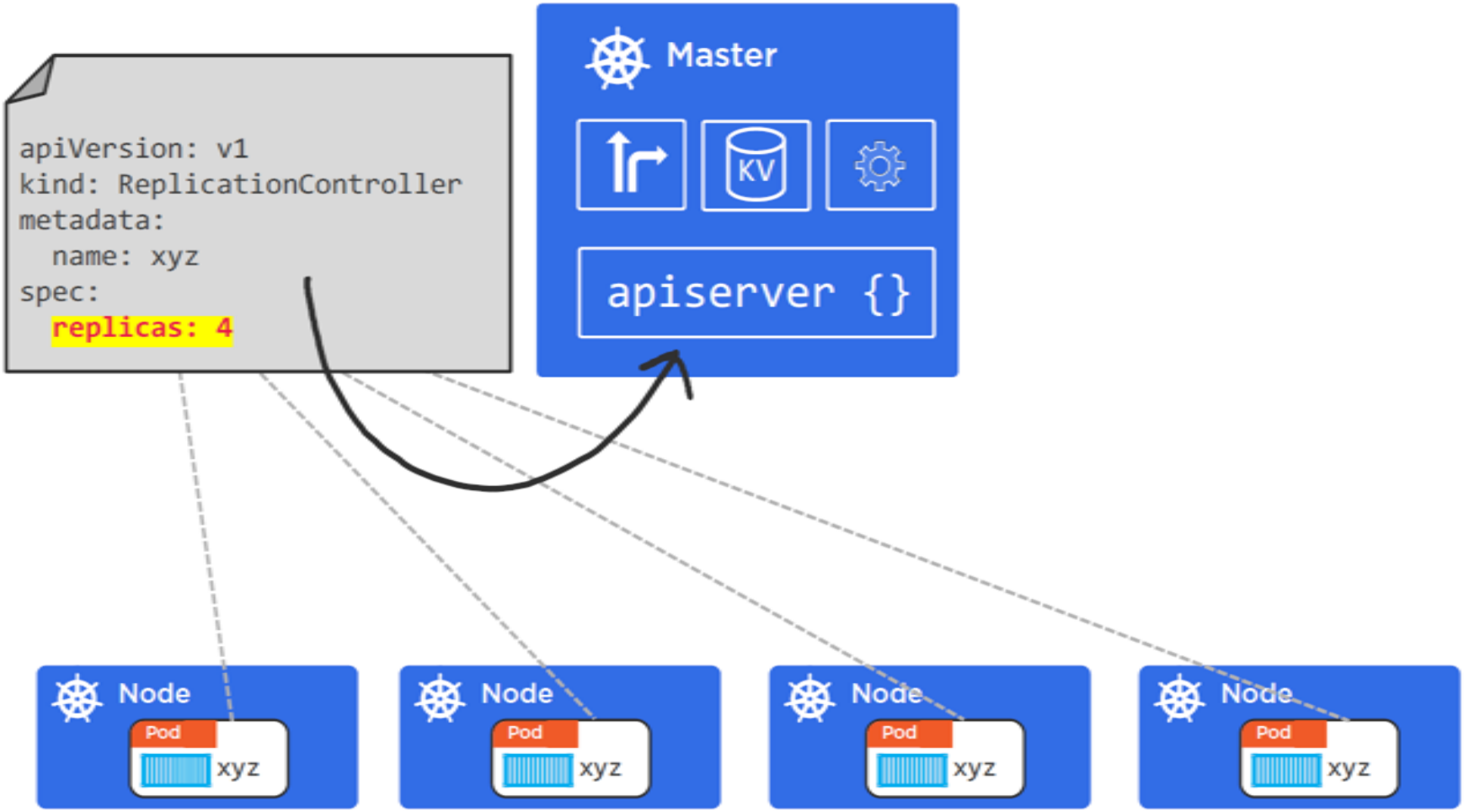




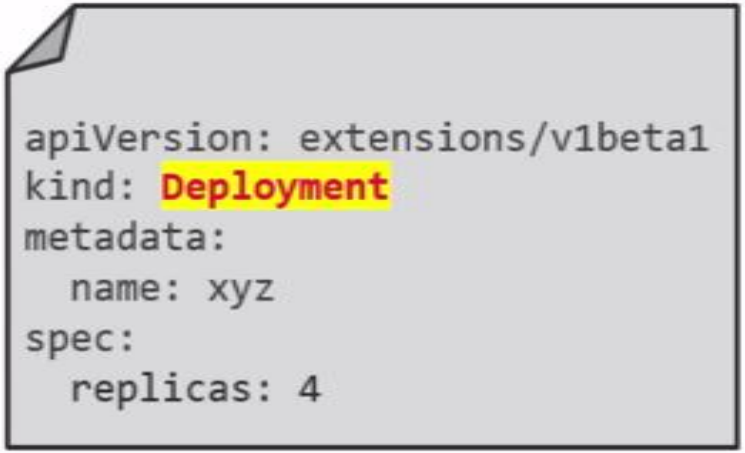

Service
IP: 10.0.0.60
DNS: db.myservice
Prod
BE
Pod
db
Prod
BE
1.4
Pod
db
Prod
BE
1.3
Pod
db
Prod
BE
1.3
Pod
db
Prod
BE
1.4


Only send to healthy pods
Can be configured for session affinity
Can point to things outside the cluster
Random load balancing
Uses TCP by default
Service
IP: 10.0.0.60
DNS: db.myservice
Prod
BE
1.4
Pod
db
Prod
BE
1.4
Pod
db
Prod
BE
1.3
Pod
db
Prod
BE
1.3
Pod
db
Prod
BE
1.4

Deployments


apiVersion: v1
kind: ReplicationController
metadata:
name: xyz
spec:
replicas: 4
Master
KV
apiserver {}
Node
Pod
xyz
Node
Pod
xyz
Node
Pod
xyz
Node
Pod
xyz

Deployments are all about declarations



REST objects

Self documenting

Deployed via YAML or JSON manifests

Spec-once deploy-many

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: xyz
spec:
  replicas: 4
```

Simple rolling updates and rollbacks

Add features to Replication Controllers (Replica Sets) RC v2

Versioned

Deployed via the `apiserver`

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: xyz
spec:
  replicas: 4
```

Simple rolling updates and rollbacks

Multiple concurrent versions
- Blue-green deployments
- Canary releases

Simple versioned rollbacks

Objects in the K8s API

Pods : Atomic unit of scheduling...

Replication Controllers : Scale pods, desired state etc...

Deployments : RC + rolling updates, rollbacks...

Services : Stable networking...

Installing Kubernetes

VS

VS

Simplest

Most involved

Minikube

Google Container Engine (GKE)

AWS Provider

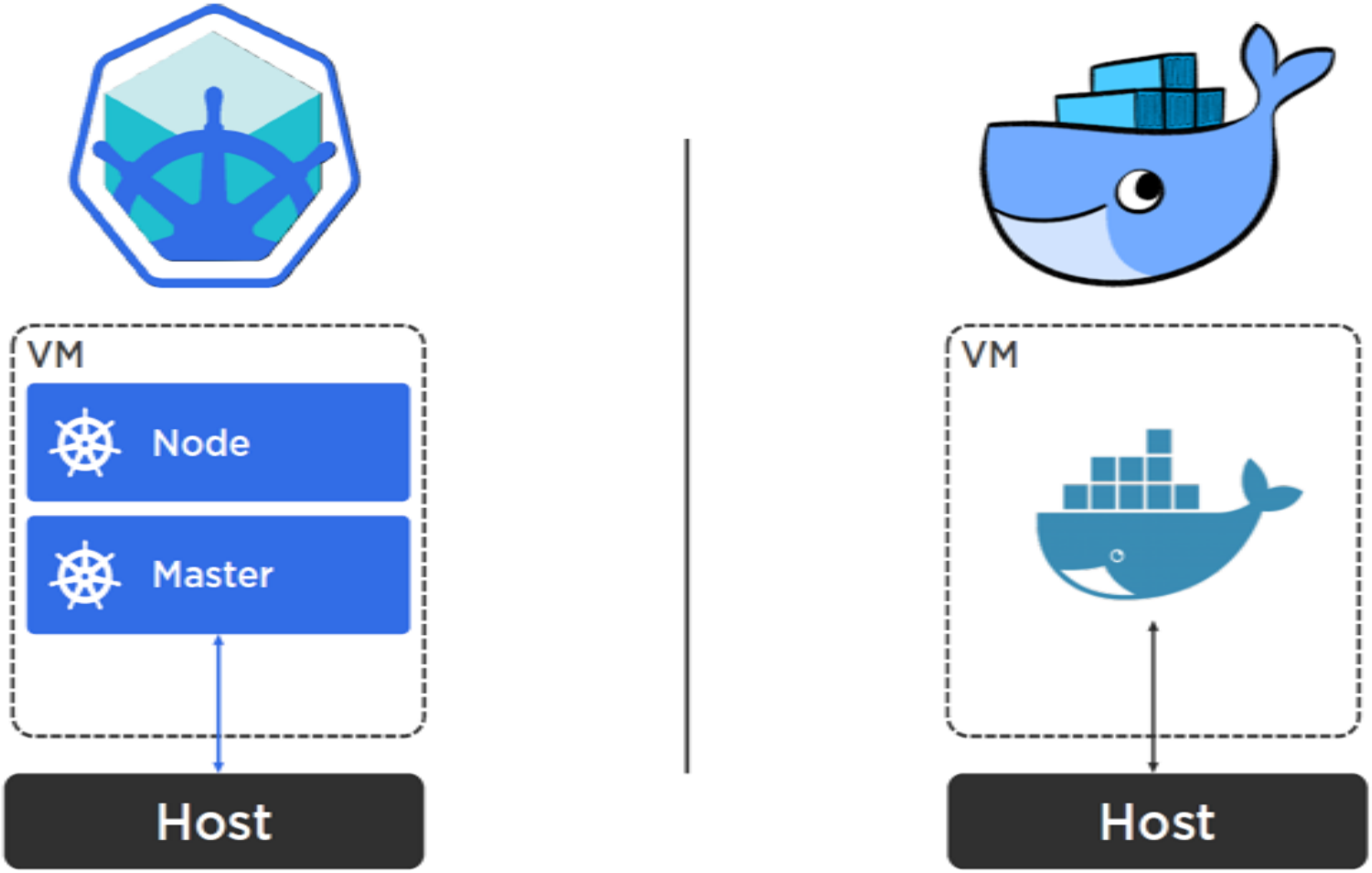
Manual install

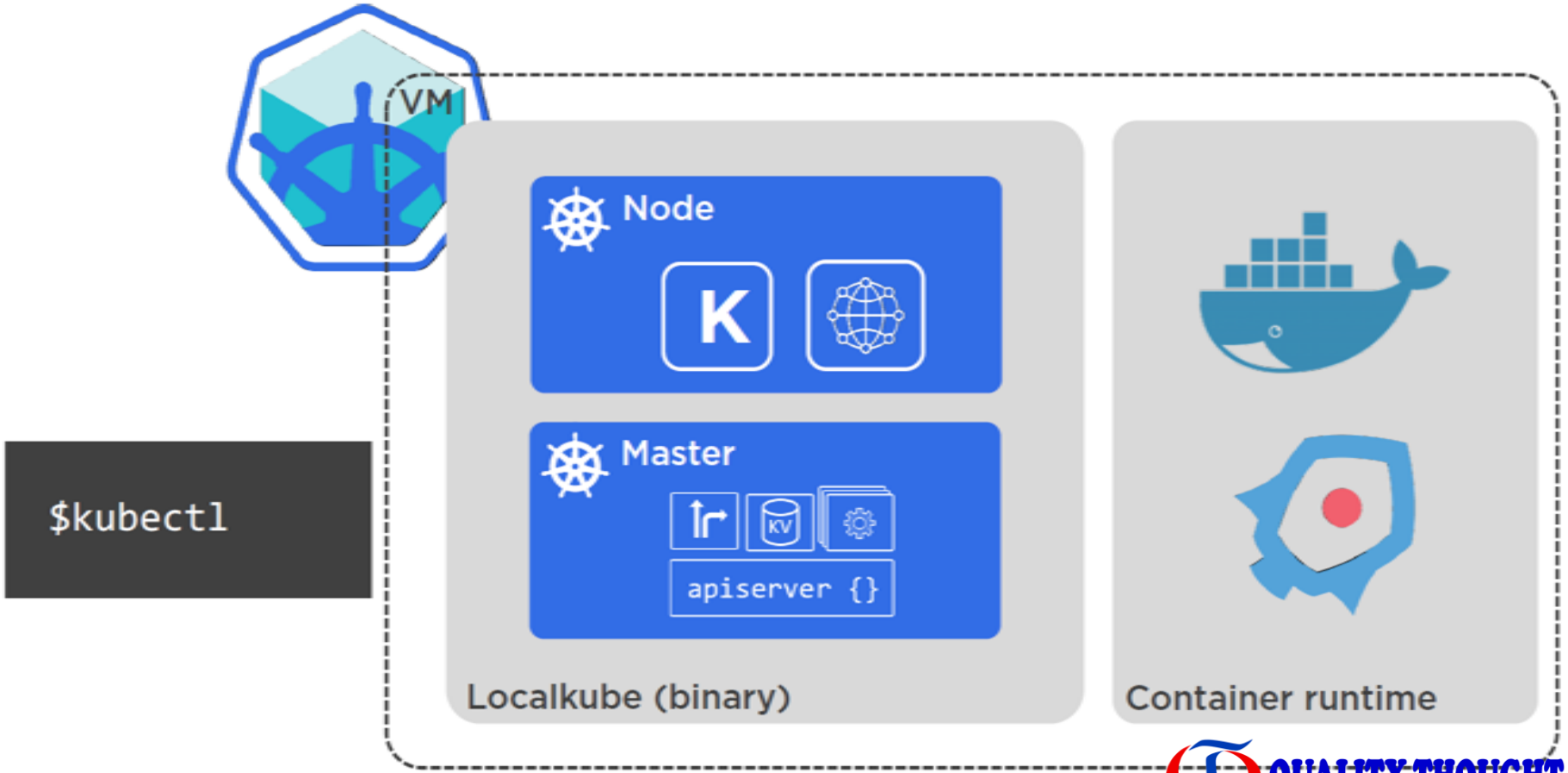
Minikube

minikube

DFM
DFW

Best way of spinning up a local environment


VM
Node
Master
Host
VM
Host



minikube

Best way of spinning up a local k8s environment

INSTALLING MINIKUBE ON WINDOWS 10

```
# Get the kubectl.exe binary from this URL and copy it into your %PATH%

https://storage.googleapis.com/kubernetes-release/release/v1.6.0/bin/windows/amd64/kubectl.exe

# Get the Minikube installer from GitHub and install

https://github.com/kubernetes/minikube/releases

minikube version

minikube start --vm-driver=hyperv --kubernetes-version="v1.6.0"

kubectl get nodes

minikube dashboard
```

Manually Installing k8s with `kubeadm`

Pre-reqs

Every node (master and minions) needs:

- Docker (or rkt)
- Kubelet
- Kubeadm
- Kubectl
- CNI

INSTALLING K8S WITH KUBEADM

Install kubeadm as mentioned in
https://kubernetes.io/docs/setup/independent/install-kubeadm/ or

```
apt-get update && apt-get install -y apt-transport-https
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb http://apt.kubernetes.io/ kubernetes-xenial main
EOF
apt-get update
apt-get install -y docker.io kubeadm kubectl kubelet kubernetes-cni
kubeadm init (as root)

kubectl get nodes
kubectl get pods --all-namespaces
kubectl apply --filename https://git.io/weave-kube-1.6

kubectl get nodes
kubectl get pods --all-namespaces
kubectl get nodes
```

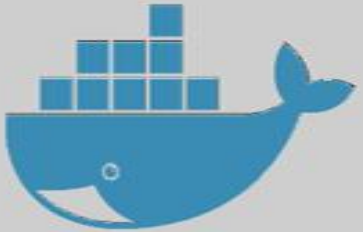
Working with Pods

Theory

Hypervisor Virtualization

VM

Atomic unit of scheduling



Container

Atomic unit of scheduling

Pod

Atomic unit of scheduling