

# Building a Kubernetes platform

Chris Nesbitt-Smith





# Chris Nesbitt-Smith



UK Gov | LearnK8s | Control Plane | lots of open source



learnk8s

# Multi-tenancy in Kubernetes

---



# **Multi-tenancy in Kubernetes**

---

**Isolation**

Ease of management

Cost efficiency



# **Multi-tenancy in Kubernetes**

---

Isolation

**Ease of management**

Cost efficiency



# **Multi-tenancy in Kubernetes**

---

Isolation

Ease of management

**Cost efficiency**

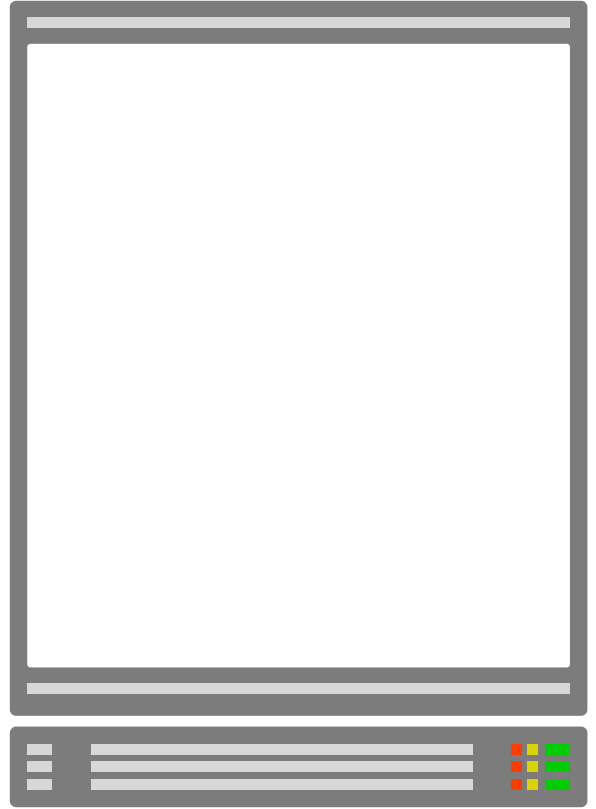
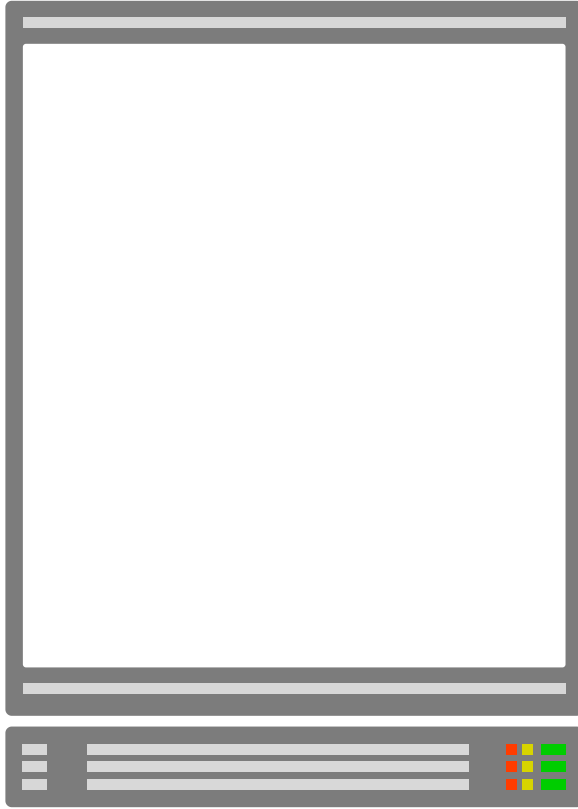


# Datacentre as a single computer

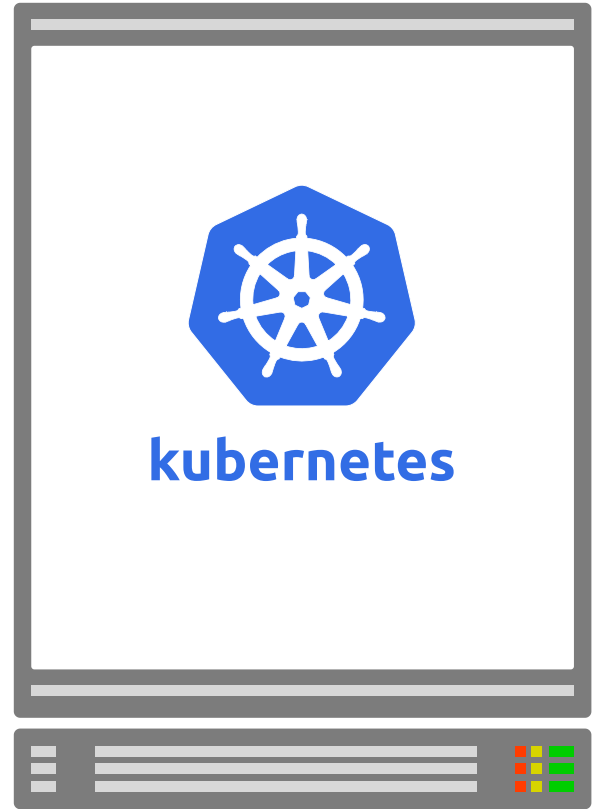
---

01





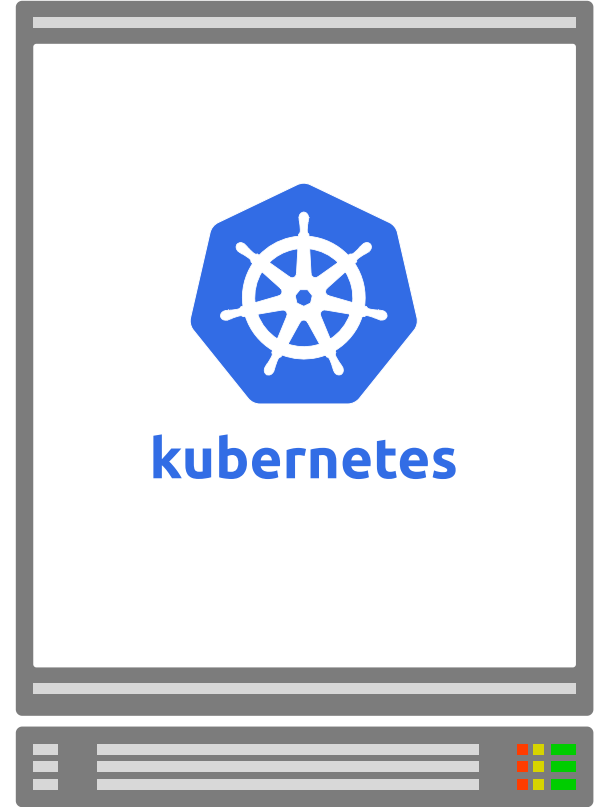
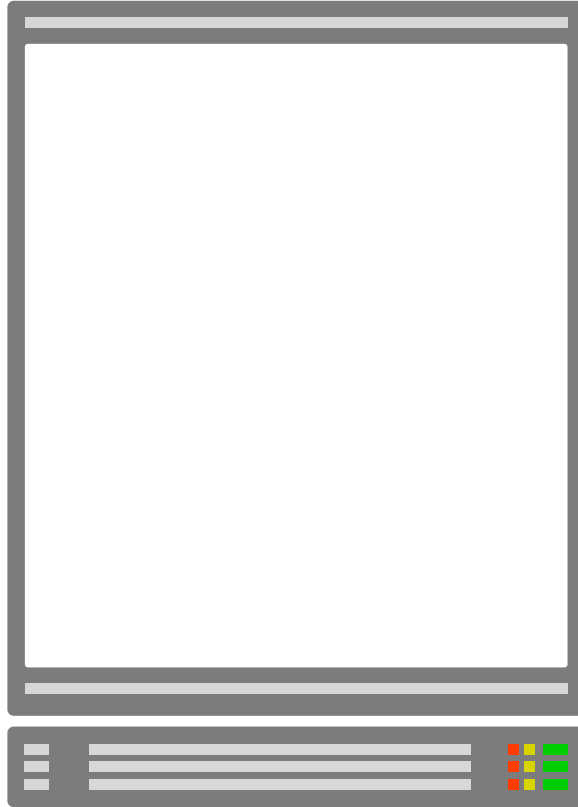




**Worker Node**

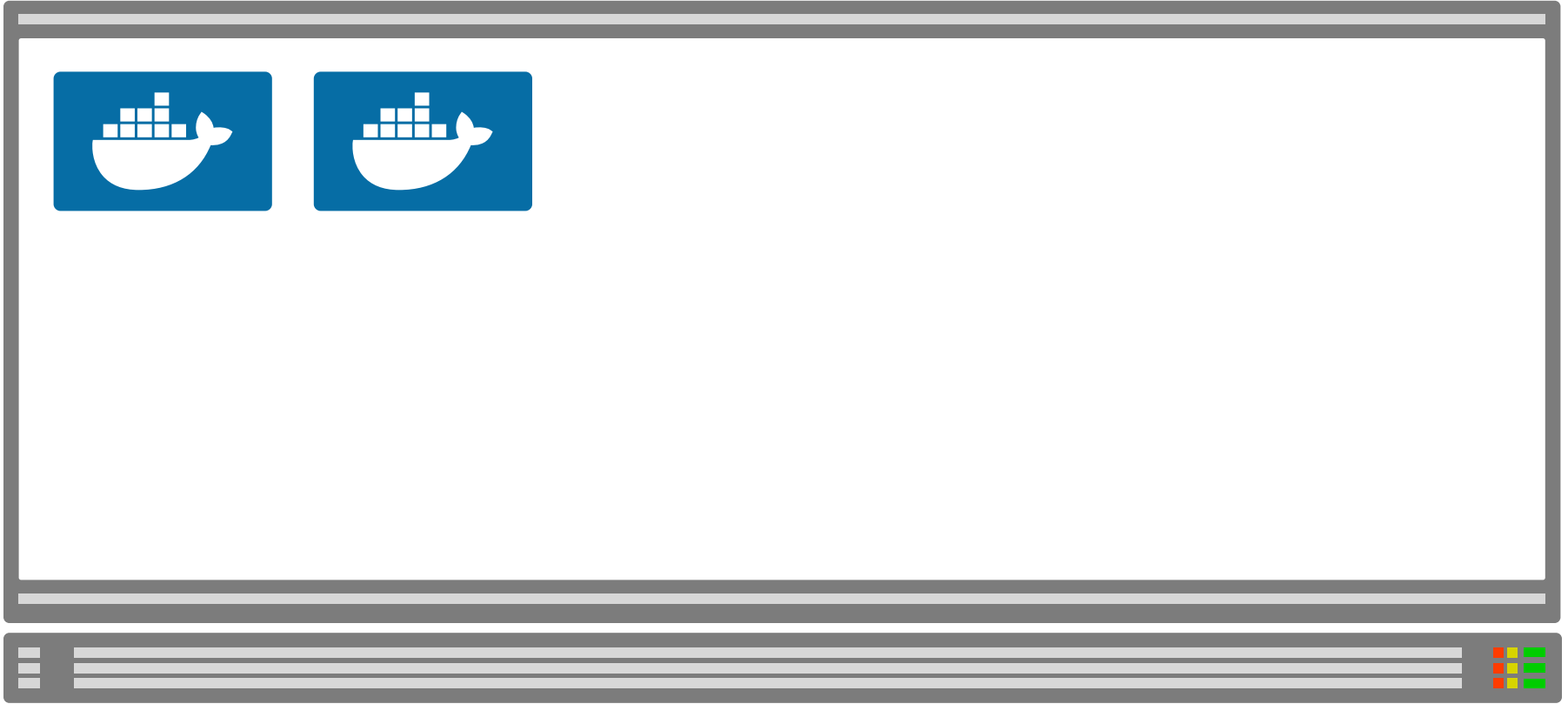


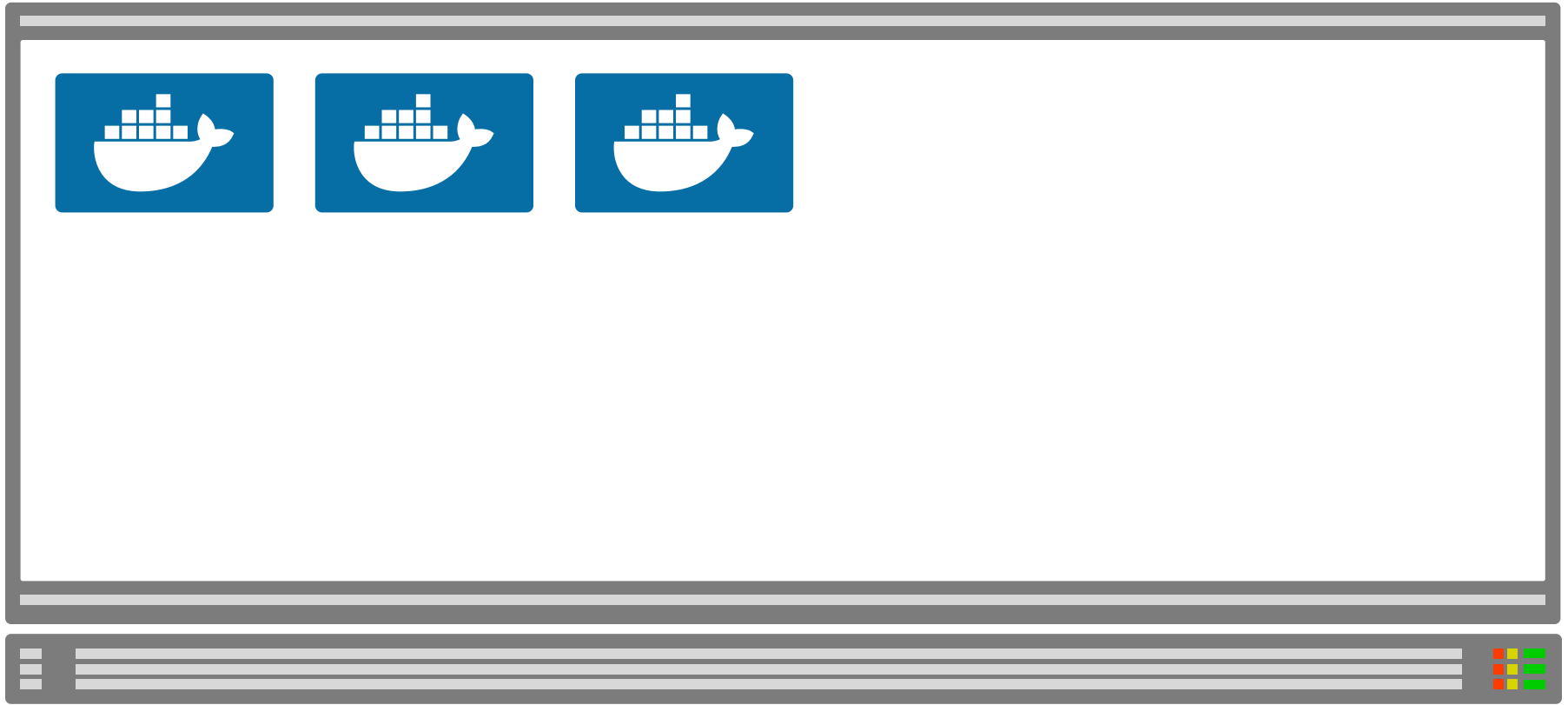
**Worker Node**



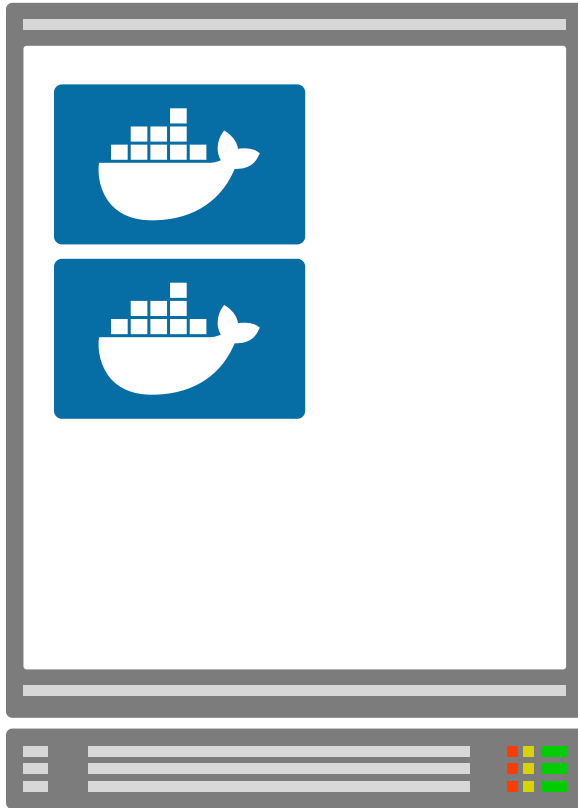




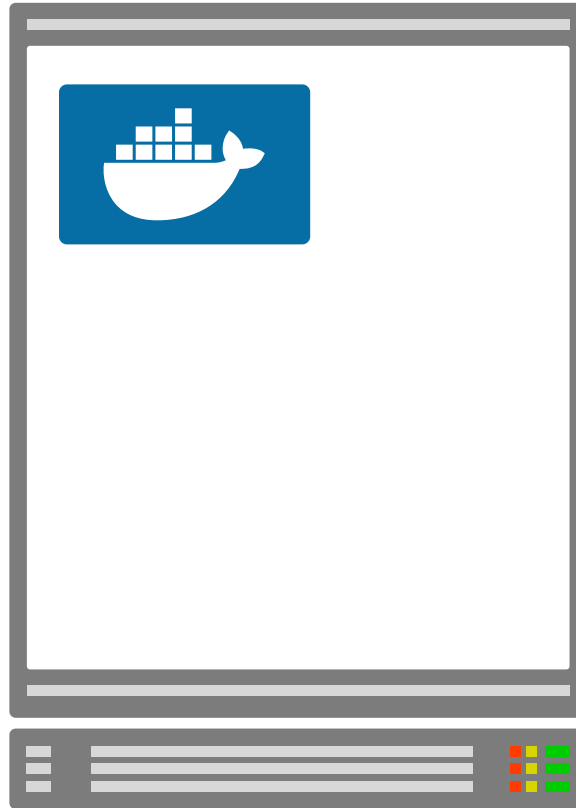




**Worker Node**



**Worker Node**

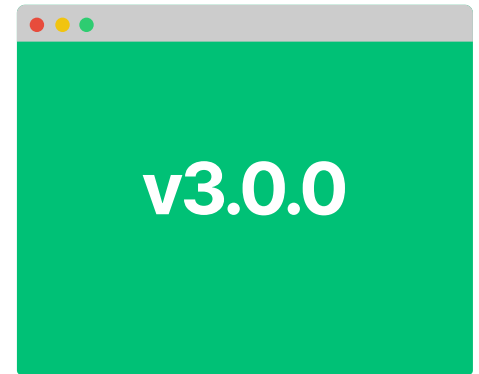


# Namespaces

---







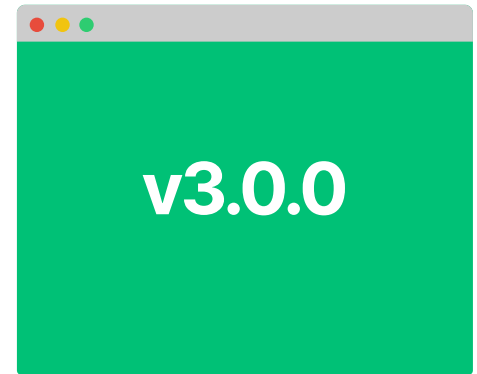
**namespace**



**namespace**



**namespace**



**namespace**

**namespace**

**namespace**

**v1.0.0**

**v2.0.0**

**v3.0.0**



# Environments x tenants

---



**Team A**

**Team B**

**Team B**

**v1.0.0**

**v2.0.0**

**v3.0.0**



dev

test

prod

**v1.0.0**

**v2.0.0**

**v3.0.0**



# Team A

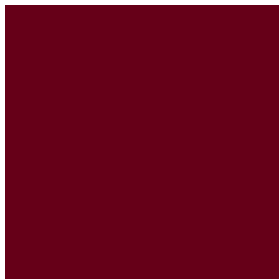
**dev**



**test**



**prod**



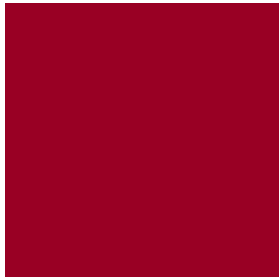
**Team A**

**Team B**

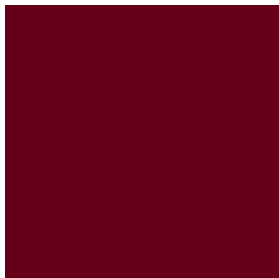
**dev**



**test**



**prod**





**Team A**

**Team B**

**Team C**

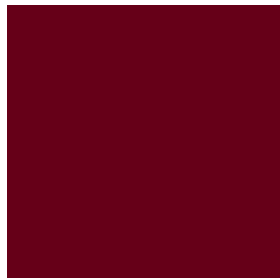
**dev**



**test**



**prod**



# Environments x tenants *at scale*

---



10 TENANTS

**dev**



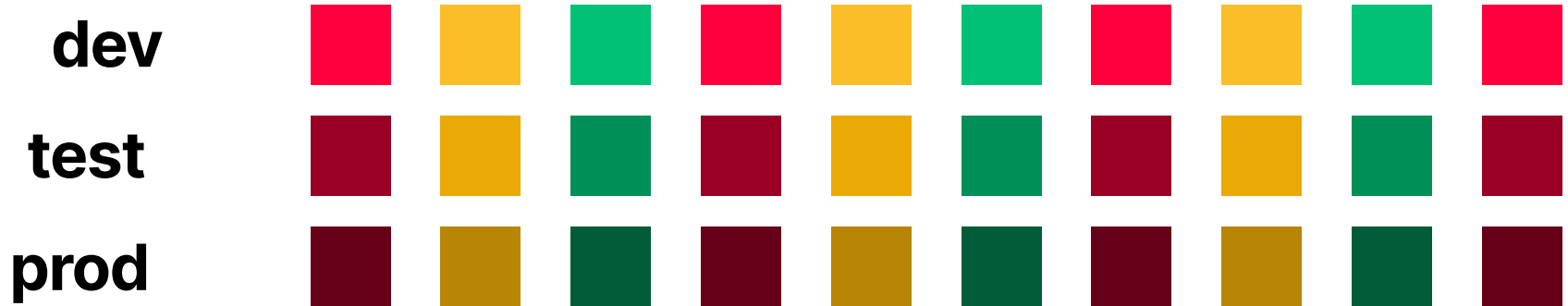
**test**



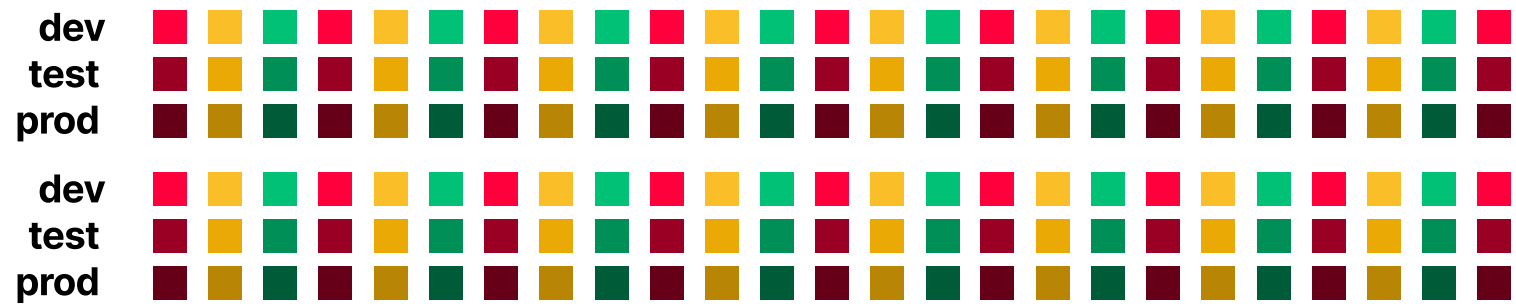
**prod**

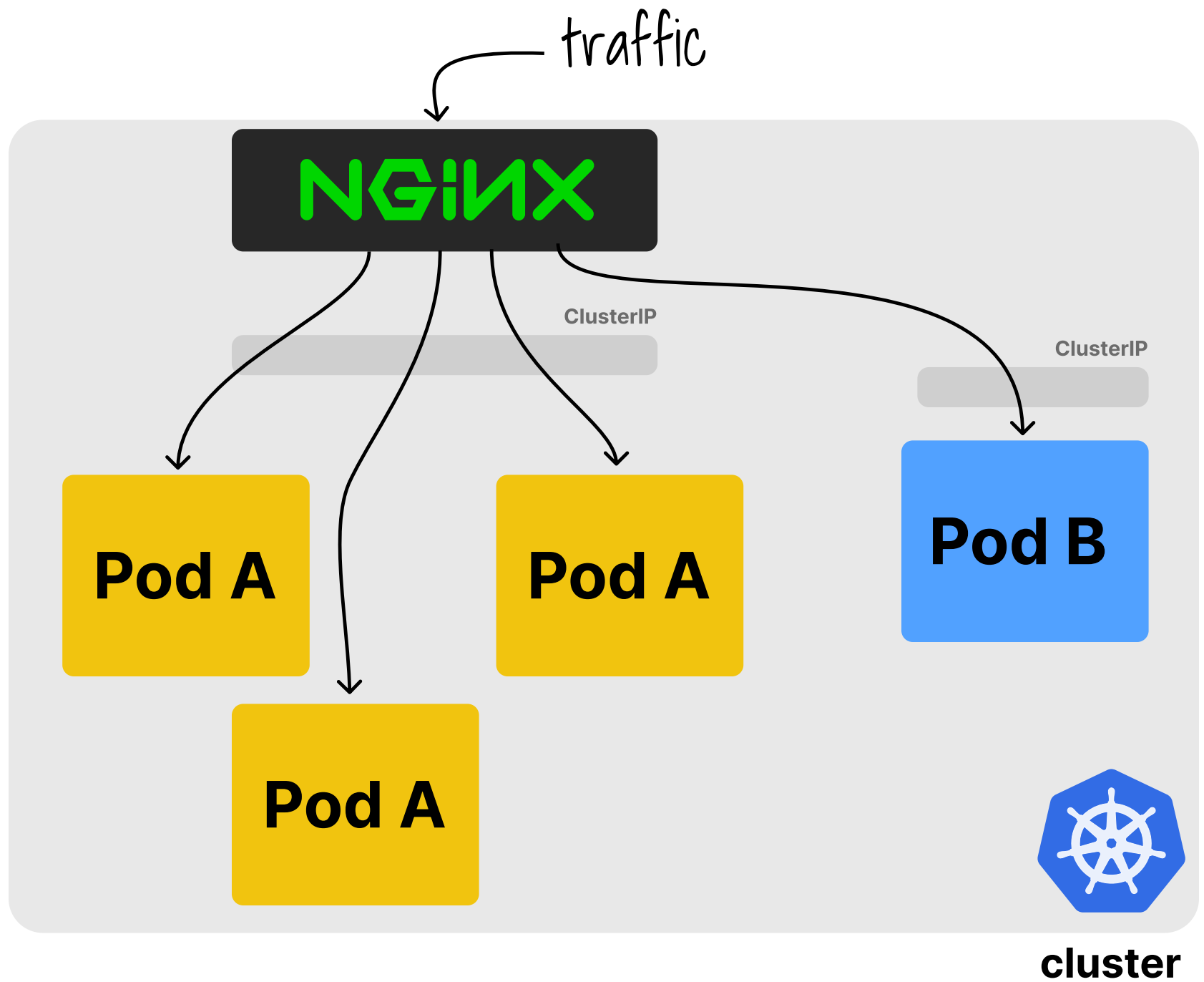


## 10 TENANTS



## 50 TENANTS





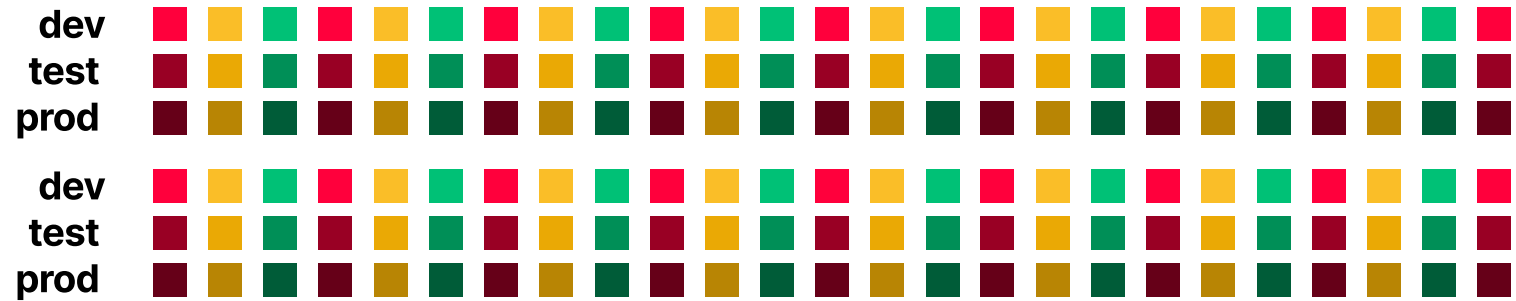
# NGINX

10 TENANTS

**dev**  
**test**  
**prod**



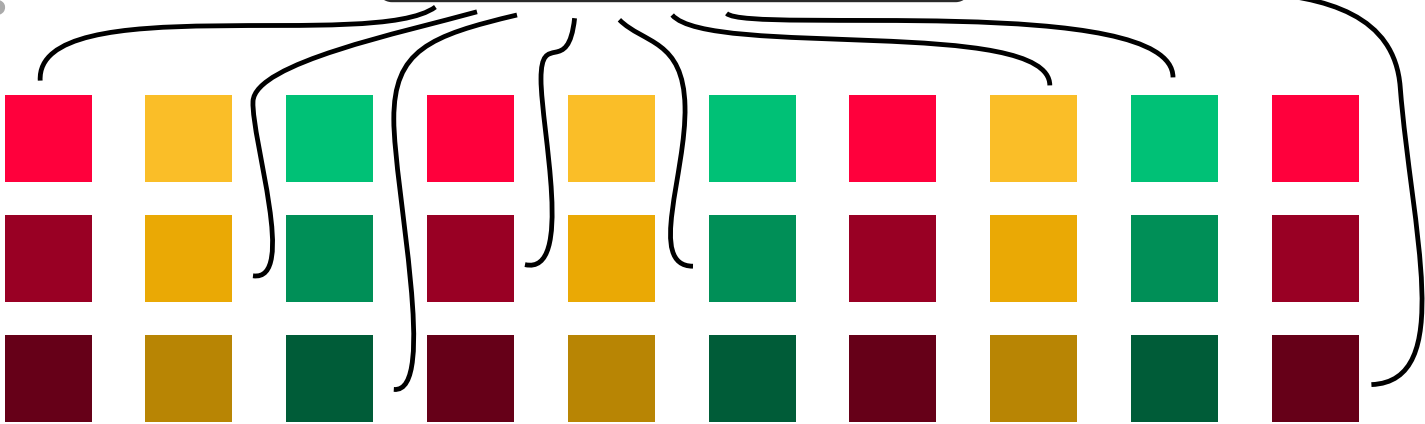
50 TENANTS





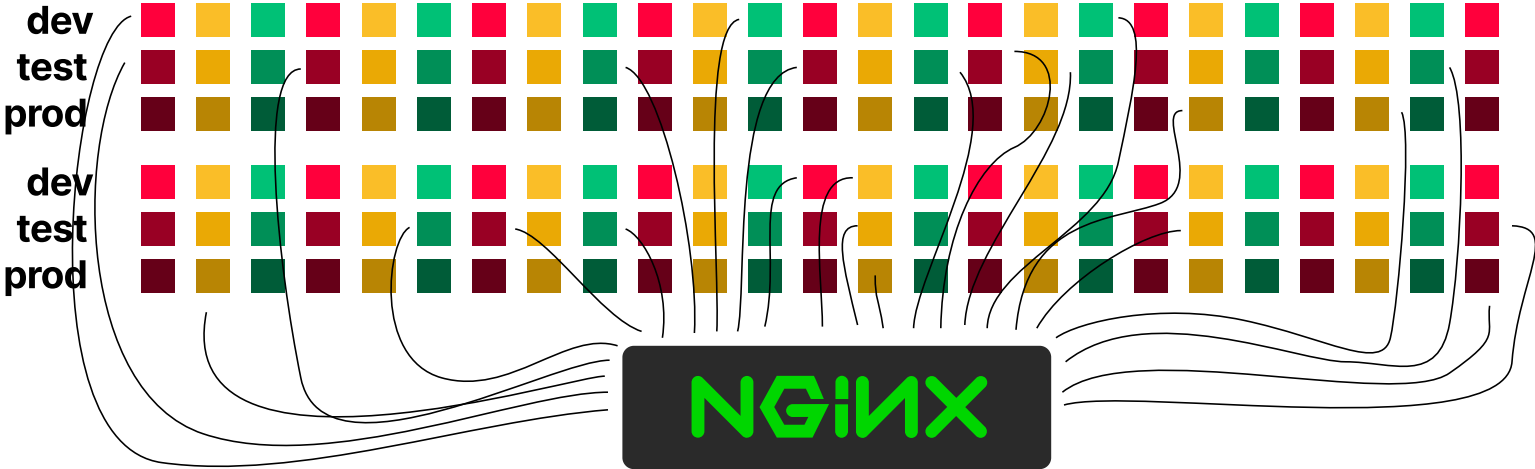
10 TENANTS

dev  
test  
prod

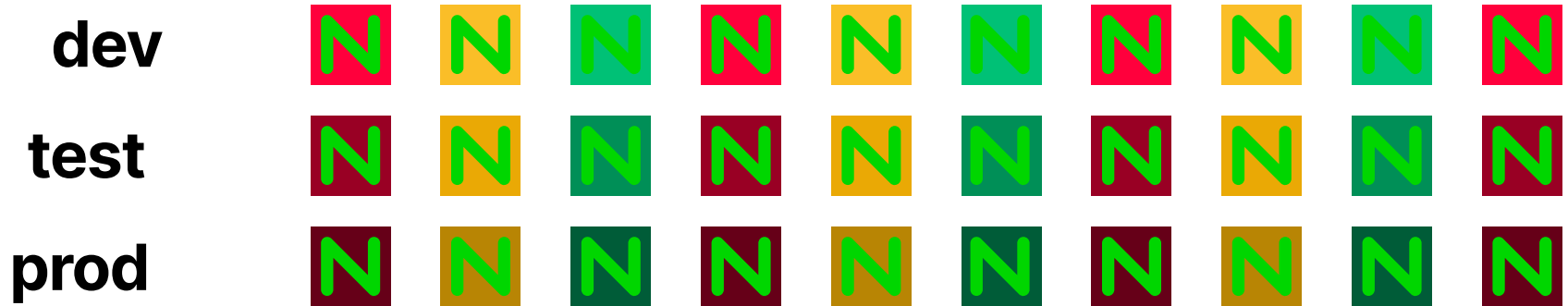


50 TENANTS

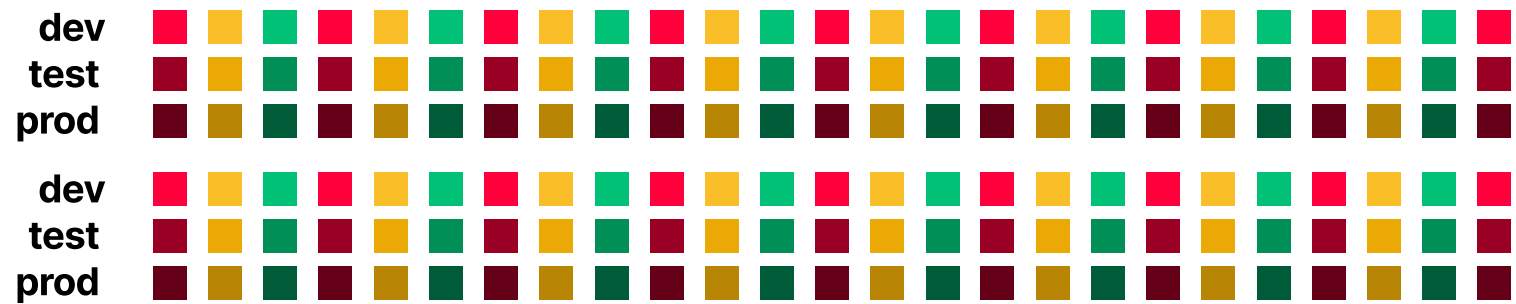
dev  
test  
prod  
  
dev  
test  
prod



## 10 TENANTS



## 50 TENANTS





## 10 TENANTS

# dev

**test**

**prod**

## 50 TENANTS

dev

**test**

**prod**

dev

**test**

**prod**



# **1 vs many: resources**

---





```
~$ cat values.yaml
```

```
...
```

```
resources:
```

```
  requests:
```

```
    cpu: 100m
```

```
    memory: 90Mi
```

ingress-nginx Helm chart



# Single Ingress

CPU

**100m**

MEMORY

**90Mi**



**Single Ingress**

**10 × 3**

CPU

**100m**

CPU

**3vCPU**

MEMORY

**90Mi**

MEMORY

**2.7GB**



**Single Ingress**

**10 × 3**

**50 × 3**

CPU

**100m**

CPU

**3vCPU**

CPU

**5vCPU**

MEMORY

**90Mi**

MEMORY

**2.7GB**

MEMORY

**4.5GB**



Instance Size	vCPU	Memory (GiB)	Instance Storage (GB)	Network Bandwidth (Gbps)***	EBS Bandwidth (Gbps)
c6i.large	2	4	EBS-Only	Up to 12.5	Up to 10
c6i.xlarge	4	8	EBS-Only	Up to 12.5	Up to 10
c6i.2xlarge	8	16	EBS-Only	Up to 12.5	Up to 10
c6i.4xlarge	16	32	EBS-Only	Up to 12.5	Up to 10
c6i.8xlarge	32	64	EBS-Only	12.5	10

**\$0.34/hr**

**\$248.2/m**

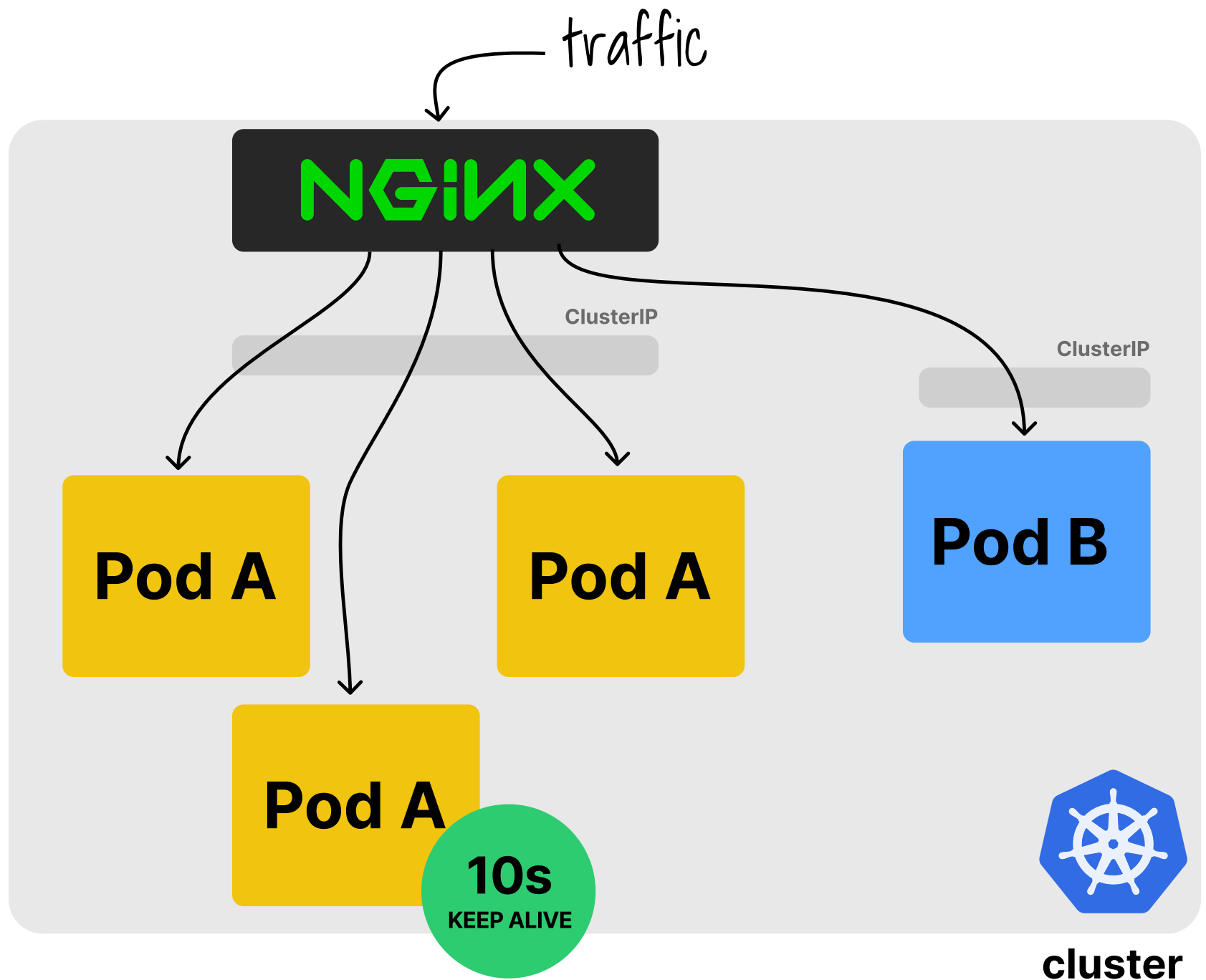


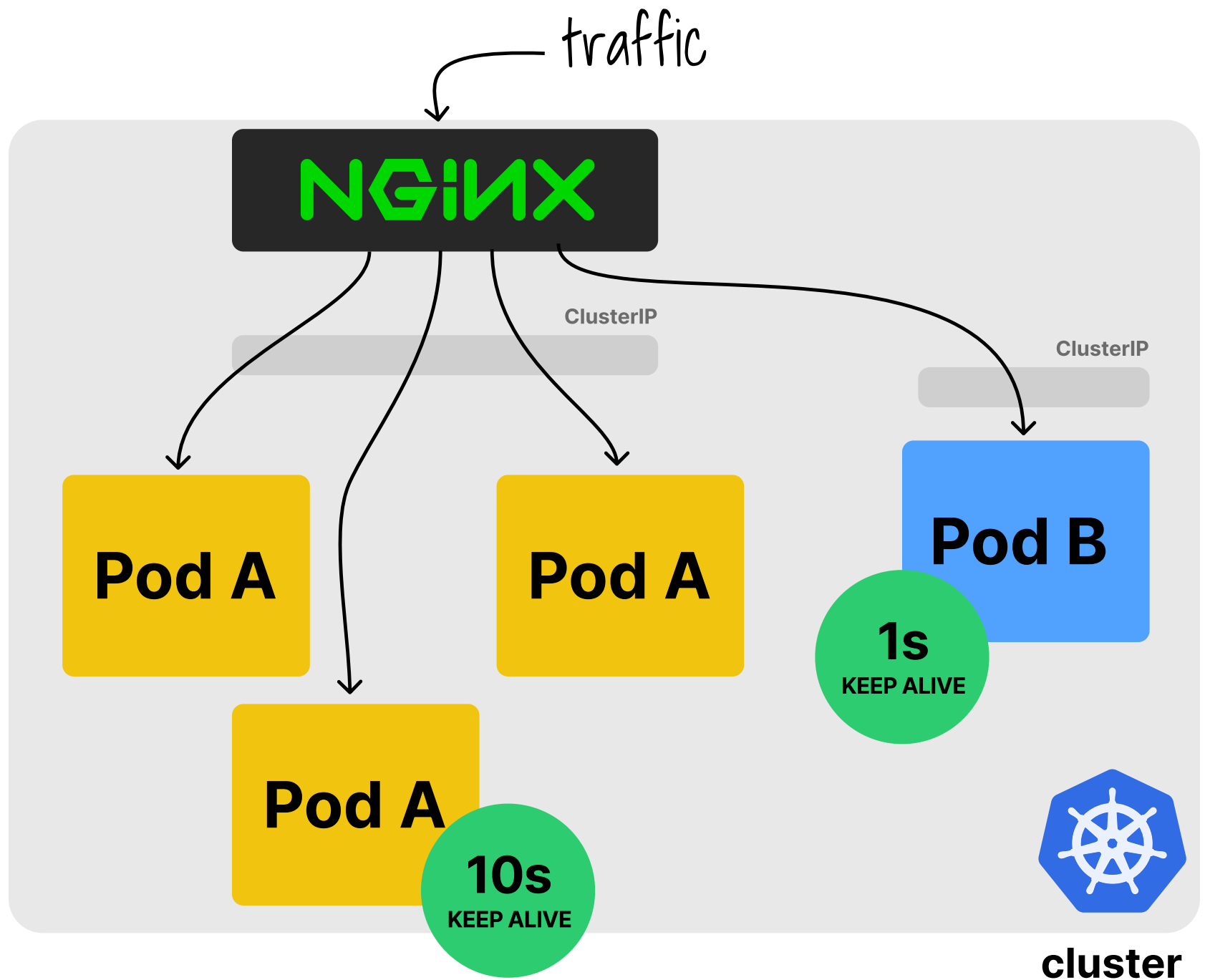
# 1 vs many: config

---











kind: ConfigMap

apiVersion: v1

metadata:

name: nginx-configuration

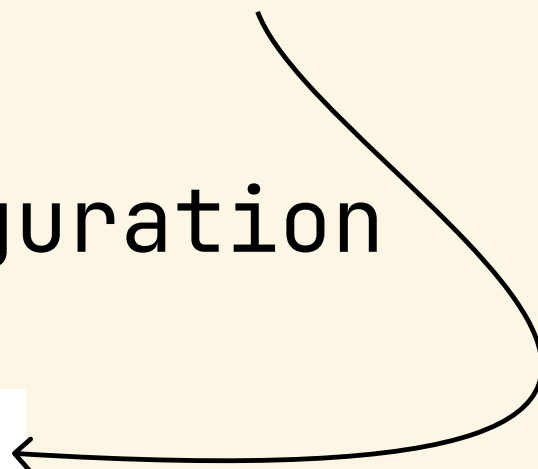
data:

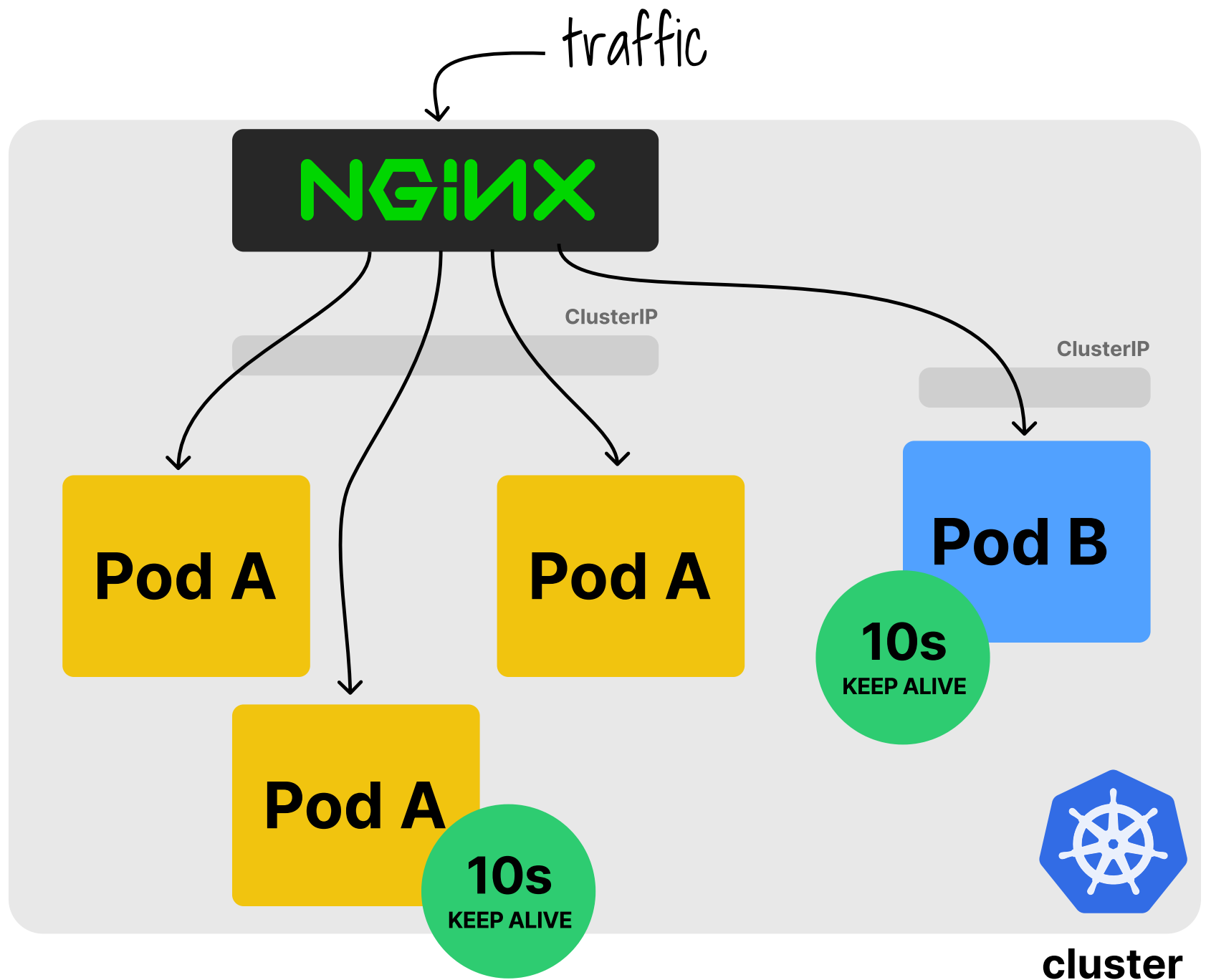
keep-alive: "10s"

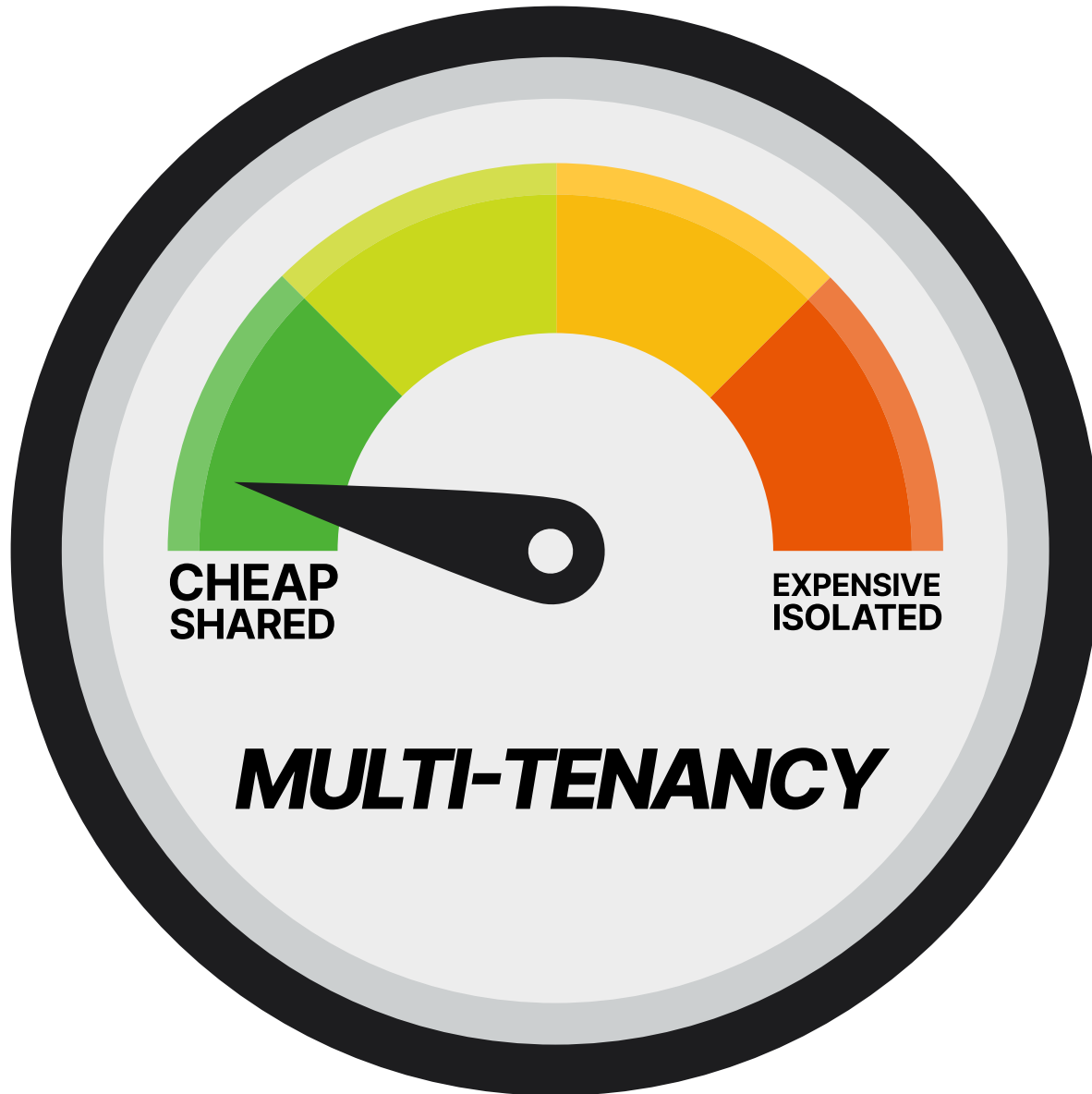
proxy-read-timeout: "10s"

client-max-body-size: "2m"

global setting







# Kubernetes platform tools

---



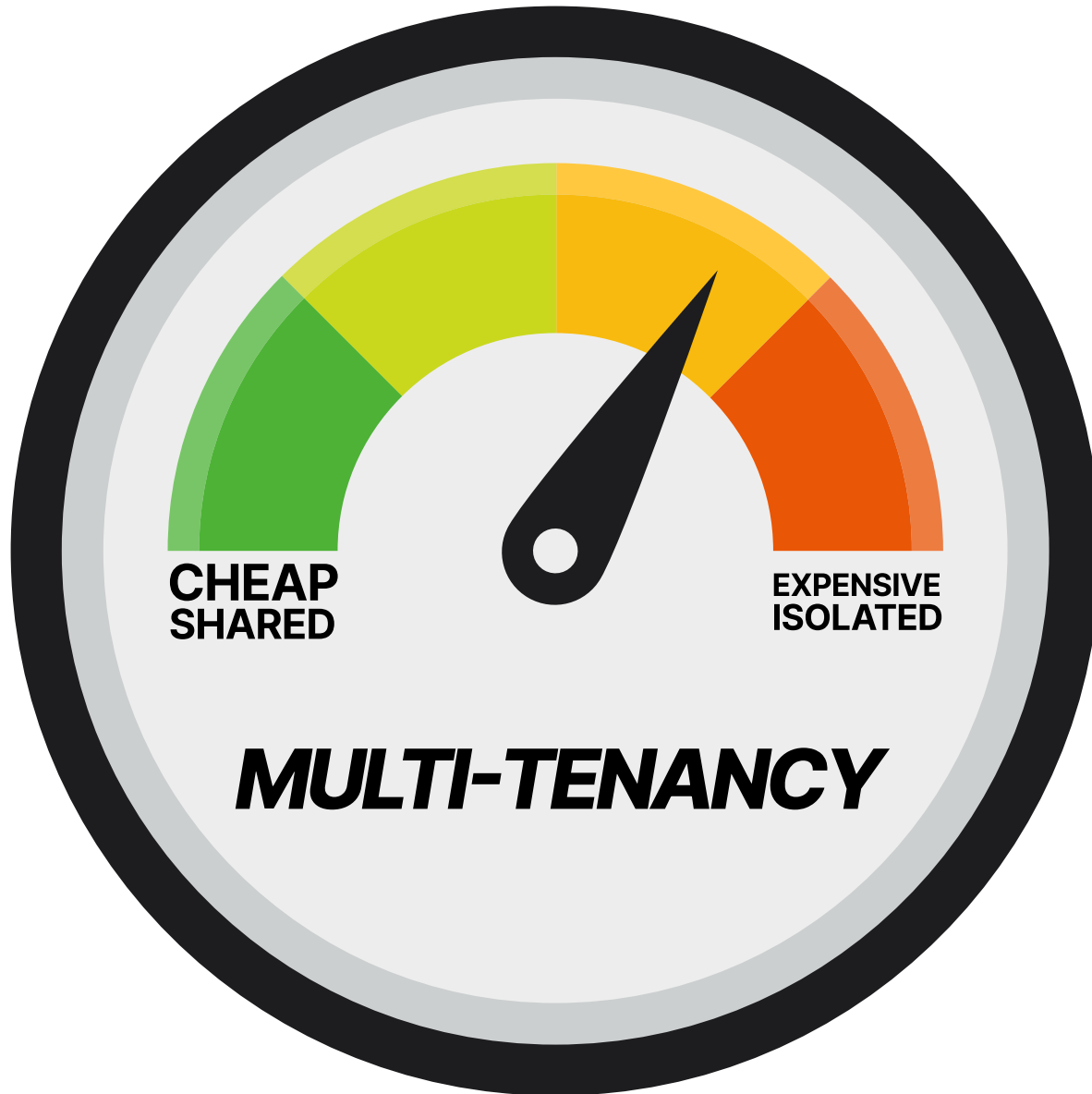


HOON JO







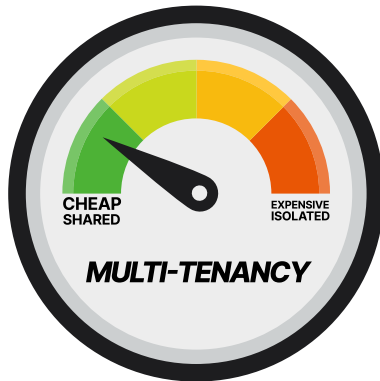
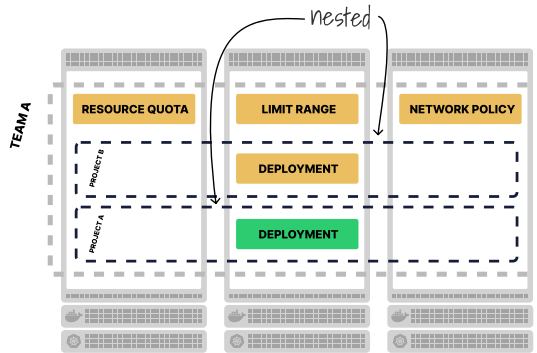


# Comparing multi-tenancy tools

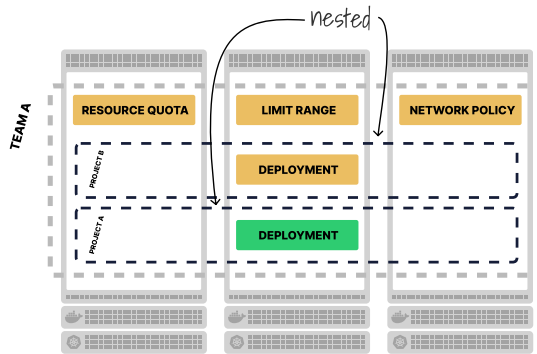
---



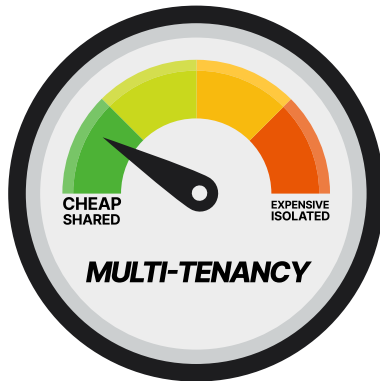
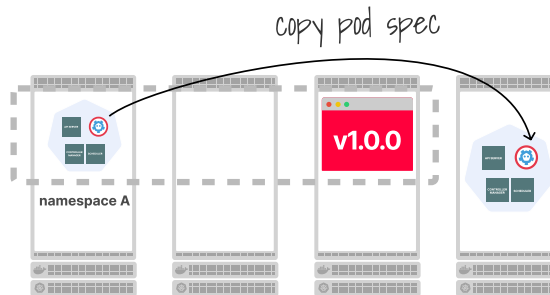
# Hierarchical Namespace Controller



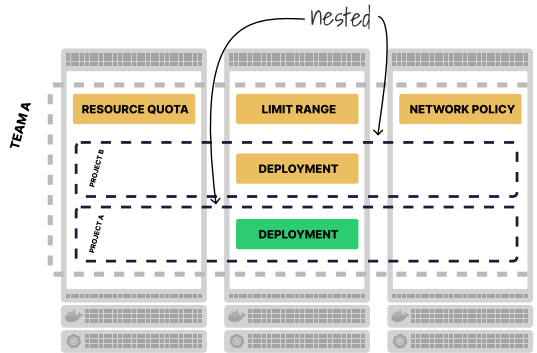
## Hierarchical Namespace Controller



## vCluster



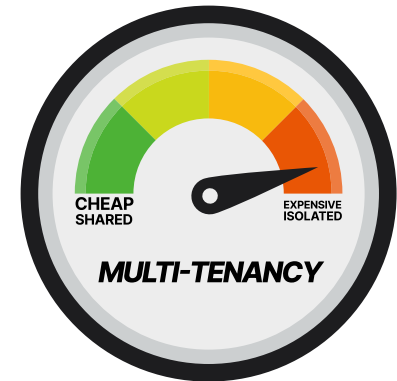
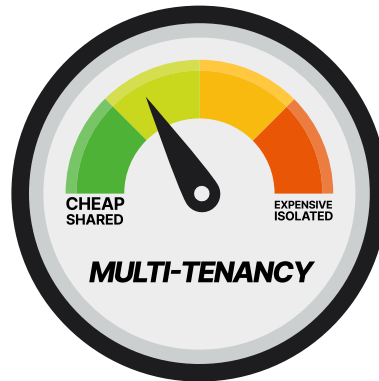
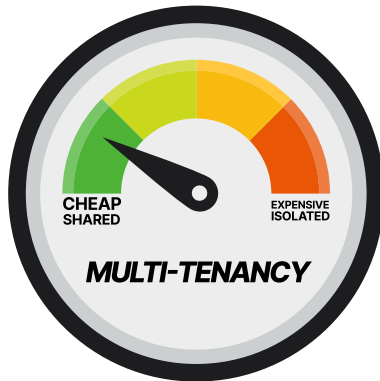
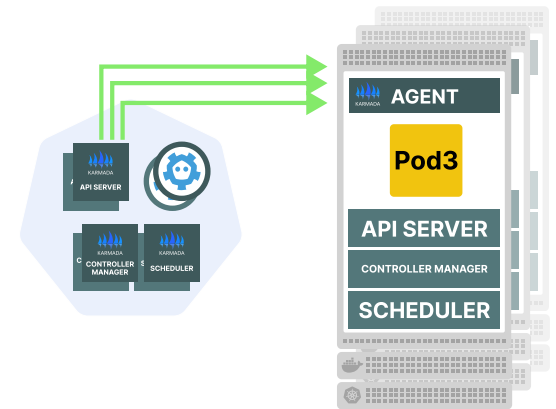
## Hierarchical Namespace Controller



## vCluster



## Karmada



# Hierarchical Namespace Controller

---



root namespace





root namespace

**RESOURCE QUOTA**

child 1

**ROLE**

child 2



root namespace

**RESOURCE QUOTA**

**ROLE**

child 1

child 2



child 3



child 4



child 5



# Demo

---



# **Hierarchical Namespace Controller**

---

**“Nested” namespaces**



# **Hierarchical Namespace Controller**

---

**“Nested” namespaces**  
**Single controller**



# **Hierarchical Namespace Controller**

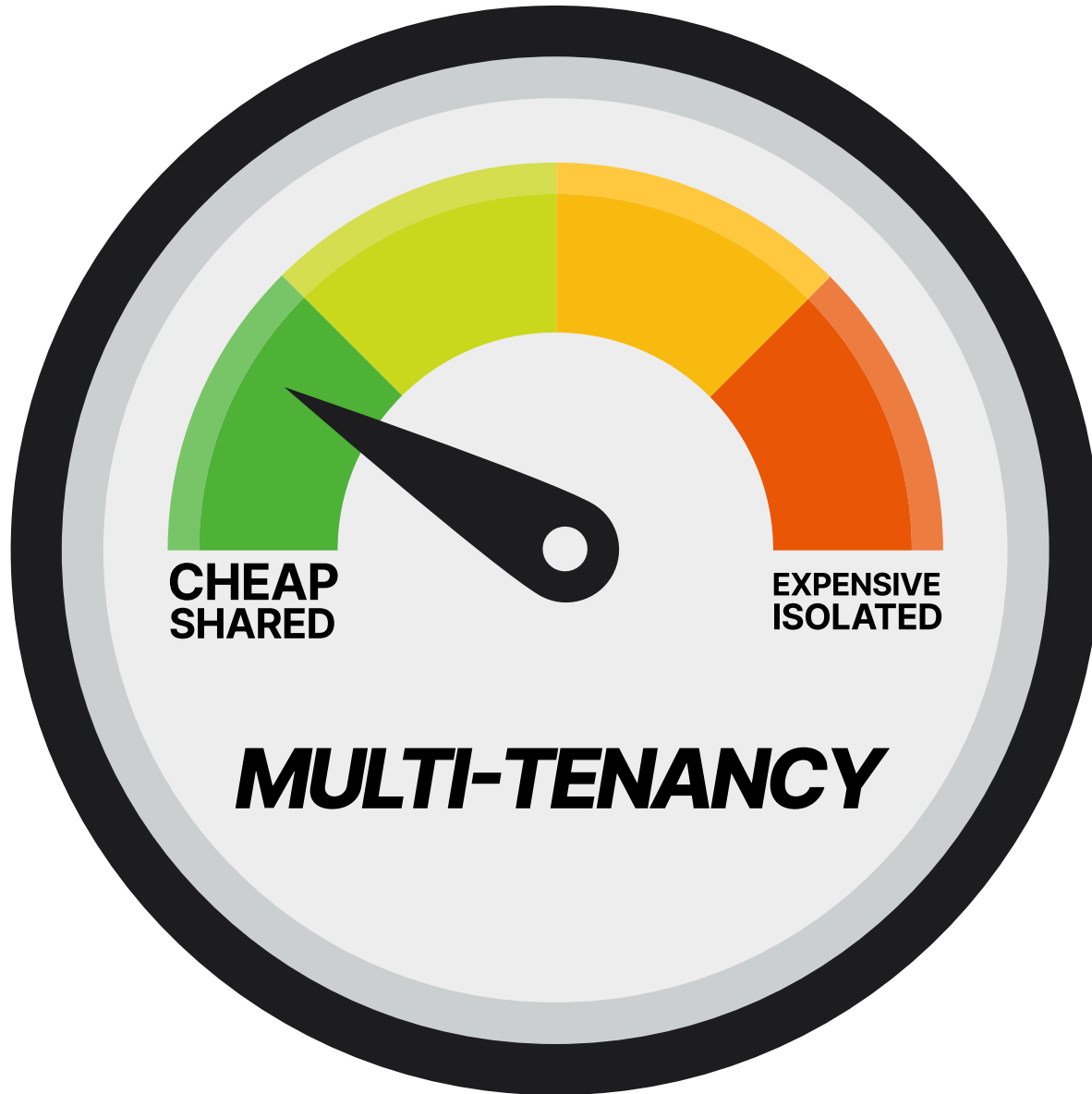
---

**“Nested” namespaces**

**Single controller**

**Regular namespaces**





## COSTS FOR 50 TENANTS

~\$0









# HNC & Roles

---







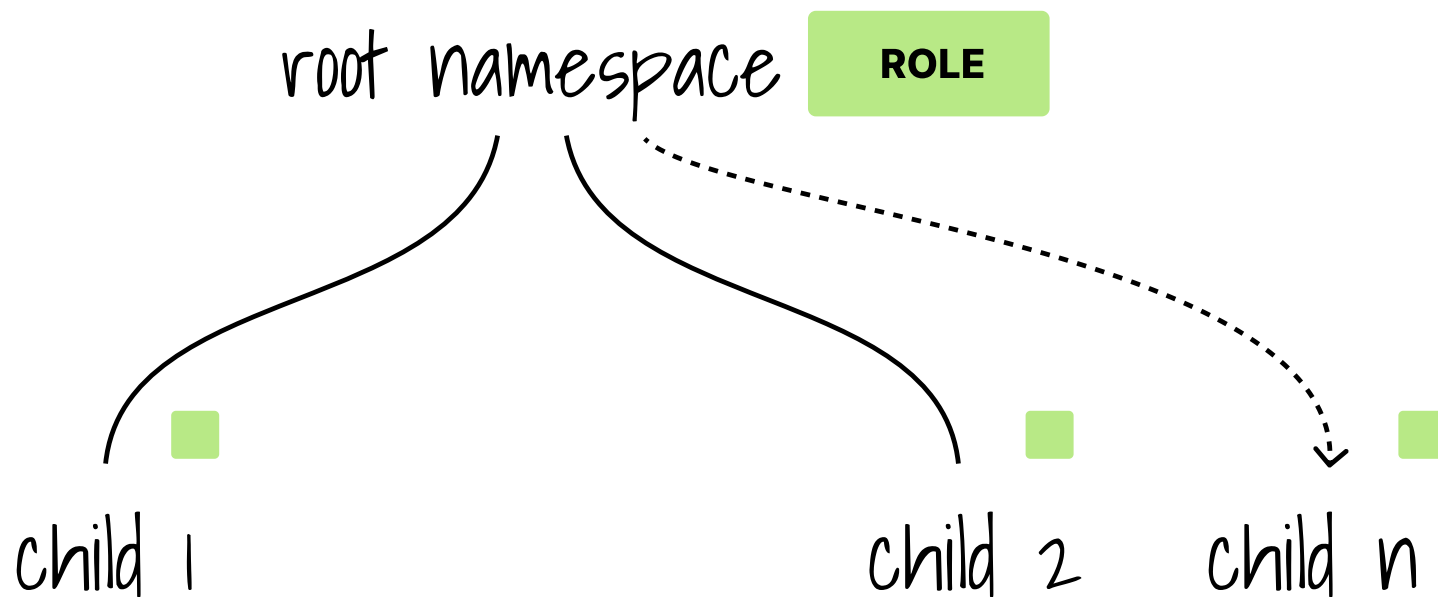
UID	ROLE	PODS		PVs	
		read	write	read	write
1	teamA				

root namespace

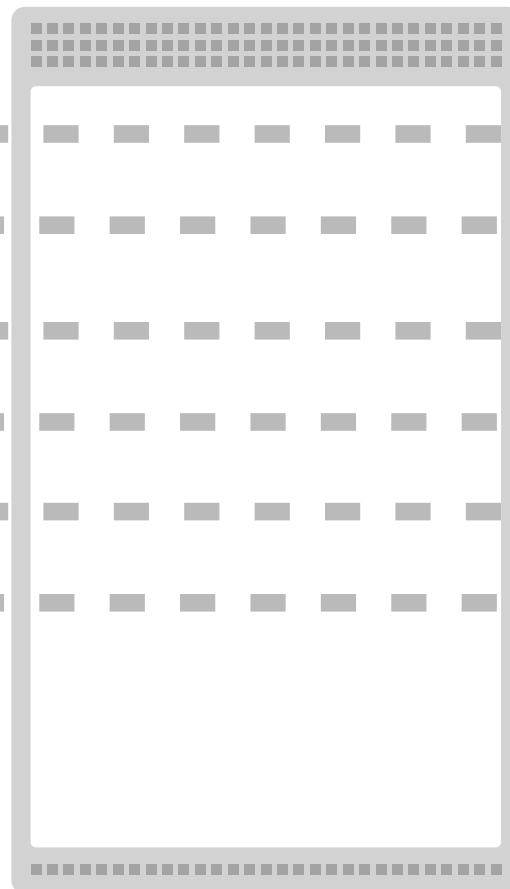
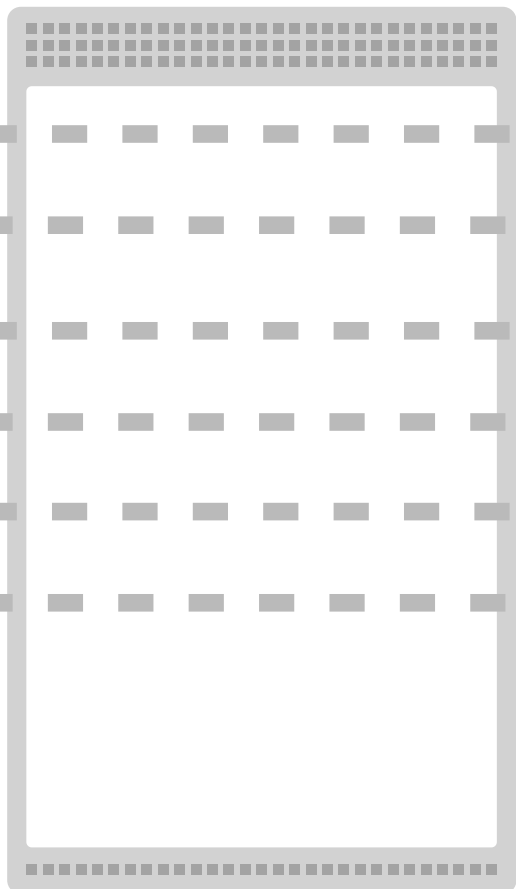
ROLE



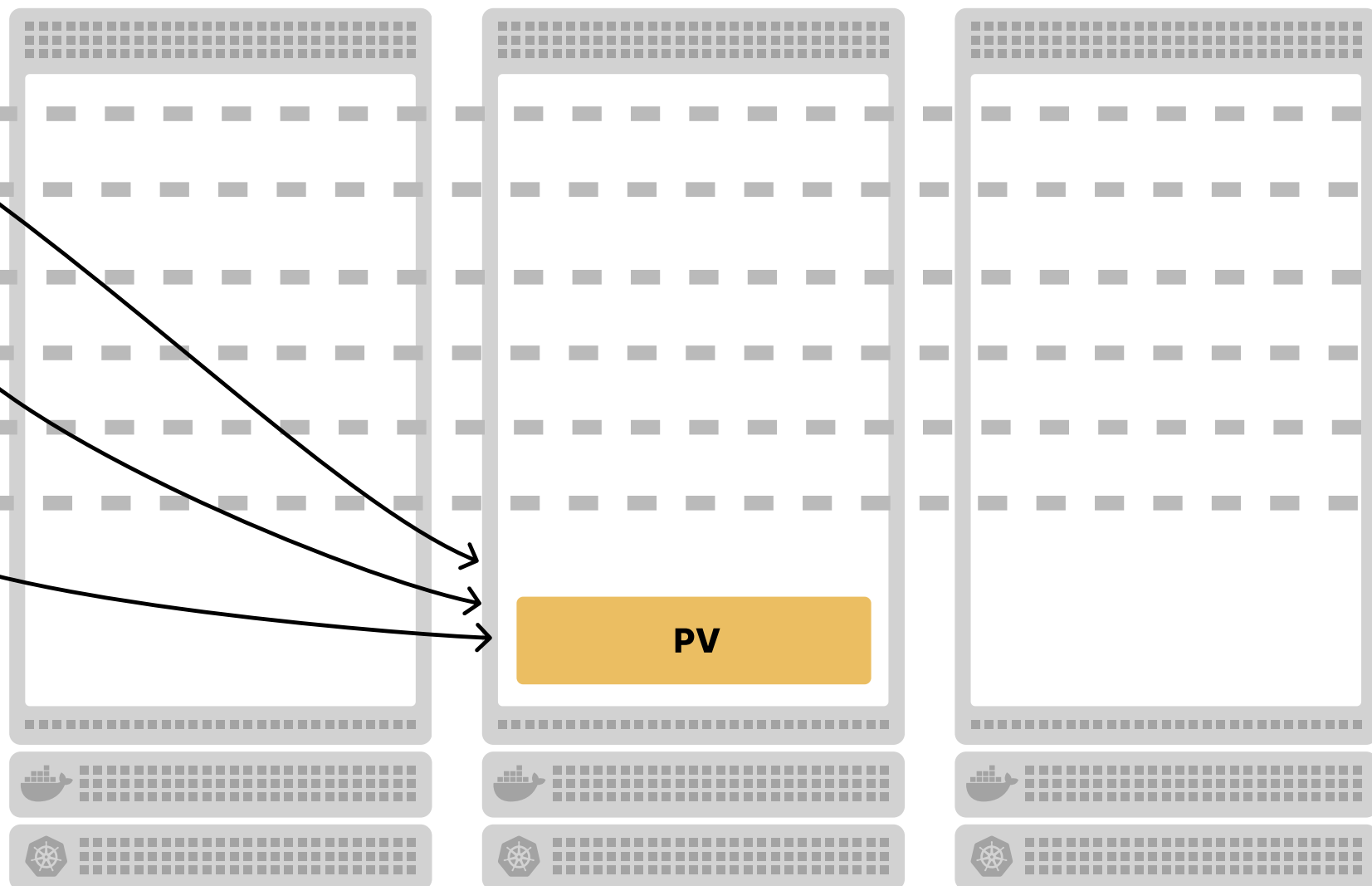
UID	ROLE	PODS		PVs	
		read	write	read	write
1	teamA				



TEAM C  
TEAM B  
TEAM A



TEAM A  
TEAM B  
TEAM C

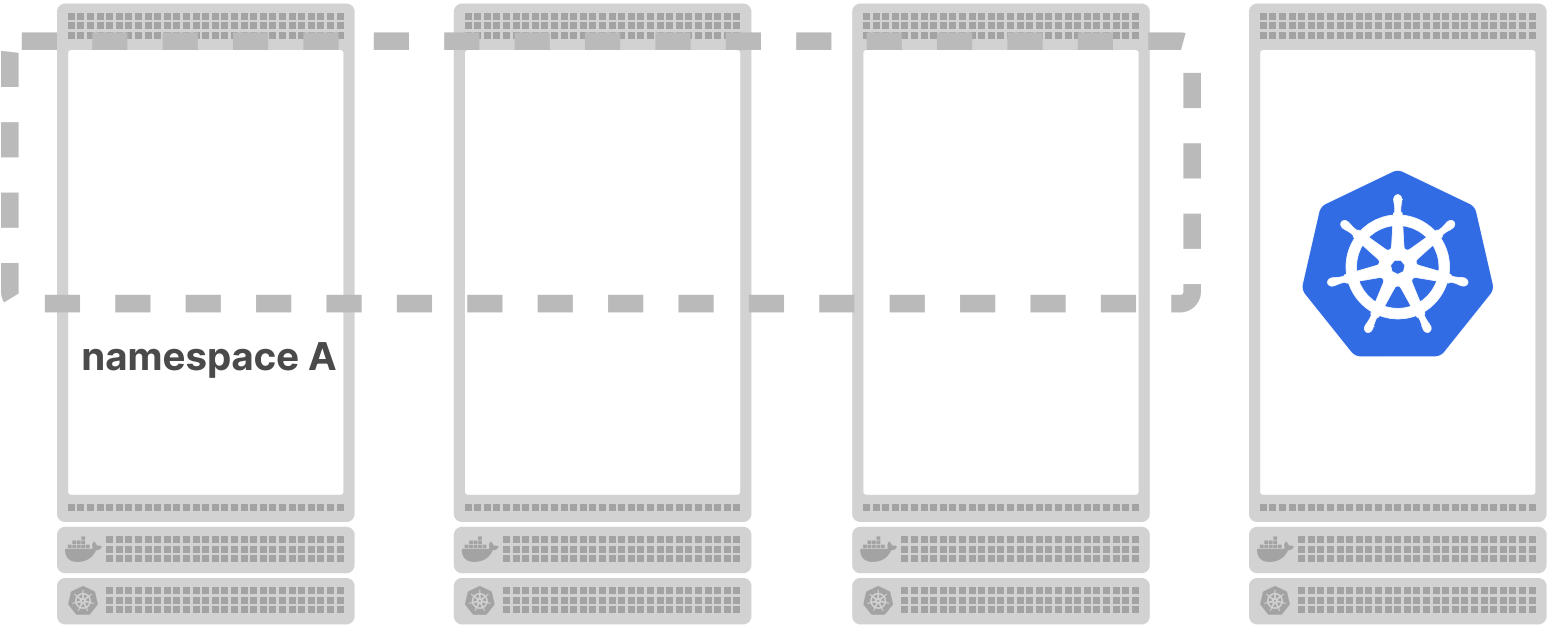


# Isolating control planes

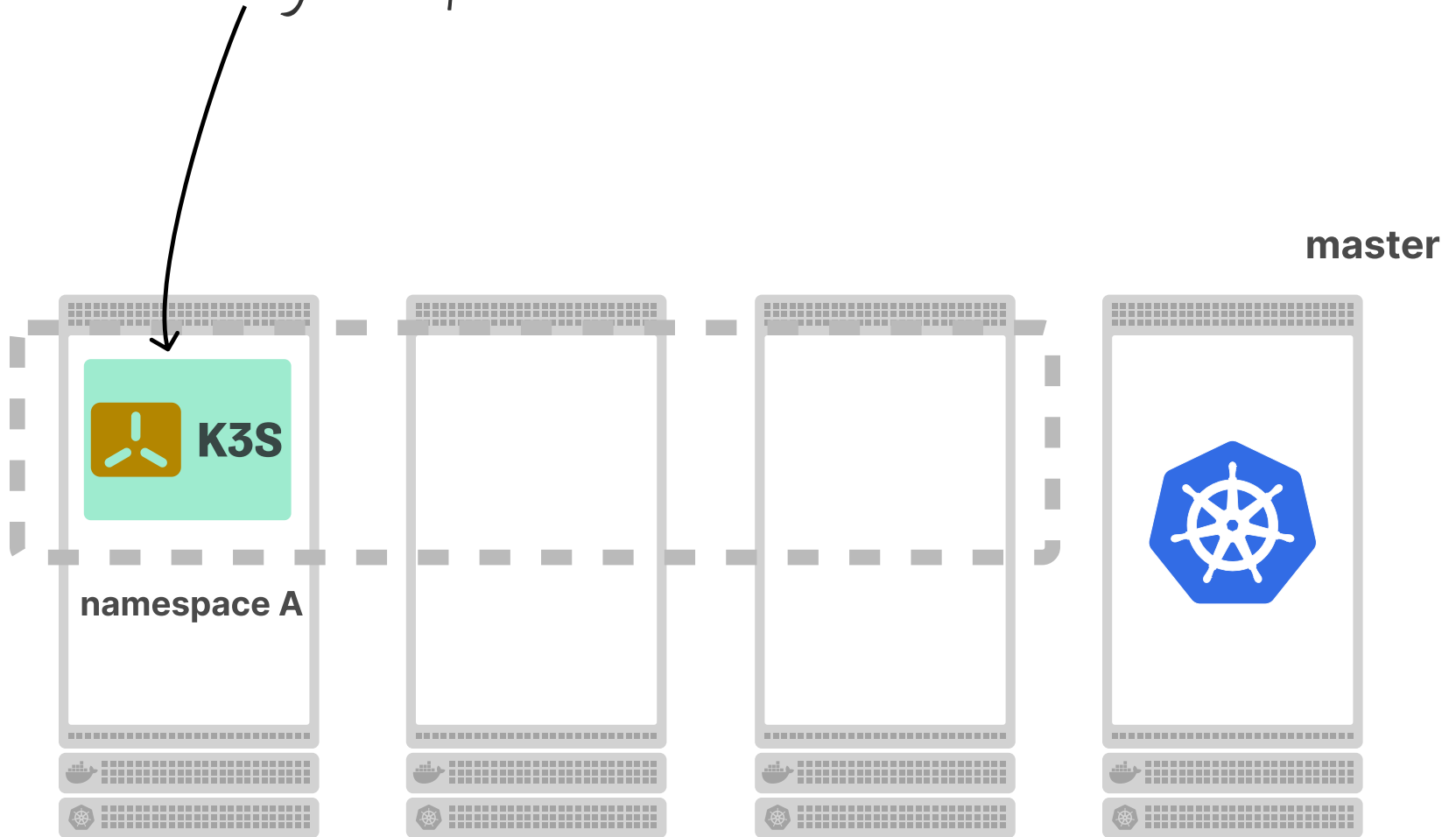
---



master



Just a regular pod

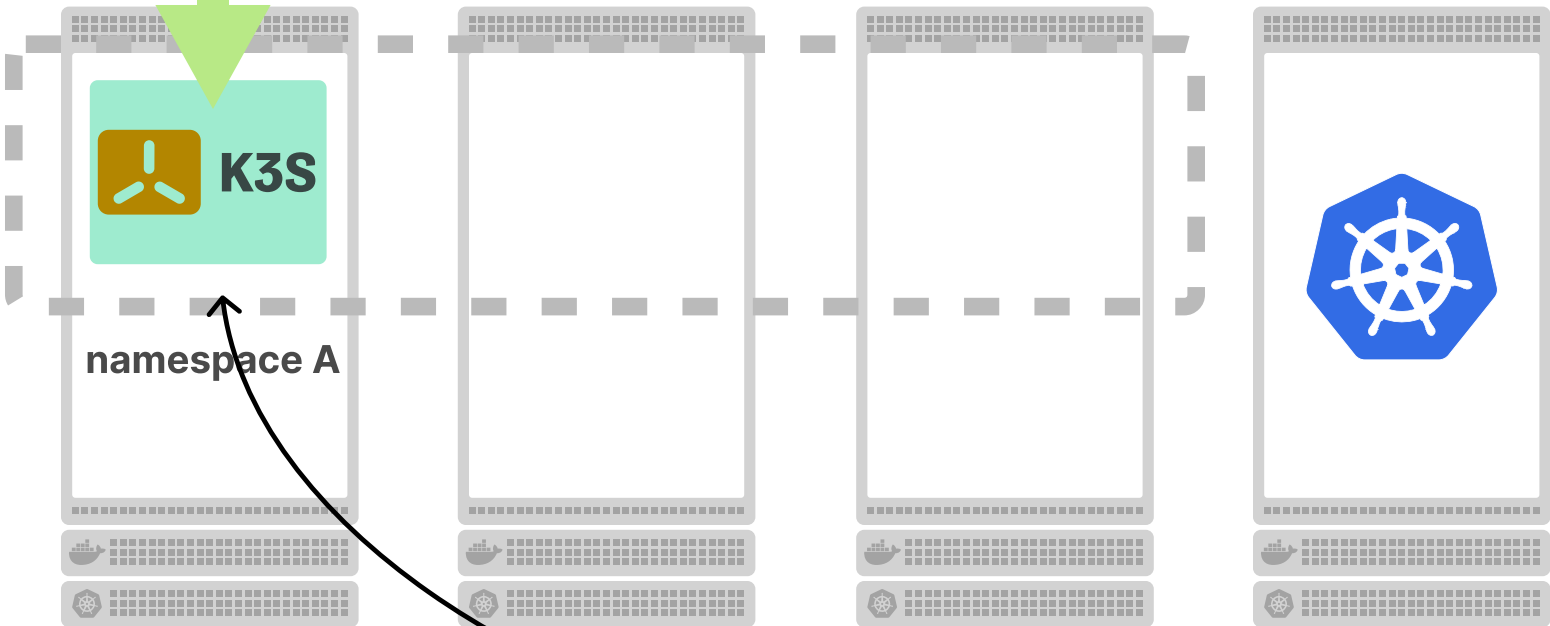






kubectl apply

master



it has no nodes!

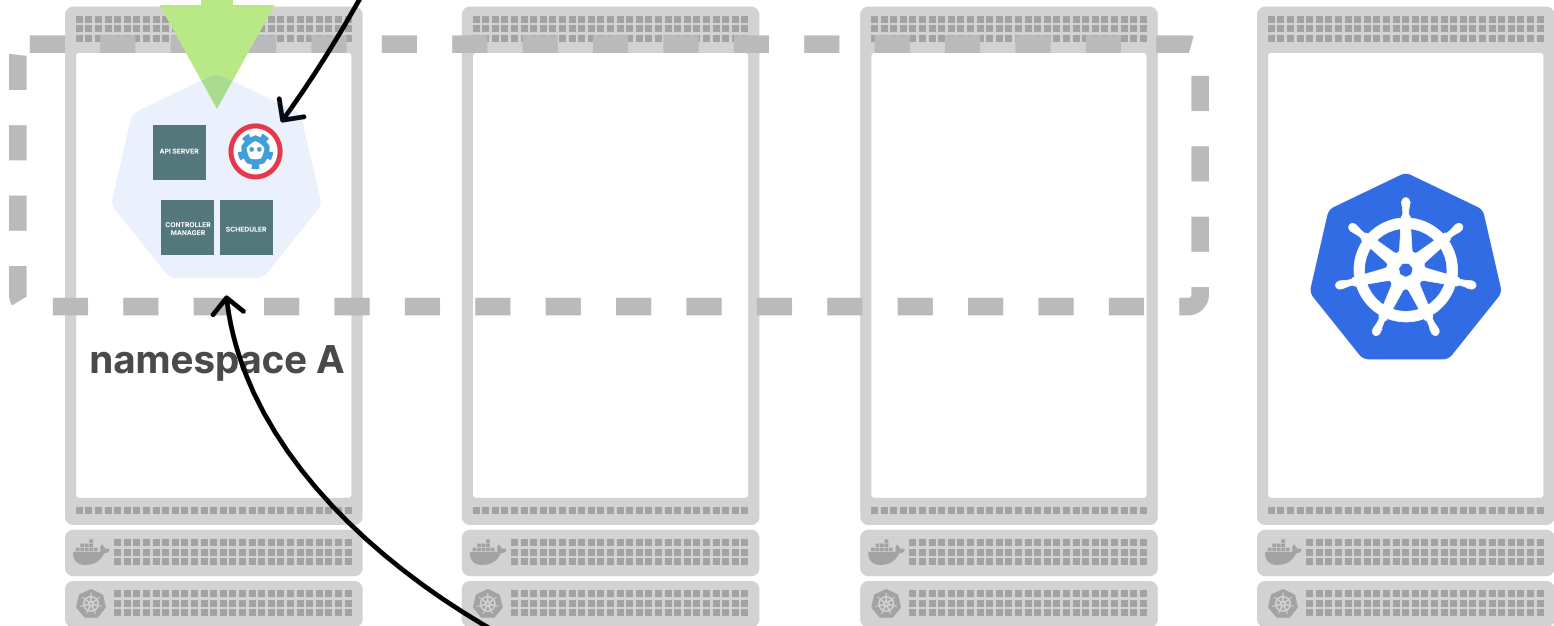




kubectl apply

the pods are in (wrong) db!

master



it has no nodes!

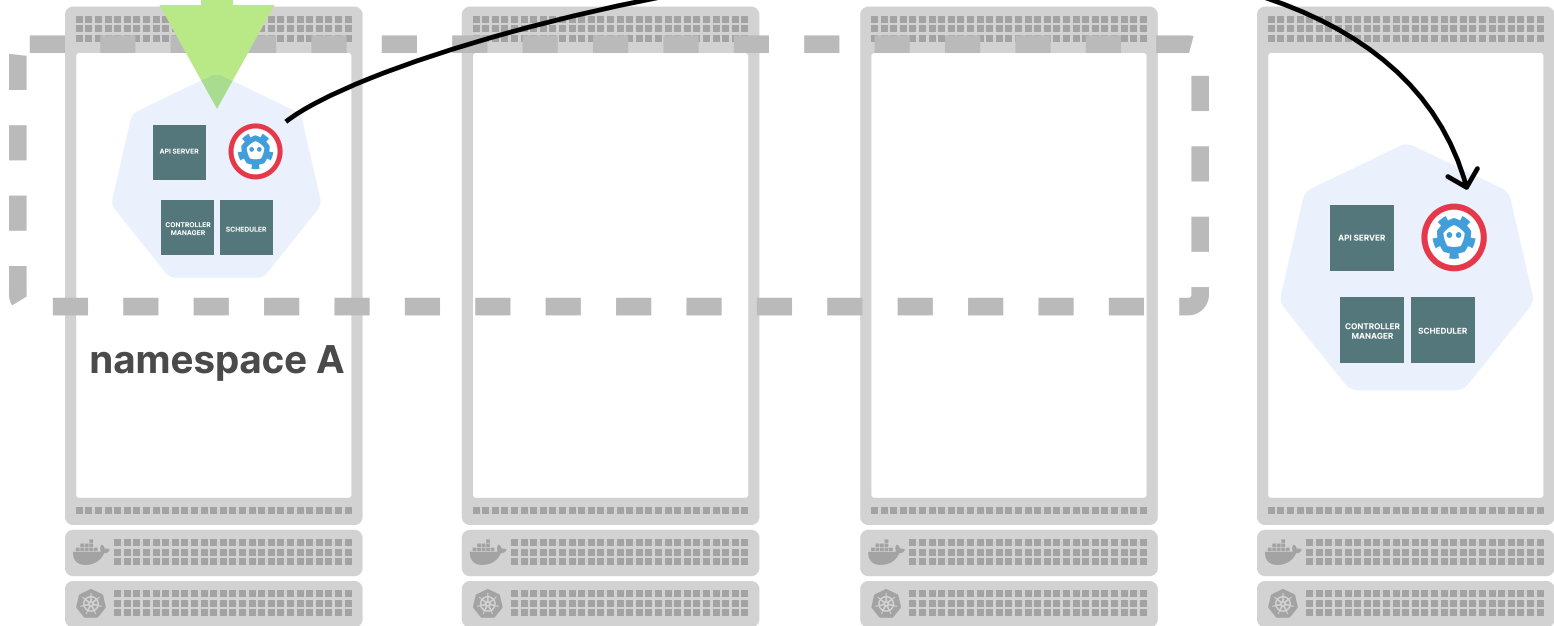




kubectl apply

copy pod spec

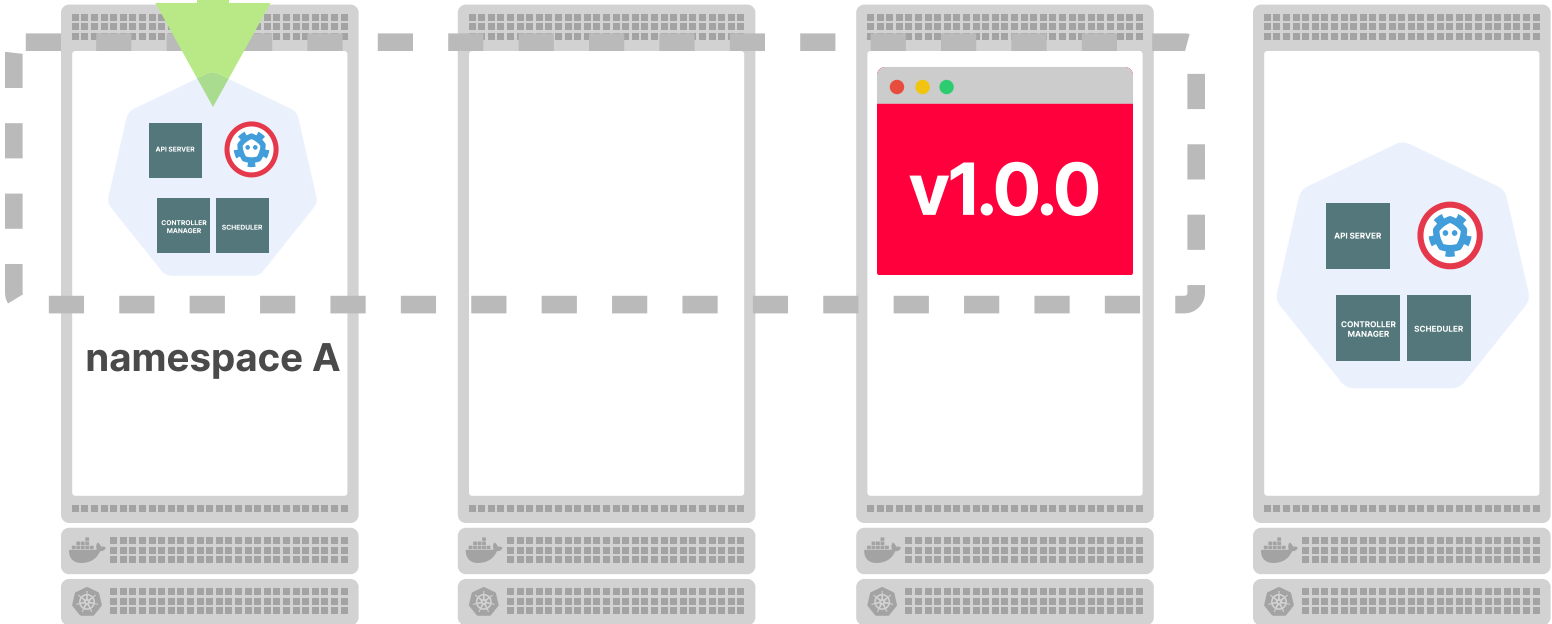
master





kubectl apply

master

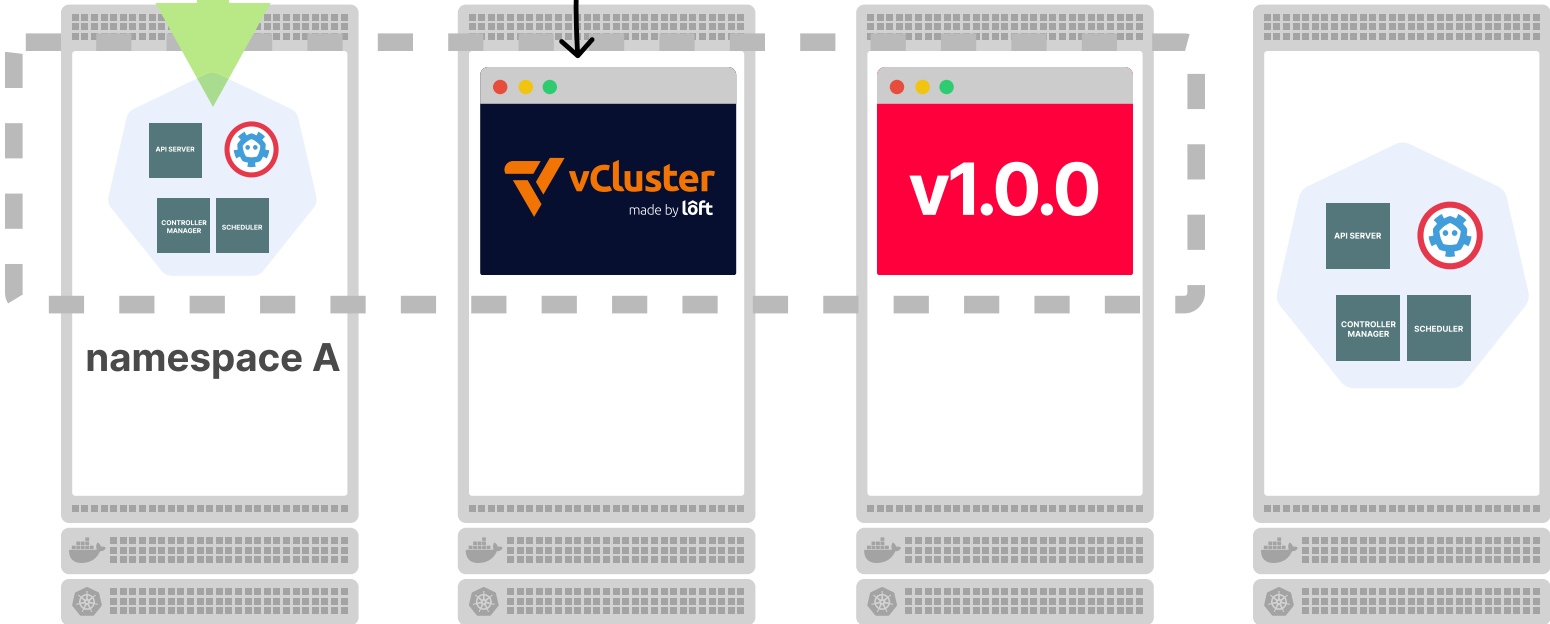




kubectl apply

syncer

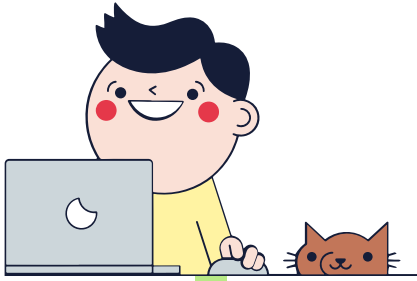
master



# **vCluster and global resources**

---



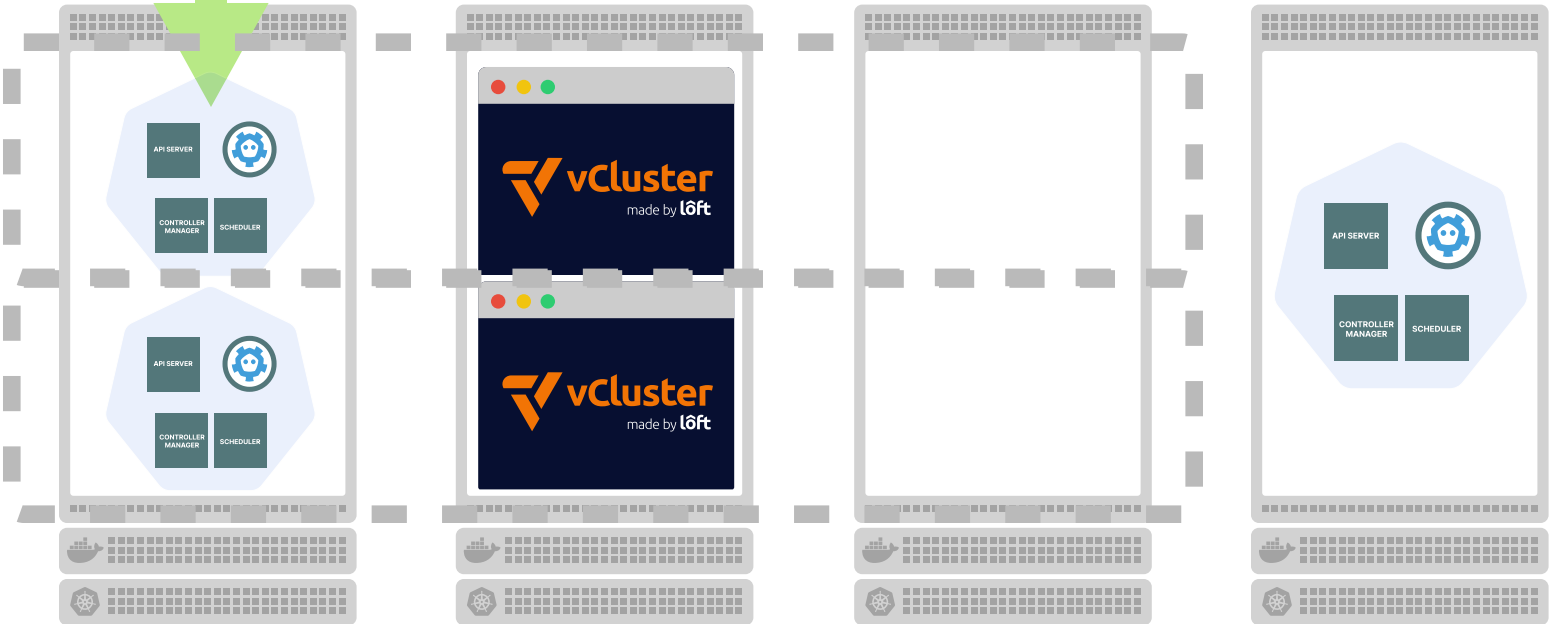


```
kubectl apply -f my-pv.yaml
```

master

TEAM A  
NAMESPACE

TEAM B  
NAMESPACE





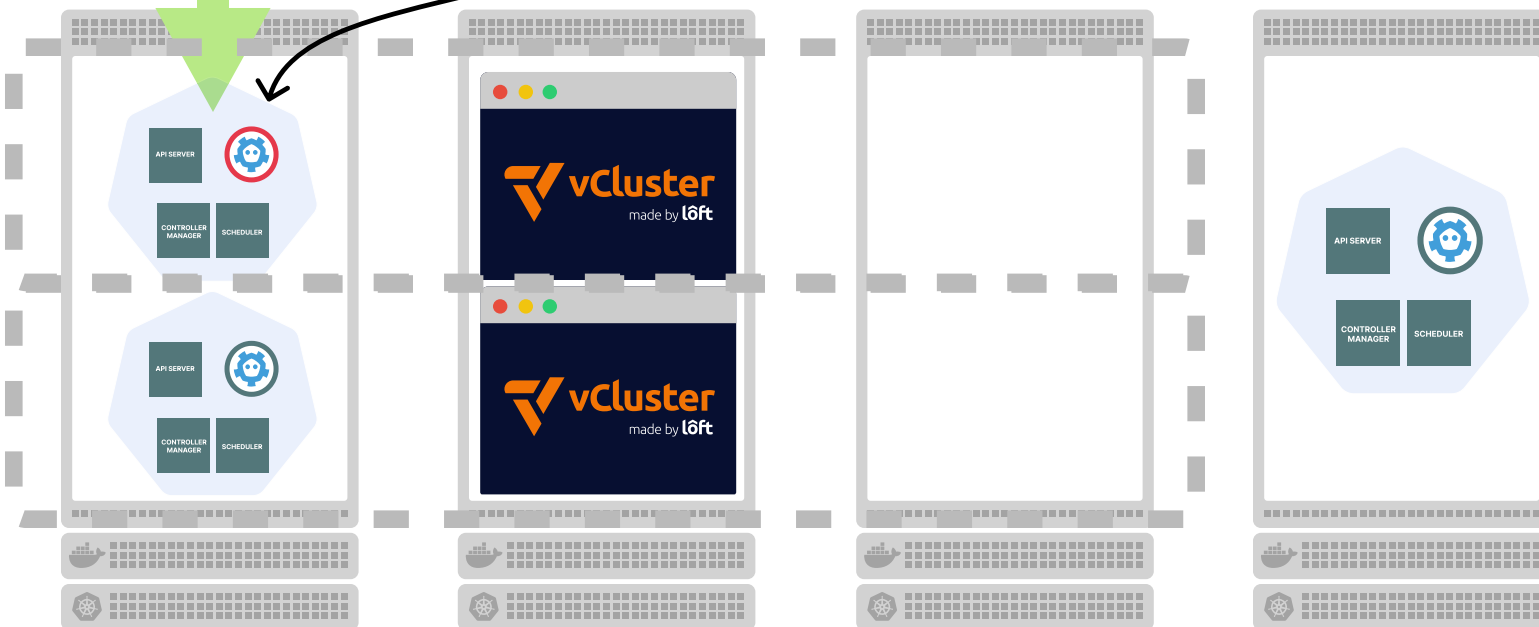
```
kubectl apply -f my-pv.yaml
```

The PV is stored in the  
tenant control plane

TEAM A  
NAMESPACE

TEAM B  
NAMESPACE

master







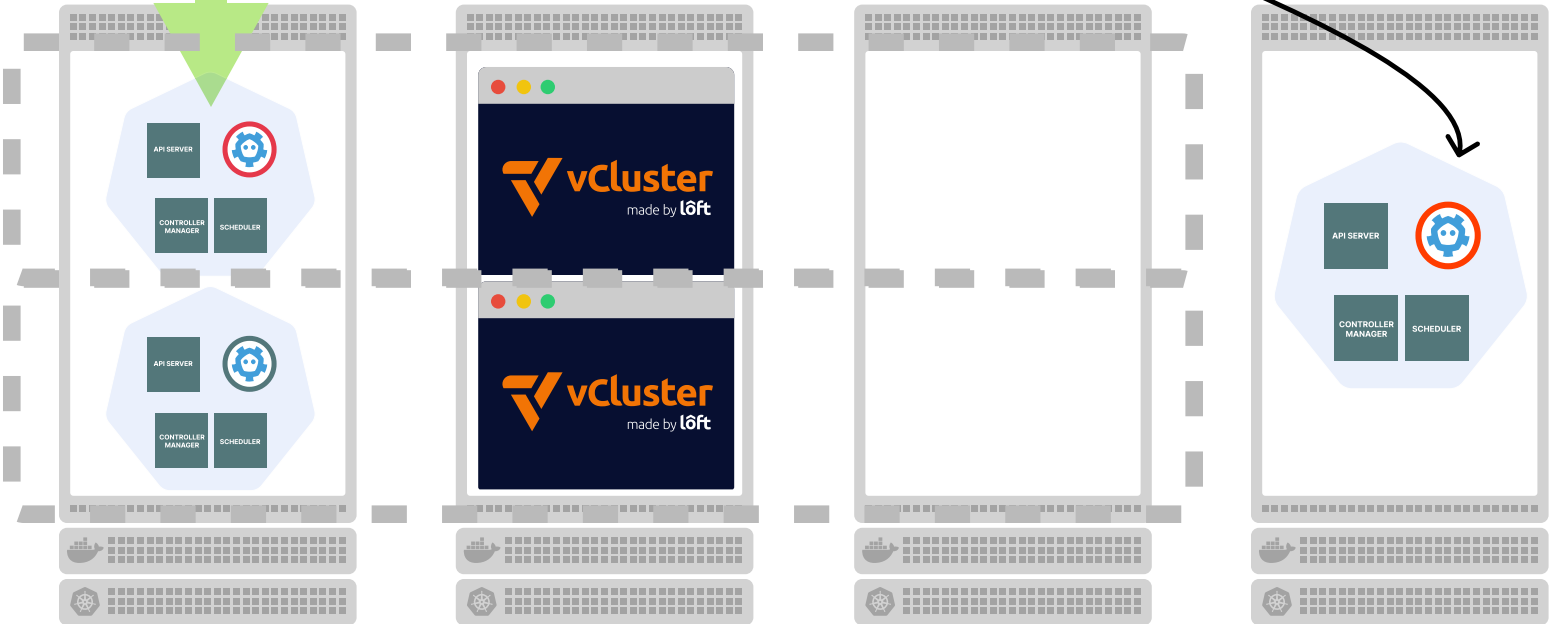
```
kubectl apply -f my-pv.yaml
```

the PV is synced

TEAM A  
NAMESPACE

TEAM B  
NAMESPACE

master





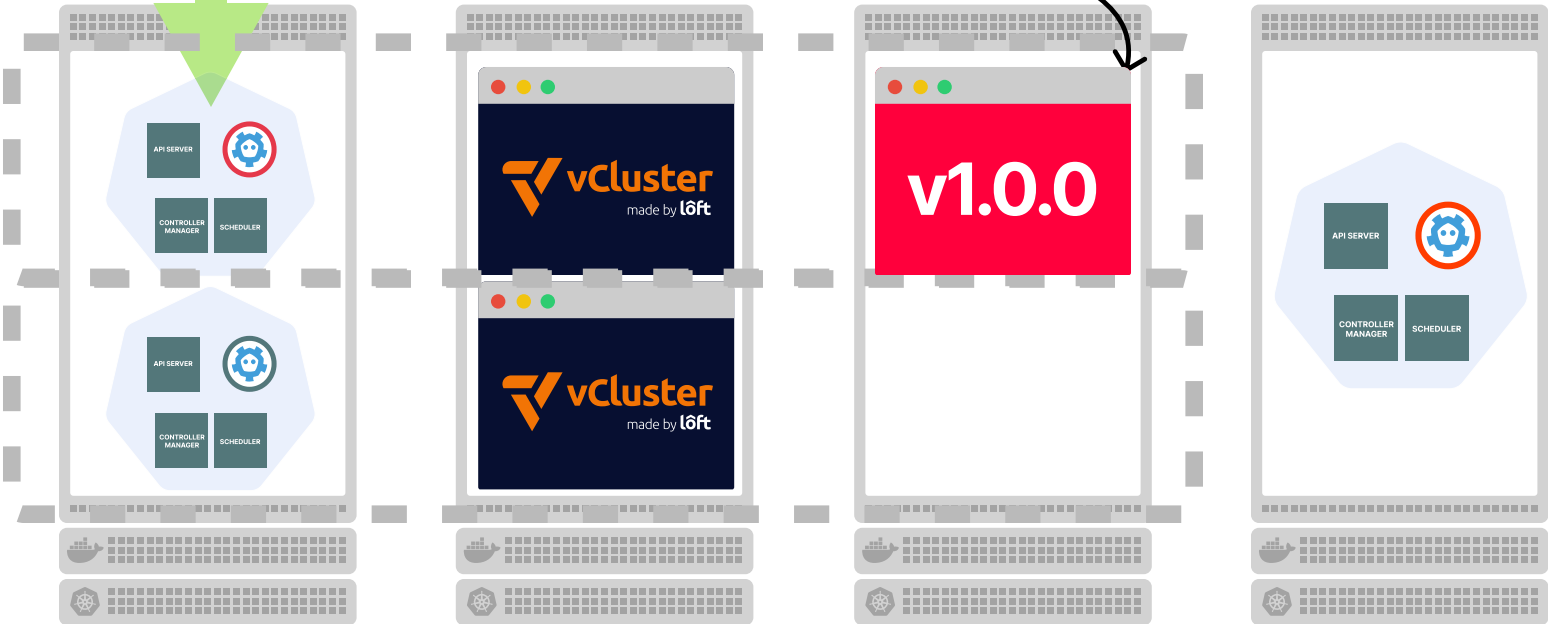
```
kubectl apply -f my-pv.yaml
```

the pod can consume it  
with a PVC

master

TEAM A  
NAMESPACE

TEAM B  
NAMESPACE



# Demo

---



**vCluster**

---

**“Nested” control planes**

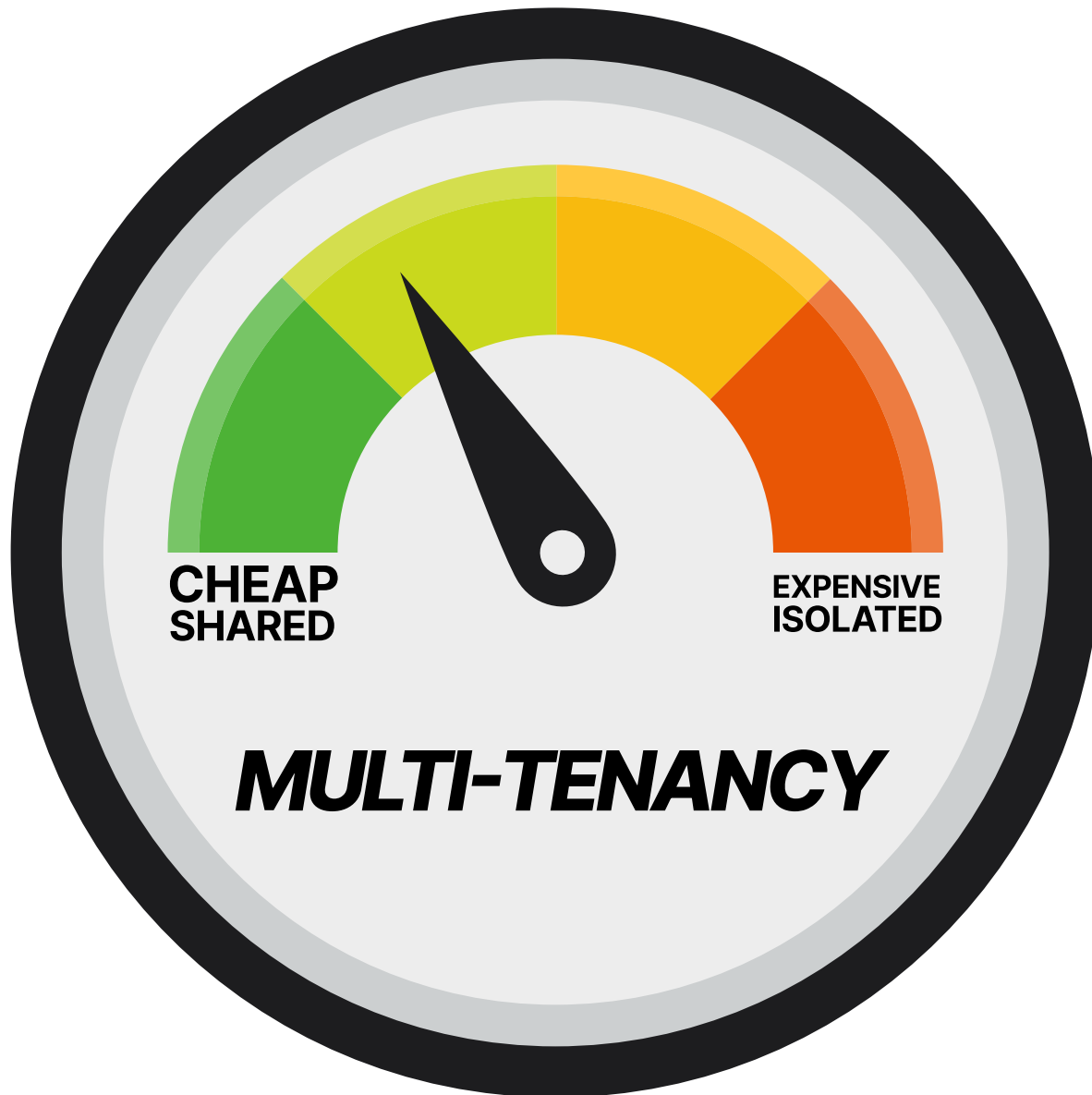


# **“Nested” control planes Admin vs tenants**



**“Nested” control planes**  
**Admin vs tenants**  
**Shared host cluster**





## COSTS FOR 50 TENANTS

**+ 17 nodes x \$12**





## COSTS FOR 50 TENANTS

**+ 17 nodes x \$12**

**+ 50 PVs x \$1**



## COSTS FOR 50 TENANTS

**+ 17 nodes x \$12**

**+ 50 PVs x \$1**



**= \$254 / month**

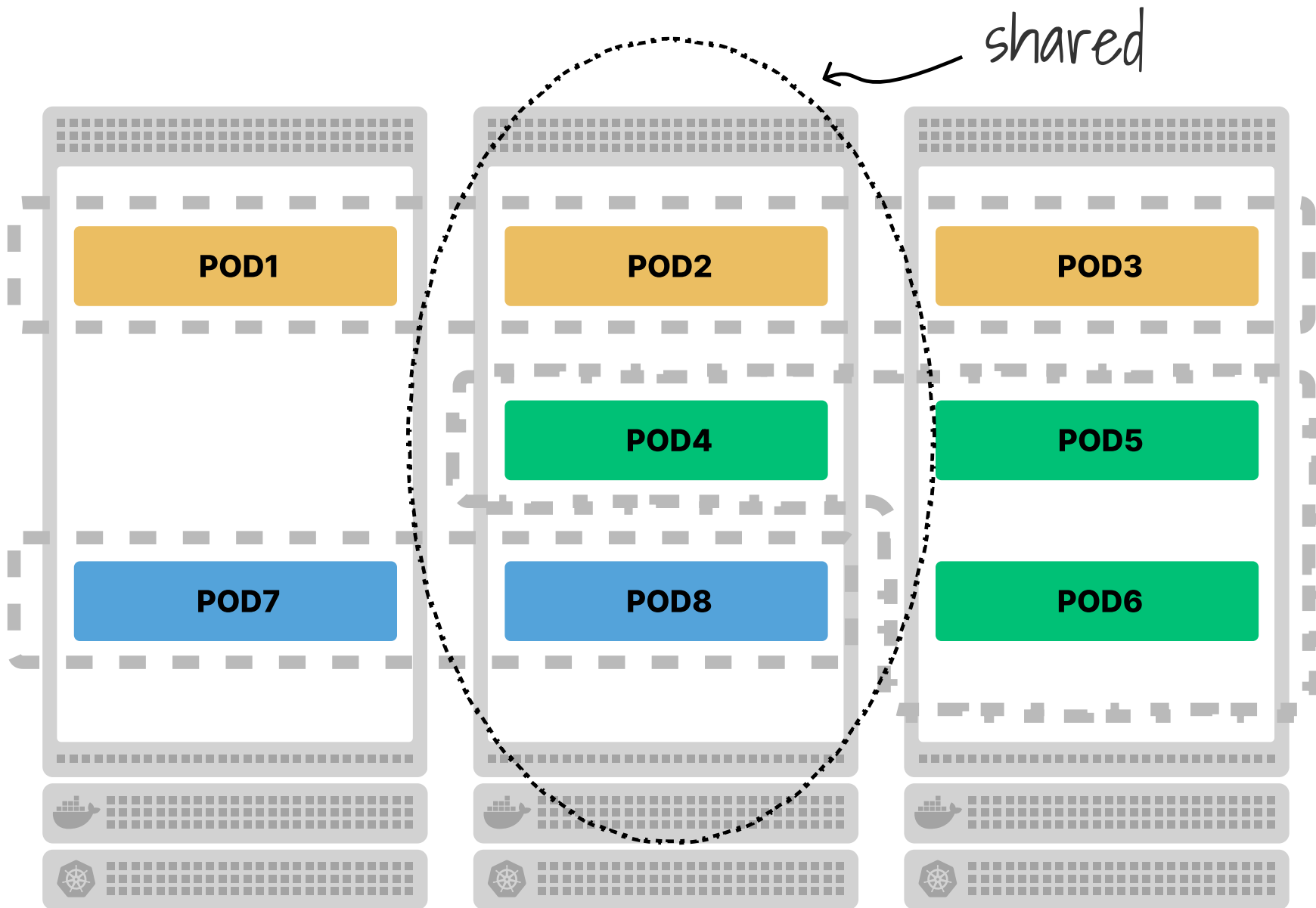
**~\$5 / month / tenant**

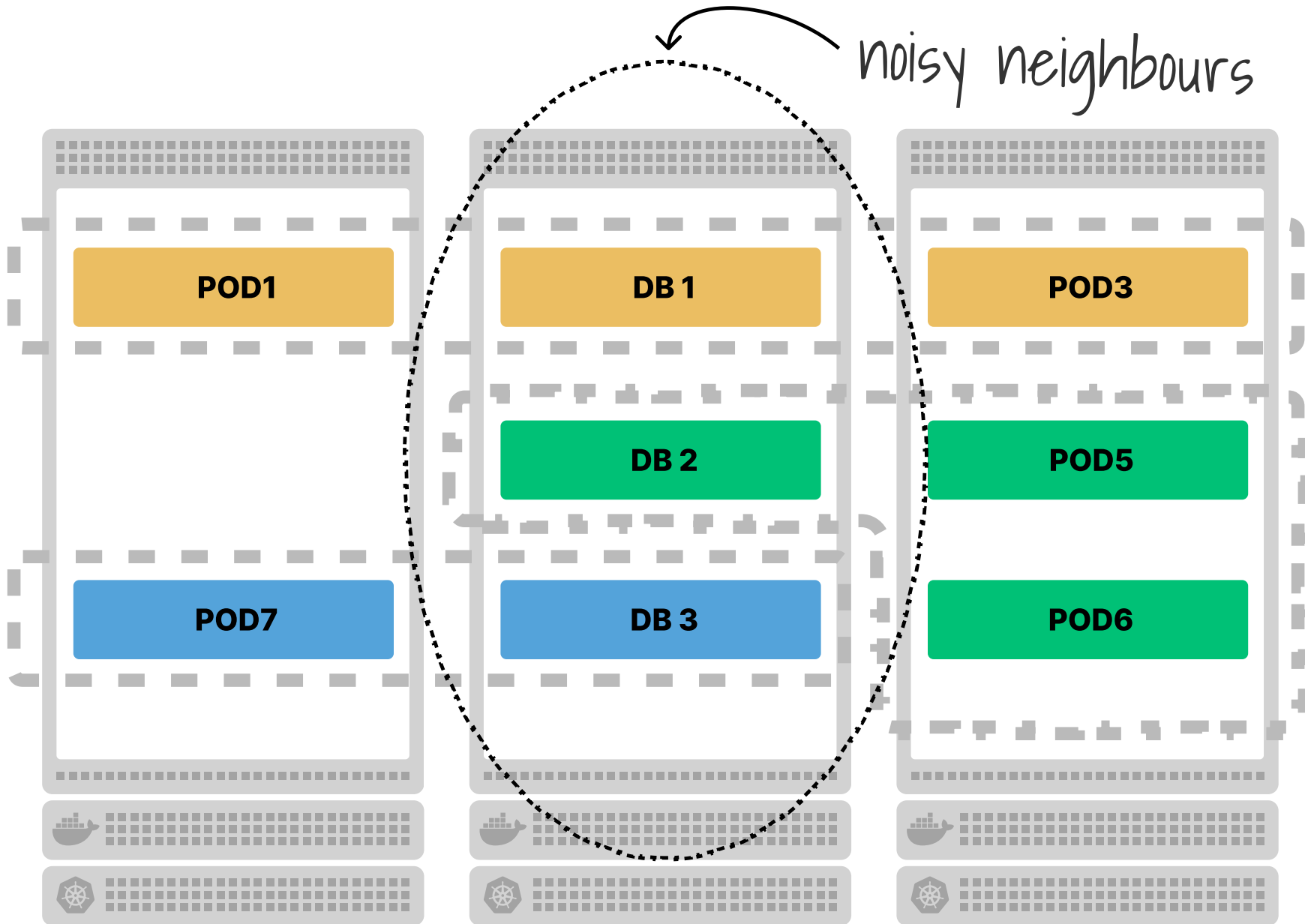


# **vCluster and shared nodes**

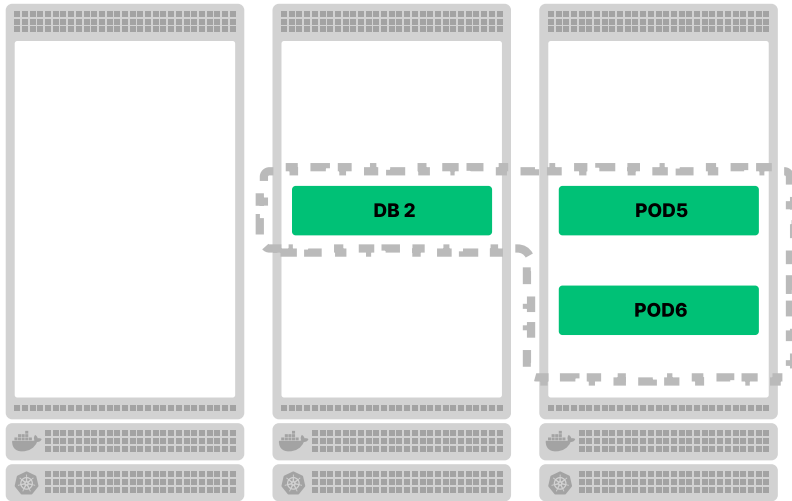
---



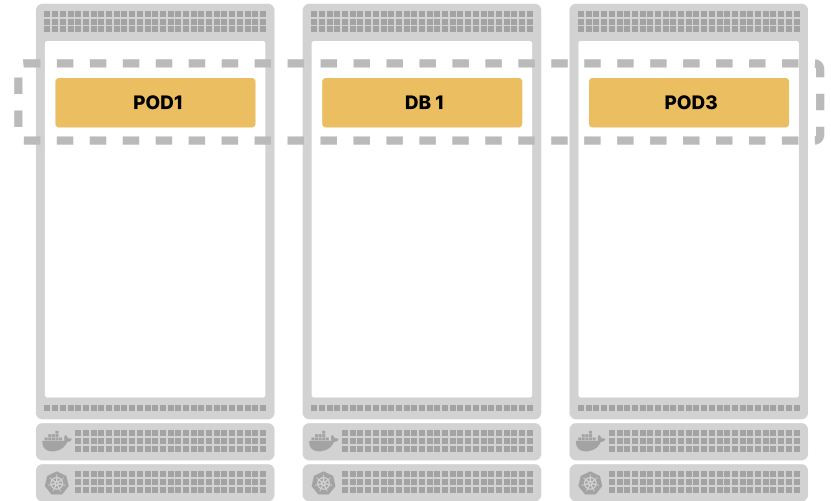




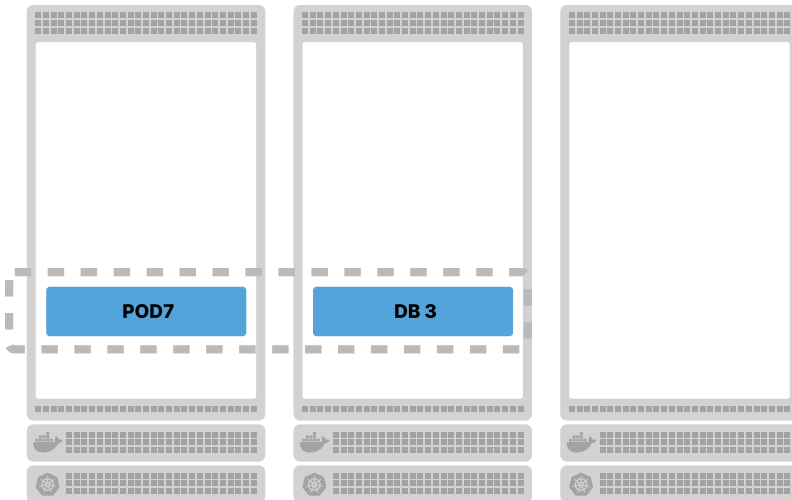
## POOL 1



## POOL 2



## POOL 3



## CONTROL PLANE



--node-selector  
--enforce-node-selector

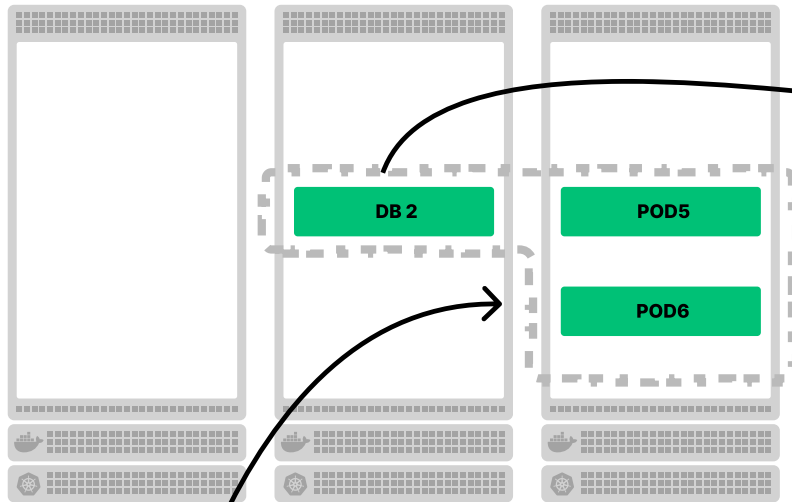


# **vCluster and shared network**

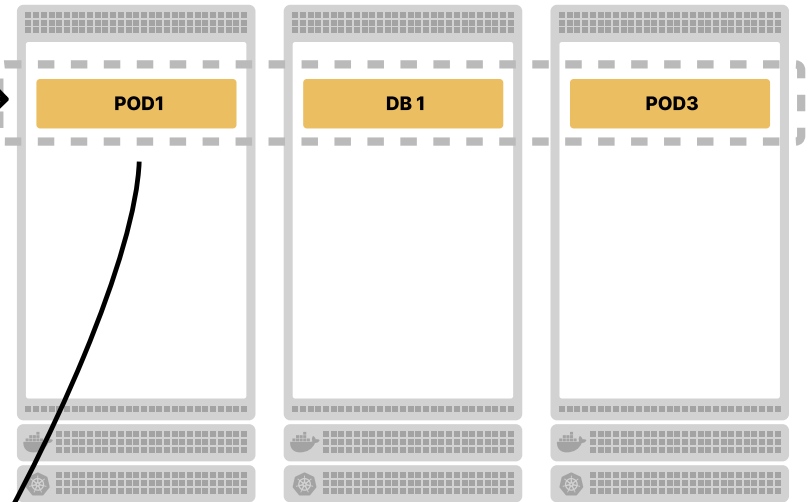
---



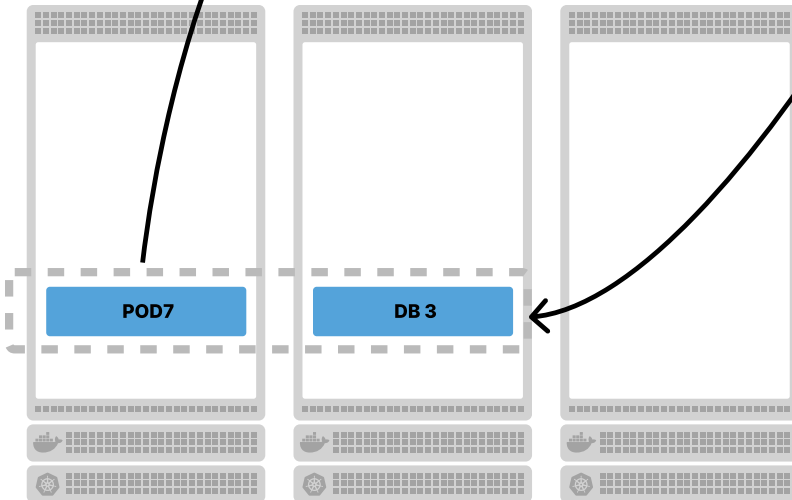
## POOL 1



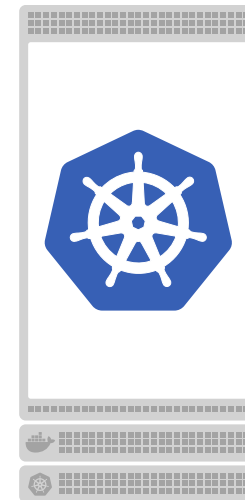
## POOL 2



## POOL 3

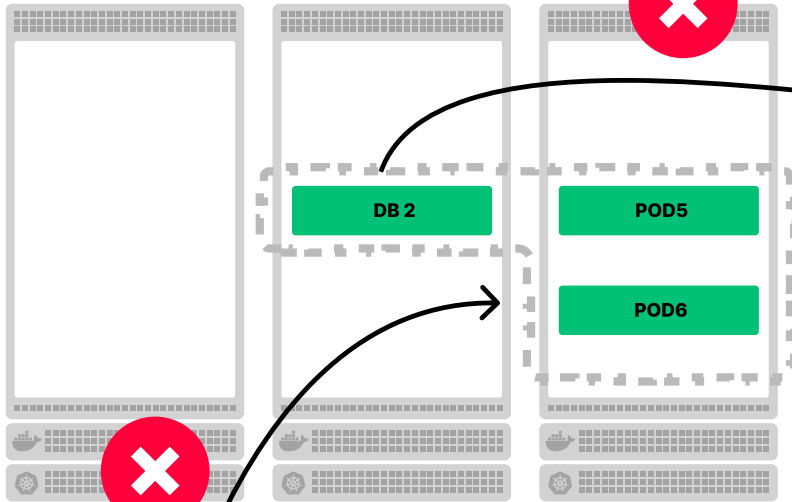


## CONTROL PLANE

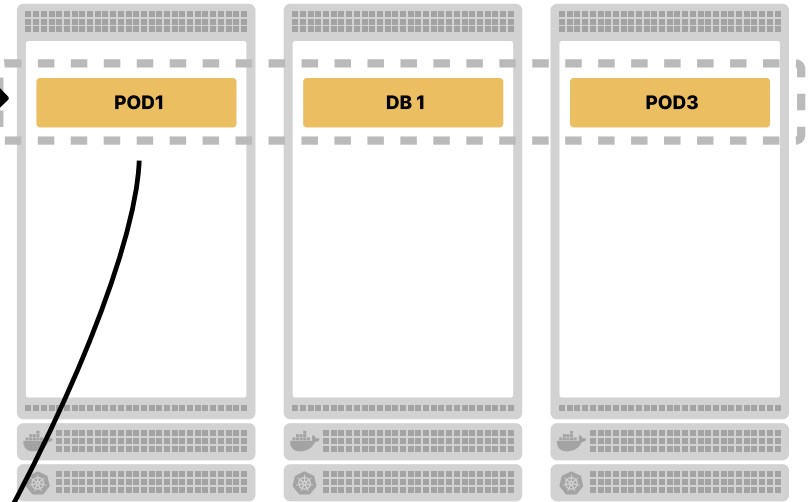




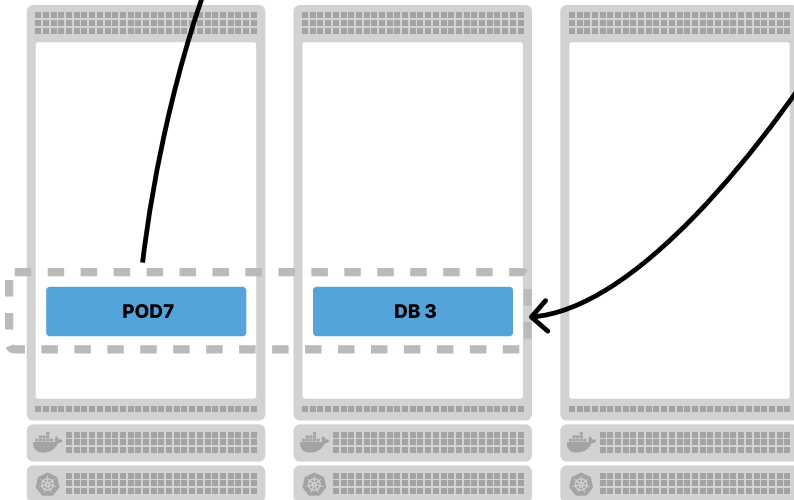
## POOL 1



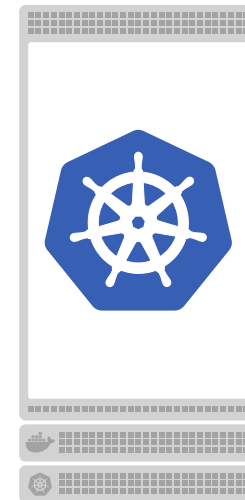
## POOL 2



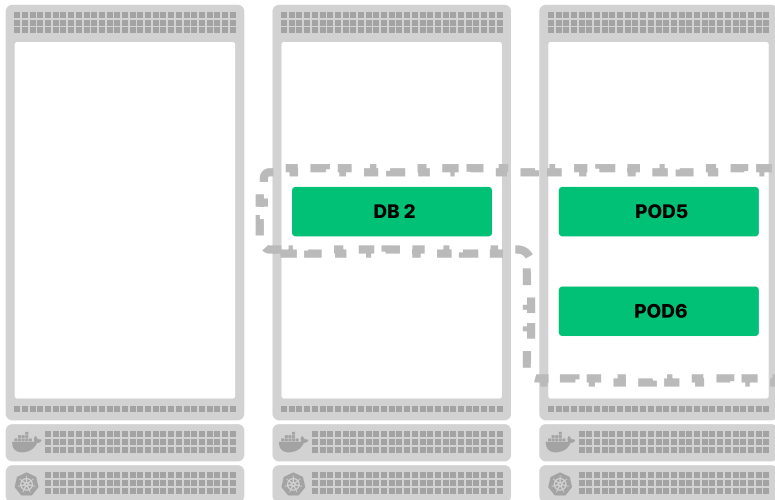
## POOL 3



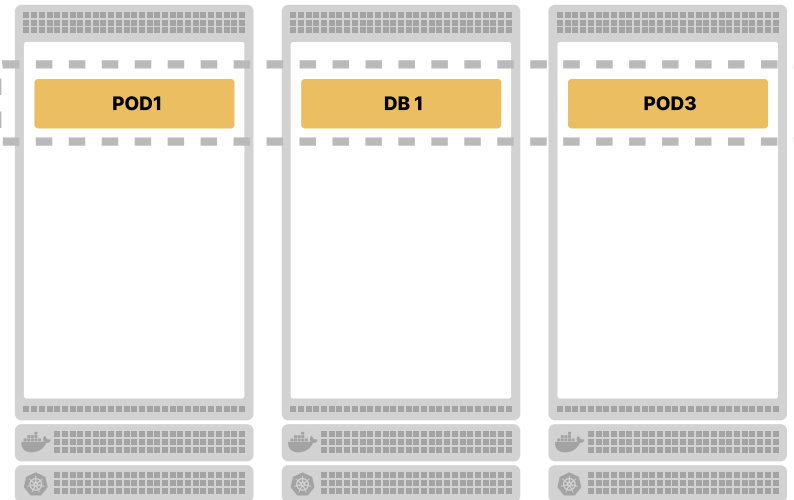
## CONTROL PLANE



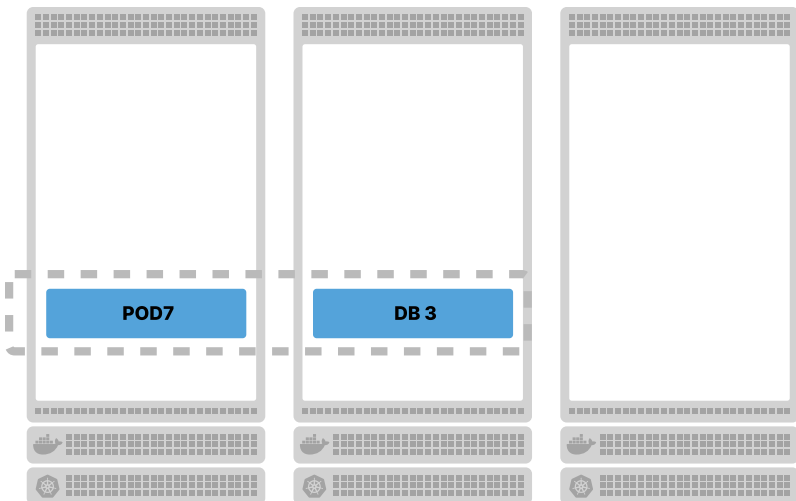
## POOL 1



## POOL 2



## POOL 3



## CONTROL PLANE



--isolate

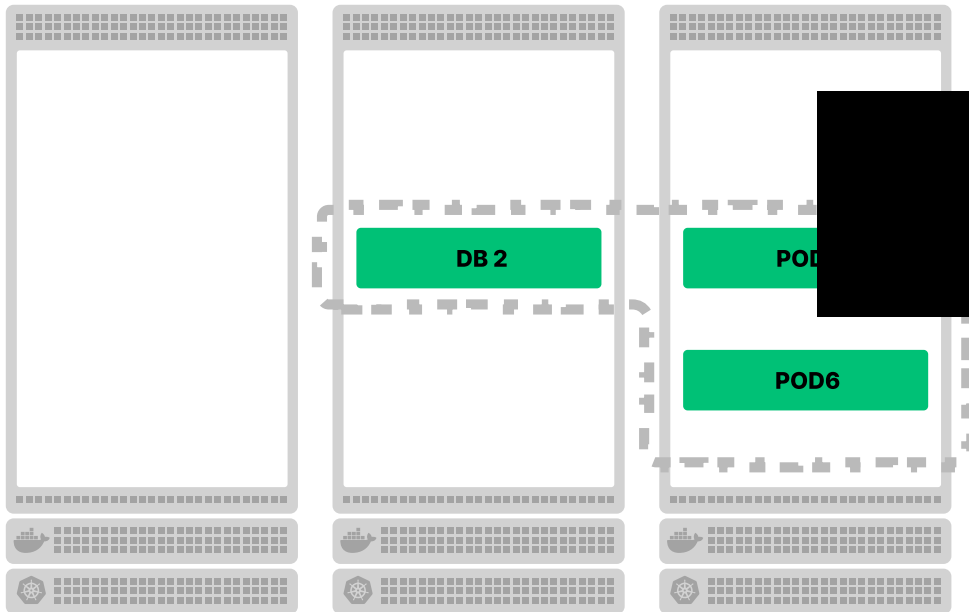
# **vCluster and shared cluster**

---



container escape

## POOL 1

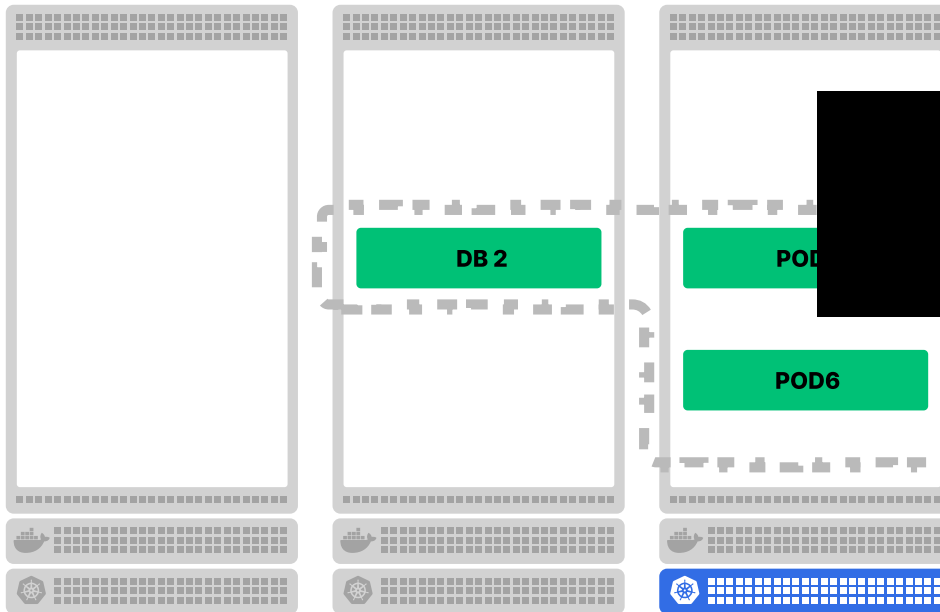


## CONTROL PLANE



container escape

## POOL 1



## CONTROL PLANE

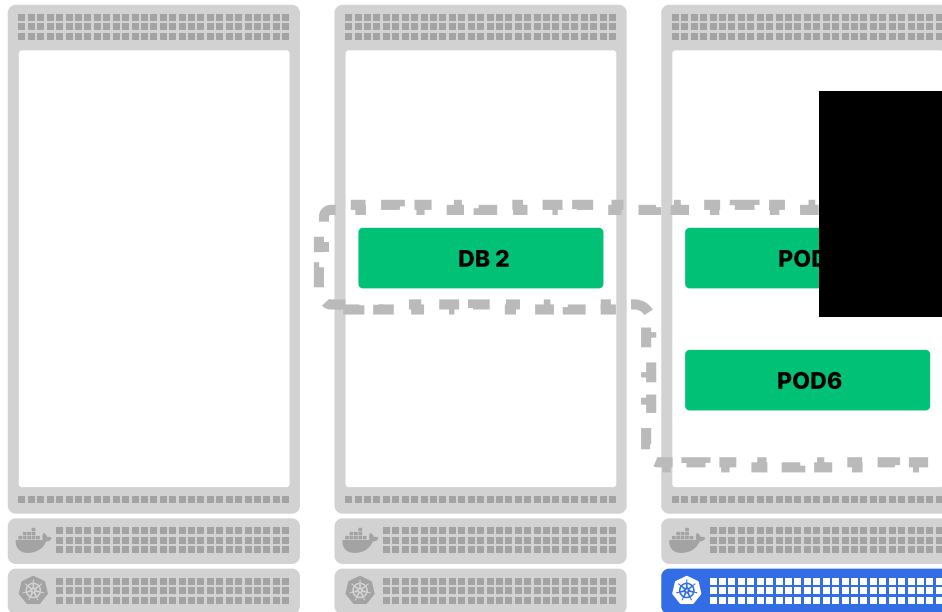


kubelet take over

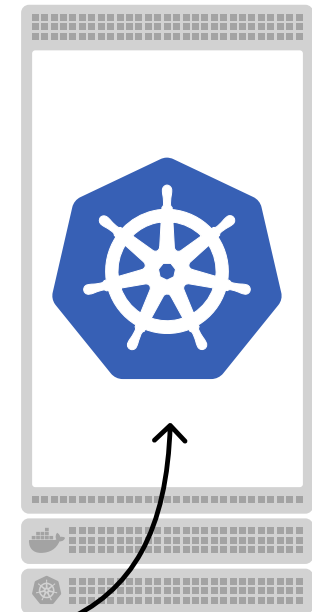


container escape

**POOL 1**



**CONTROL PLANE**



kubelet take over

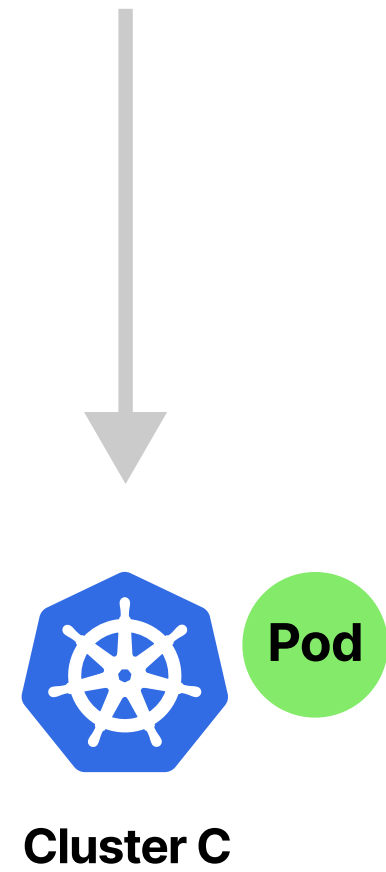
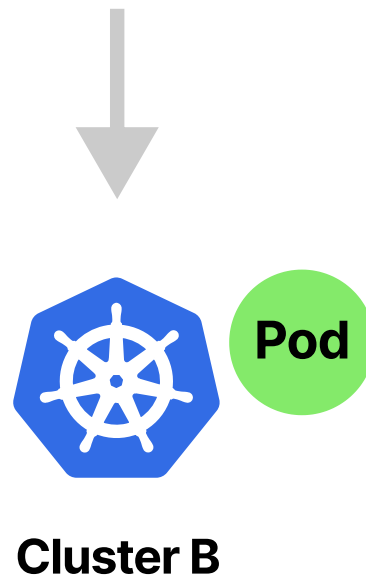
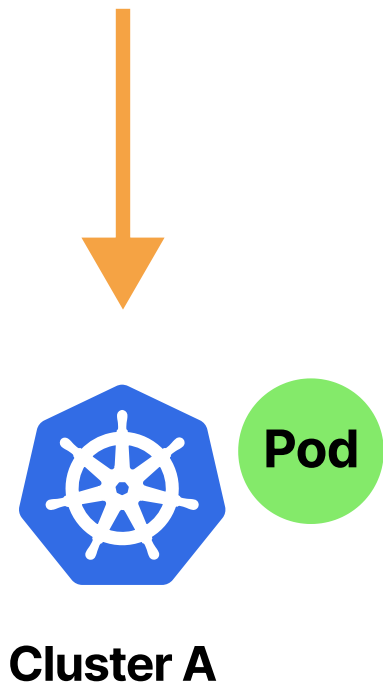
control plane  
escalation



# Dedicated clusters

---







# Karmada

---



**kubectl**



**API SERVER**



**CONTROLLER  
MANAGER**

**SCHEDULER**





KARMADA

**API SERVER**



KARMADA

**CONTROLLER  
MANAGER**



KARMADA

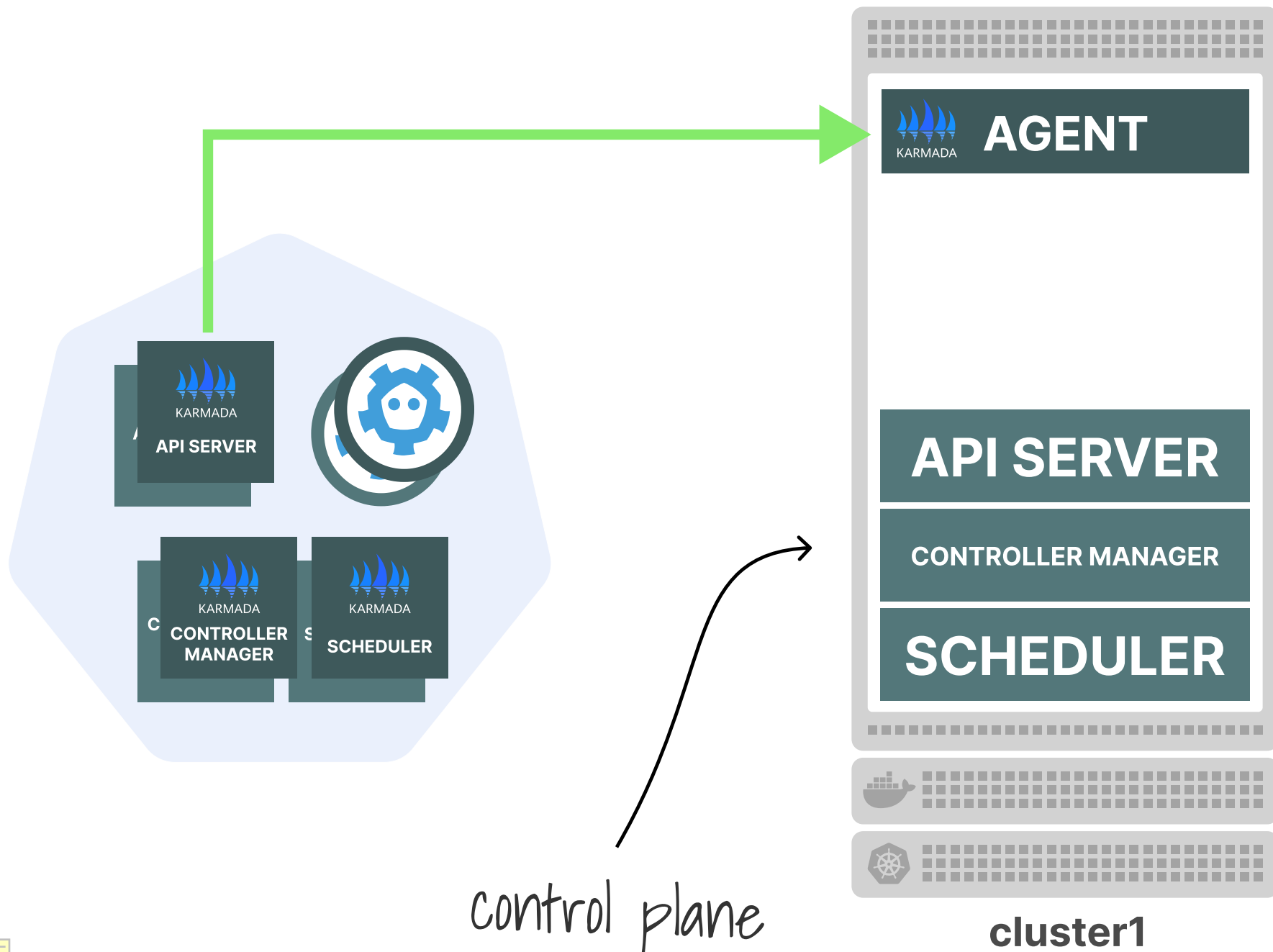
**SCHEDULER**

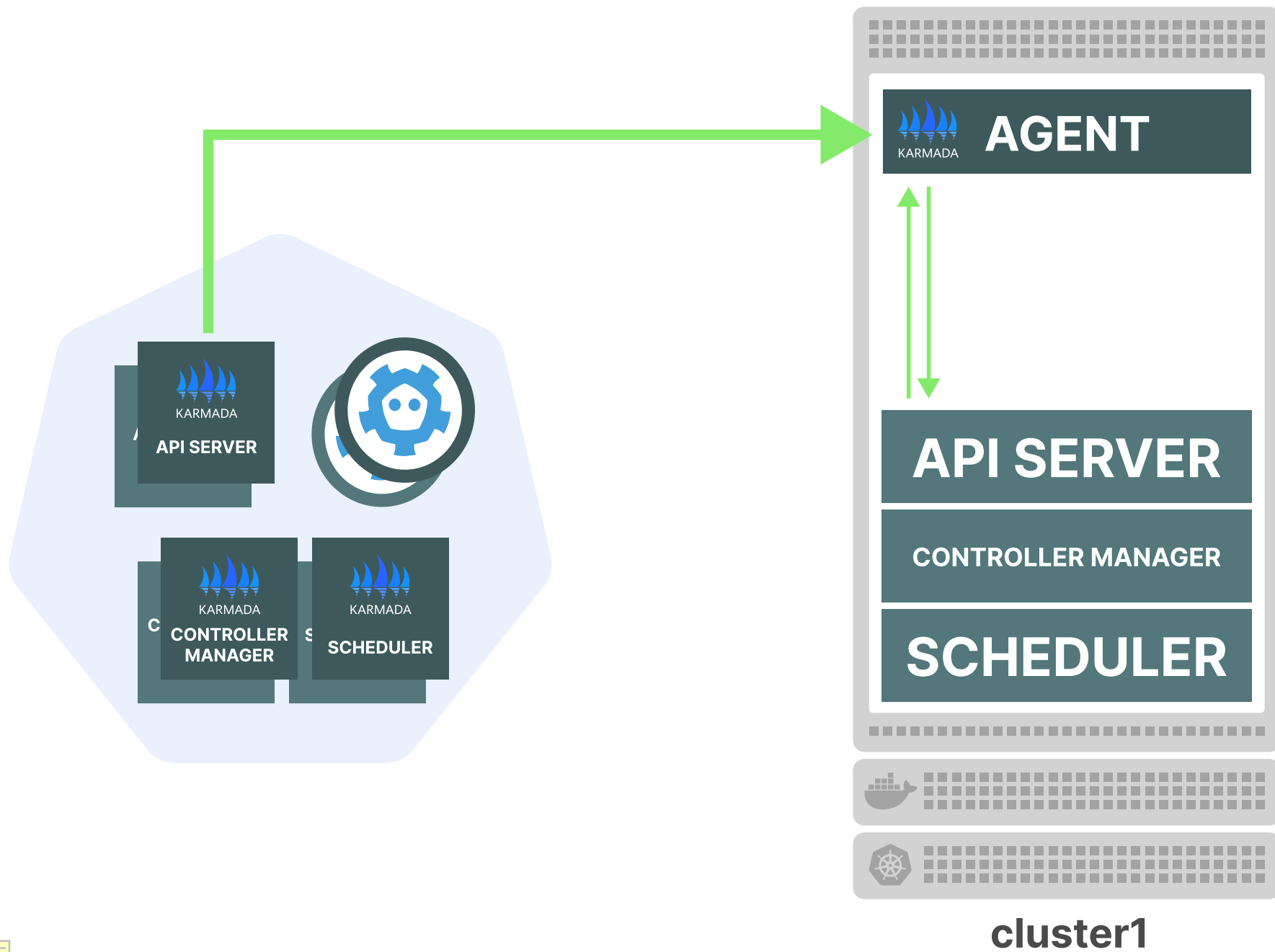


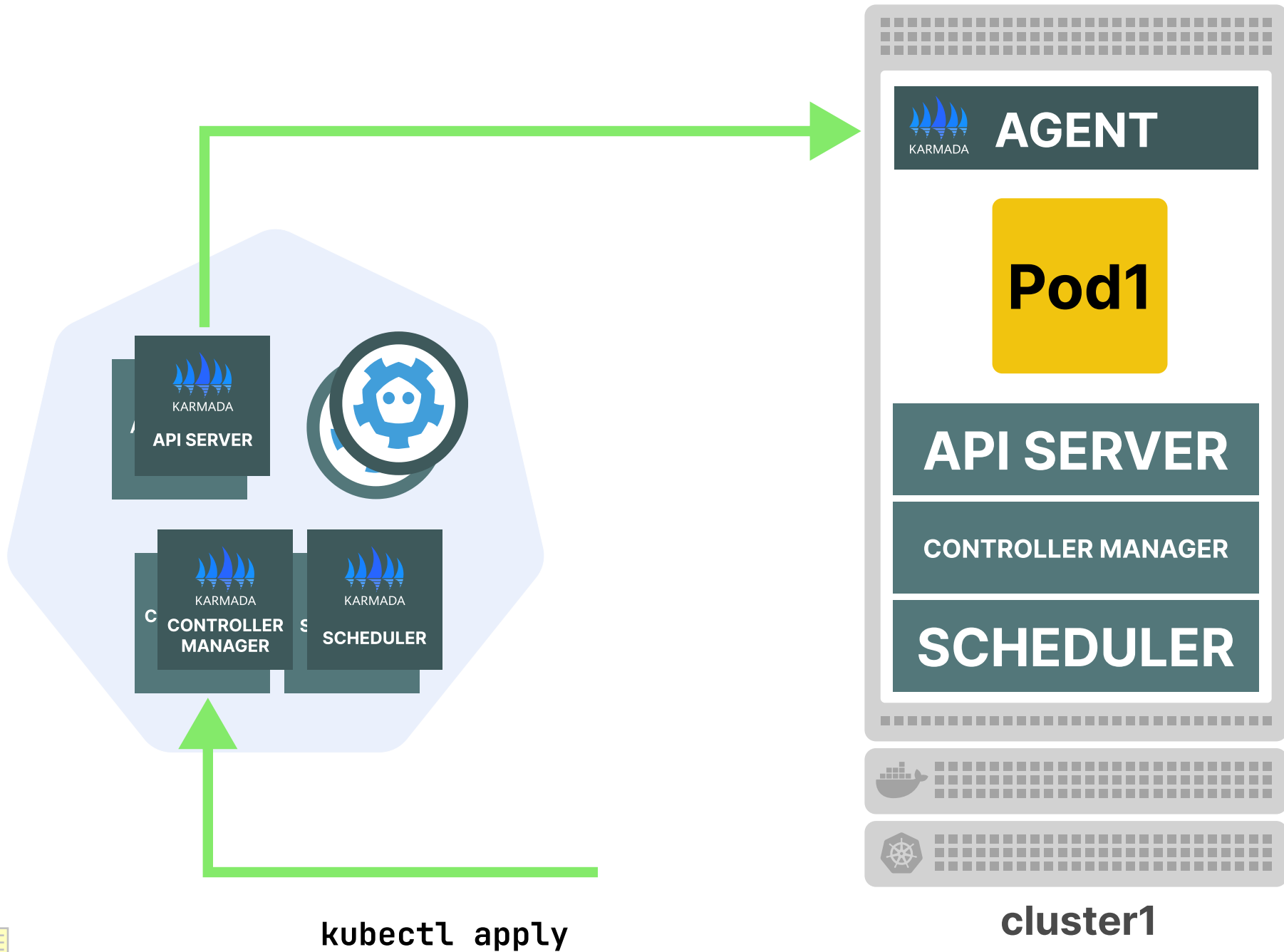
# Karmada architecture

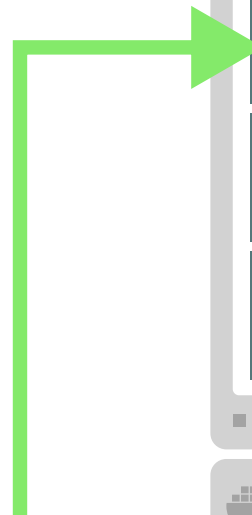
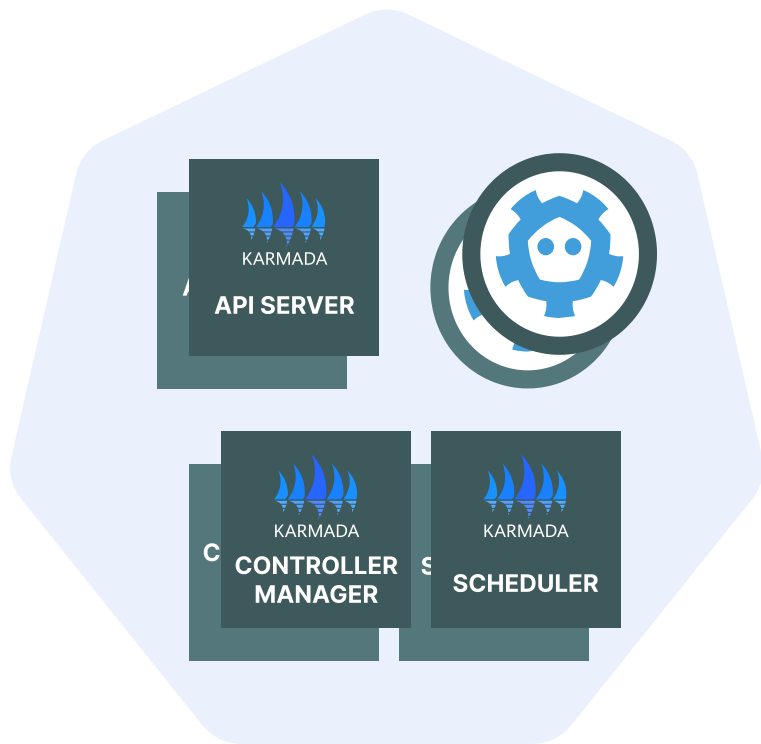
---



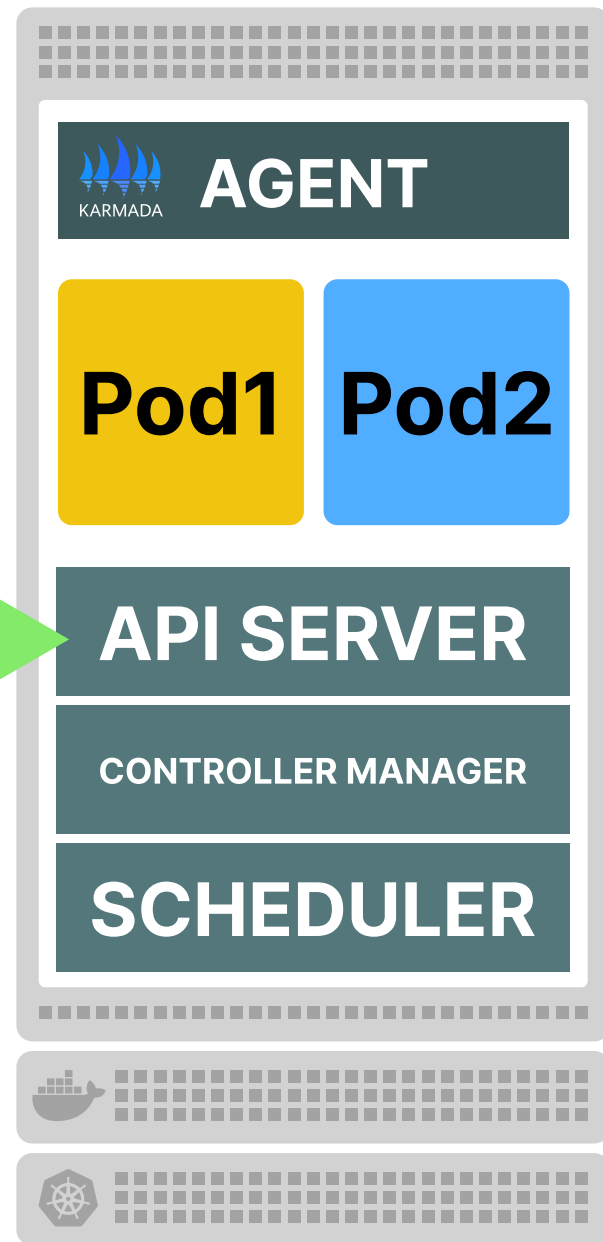








`kubectl apply`



**cluster1**





# **Independent cluster with central management**

---



kubectl



manager

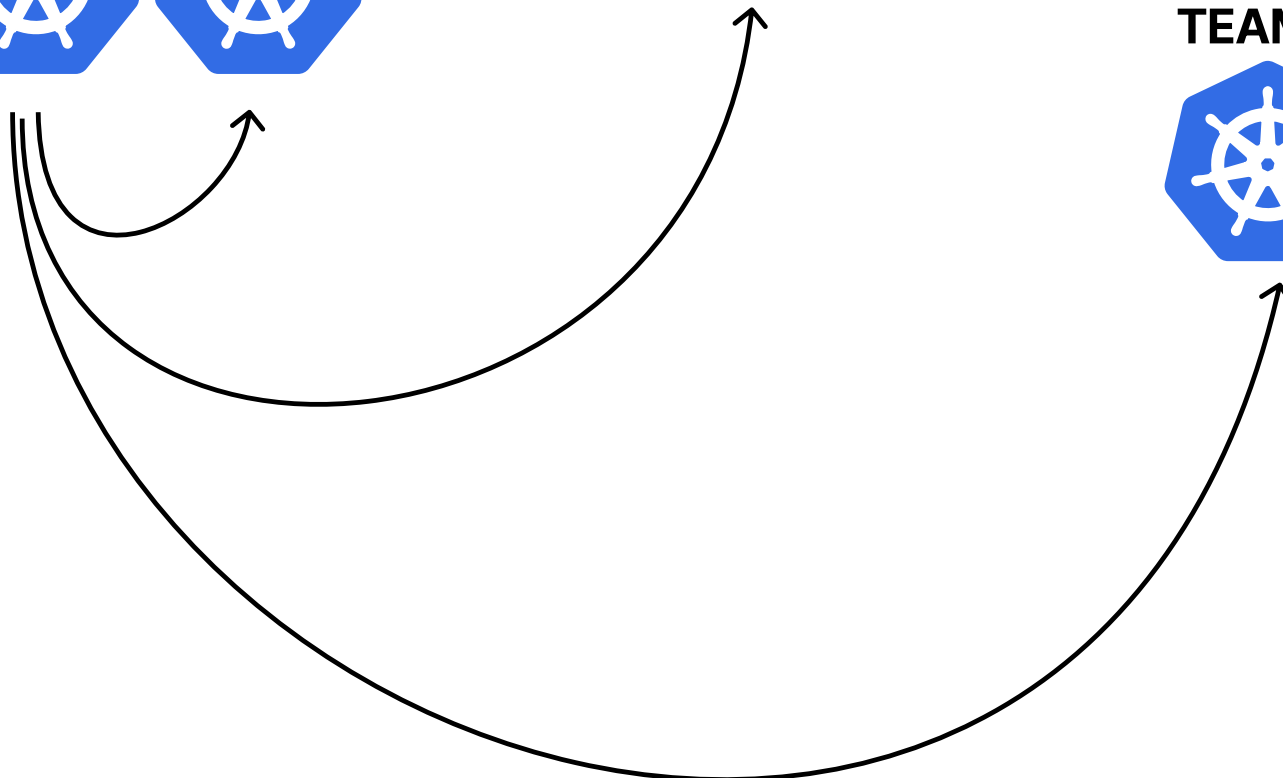
TEAM A



TEAM B



TEAM C



kubectl



manager

TEAM A



Pod

TEAM B

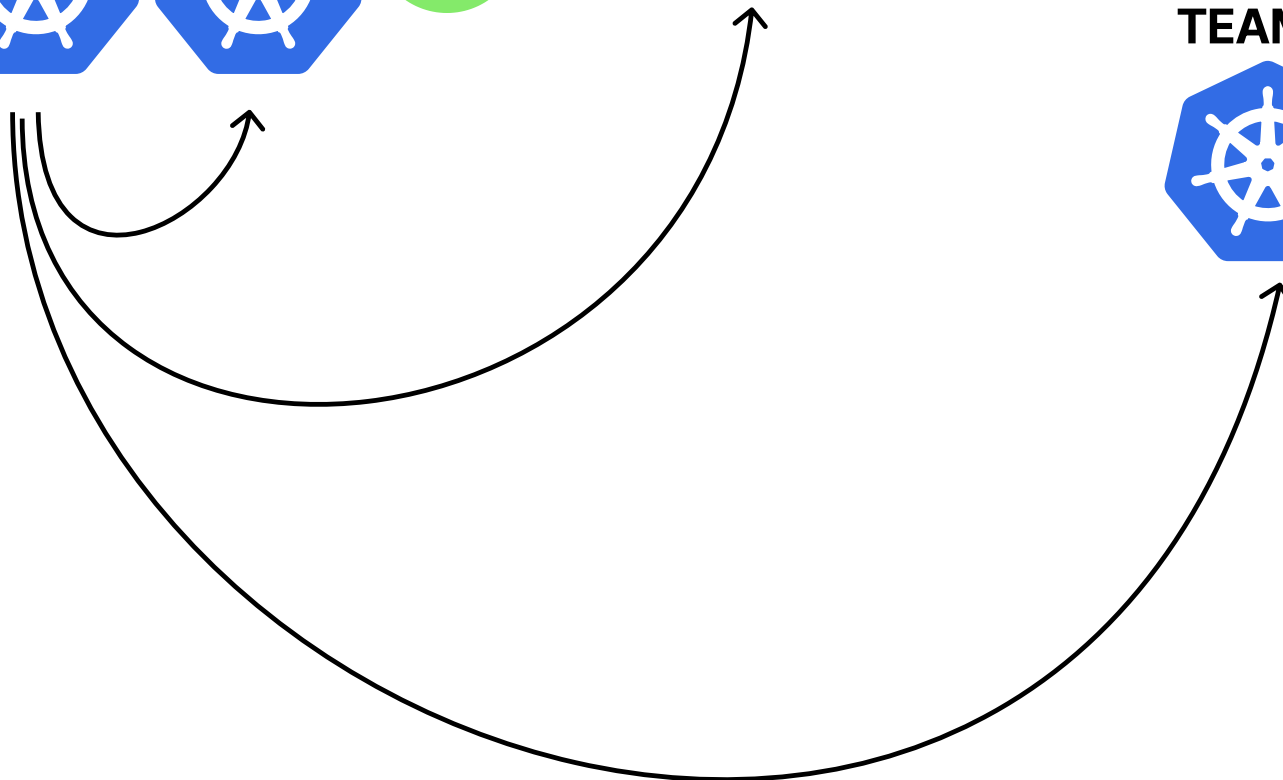


Pod

TEAM C



Pod



kubectl



manager

TEAM A



Pod

Pod

TEAM B



Pod

TEAM C



Pod



kubectl



manager

TEAM A



Pod

Pod

TEAM B



Pod

Pod

TEAM C



Pod



kubectl



manager

TEAM A



Pod

Pod

TEAM B



Pod

Pod

TEAM C



Pod

Pod



# Demo

---



# Karmada

---

## Cluster of clusters





# Karmada

---

## **Cluster of clusters**

## **Admin vs tenants**

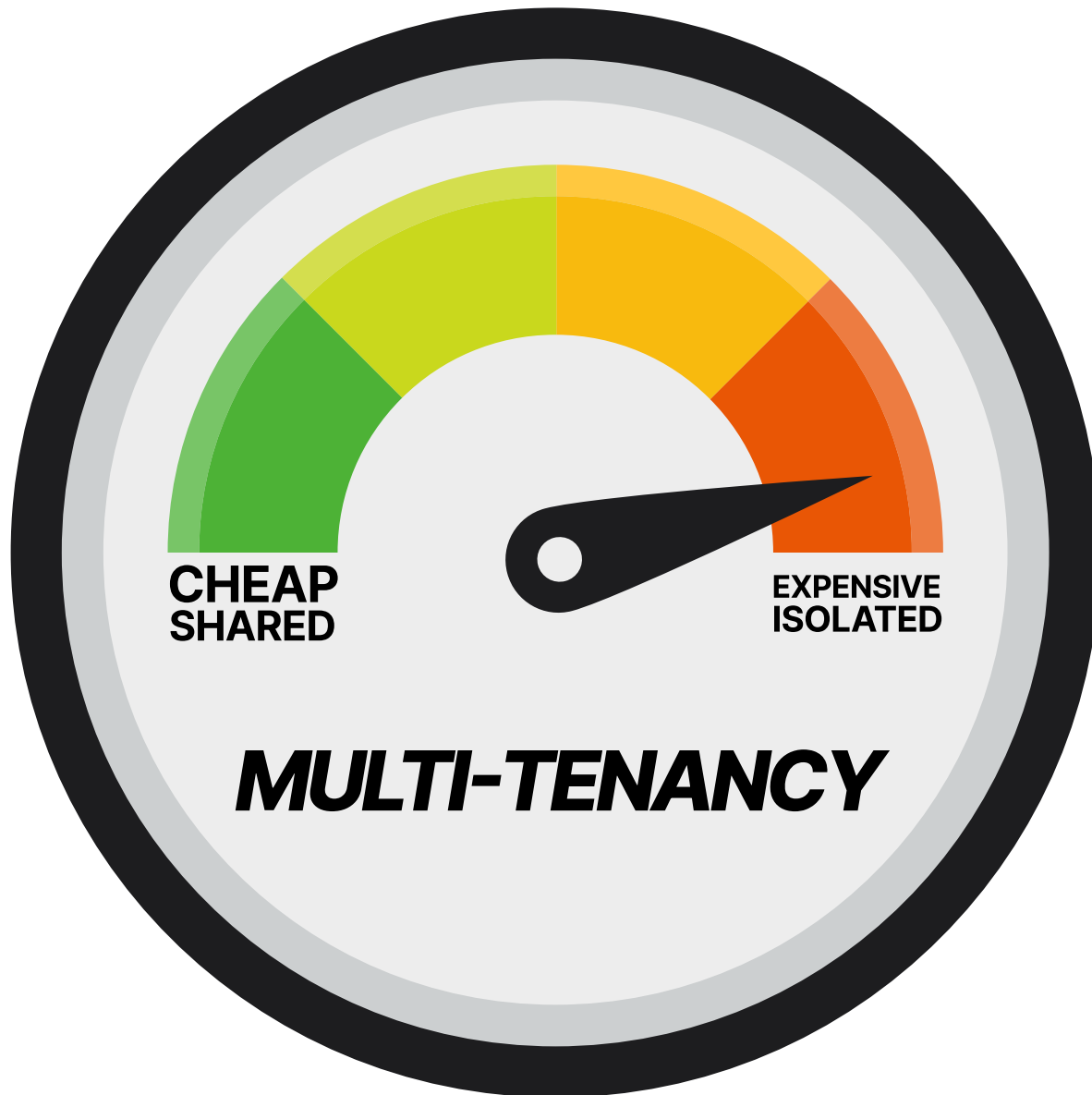


**Cluster of clusters**

**Admin vs tenants**

**No sharing**





## COSTS FOR 50 TENANTS

**+ 51 clusters x \$0**



## COSTS FOR 50 TENANTS

**+ 51 clusters x \$0**

**+ 51 nodes x \$12**



## COSTS FOR 50 TENANTS

**+ 51 clusters x \$0**

**+ 51 nodes x \$12**



**= \$612 / month**

**~\$12 / month / tenant**



# Multi-tenancy baseline

---



**Multi-tenancy**

**Kubernetes**





**Node pools, Sandbox runtime**

**Multi-tenancy**

**Kubernetes**



monitoring



**Node pools, Sandbox runtime**

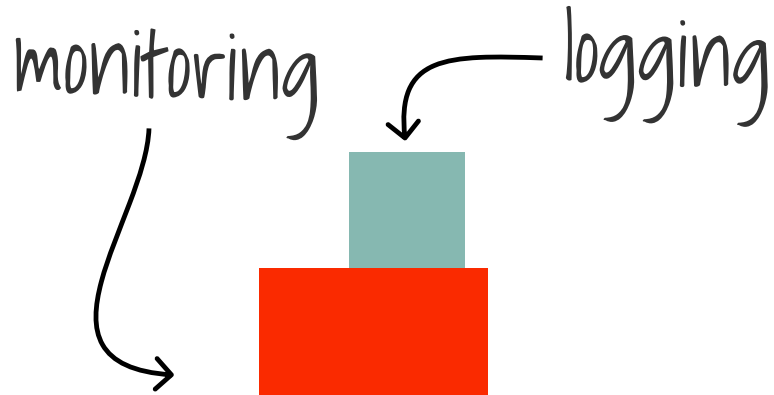
**Multi-tenancy**

**Kubernetes**



monitoring

logging

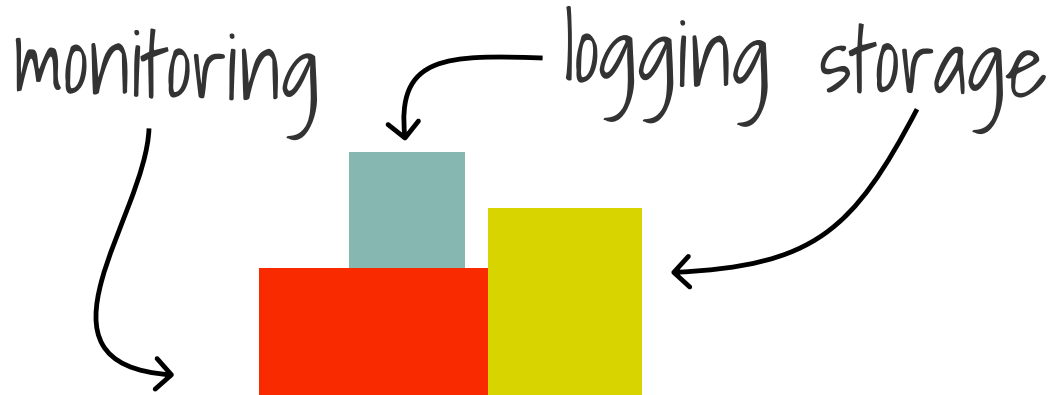


**Node pools, Sandbox runtime**

**Multi-tenancy**

**Kubernetes**





**Node pools, Sandbox runtime**

**Multi-tenancy**

**Kubernetes**



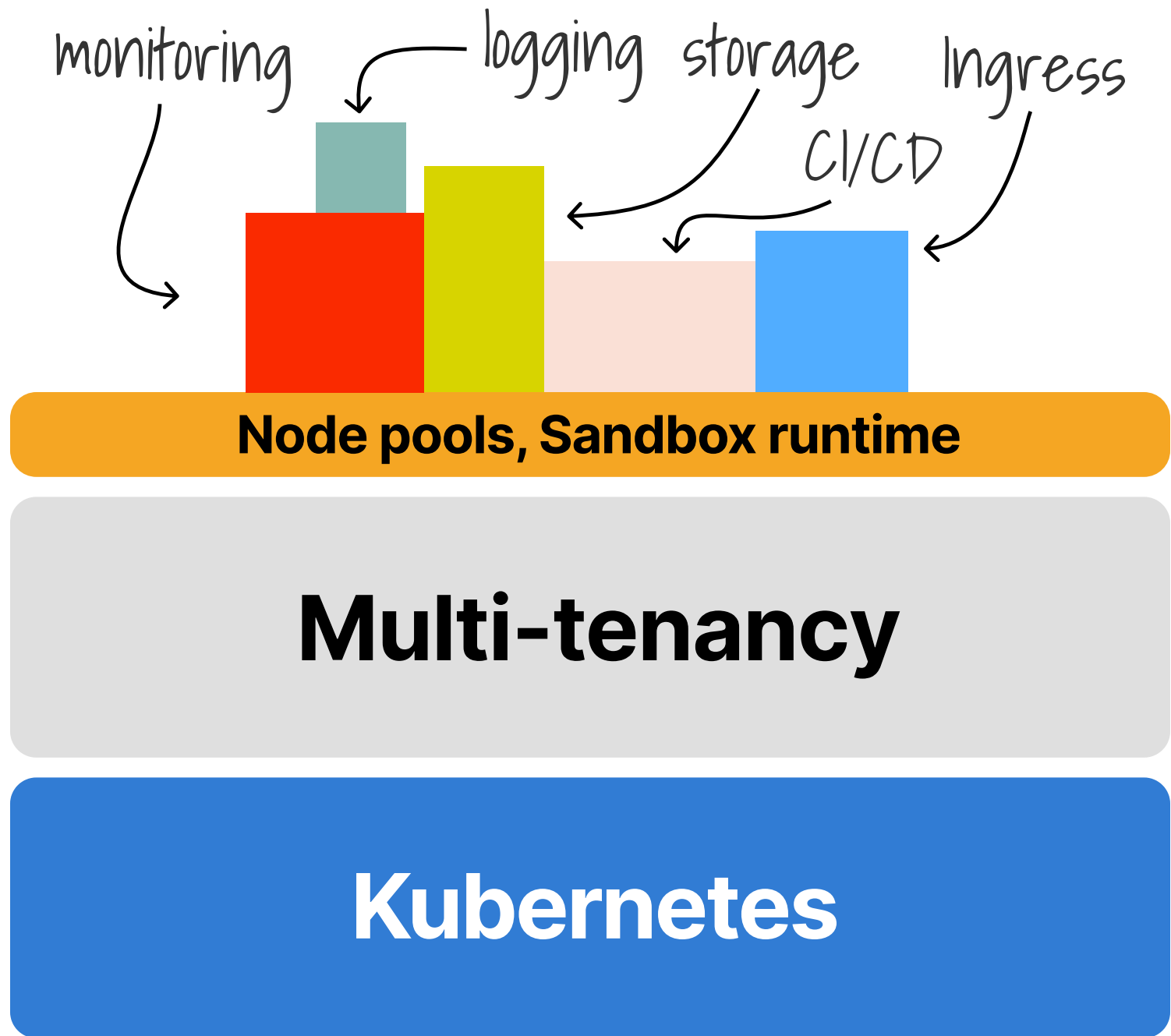


**Node pools, Sandbox runtime**

**Multi-tenancy**

**Kubernetes**





# Costs

---



**Cost**



**\$0**

**HNC**

**\$252**

**vCluster**

**\$612**

**Karmada**





# DEDICATED INGRESS FOR 50 TENANTS

50 × 3

CPU

5vCPU

MEMORY

4.5GB

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)***	EBS Bandwidth (Gbps)
c6i.large	2	4	EBS-Only	Up to 12.5	Up to 10
c6i.xlarge	4	8	EBS-Only	Up to 12.5	Up to 10
c6i.2xlarge	8	16	EBS-Only	Up to 12.5	Up to 10
c6i.4xlarge	16	32	EBS-Only	Up to 12.5	Up to 10
c6i.8xlarge	32	64	EBS-Only	12.5	10

\$0.34/hr

\$248.2/m



**Cost**

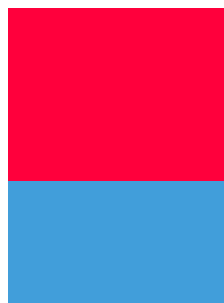


**\$248**



**HNC**

**\$469**



**vCluster**

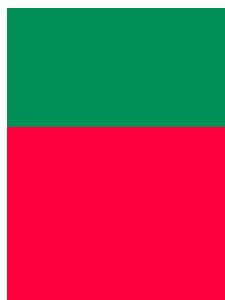
**\$860**



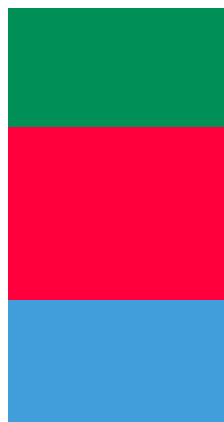
**Karmada**



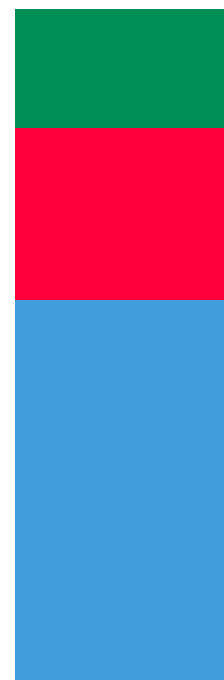
**Cost**



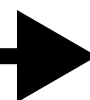
**HNC**



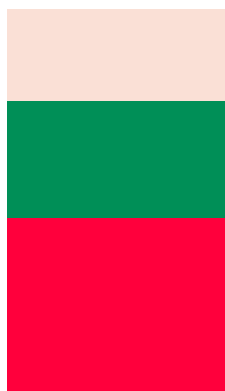
**vCluster**



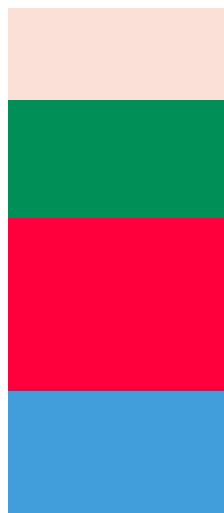
**Karmada**



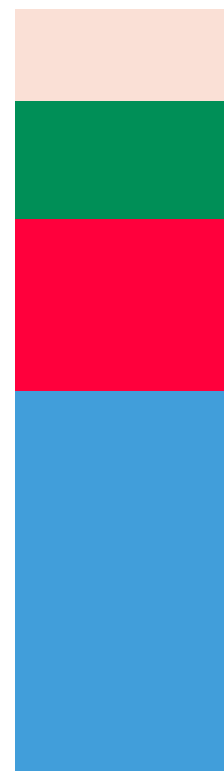
**Cost**



**HNC**



**vCluster**



**Karmada**

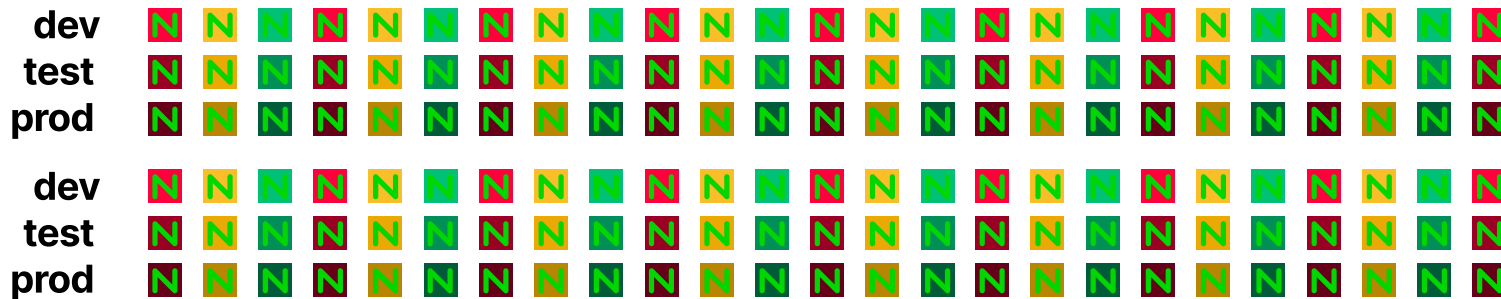


# Costs\*

---



**Cost**



**\$248**

**\$469**

**\$860**

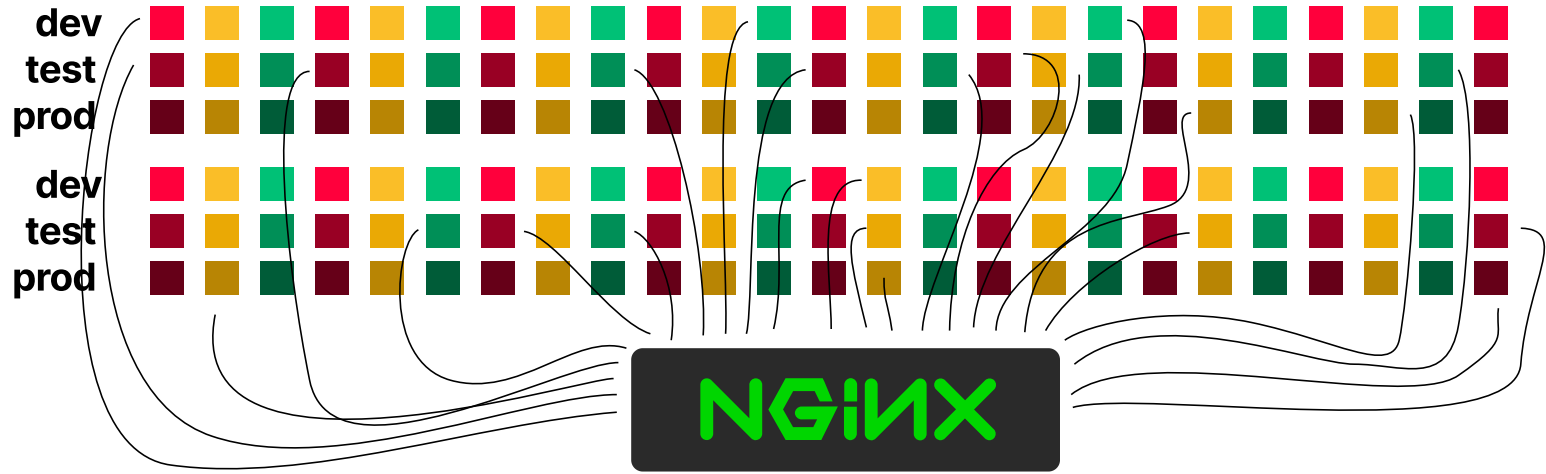
**HNC**

**vCluster**

**Karmada**



**Cost**



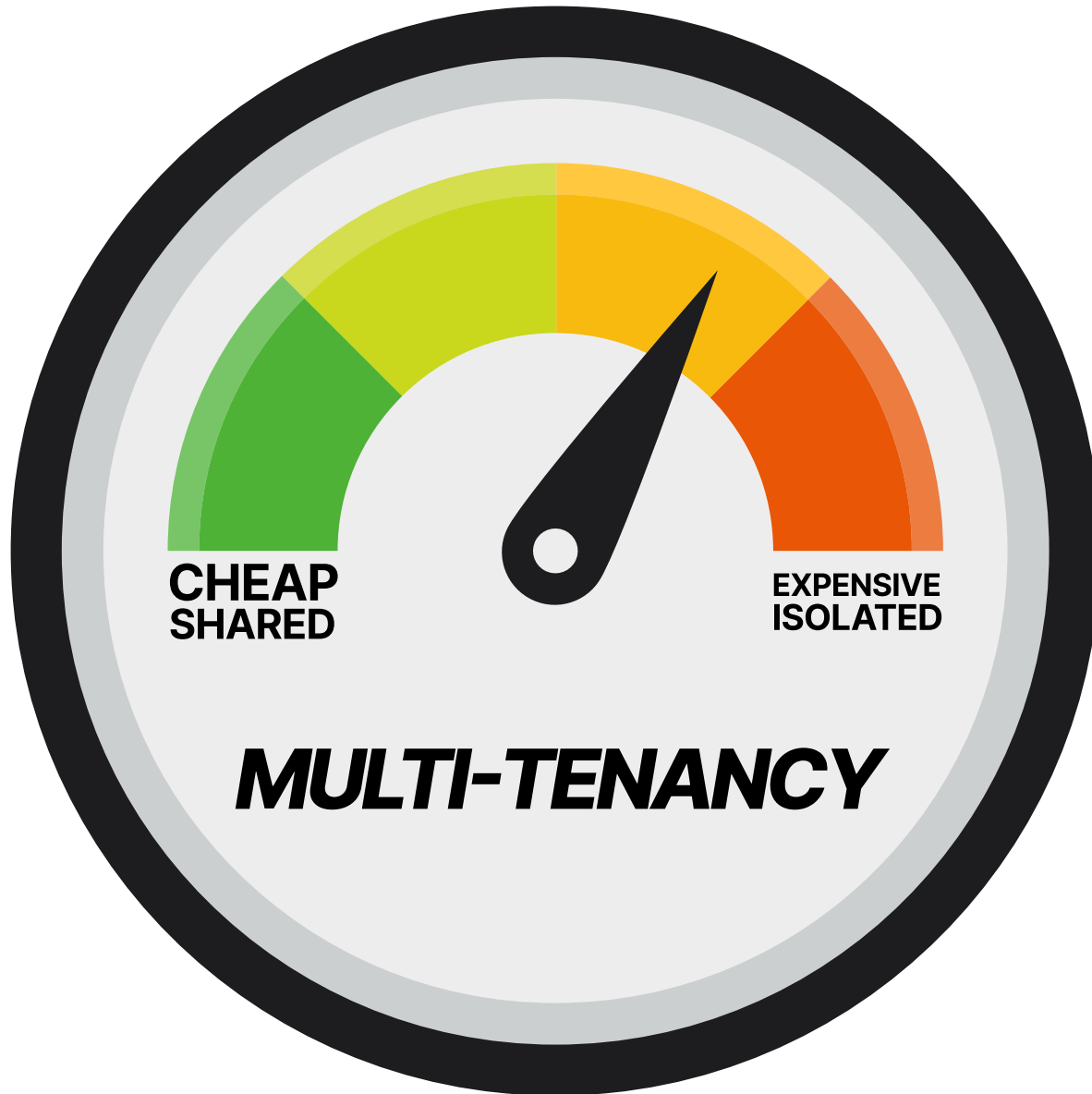
50x ingress  
controllers

**HNC**

**vCluster**

**Karmada**







# Multi-tenant platform from scratch

---

Recap



# Recap

---

**1. Isolation VS costs**

2. Multi-tenant components (e.g. Ingress)

3. Constant vs linear vs exponential costs

4. HNC and vCluster

5. Karmada



# Recap

---

1. Isolation VS costs

**2. Multi-tenant components (e.g. Ingress)**

3. Constant vs linear vs exponential costs

4. HNC and vCluster

5. Karmada



# Recap

---

1. Isolation VS costs
2. Multi-tenant components (e.g. Ingress)
- 3. Constant vs linear vs exponential costs**
4. HNC and vCluster
5. Karmada



# Recap

---

1. Isolation VS costs

2. Multi-tenant components (e.g. Ingress)

3. Constant vs linear vs exponential costs

**4. HNC and vCluster**

5. Karmada



# Recap

---

1. Isolation VS costs
2. Multi-tenant components (e.g. Ingress)
3. Constant vs linear vs exponential costs
4. HNC and vCluster
- 5. Karmada**



loft

**Thank you!**



# Thank you!

---

 Chris Nesbitt-Smith





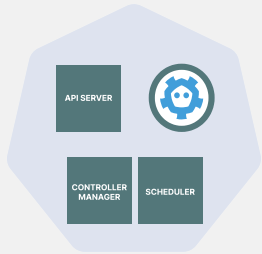


# **Hypershift/Kamaji**

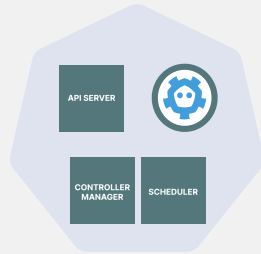
---

# CLUSTER

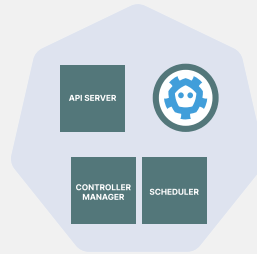
control plane as a pod



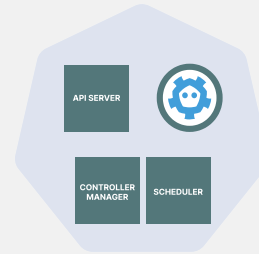
POD1



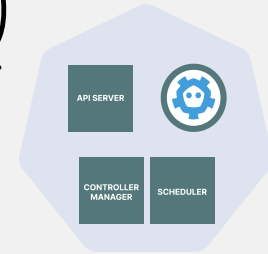
POD2



POD3



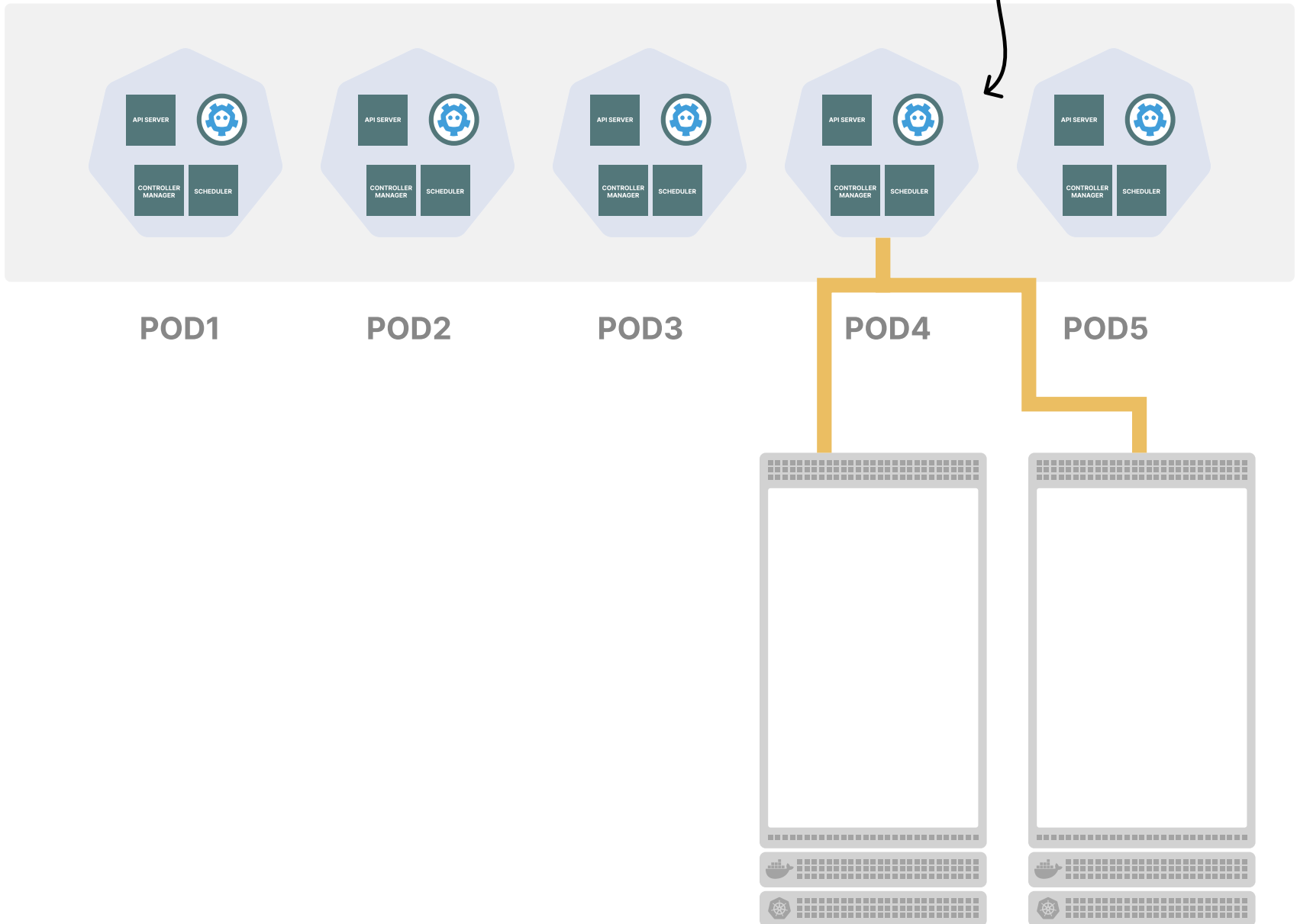
POD4



POD5

# CLUSTER

control plane as a pod



# CLUSTER

control plane as a pod

