# Building a Kubernetes platform

Chris Nesbitt-Smith

learnk8s

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

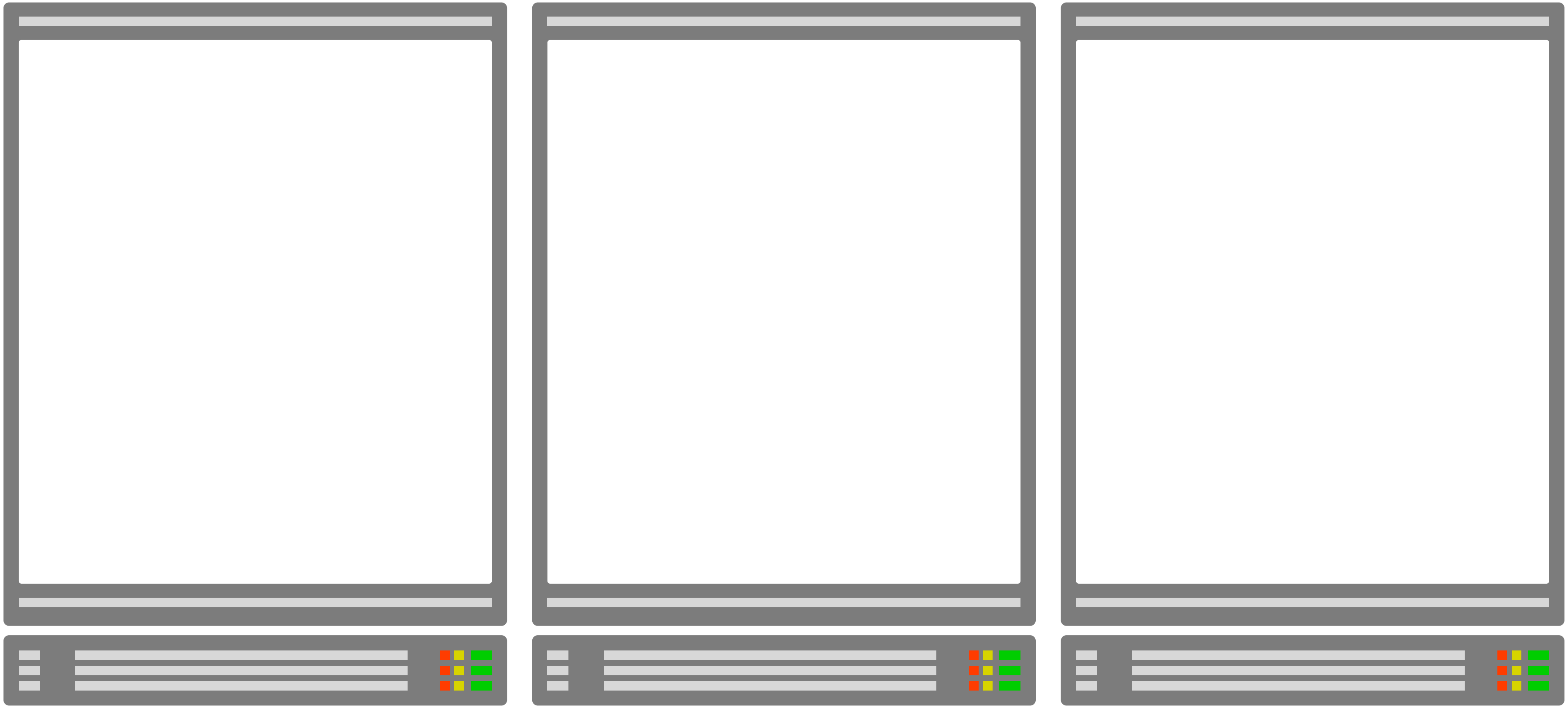
Isolation

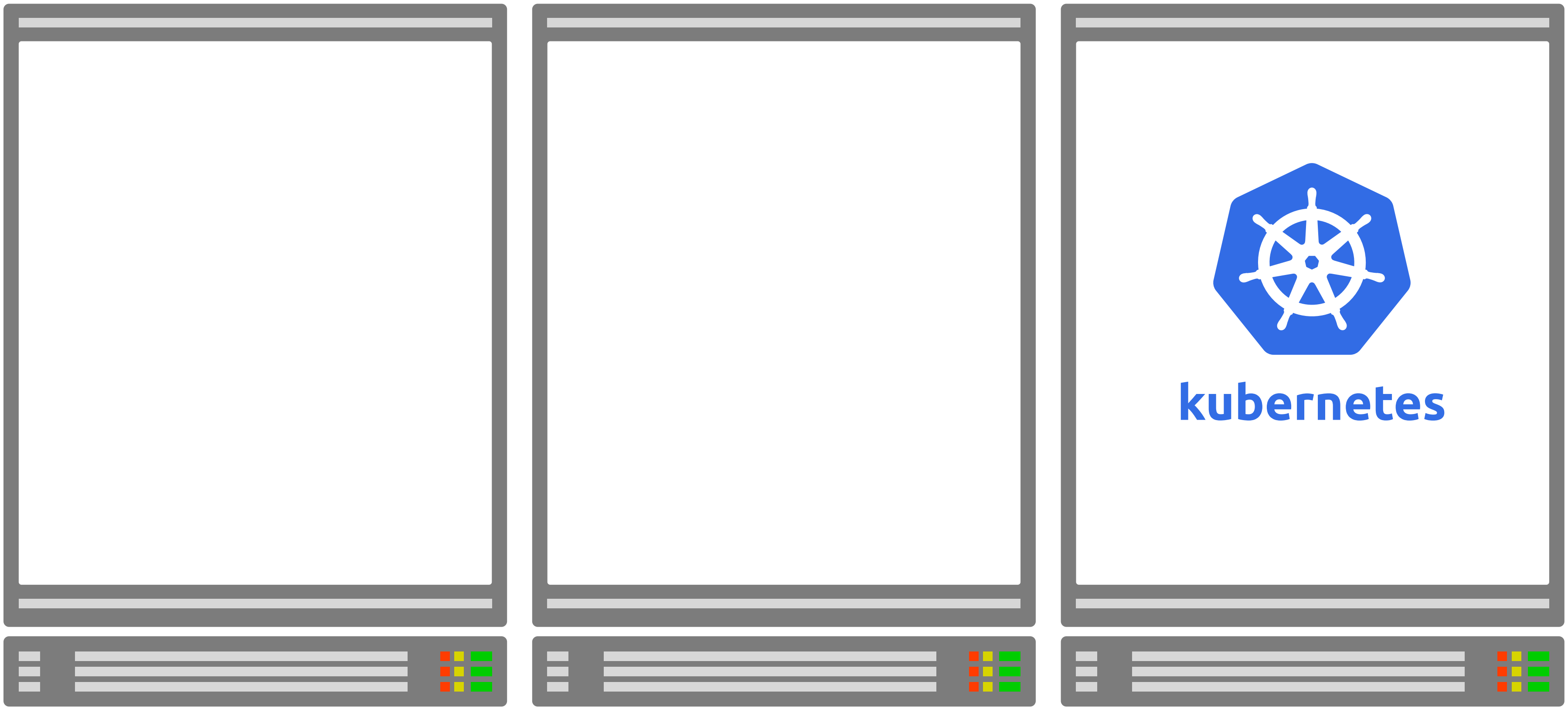
Ease of management

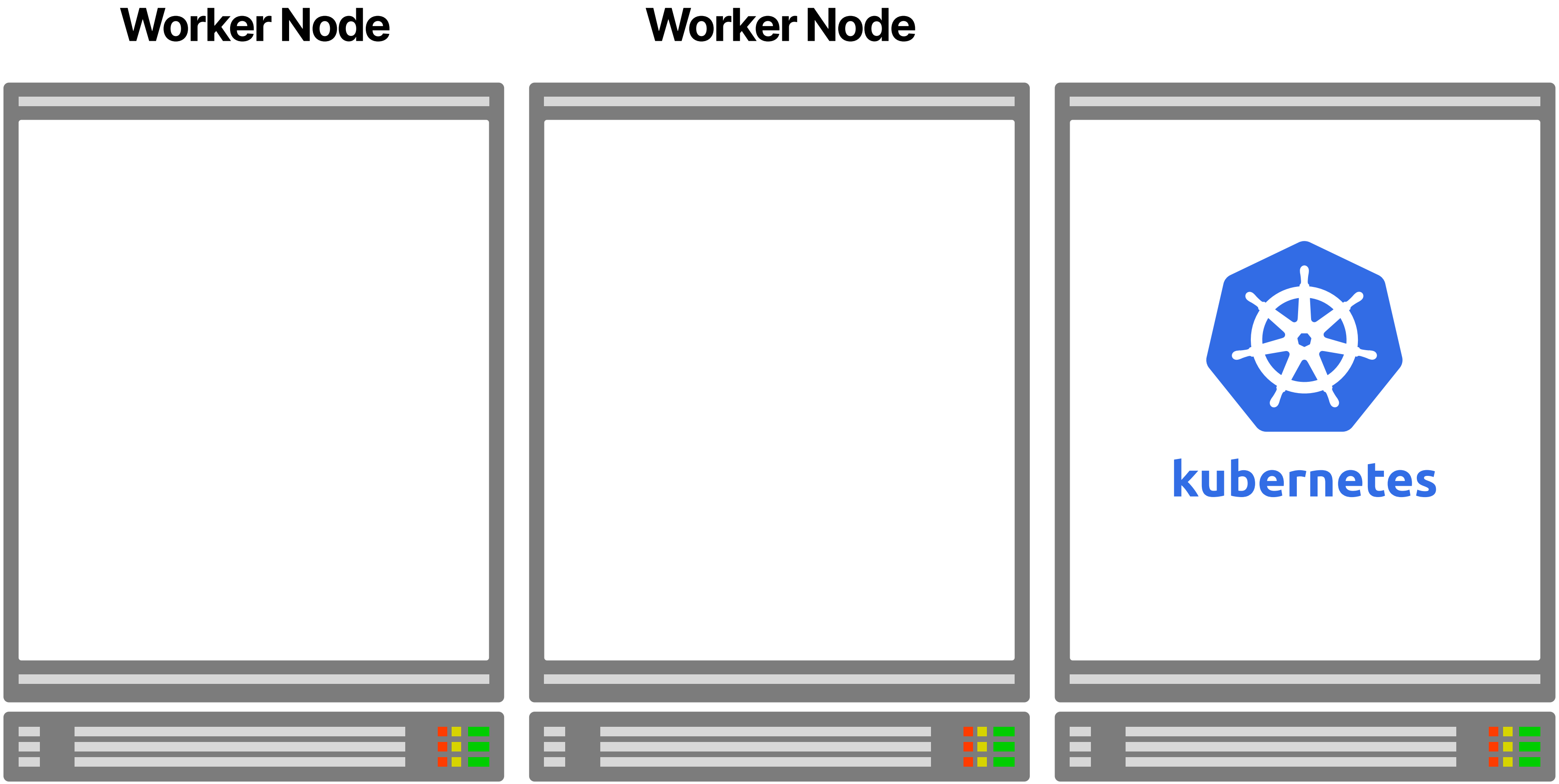
**Cost efficiency**

# Datacentre as a single computer

01

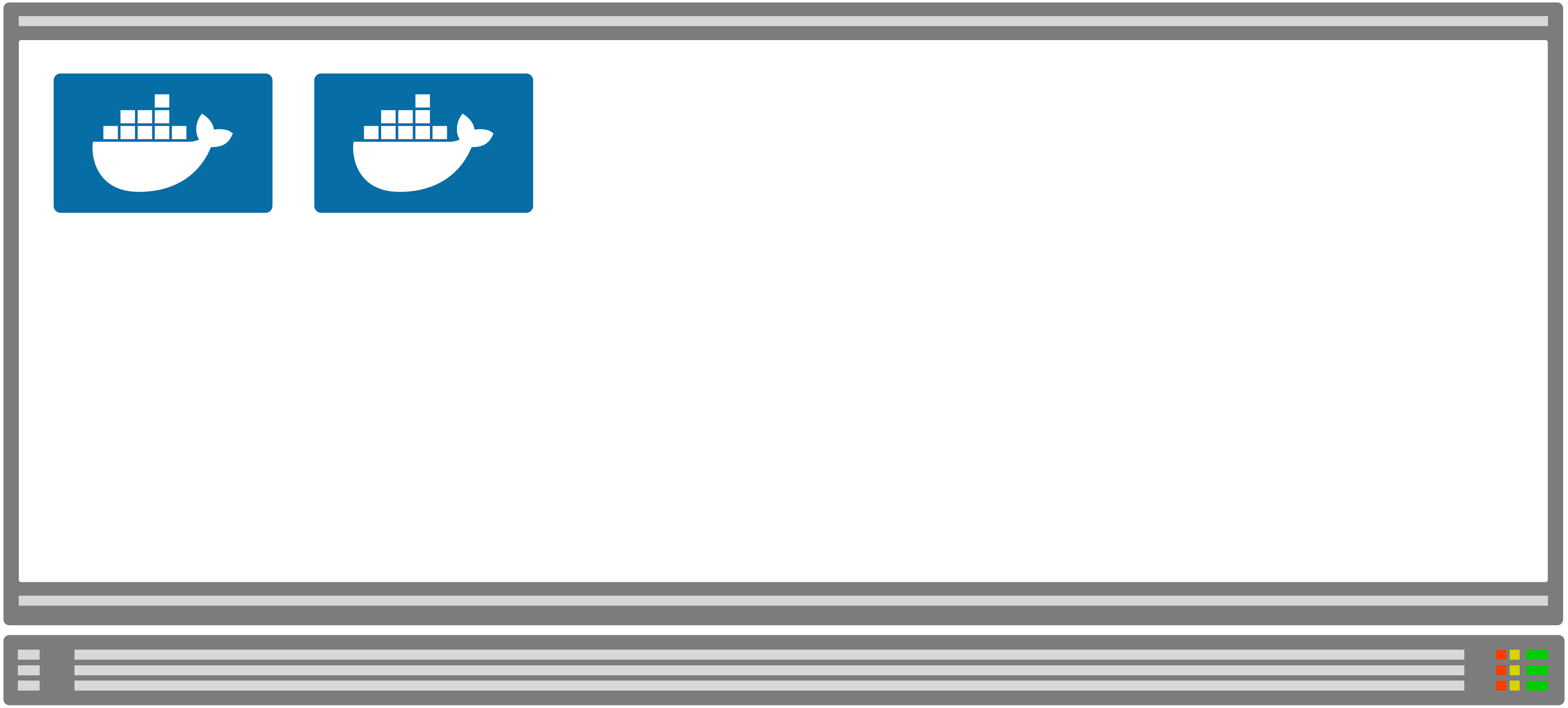


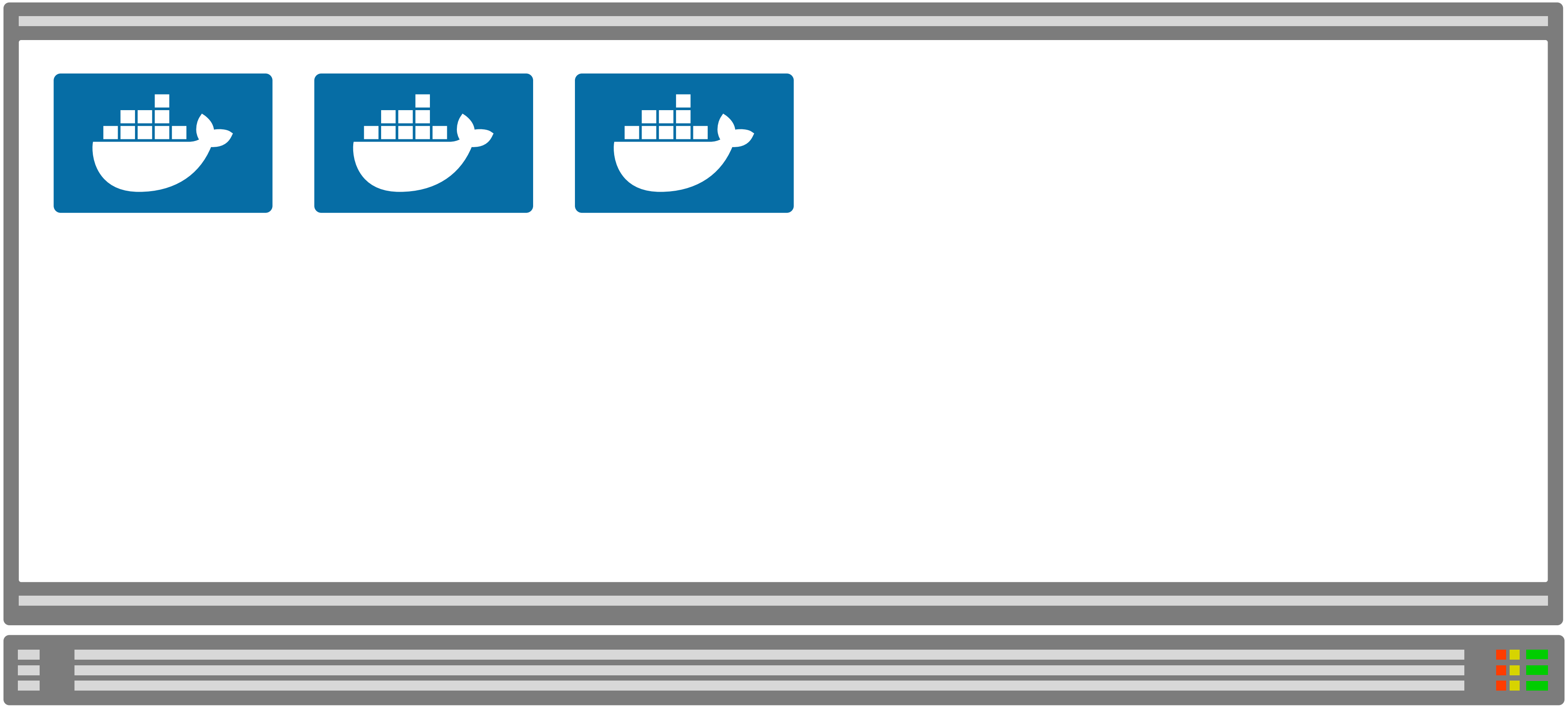

kubernetes



Worker Node

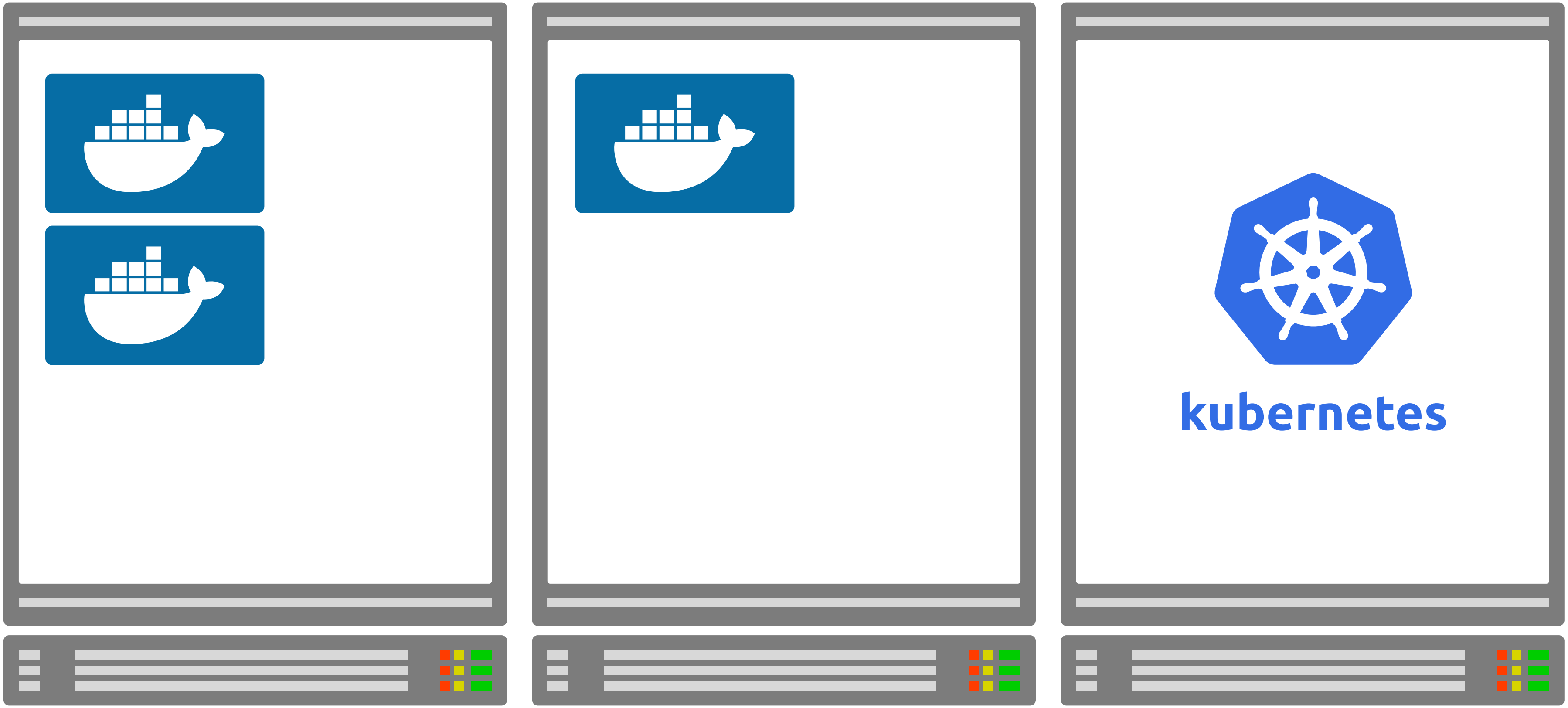
Worker Node





Worker Node

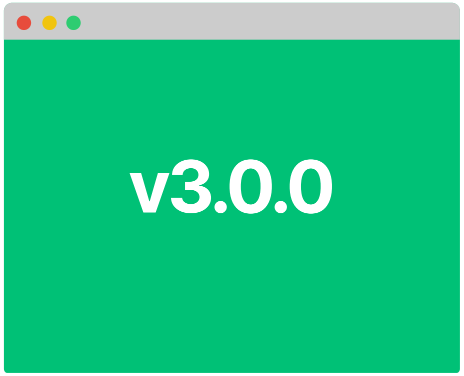
Worker Node

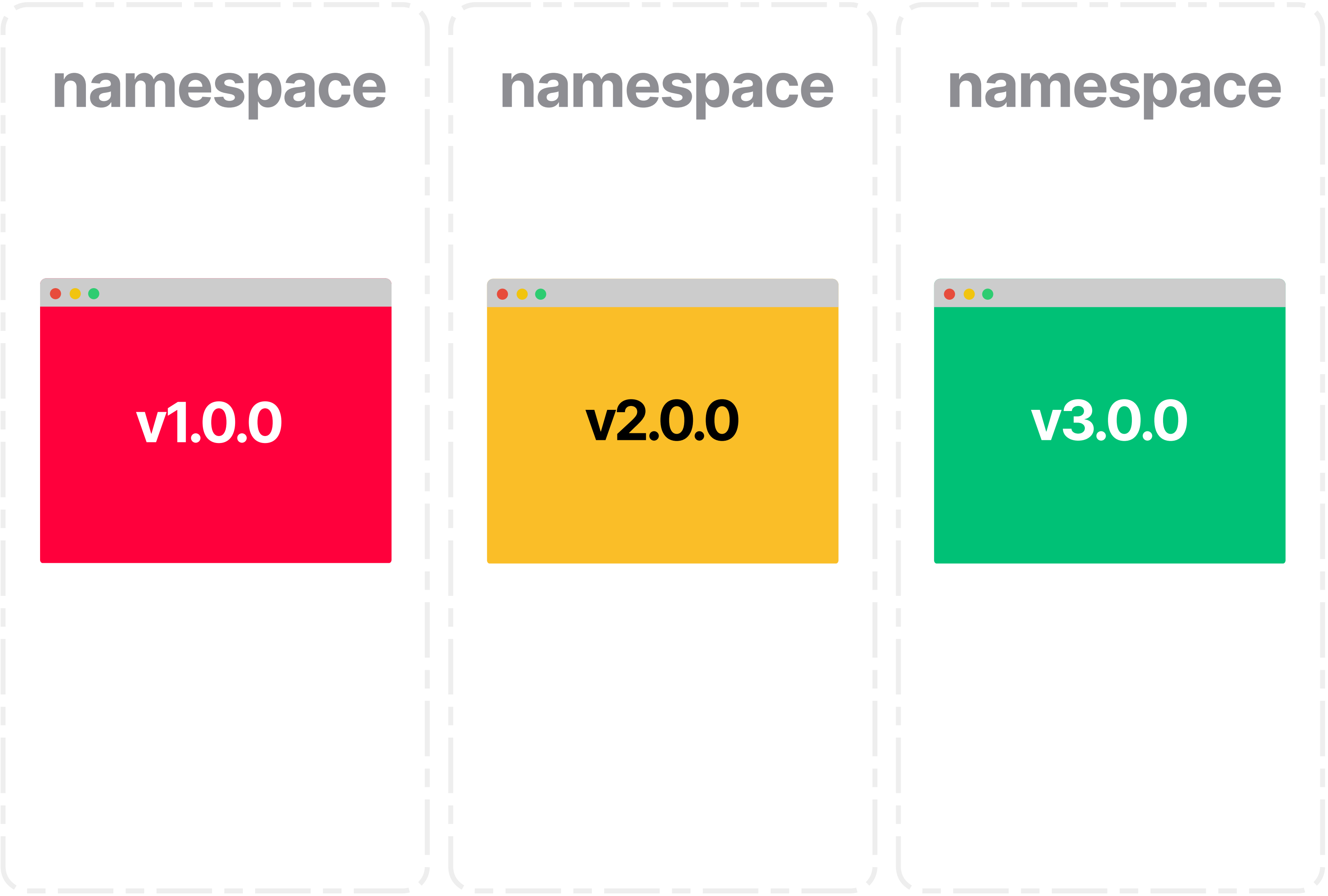


# Namespaces

v1.0.0

v2.0.0


v3.0.0



namespace

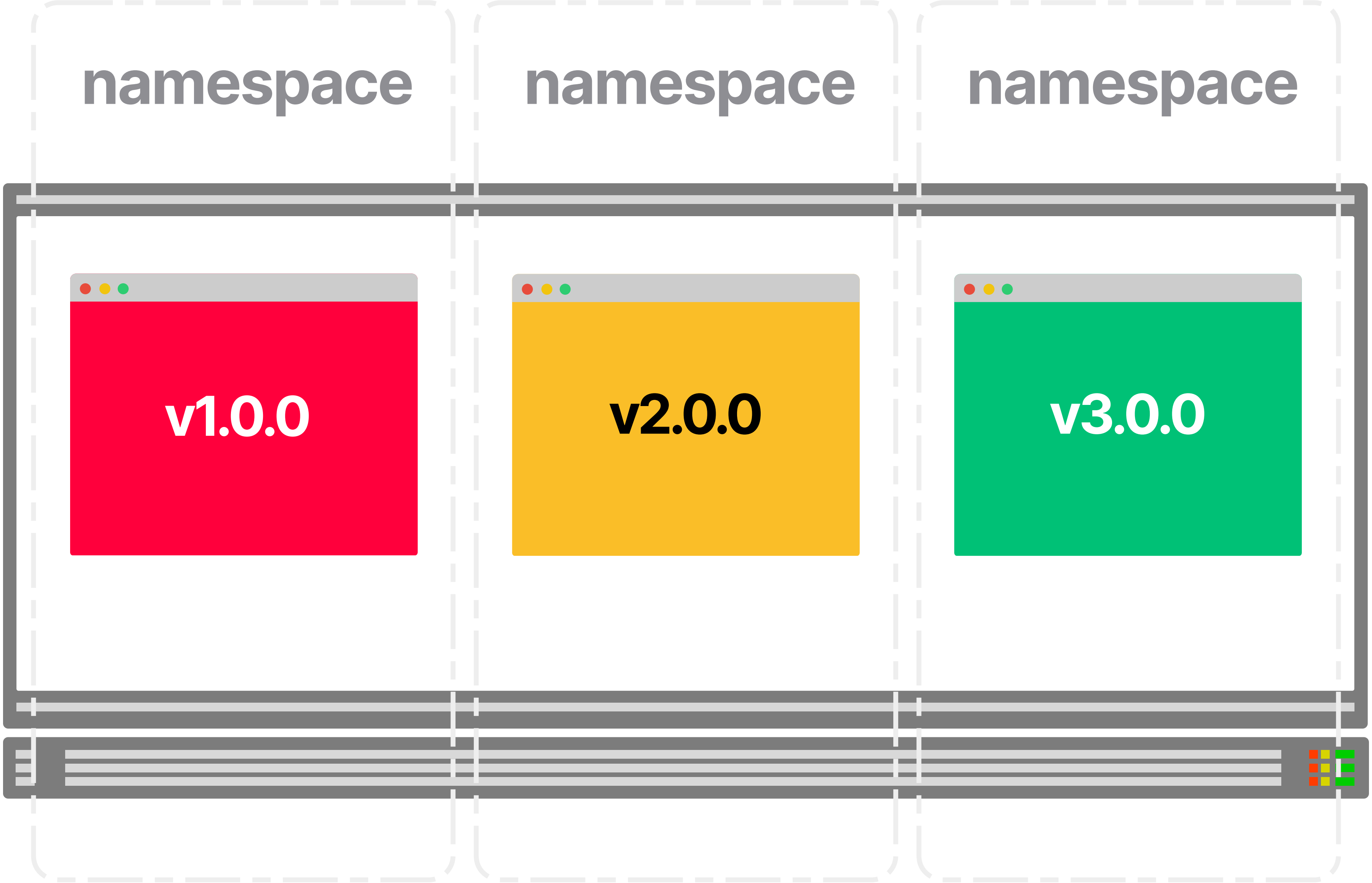
v1.0.0

namespace

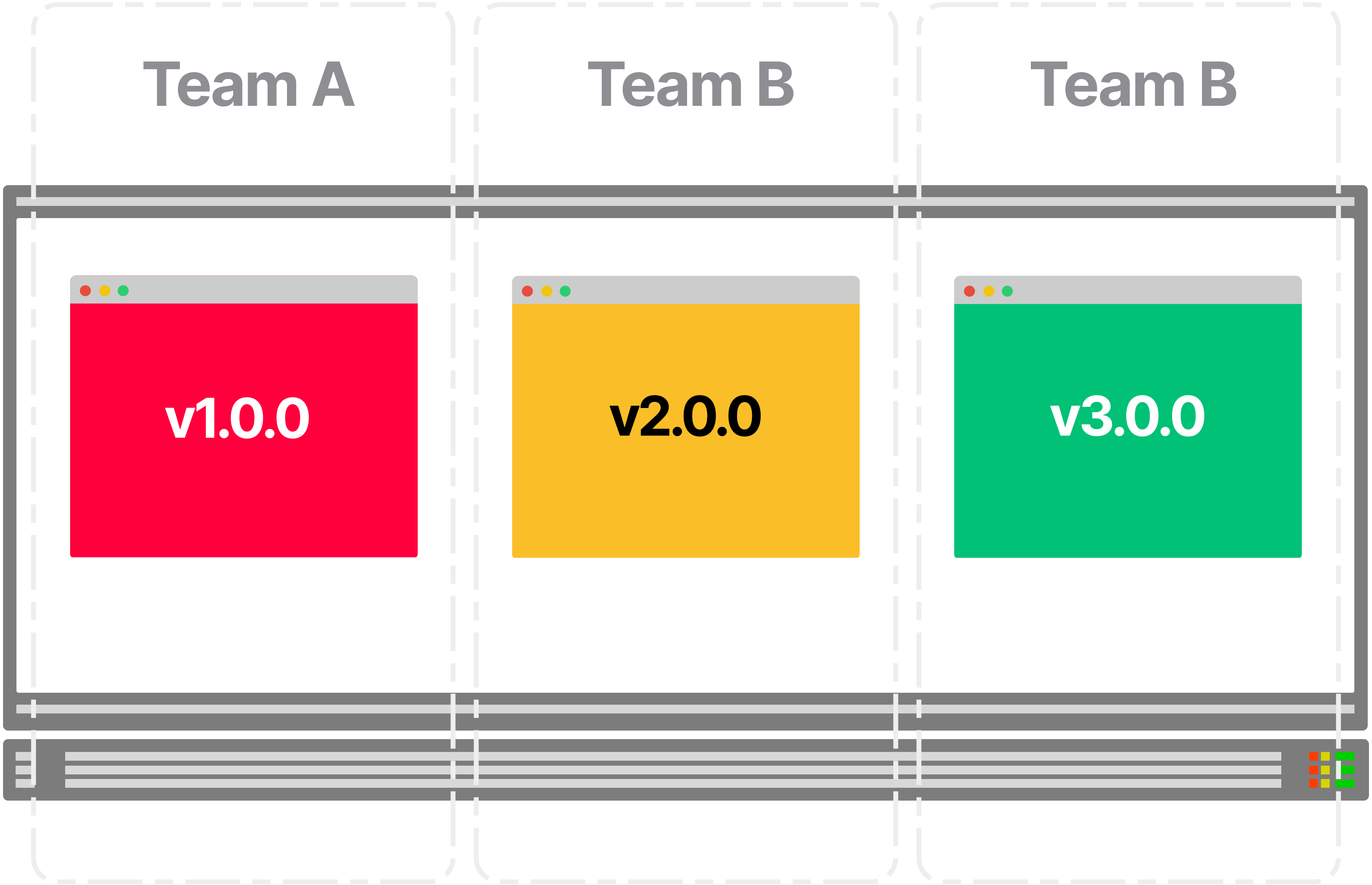
v2.0.0

namespace

v3.0.0


namespace
namespace
namespace
v1.0.0
v2.0.0
v3.0.0

# Environments x tenants



Team A

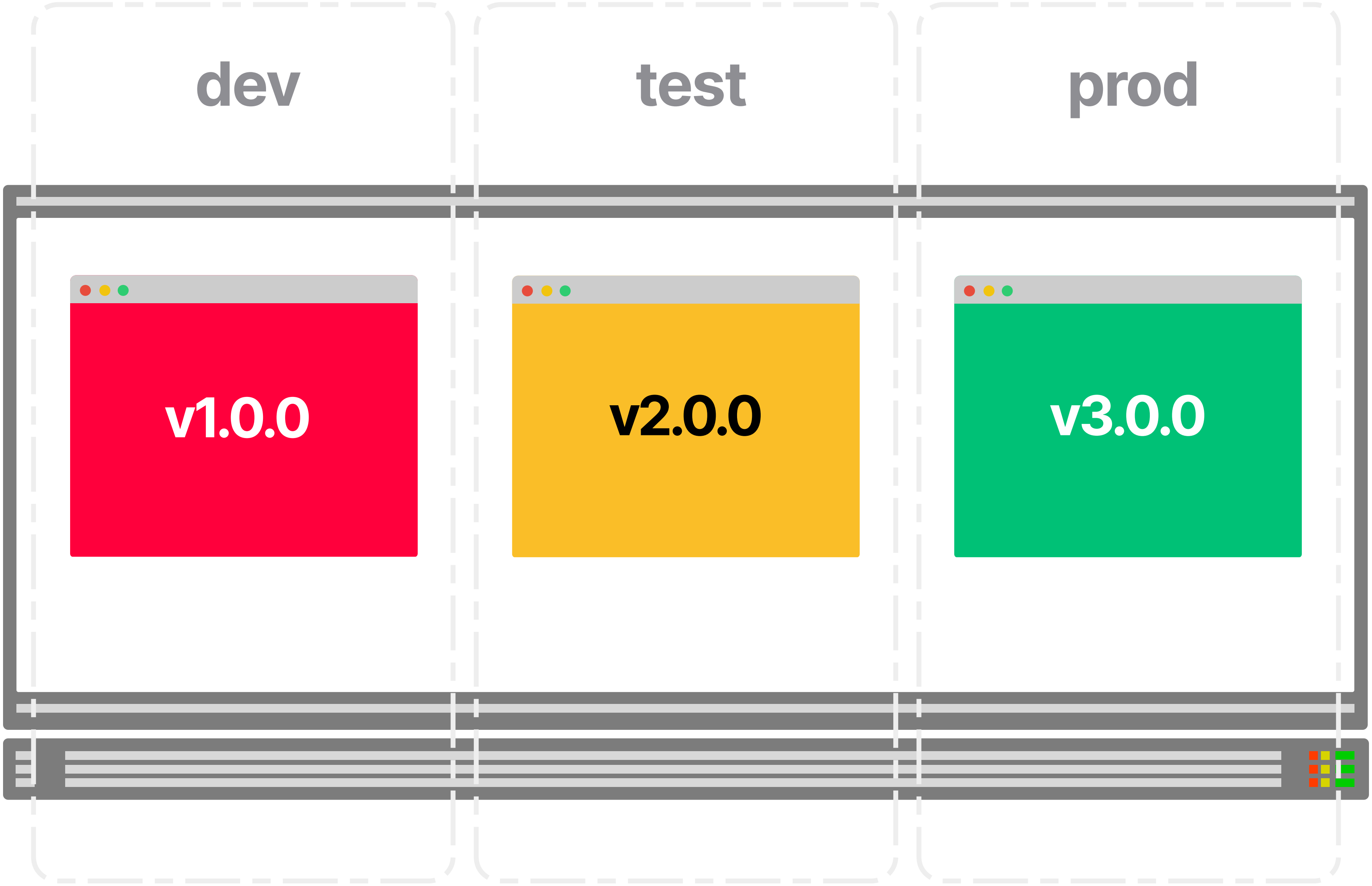
Team B

Team B

v1.0.0

v2.0.0

v3.0.0



dev

test

prod

v1.0.0

v2.0.0

v3.0.0

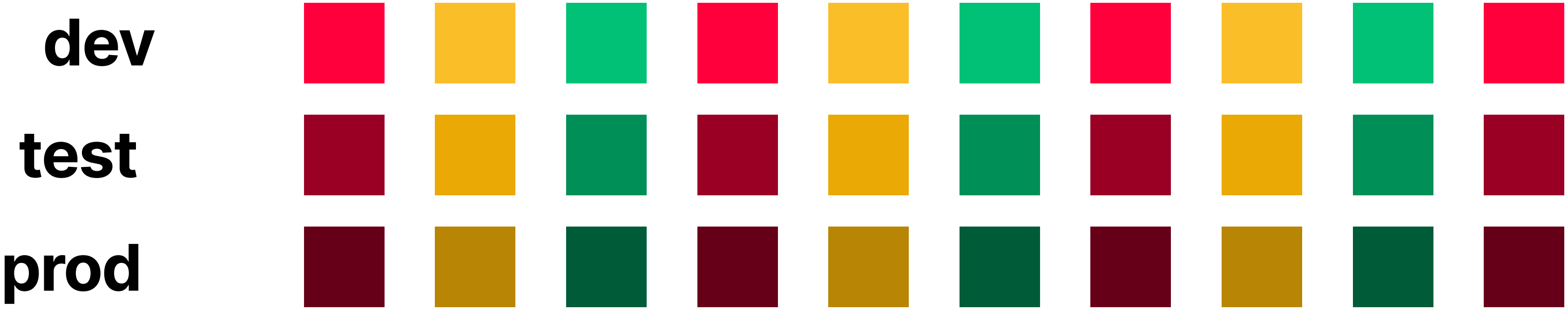
|  | Team A |
|---|---|
| dev | 🟥 |
| test | 🟥 |
| prod | 🟥 |

| | Team A | Team B |
|---|---|---|
| dev | 🟥 | 🟨 |
| test | 🟥 | 🟨 |
| prod | 🟥 | 🟨 |

|  | Team A | Team B | Team C |
|---|---|---|---|
| dev | 🟥 | 🟨 | 🟩 |
| test | 🟥 | 🟨 | 🟩 |
| prod | 🟥 | 🟨 | 🟩 |

# Environments x tenants *at scale*

10 TENANTS

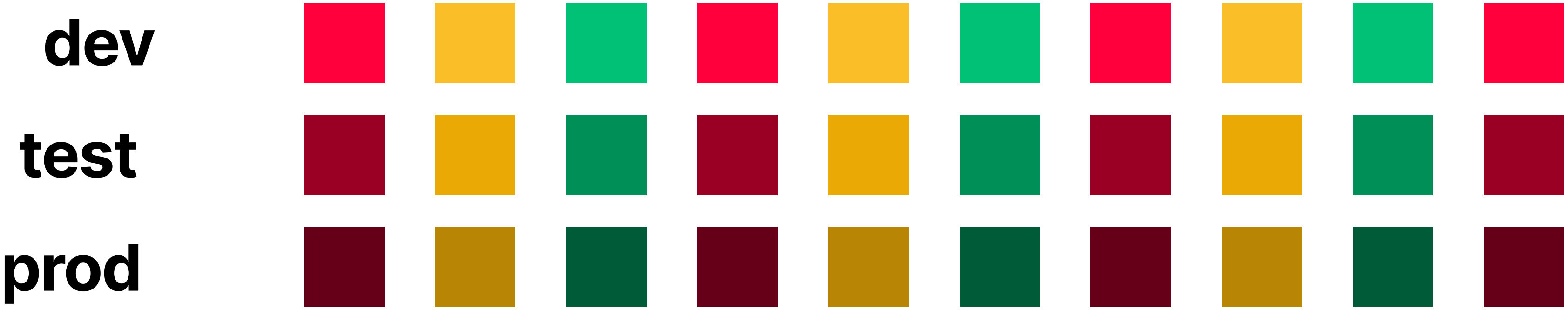


dev

test

prod

10 TENANTS

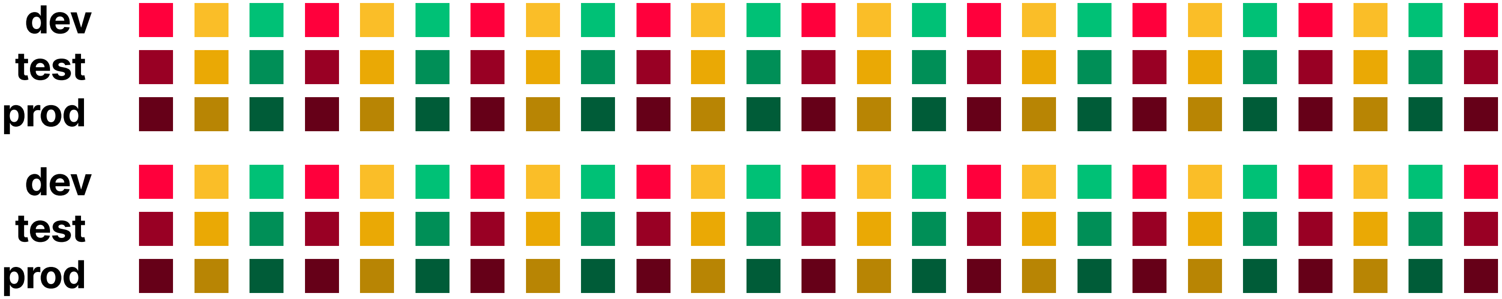


dev

test

prod

50 TENANTS



dev

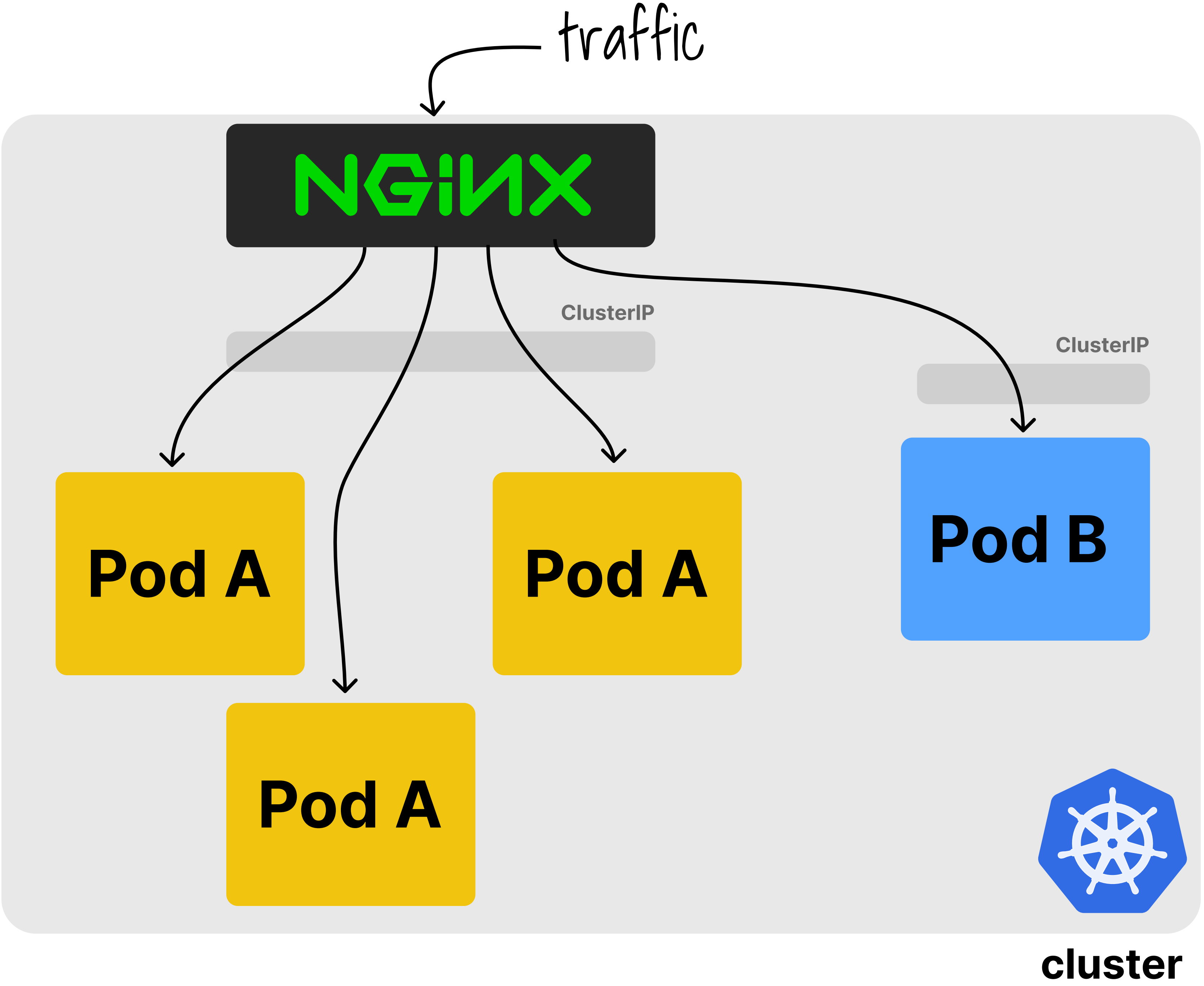
test

prod

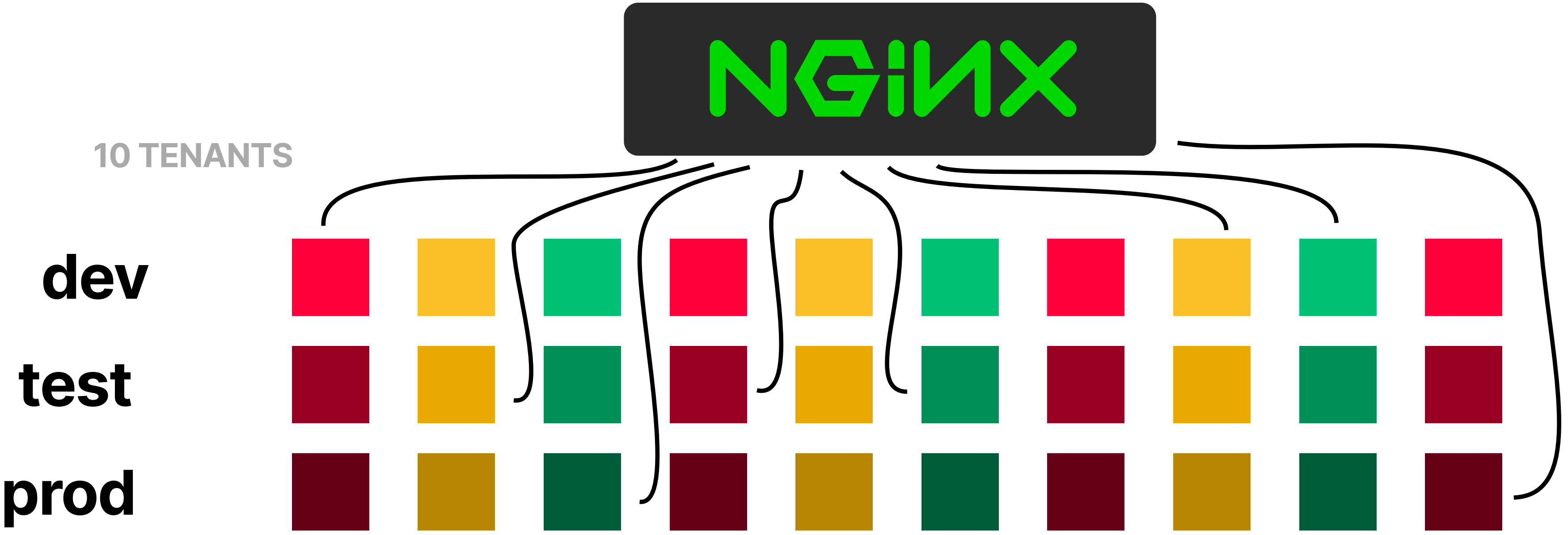
dev

test

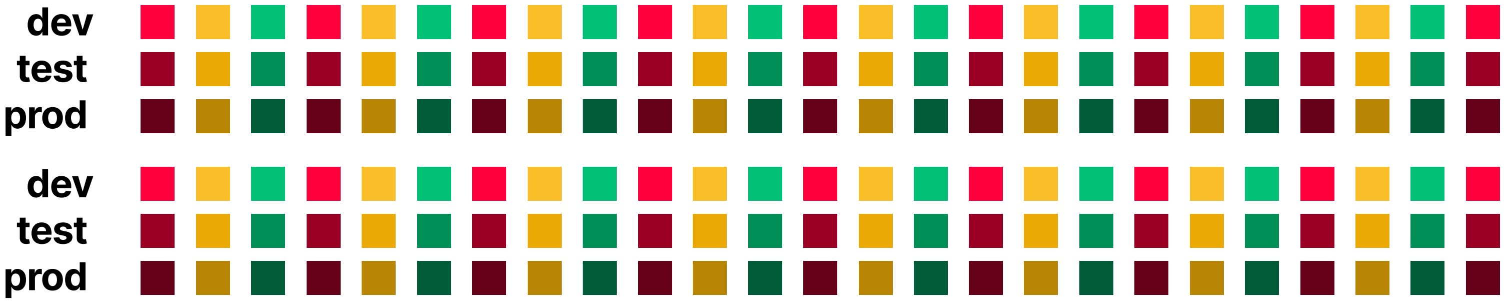
prod


traffic
NGINX
ClusterIP
ClusterIP
Pod A
Pod A
Pod B
Pod A
cluster

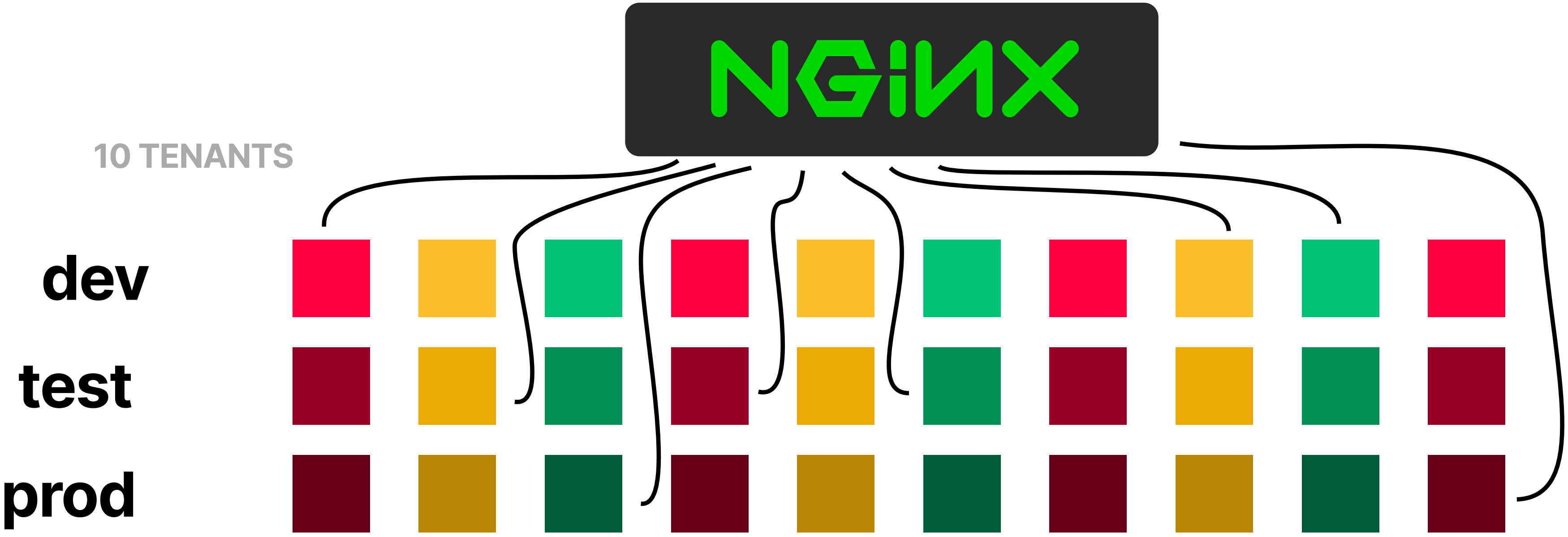
10 TENANTS



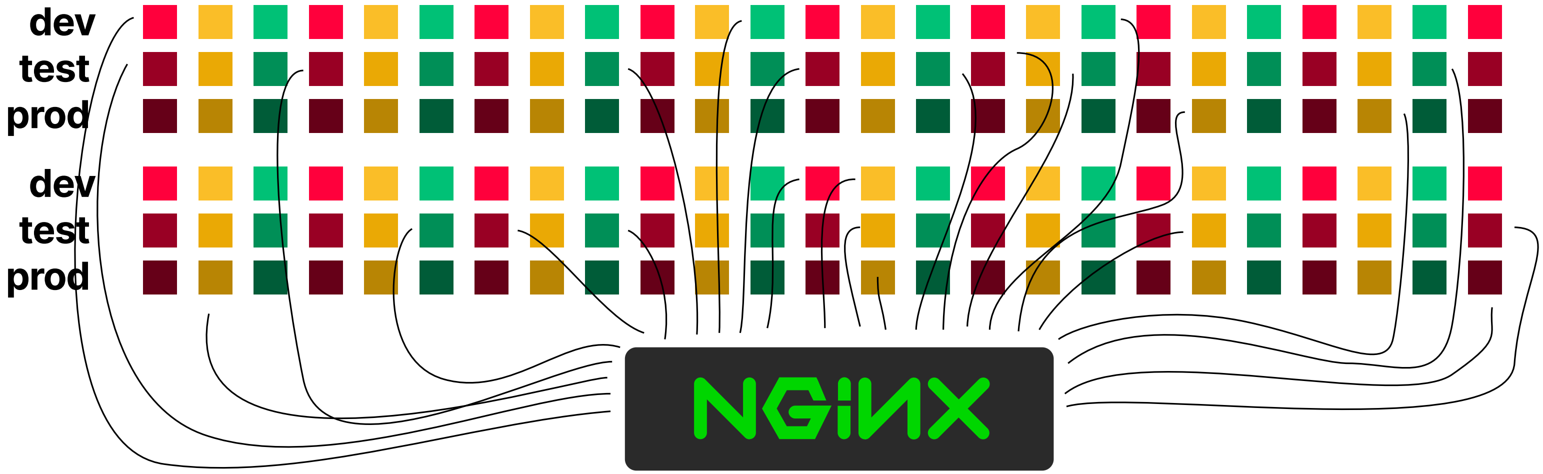
50 TENANTS



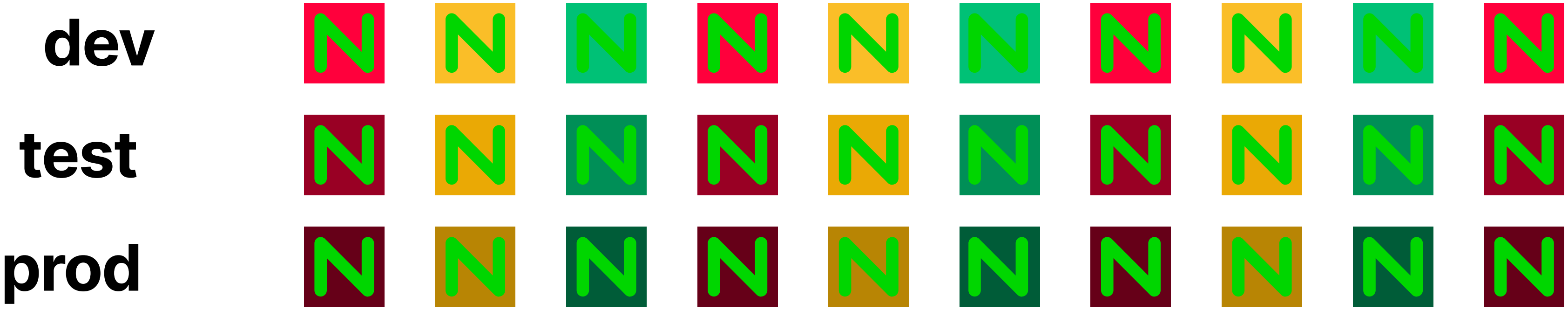
10 TENANTS



50 TENANTS

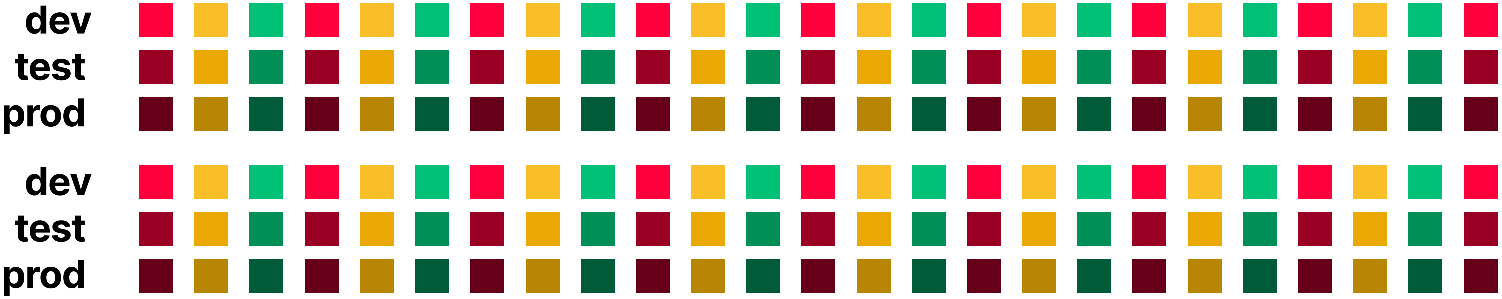


10 TENANTS



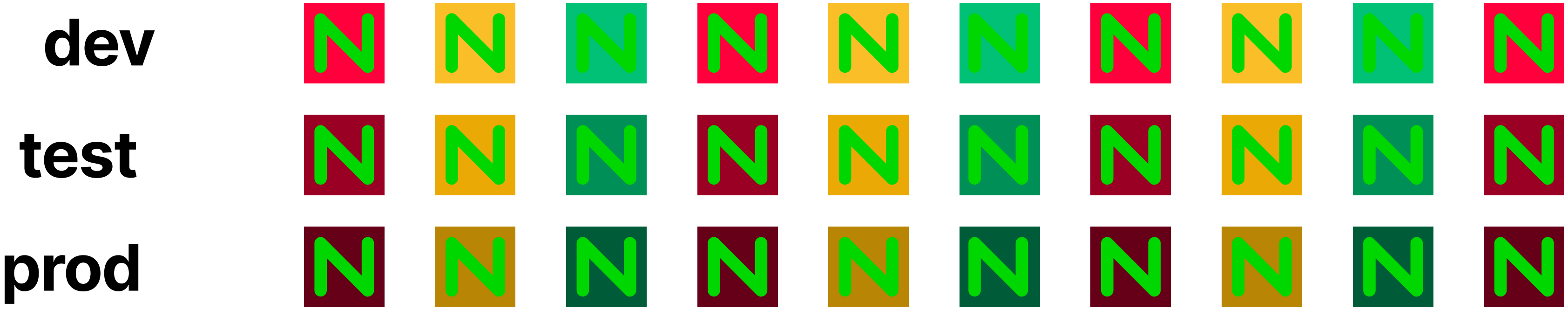
dev
test
prod

50 TENANTS

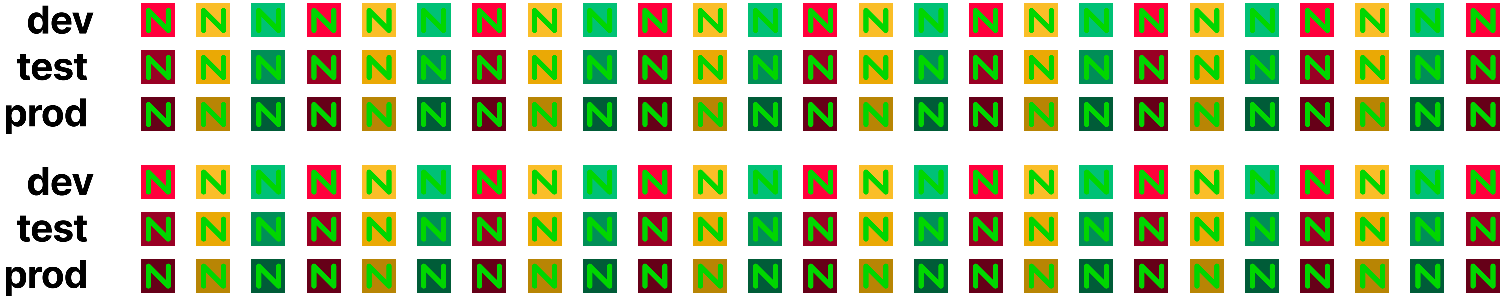


dev
test
prod
dev
test
prod

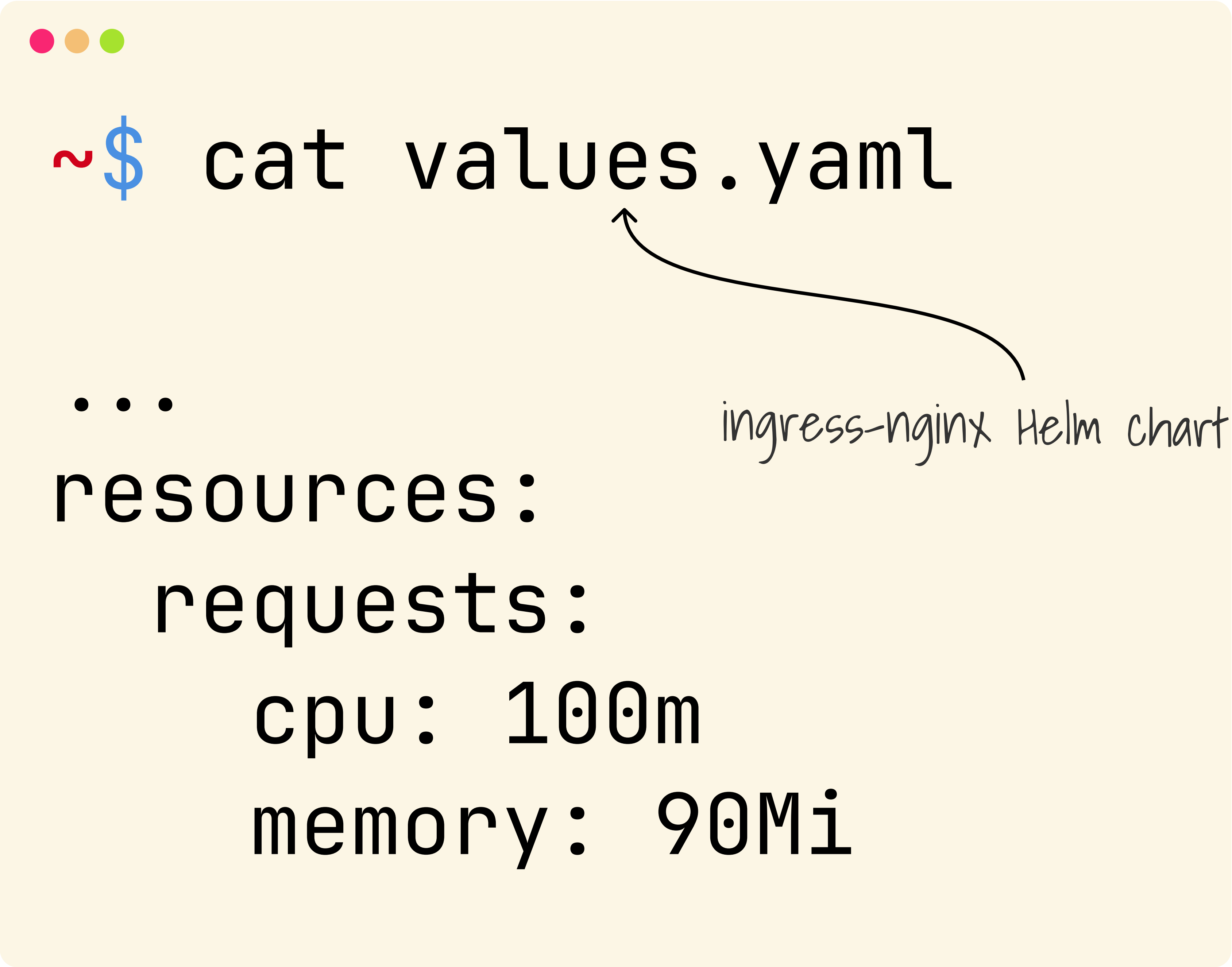
## 10 TENANTS



## 50 TENANTS



# 1 vs many: resources



```
~$ cat values.yaml

...
resources:
  requests:
    cpu: 100m
    memory: 90Mi
```

ingress-nginx Helm chart

# Single Ingress

CPU

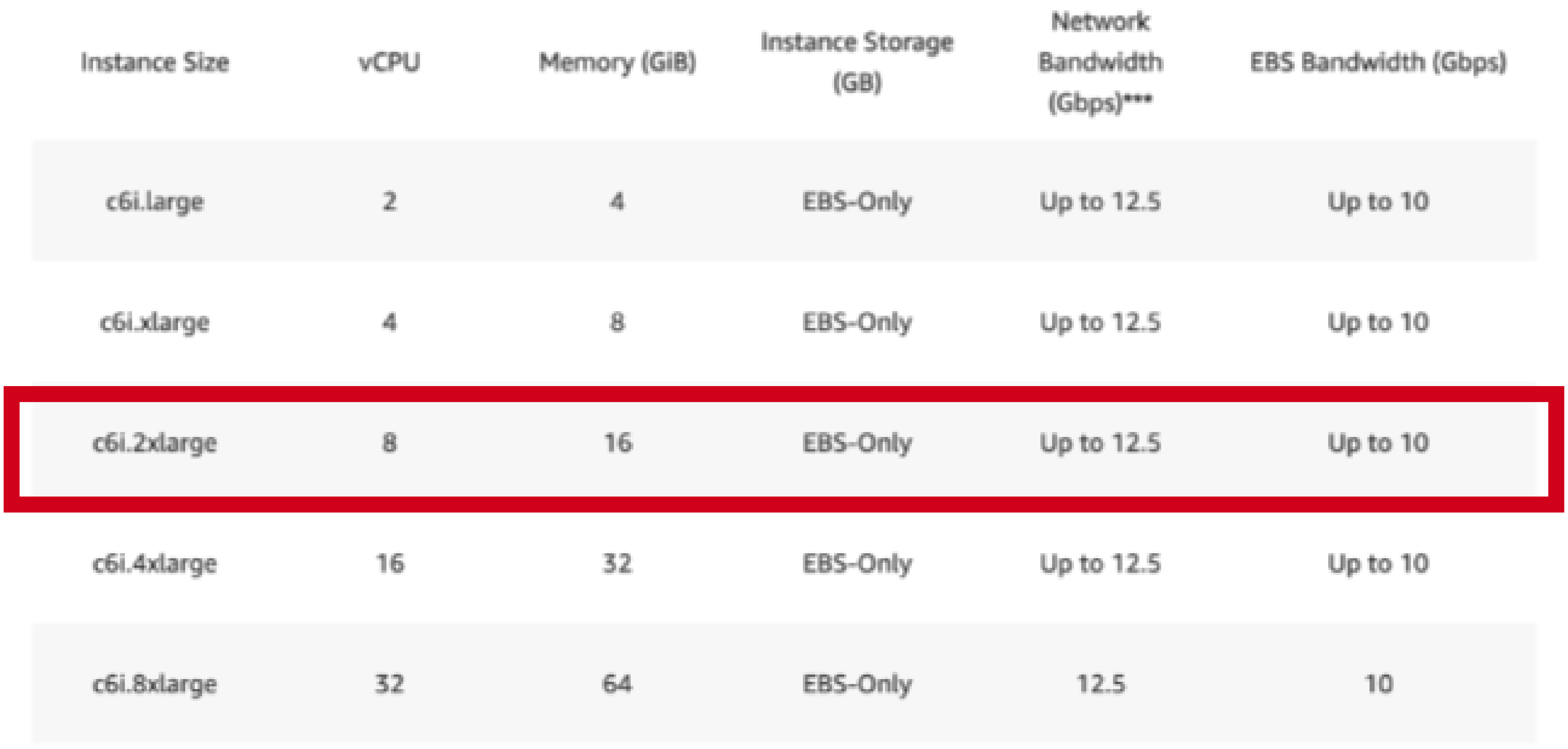
**100m**

MEMORY

**90Mi**

| Single Ingress | 10 × 3 |
|---|---|
| CPU<br>100m | CPU<br>3vCPU |
| MEMORY<br>90Mi | MEMORY<br>2.7GB |

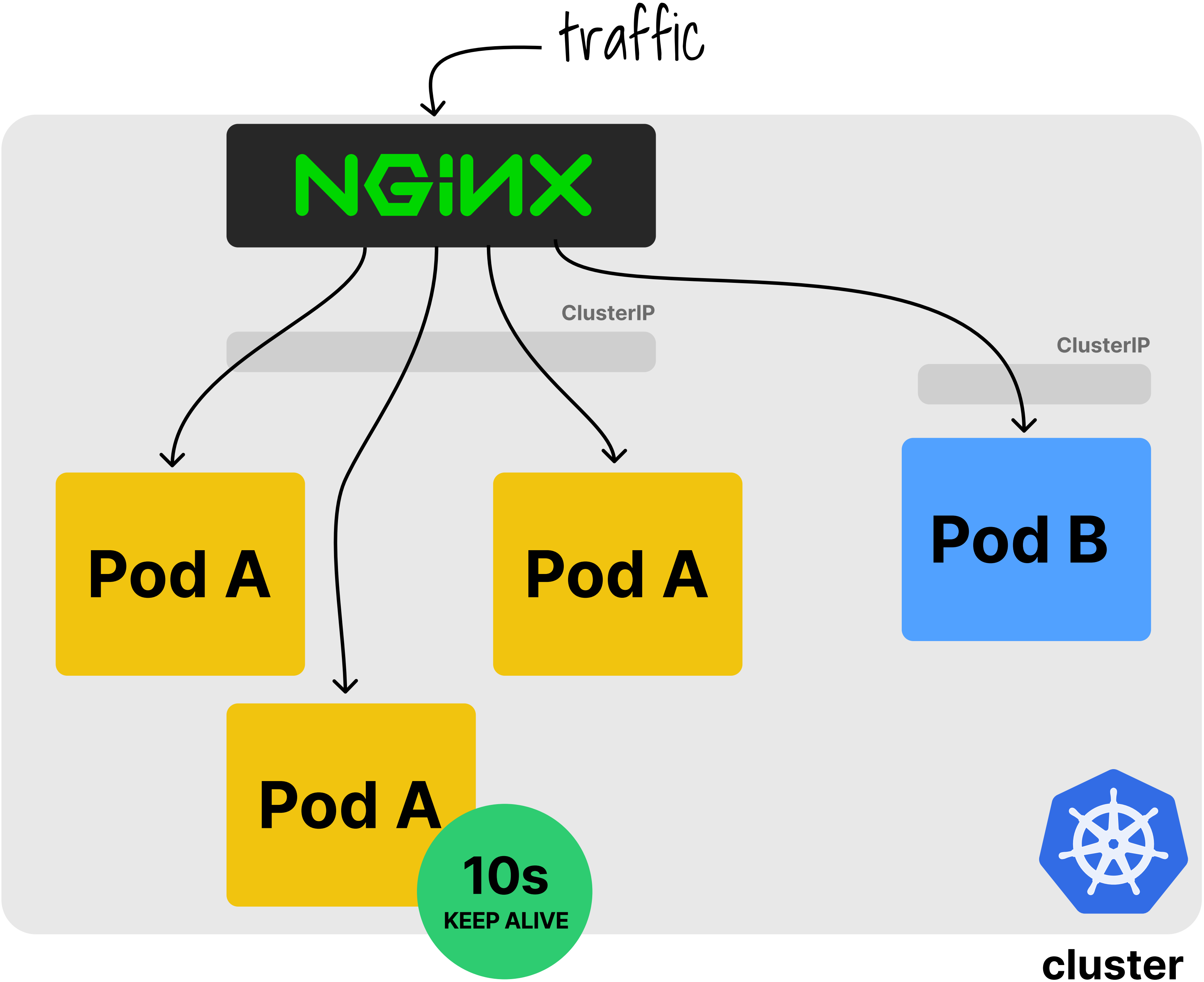
| Single Ingress | 10 × 3 | 50 × 3 |
|---|---|---|
| CPU | CPU | CPU |
| 100m | 3vCPU | 5vCPU |
| MEMORY | MEMORY | MEMORY |
| 90Mi | 2.7GB | 4.5GB |

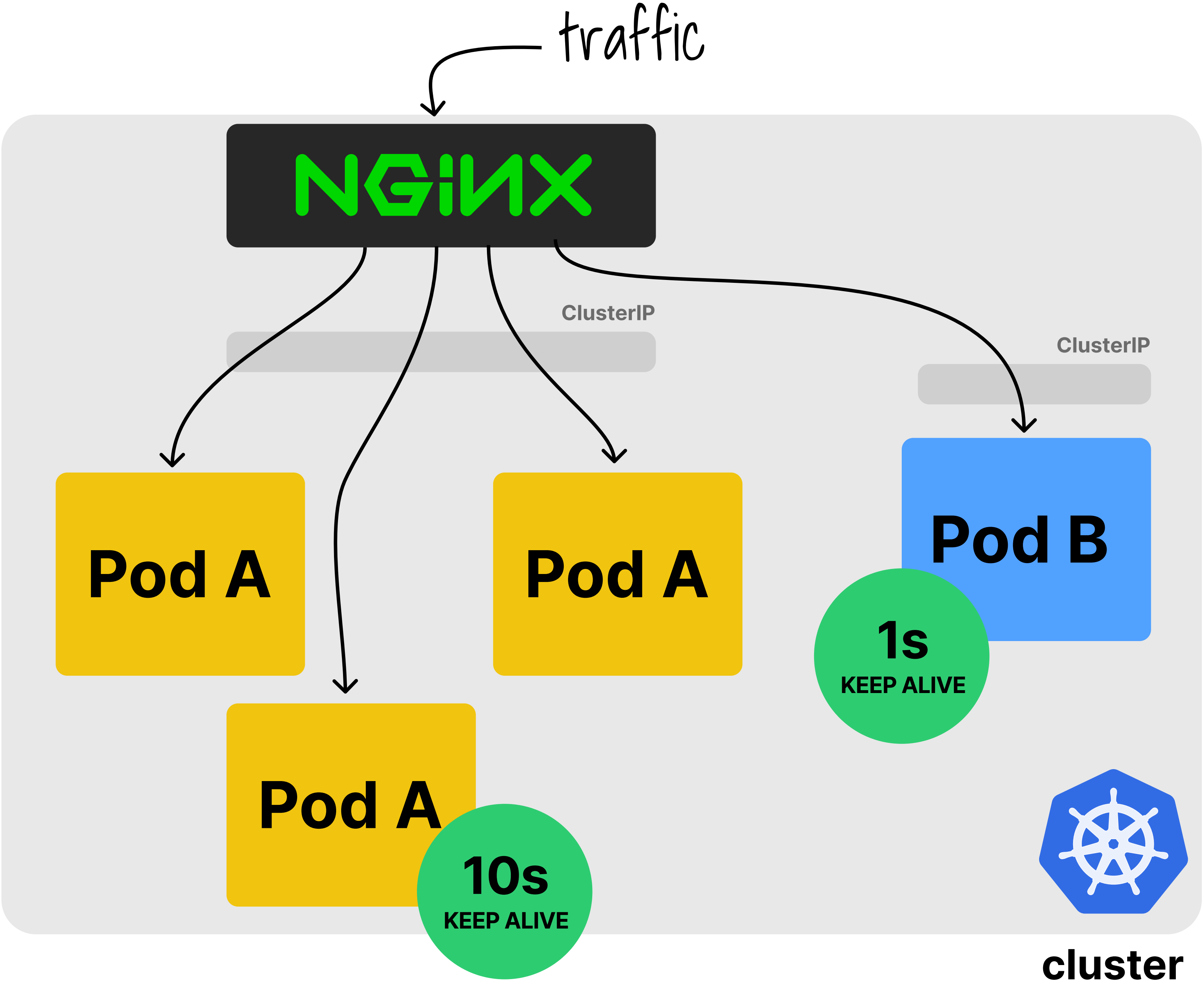


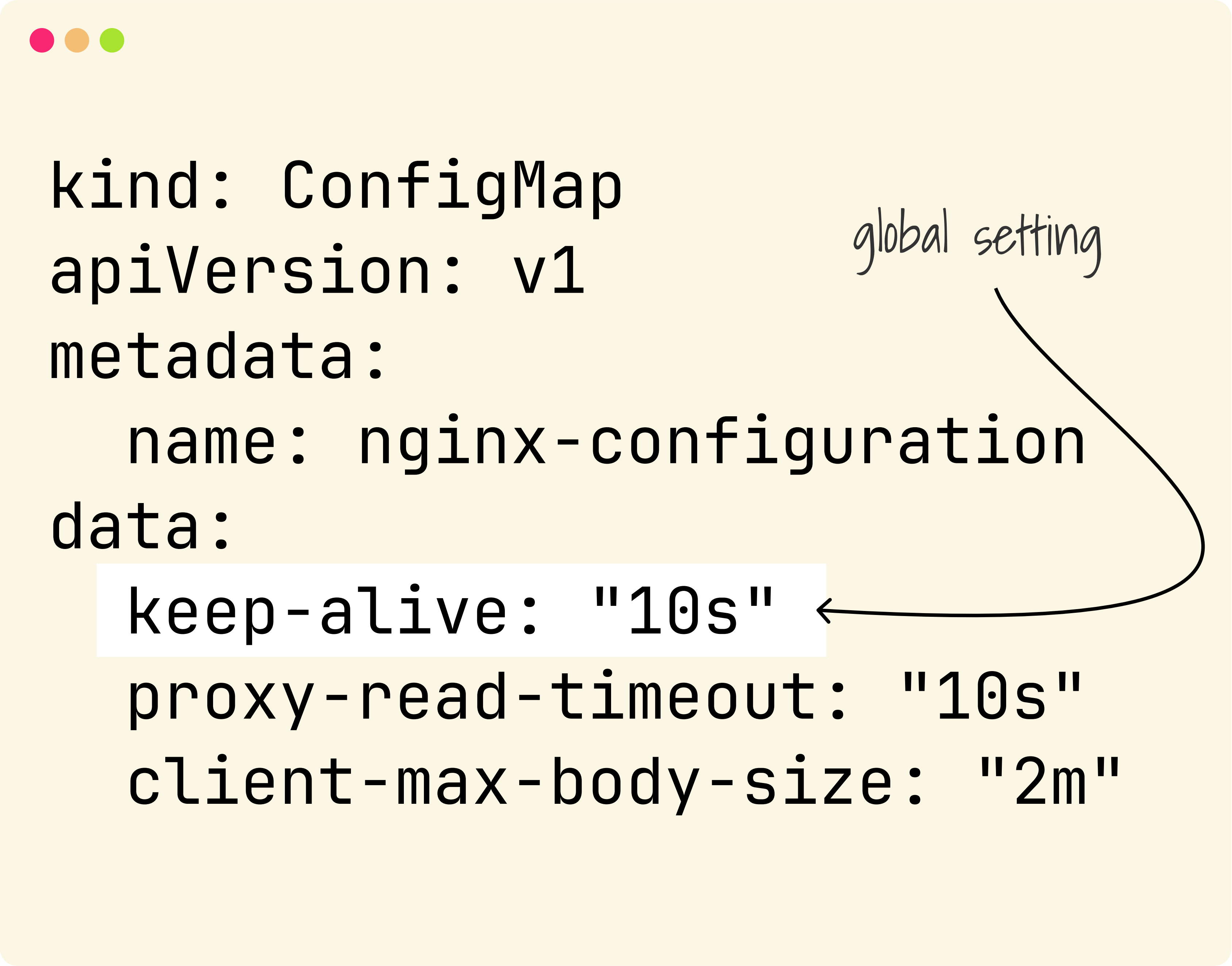
| Instance Size | vCPU | Memory (GiB) | Instance Storage (GB) | Network Bandwidth (Gbps)*** | EBS Bandwidth (Gbps) |
|---|---|---|---|---|---|
| c6i.large | 2 | 4 | EBS-Only | Up to 12.5 | Up to 10 |
| c6i.xlarge | 4 | 8 | EBS-Only | Up to 12.5 | Up to 10 |
| c6i.2xlarge | 8 | 16 | EBS-Only | Up to 12.5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

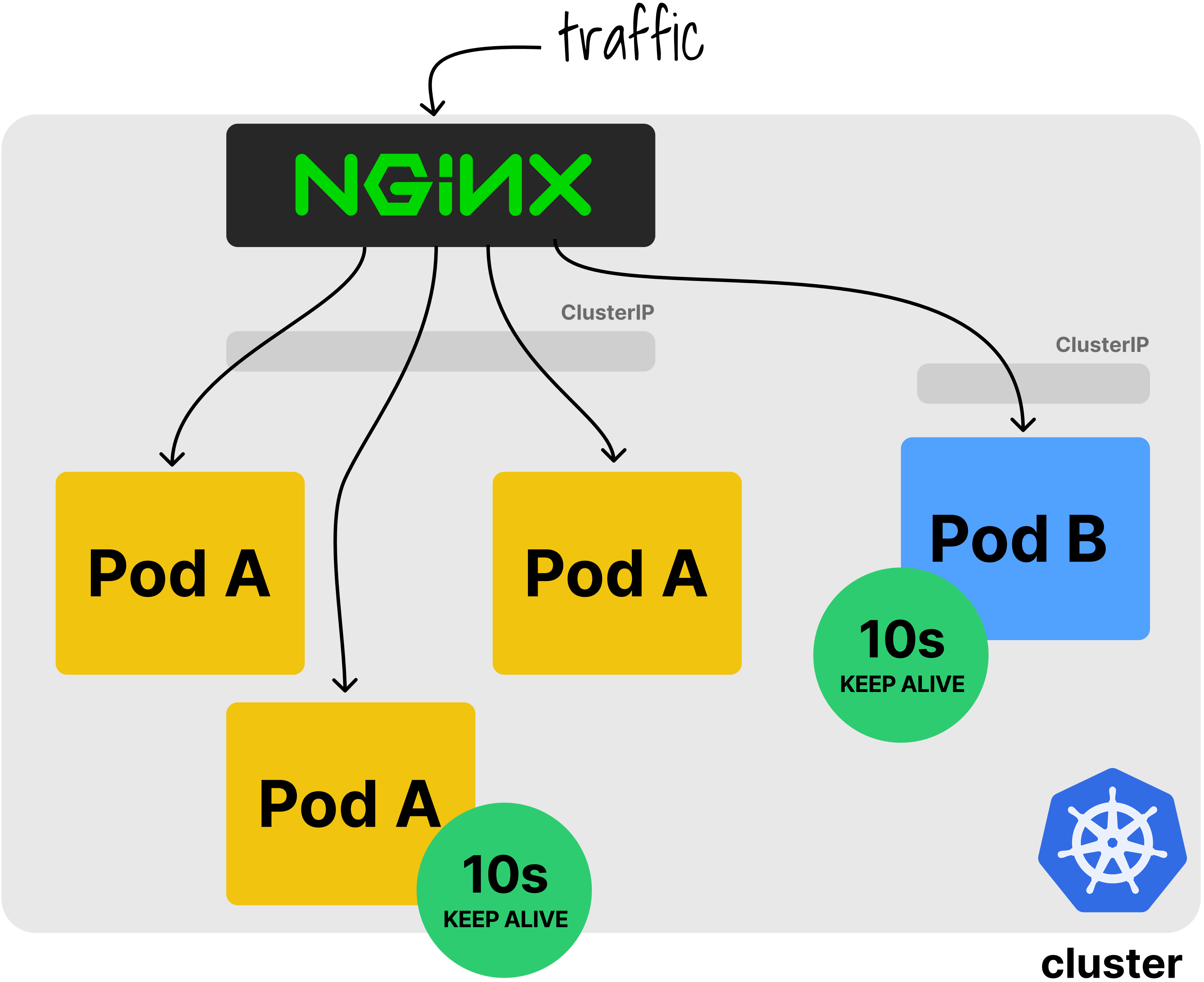

traffic
NGINX
ClusterIP
ClusterIP
Pod A
Pod A
Pod B
Pod A
10s
KEEP ALIVE
cluster


traffic
NGINX
ClusterIP
ClusterIP
Pod A
Pod A
Pod B
1s
KEEP ALIVE
Pod A
10s
KEEP ALIVE
cluster



```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
data:
  keep-alive: "10s"
  proxy-read-timeout: "10s"
  client-max-body-size: "2m"
```

global setting



traffic

NGINX

ClusterIP

ClusterIP

Pod A

Pod A

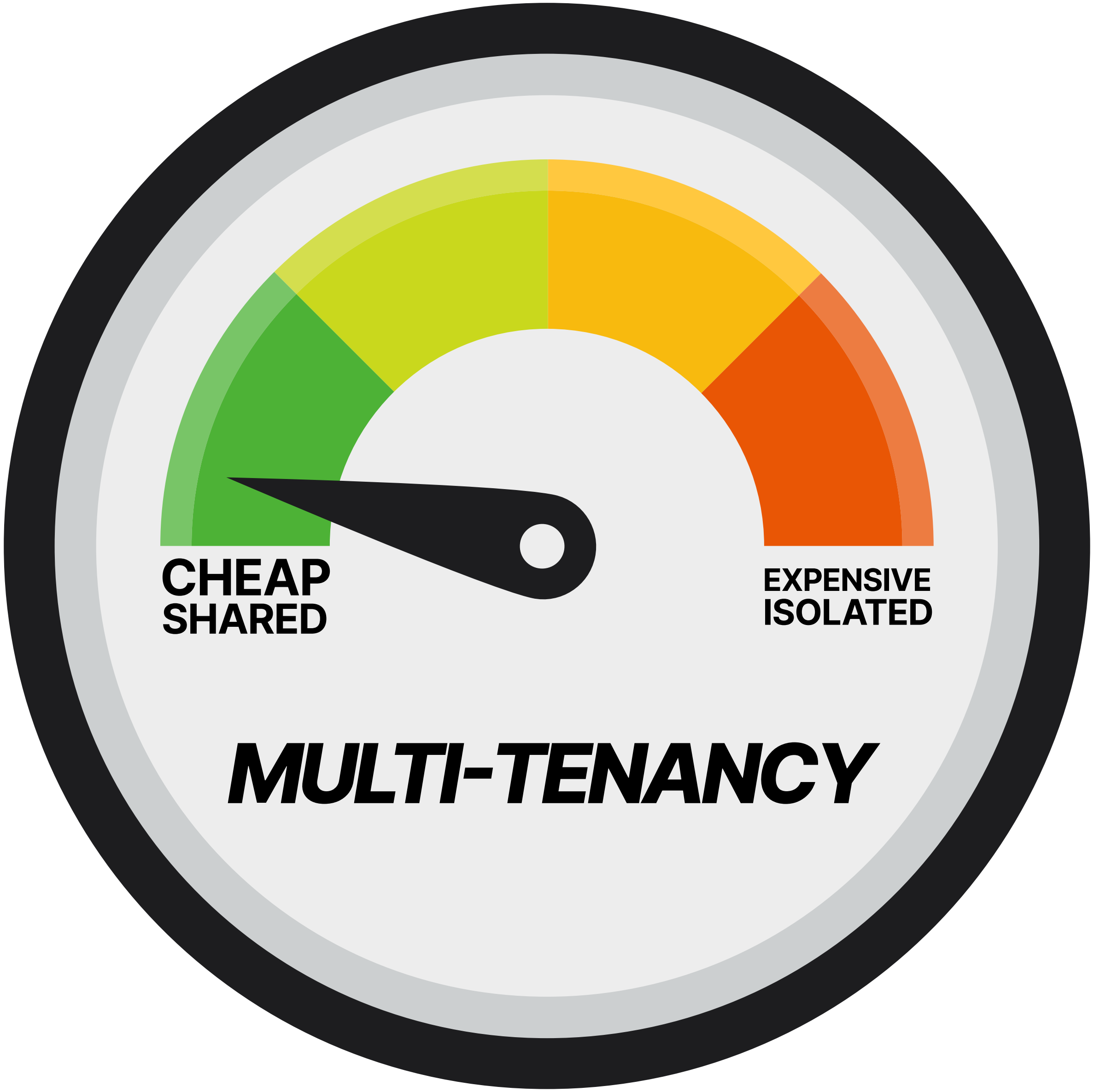
Pod B

10s
KEEP ALIVE

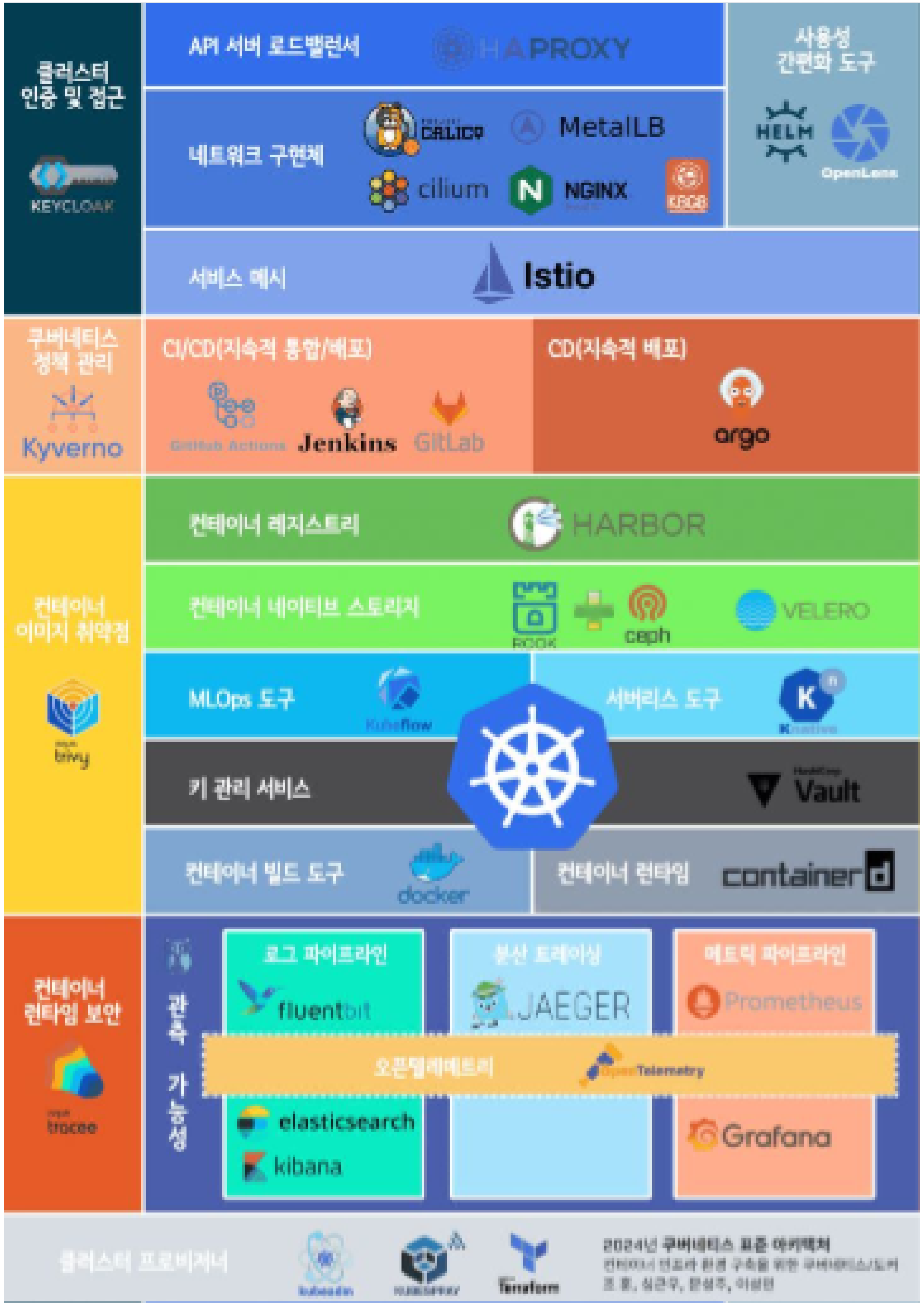
Pod A

10s
KEEP ALIVE

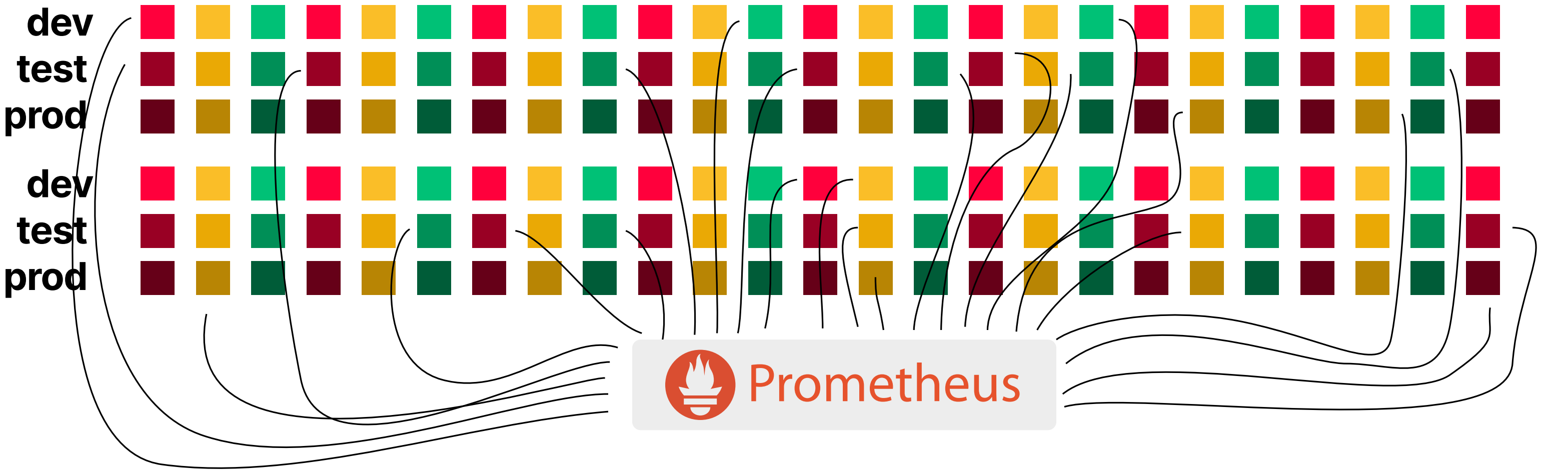
cluster

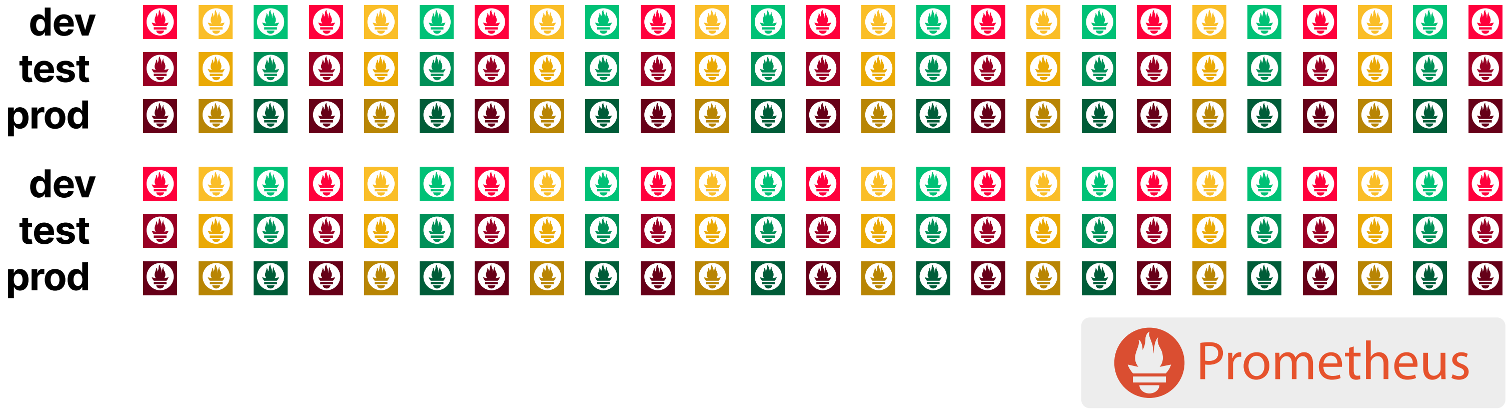

CHEAP
SHARED
EXPENSIVE
ISOLATED
MULTI-TENANCY

# Kubernetes platform tools


클러스터 인증 및 접근
KEYCLOAK
API 서버 로드밸런서
HAPROXY
사용성 간편화 도구
HELM
OpenLens
네트워크 구현체
CALICO
MetalLB
cilium
NGINX
KGCB
서비스 메시
Istio
쿠버네티스 정책 관리
Kyverno
CI/CD(지속적 통합/배포)
GitHub Actions
Jenkins
GitLab
CD(지속적 배포)
argo
컨테이너 이미지 취약점
trivy
컨테이너 레지스트리
HARBOR
컨테이너 네이티브 스토리지
ROOK
ceph
VELERO
MLOps 도구
Kubeflow
서버리스 도구
Knative
키 관리 서비스
HashiCorp Vault
컨테이너 빌드 도구
docker
컨테이너 런타임
containerd
컨테이너 런타임 보안
tracee
관측 가능성
로그 파이프라인
fluentbit
elasticsearch
kibana
분산 트레이싱
JAEGER
메트릭 파이프라인
Prometheus
Grafana
오픈텔레메트리
OpenTelemetry
클러스터 프로비저너
kubeadm
KUBESPRAY
Terraform
2024년 쿠버네티스 표준 아키텍처
컨테이너 인프라 환경 구축을 위한 쿠버네티스/도커
조 훈, 심근우, 문성주, 이성민

HOON JO

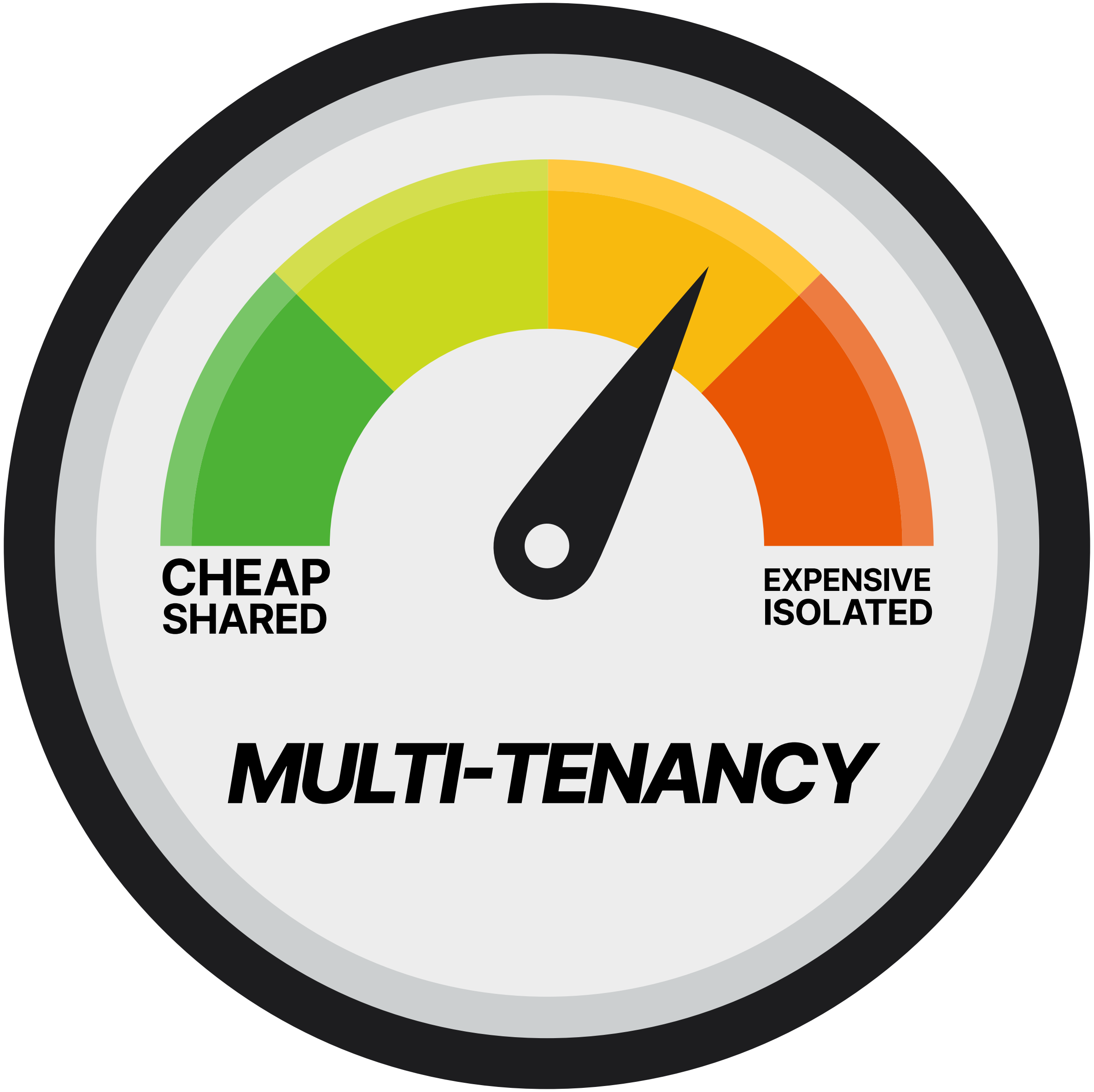

dev
test
prod
dev
test
prod
Prometheus



dev
test
prod

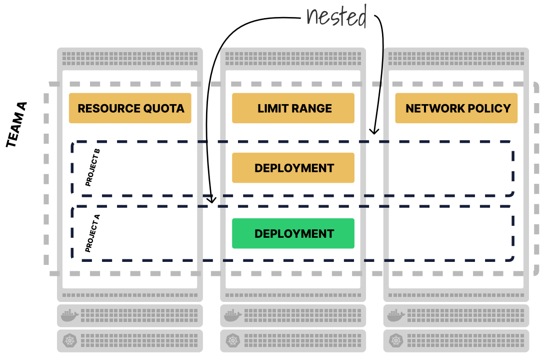
dev
test
prod

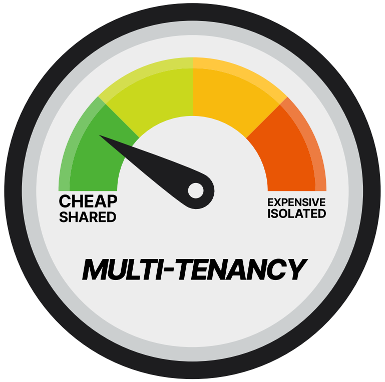
Prometheus


CHEAP
SHARED
EXPENSIVE
ISOLATED
MULTI-TENANCY

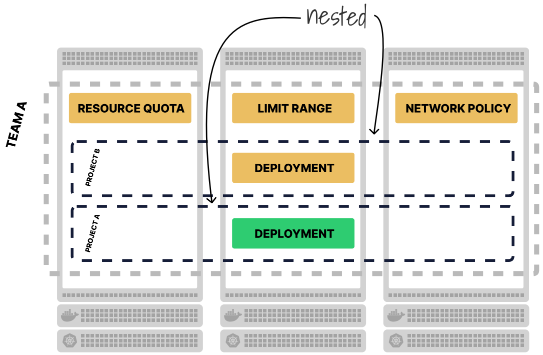
# Comparing multi-tenancy tools

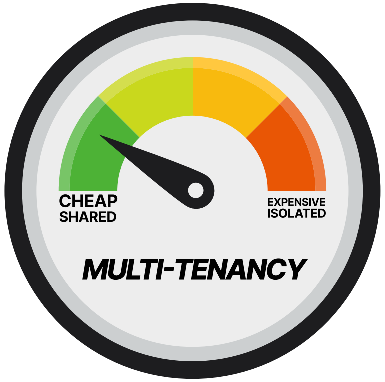
# Hierarchial Namespace Controller



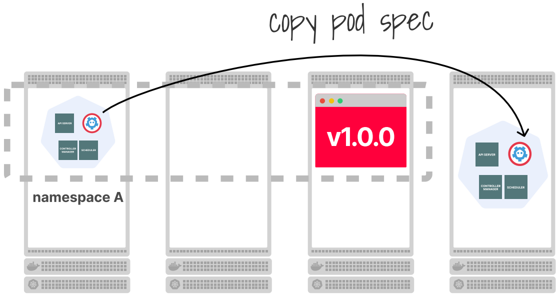


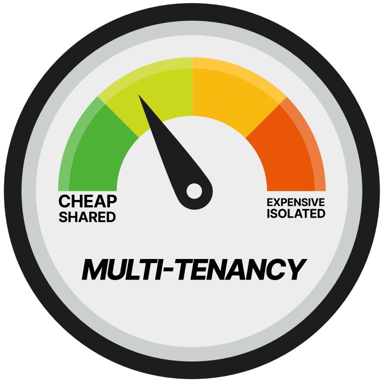
# Hierarchial Namespace Controller



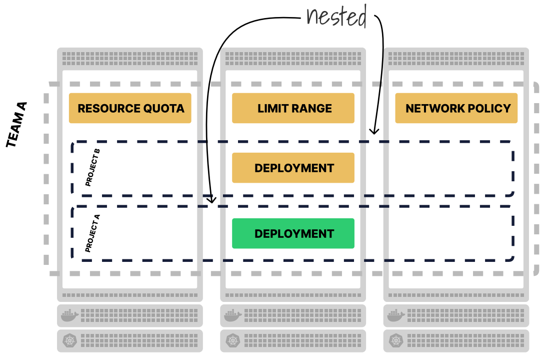


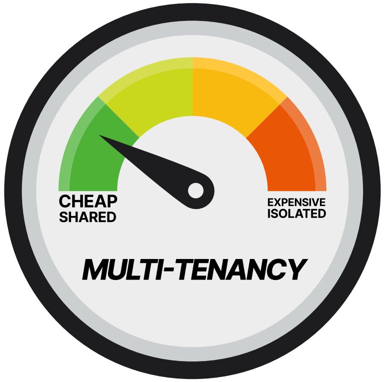
# vCluster



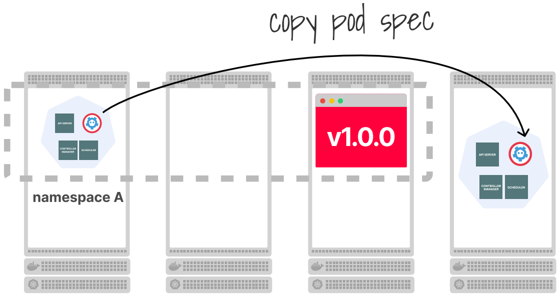


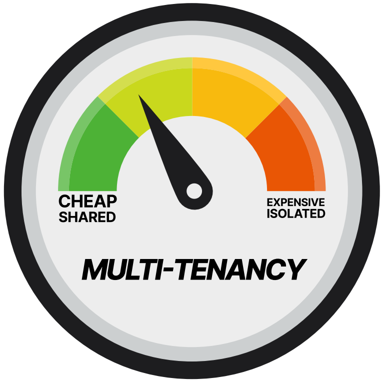
## Hierarchial Namespace Controller



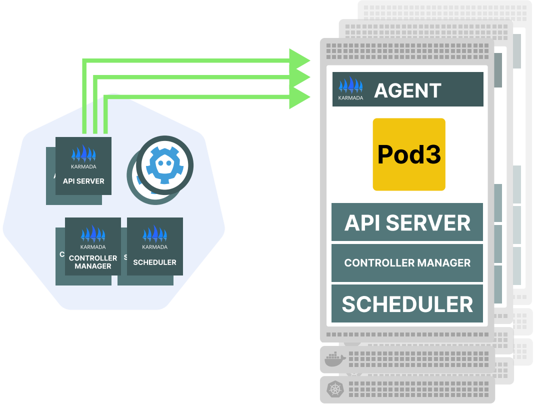


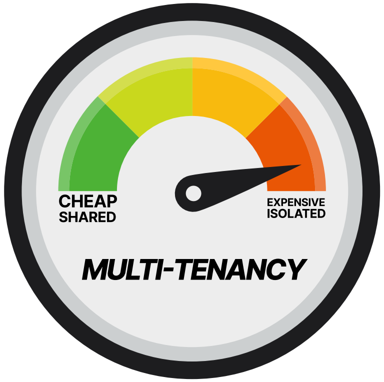
## vCluster





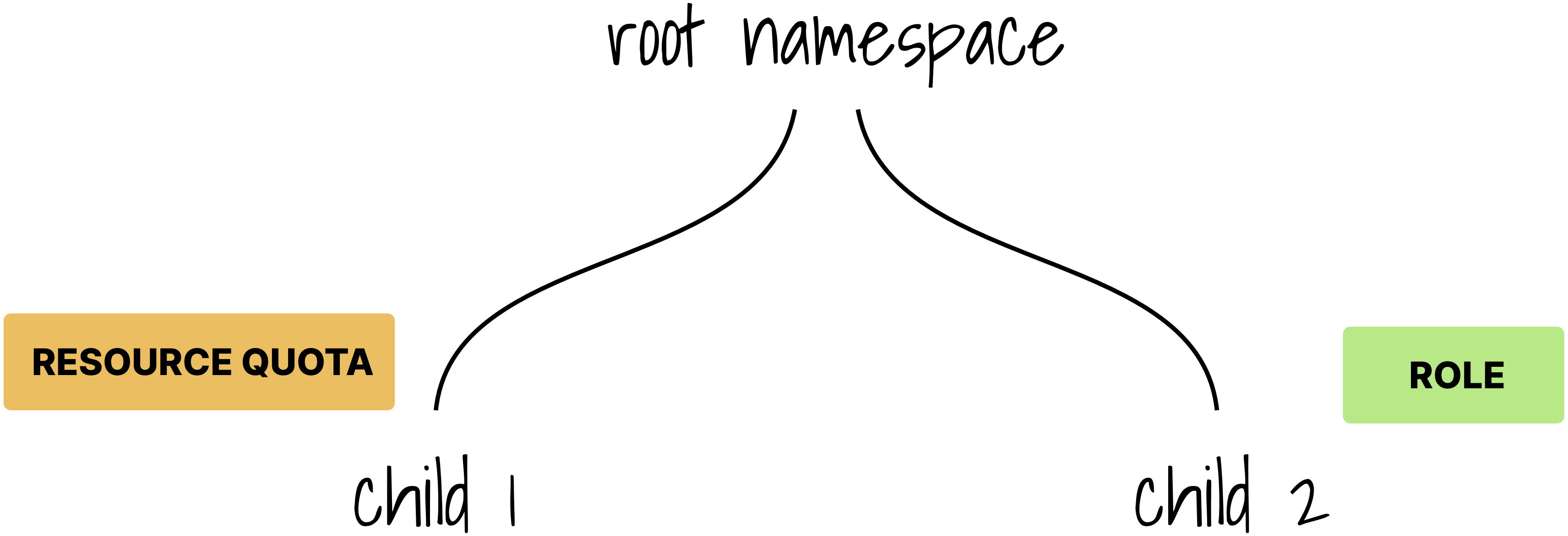
## Karmada





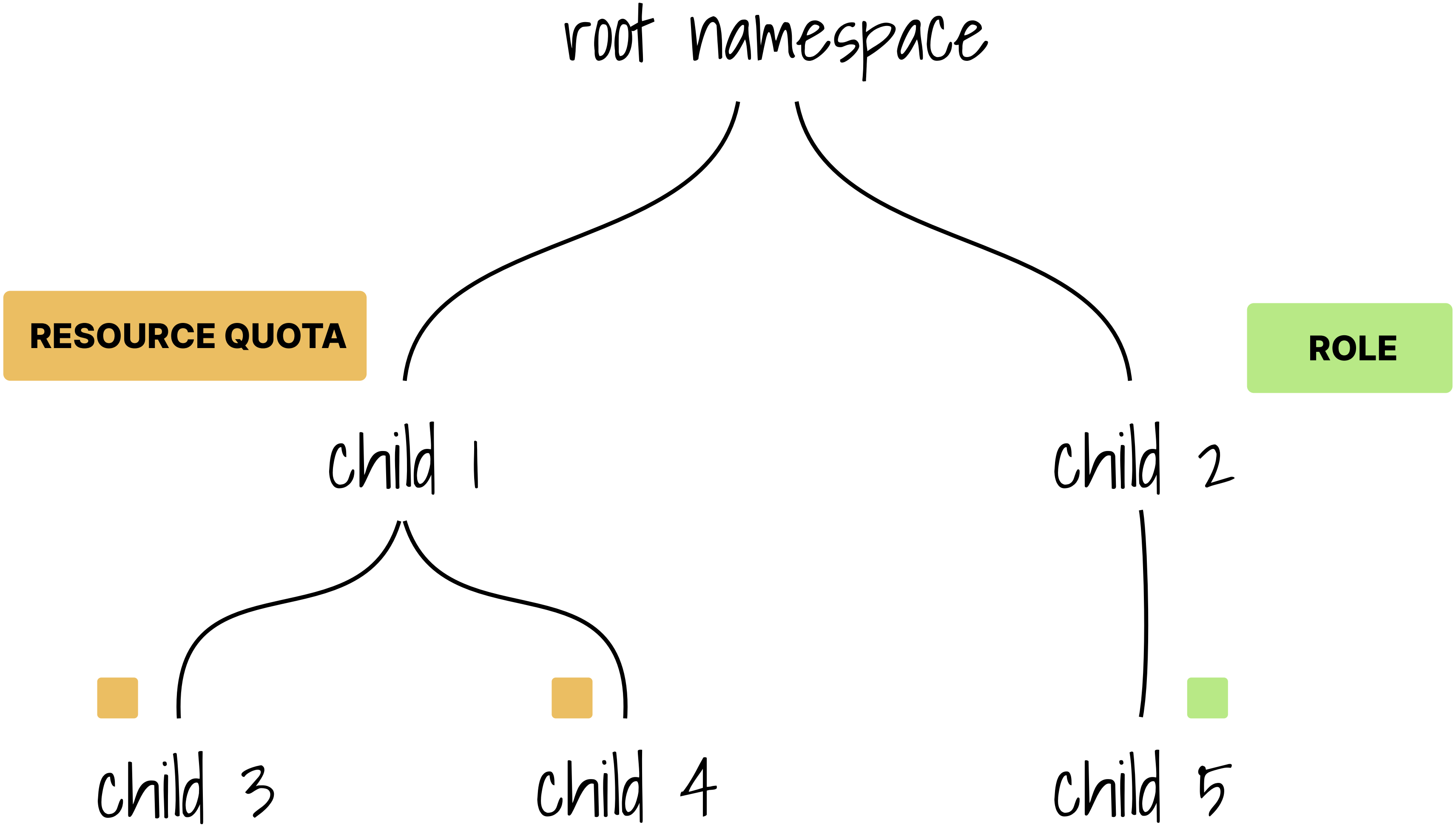
# Hierarchical Namespace Controller

# root namespace



RESOURCE QUOTA

ROLE



RESOURCE QUOTA

ROLE

# Demo

## Hierarchial Namespace Controller

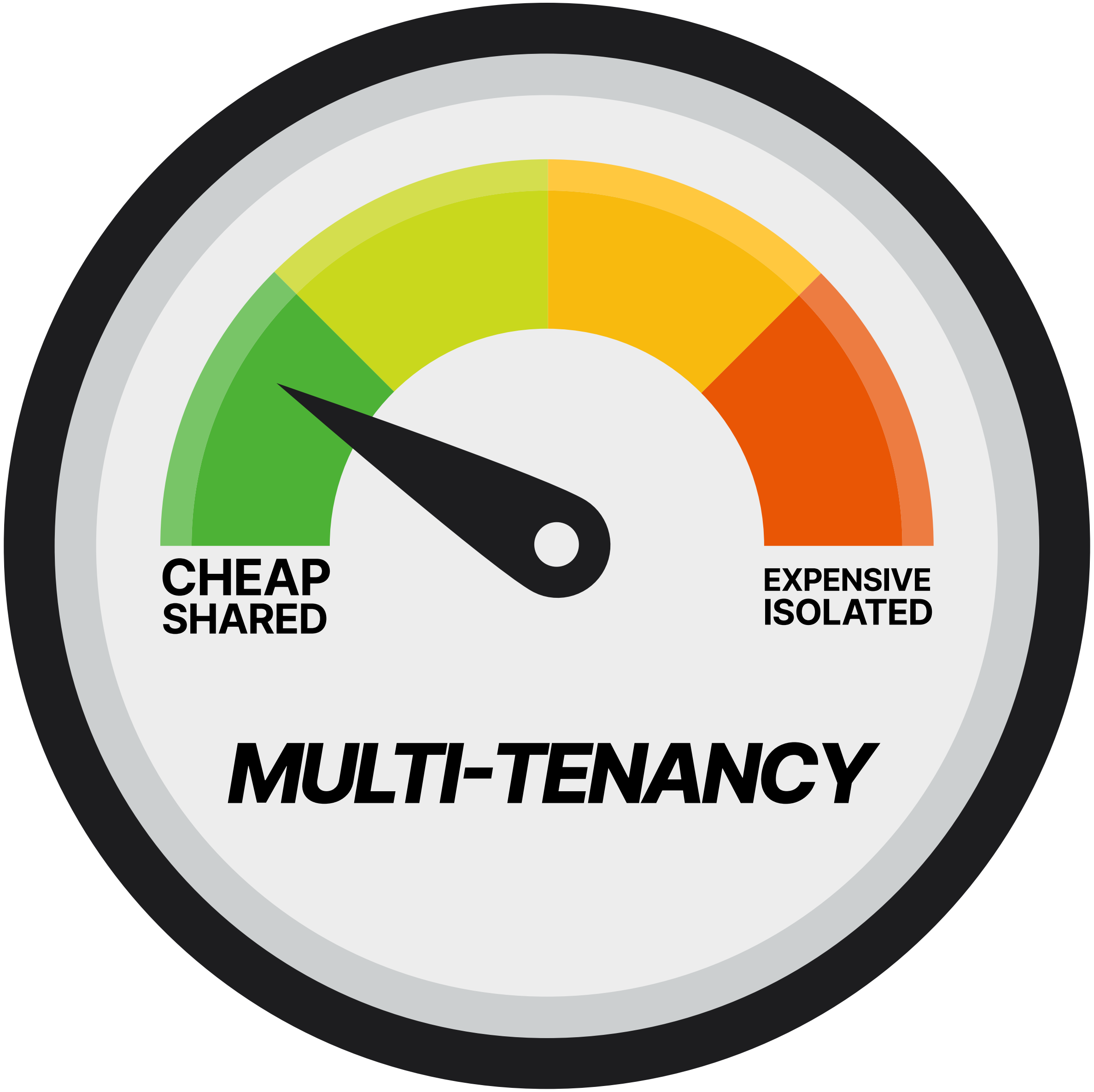
# “Nested” namespaces

## Hierarchial Namespace Controller

# “Nested” namespaces
# Single controller

# Hierarchial Namespace Controller

## “Nested” namespaces
## Single controller
## Regular namespaces


CHEAP
SHARED
EXPENSIVE
ISOLATED
MULTI-TENANCY

## COSTS FOR 50 TENANTS

# ~$0

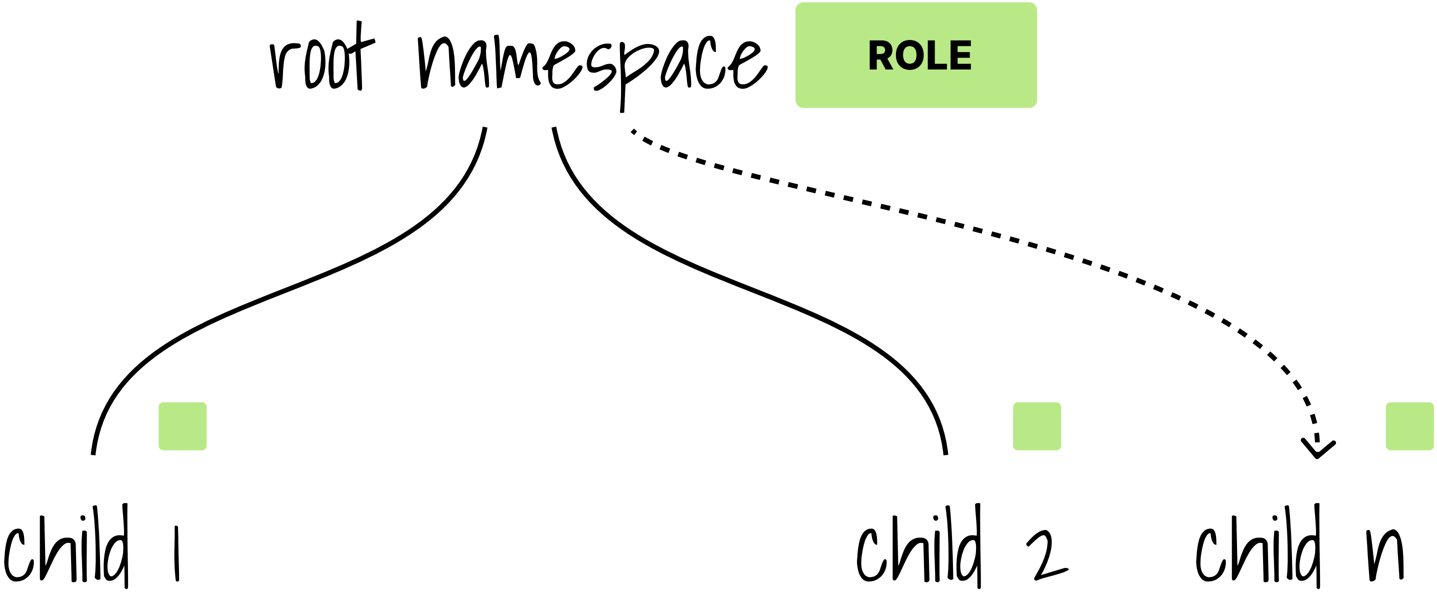
# HNC & Roles

| UID | ROLE | PODS | | PVs | |
|---|---|---|---|---|---|
| | | read | write | read | write |
| 1 | teamA | ✓ | ✓ | ✓ | ✓ |

root namespace

ROLE

|  |  | PODS |  | PVs |  |
|---|---|---|---|---|---|
| UID | ROLE | read | write | read | write |
| 1 | teamA | ✓ | ✓ | ✓ | ✓ |



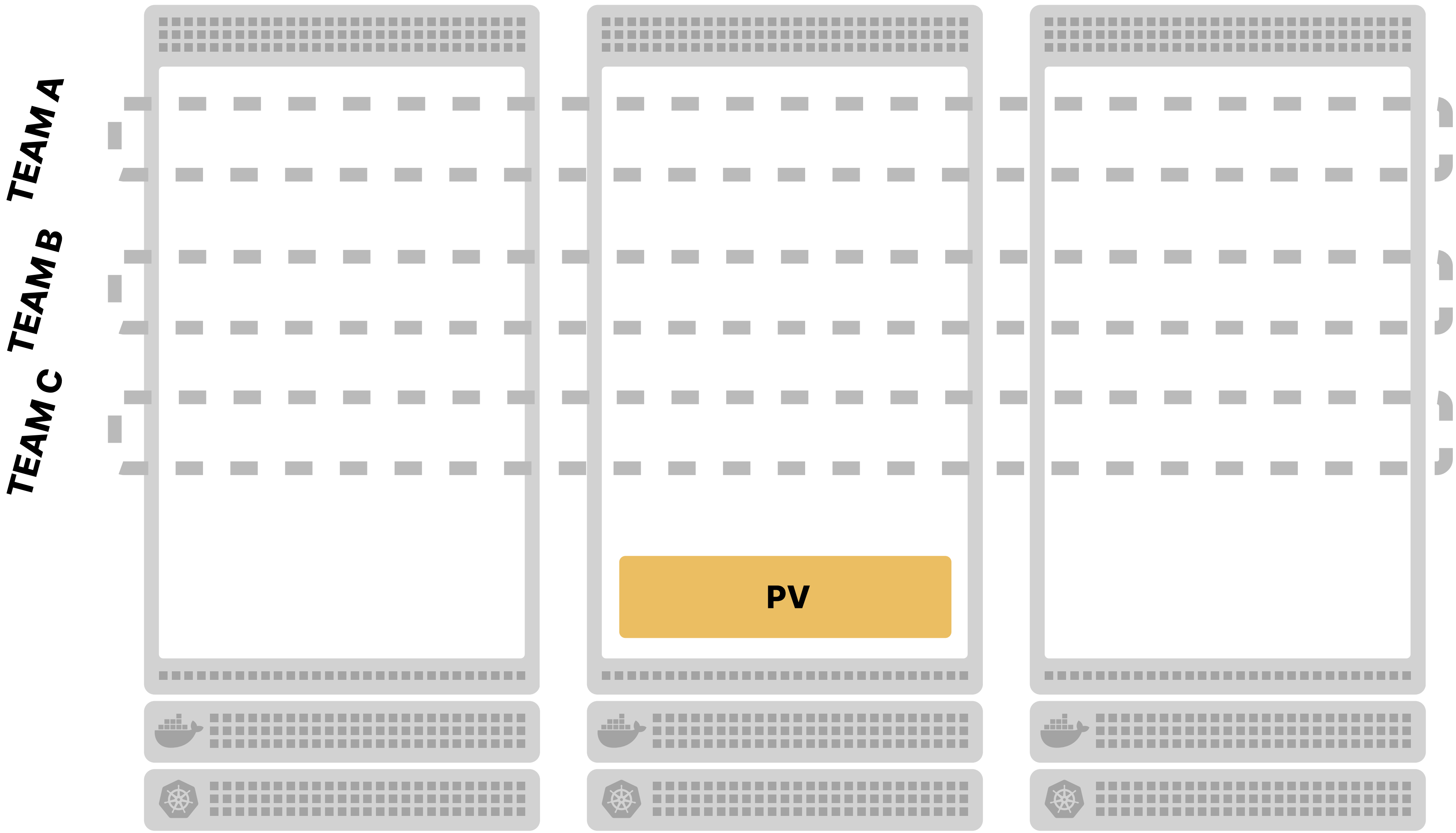
root namespace

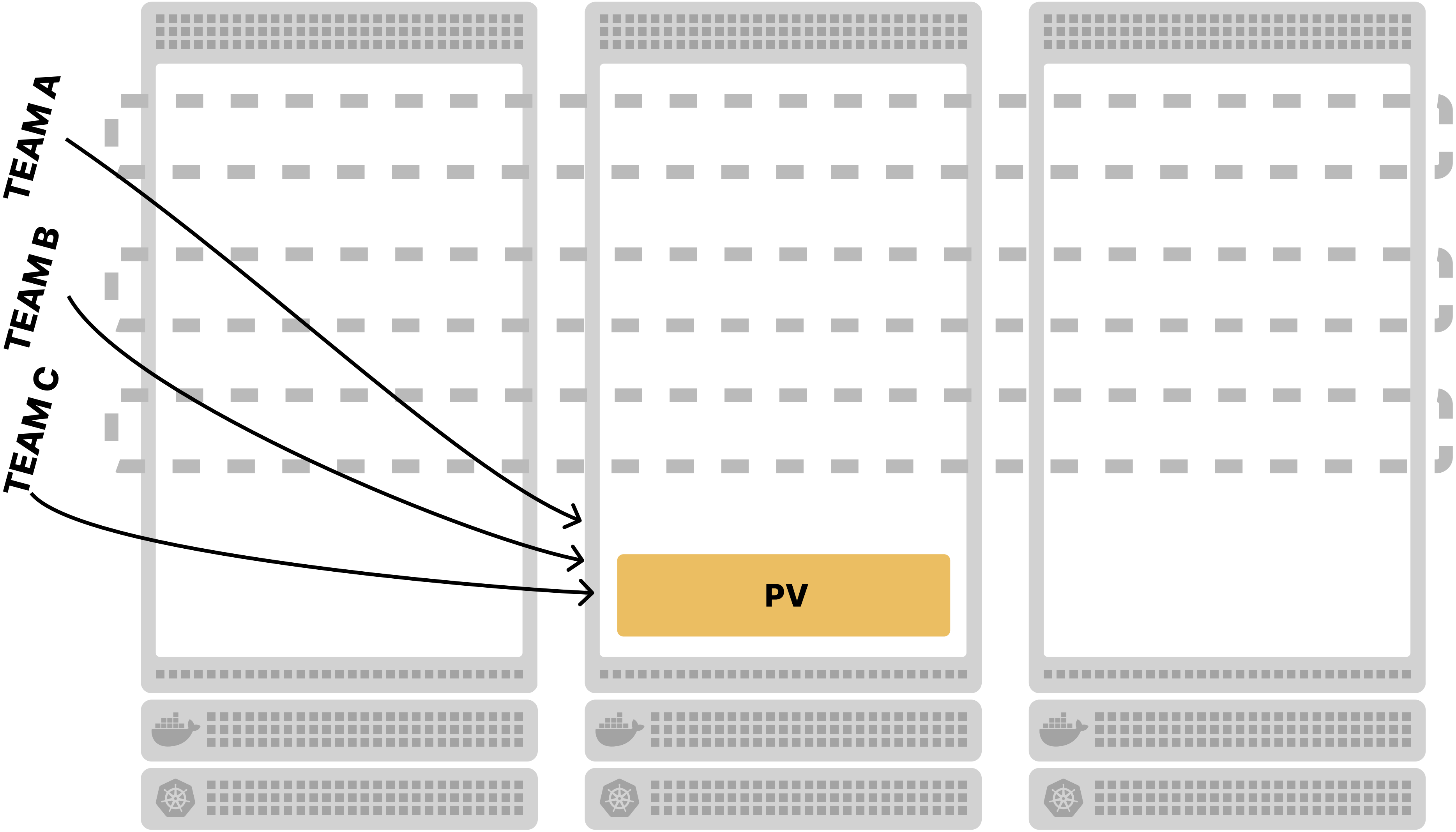
ROLE

child 1

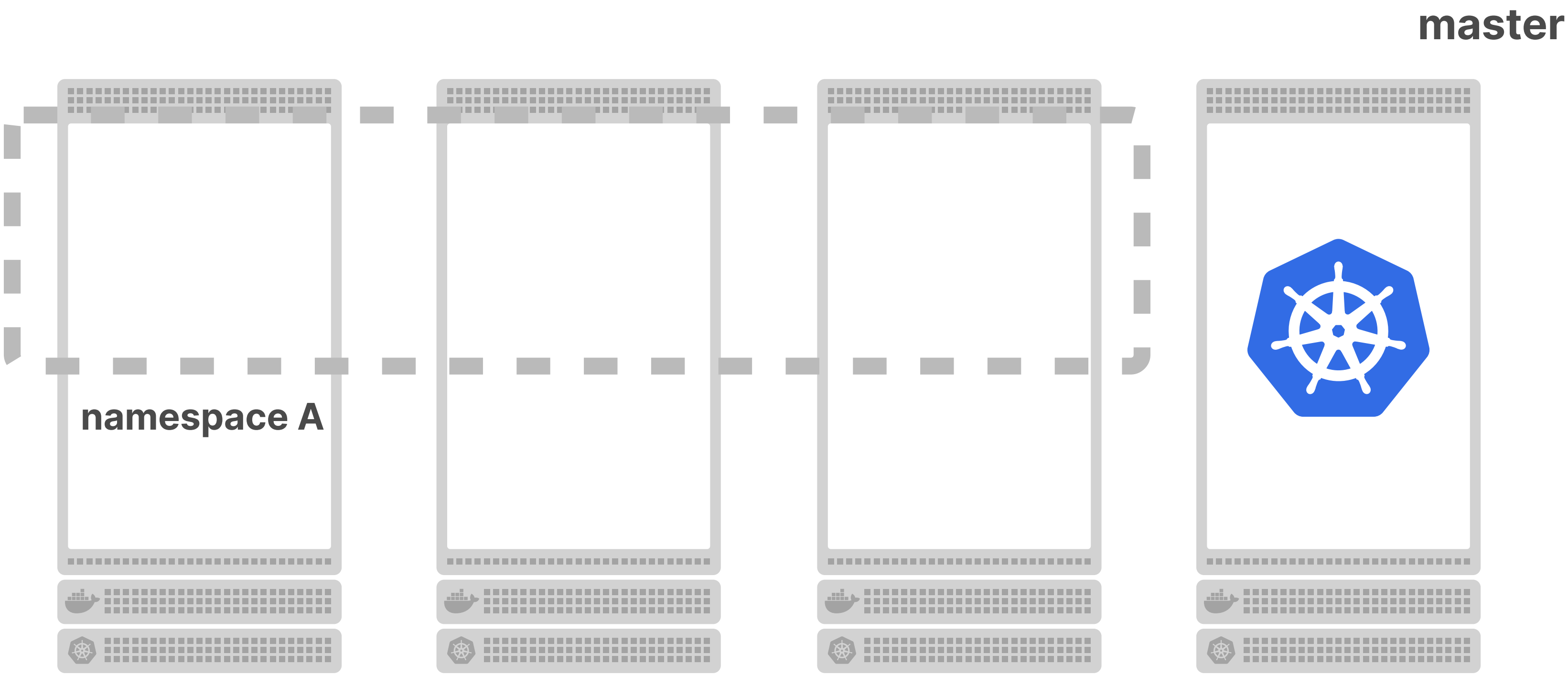
child 2

child n


TEAM A
TEAM B
TEAM C
PV

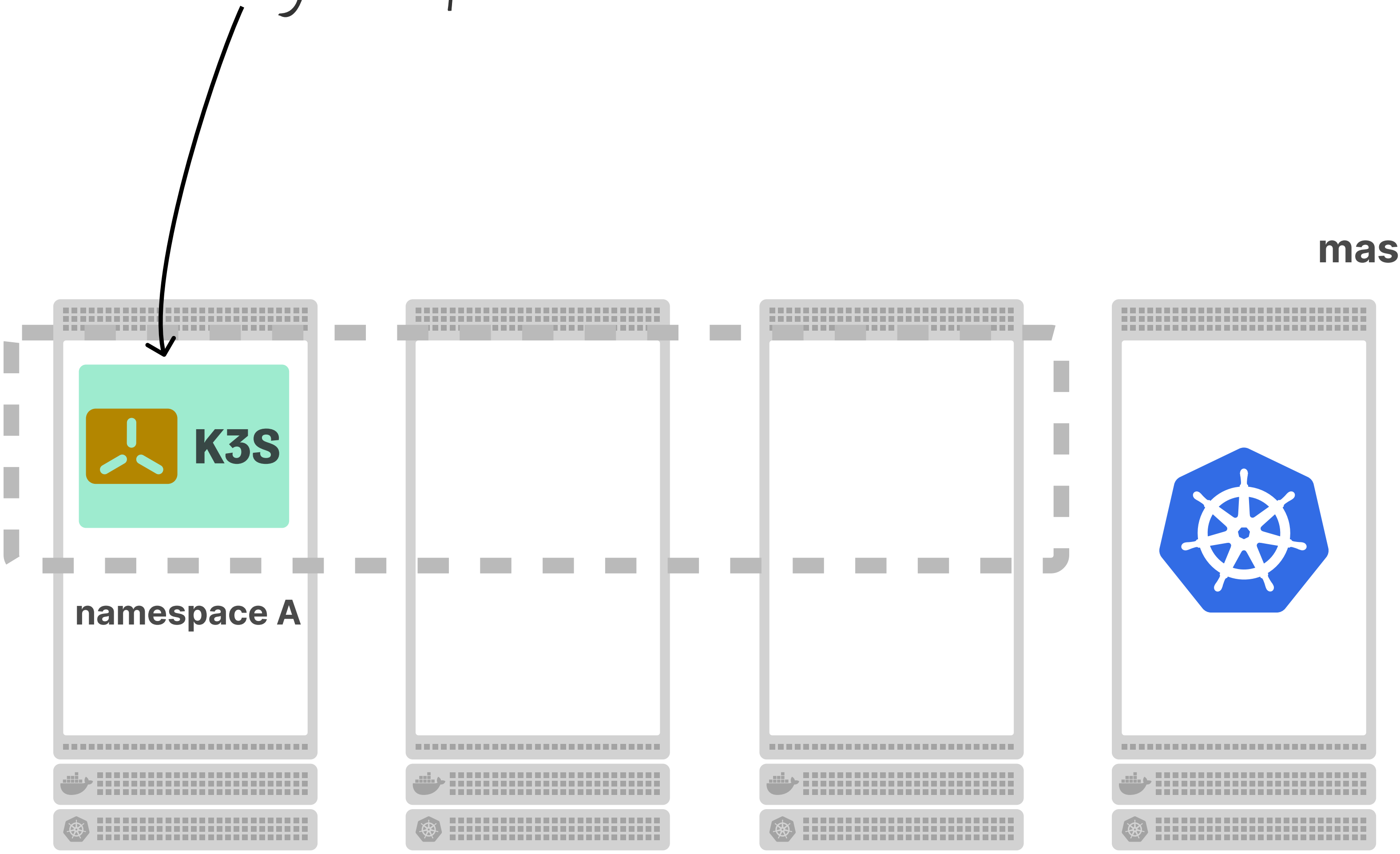

TEAM A
TEAM B
TEAM C
PV

# Isolating control planes



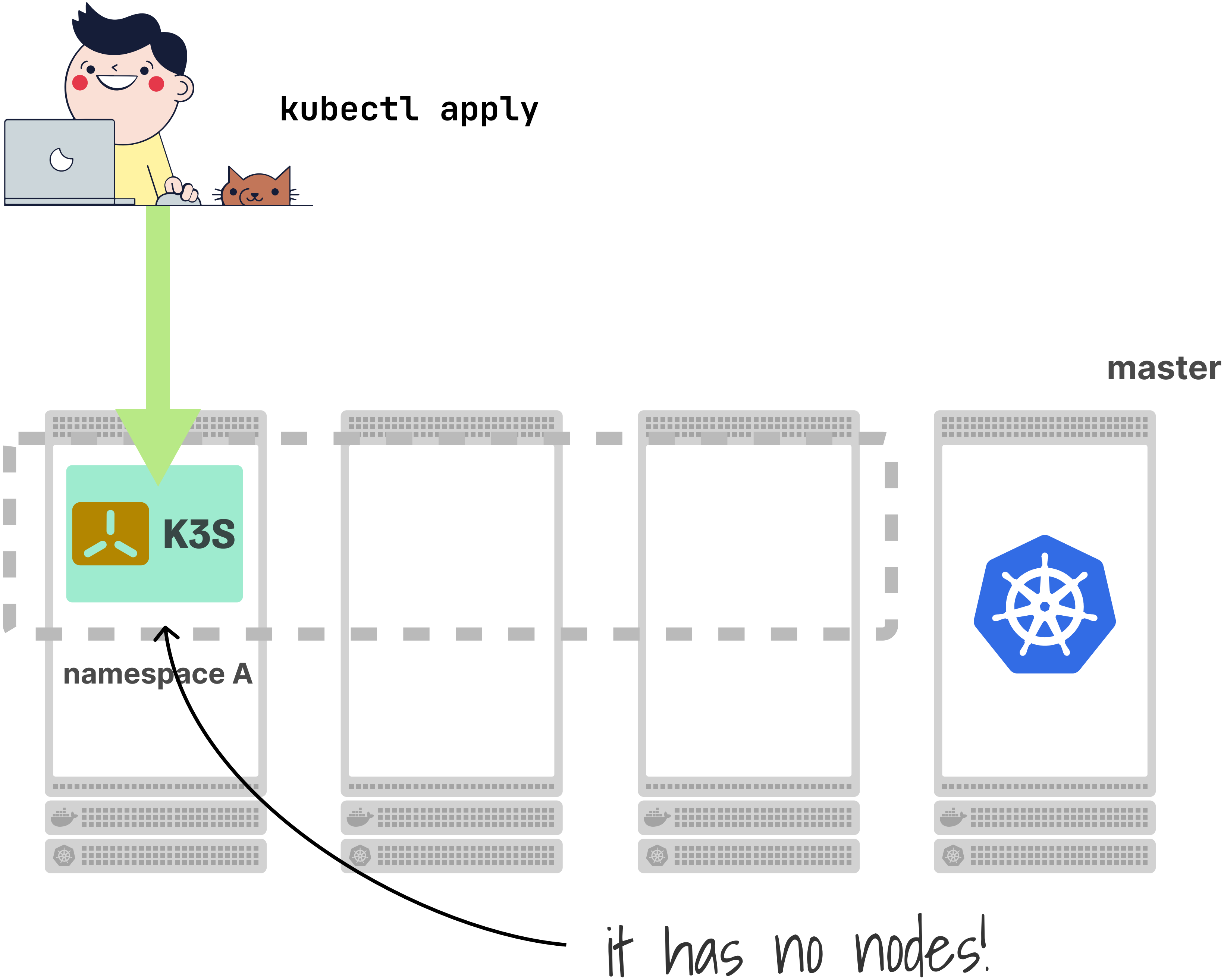
master
namespace A

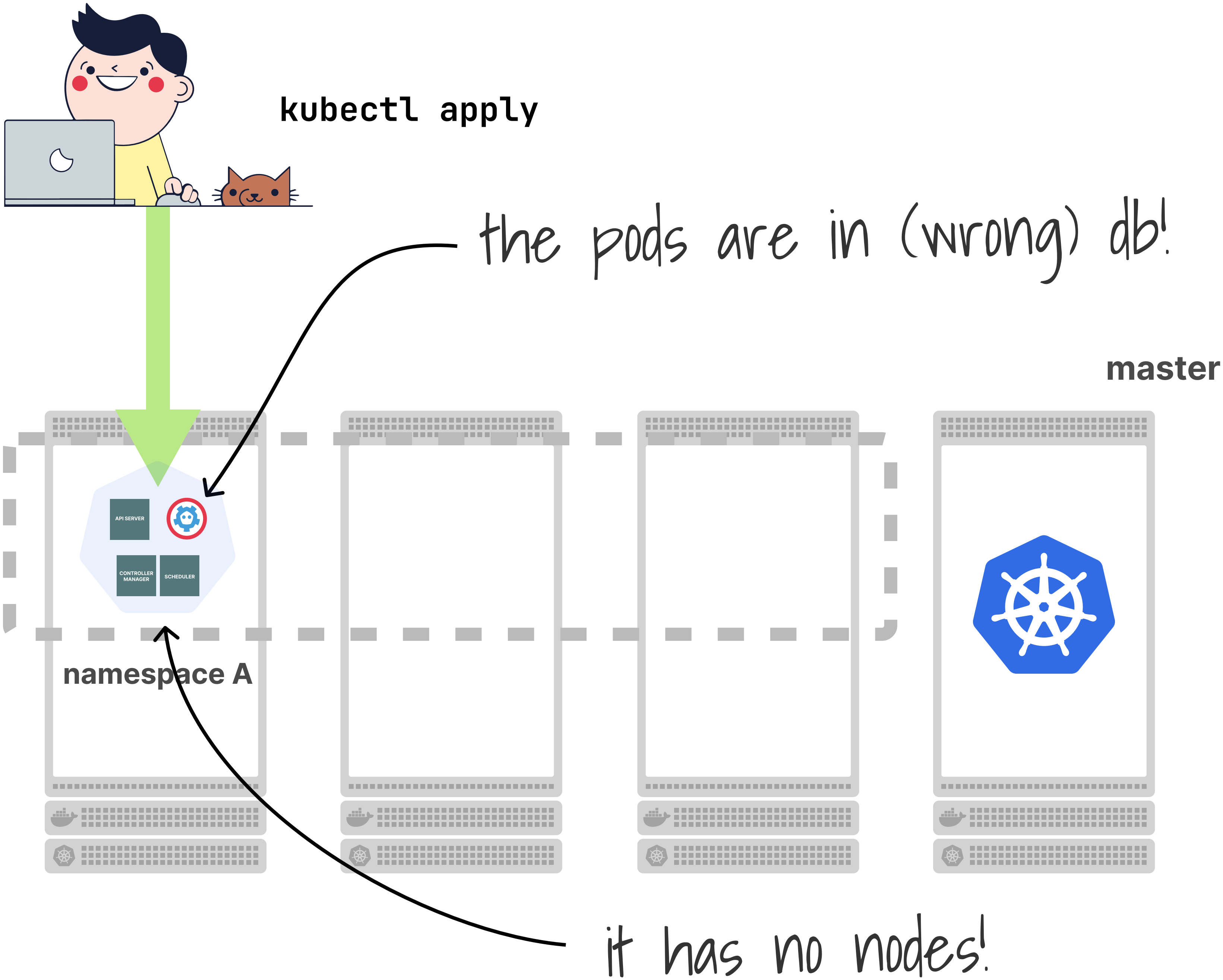
Just a regular pod

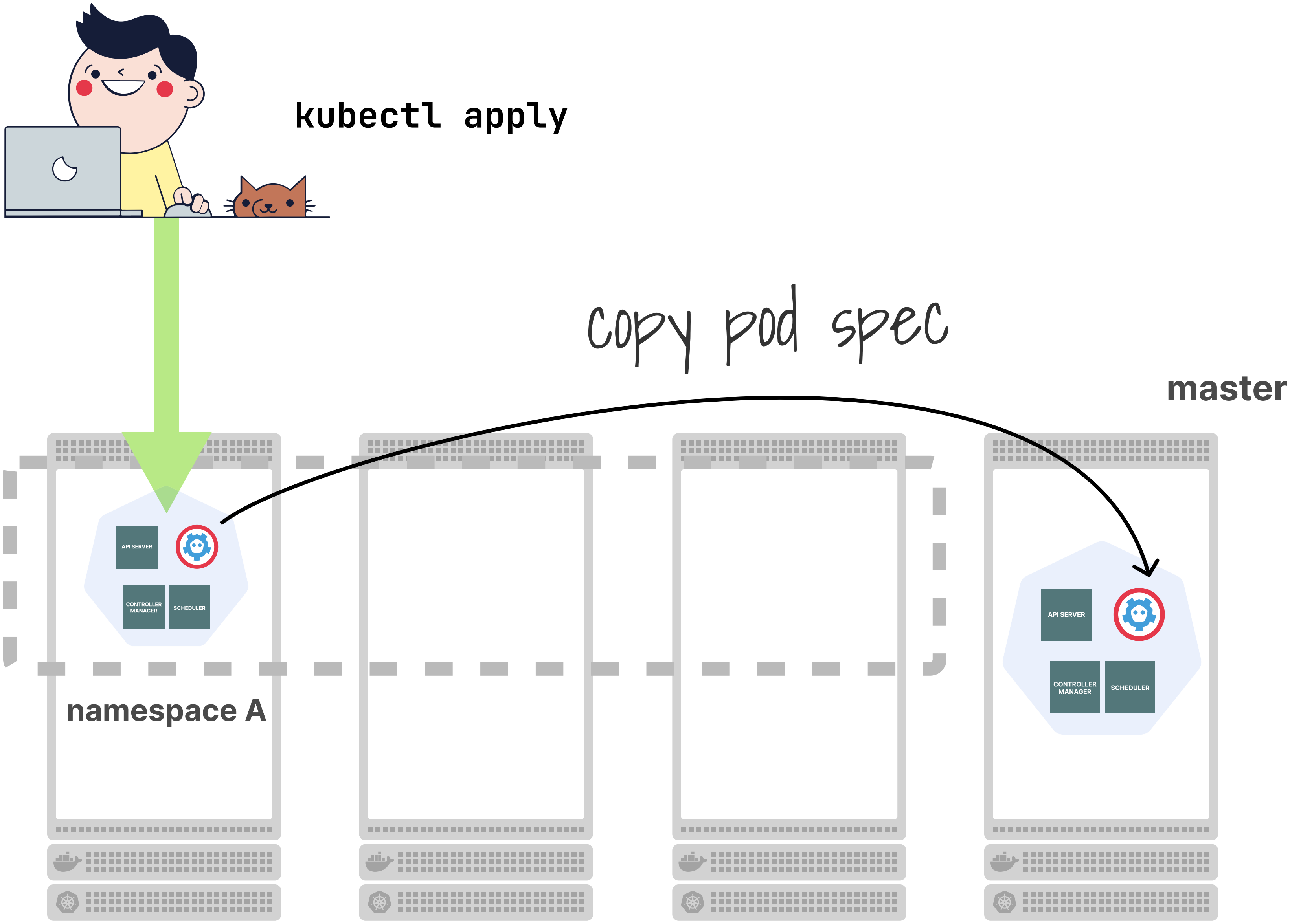


master

namespace A


kubectl apply
master
K3S
namespace A
it has no nodes!


kubectl apply
the pods are in (wrong) db!
master
API SERVER
CONTROLLER MANAGER
SCHEDULER
namespace A
it has no nodes!

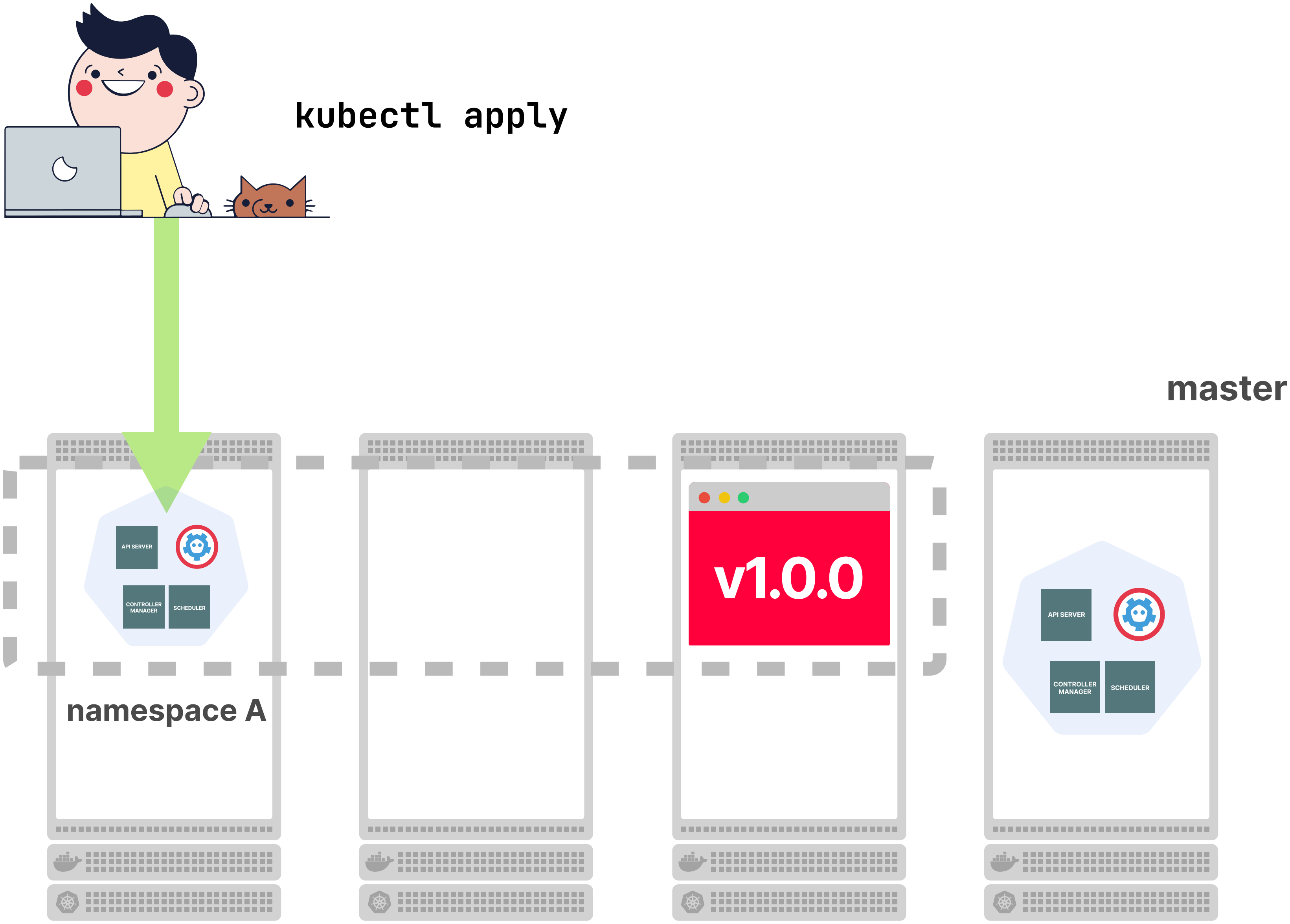


kubectl apply

copy pod spec

master

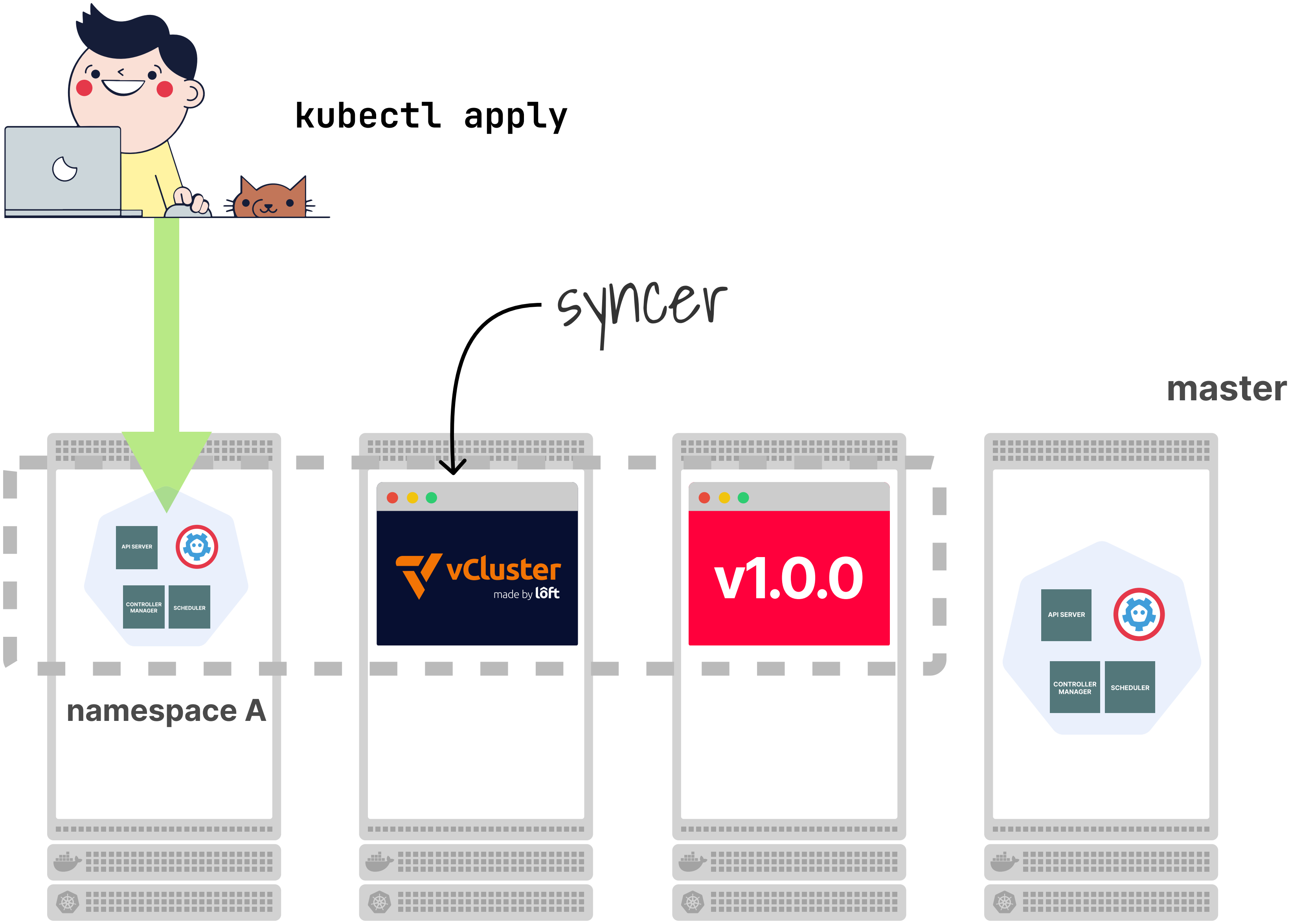
namespace A



kubectl apply

master

namespace A



```
kubectl apply
```

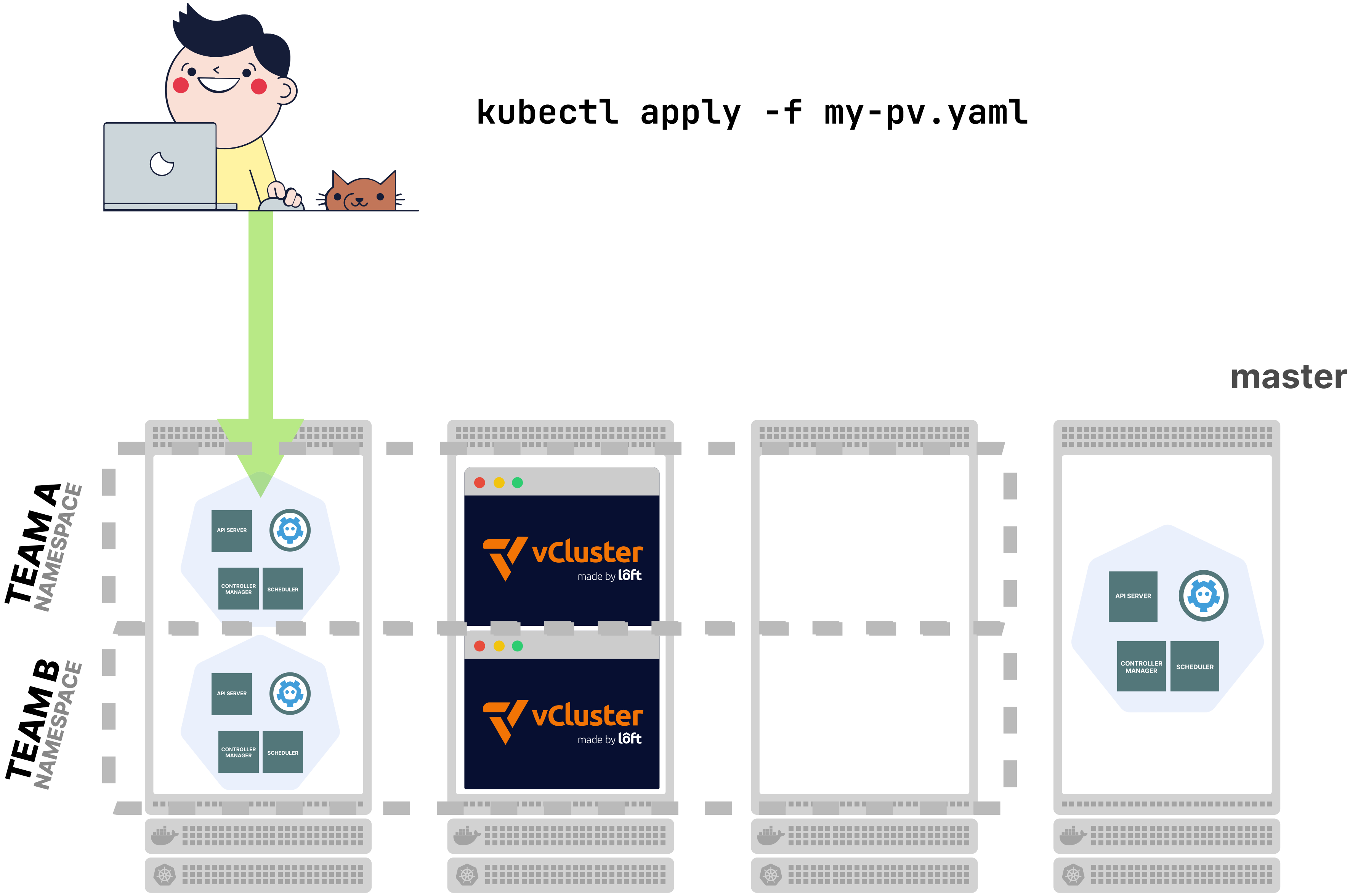
syncer

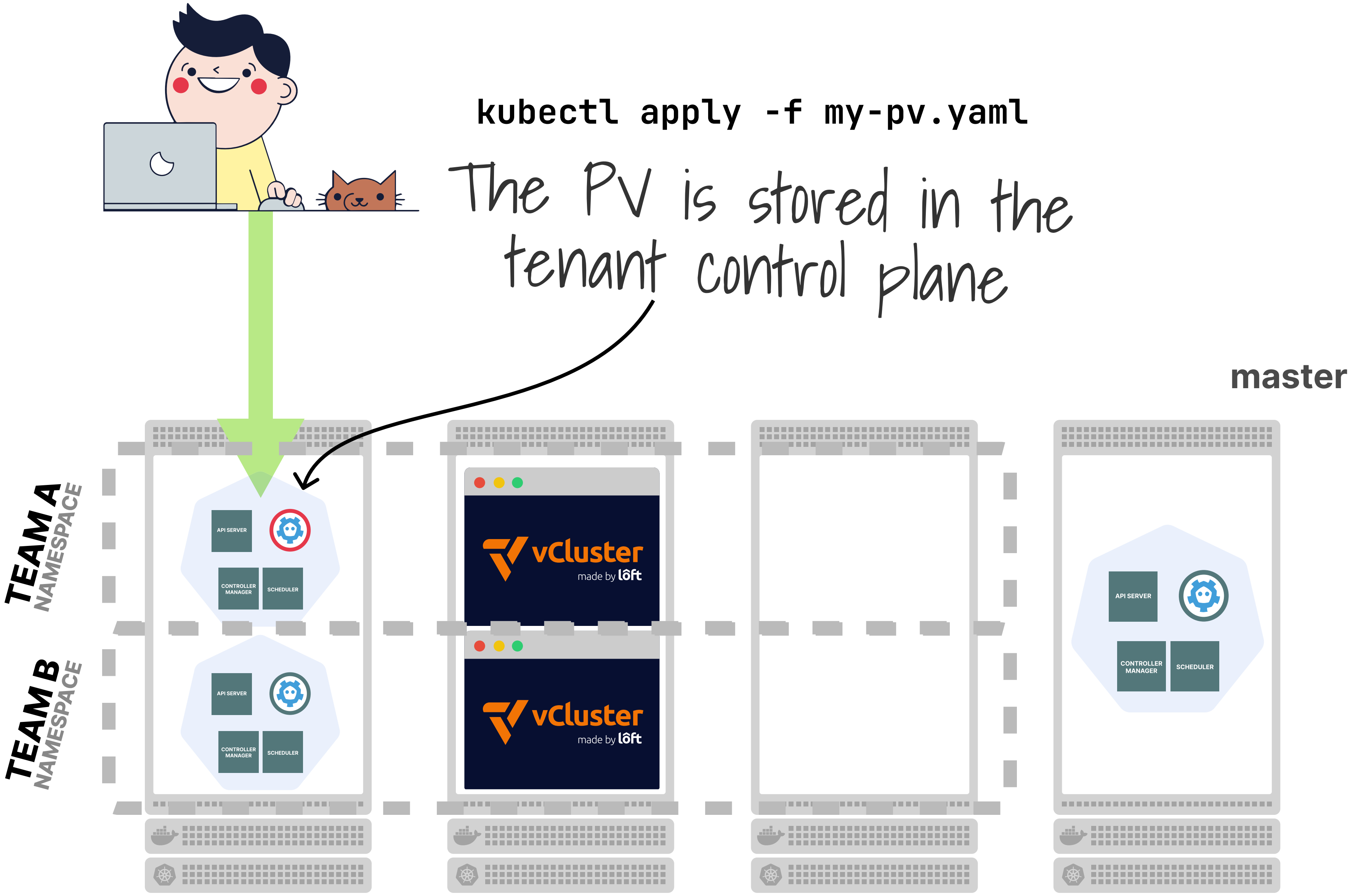
master

namespace A

# vCluster and global resources

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



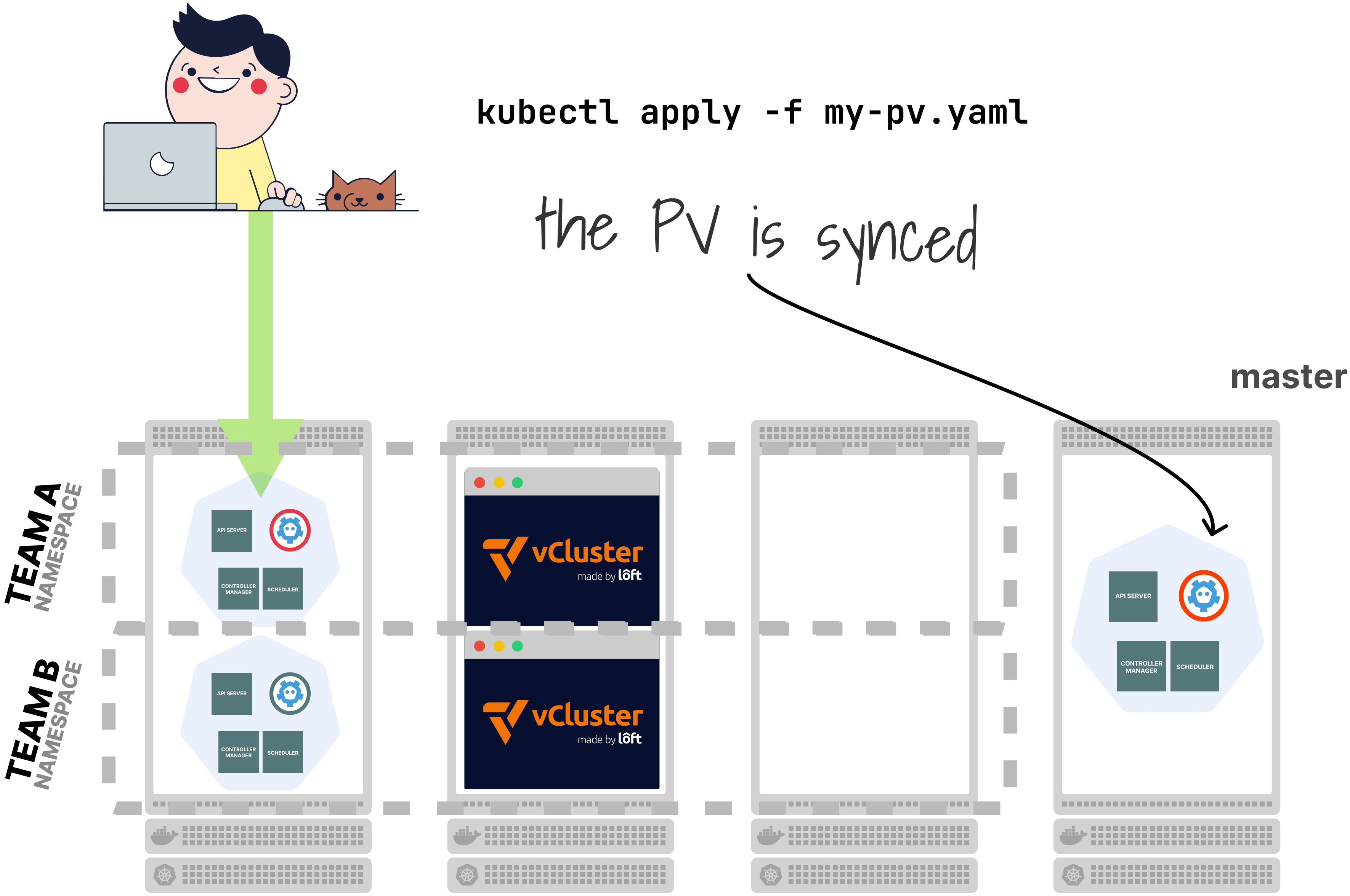


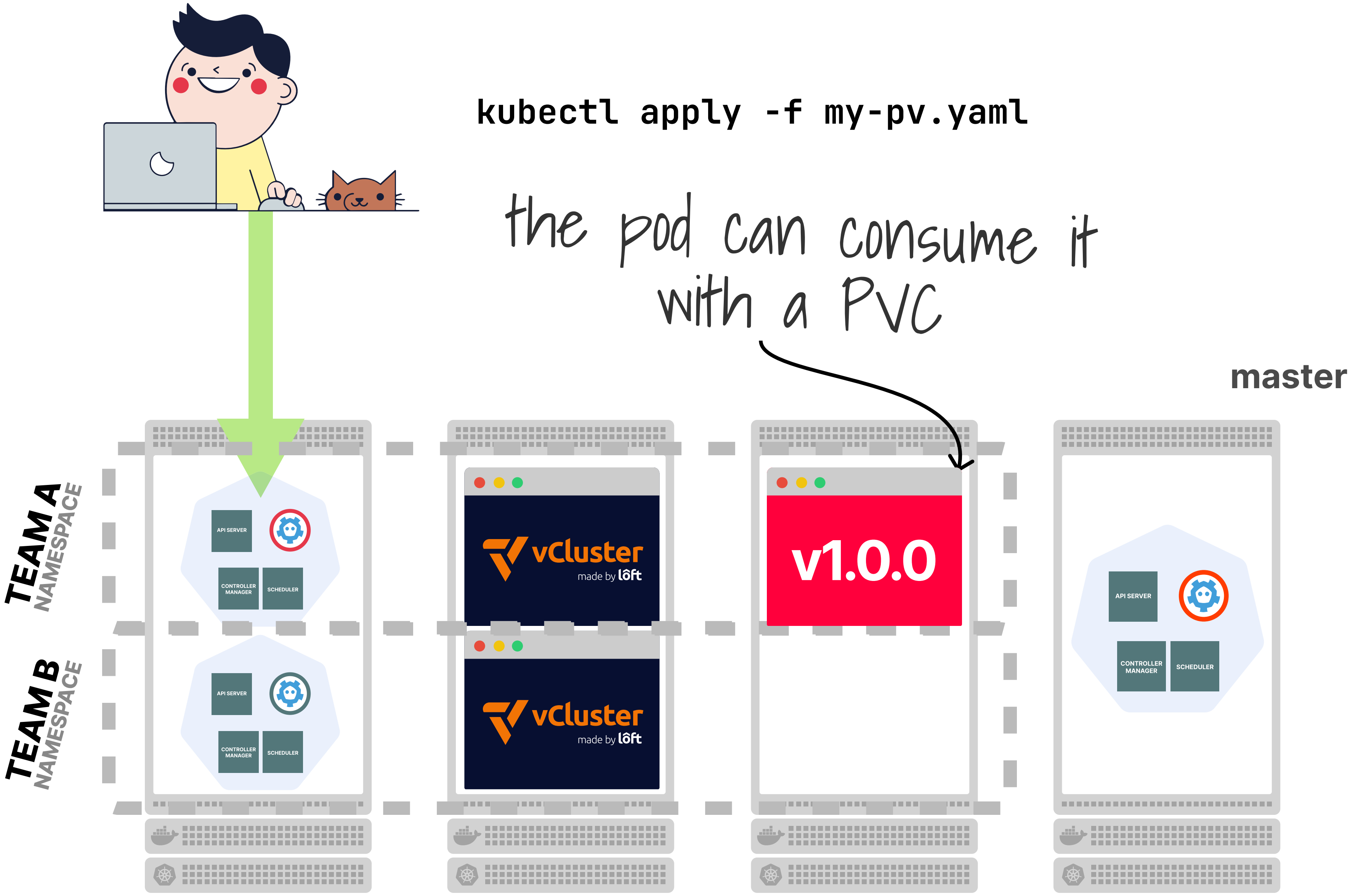
```
kubectl apply -f my-pv.yaml
```

The PV is stored in the tenant control plane

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

the PV is synced





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

the pod can consume it with a PVC

# Demo

## vCluster

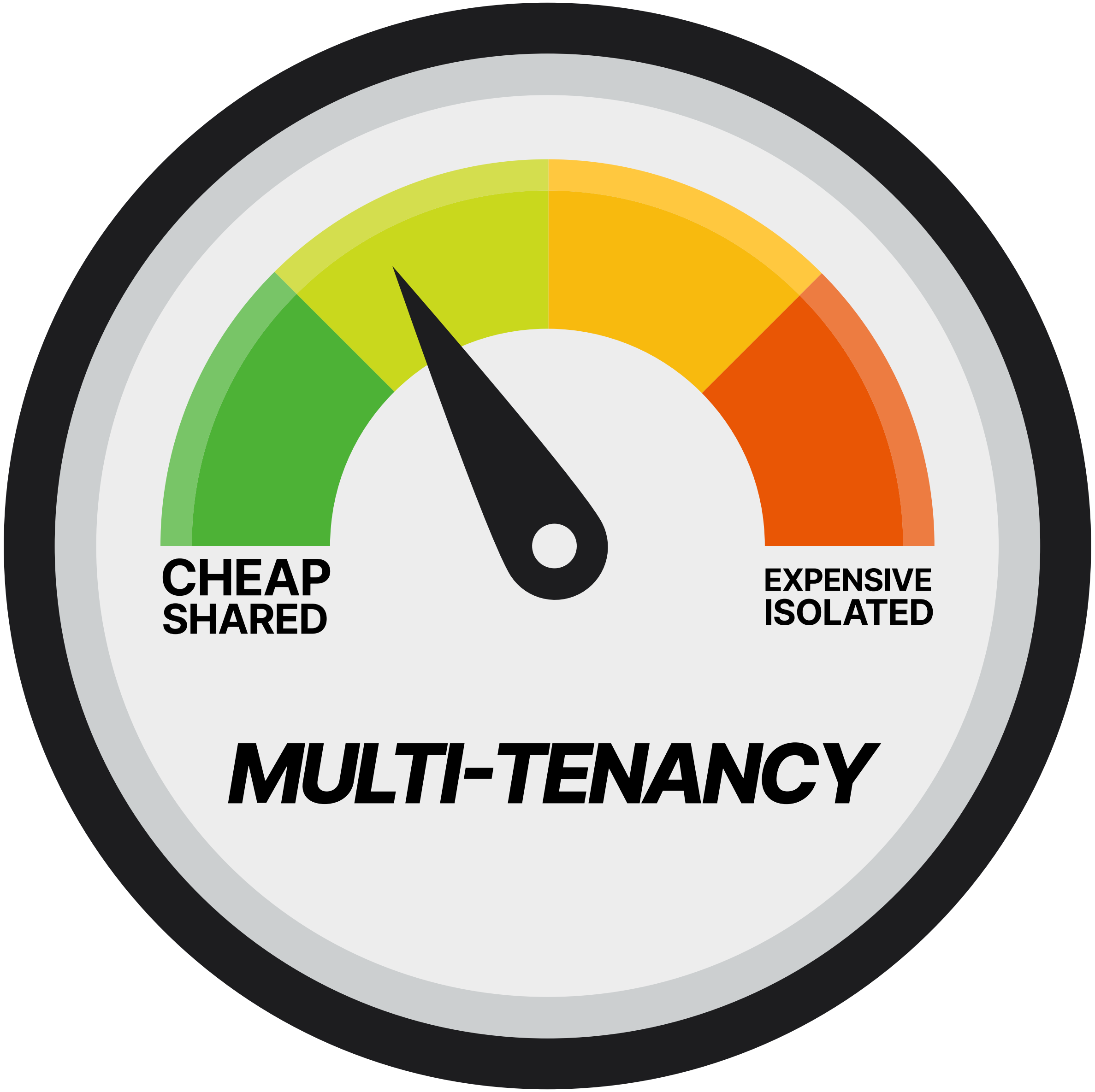
# “Nested” control planes

## vCluster

# “Nested” control planes
# Admin vs tenants

## vCluster

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



CHEAP
SHARED

EXPENSIVE
ISOLATED

MULTI-TENANCY

## COSTS FOR 50 TENANTS

# + 17 nodes x $12

# COSTS FOR 50 TENANTS

+ 17 nodes x $12

+ 50 PVs x $1

## COSTS FOR 50 TENANTS

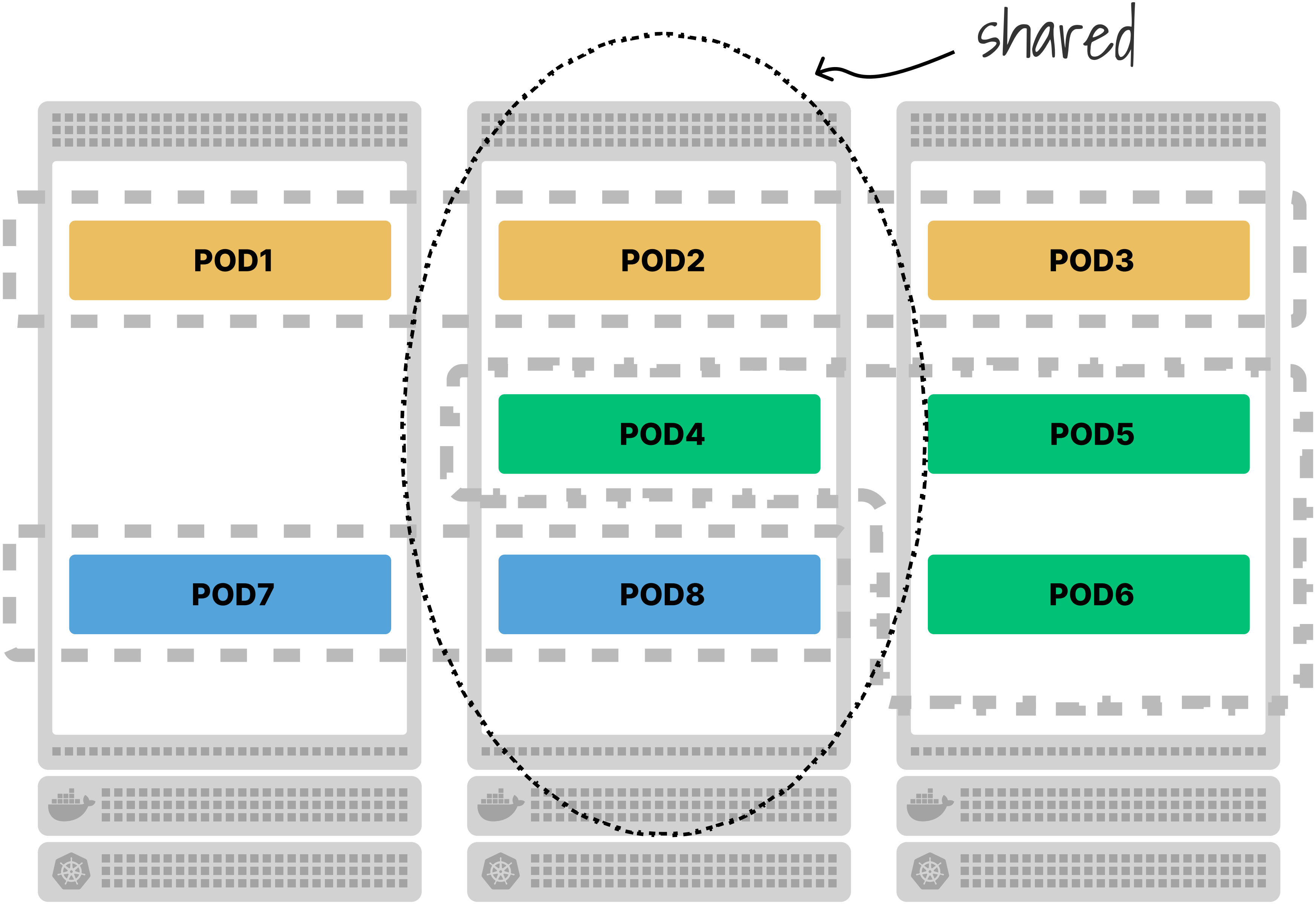
+ 17 nodes x $12

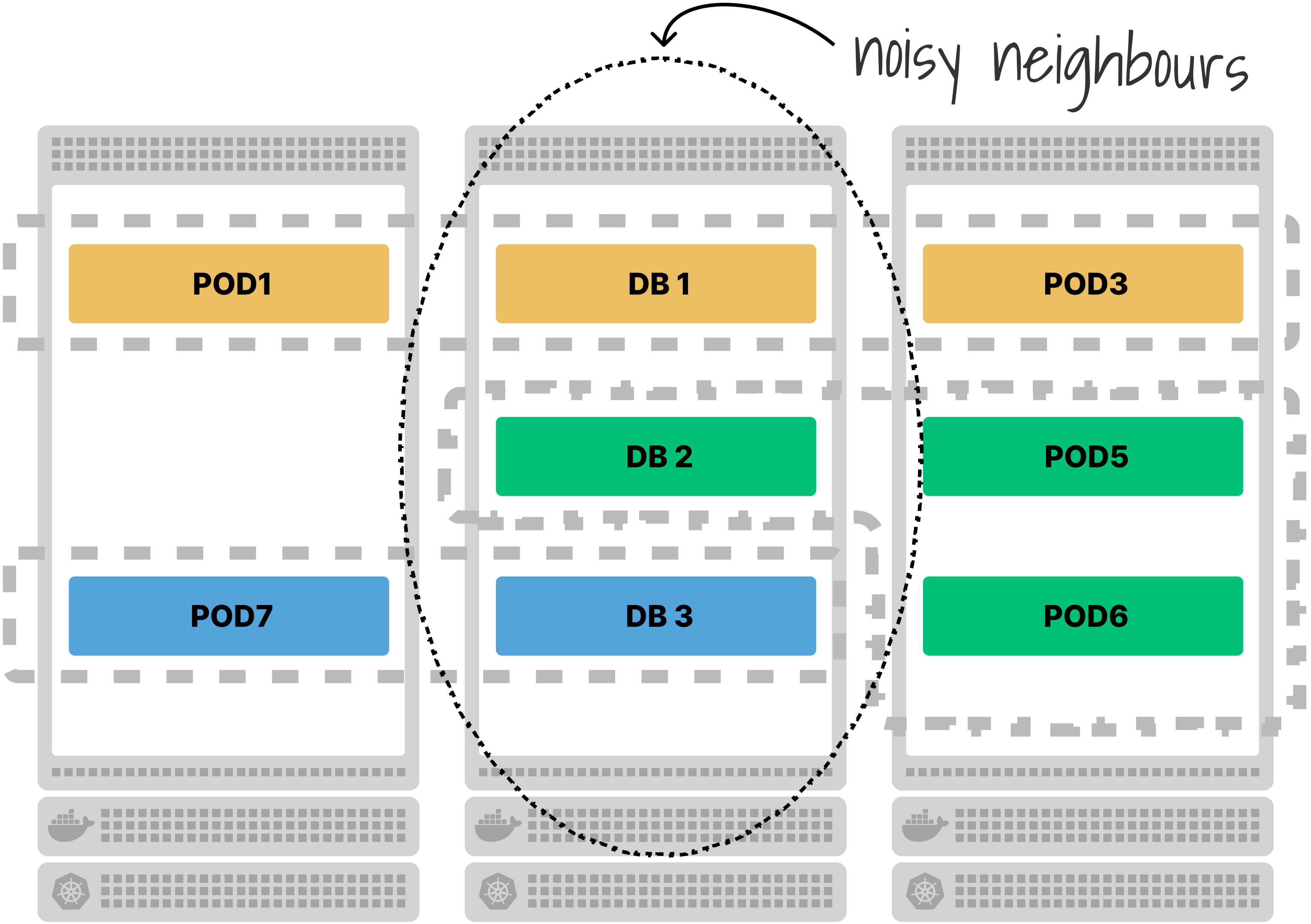
+ 50 PVs x $1

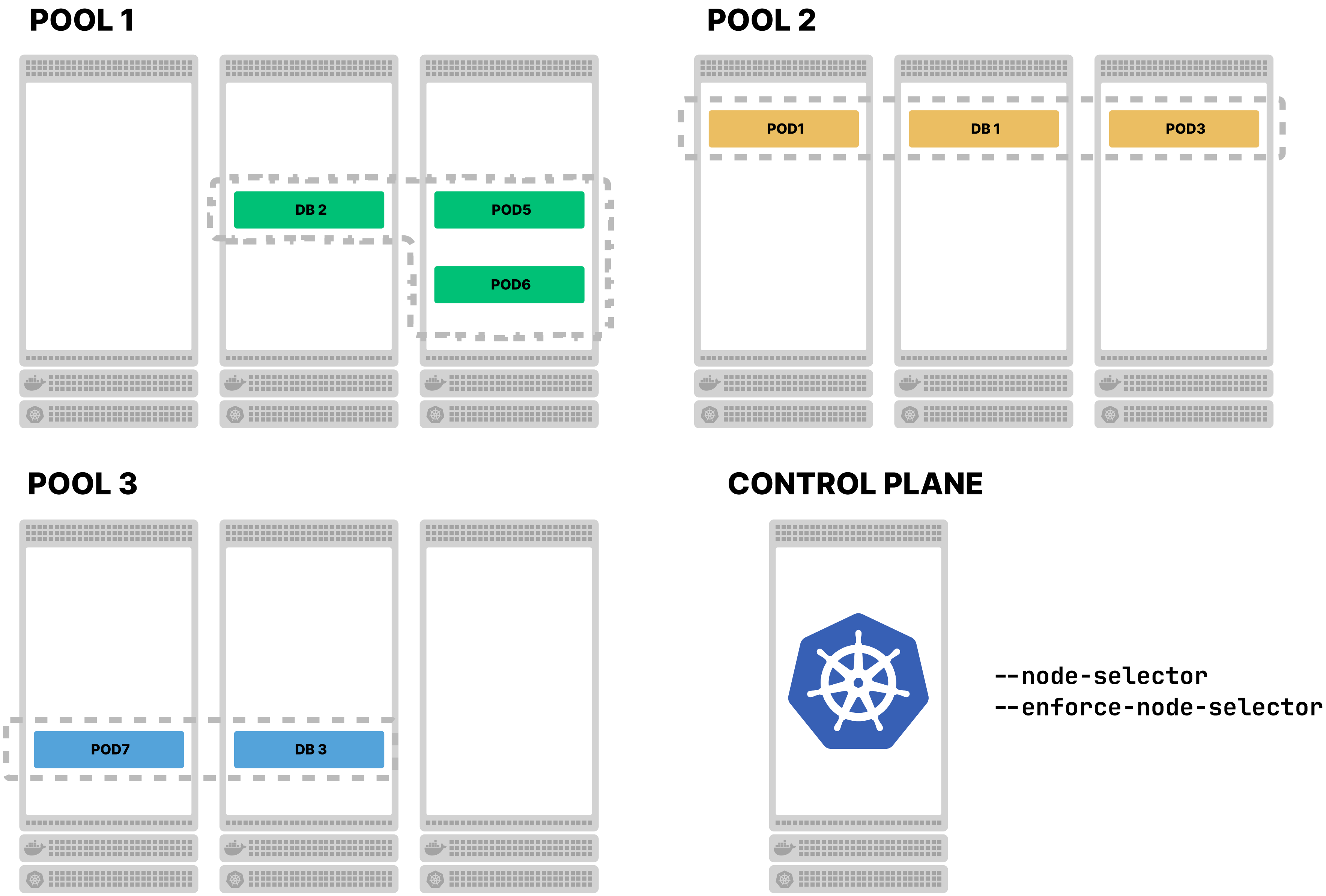
= $254 / month

~$5 / month / tenant

# vCluster and shared nodes


shared
POD1
POD2
POD3
POD4
POD5
POD7
POD8
POD6


noisy neighbours
POD1
DB 1
POD3
DB 2
POD5
POD7
DB 3
POD6



POOL 1

DB 2

POD5

POD6

POOL 2

POD1

DB 1

POD3

POOL 3

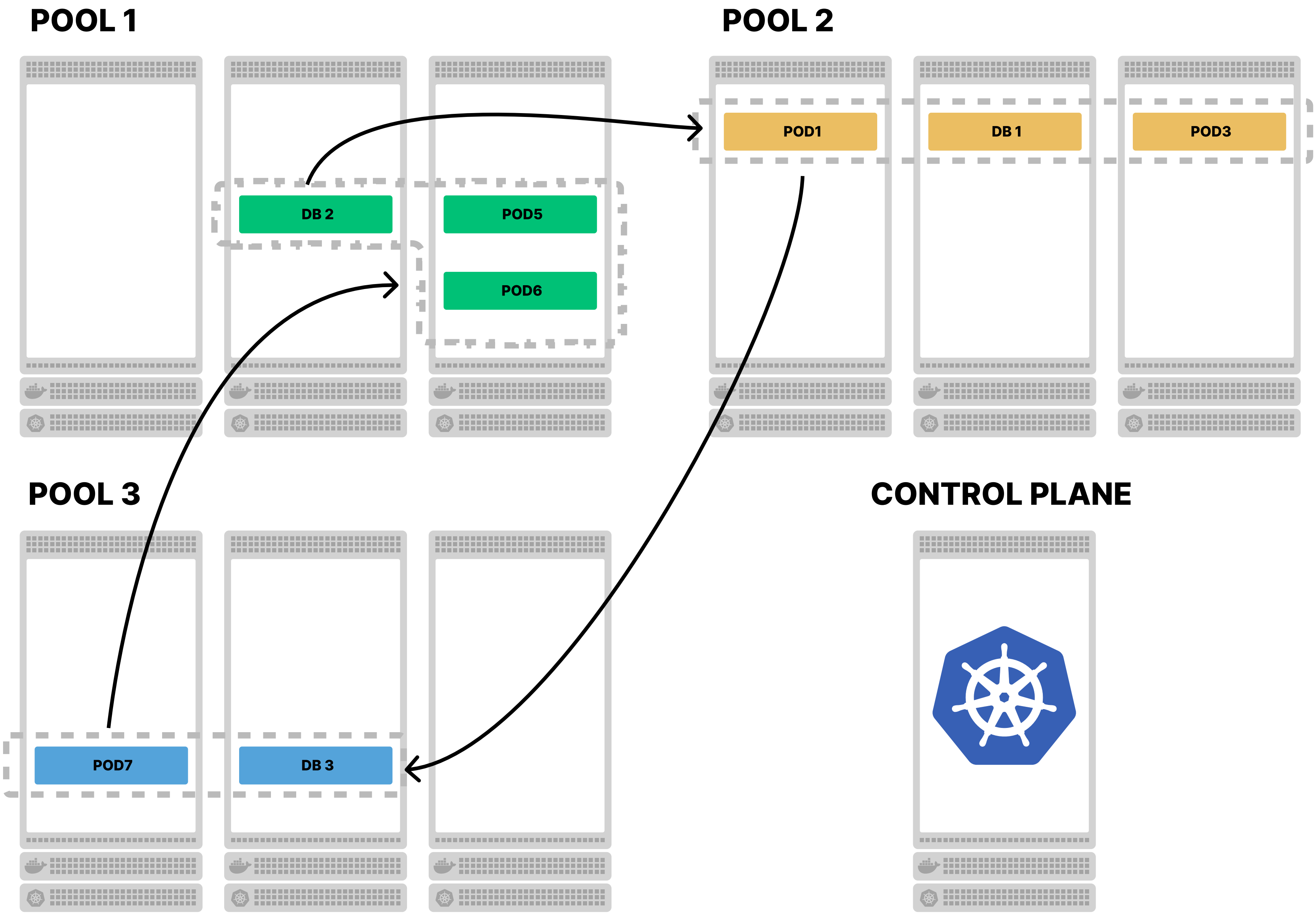
POD7

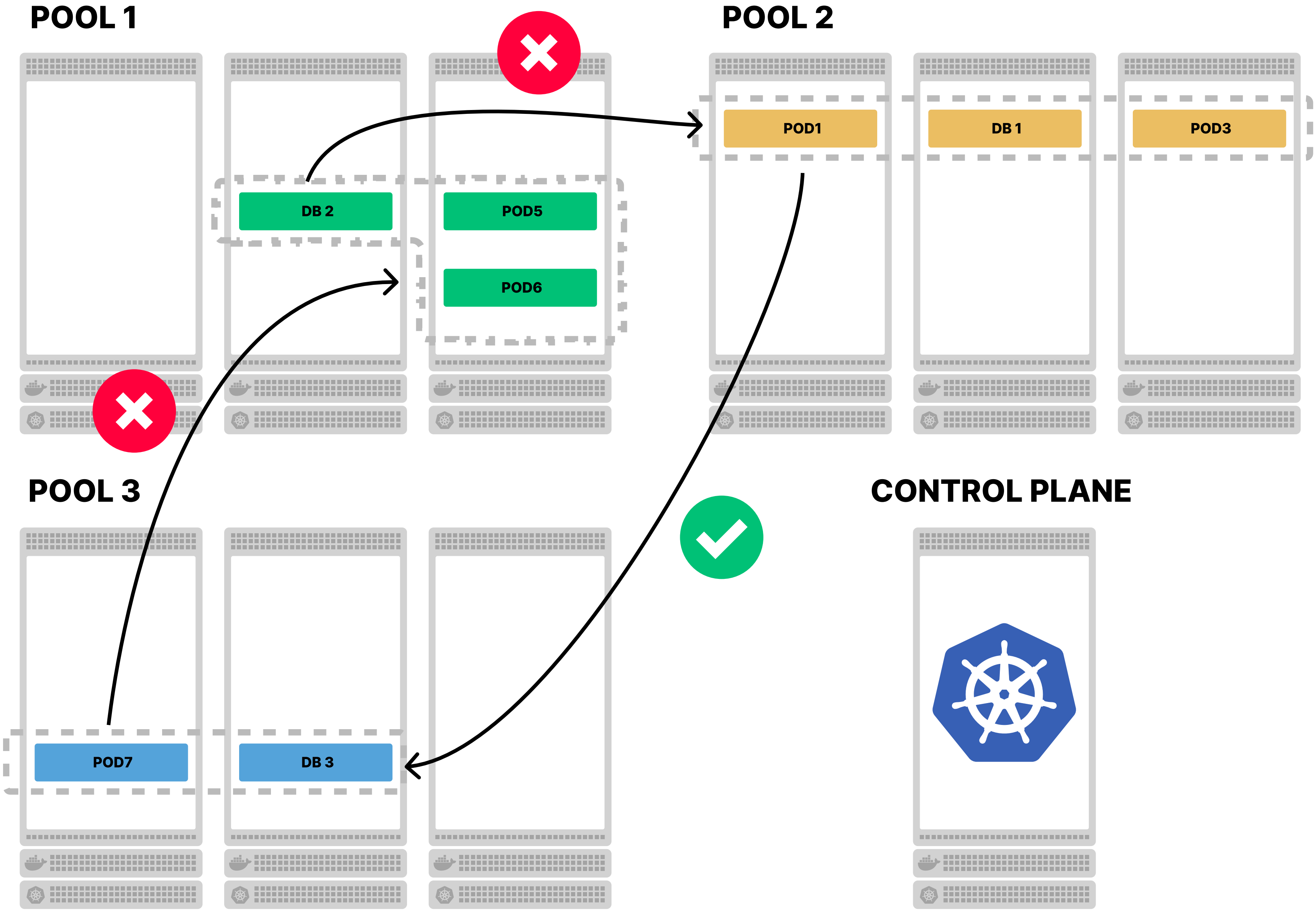
DB 3

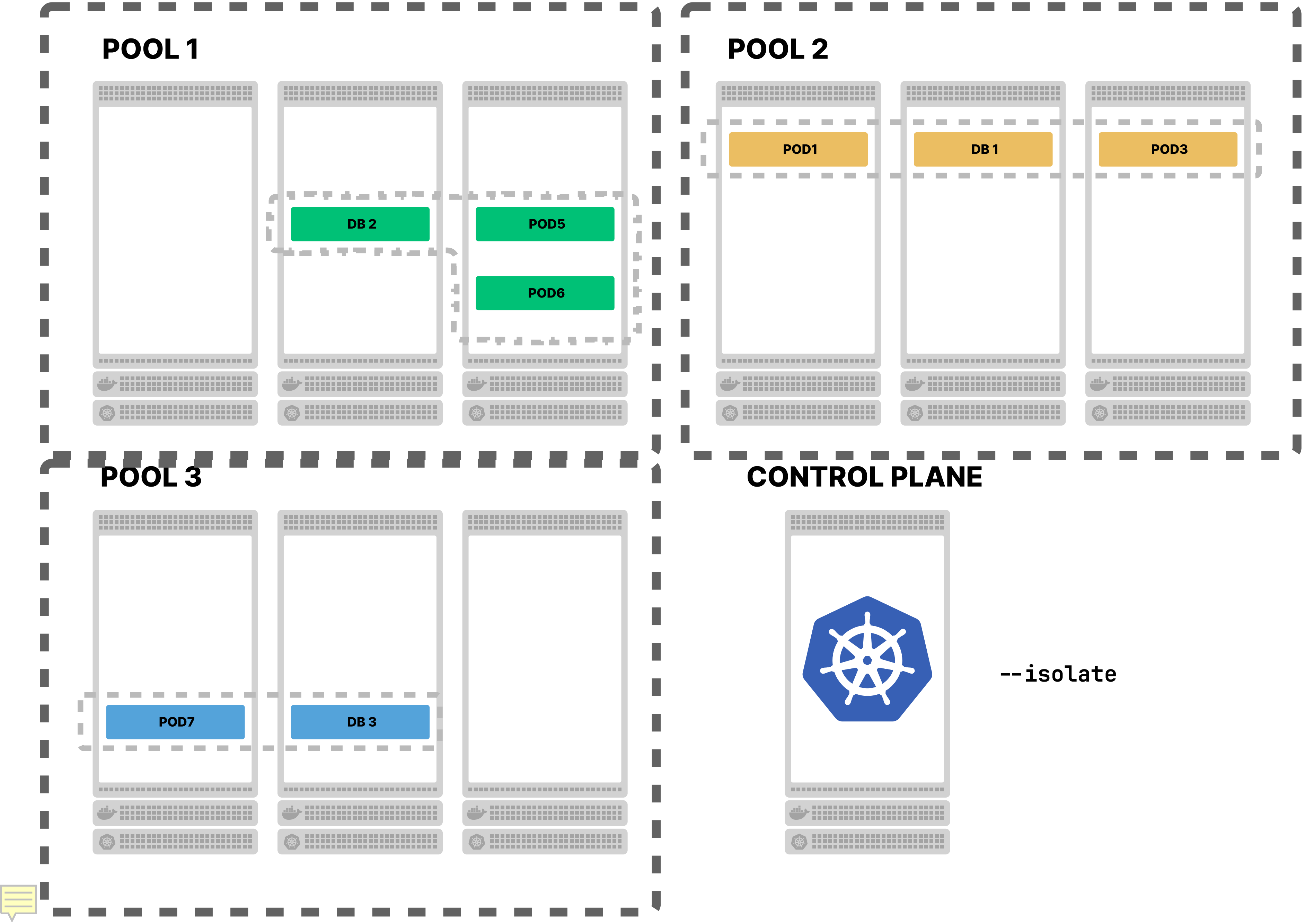
CONTROL PLANE

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

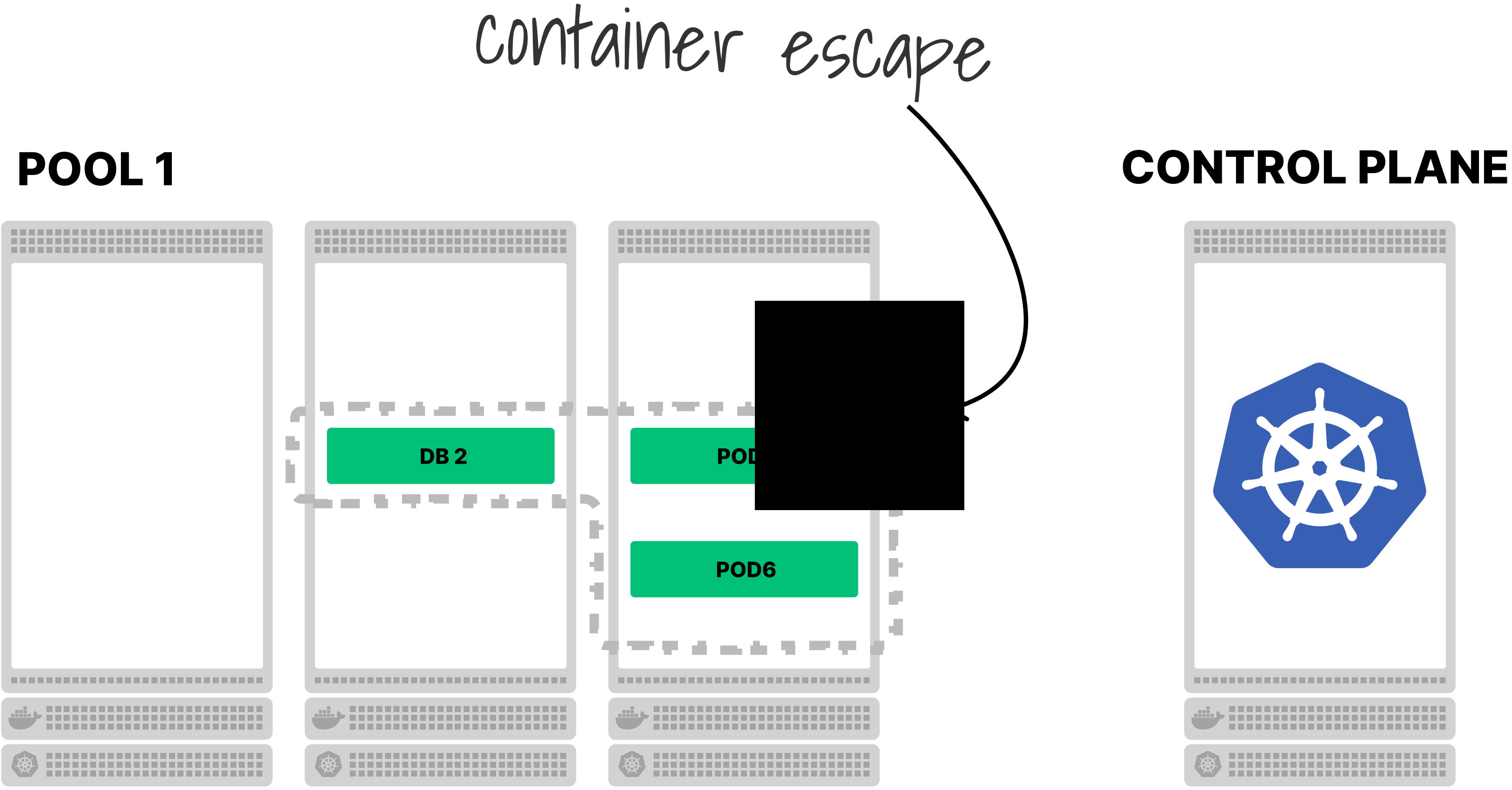
# vCluster  and shared network


POOL 1
POOL 2
POD1
DB 1
POD3
DB 2
POD5
POD6
POOL 3
CONTROL PLANE
POD7
DB 3


POOL 1
POOL 2
DB 2
POD5
POD6
POD1
DB 1
POD3
POOL 3
CONTROL PLANE
POD7
DB 3


POOL 1
DB 2
POD5
POD6
POOL 2
POD1
DB 1
POD3
POOL 3
POD7
DB 3
CONTROL PLANE
--isolate

# vCluster and shared cluster



container escape

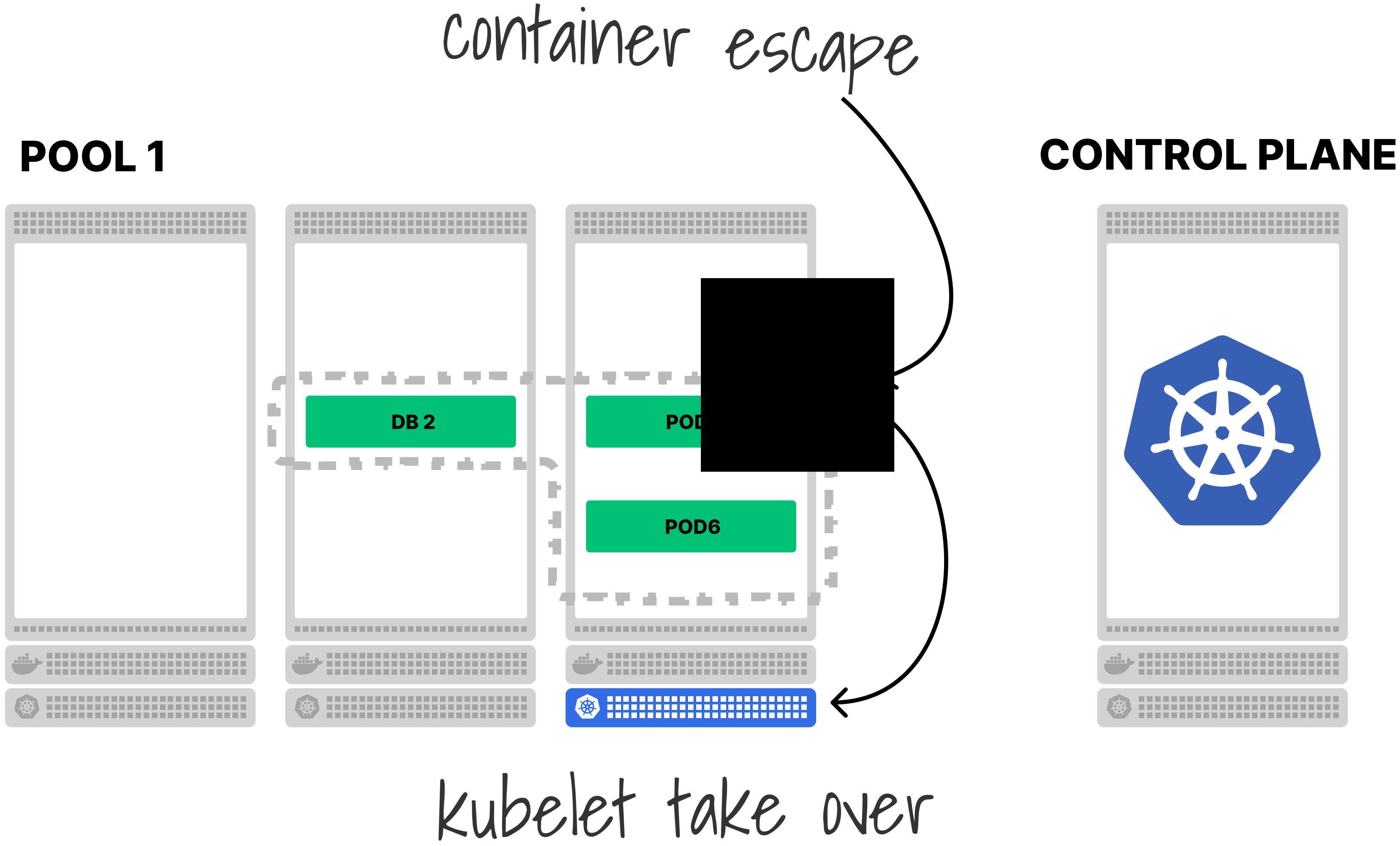
POOL 1

CONTROL PLANE

DB 2

POD [redacted]

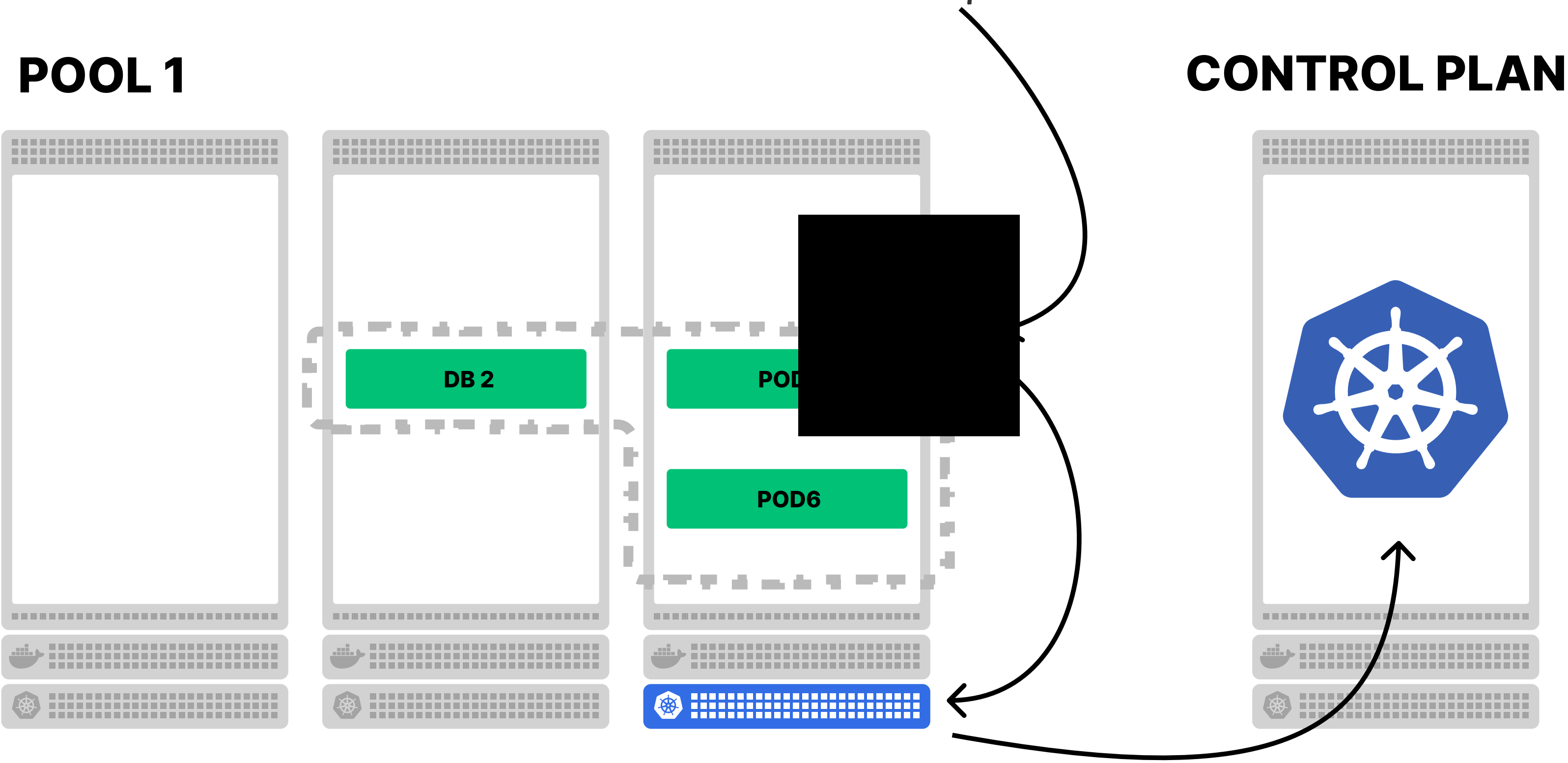
POD6



container escape
POOL 1
CONTROL PLANE
DB 2
POD [redacted]
POD6
kubelet take over

POOL 1

CONTROL PLANE

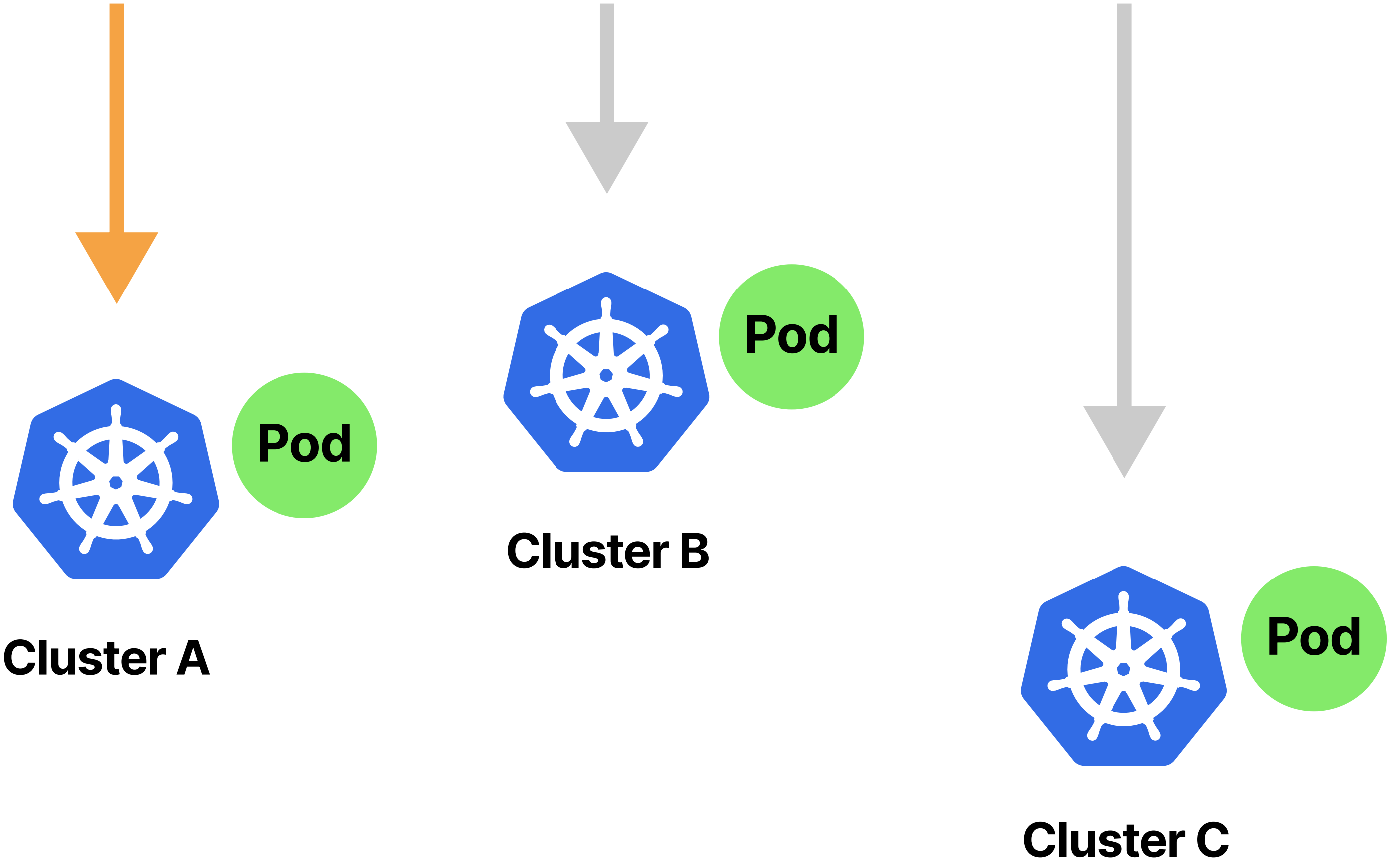


container escape

kubelet take over

control plane escalation

# Dedicated clusters



Pod

Cluster A

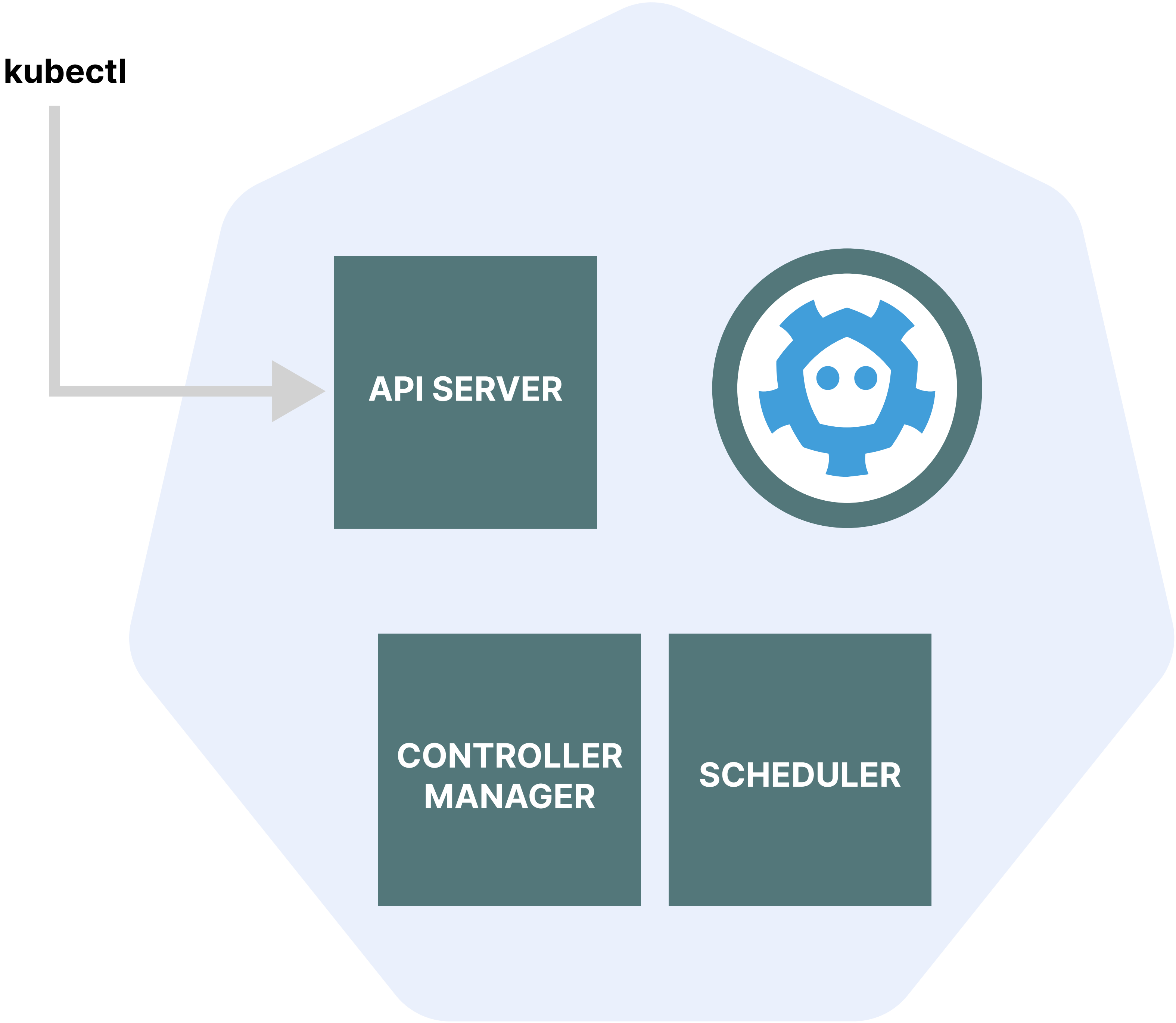
Pod

Cluster B

Pod

Cluster C

# Karmada

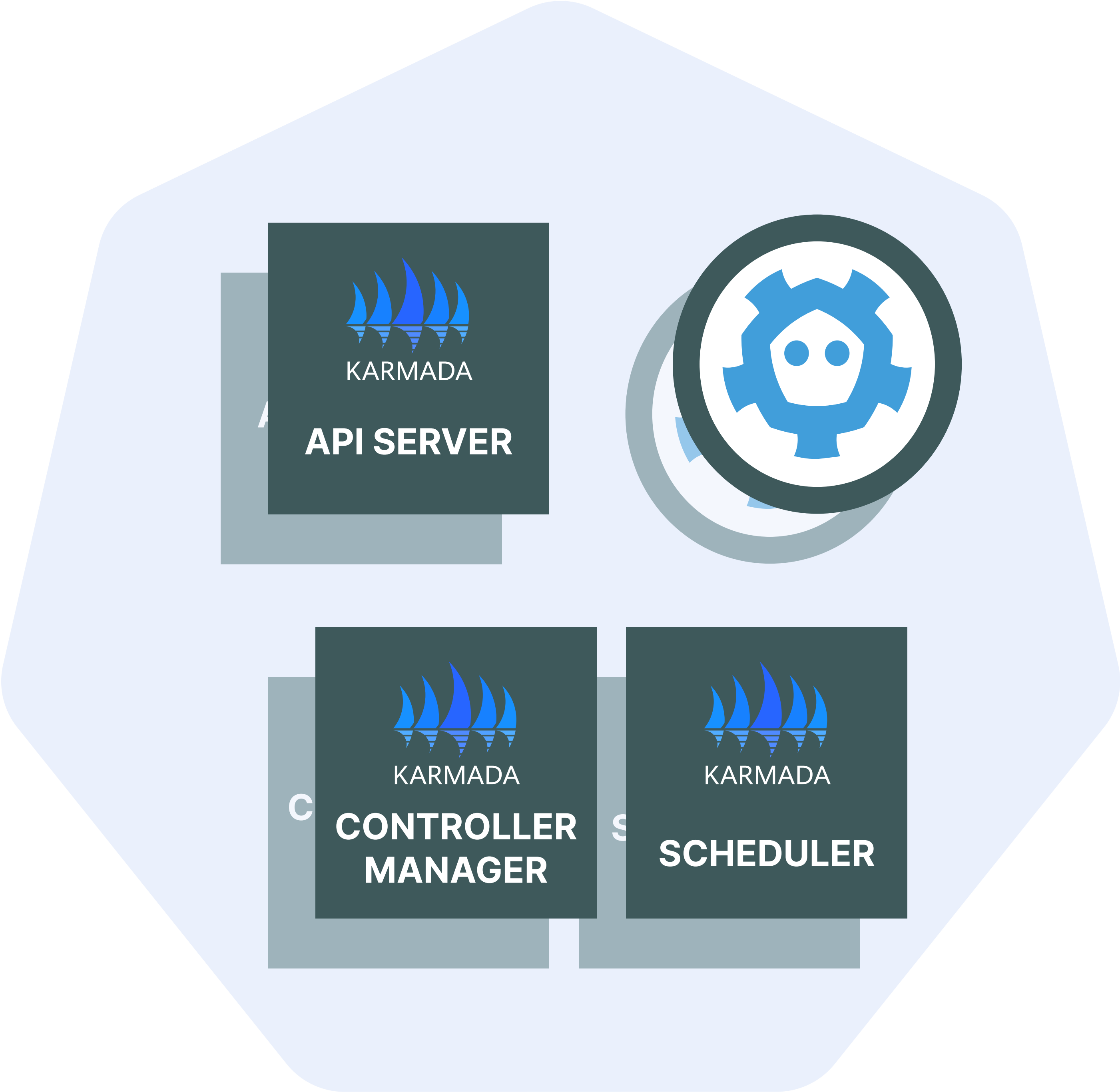


kubectl

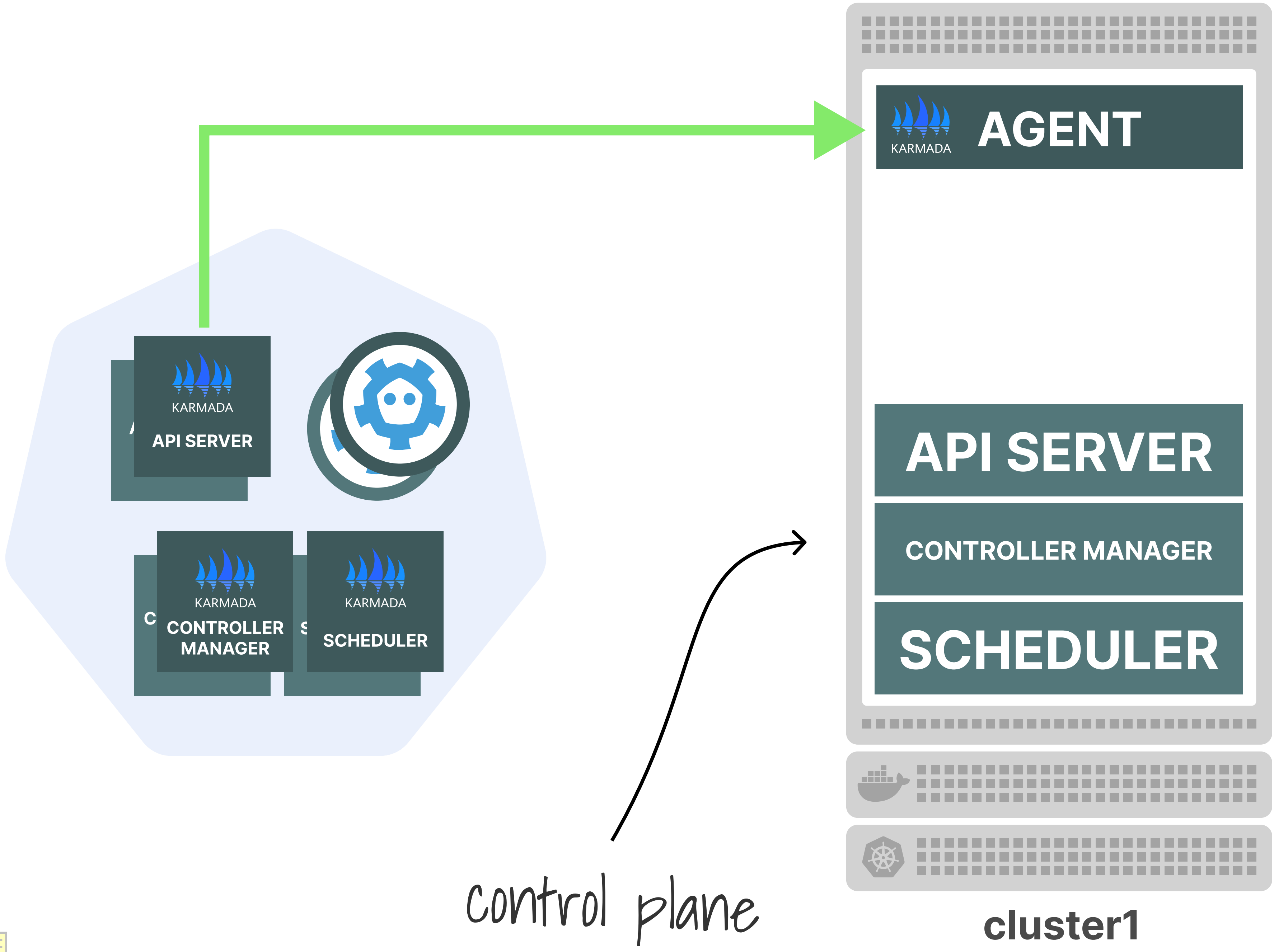
API SERVER

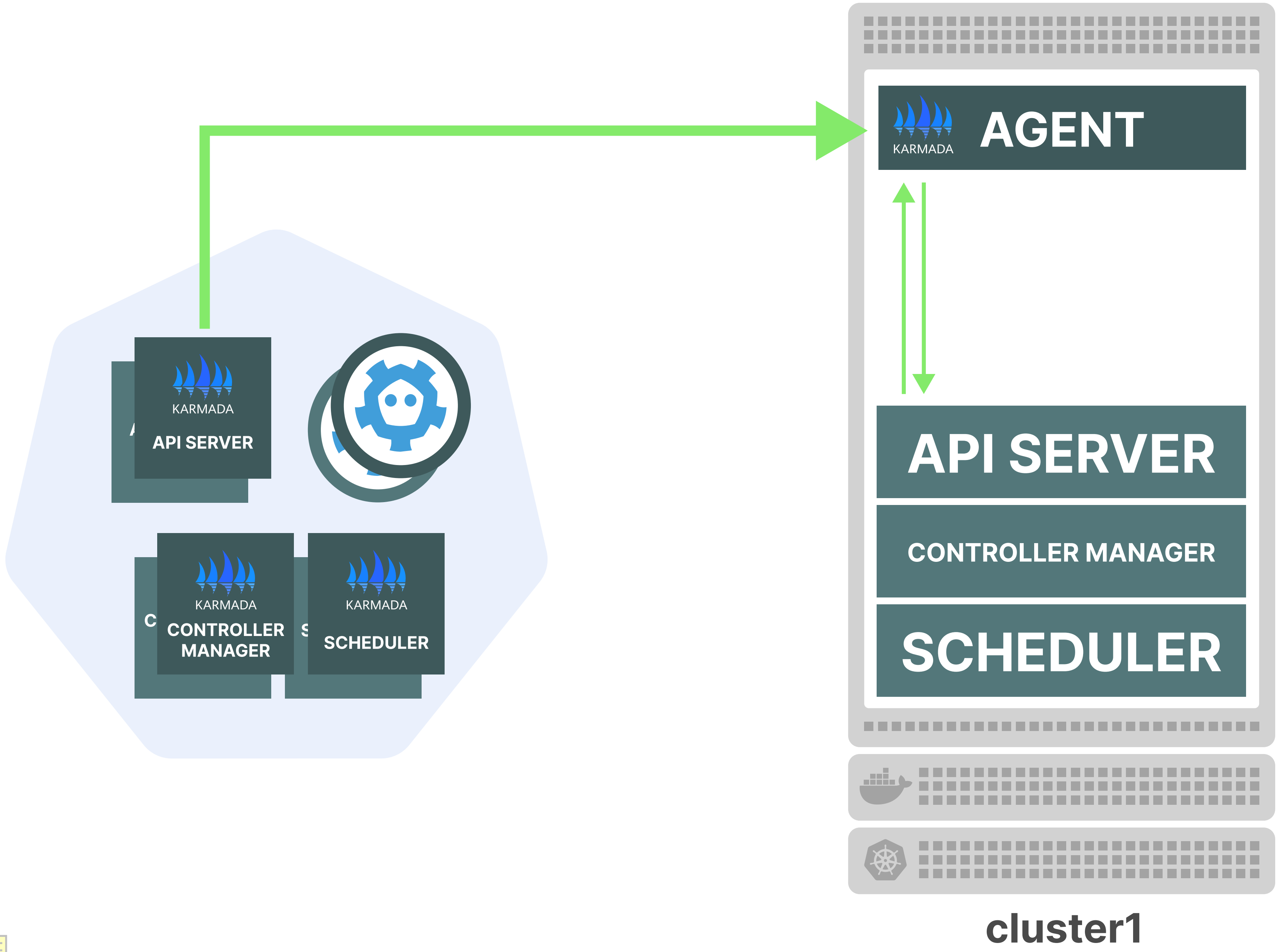
CONTROLLER MANAGER

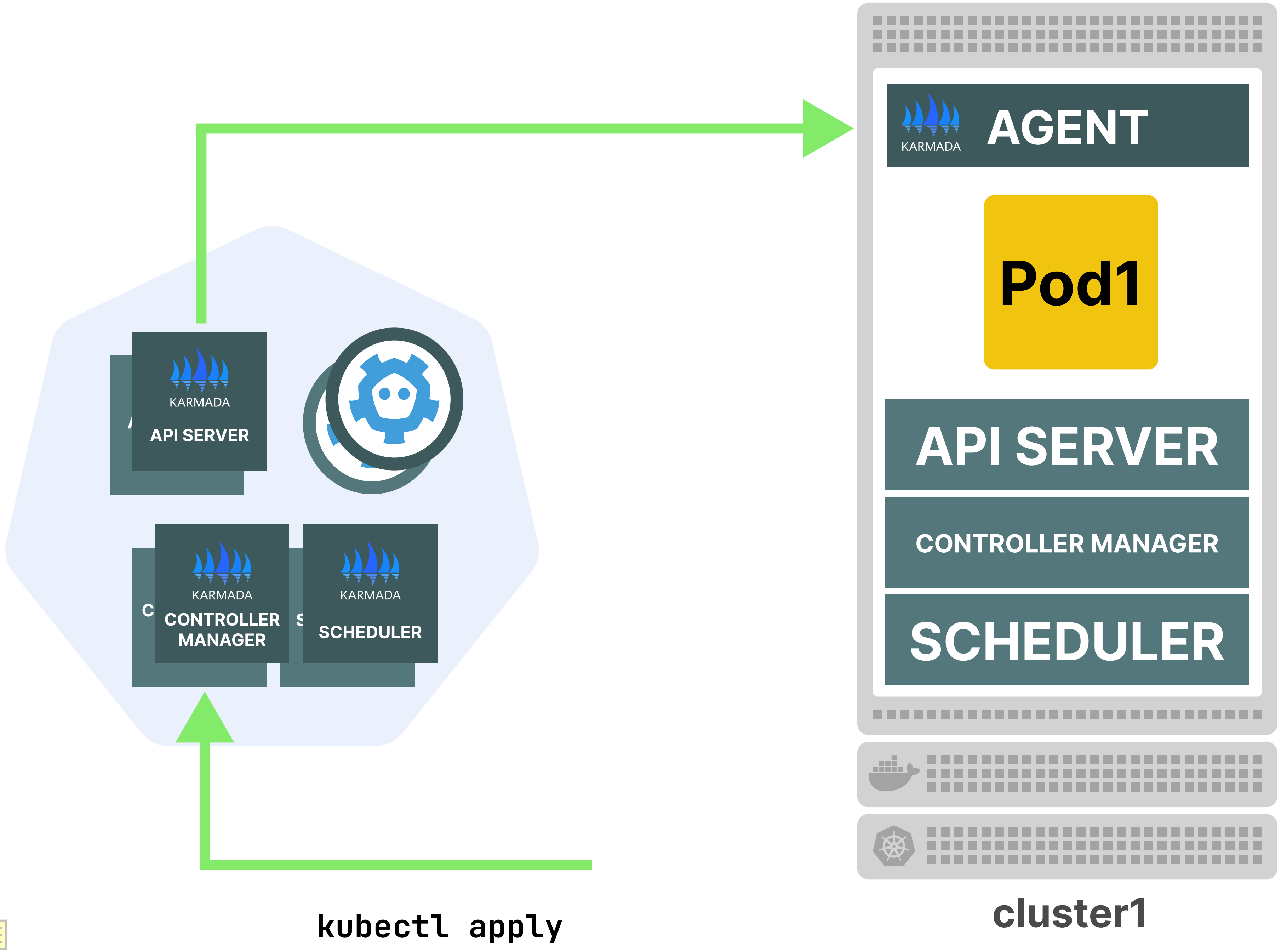
SCHEDULER

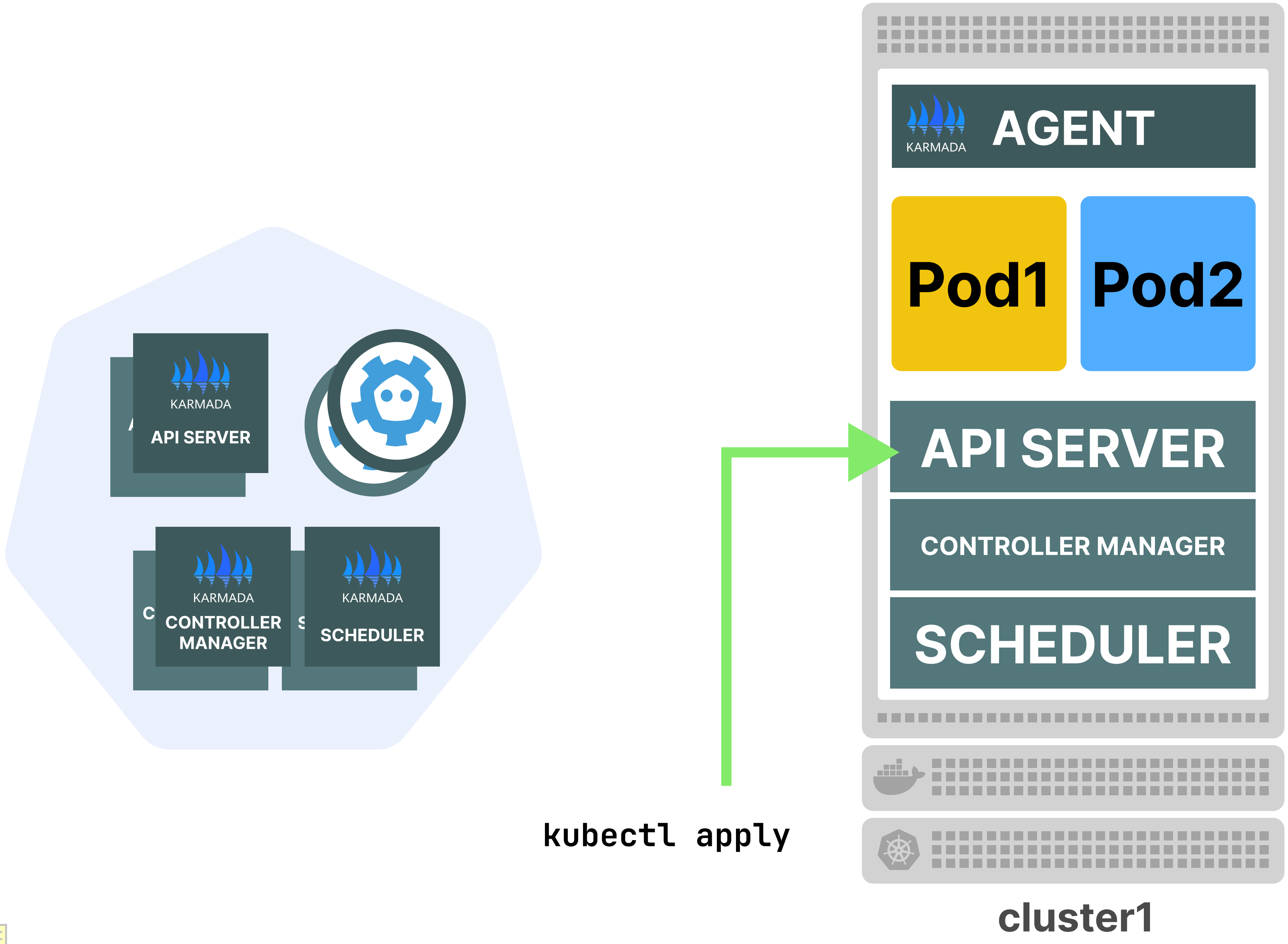

KARMADA
API SERVER
KARMADA
CONTROLLER
MANAGER
KARMADA
SCHEDULER

# Karmada architecture


KARMADA
API SERVER
KARMADA
CONTROLLER MANAGER
KARMADA
SCHEDULER
KARMADA
AGENT
API SERVER
CONTROLLER MANAGER
SCHEDULER
control plane
cluster1


KARMADA
API SERVER
KARMADA
CONTROLLER MANAGER
KARMADA
SCHEDULER
KARMADA
AGENT
API SERVER
CONTROLLER MANAGER
SCHEDULER
cluster1


KARMADA
AGENT
Pod1
API SERVER
CONTROLLER MANAGER
SCHEDULER
KARMADA
API SERVER
KARMADA
CONTROLLER MANAGER
KARMADA
SCHEDULER
kubectl apply
cluster1



KARMADA

API SERVER

KARMADA

CONTROLLER MANAGER

KARMADA

SCHEDULER

KARMADA

AGENT

Pod1

Pod2

API SERVER

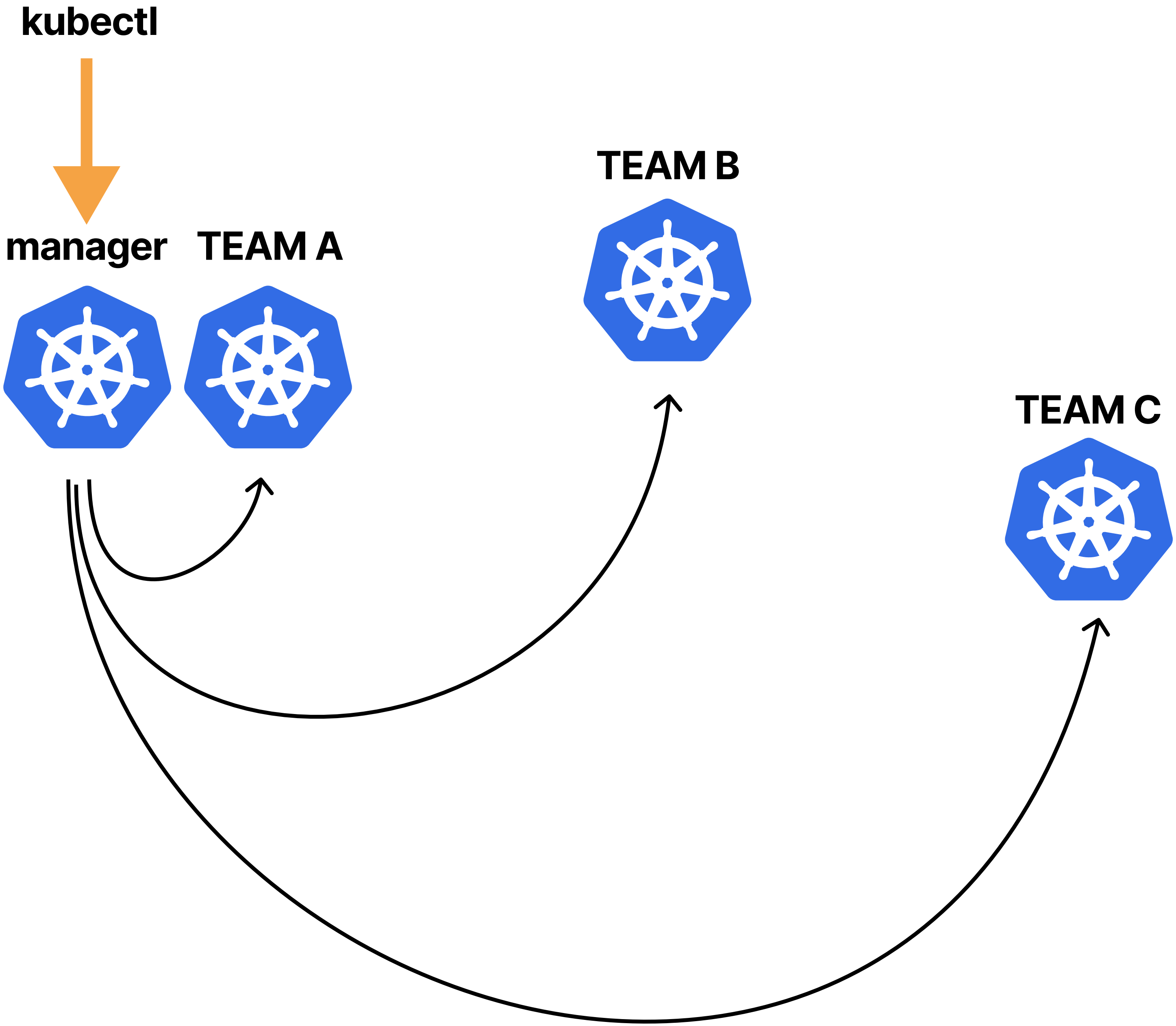
CONTROLLER MANAGER

SCHEDULER

kubectl apply

cluster1

# Independent cluster with central management



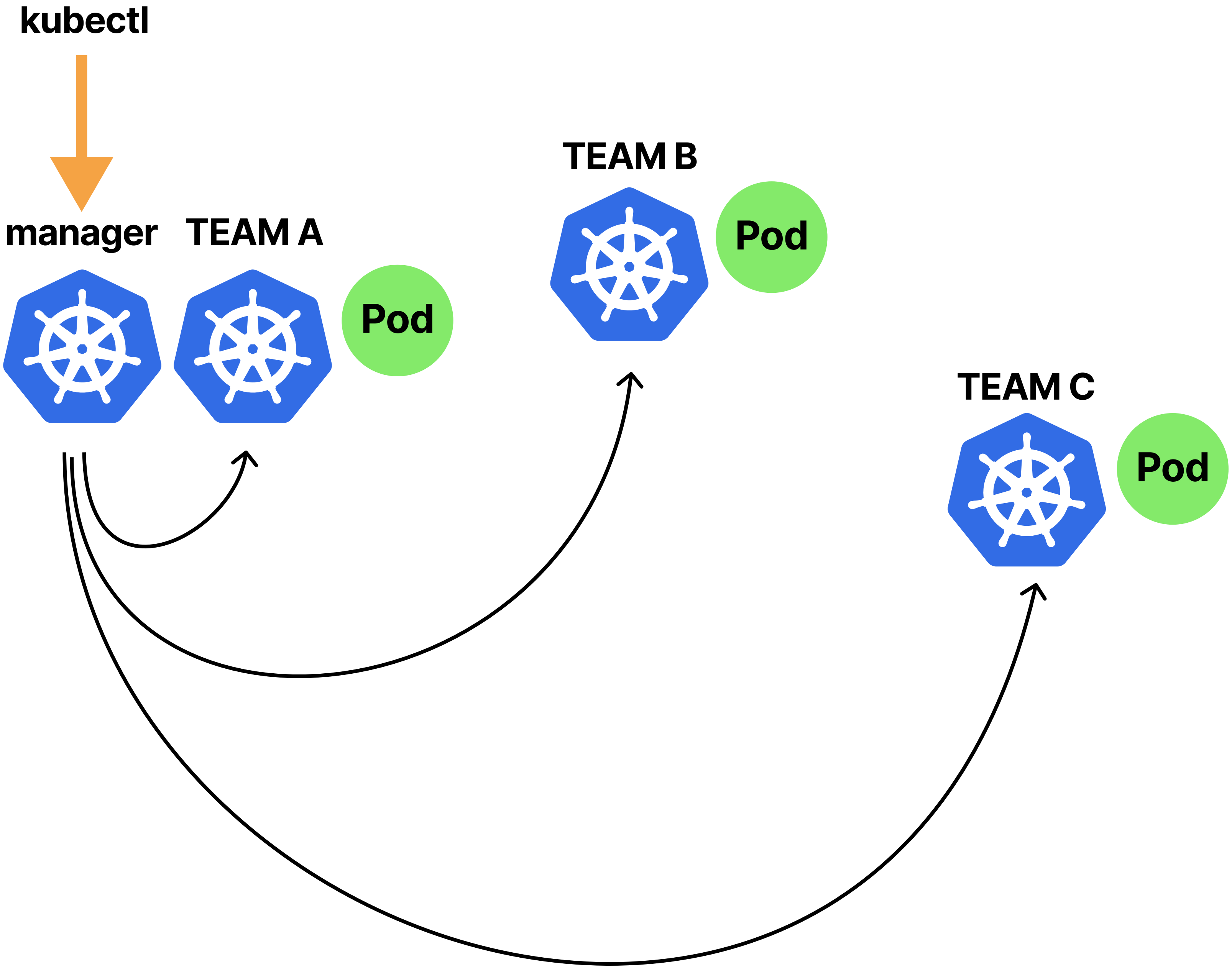
kubectl

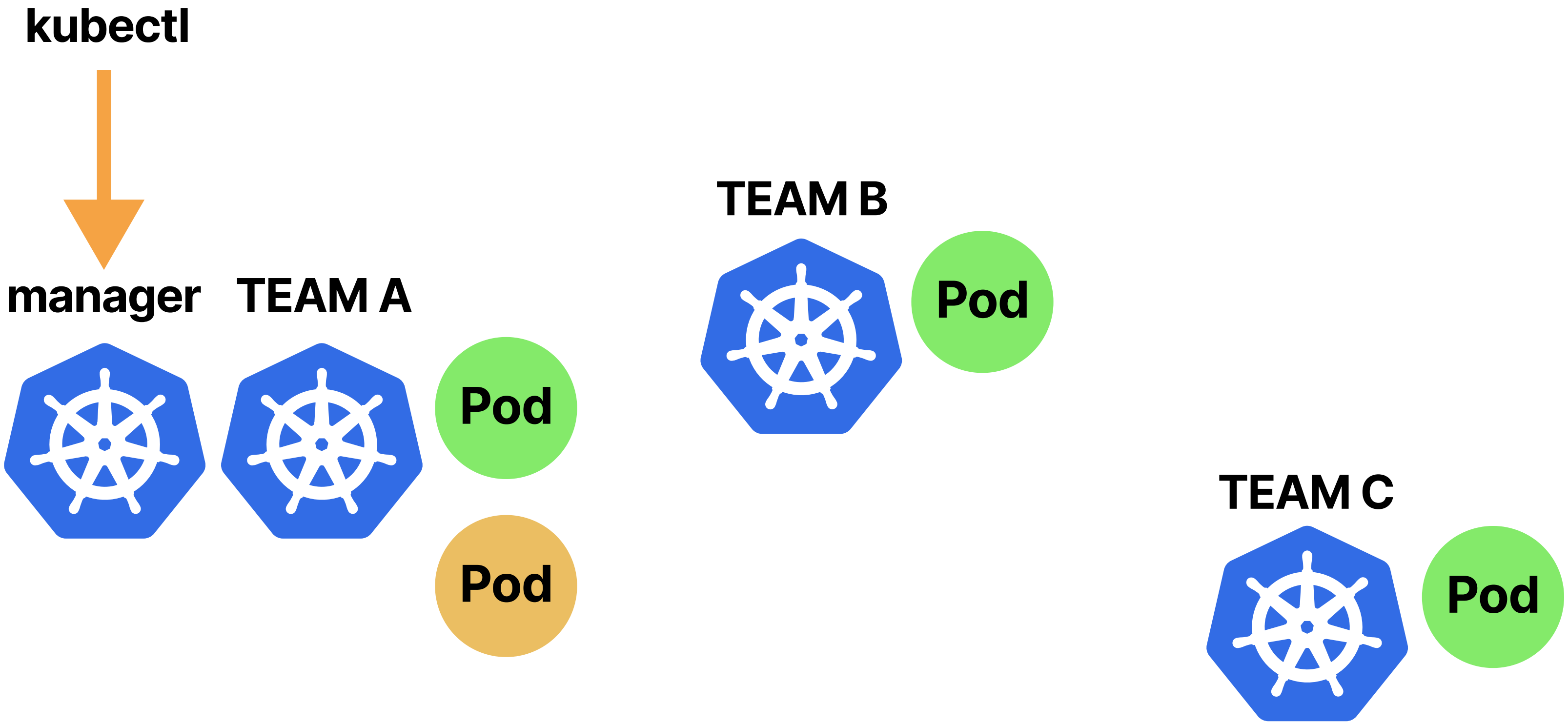
manager

TEAM A

TEAM B

TEAM C


kubectl
manager
TEAM A
Pod
TEAM B
Pod
TEAM C
Pod



kubectl

manager

TEAM A

Pod

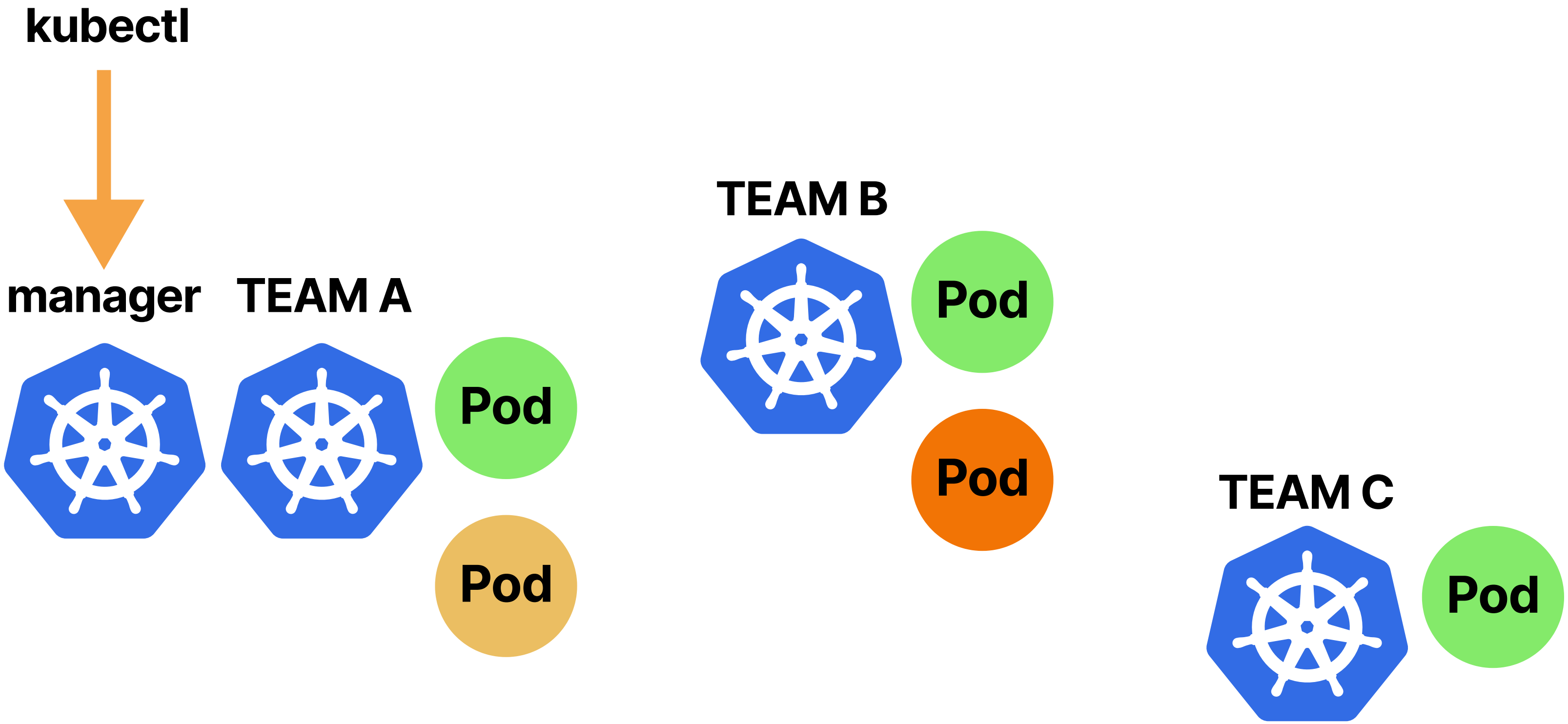
Pod

TEAM B

Pod

TEAM C

Pod



kubectl

manager

TEAM A

Pod

Pod

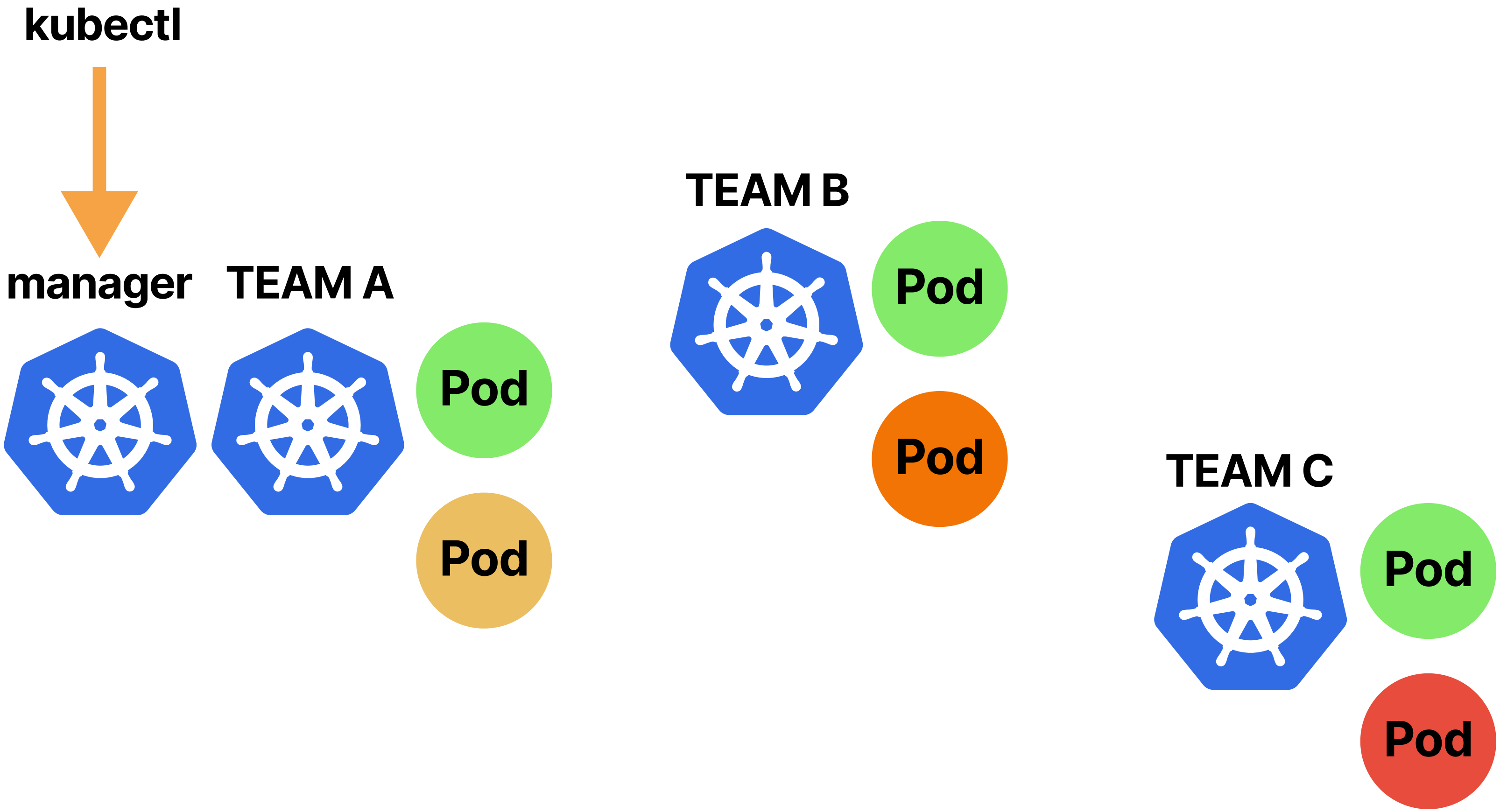
TEAM B

Pod

Pod

TEAM C

Pod



kubectl

manager

TEAM A

Pod

Pod

TEAM B

Pod

Pod

TEAM C

Pod

Pod

# Demo

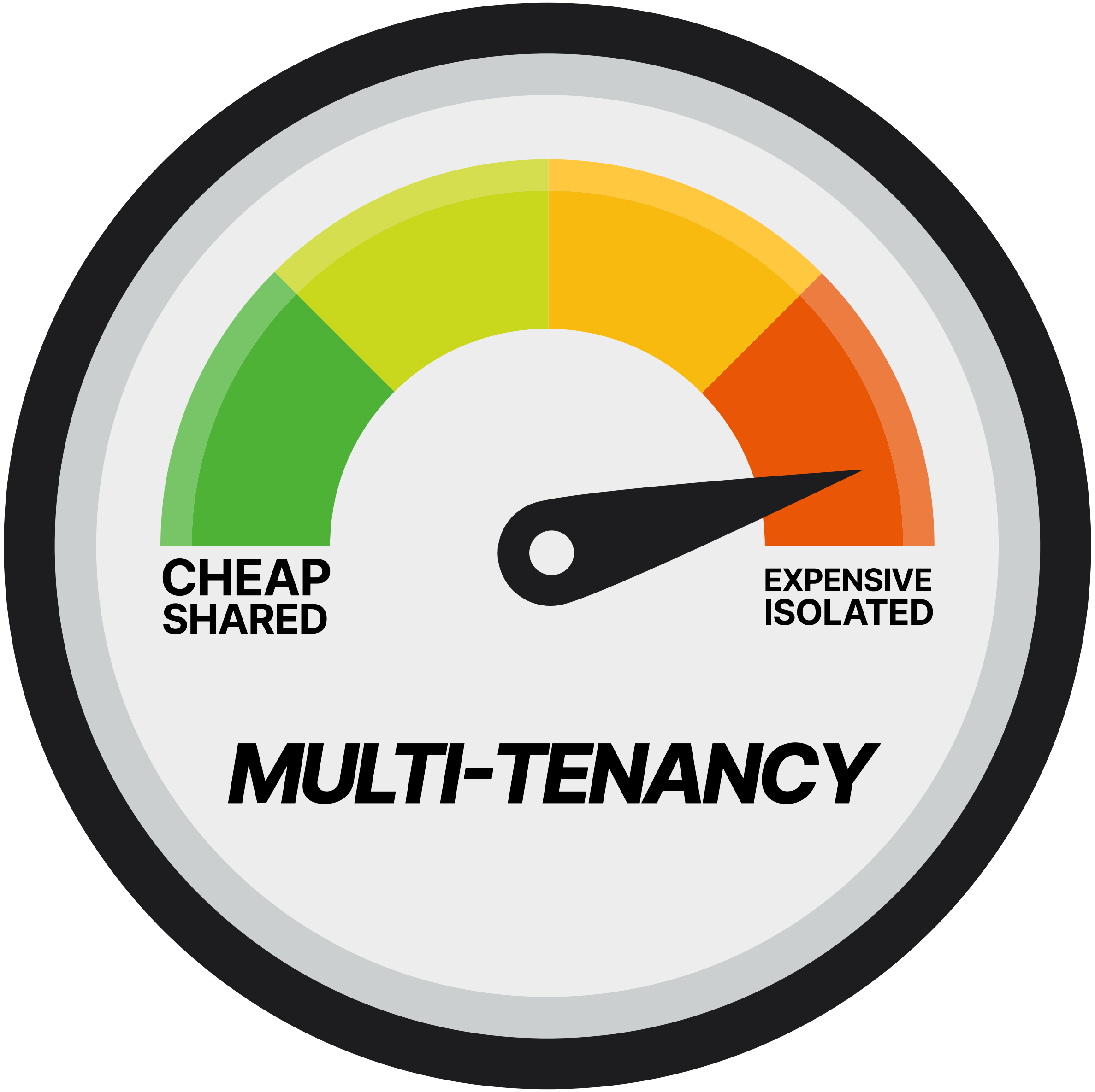
# Cluster of clusters

Karmada

# Cluster of clusters
# Admin vs tenants

## Karmada

# Cluster of clusters
# Admin vs tenants
# No sharing


CHEAP
SHARED
EXPENSIVE
ISOLATED
MULTI-TENANCY

## COSTS FOR 50 TENANTS

# + 51 clusters x $0

## COSTS FOR 50 TENANTS

**+ 51 clusters x $0**

**+ 51 nodes x $12**

# COSTS FOR 50 TENANTS

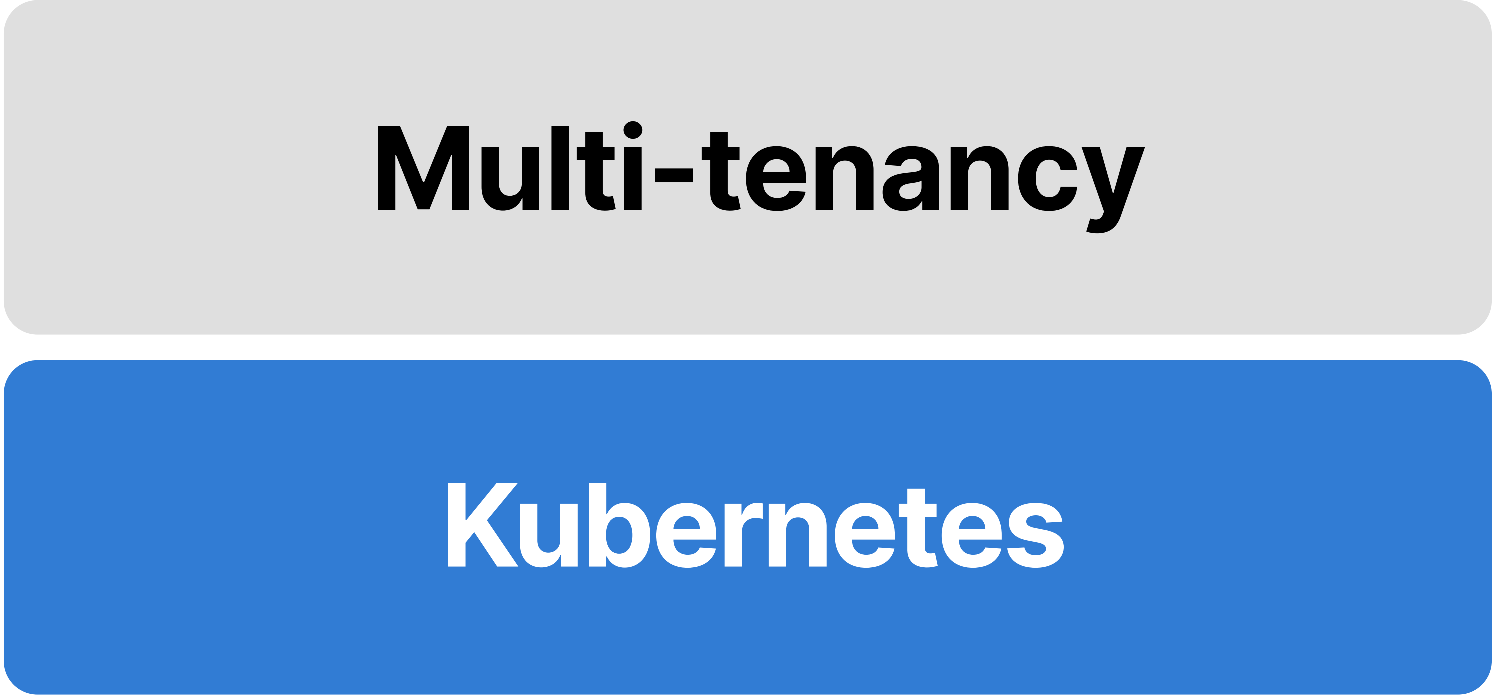
+ 51 clusters x $0

+ 51 nodes x $12

= $612 / month

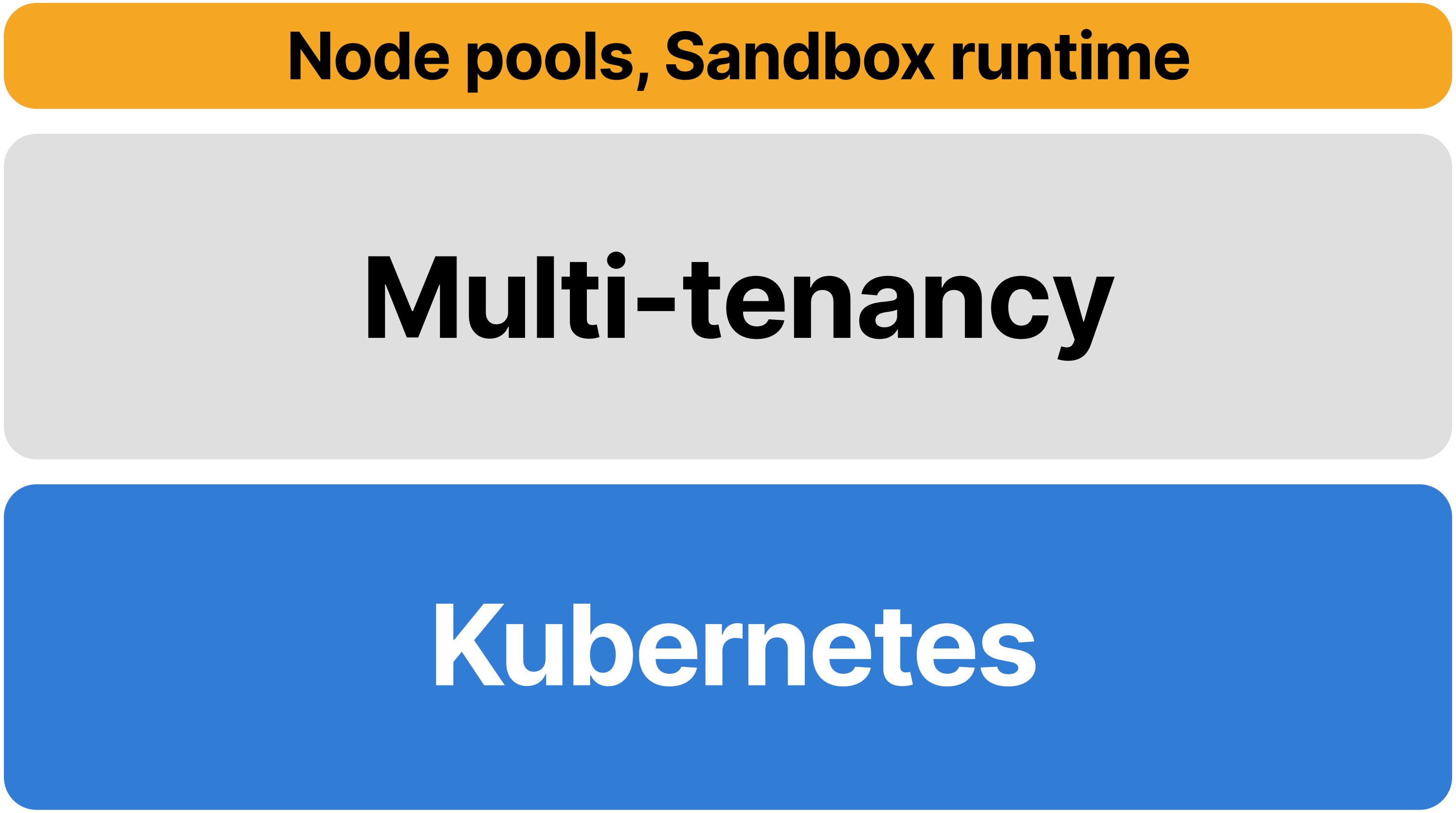
~$12 / month / tenant

# Multi-tenancy baseline



Multi-tenancy

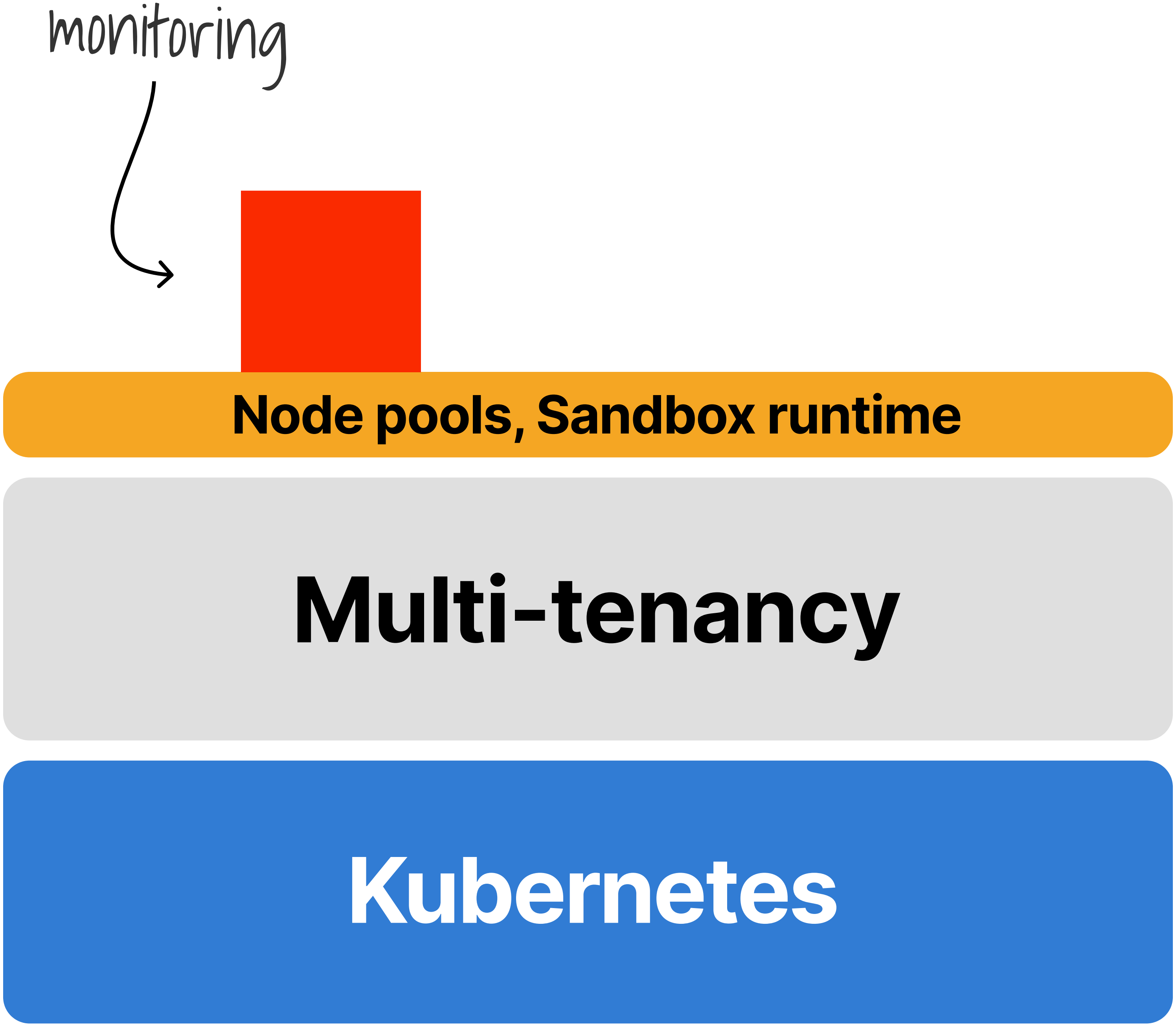
Kubernetes



Node pools, Sandbox runtime

Multi-tenancy

Kubernetes

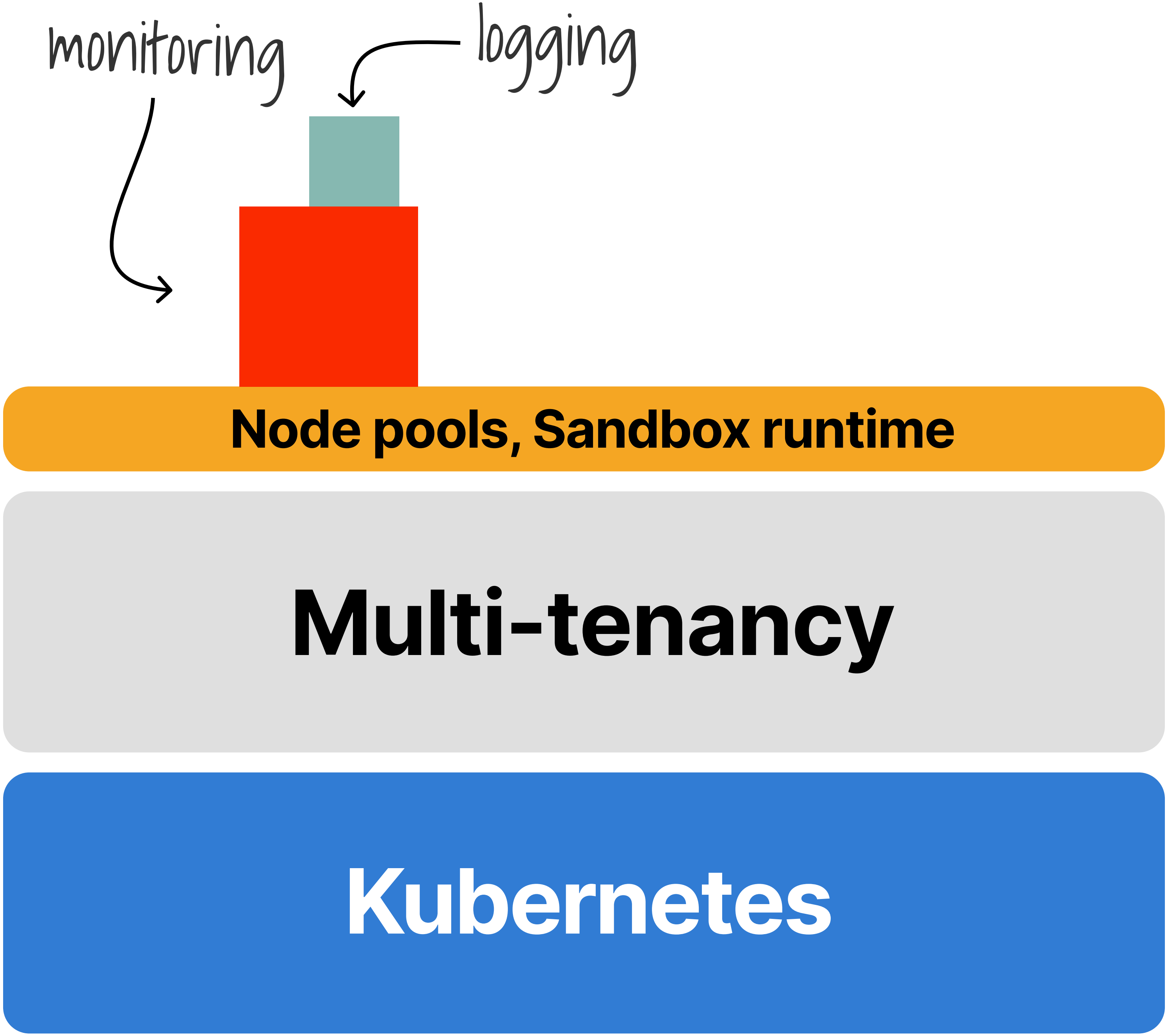


monitoring

Node pools, Sandbox runtime

Multi-tenancy

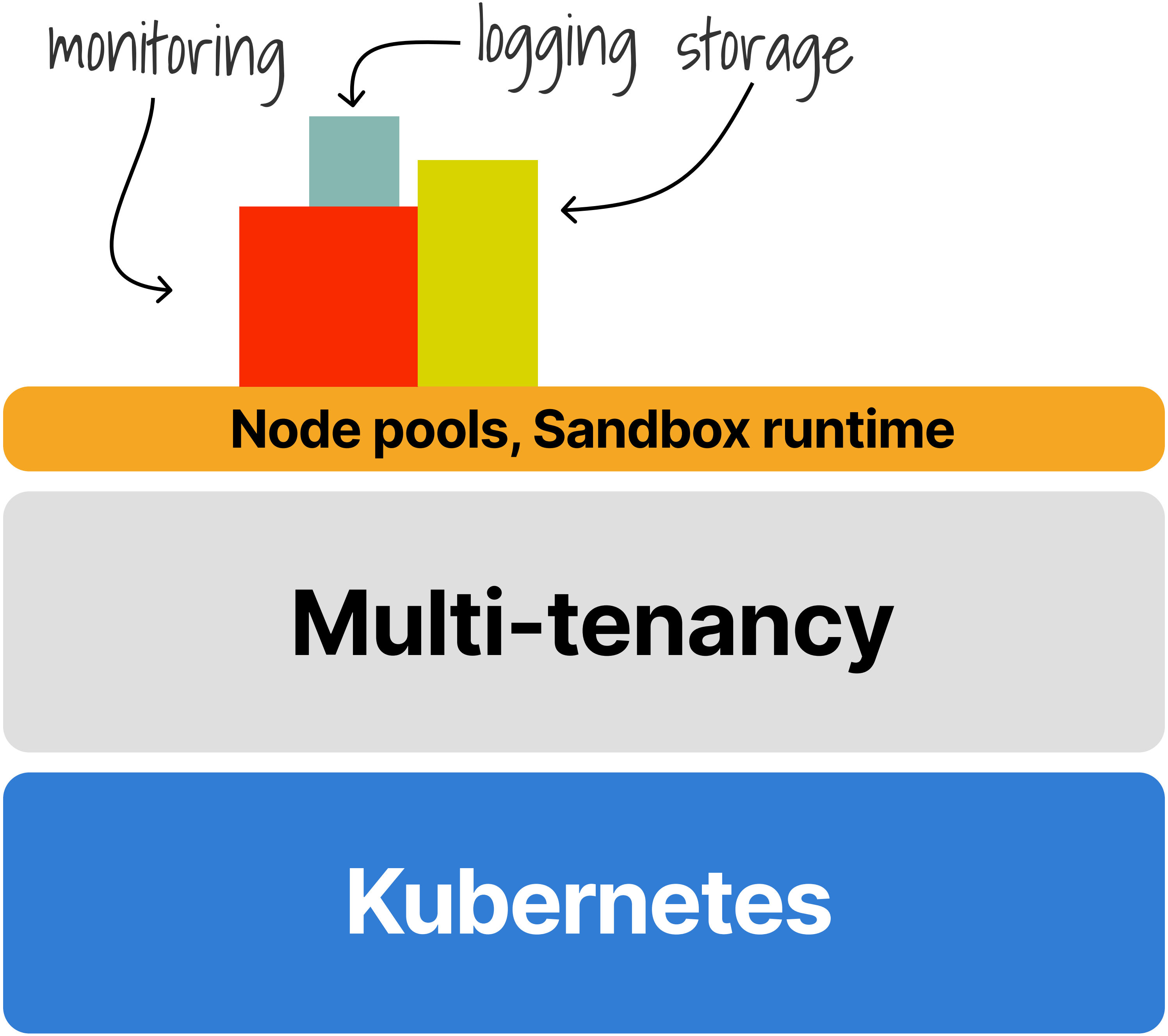
Kubernetes



Node pools, Sandbox runtime

Multi-tenancy

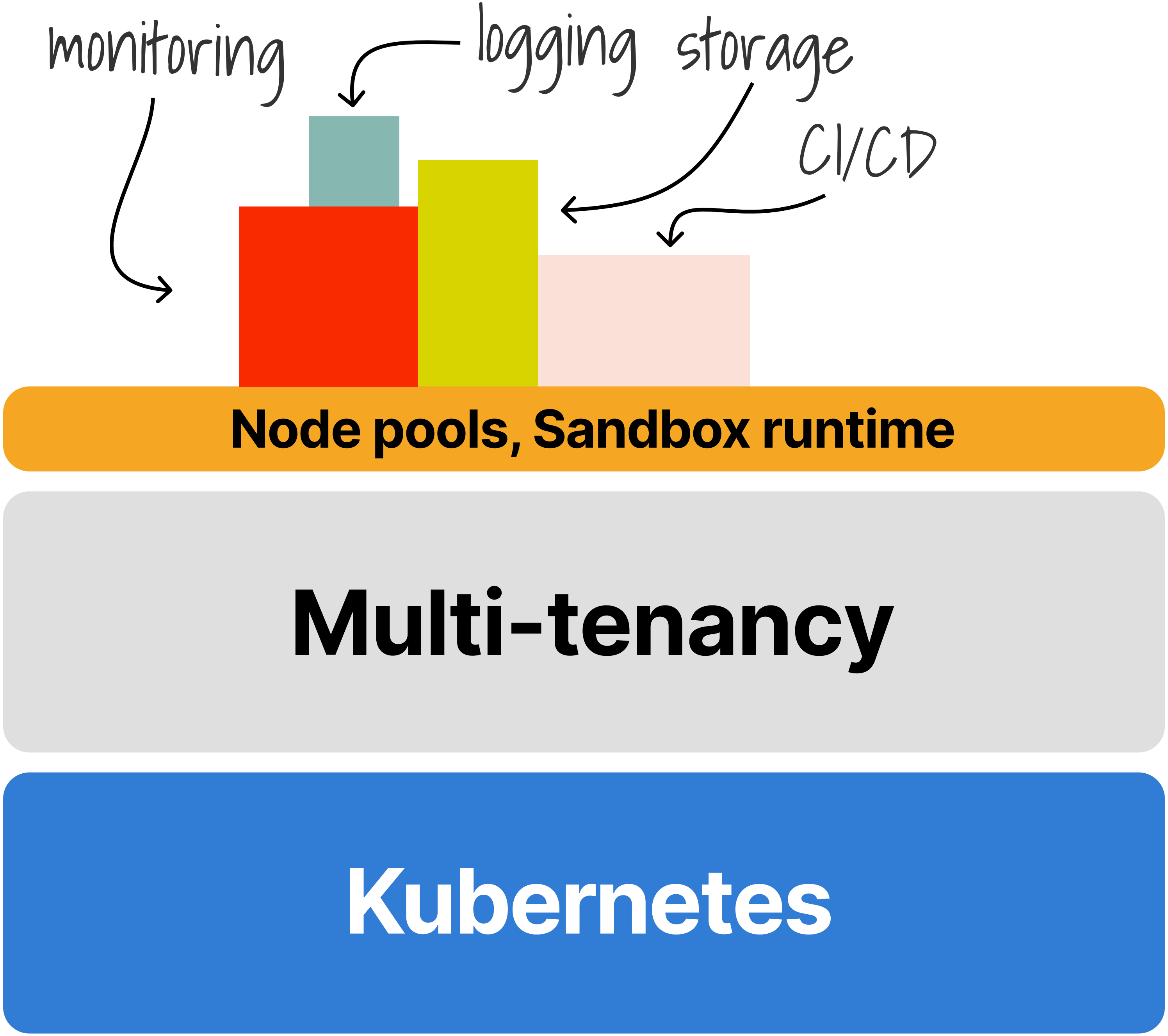
Kubernetes



Node pools, Sandbox runtime

Multi-tenancy

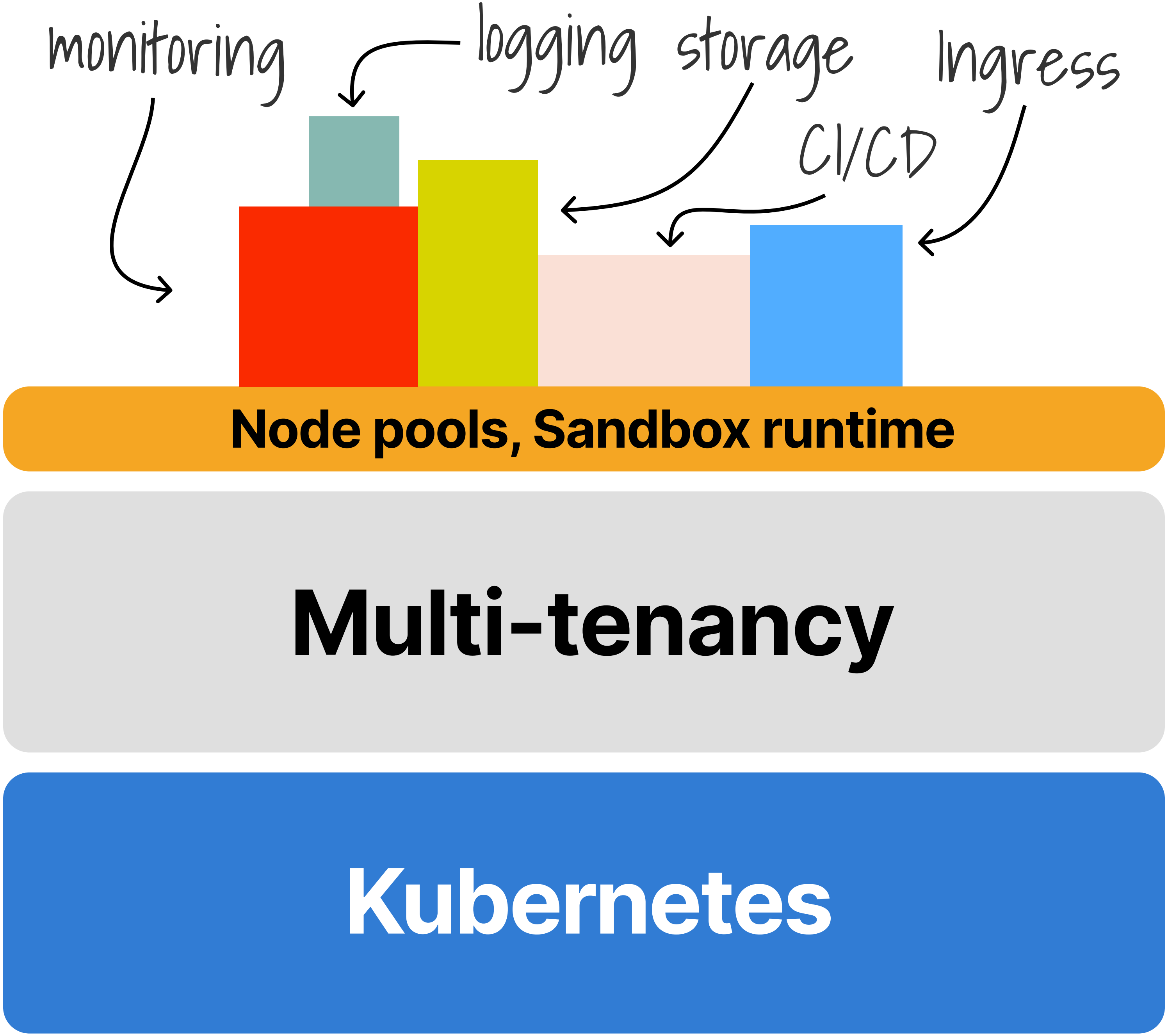
Kubernetes



Node pools, Sandbox runtime

Multi-tenancy

Kubernetes

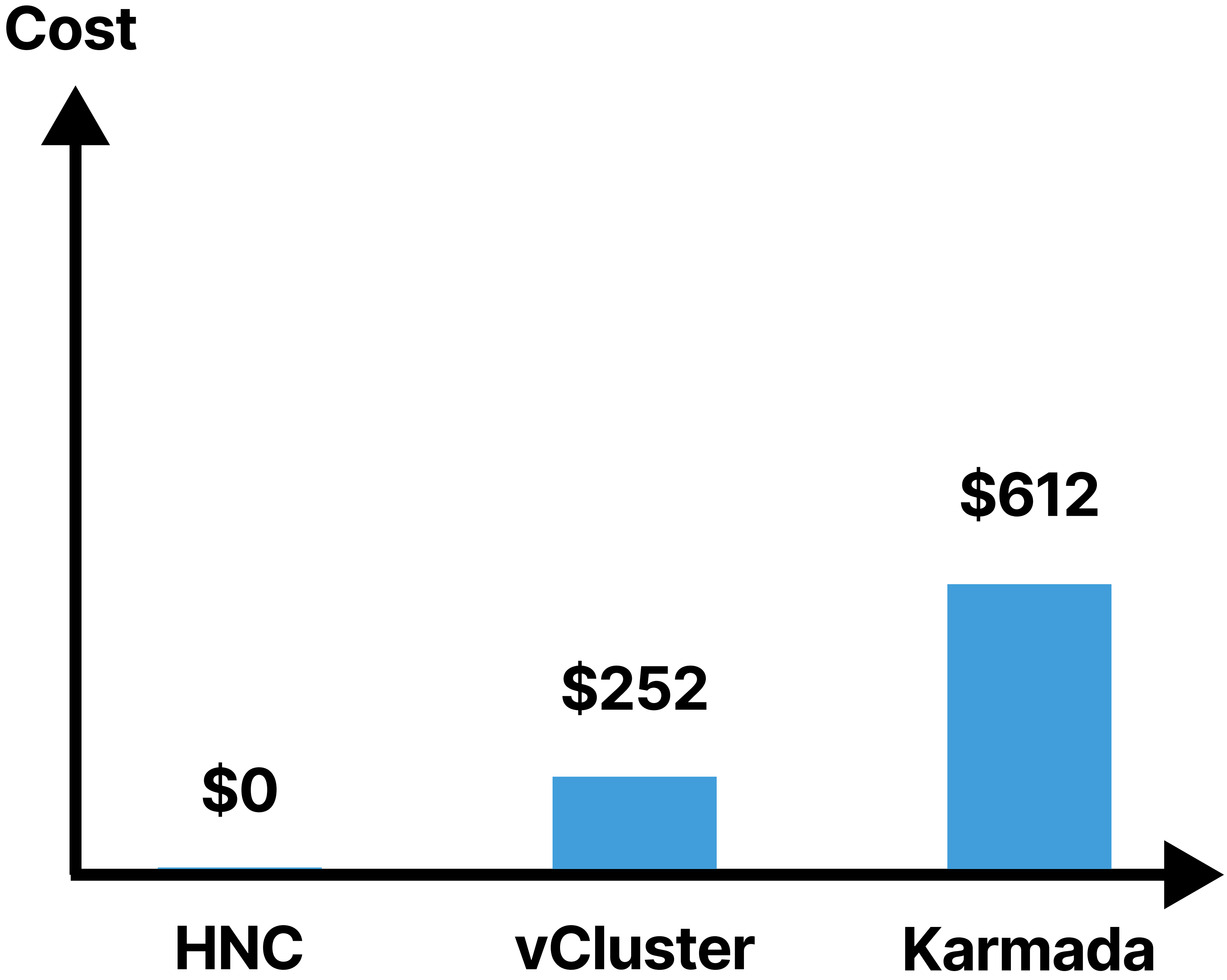


Node pools, Sandbox runtime

Multi-tenancy

Kubernetes

# Costs


Cost
$0
$252
$612
HNC
vCluster
Karmada

# DEDICATED INGRESS FOR 50 TENANTS

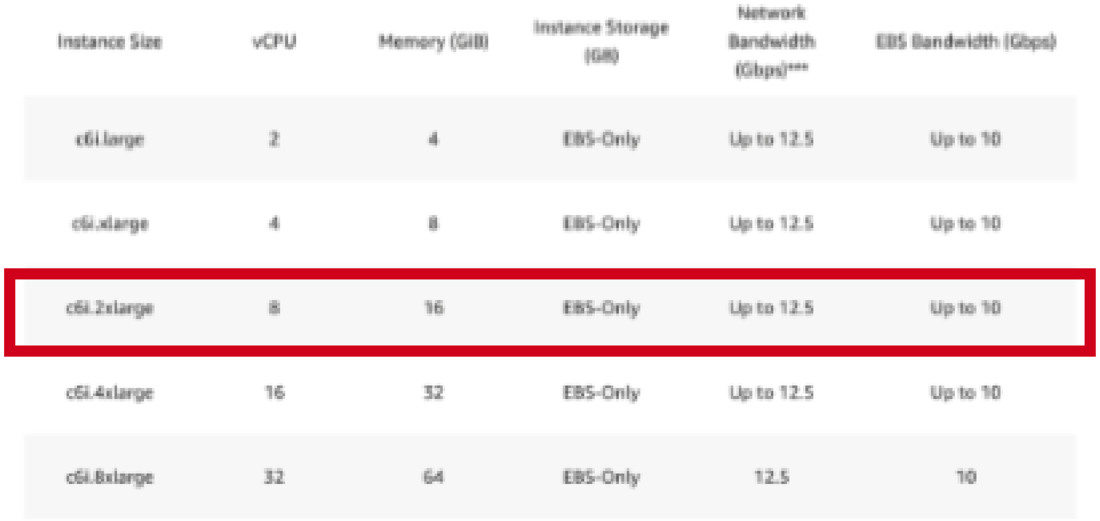
50 × 3

CPU

5vCPU

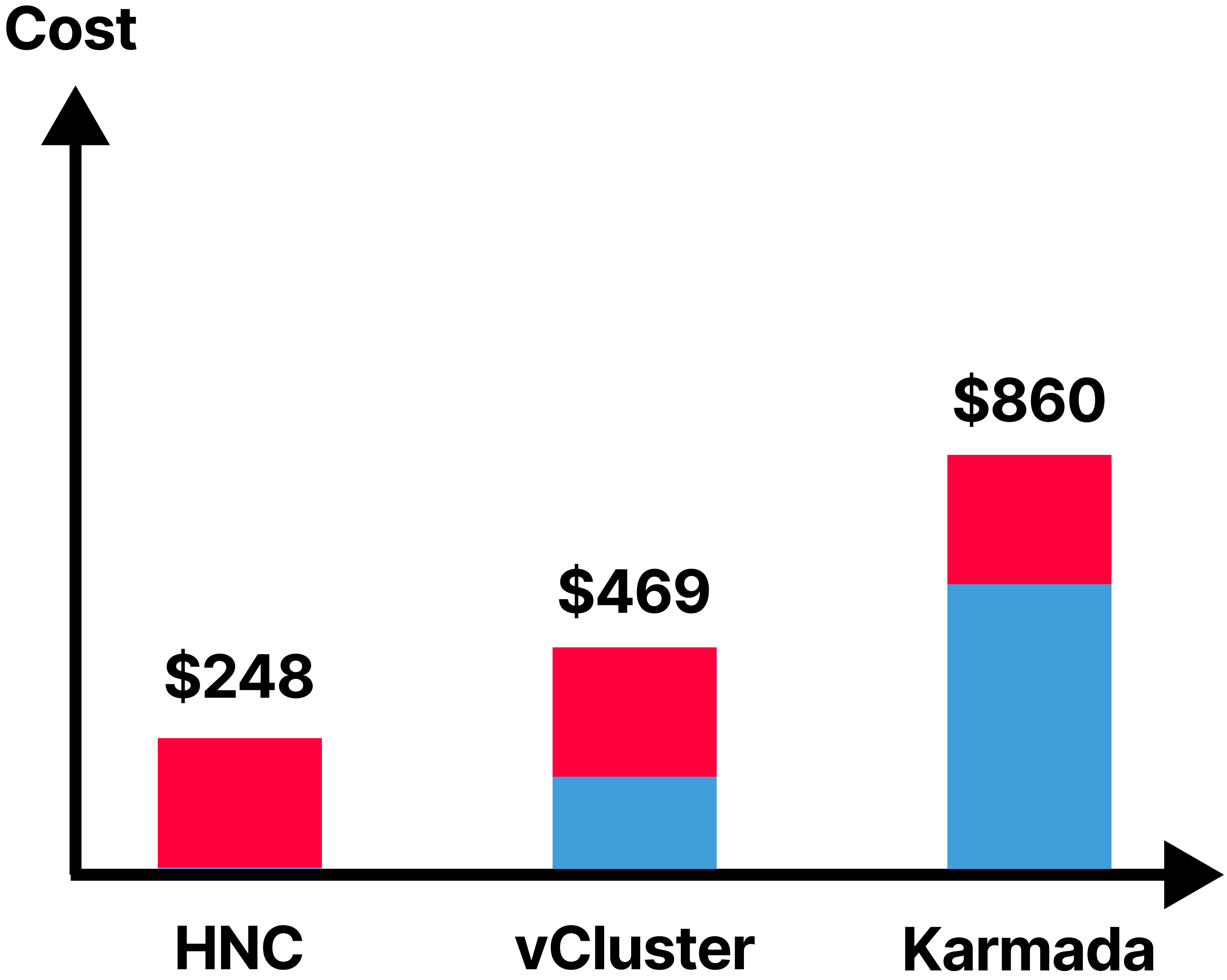
MEMORY

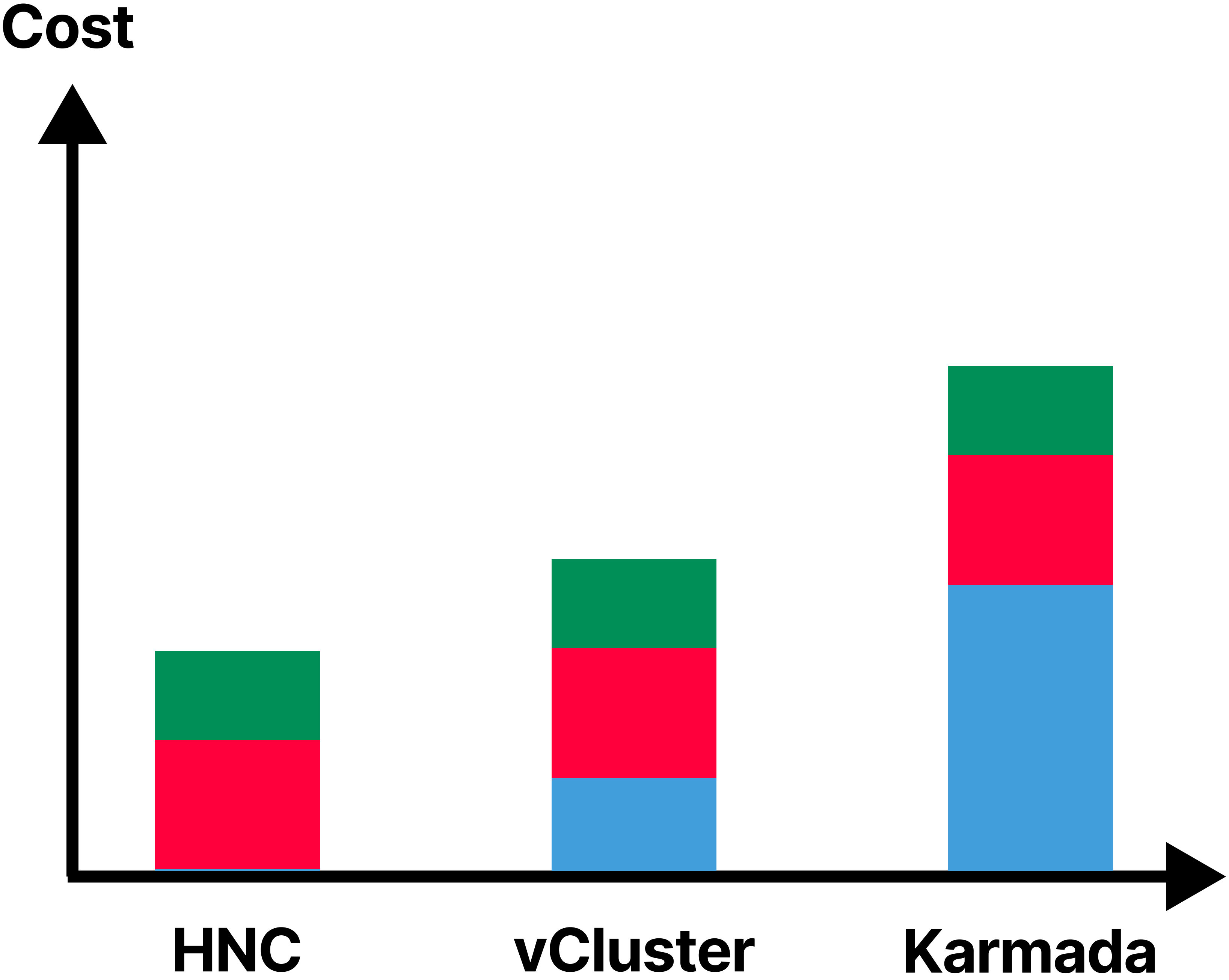
4.5GB

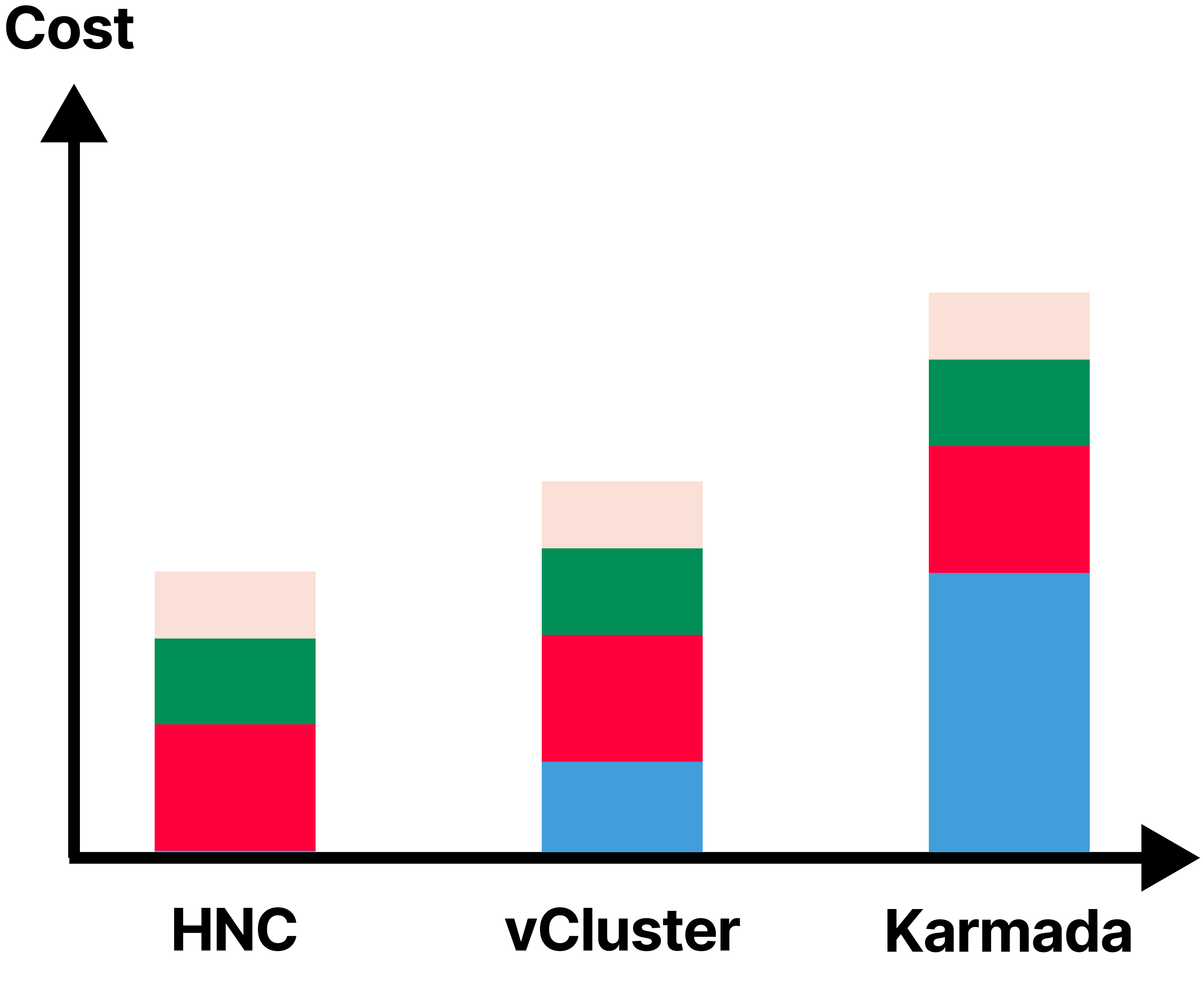


| Instance Size | vCPU | Memory (GiB) | Instance Storage (GB) | Network Bandwidth (Gbps)*** | EBS Bandwidth (Gbps) |
|---|---|---|---|---|---|
| c6i.large | 2 | 4 | EBS-Only | Up to 12.5 | Up to 10 |
| c6i.xlarge | 4 | 8 | EBS-Only | Up to 12.5 | Up to 10 |
| c6i.2xlarge | 8 | 16 | EBS-Only | Up to 12.5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

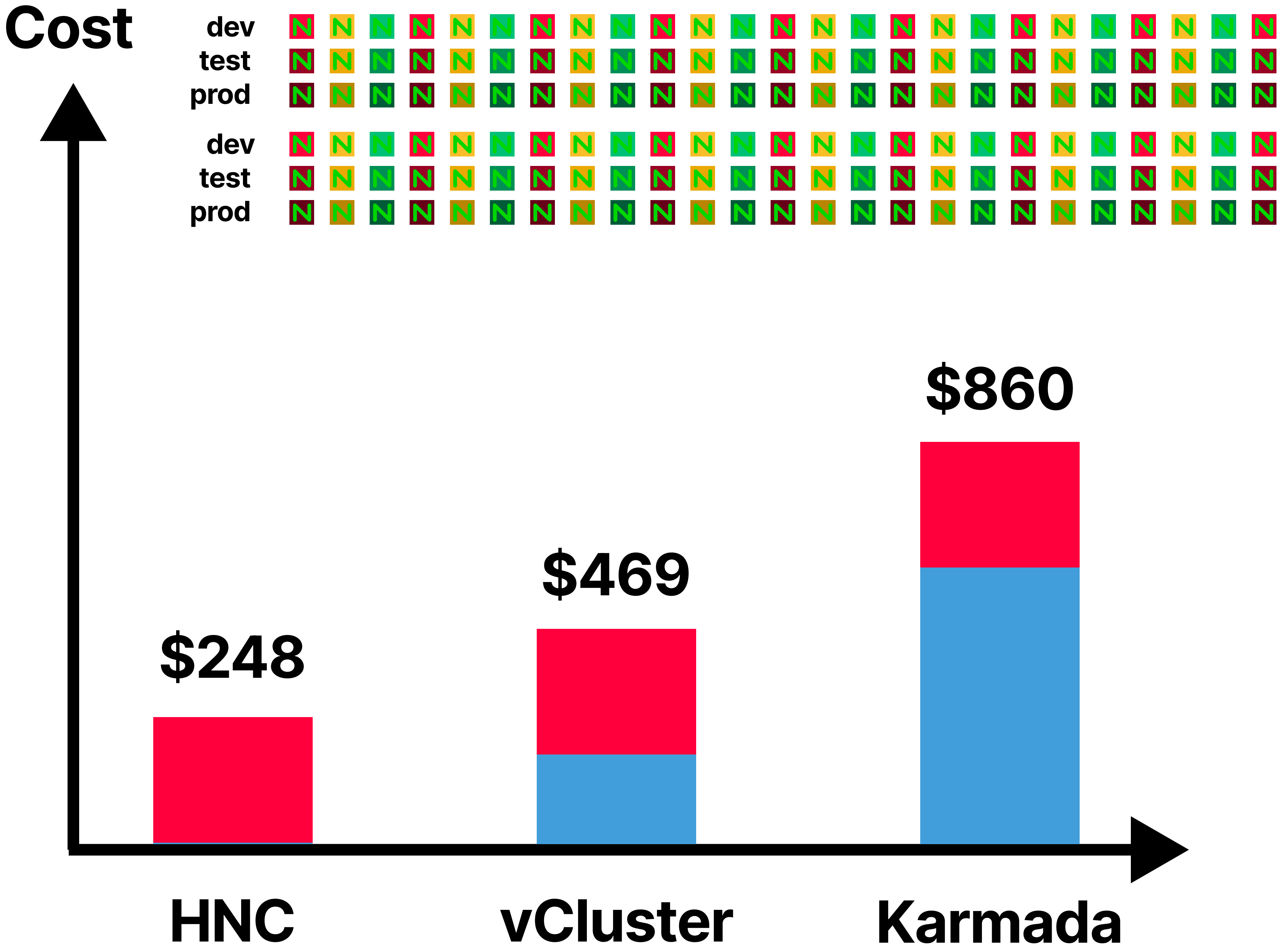
$0.34/hr $248.2/m

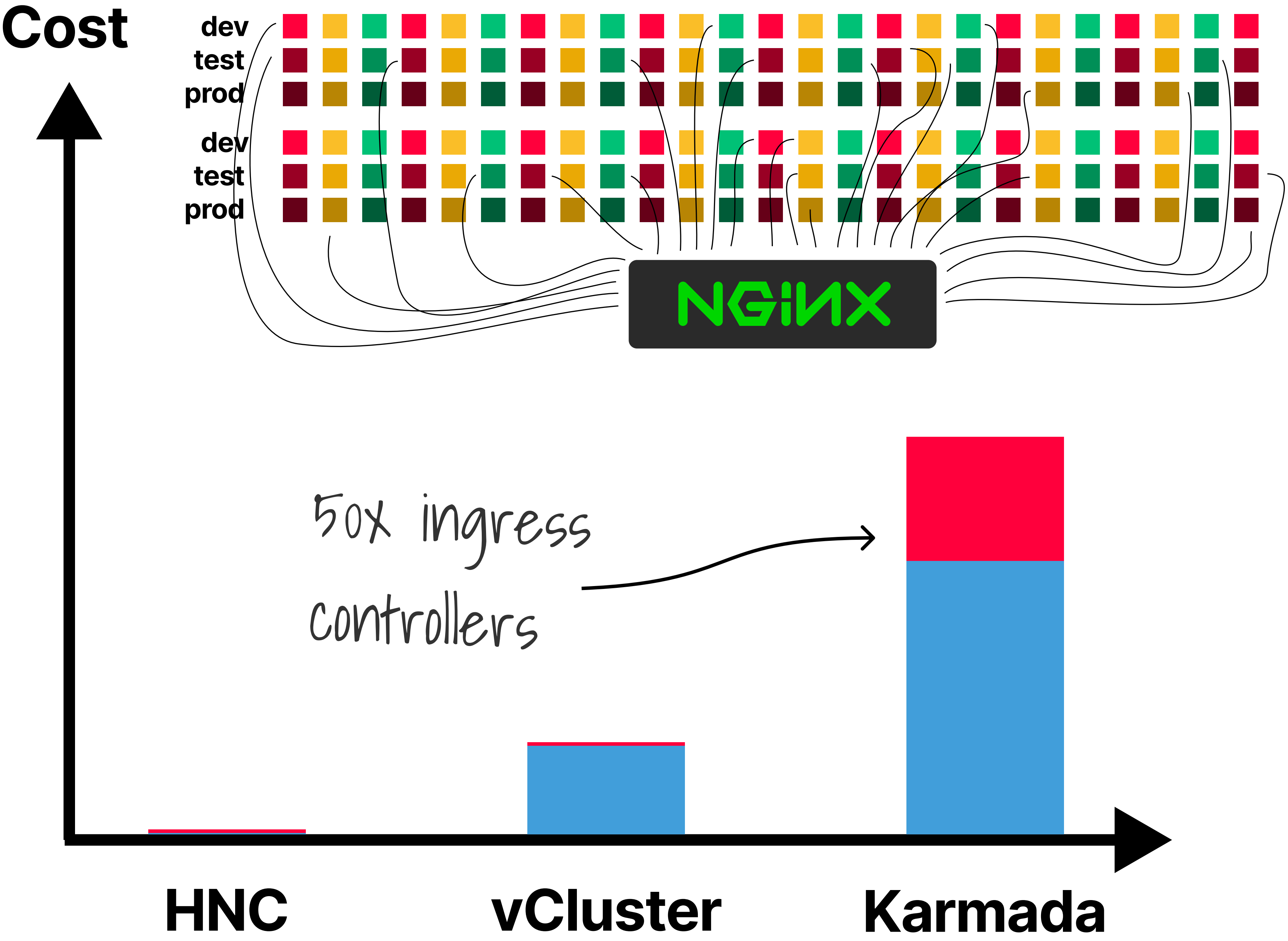

Cost
$248
$469
$860
HNC
vCluster
Karmada

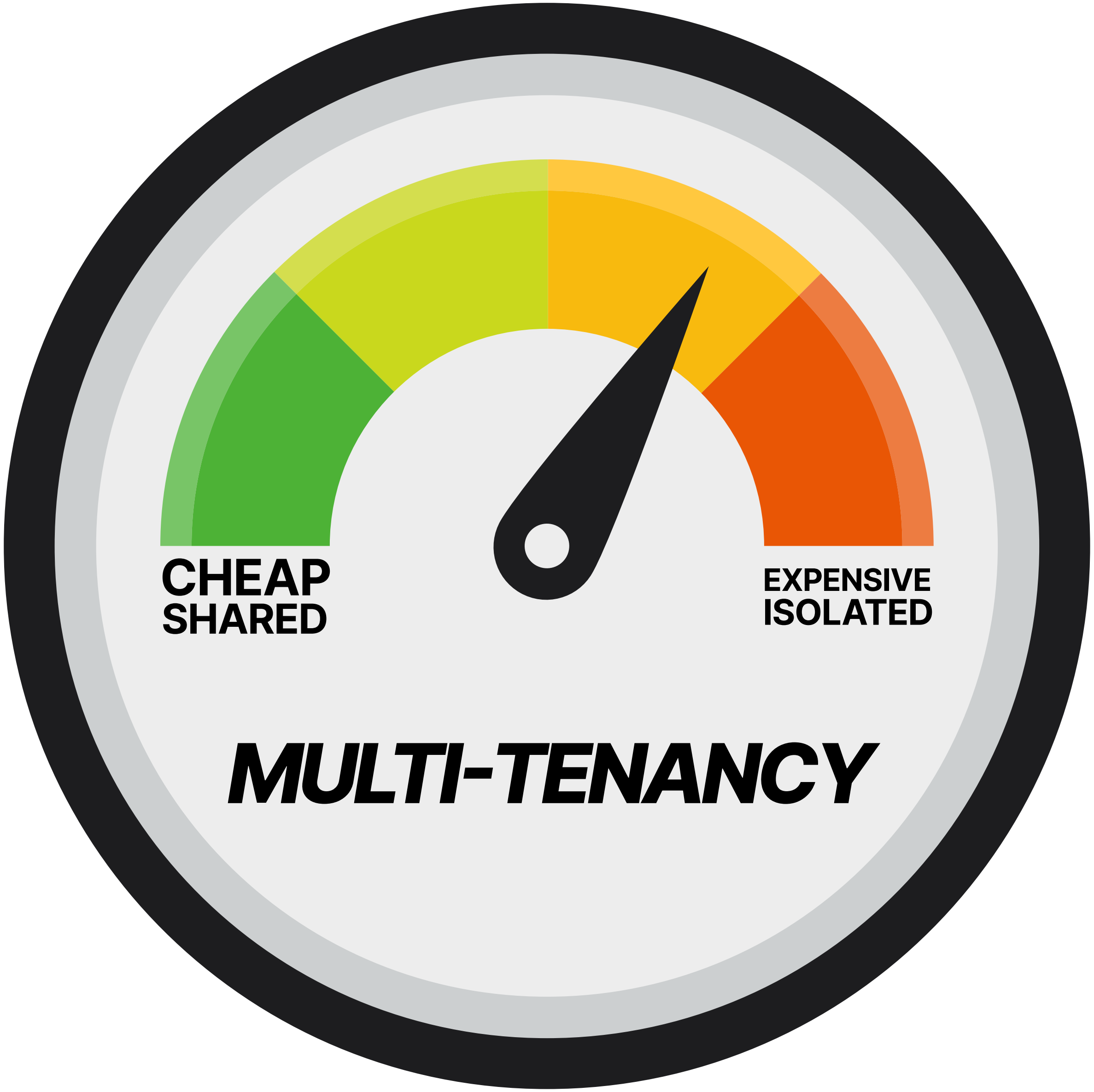

Cost
HNC
vCluster
Karmada


Cost
HNC
vCluster
Karmada

# Costs*


Cost
dev N N N N N N N N N N N N N N N N N N N N N N N N N
test N N N N N N N N N N N N N N N N N N N N N N N N N
prod N N N N N N N N N N N N N N N N N N N N N N N N N
dev N N N N N N N N N N N N N N N N N N N N N N N N N
test N N N N N N N N N N N N N N N N N N N N N N N N N
prod N N N N N N N N N N N N N N N N N N N N N N N N N
$248
$469
$860
HNC
vCluster
Karmada


Cost
dev
test
prod
dev
test
prod
NGINX
50x ingress
controllers
HNC
vCluster
Karmada



CHEAP
SHARED
EXPENSIVE
ISOLATED
MULTI-TENANCY

# 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**

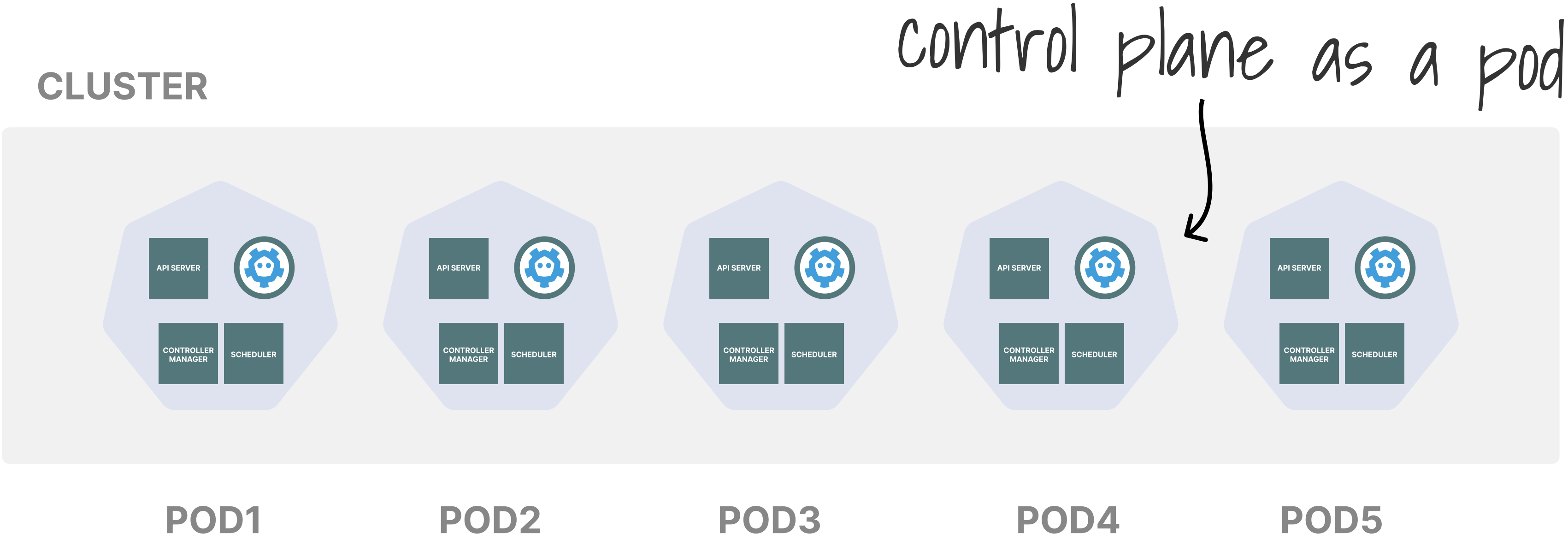
loft

# Thank you!

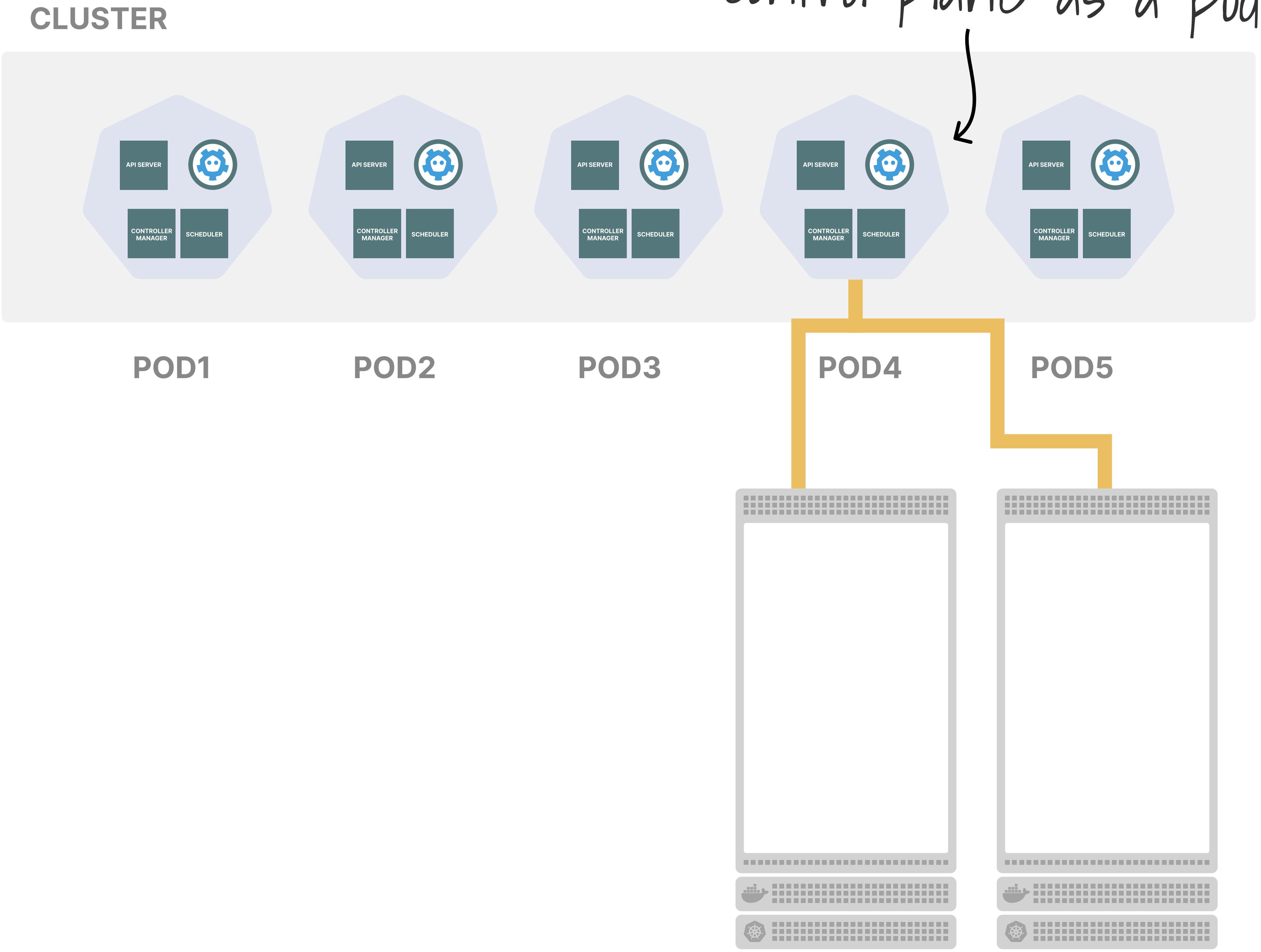
# Thank you!

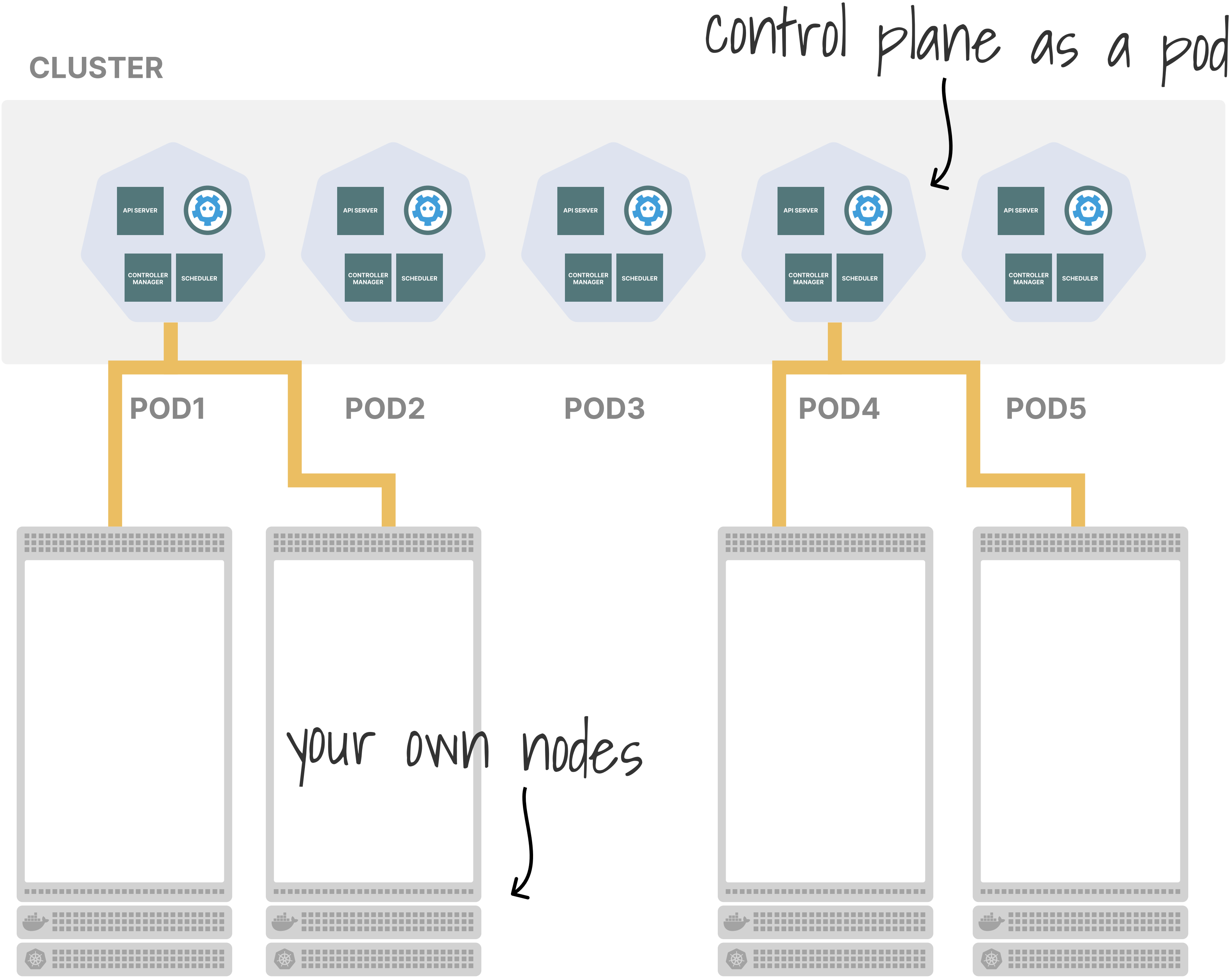
Chris Nesbitt-Smith

# Hypershift/Kamaji


control plane as a pod
CLUSTER
API SERVER
CONTROLLER MANAGER
SCHEDULER
API SERVER
CONTROLLER MANAGER
SCHEDULER
API SERVER
CONTROLLER MANAGER
SCHEDULER
API SERVER
CONTROLLER MANAGER
SCHEDULER
API SERVER
CONTROLLER MANAGER
SCHEDULER
POD1
POD2
POD3
POD4
POD5

control plane as a pod





control plane as a pod

your own nodes