Balancing Isolation and Complexity in a Kubernetes Platform

Chris Nesbitt-Smith

learnk8s

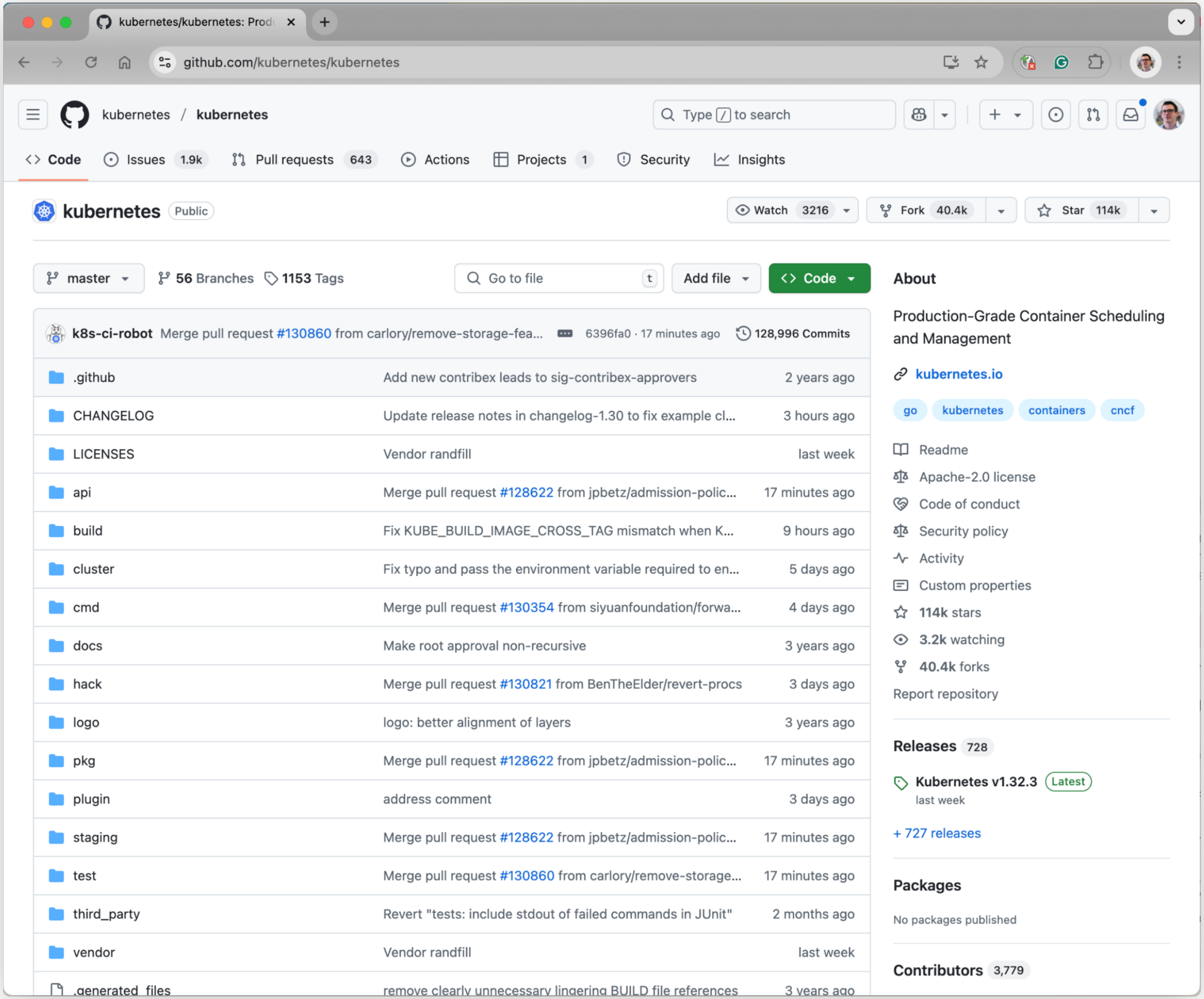
Chris Nesbitt-Smith

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

learnk8s

Case study

GitPod



k8s-ci-robot Merge pull request #130860 from carlory/remove-storage-fea... 6396fa0 · 17 minutes ago 128,996 Commits

.github	Add new contribex leads to sig-contribex-approvers	2 years ago
CHANGELOG	Update release notes in changelog-1.30 to fix example cl...	3 hours ago
LICENSES	Vendor randfill	last week
api	Merge pull request #128622 from jpbetz/admission-polic...	17 minutes ago
build	Fix KUBE_BUILD_IMAGE_CROSS_TAG mismatch when K...	9 hours ago
cluster	Fix typo and pass the environment variable required to en...	5 days ago
cmd	Merge pull request #130354 from siyuanfoundation/forwa...	4 days ago
docs	Make root approval non-recursive	3 years ago
hack	Merge pull request #130821 from BenTheElder/revert-procs	3 days ago
logo	logo: better alignment of layers	3 years ago
pkg	Merge pull request #128622 from jpbetz/admission-polic...	17 minutes ago
plugin	address comment	3 days ago
staging	Merge pull request #128622 from jpbetz/admission-polic...	17 minutes ago
test	Merge pull request #130860 from carlory/remove-storage...	17 minutes ago
third_party	Revert "tests: include stdout of failed commands in JUnit"	2 months ago
vendor	Vendor randfill	last week
.generated_files	remove clearly unnecessary lingering BUILD file references	3 years ago

About

Production-Grade Container Scheduling and Management

kubernetes.io

go kubernetes containers cncf

Readme

Apache-2.0 license

Code of conduct

Security policy

Activity

Custom properties

114k stars

3.2k watching

40.4k forks

Report repository

Releases 728

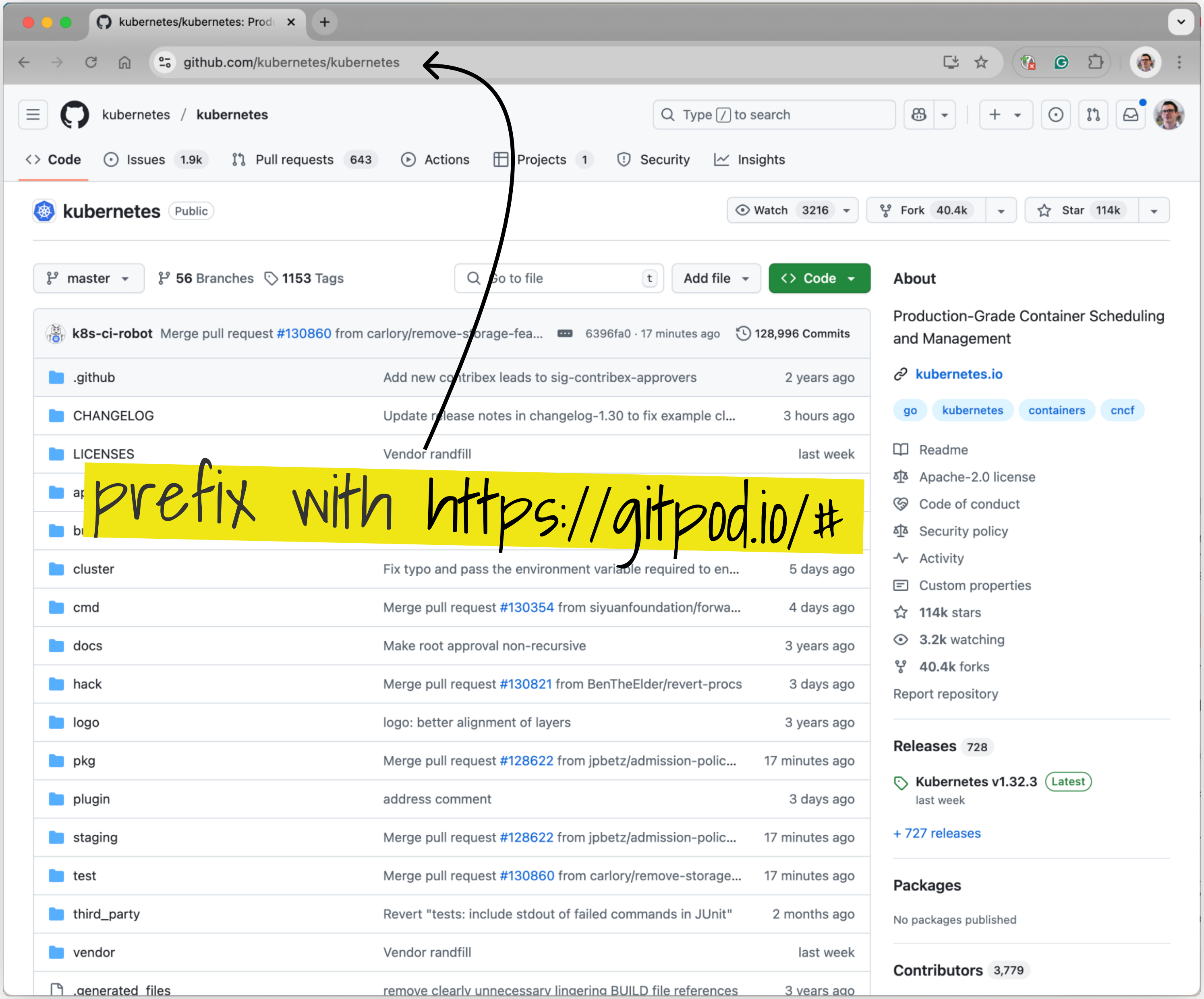
Kubernetes v1.32.3 Latest
last week

+ 727 releases

Packages

No packages published

Contributors 3,779



prefix with https://gitpod.io/#

kubernetes/kubernetes: Prod

github.com/kubernetes/kubernetes

kubernetes / kubernetes

Type / to search

Code | Issues 1.9k | Pull requests 643 | Actions | Projects 1 | Security | Insights

kubernetes Public

Watch 3216 | Fork 40.4k | Star 114k

master | 56 Branches | 1153 Tags | Go to file | Add file | Code

k8s-ci-robot Merge pull request #130860 from carlory/remove-storage-fea... 6396fa0 · 17 minutes ago 128,996 Commits

.github	Add new contribex leads to sig-contribex-approvers	2 years ago
CHANGELOG	Update release notes in changelog-1.30 to fix example cl...	3 hours ago
LICENSES	Vendor randfill	last week
ap		
bu		
cluster	Fix typo and pass the environment variable required to en...	5 days ago
cmd	Merge pull request #130354 from siyuanfoundation/forwa...	4 days ago
docs	Make root approval non-recursive	3 years ago
hack	Merge pull request #130821 from BenTheElder/revert-procs	3 days ago
logo	logo: better alignment of layers	3 years ago
pkg	Merge pull request #128622 from jpbetz/admission-polic...	17 minutes ago
plugin	address comment	3 days ago
staging	Merge pull request #128622 from jpbetz/admission-polic...	17 minutes ago
test	Merge pull request #130860 from carlory/remove-storage...	17 minutes ago
third_party	Revert "tests: include stdout of failed commands in JUnit"	2 months ago
vendor	Vendor randfill	last week
.generated_files	remove clearly unnecessary lingering BUILD file references	3 years ago

About

Production-Grade Container Scheduling and Management

kubernetes.io

go | kubernetes | containers | cncf

- Readme
- Apache-2.0 license
- Code of conduct
- Security policy
- Activity
- Custom properties
- 114k stars
- 3.2k watching
- 40.4k forks

Report repository

Releases 728

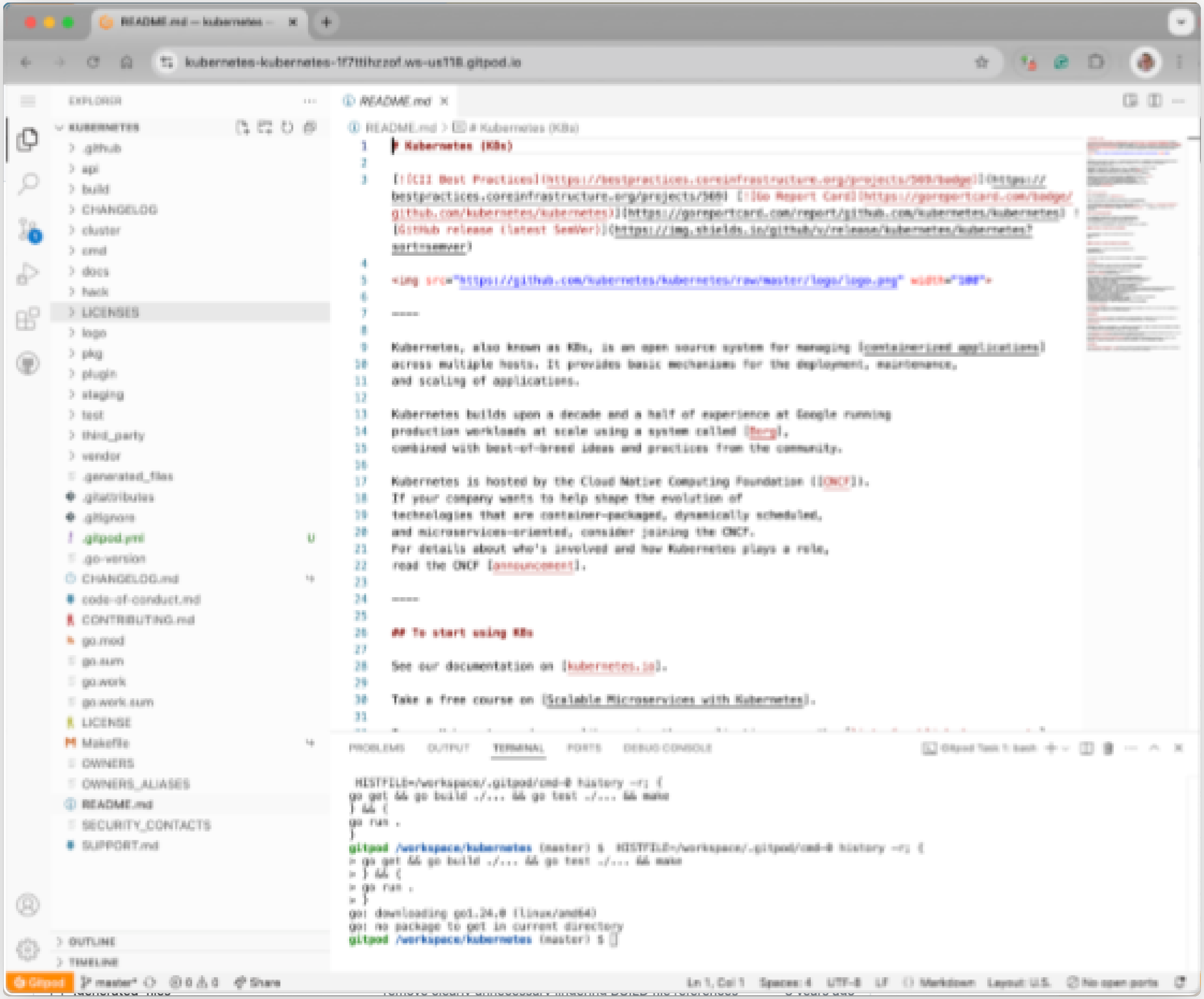
Kubernetes v1.32.3 Latest
last week

+ 727 releases

Packages

No packages published

Contributors 3,779



```
# Kubernetes (K8s)

[![CII Best Practices](https://bestpractices.coreinfrastructure.org/projects/569/badge)](https://bestpractices.coreinfrastructure.org/projects/569) [![Go Report Card](https://goreportcard.com/badge/github.com/kubernetes/kubernetes)](https://goreportcard.com/report/github.com/kubernetes/kubernetes) ![GitHub release (latest SemVer)](https://img.shields.io/github/v/release/kubernetes/kubernetes?sort=semver)

<img src="https://github.com/kubernetes/kubernetes/raw/master/logo/logo.png" width="100">

----

Kubernetes, also known as K8s, is an open source system for managing [containerized applications]
across multiple hosts. It provides basic mechanisms for the deployment, maintenance,
and scaling of applications.

Kubernetes builds upon a decade and a half of experience at Google running
production workloads at scale using a system called [Borg],
combined with best-of-breed ideas and practices from the community.

Kubernetes is hosted by the Cloud Native Computing Foundation ([CNCF]).
If your company wants to help shape the evolution of
technologies that are container-packaged, dynamically scheduled,
and microservices-oriented, consider joining the CNCF.
For details about who's involved and how Kubernetes plays a role,
read the CNCF [announcement].

----

## To start using K8s

See our documentation on [kubernetes.io].

Take a free course on [Scalable Microservices with Kubernetes].
```

```
 HISTFILE=/workspace/.gitpod/cmd-0 history -r; {
go get && go build ./... && go test ./... && make
} && {
go run .
}
gitpod /workspace/kubernetes (master) $  HISTFILE=/workspace/.gitpod/cmd-0 history -r; {
> go get && go build ./... && go test ./... && make
> } && {
> go run .
> }
go: downloading go1.24.0 (linux/amd64)
go: no package to get in current directory
gitpod /workspace/kubernetes (master) $ 
```

GitPod

Open source

VS Code in the cloud

Dedicated environment

GitPod

Open source

VS Code in the cloud

Dedicated environment

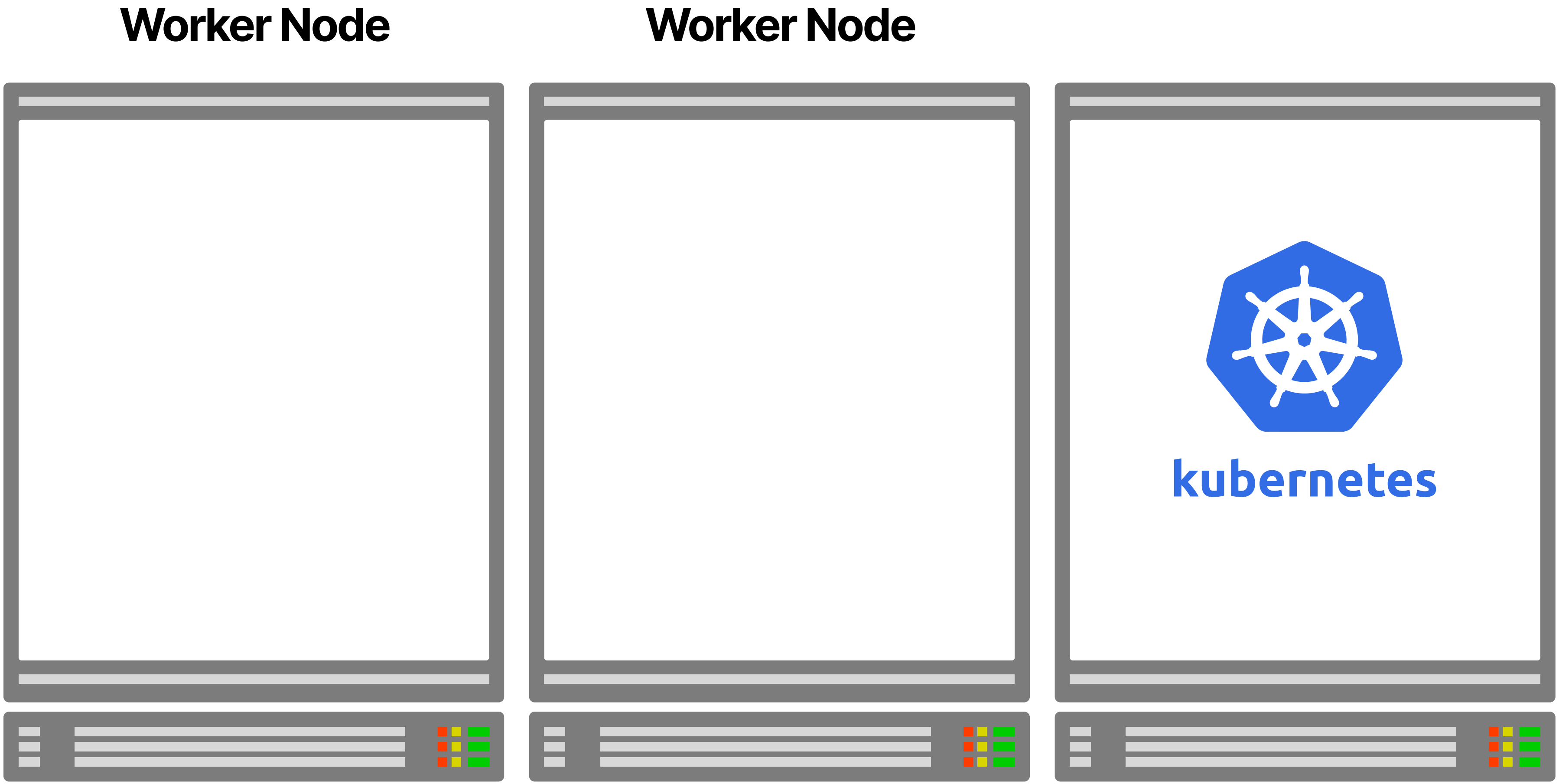
GitPod

Open source

VS Code in the cloud

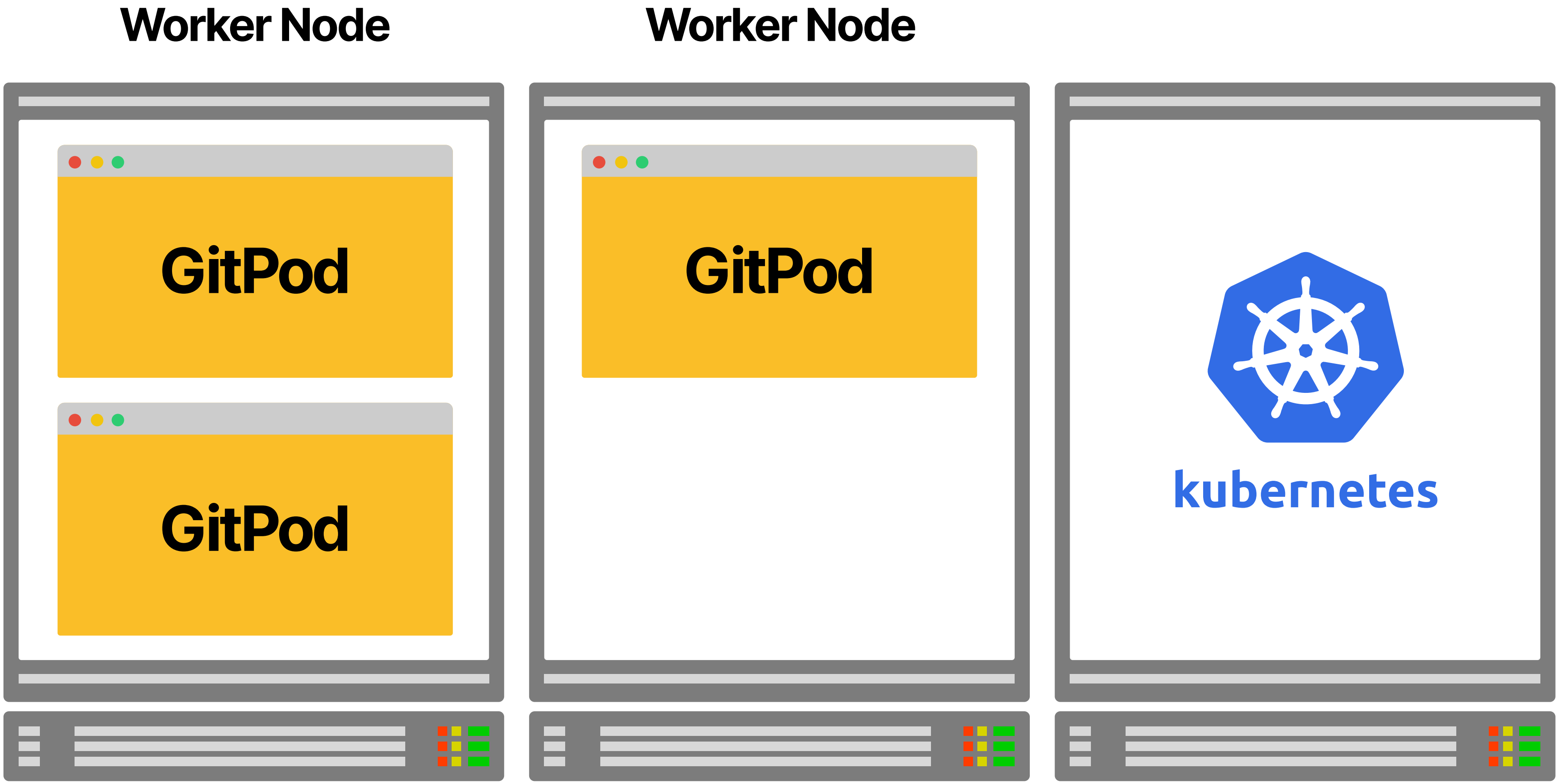
Dedicated environment

GitPod in Kubernetes



Worker Node

Worker Node



Worker Node

Worker Node

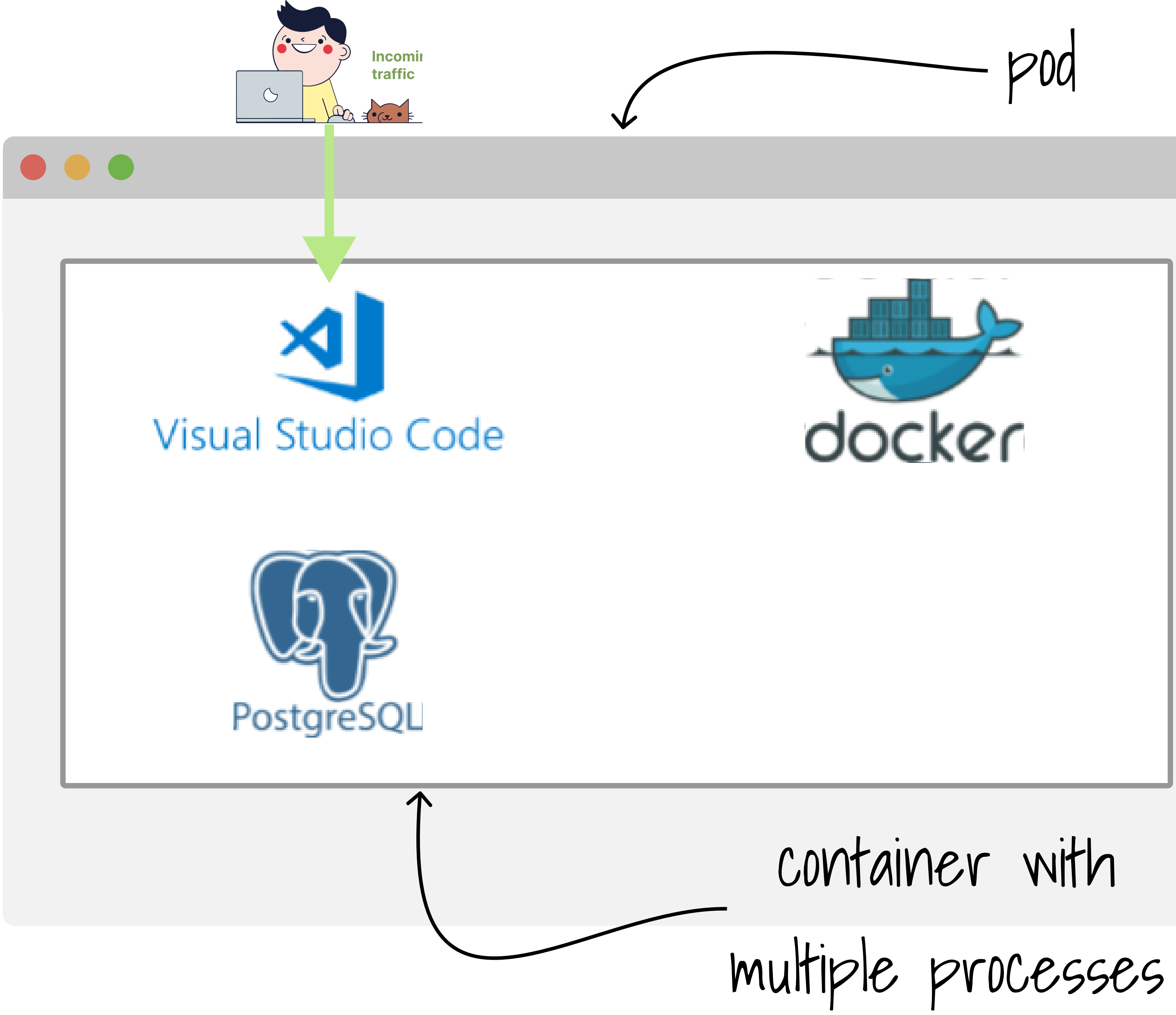
GitPod

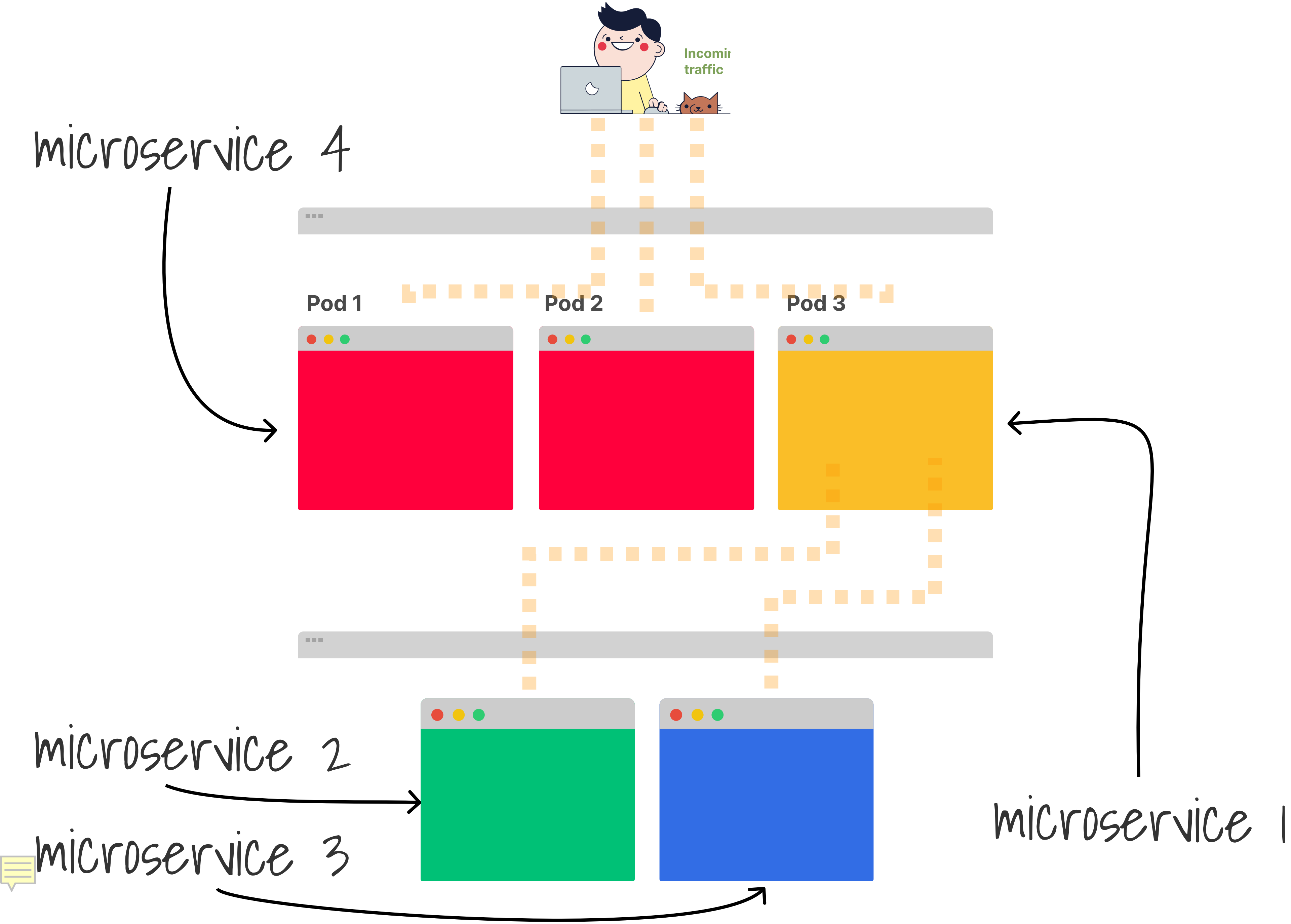
GitPod

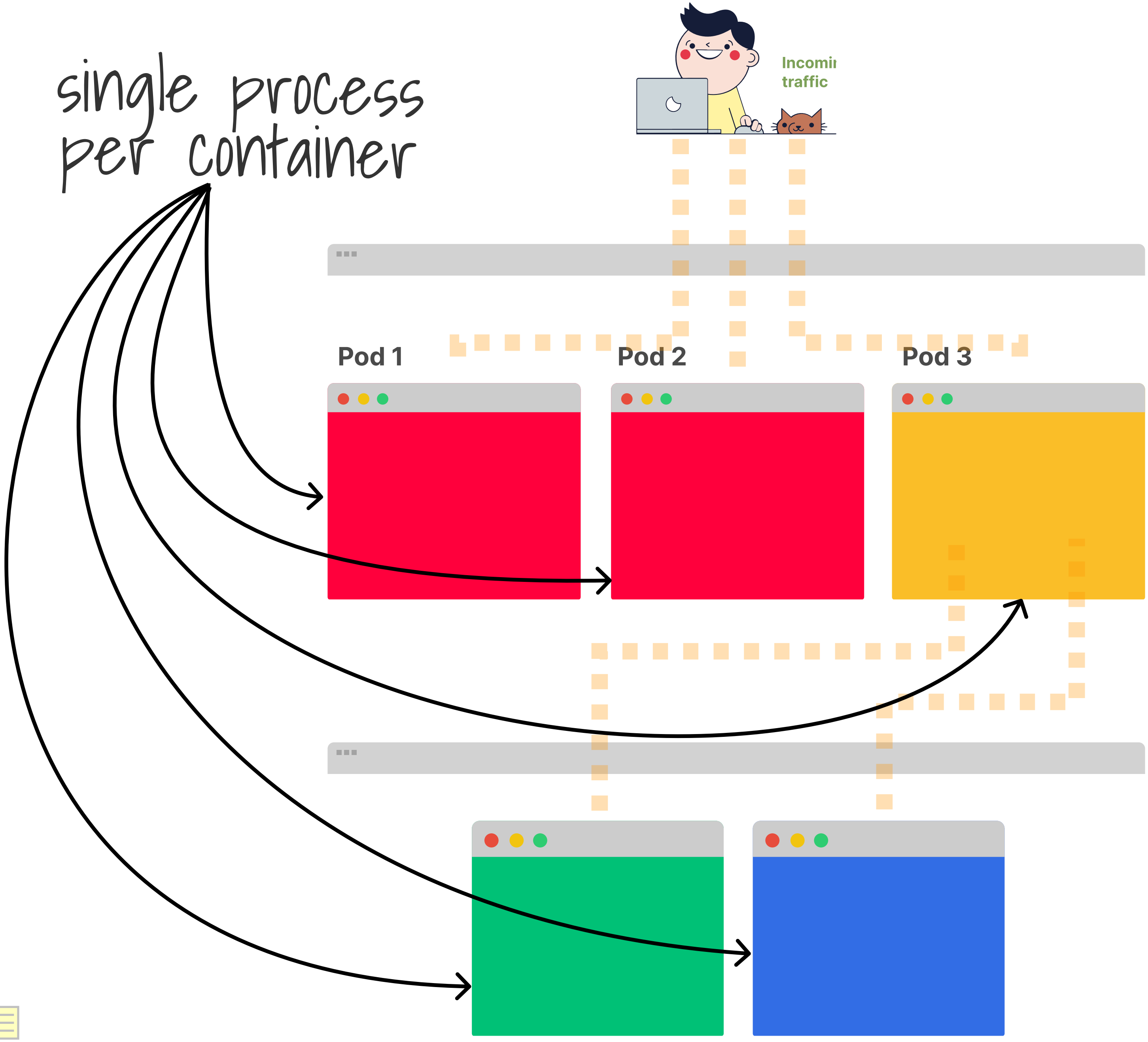
GitPod

kubernetes

Incomi
traffic
pod
GitPod


Incomi
traffic
pod
Visual Studio Code
docker
PostgreSQL
container with
multiple processes


microservice 4
Incomir
traffic
Pod 1
Pod 2
Pod 3
microservice 2
microservice 3
microservice 1



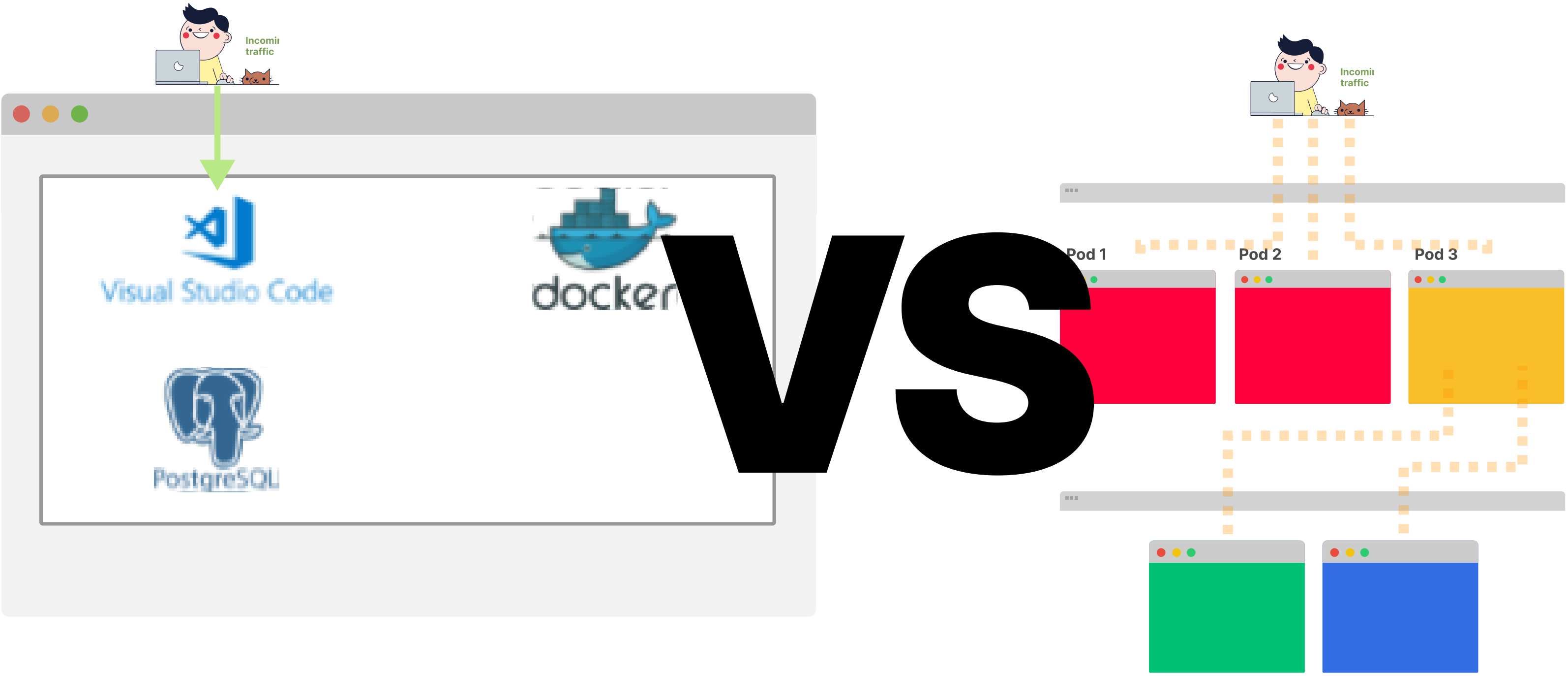
single process per container

Incomir
traffic

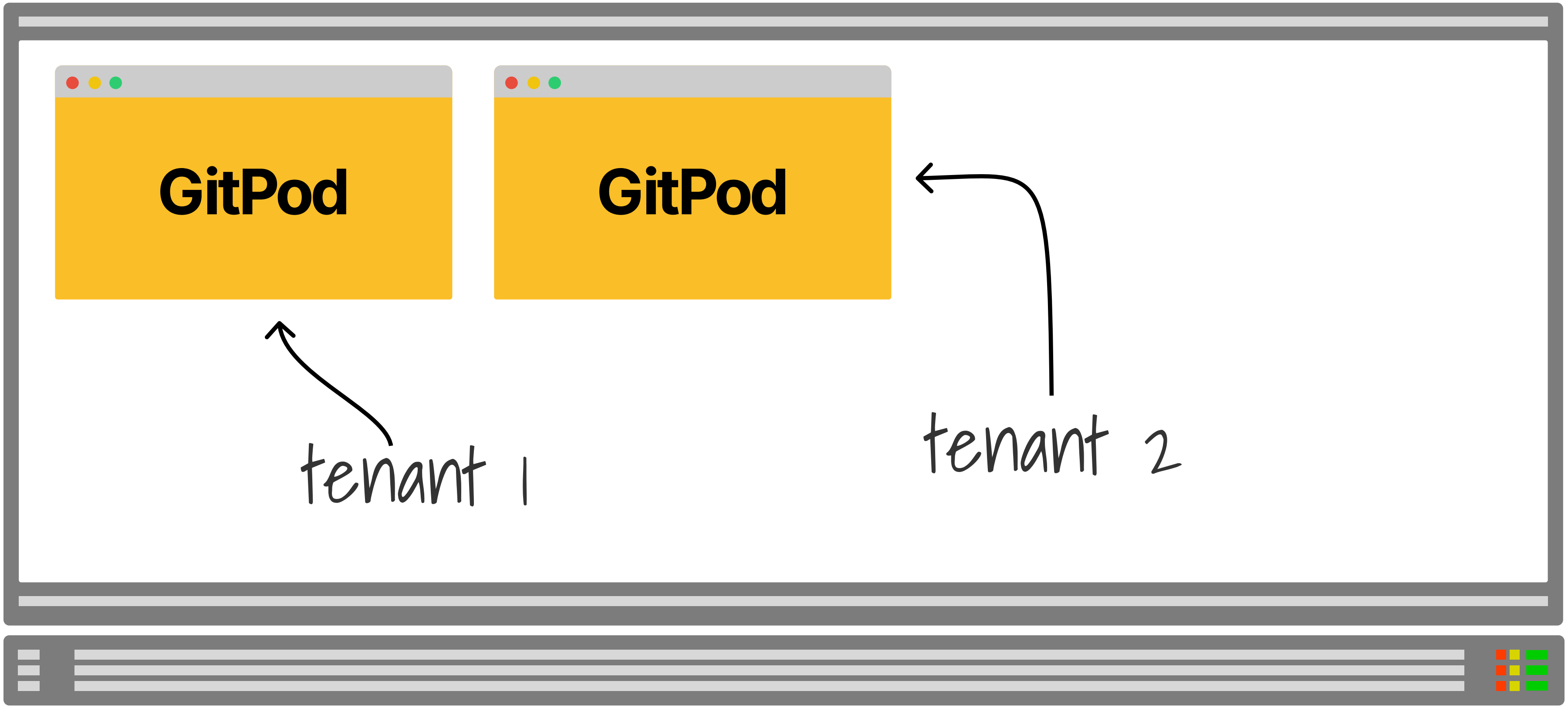
Pod 1

Pod 2

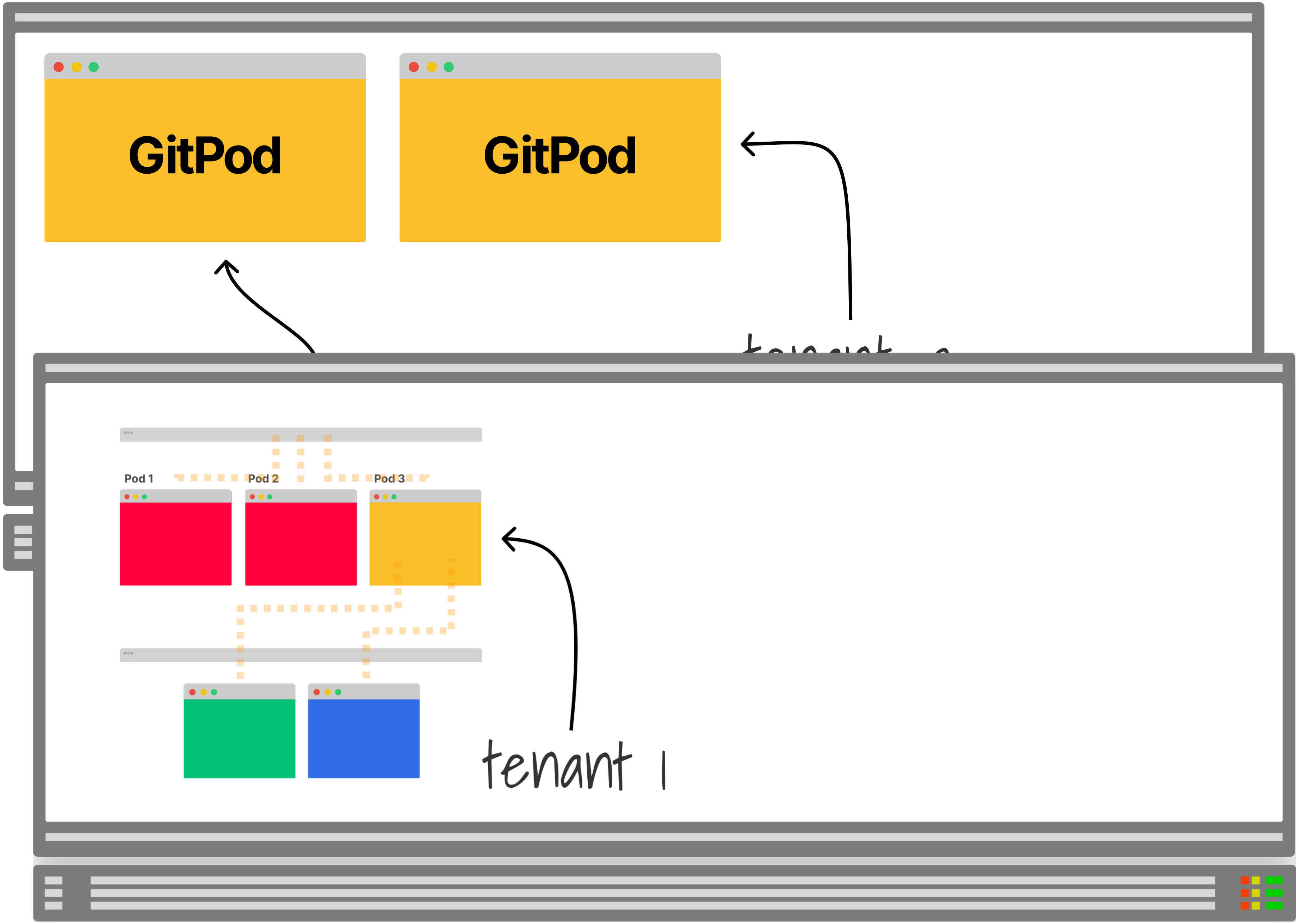
Pod 3

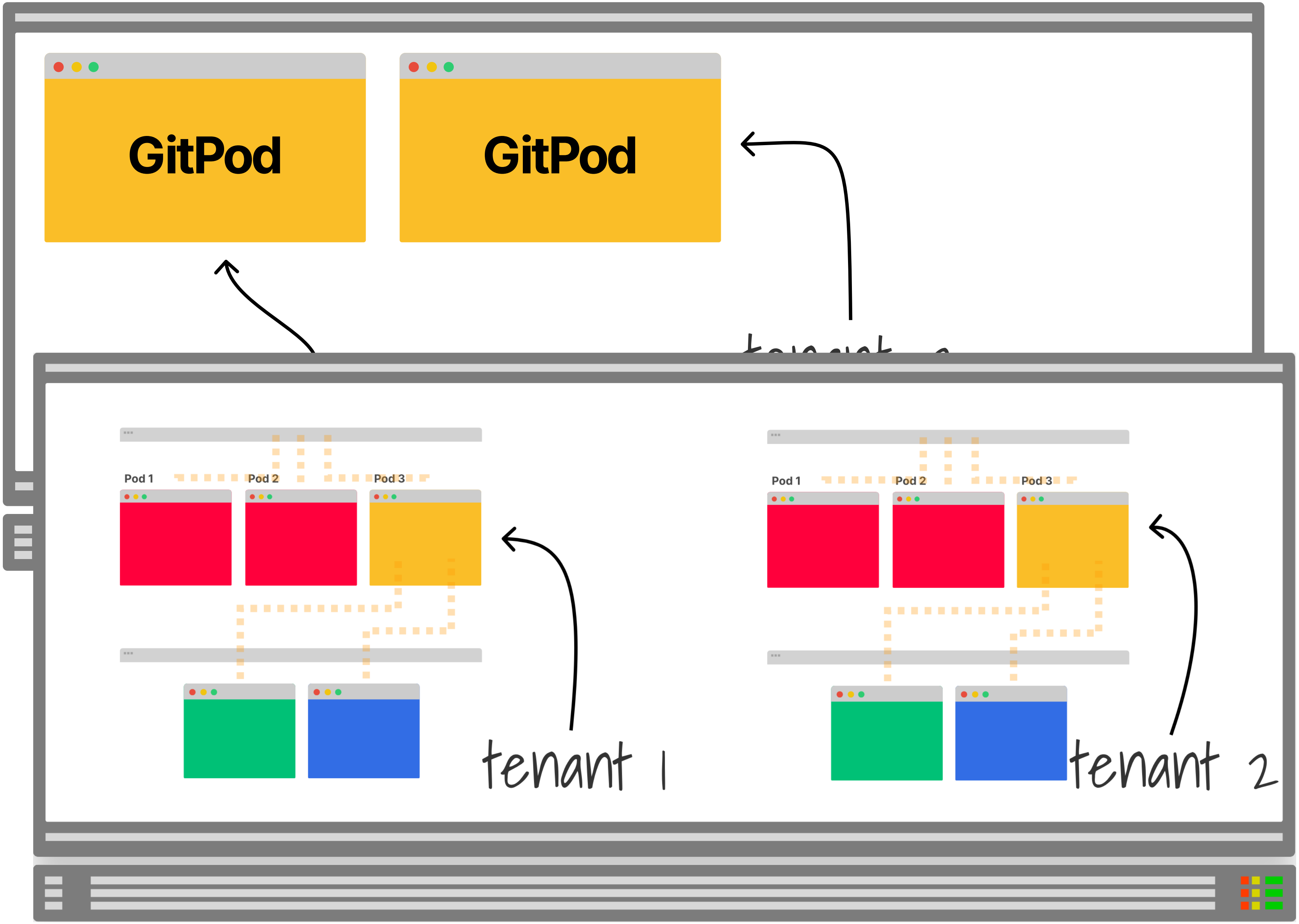

Incomi
traffic
Visual Studio Code
docker
PostgreSQL
Incomi
traffic
Pod 1
Pod 2
Pod 3

VS



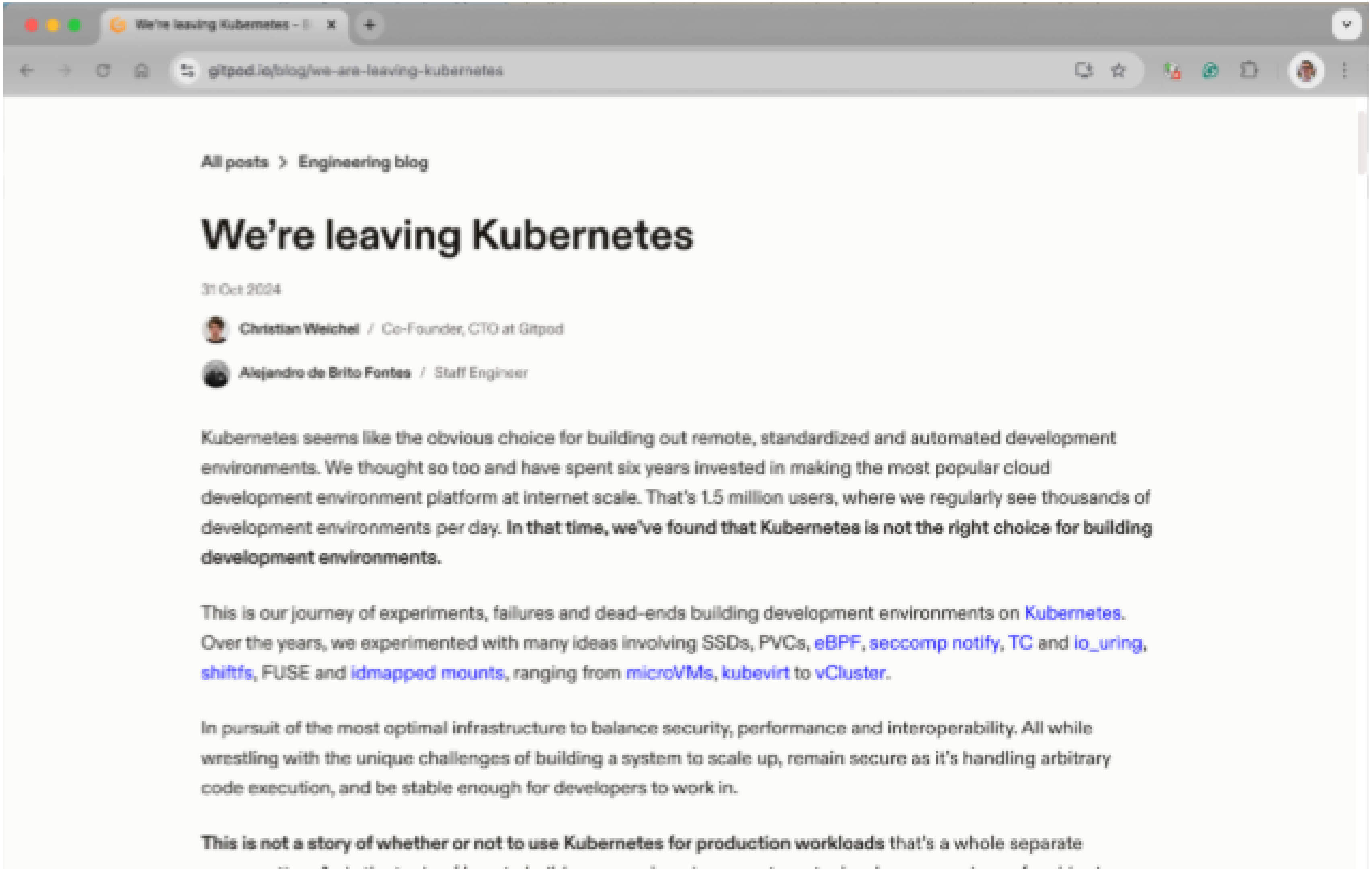
GitPod
GitPod
tenant 1
tenant 2


GitPod
GitPod
tenant 2
Pod 1
Pod 2
Pod 3
tenant 1


GitPod
GitPod
Pod 1
Pod 2
Pod 3
tenant 1
Pod 1
Pod 2
Pod 3
tenant 2

Case study

GitPod


We're leaving Kubernetes - ...
gitpod.io/blog/we-are-leaving-kubernetes

All posts > Engineering blog

We're leaving Kubernetes

31 Oct 2024

Christian Weichel / Co-Founder, CTO at Gitpod

Alejandro de Brito Fontes / Staff Engineer

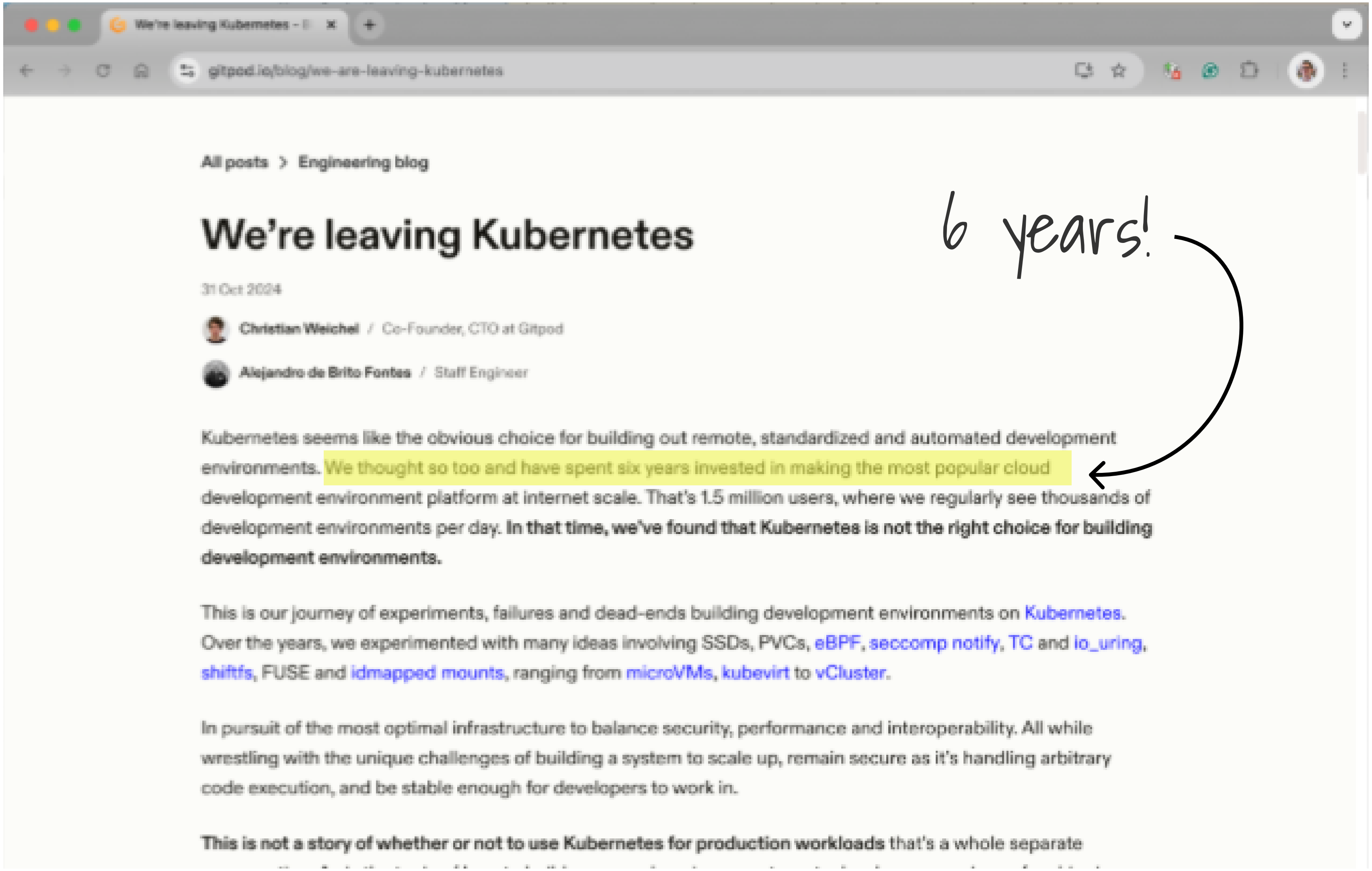
Kubernetes seems like the obvious choice for building out remote, standardized and automated development environments. We thought so too and have spent six years invested in making the most popular cloud development environment platform at internet scale. That's 1.5 million users, where we regularly see thousands of development environments per day. **In that time, we've found that Kubernetes is not the right choice for building development environments.**

This is our journey of experiments, failures and dead-ends building development environments on Kubernetes. Over the years, we experimented with many ideas involving SSDs, PVCs, eBPF, seccomp notify, TC and io_uring, shiftfs, FUSE and idmapped mounts, ranging from microVMs, kubevirt to vCluster.

In pursuit of the most optimal infrastructure to balance security, performance and interoperability. All while wrestling with the unique challenges of building a system to scale up, remain secure as it's handling arbitrary code execution, and be stable enough for developers to work in.

This is not a story of whether or not to use Kubernetes for production workloads that's a whole separate

31 Oct 2024


We're leaving Kubernetes - G
gitpod.io/blog/we-are-leaving-kubernetes

All posts > Engineering blog

We're leaving Kubernetes

31 Oct 2024

Christian Weichel / Co-Founder, CTO at Gitpod

Alejandro de Brito Fontes / Staff Engineer

Kubernetes seems like the obvious choice for building out remote, standardized and automated development environments. We thought so too and have spent six years invested in making the most popular cloud development environment platform at internet scale. That's 1.5 million users, where we regularly see thousands of development environments per day. **In that time, we've found that Kubernetes is not the right choice for building development environments.**

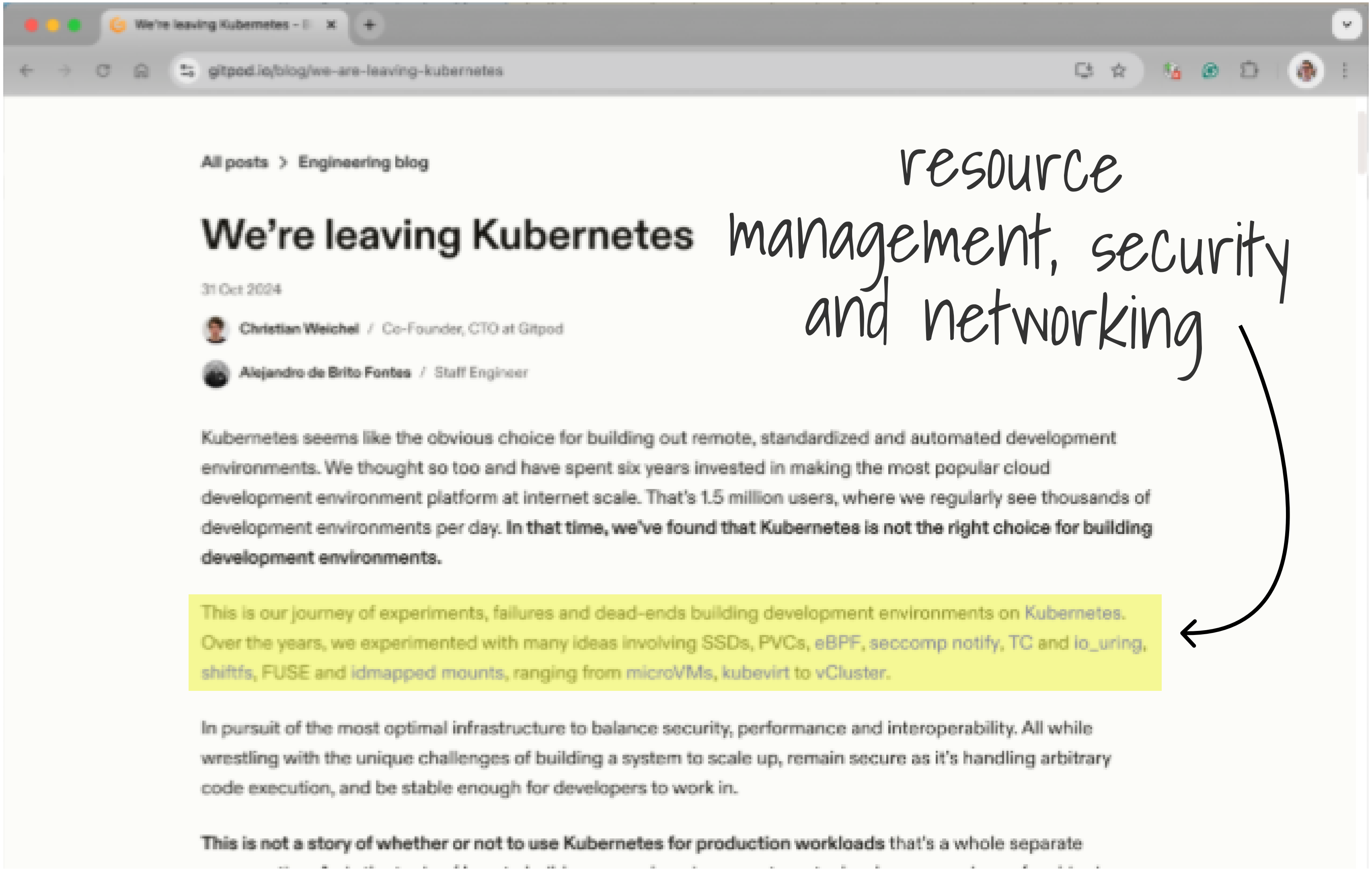
This is our journey of experiments, failures and dead-ends building development environments on Kubernetes. Over the years, we experimented with many ideas involving SSDs, PVCs, eBPF, seccomp notify, TC and io_uring, shiftfs, FUSE and idmapped mounts, ranging from microVMs, kubevirt to vCluster.

In pursuit of the most optimal infrastructure to balance security, performance and interoperability. All while wrestling with the unique challenges of building a system to scale up, remain secure as it's handling arbitrary code execution, and be stable enough for developers to work in.

This is not a story of whether or not to use Kubernetes for production workloads that's a whole separate

6 years!

31 Oct 2024


We're leaving Kubernetes - G
gitpod.io/blog/we-are-leaving-kubernetes

All posts › Engineering blog

We're leaving Kubernetes

31 Oct 2024

Christian Weichel / Co-Founder, CTO at Gitpod

Alejandro de Brito Fontes / Staff Engineer

Kubernetes seems like the obvious choice for building out remote, standardized and automated development environments. We thought so too and have spent six years invested in making the most popular cloud development environment platform at internet scale. That's 1.5 million users, where we regularly see thousands of development environments per day. **In that time, we've found that Kubernetes is not the right choice for building development environments.**

This is our journey of experiments, failures and dead-ends building development environments on Kubernetes. Over the years, we experimented with many ideas involving SSDs, PVCs, eBPF, seccomp notify, TC and io_uring, shiftfs, FUSE and idmapped mounts, ranging from microVMs, kubevirt to vCluster.

In pursuit of the most optimal infrastructure to balance security, performance and interoperability. All while wrestling with the unique challenges of building a system to scale up, remain secure as it's handling arbitrary code execution, and be stable enough for developers to work in.

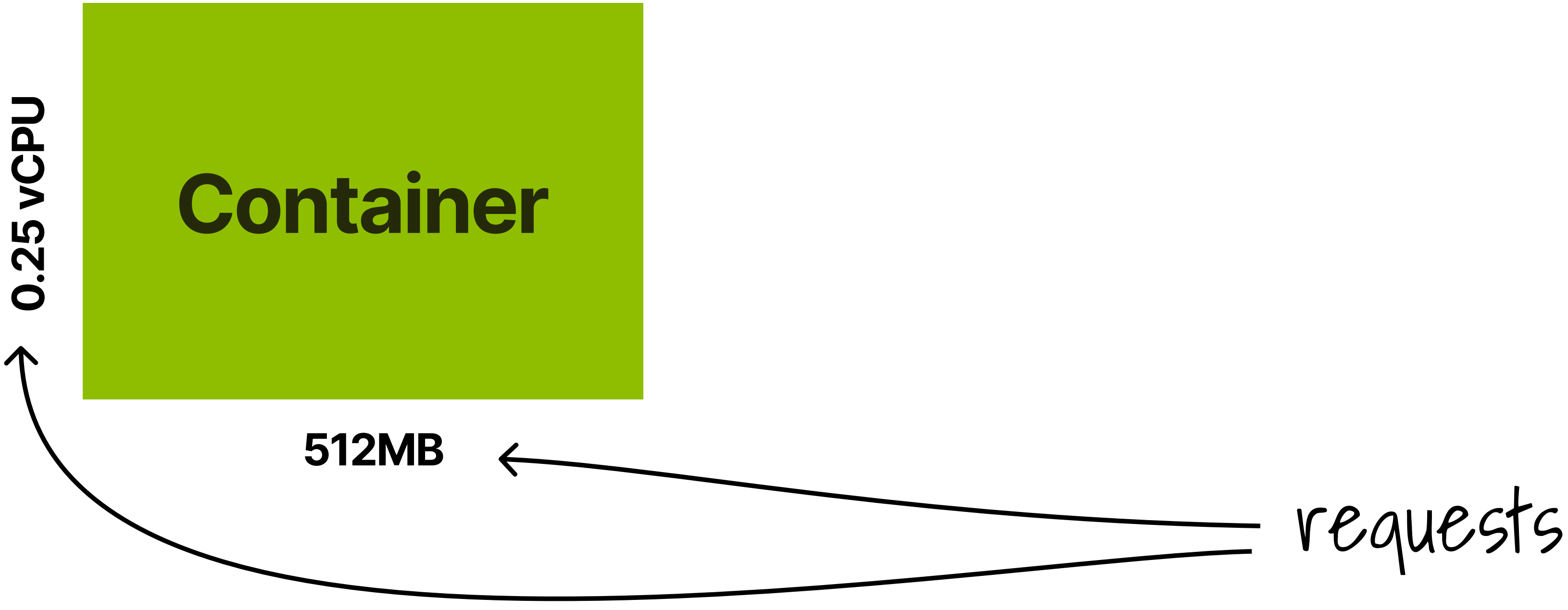
This is not a story of whether or not to use Kubernetes for production workloads that's a whole separate

resource management, security and networking

31 Oct 2024

Resource management

Challenge 1

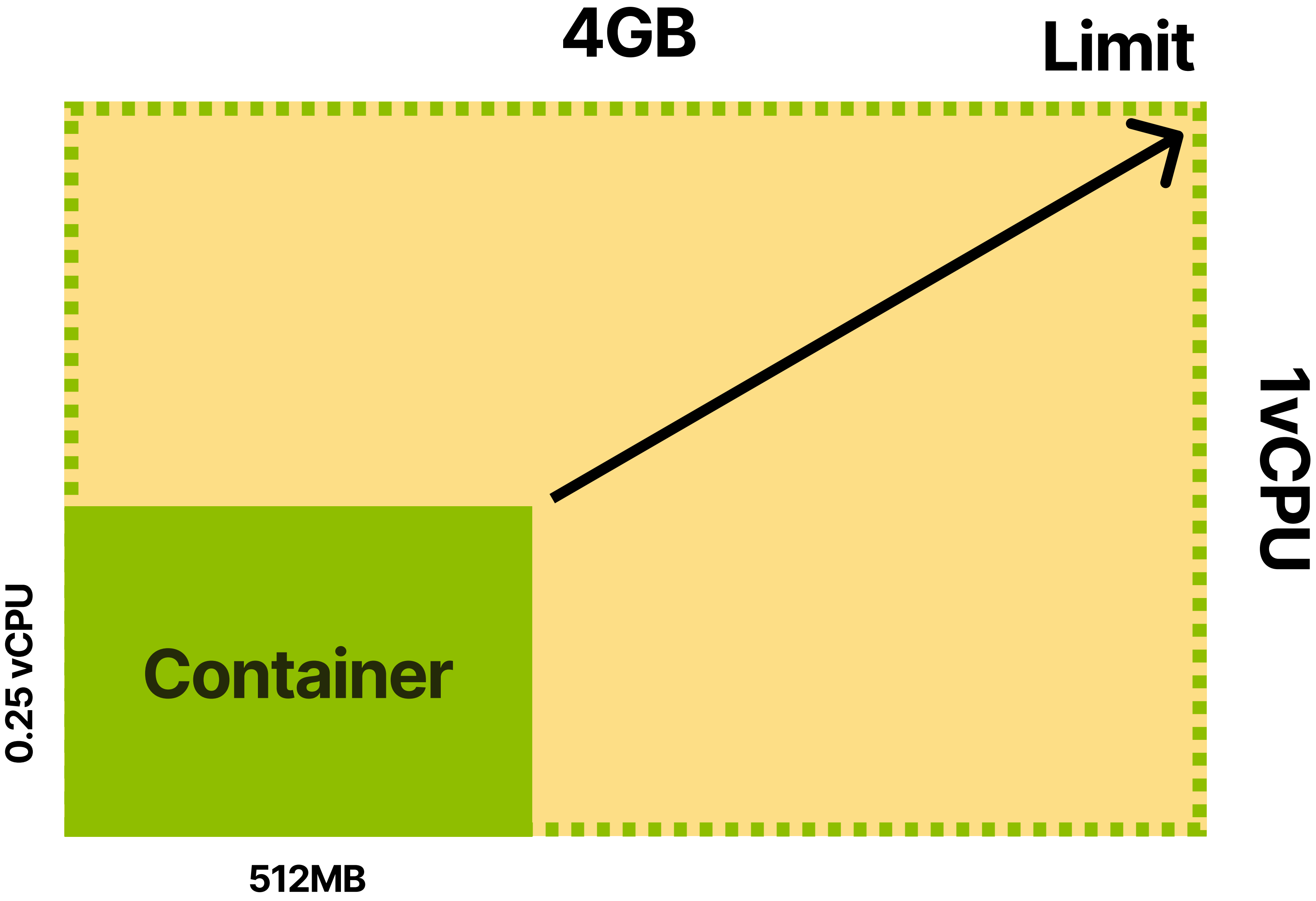


Container

0.25 vCPU

512MB

requests



4GB

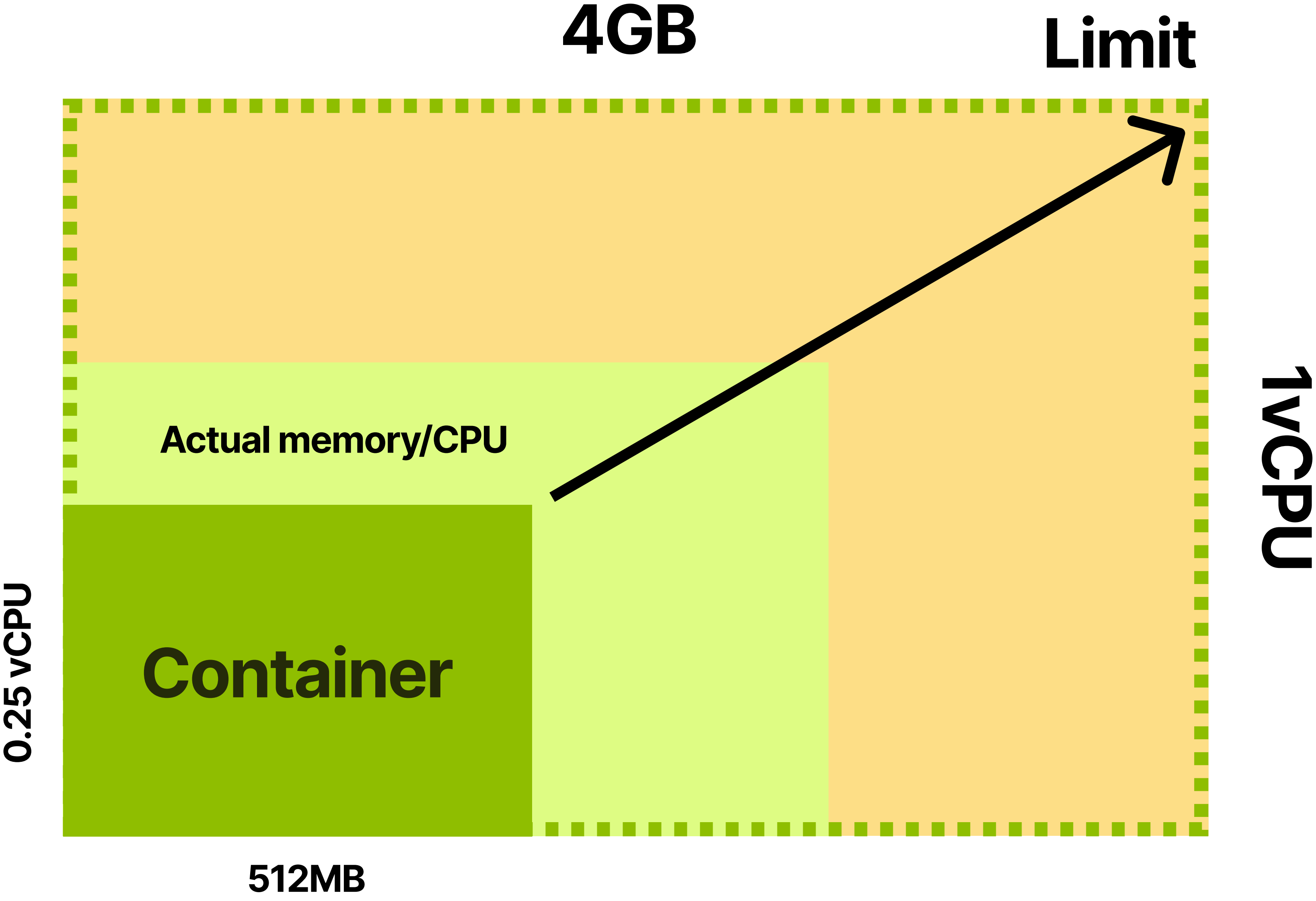
Limit

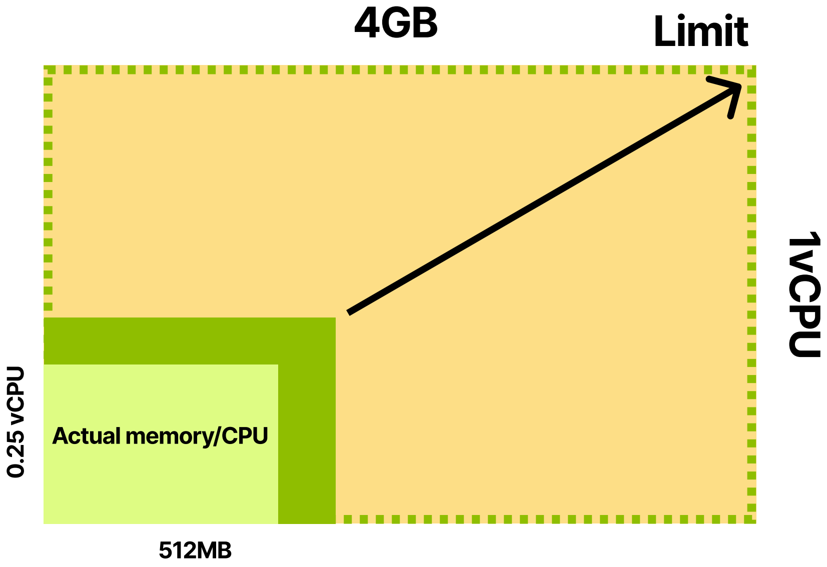
1vCPU

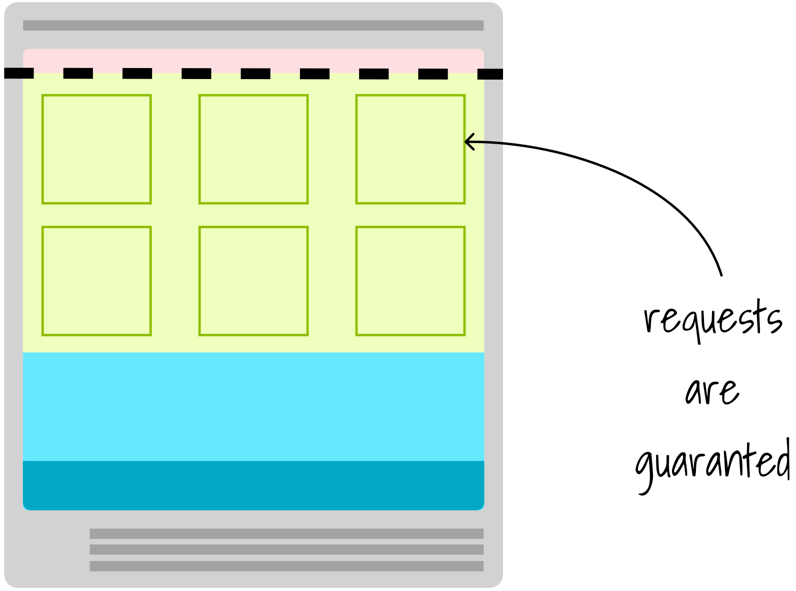
Container

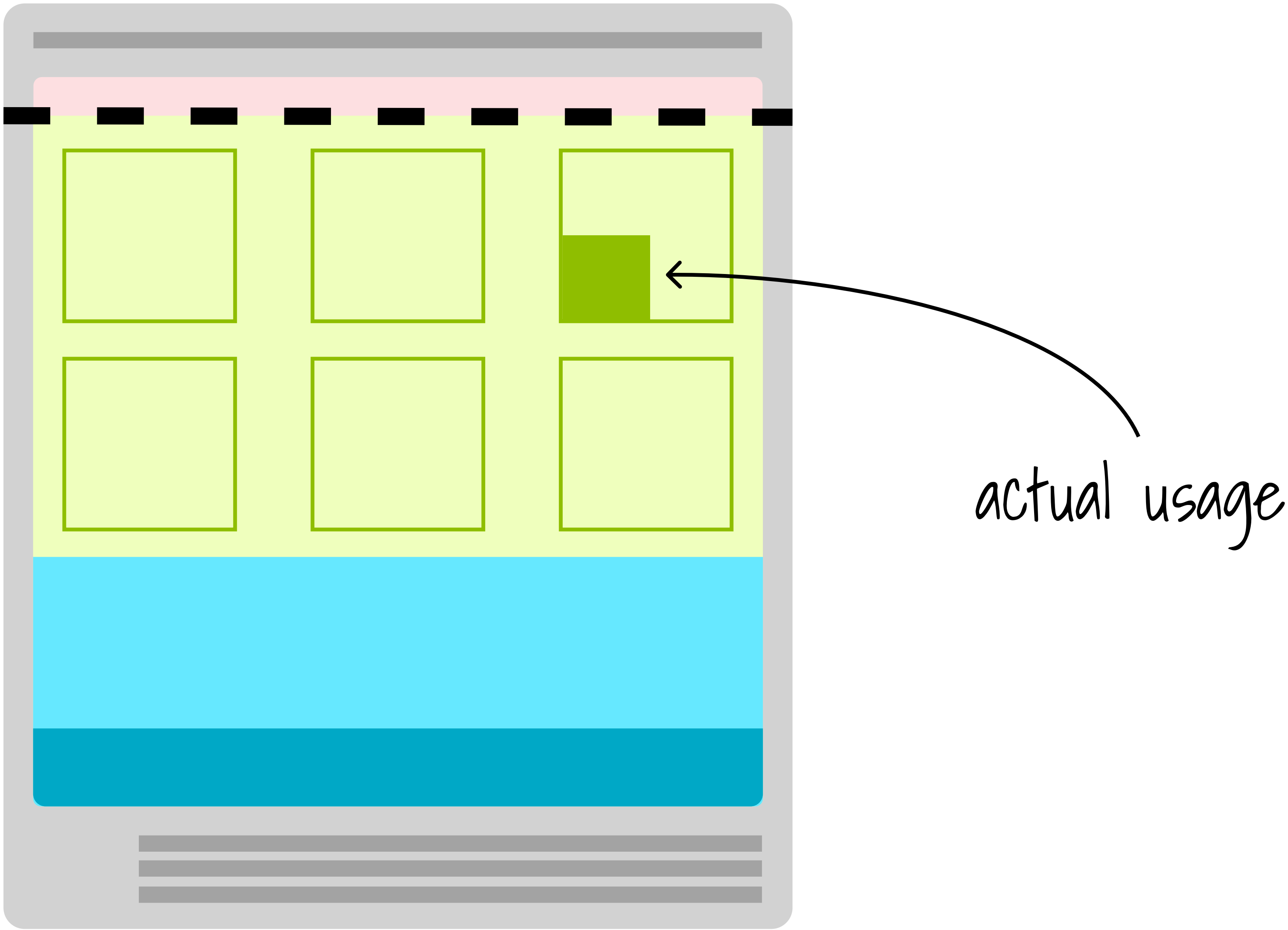
0.25 vCPU

512MB

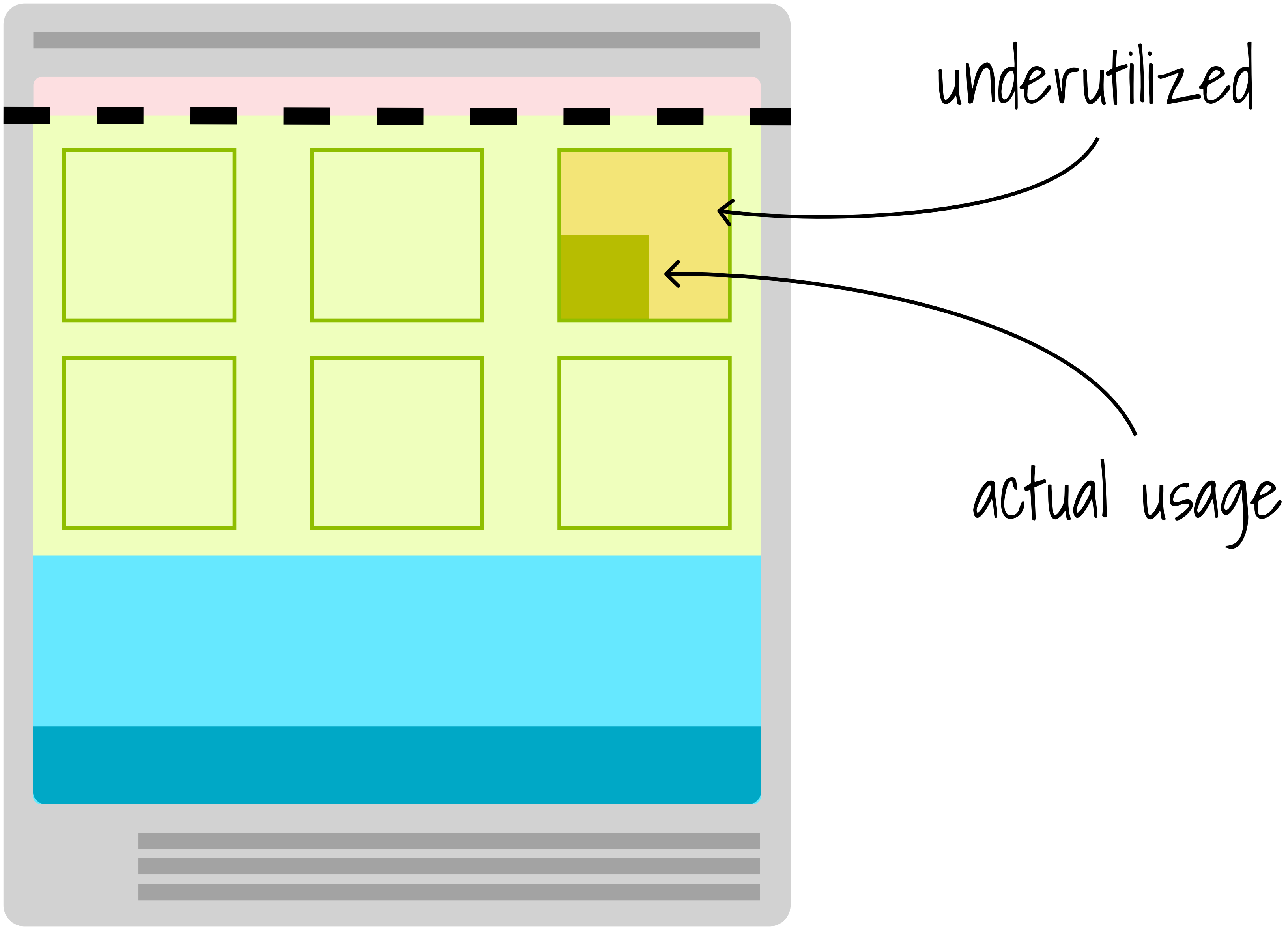

4GB
Limit
Actual memory/CPU
1vCPU
0.25 vCPU
Container
512MB

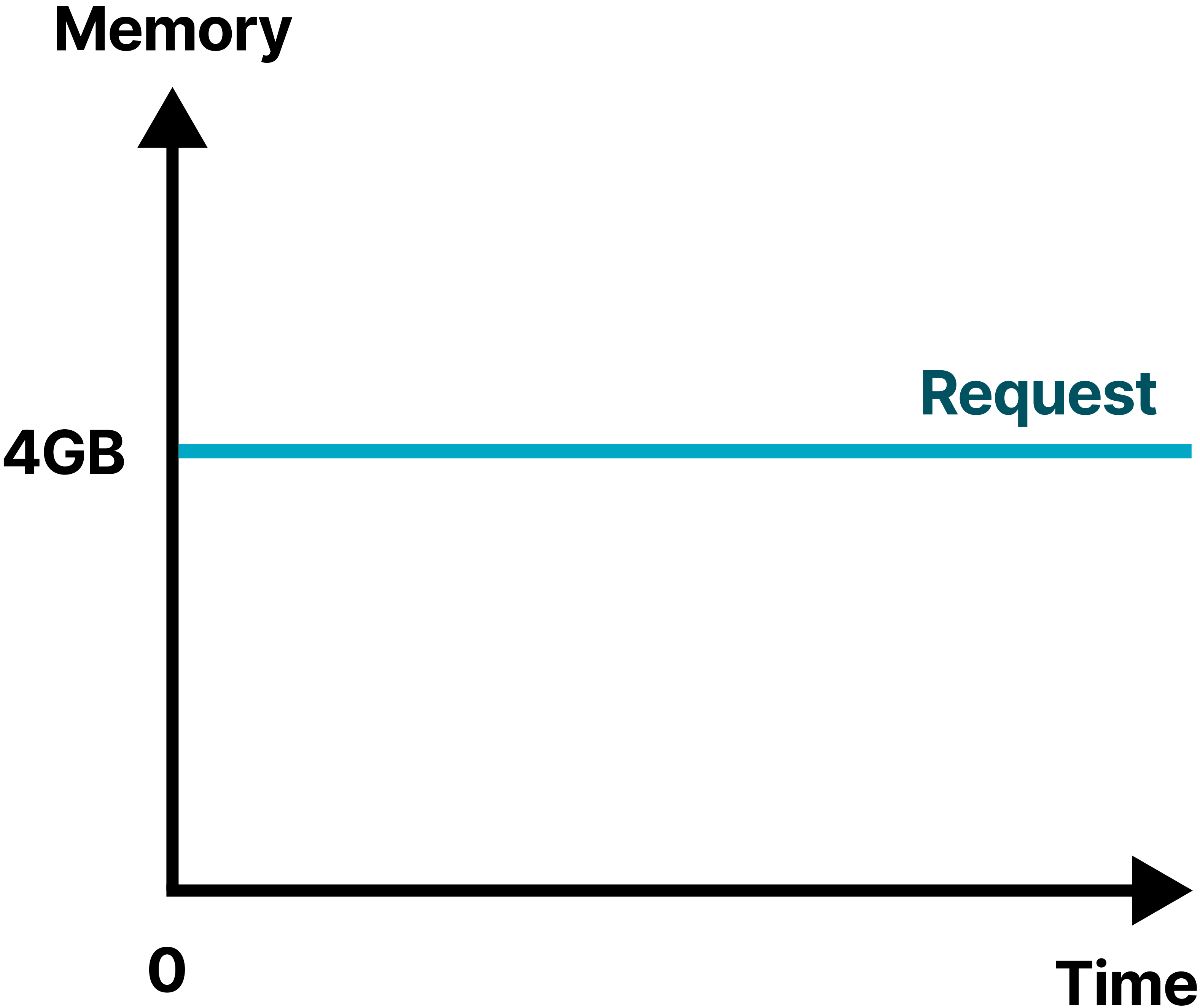

4GB
Limit
1vCPU
0.25 vCPU
Actual memory/CPU
512MB

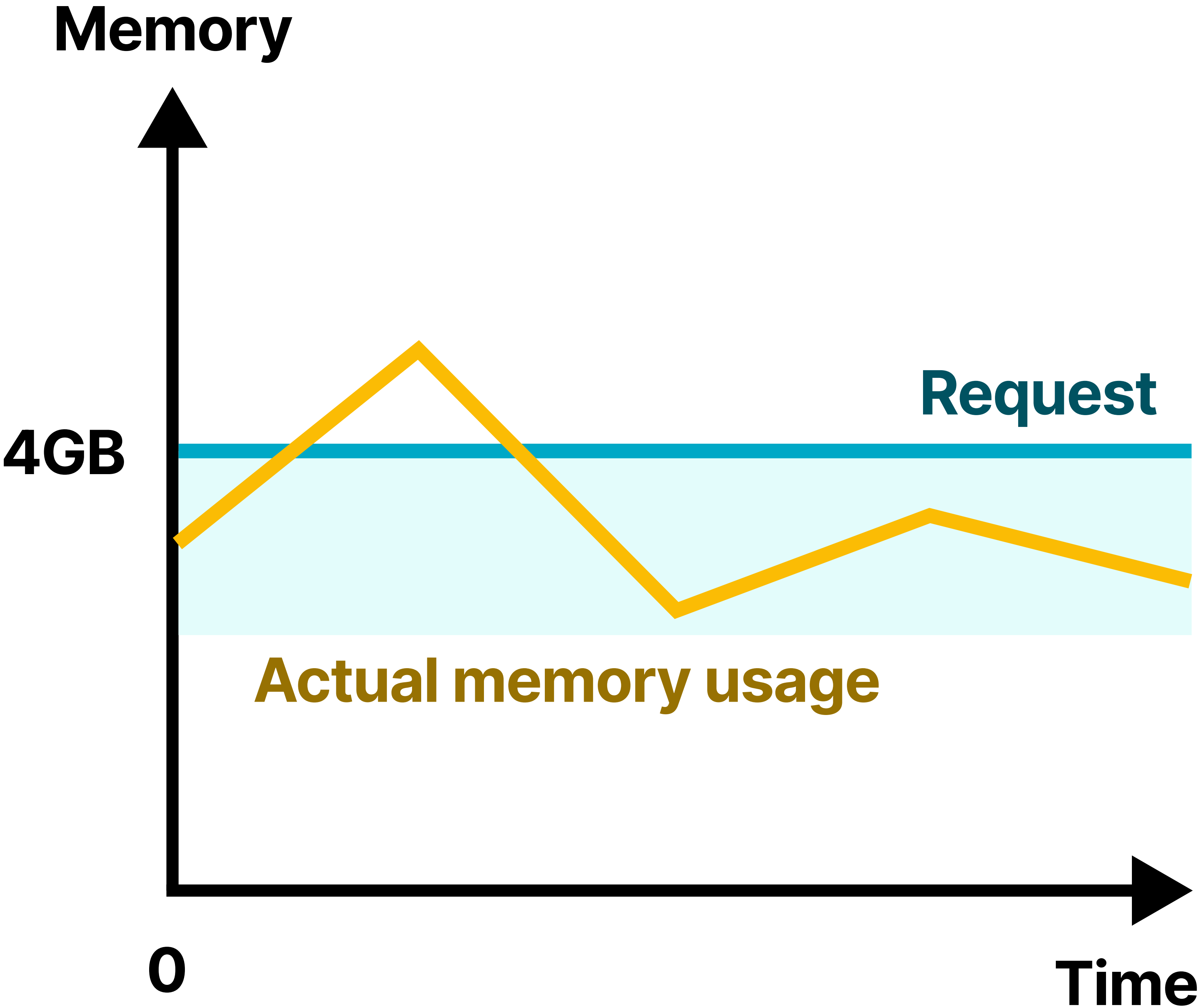

requests
are
guaranted

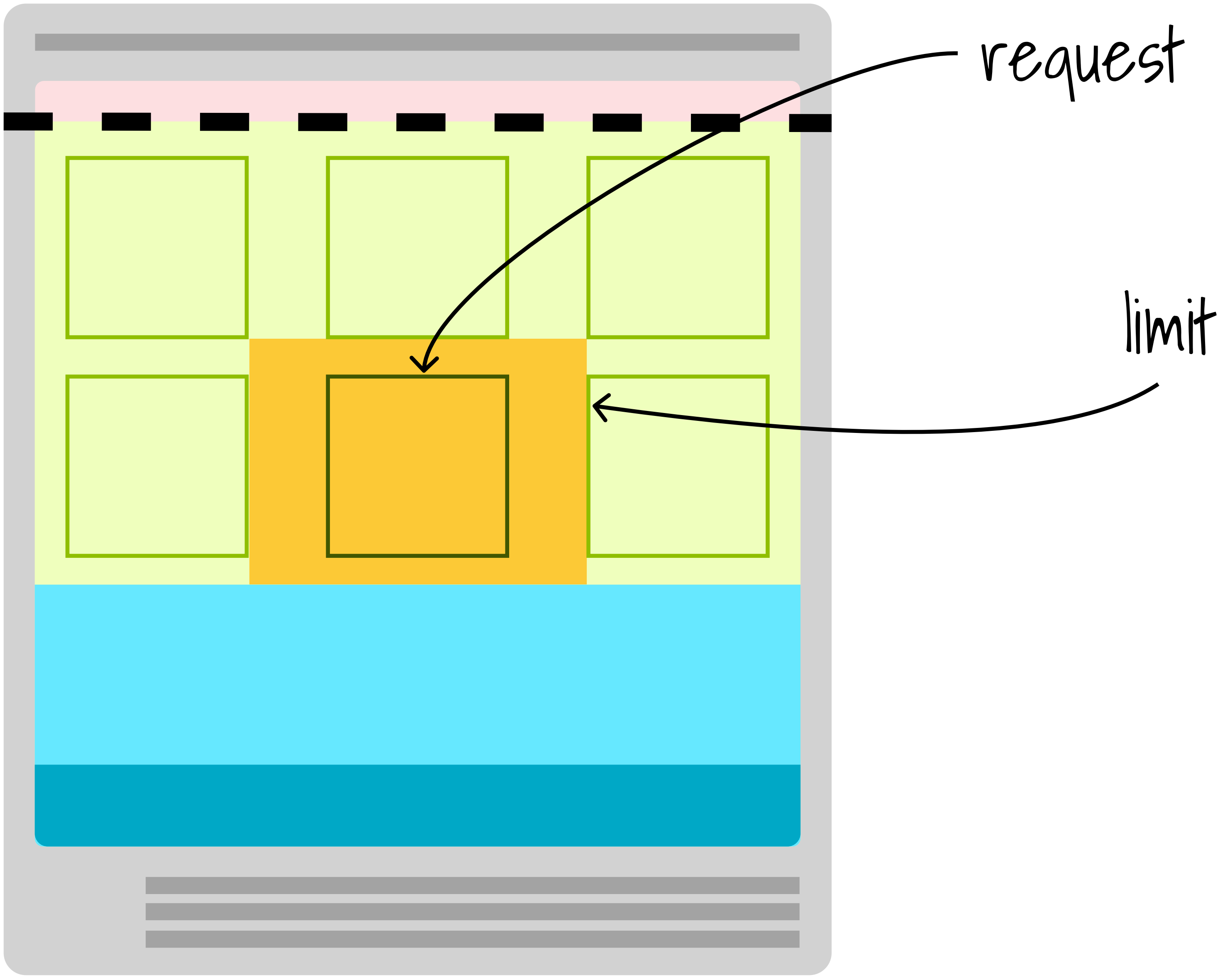


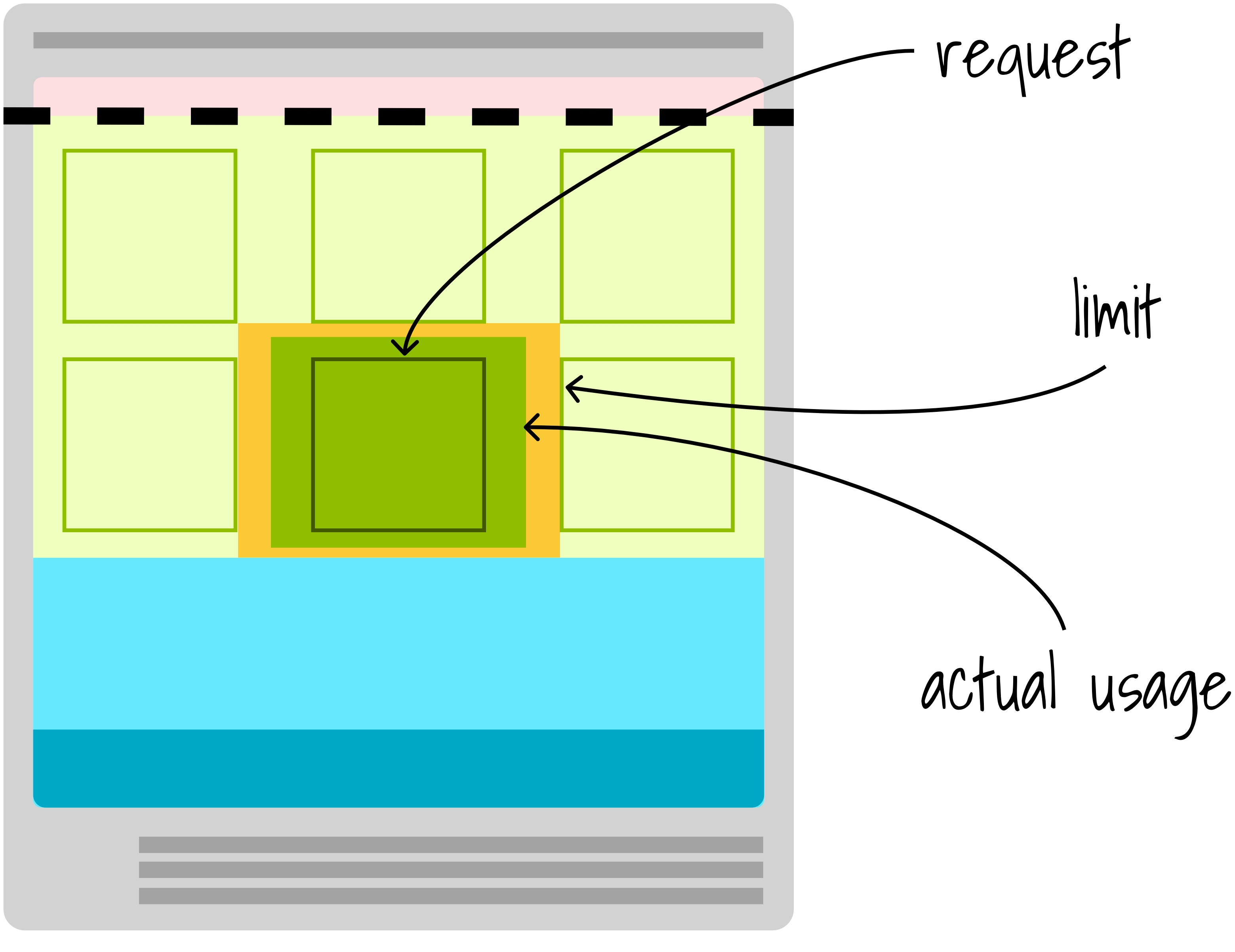
actual usage

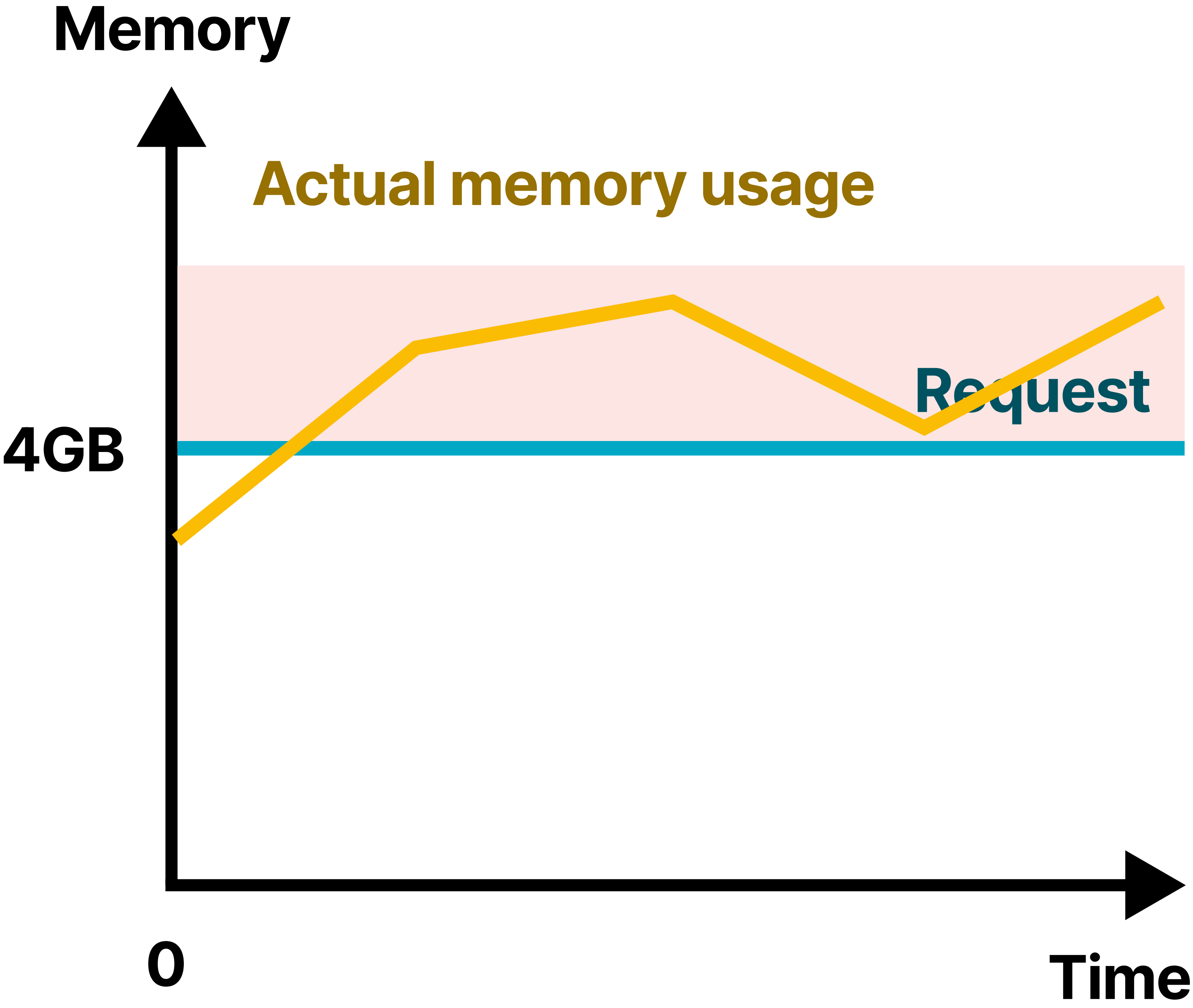

underutilized
actual usage


Memory
Request
4GB
0
Time

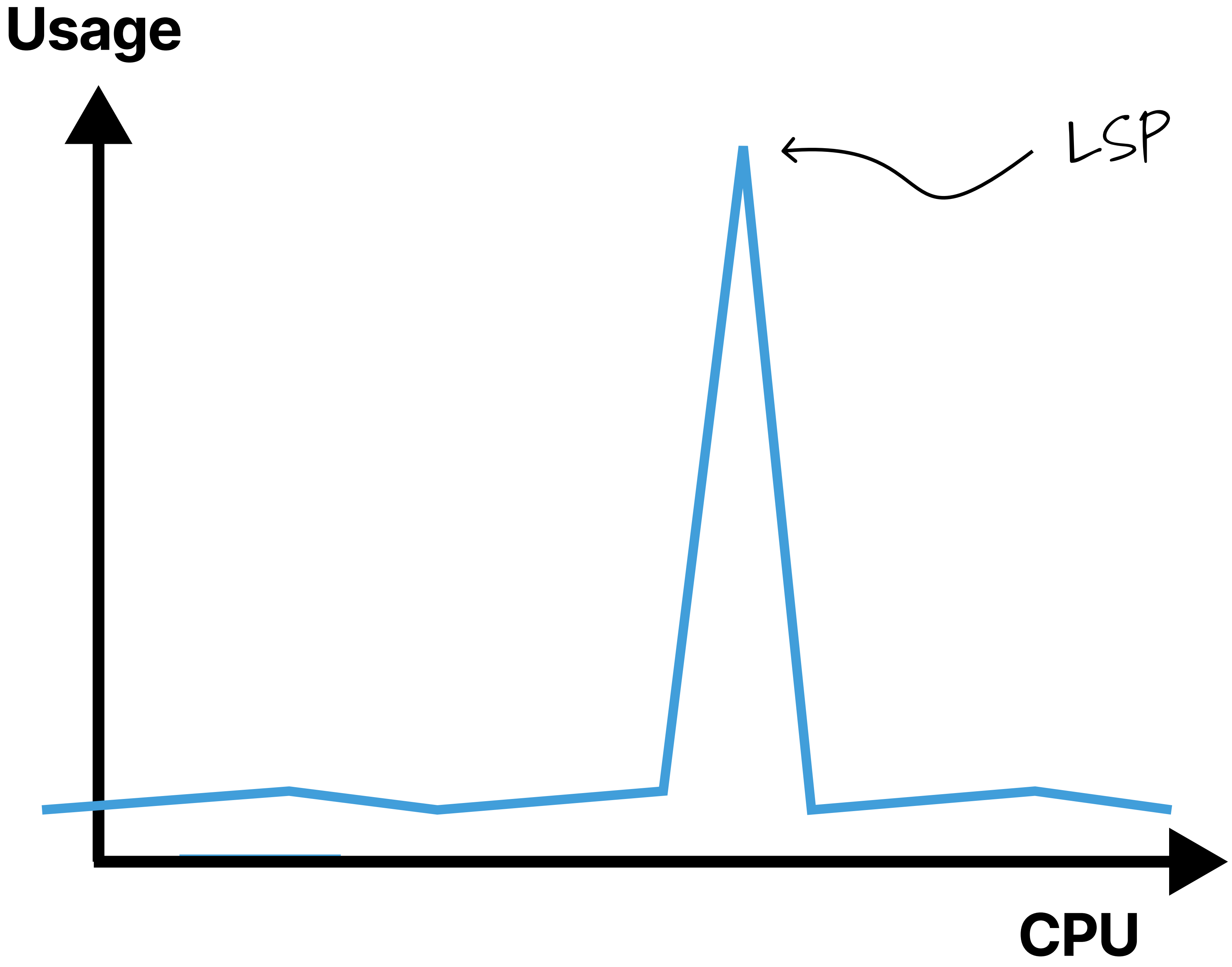

Memory
Request
4GB
Actual memory usage
0
Time

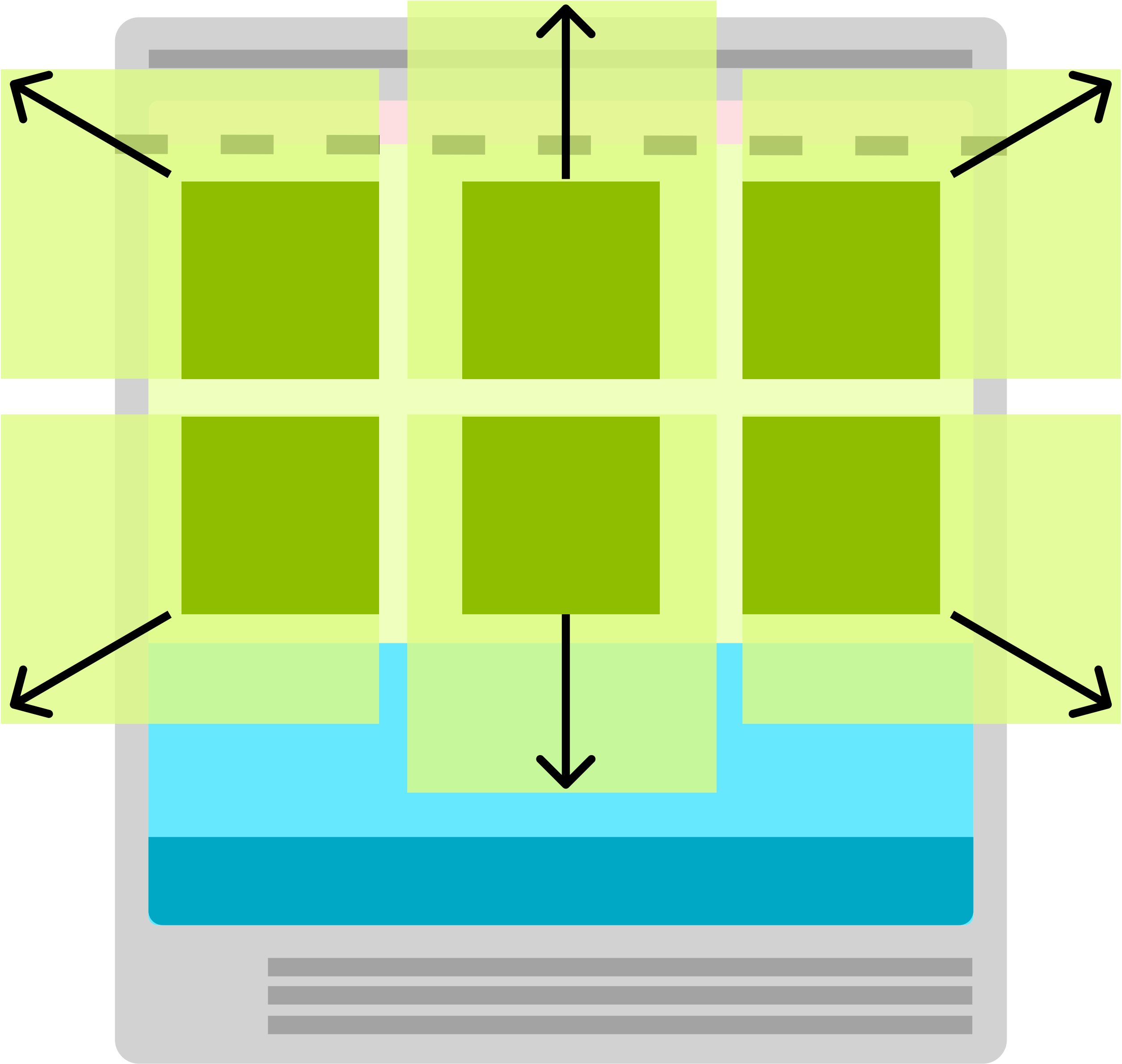

request
limit


request
limit
actual usage

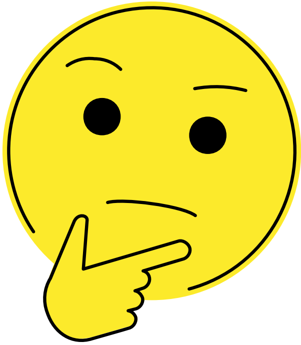

Memory
Actual memory usage
Request
4GB
0
Time

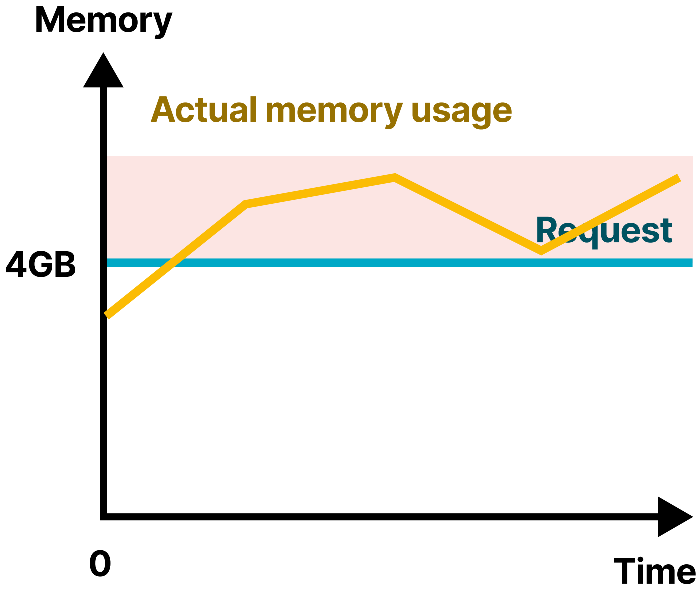
The challenge


Usage
LSP
CPU



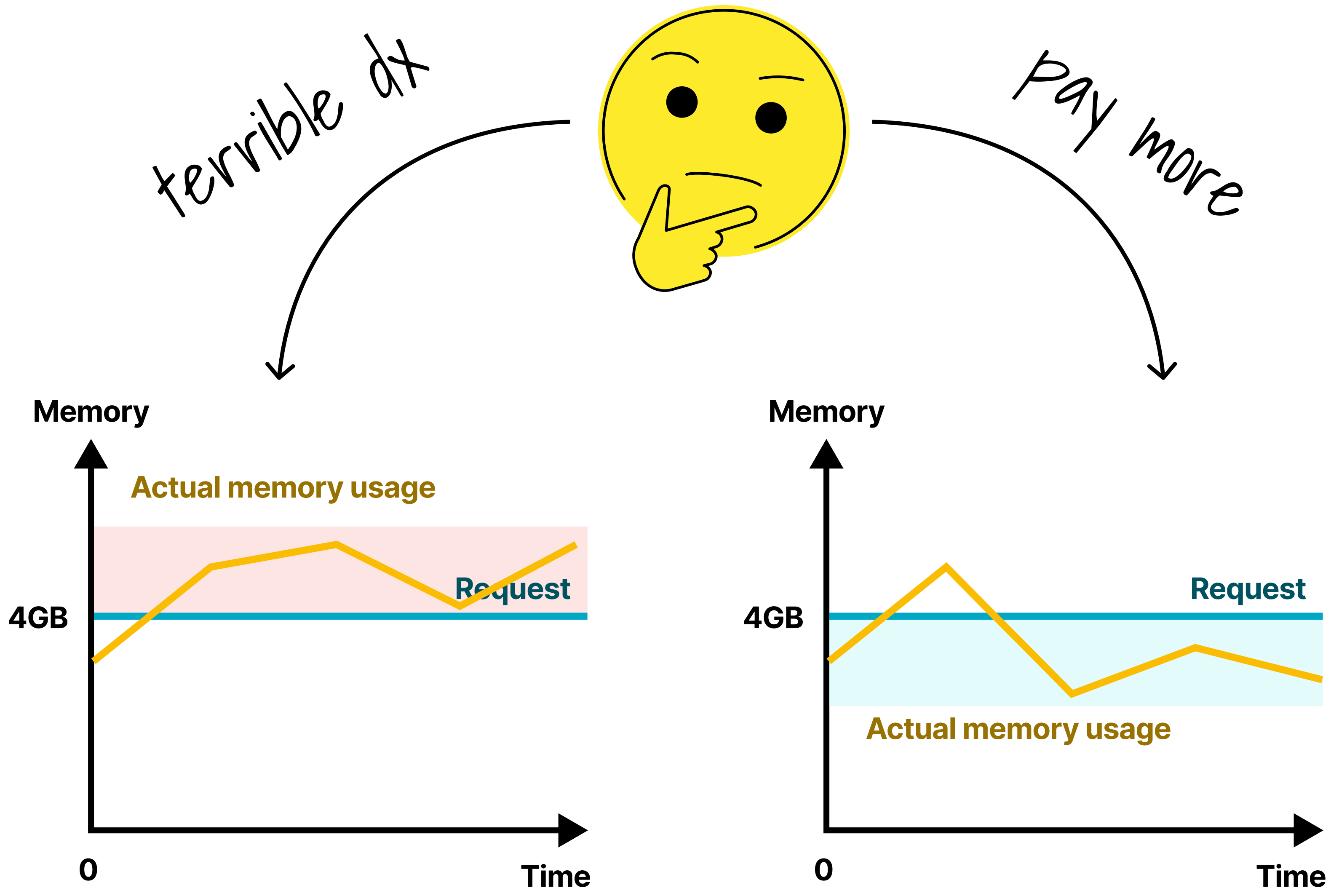
terrible dx





Slow intellisense

Slow builds



Slow intellisense

Slow builds

wasted resourced

GitPod

CFS hints

Dynamic resource allocation

Swap-space memory

GitPod

CFS hints

Dynamic resource allocation

Swap-space memory

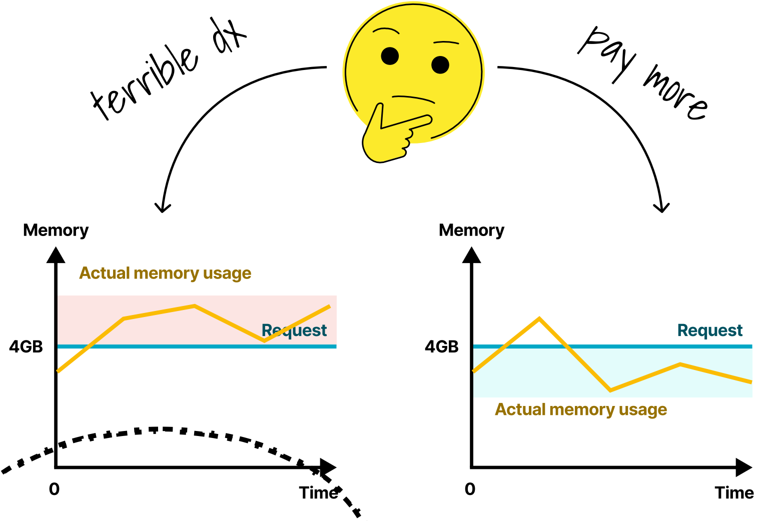
GitPod

CFS hints

Dynamic resource allocation

Swap-space memory

Your options



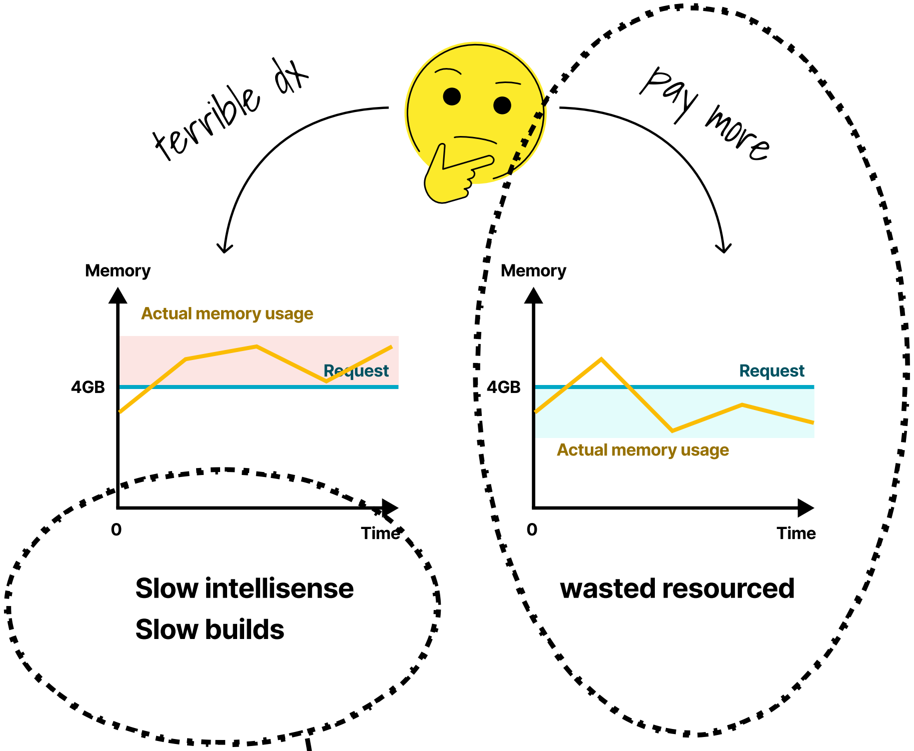
Slow intellisense

Slow builds

wasted resourced

p99 latency
(e.g. in API calls)

frequent scaling
up/down



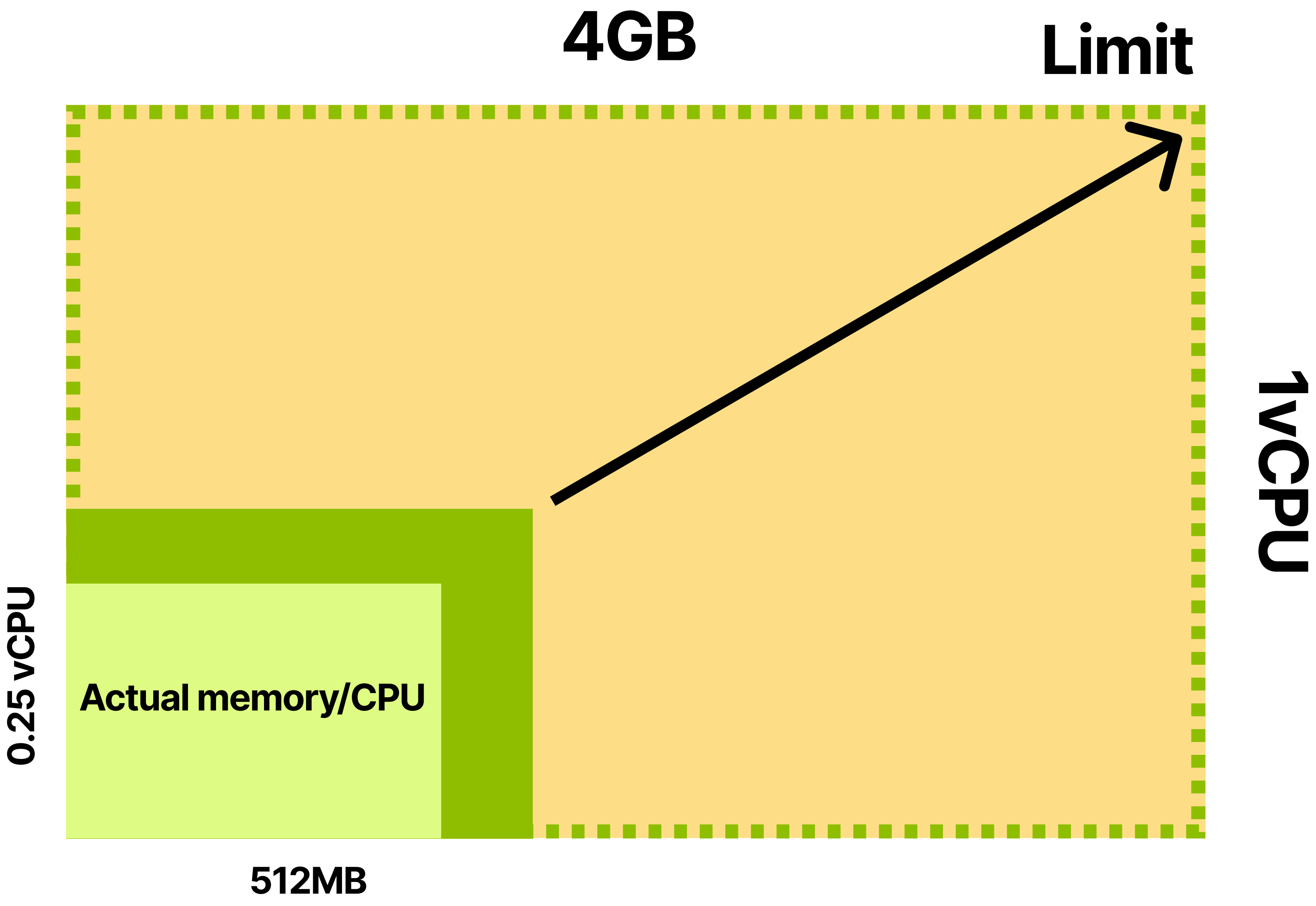
Slow intellisense
Slow builds

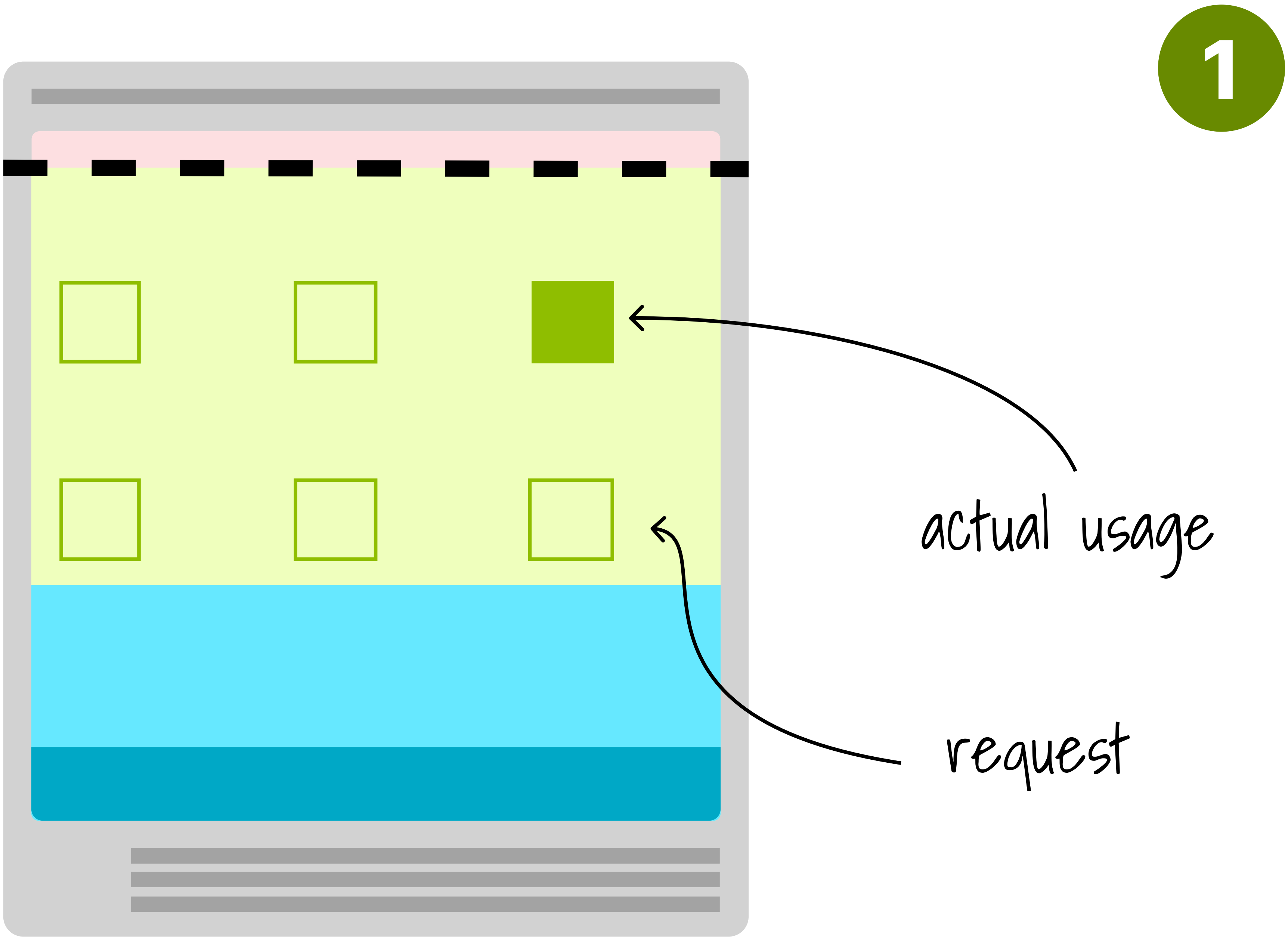
wasted resourced

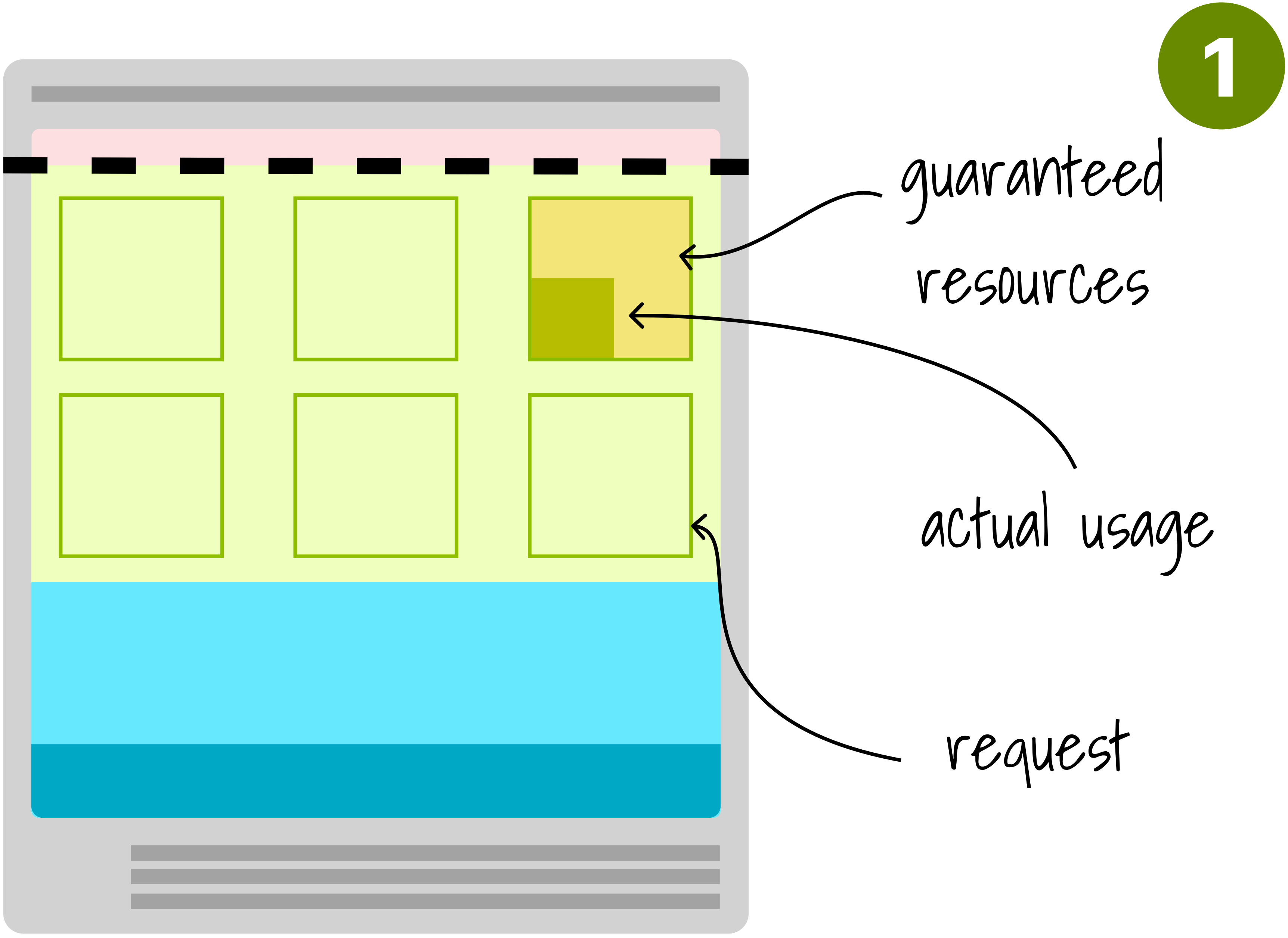
trade-off
between costs
an reliability

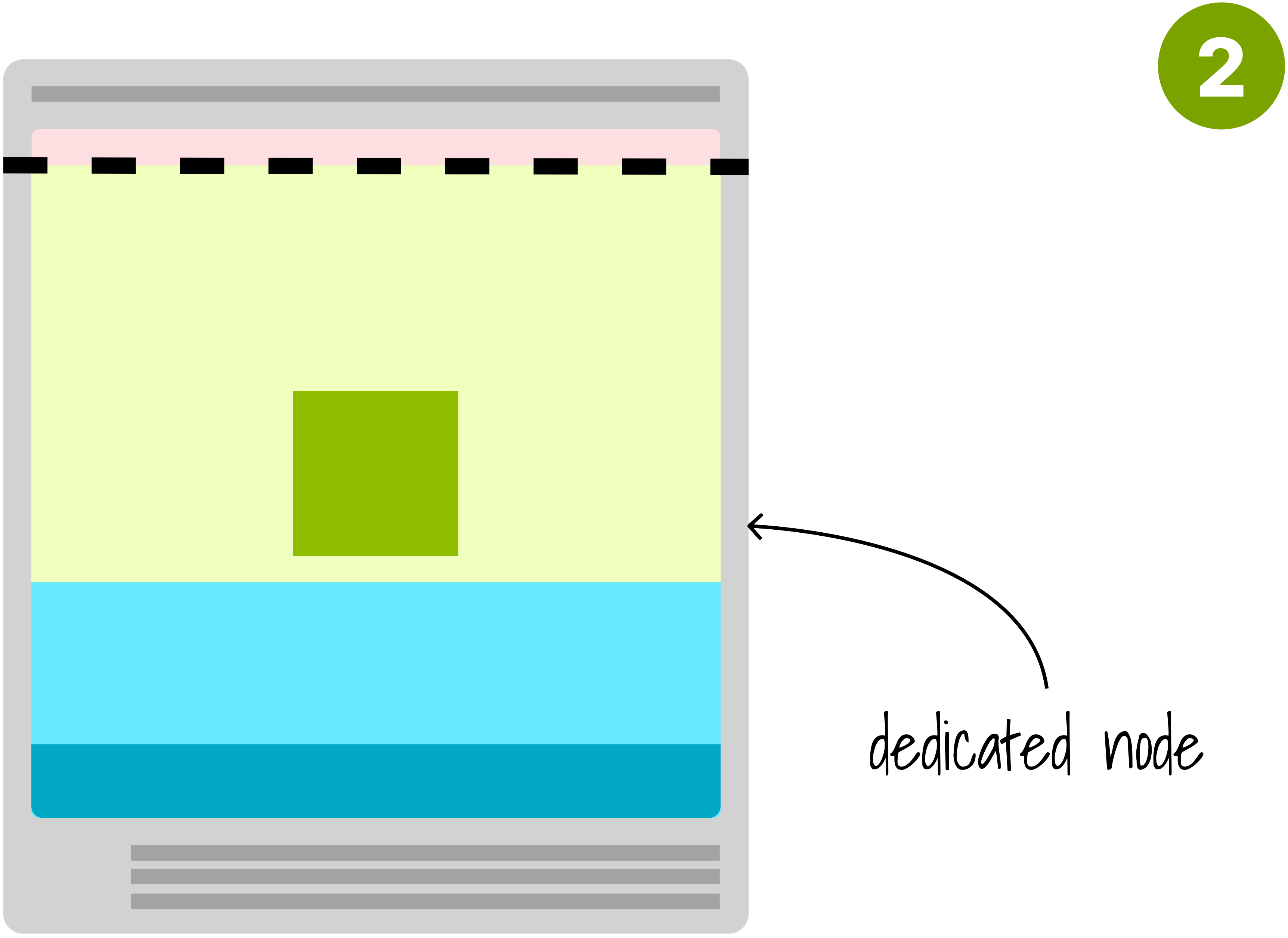
p99 latency
(e.g. in API calls)

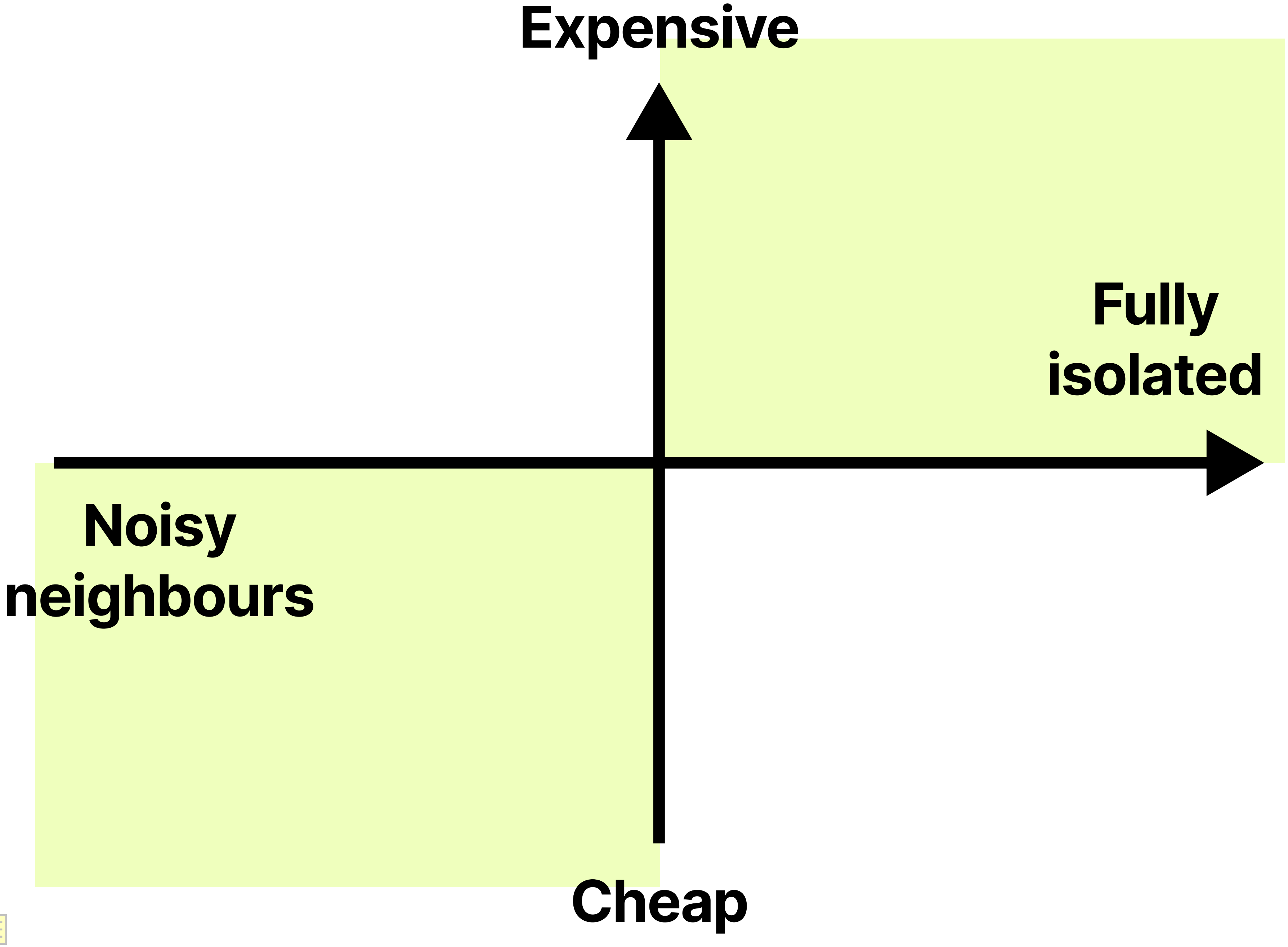
frequent scaling
up/down


4GB
Limit
1vCPU
0.25 vCPU
Actual memory/CPU
512MB


1
actual usage
request


1
guaranteed resources
actual usage
request


2
dedicated node

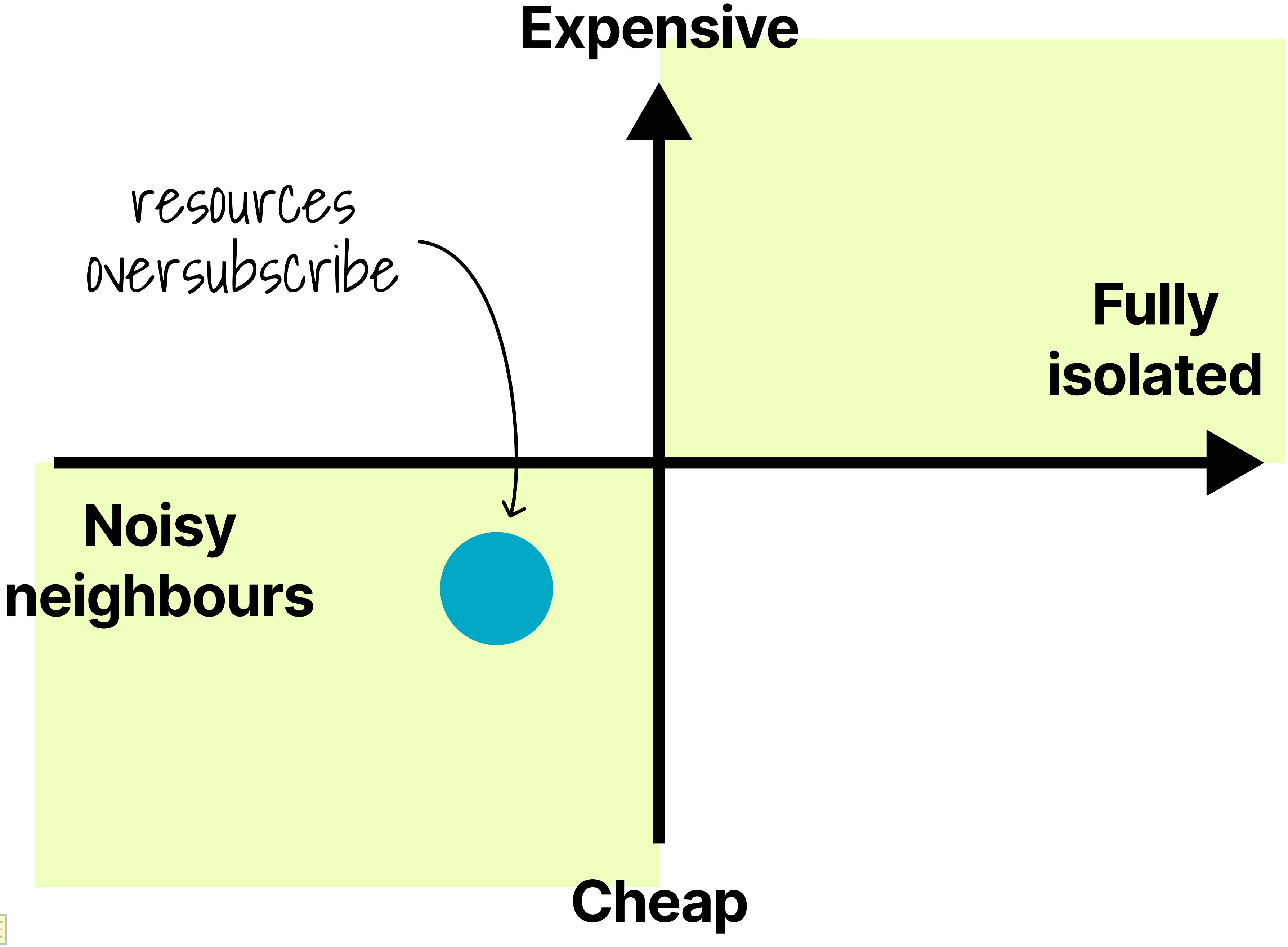


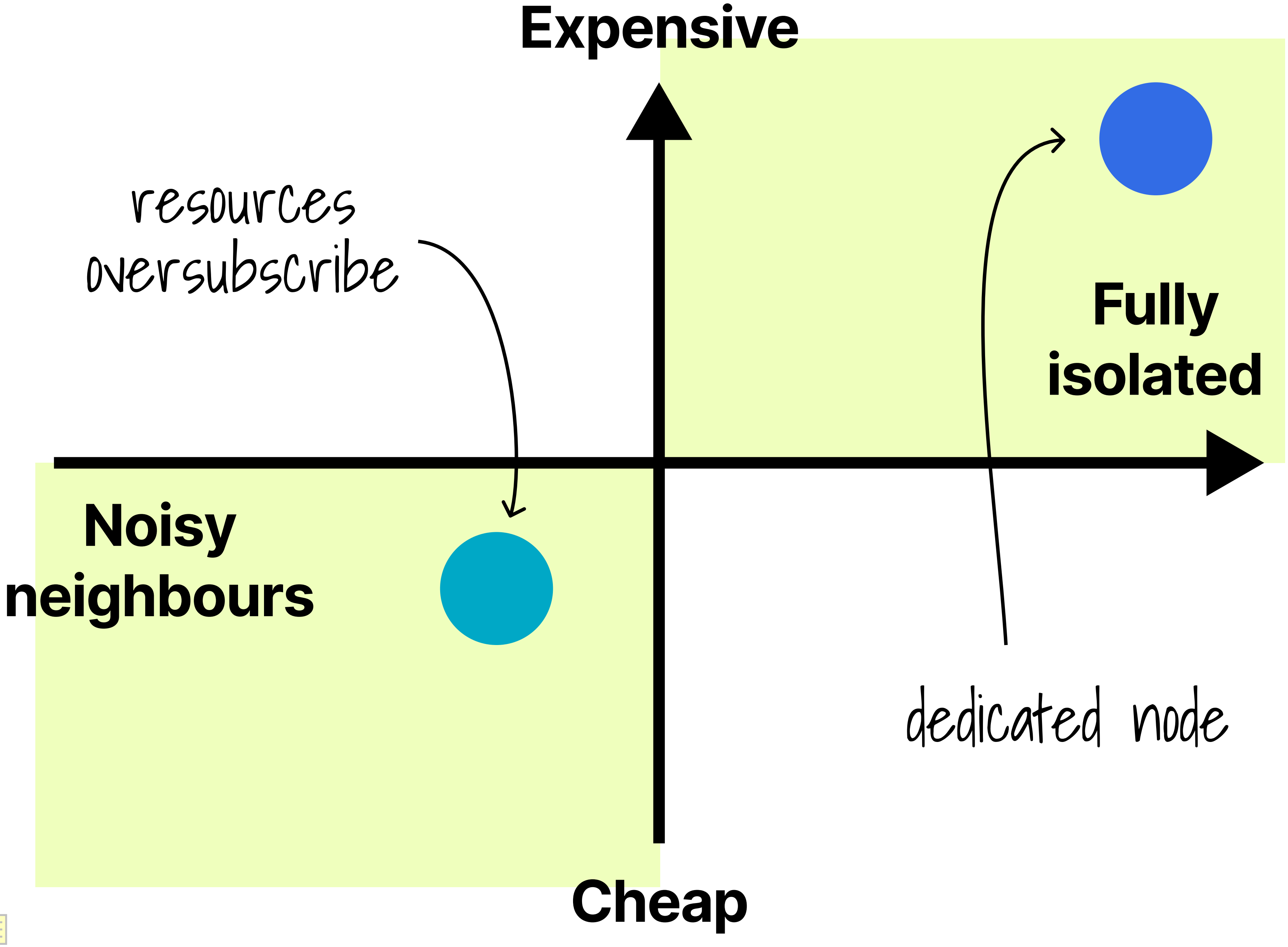
Expensive

Fully isolated

Noisy neighbours

Cheap


Expensive
resources oversubscribe
Fully isolated
Noisy neighbours
Cheap



Expensive

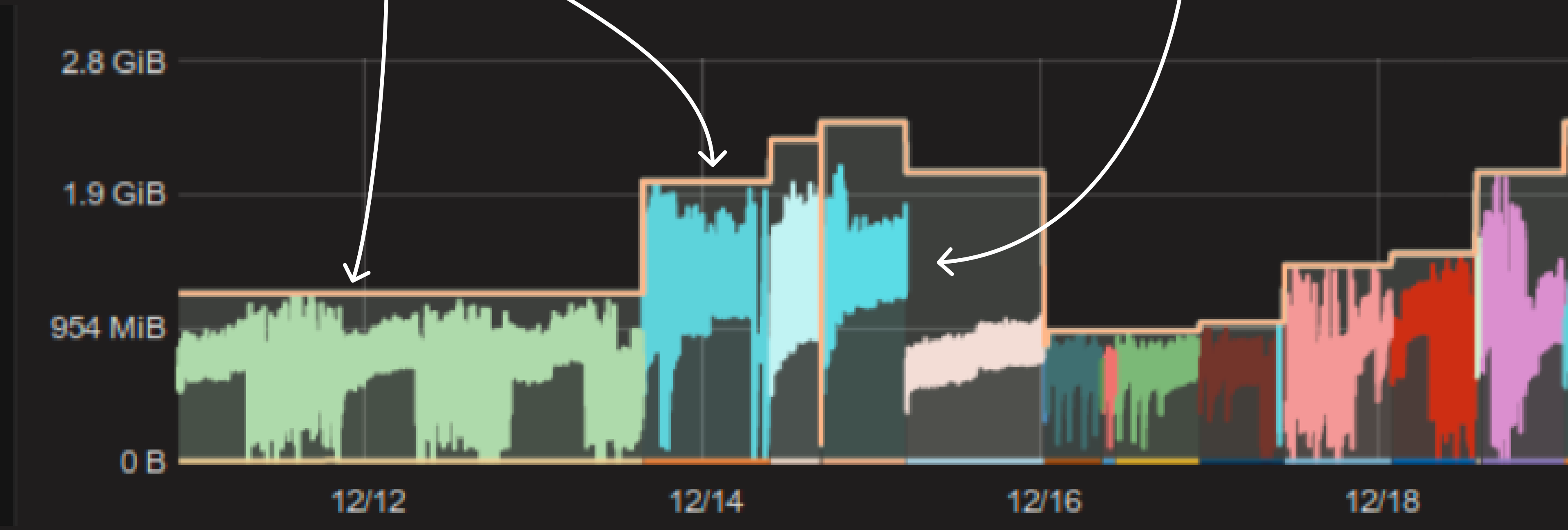
Fully isolated

Noisy neighbours

Cheap

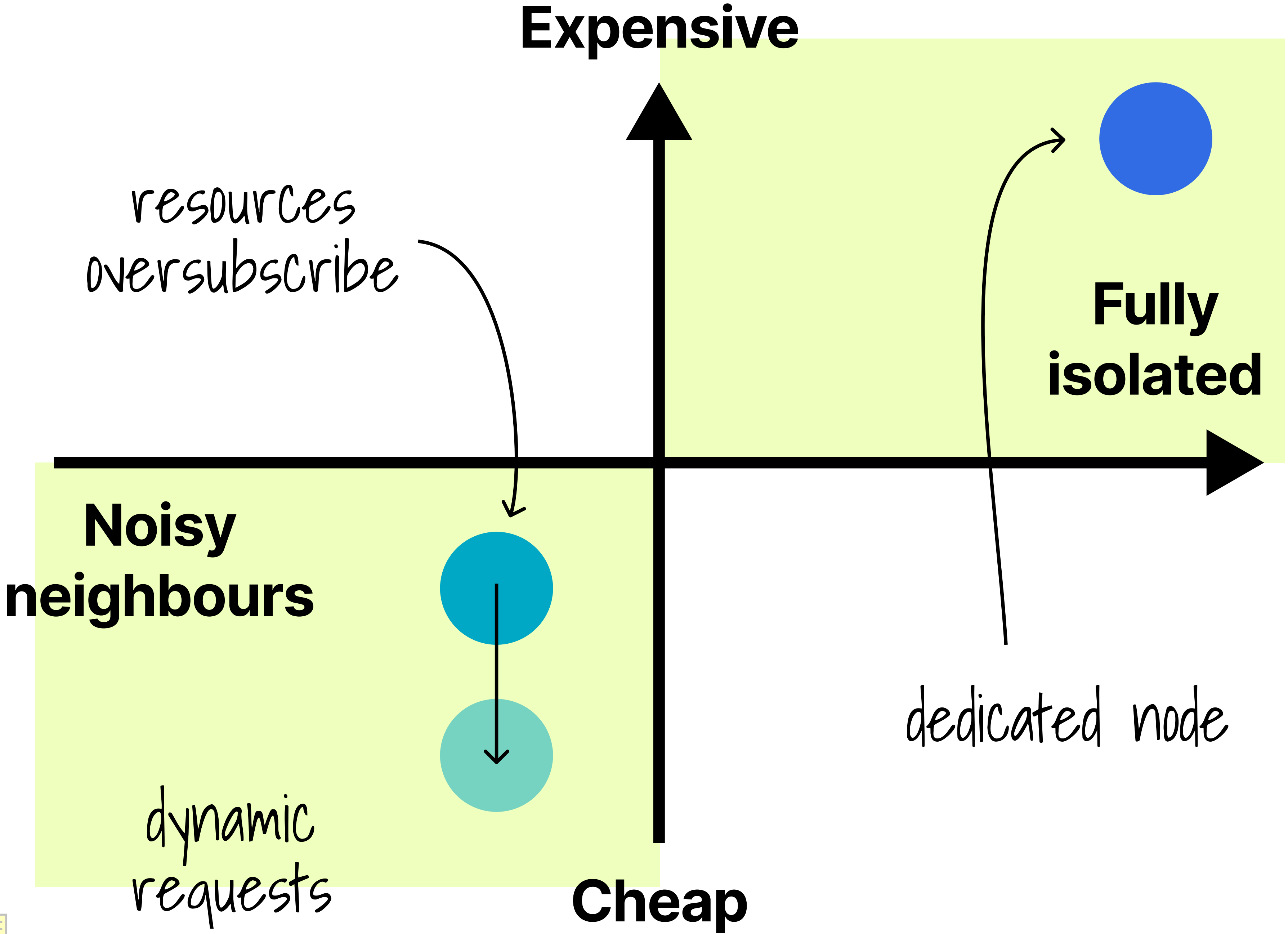
resources oversubscribe

dedicated node



memory request adjusted by the vertical pod autoscaler

actual memory usage



Expensive

resources oversubscribe

Fully isolated

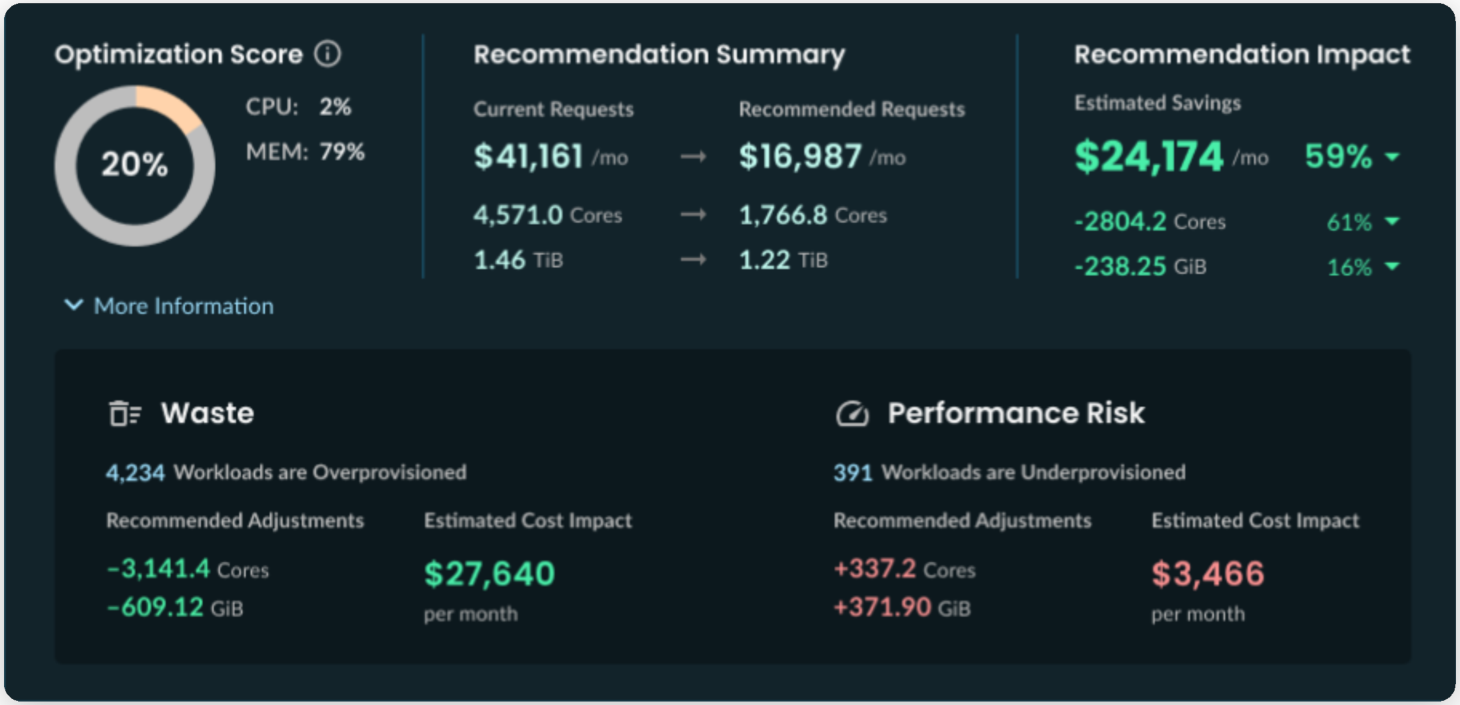
Noisy neighbours

dedicated node

dynamic requests

Cheap

Stormforge Optimize Live



Optimization Score ⓘ

20%

CPU: 2%
MEM: 79%

Recommendation Summary

Current Requests		Recommended Requests
$41,161 /mo	→	$16,987 /mo
4,571.0 Cores	→	1,766.8 Cores
1.46 TiB	→	1.22 TiB

Recommendation Impact

Estimated Savings

$24,174 /mo	59% ▾
-2804.2 Cores	61% ▾
-238.25 GiB	16% ▾

⌄ More Information

Waste

4,234 Workloads are Overprovisioned

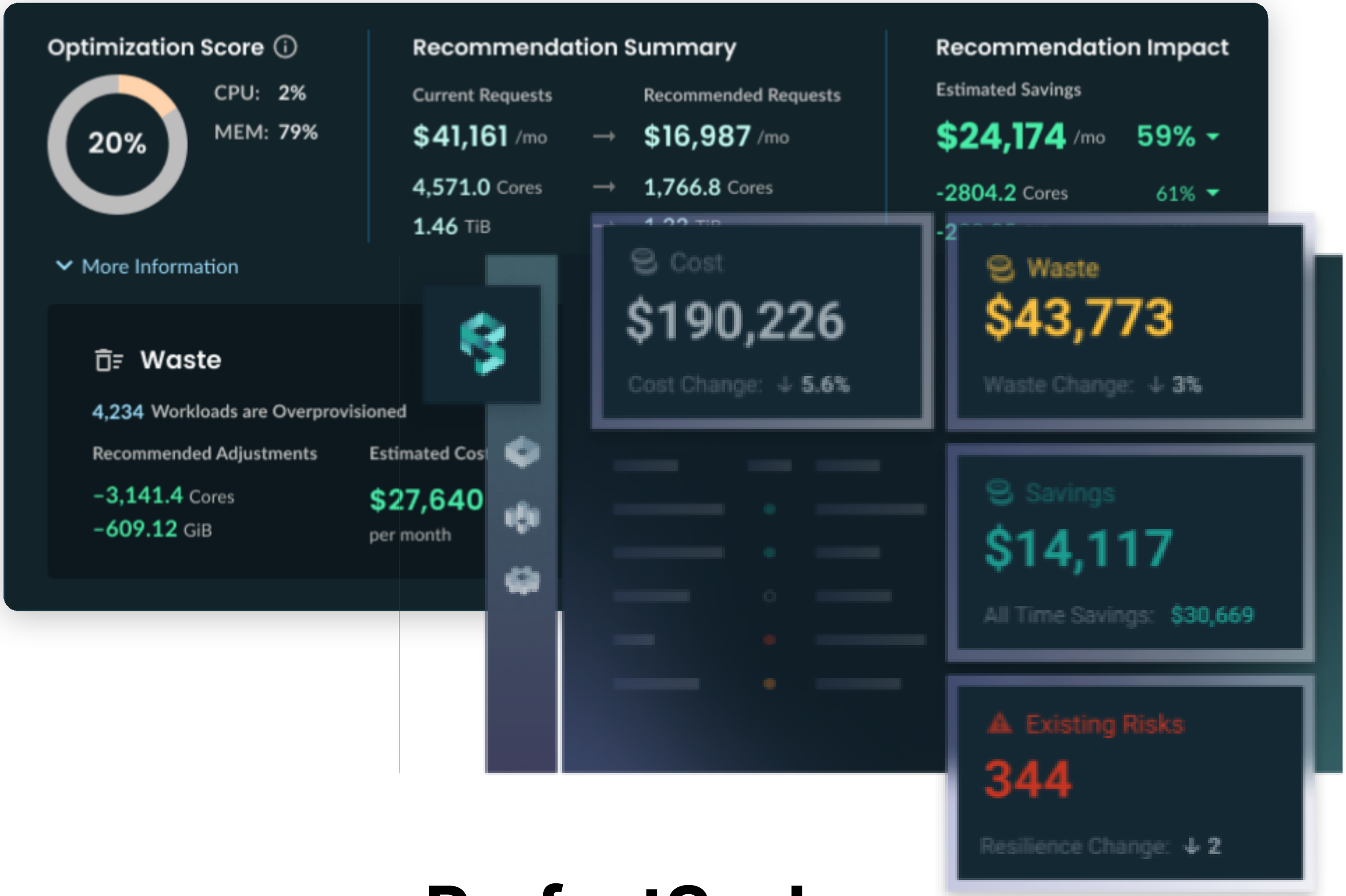
Recommended Adjustments	Estimated Cost Impact
-3,141.4 Cores -609.12 GiB	$27,640 per month

Performance Risk

391 Workloads are Underprovisioned

Recommended Adjustments	Estimated Cost Impact
+337.2 Cores +371.90 GiB	$3,466 per month

Stormforge Optimize Live



Optimization Score ⓘ

20%

CPU: 2%

MEM: 79%

More Information

Recommendation Summary

Current Requests		Recommended Requests
$41,161 /mo	→	$16,987 /mo
4,571.0 Cores	→	1,766.8 Cores
1.46 TiB		

Recommendation Impact

Estimated Savings

$24,174 /mo 59% ▾

-2804.2 Cores 61% ▾

Waste

4,234 Workloads are Overprovisioned

Recommended Adjustments	Estimated Cos
-3,141.4 Cores -609.12 GiB	$27,640 per month

Cost

$190,226

Cost Change: ↓ 5.6%

Waste

$43,773

Waste Change: ↓ 3%

Savings

$14,117

All Time Savings: $30,669

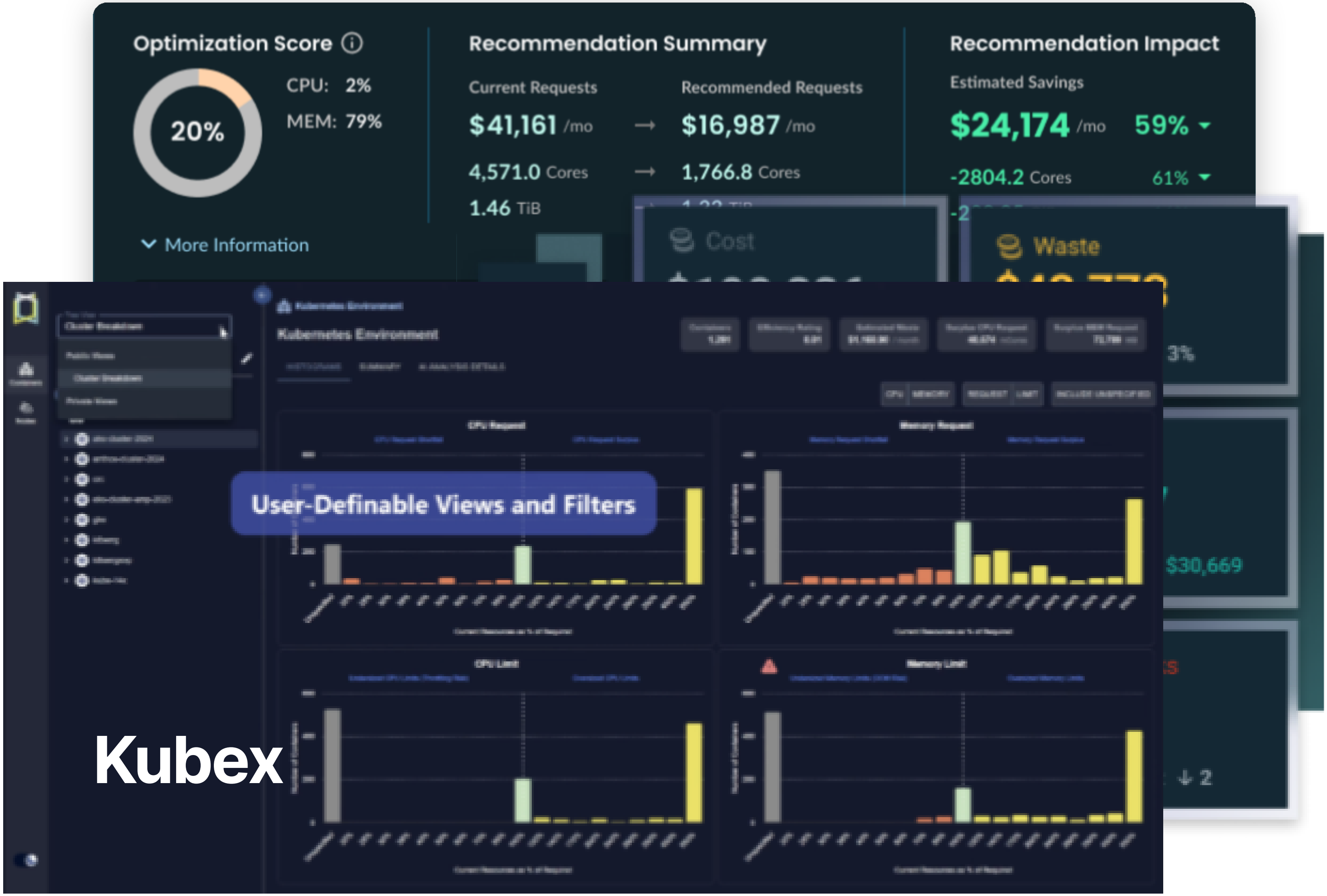
Existing Risks

344

Resilience Change: ↓ 2

PerfectScale

Stormforge Optimize Live



Optimization Score ⓘ

20%

CPU: 2%

MEM: 79%

More Information

Recommendation Summary

Current Requests		Recommended Requests
$41,161 /mo	→	$16,987 /mo
4,571.0 Cores	→	1,766.8 Cores
1.46 TiB		

Recommendation Impact

Estimated Savings

$24,174 /mo 59%

-2804.2 Cores 61%

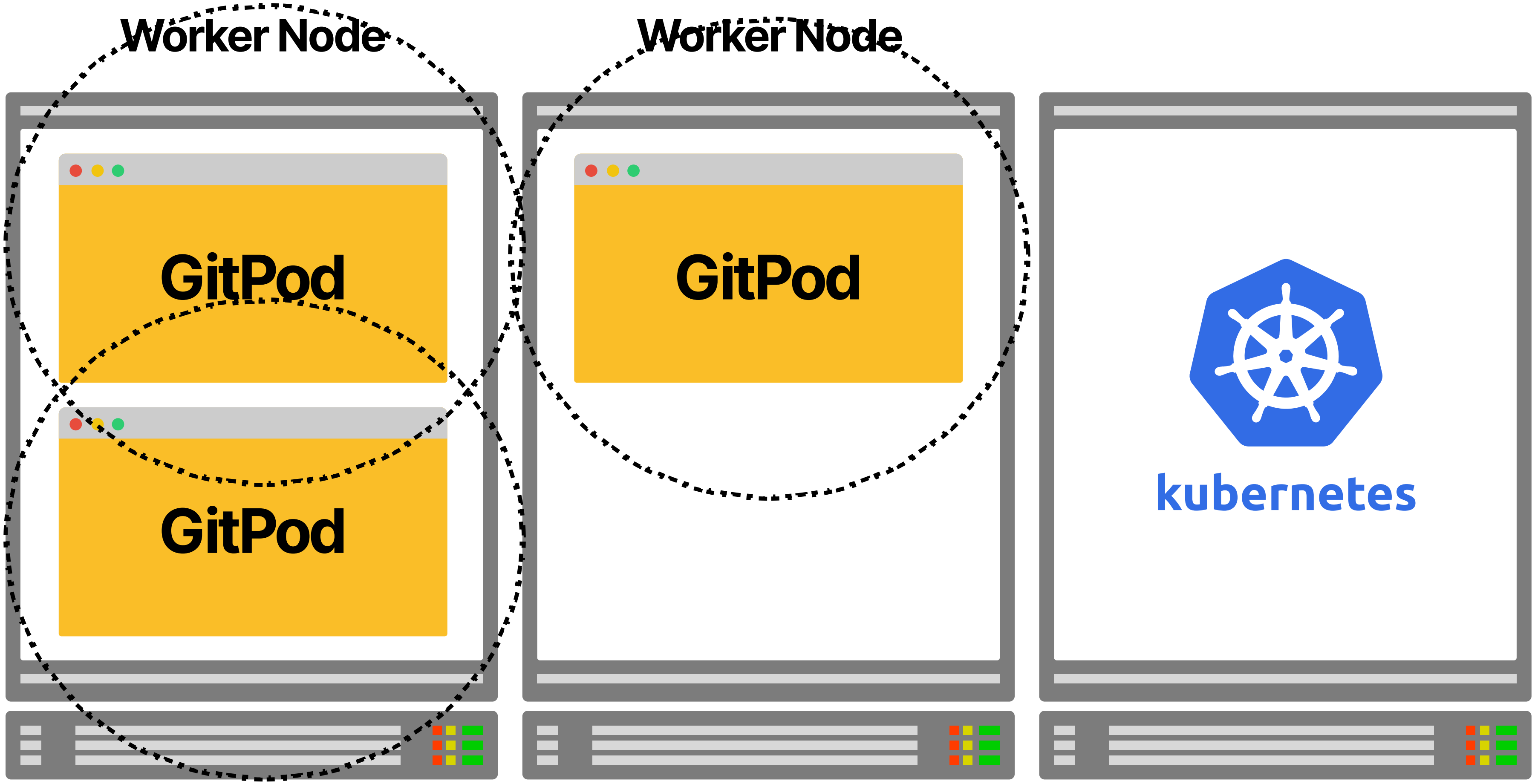
User-Definable Views and Filters

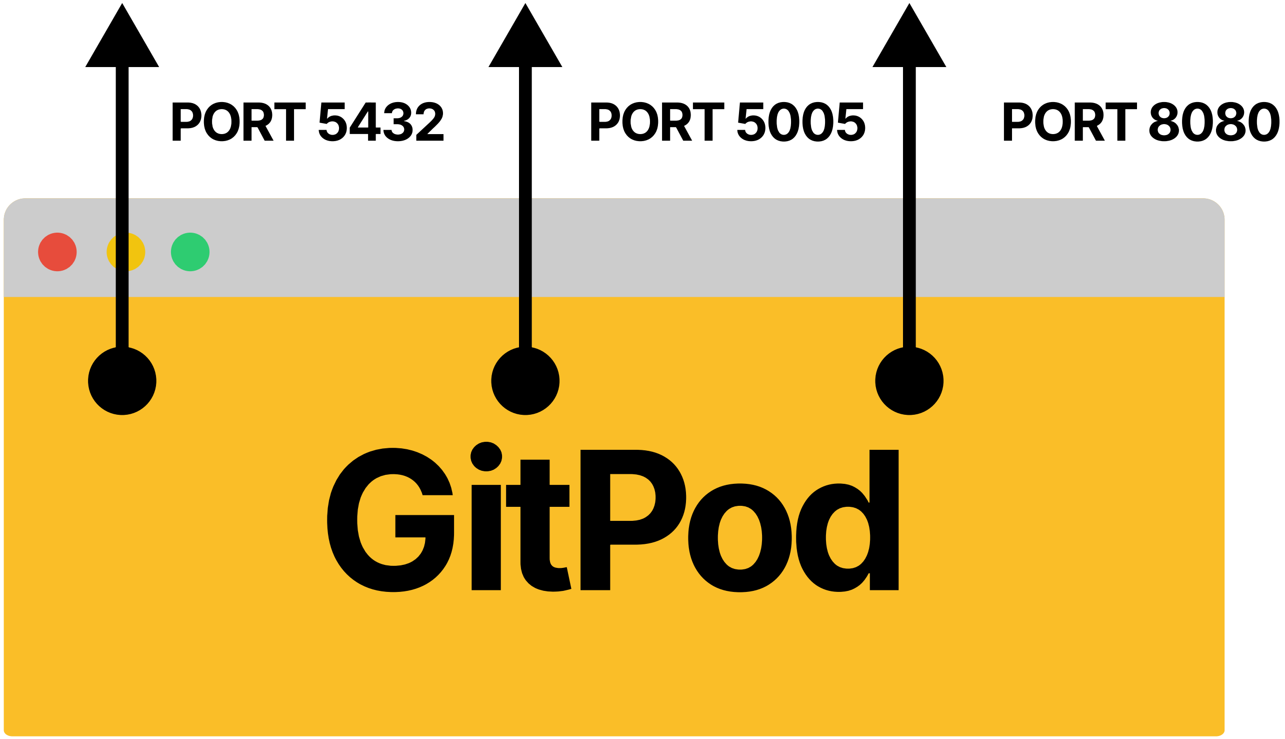
Kubex

Network complexity

Challenge 2

isolated network



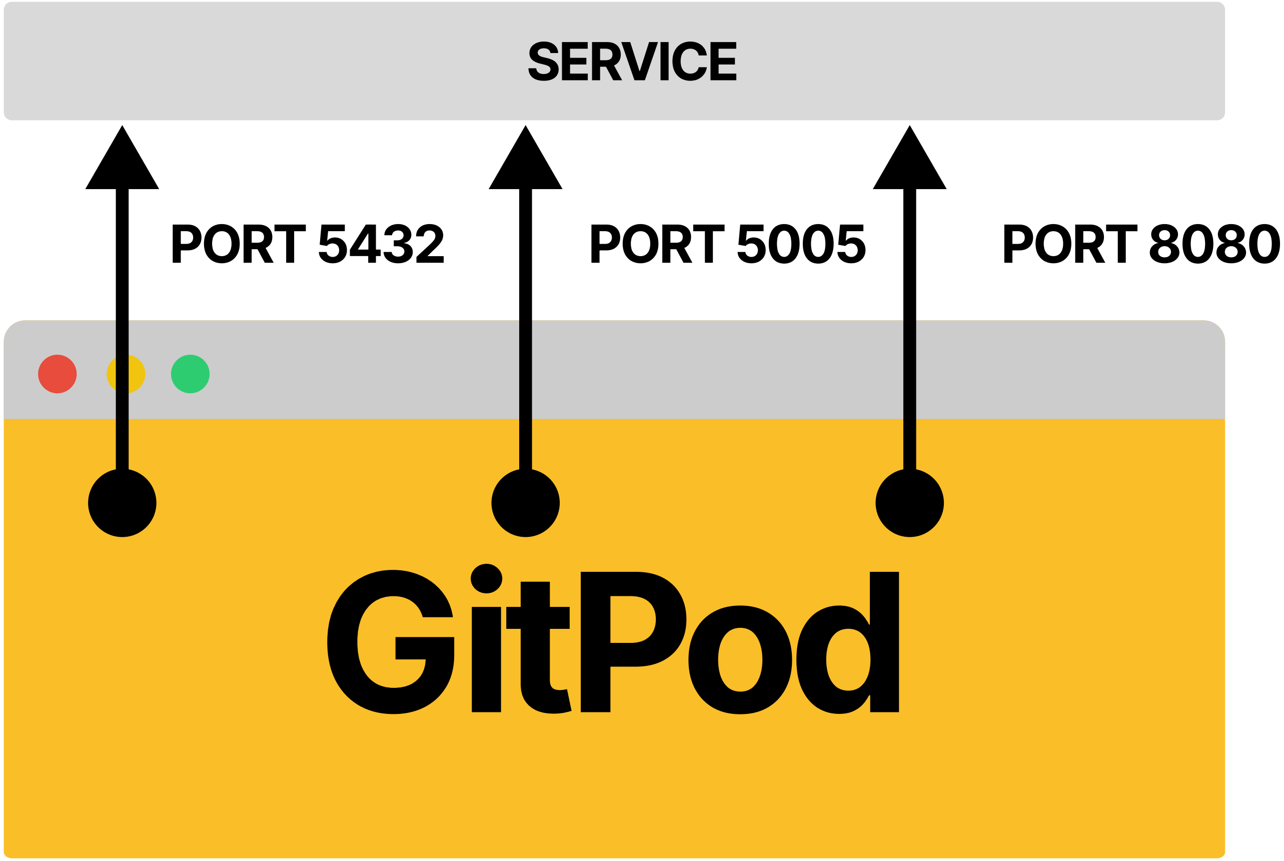


PORT 5432

PORT 5005

PORT 8080

GitPod

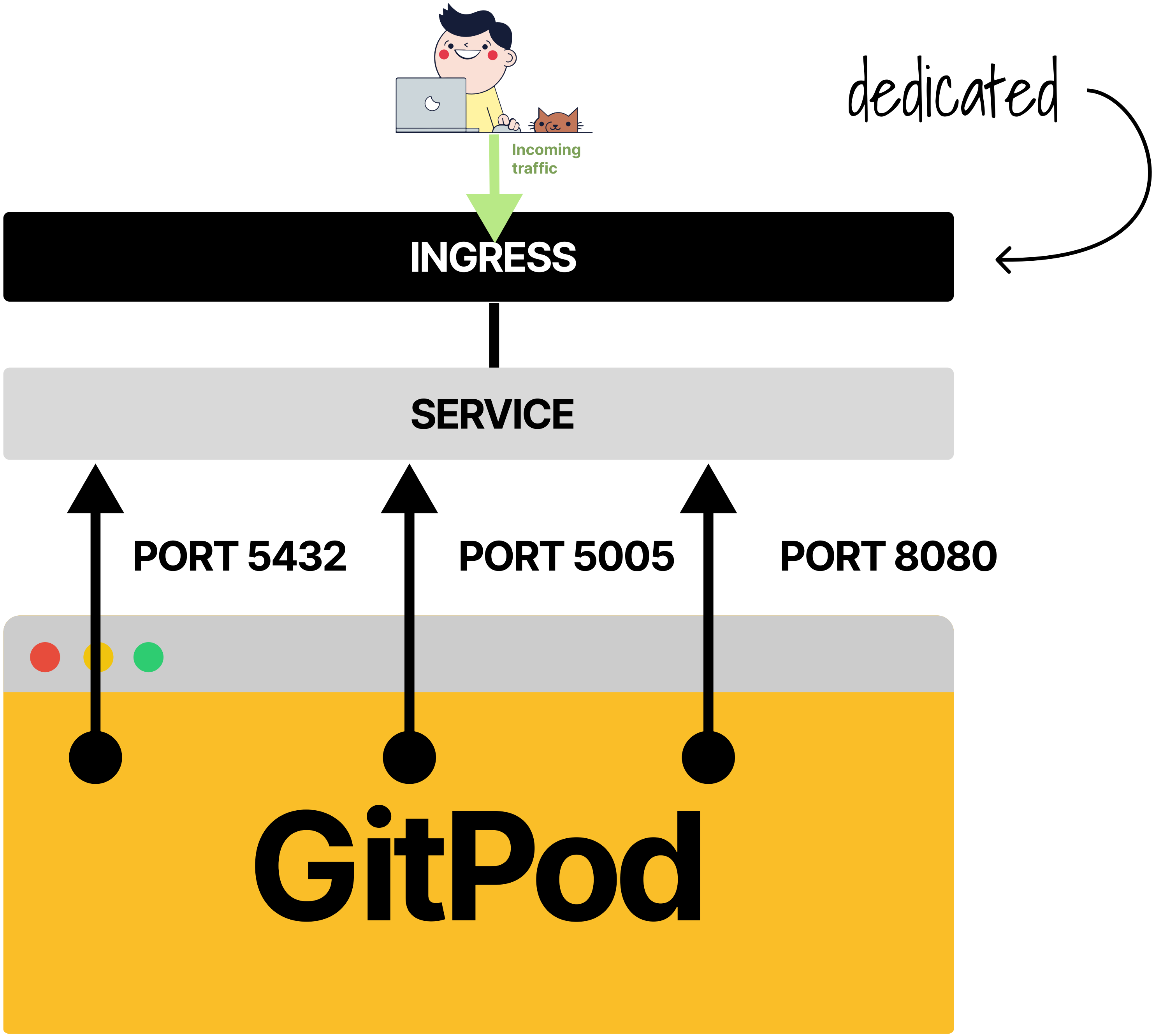

SERVICE

PORT 5432

PORT 5005

PORT 8080

GitPod



dedicated

Incoming traffic

INGRESS

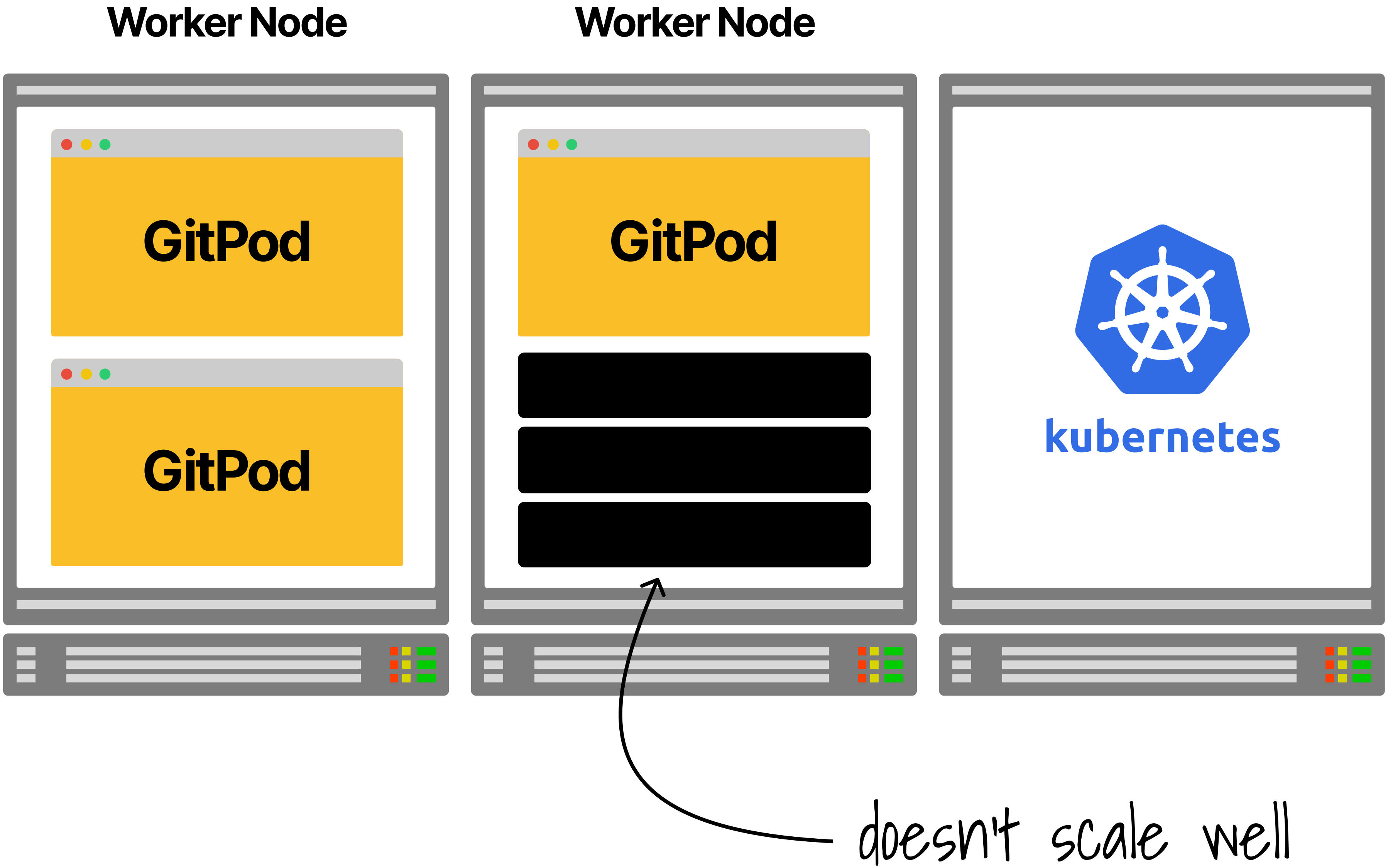
SERVICE

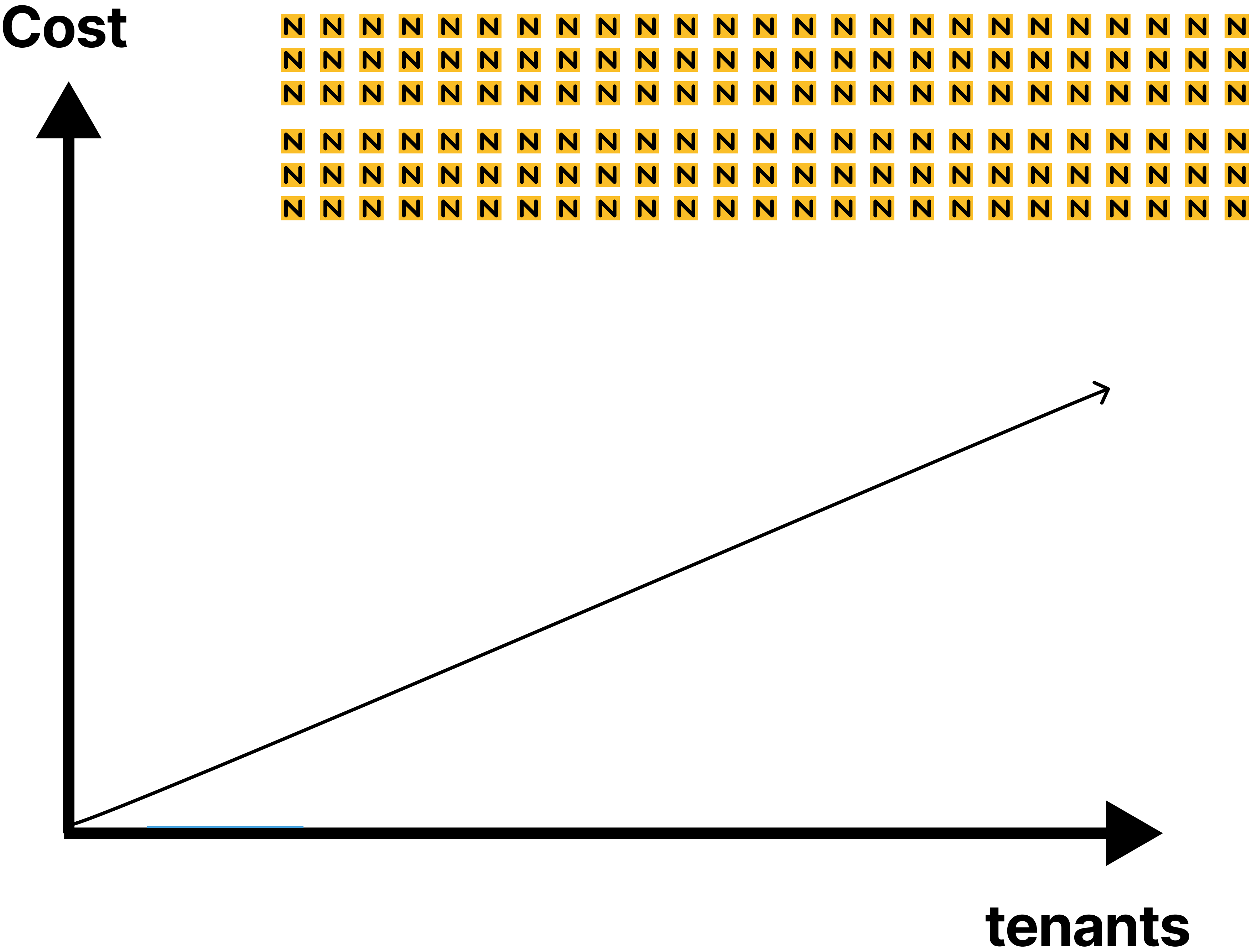
PORT 5432

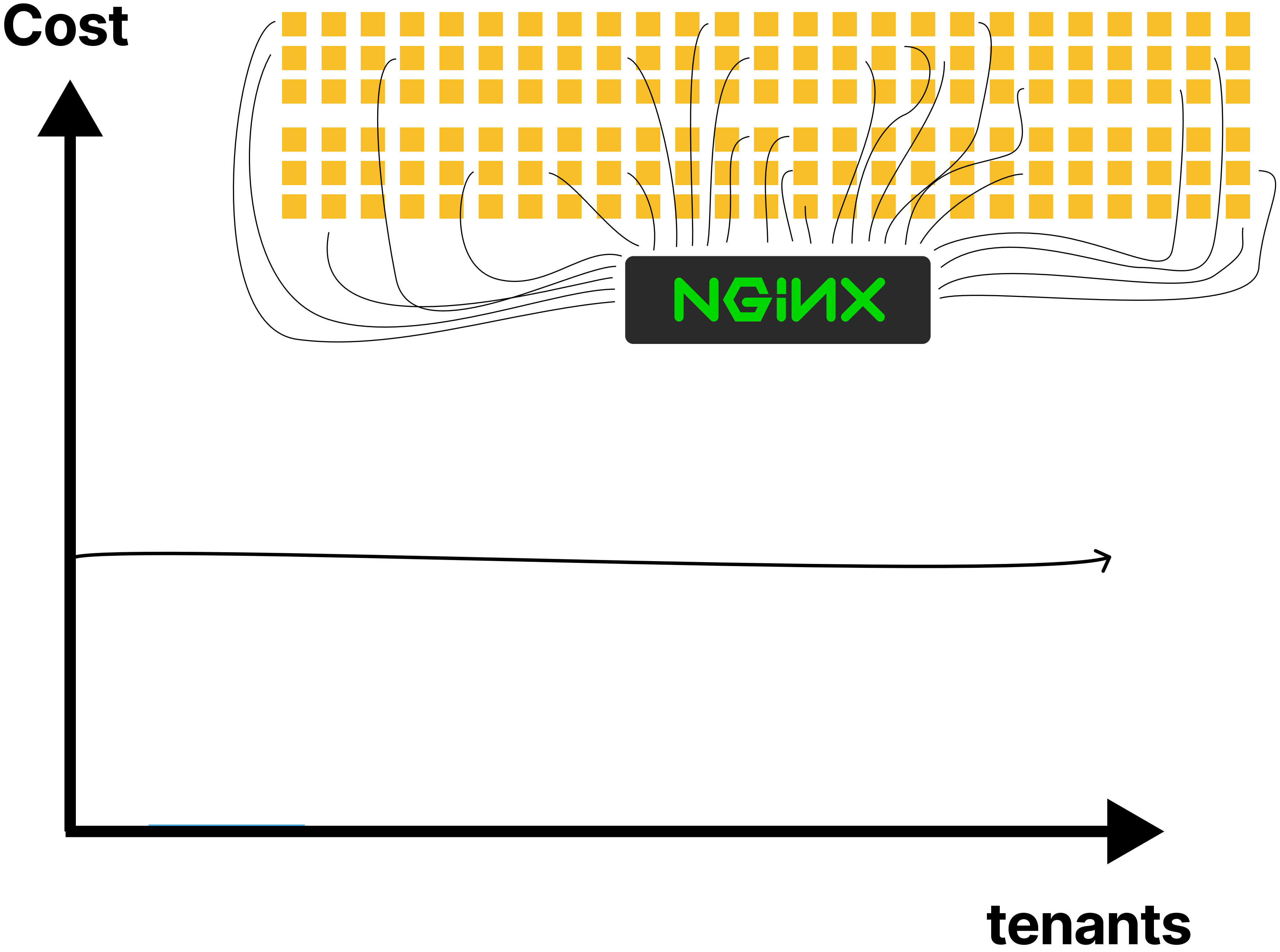
PORT 5005

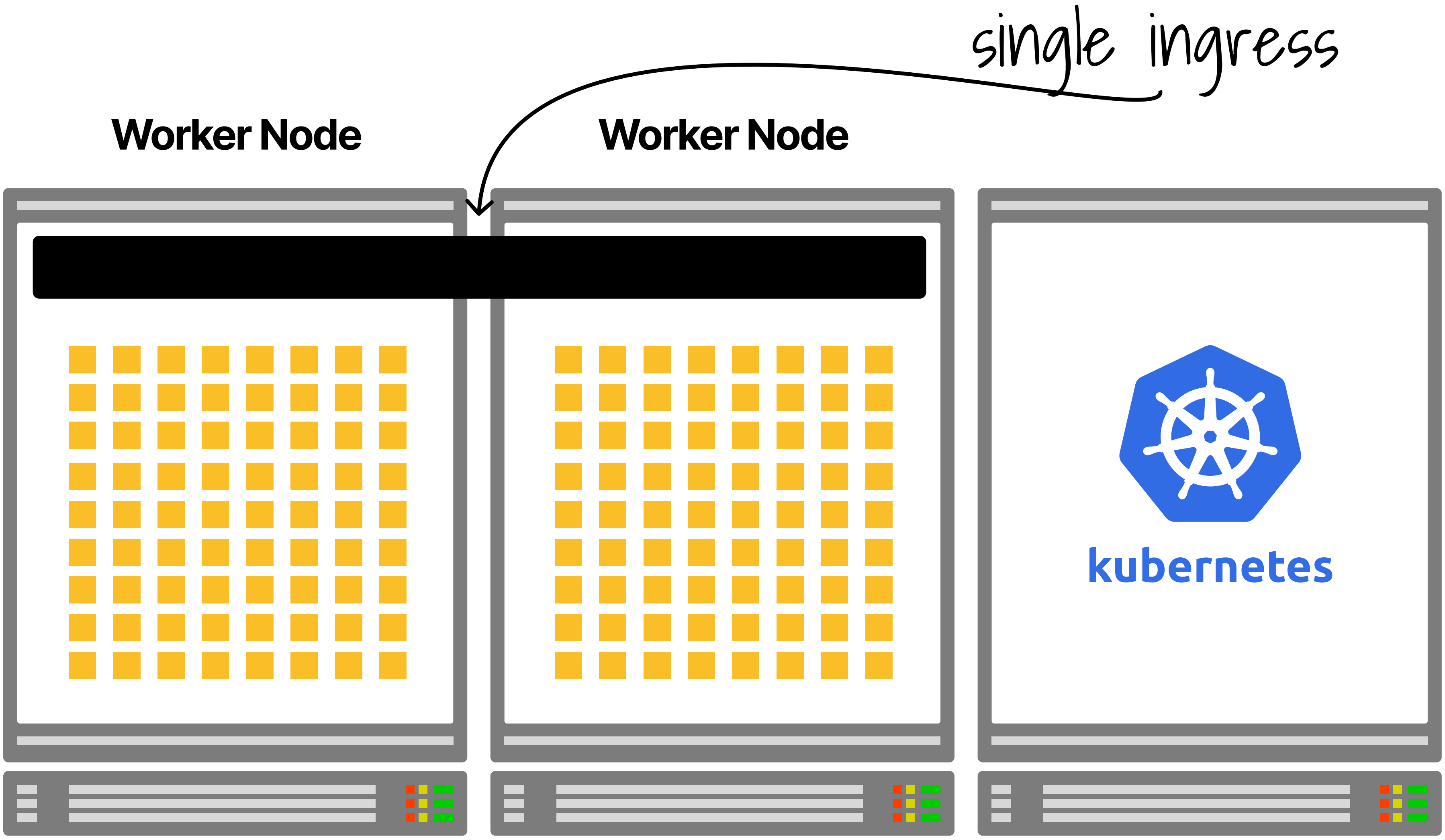
PORT 8080

GitPod


Worker Node
Worker Node
GitPod
GitPod
GitPod
kubernetes
doesn't scale well


Cost
tenants


Cost
NGINX
tenants



Worker Node

Worker Node

Network complexity

Too many services

enableServiceLinks

Failing DNS

Network complexity

Too many services

enableServiceLinks

Failing DNS

Network complexity

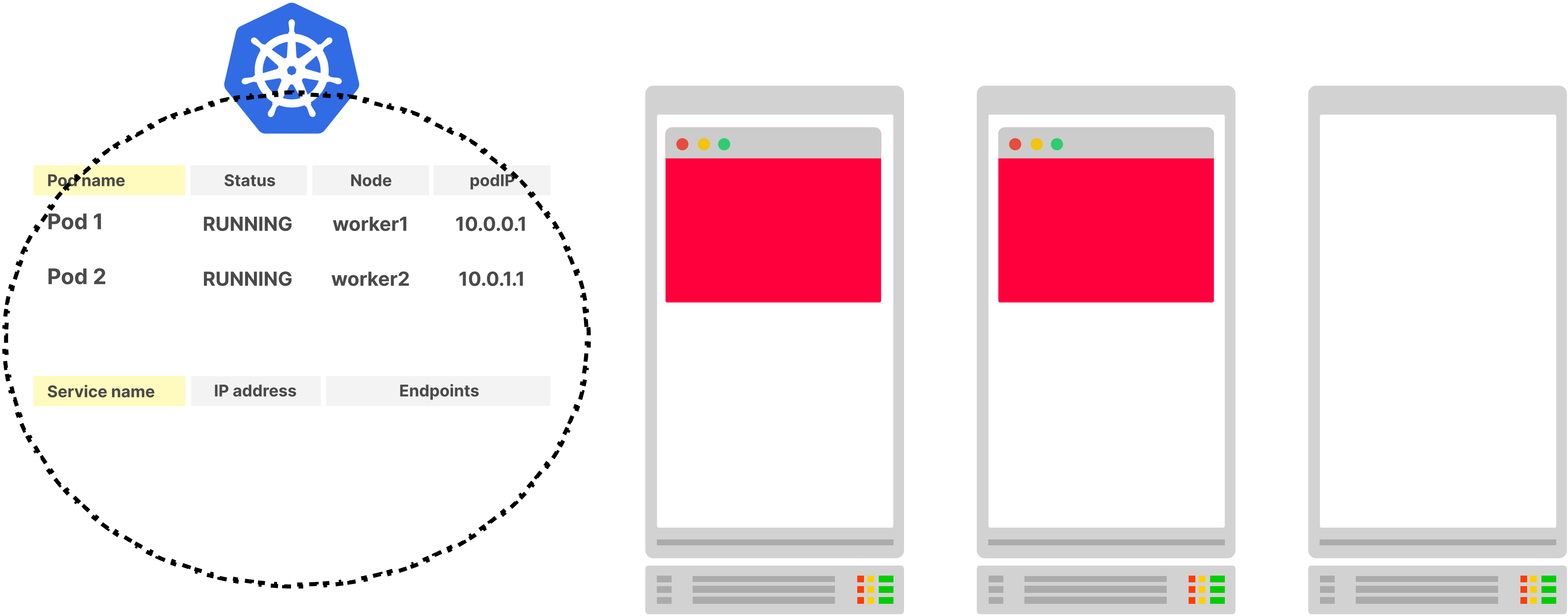
Too many services

enableServiceLinks

Failing DNS

How Services work

state in etcd



Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

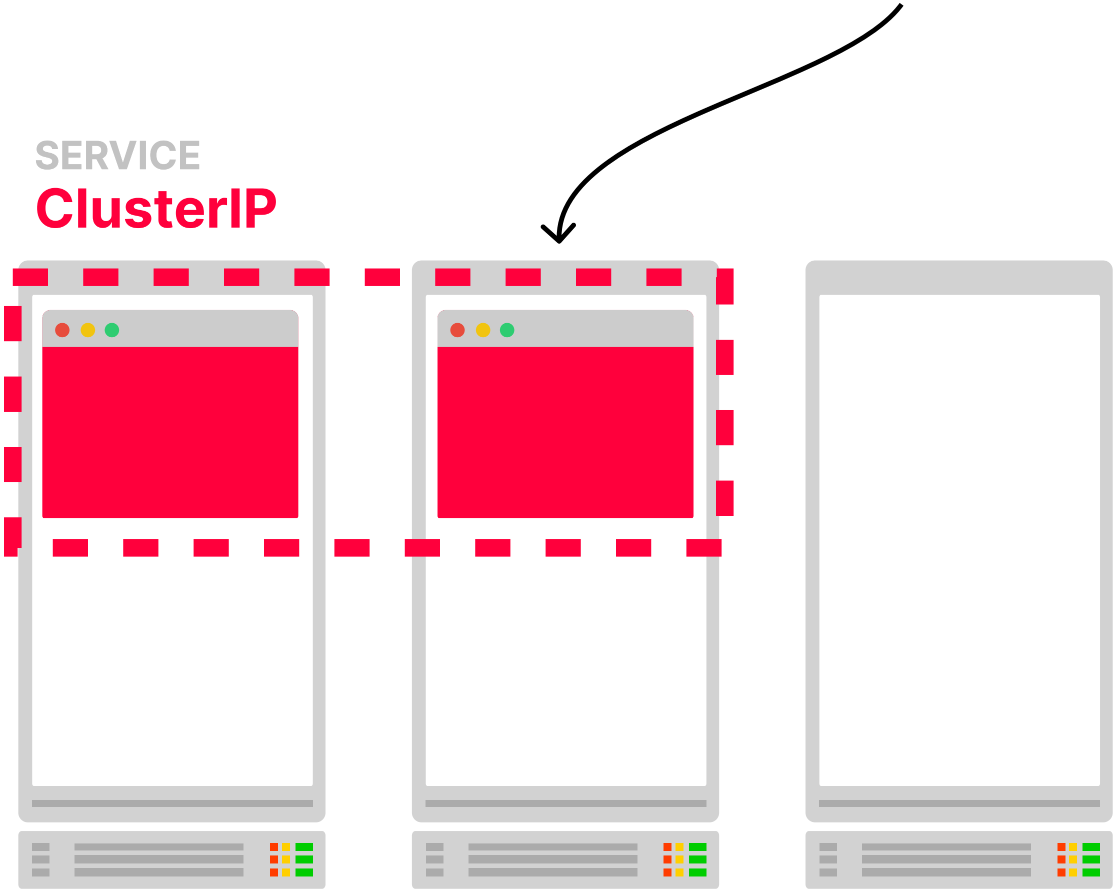
Service name	IP address	Endpoints

cluster

when you create a Service...

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

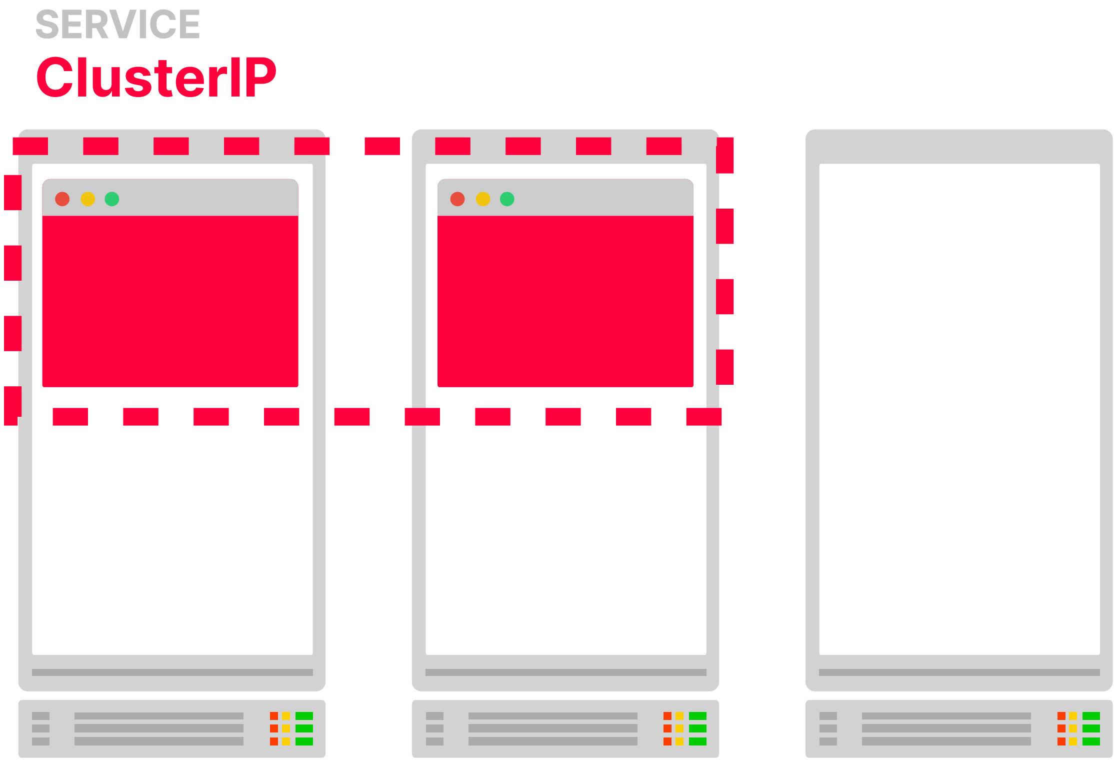
Service name	IP address	Endpoints



...the endpoint controller collects the endpoints

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

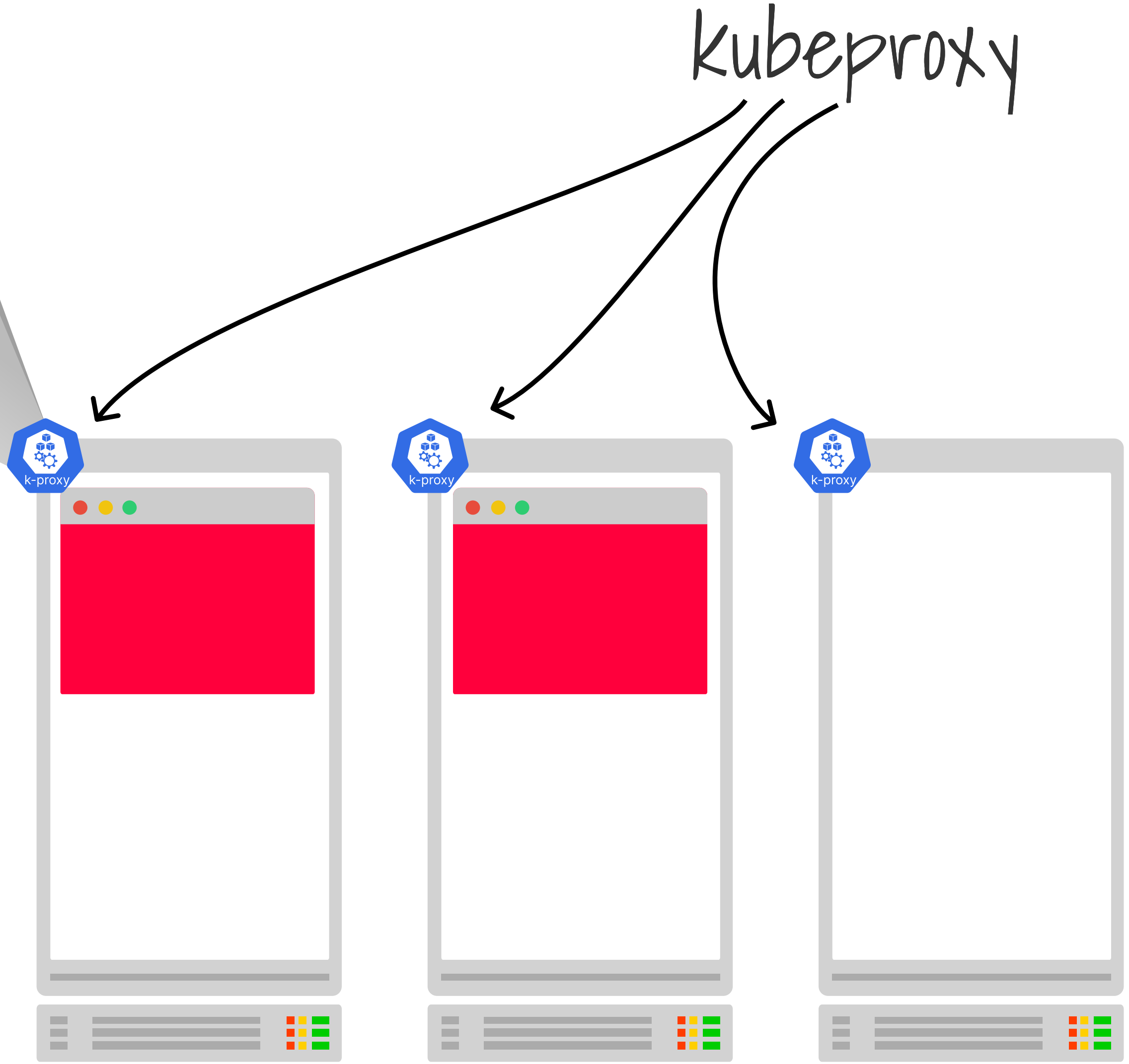
Service name	IP address	Endpoints
Red	172.17.0.1	10.0.0.1:3000,10.0.1.1:3000



IP address to intercept	Replace with

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

Service name	IP address	Endpoints
Red	172.17.0.1	10.0.0.1:3000,10.0.1.1:3000

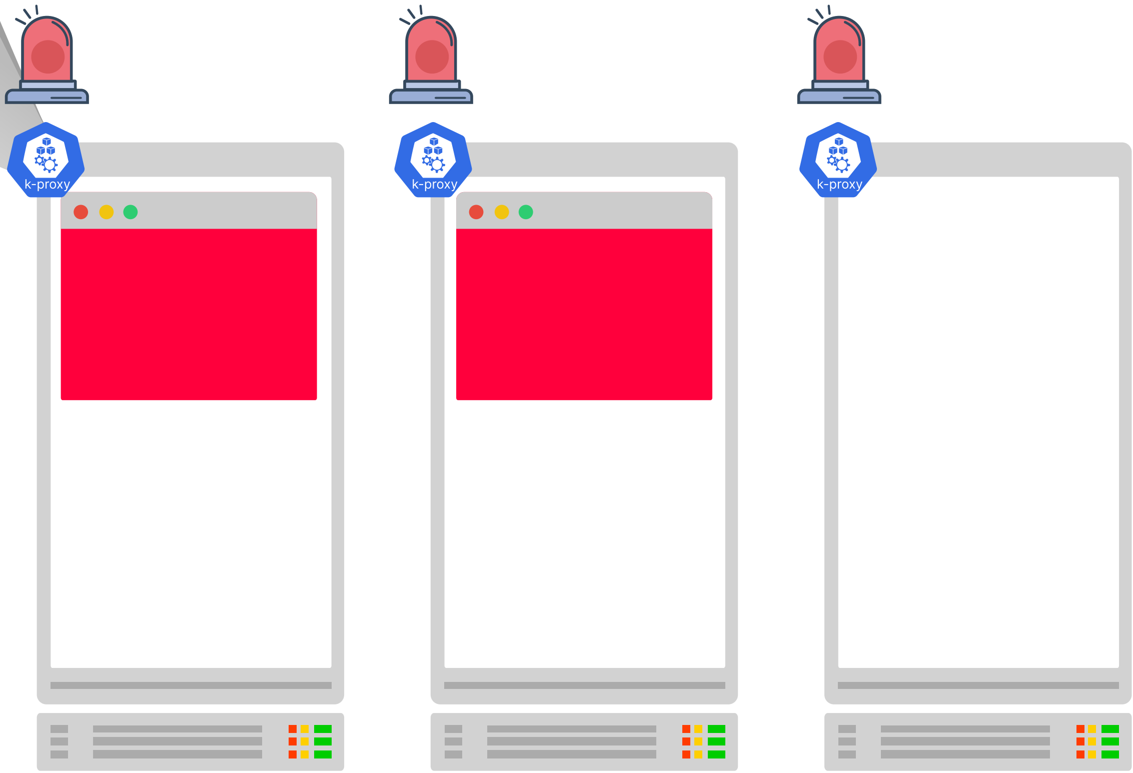


kubeproxy is notified of the endpoints

IP address to intercept	Replace with
172.17.0.1	10.0.0.1, 10.0.1.1

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

Service name	IP address	Endpoints
Red	172.17.0.1	10.0.0.1:3000,10.0.1.1:3000

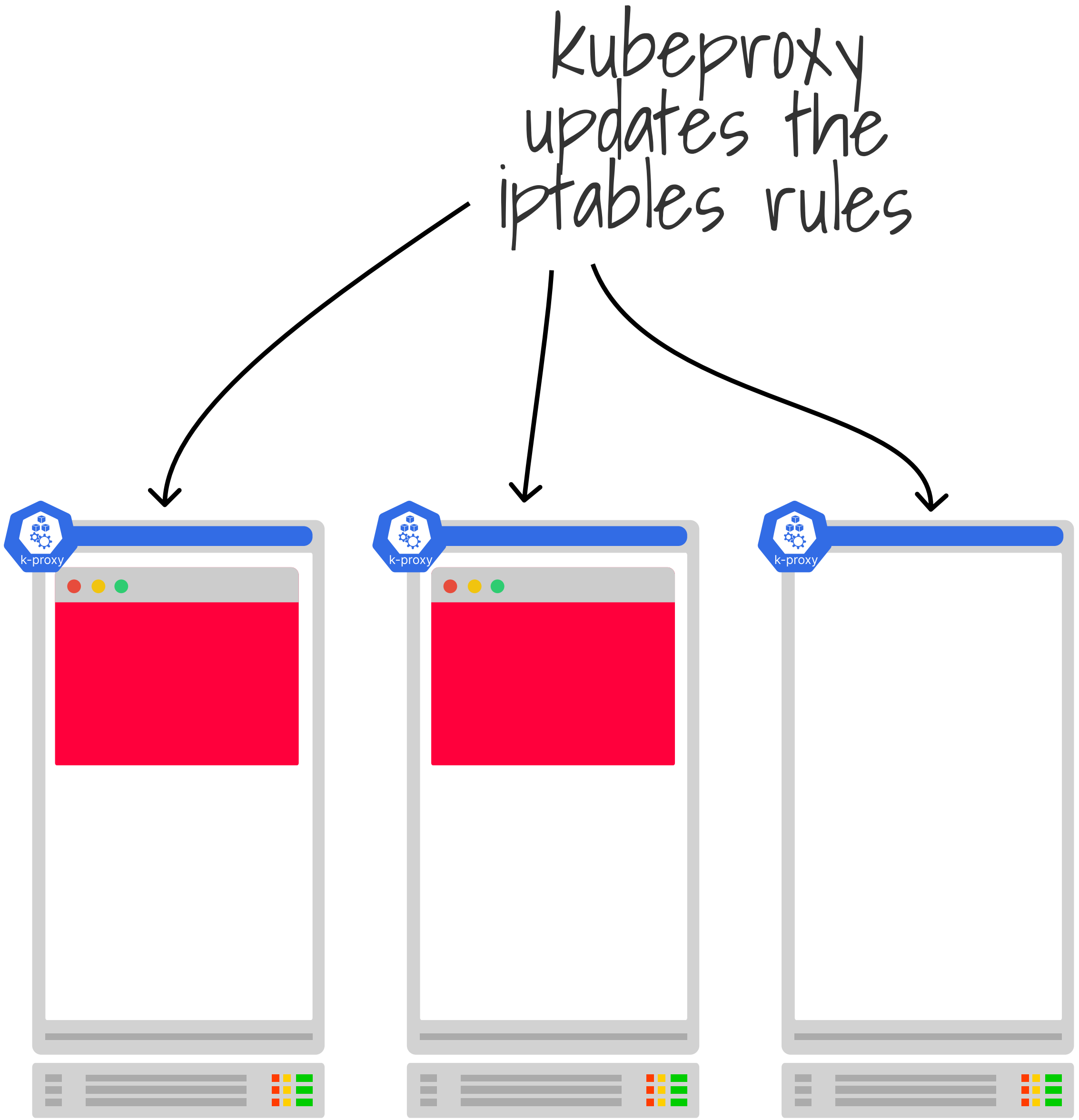


IP address to intercept	Replace with
172.17.0.1	10.0.0.1, 10.0.1.1

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

Service name	IP address	Endpoints
Red	172.17.0.1	10.0.0.1:3000,10.0.1.1:3000

kubeproxy updates the iptables rules



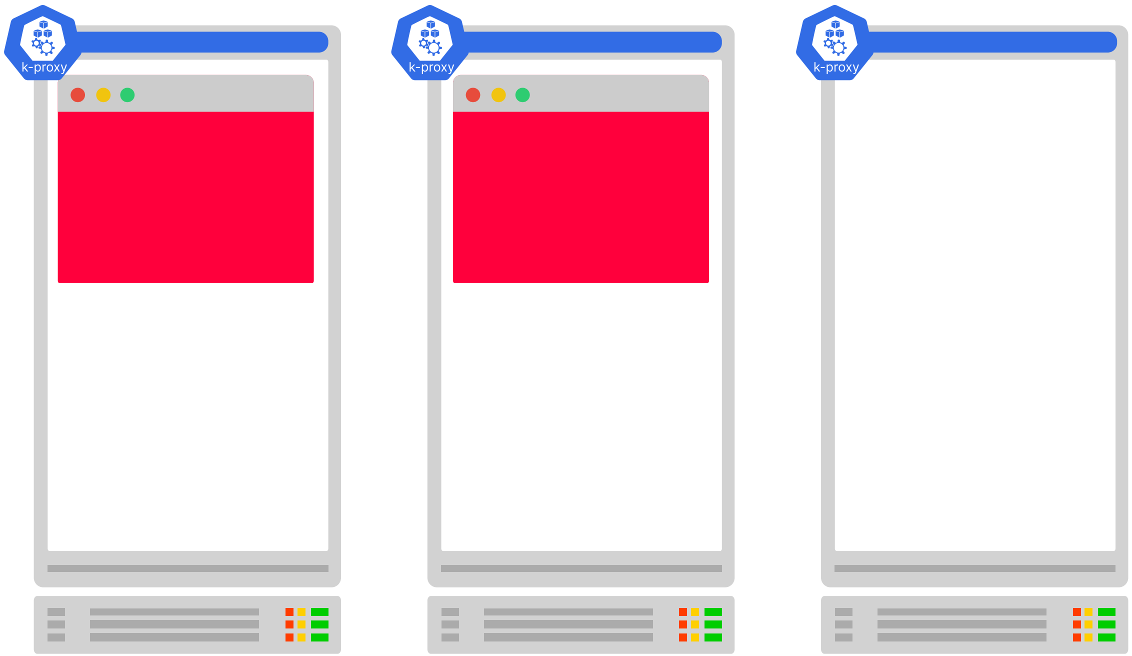
IP address to intercept	Replace with
172.17.0.1	10.0.0.1, 10.0.1.1

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

Service name	IP address	Endpoints
Red	172.17.0.1	10.0.0.1:3000,10.0.1.1:3000

Service name	IP addresses

coreDNS is also notified!



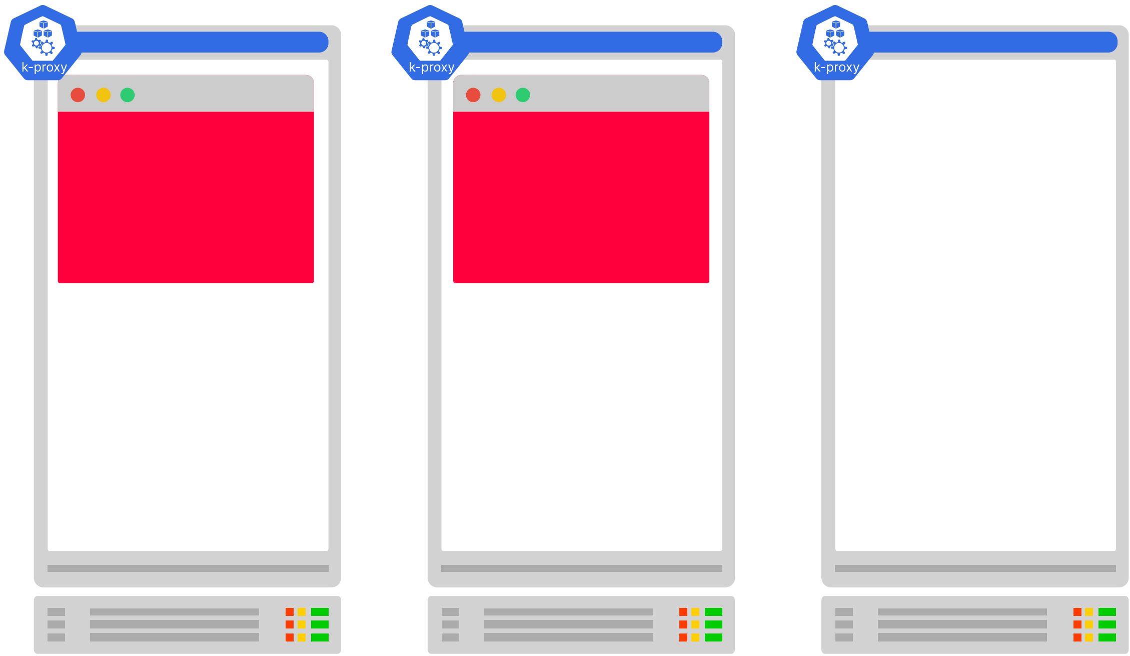
IP address to intercept	Replace with
172.17.0.1	10.0.0.1, 10.0.1.1

Service name	IP addresses
red.namespace.svc.cluster.local	172.17.0.1

a new entry is added to the DNS

Pod name	Status	Node	podIP
Pod 1	RUNNING	worker1	10.0.0.1
Pod 2	RUNNING	worker2	10.0.1.1

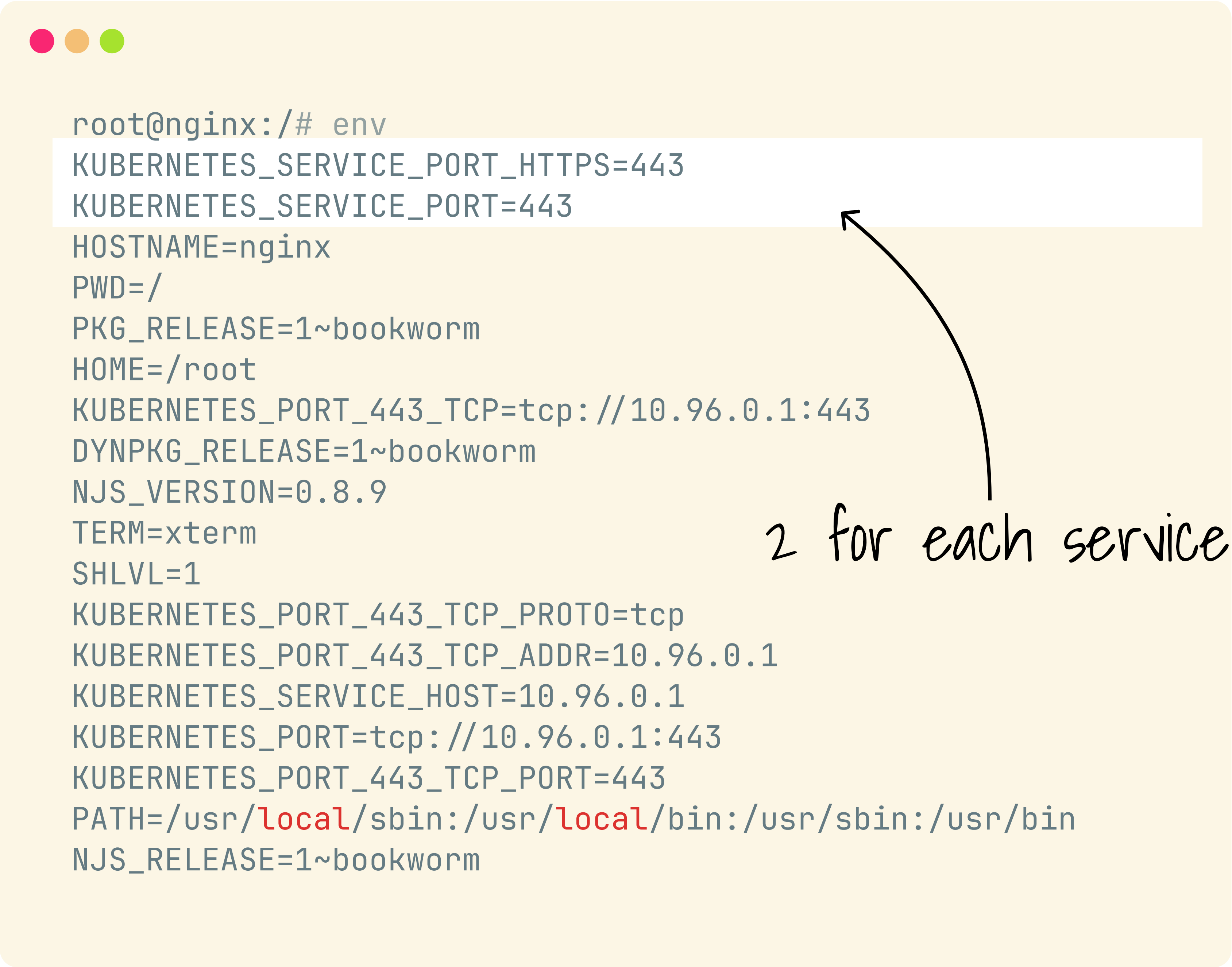
Service name	IP address	Endpoints
Red	172.17.0.1	10.0.0.1:3000,10.0.1.1:3000



Services at scale

Services at scale

if you have 5,000 services in your Kubernetes cluster, it takes 11 minutes to add a new rule with iptables

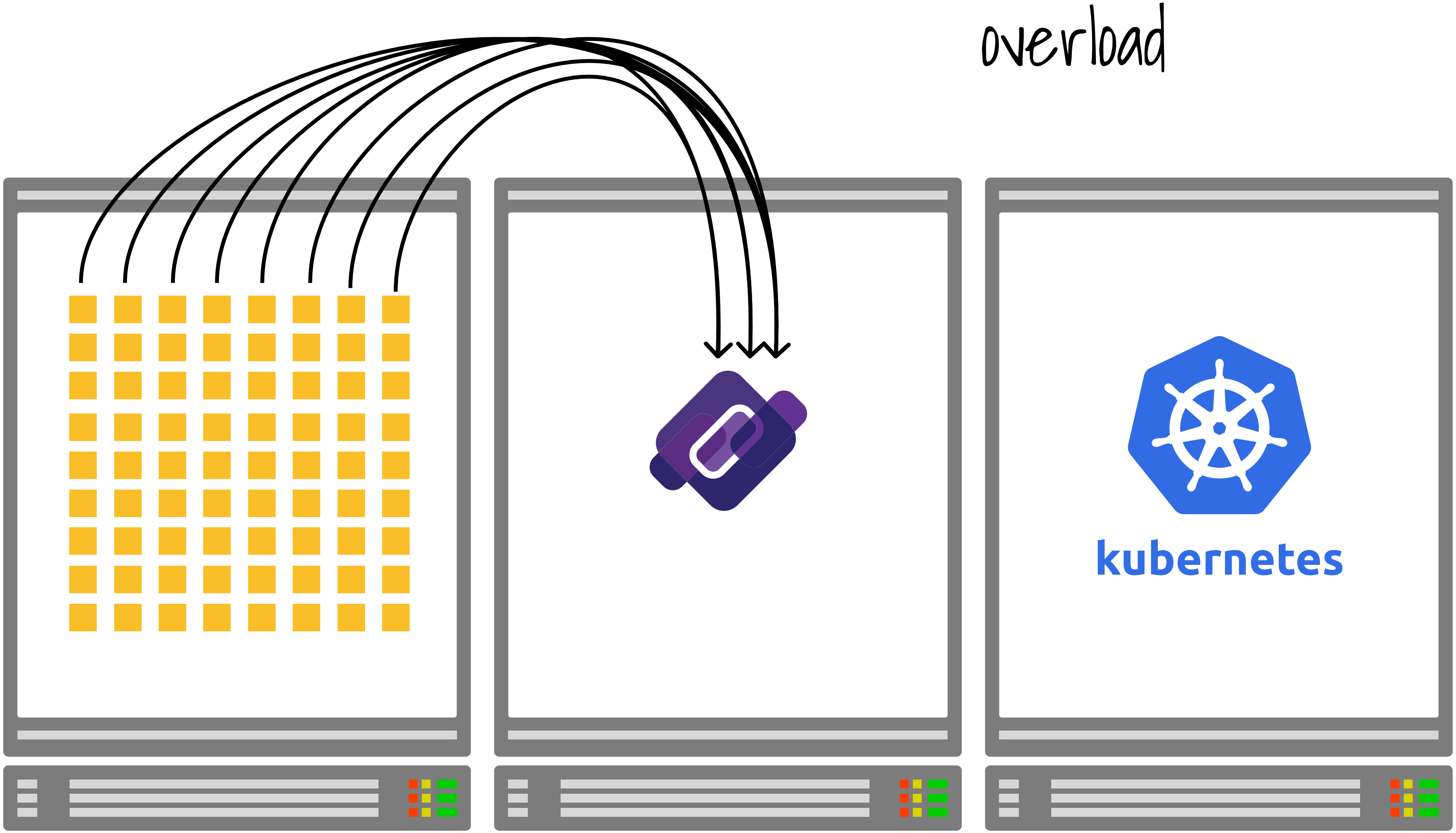


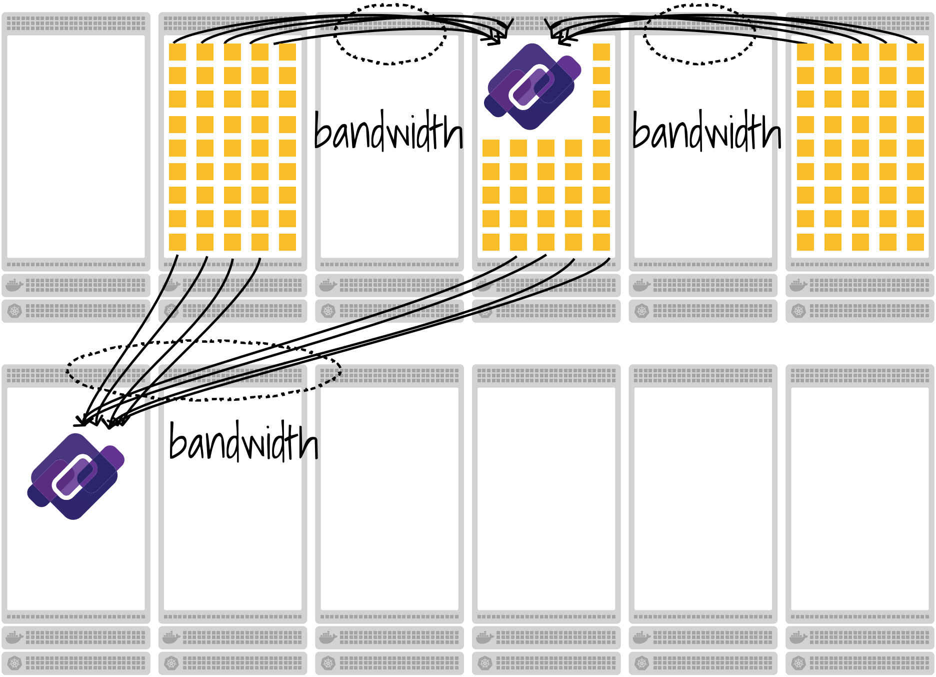
```
root@nginx:/# env
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=nginx
PWD=/
PKG_RELEASE=1~bookworm
HOME=/root
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
DYNPKG_RELEASE=1~bookworm
NJS_VERSION=0.8.9
TERM=xterm
SHLVL=1
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PORT=443
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
NJS_RELEASE=1~bookworm
```

2 for each service

CoreDNS at scale

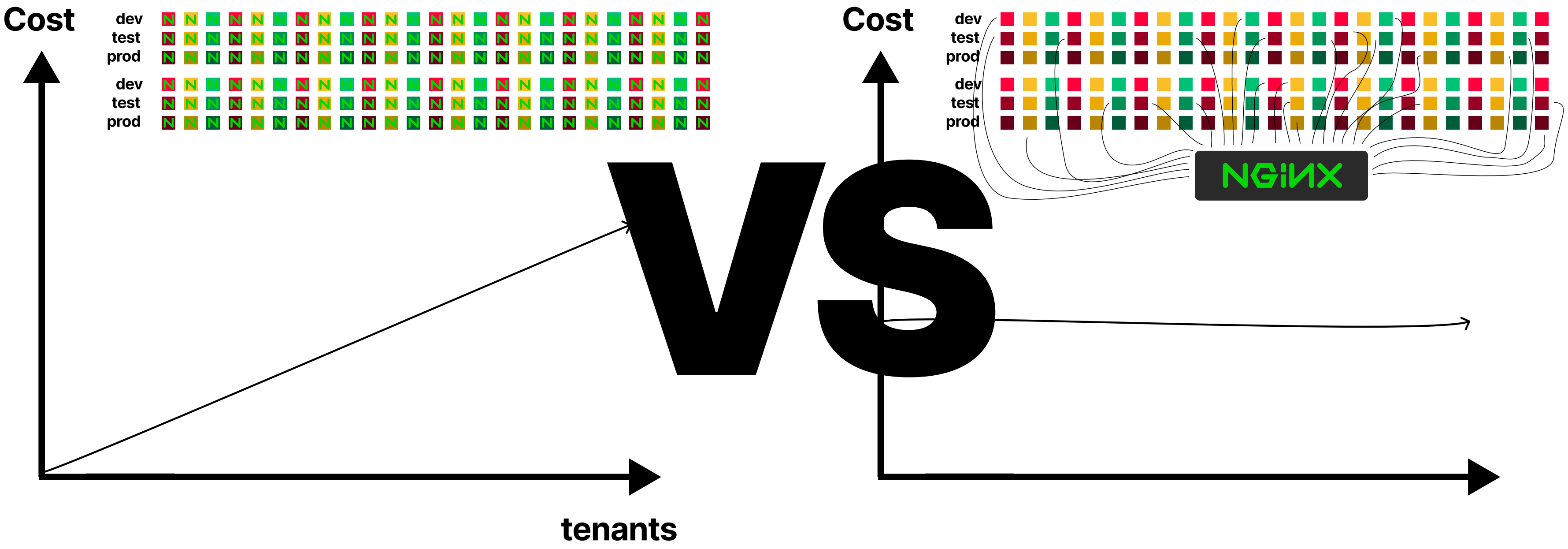
overload





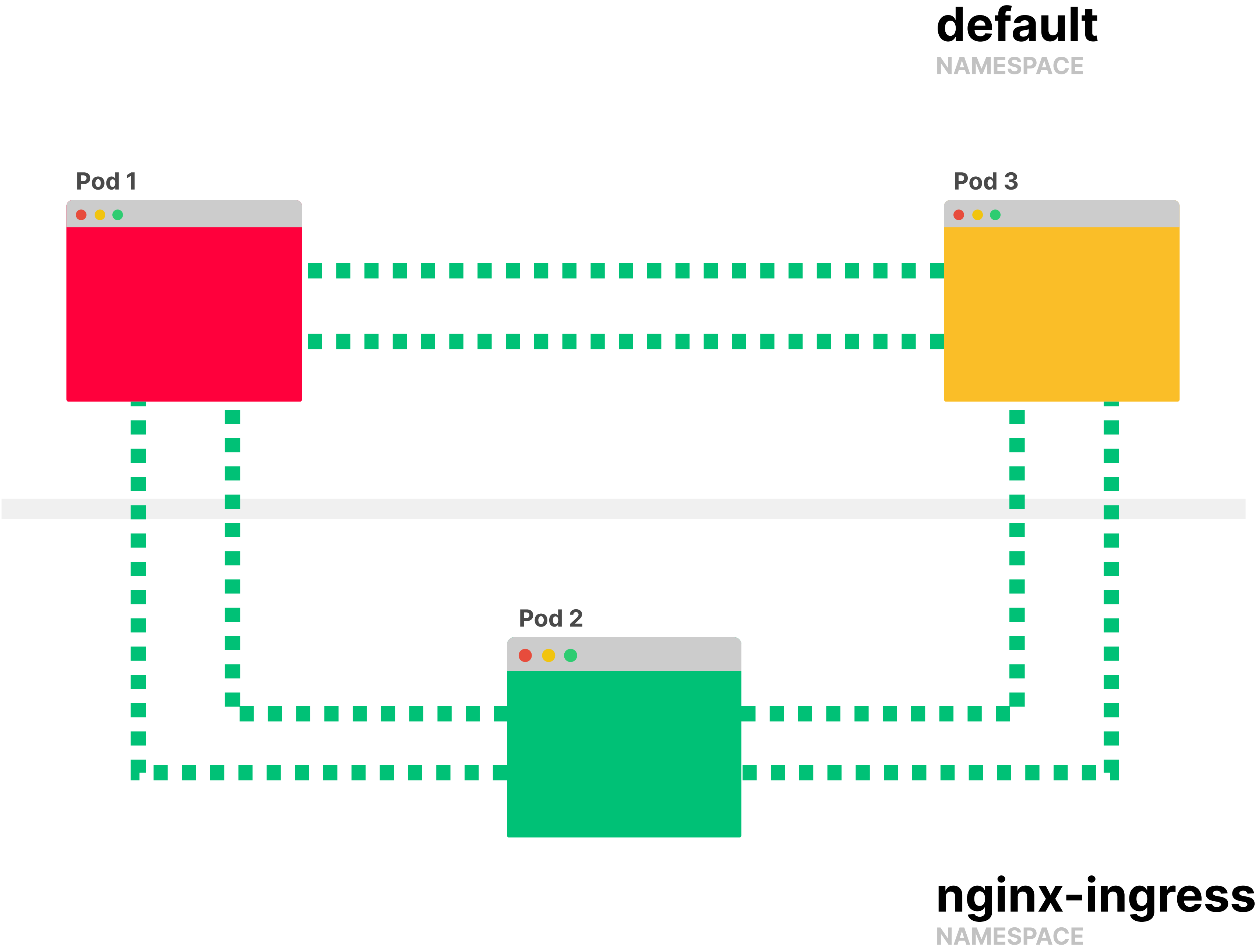
Your options

Sharing resources

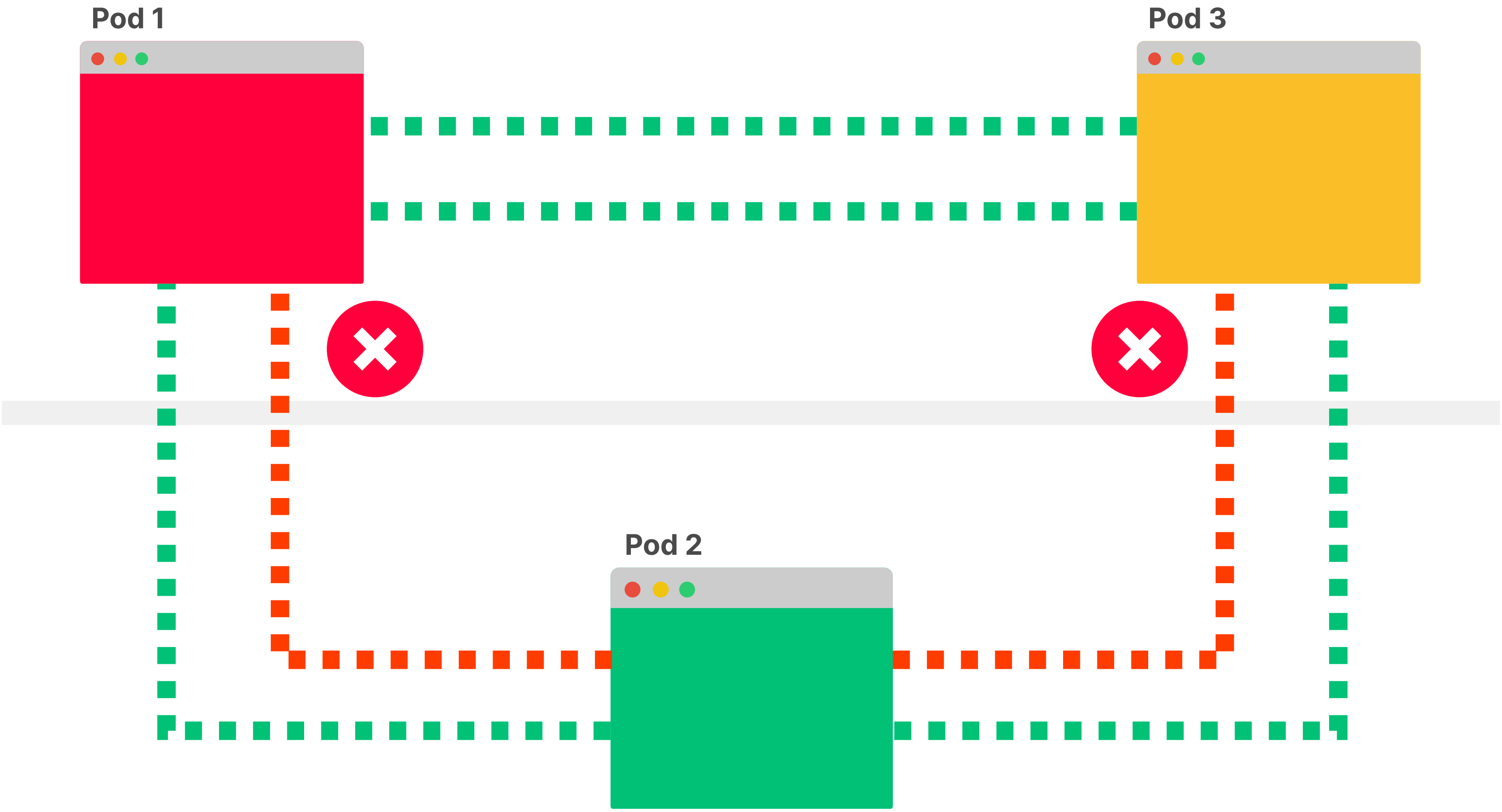

Cost
dev
test
prod
dev
test
prod
tenants
VS
Cost
dev
test
prod
dev
test
prod
NGINX

Your options

Network isolation


default
NAMESPACE
Pod 1
Pod 3
Pod 2
nginx-ingress
NAMESPACE

default
NAMESPACE



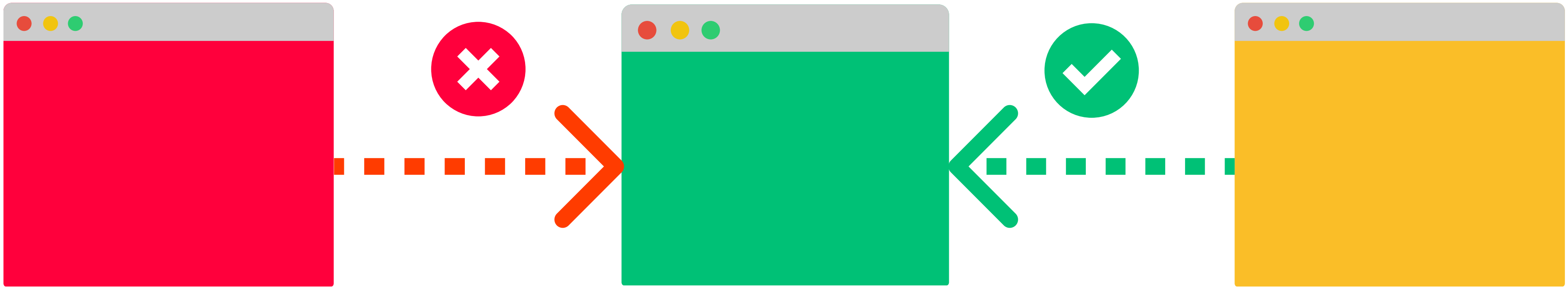
Pod 1

Pod 3

Pod 2

nginx-ingress
NAMESPACE

Example



Pod 1

LABELS

app: auth
role: api

Pod 2

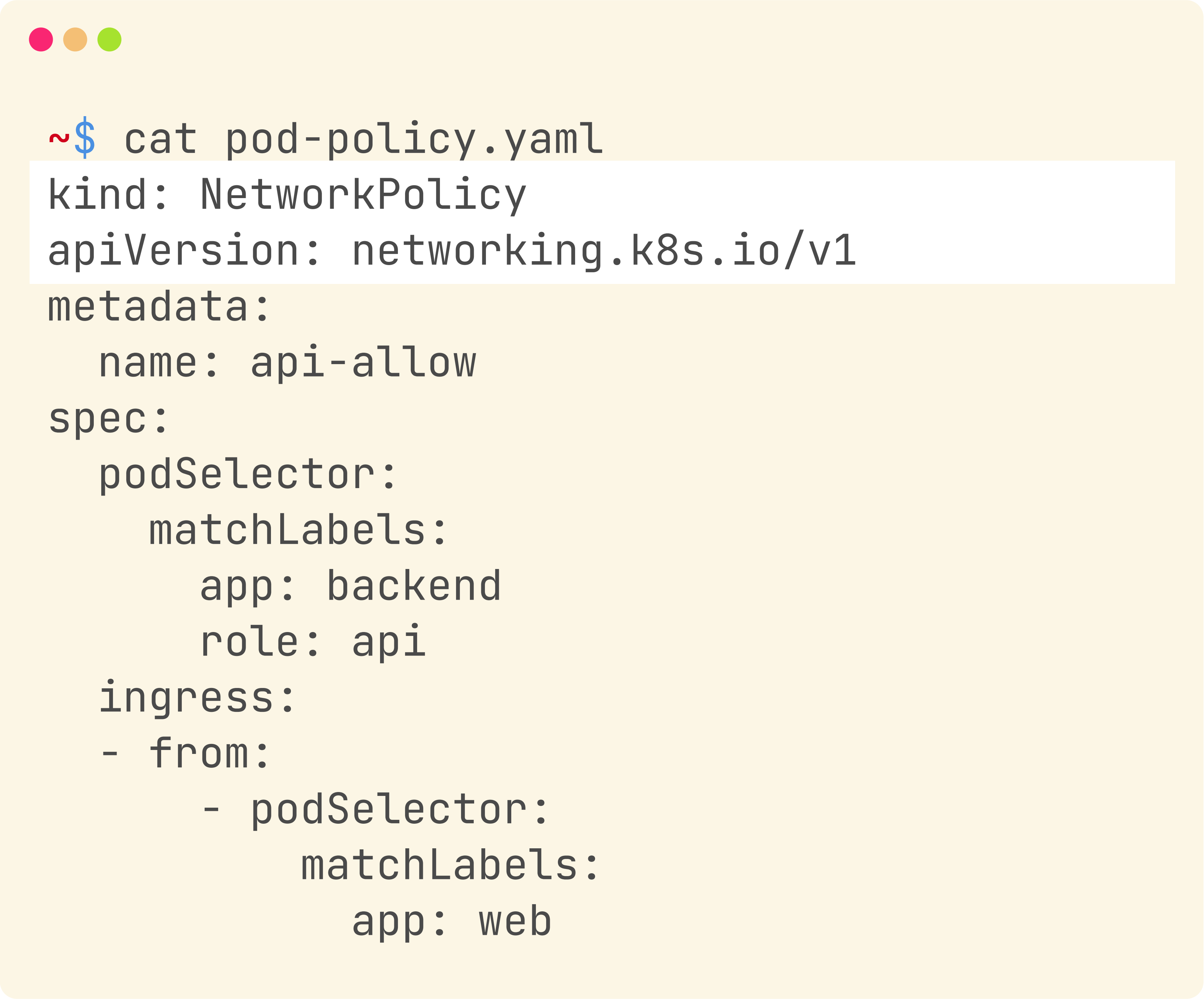
LABELS

app: backend
role: api

Pod 3

LABELS

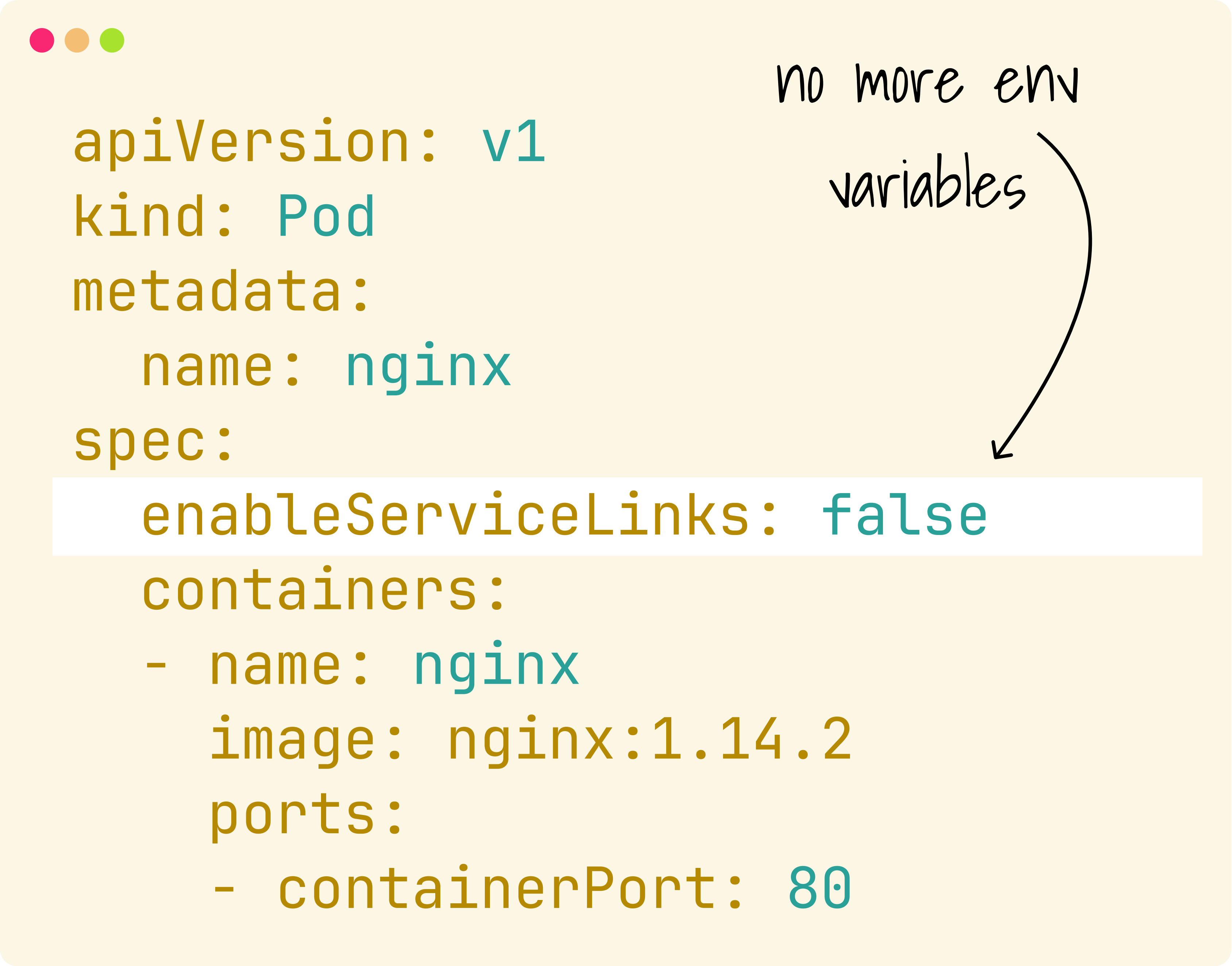
app: web
role: frontend



```
~$ cat pod-policy.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: api-allow
spec:
  podSelector:
    matchLabels:
      app: backend
      role: api
  ingress:
  - from:
      - podSelector:
          matchLabels:
            app: web
```

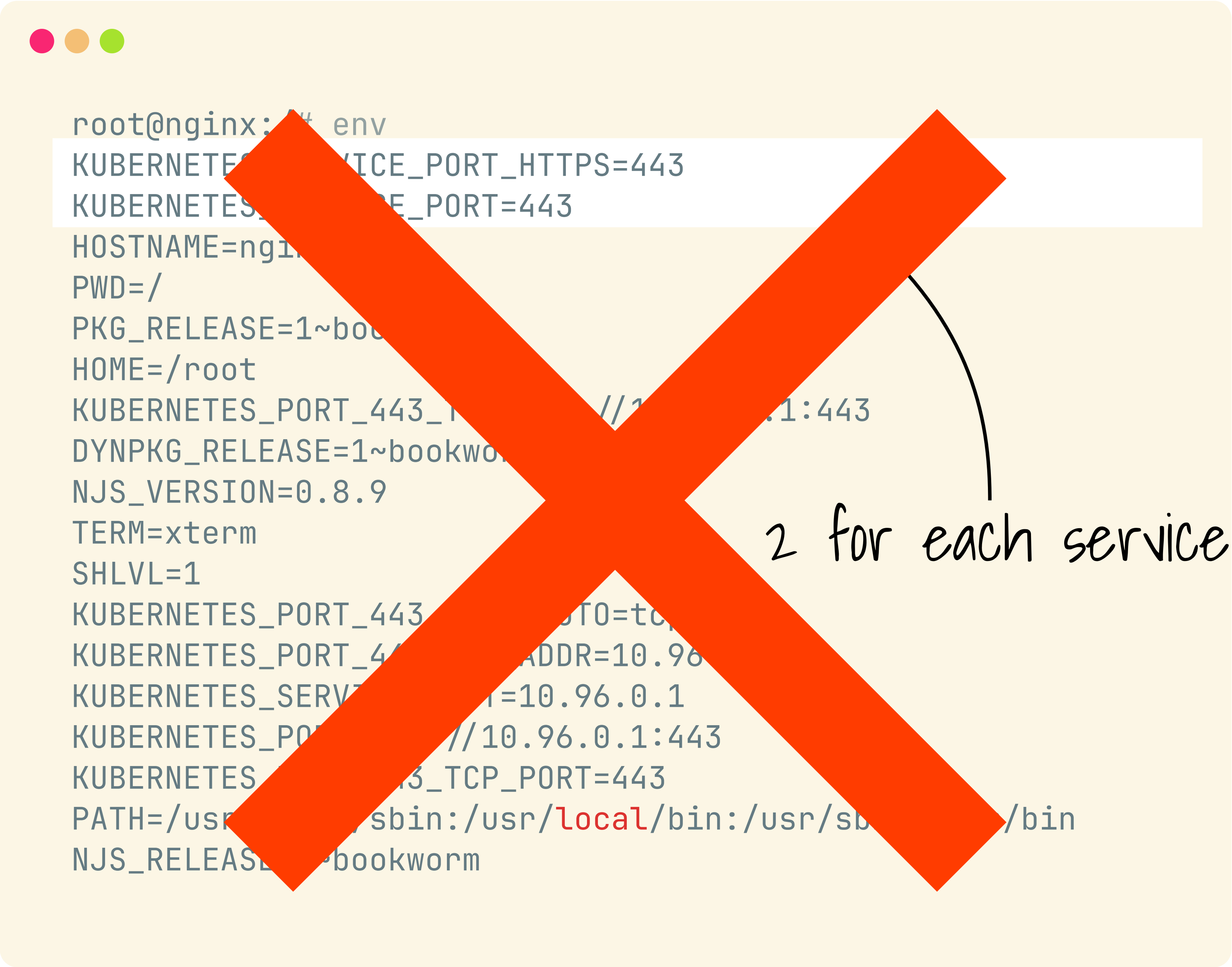
Your options

Services at scale



no more env variables

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  enableServiceLinks: false
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```



```
root@nginx:/# env
KUBERNETE      ICE_PORT_HTTPS=443
KUBERNETES      E_PORT=443
HOSTNAME=ngi
PWD=/
PKG_RELEASE=1~boo
HOME=/root
KUBERNETES_PORT_443_T        //1        .1:443
DYNPKG_RELEASE=1~bookwo
NJS_VERSION=0.8.9
TERM=xterm
SHLVL=1
KUBERNETES_PORT_443        OTO=tc
KUBERNETES_PORT_4        ADDR=10.96
KUBERNETES_SERV        T=10.96.0.1
KUBERNETES_PO        //10.96.0.1:443
KUBERNETES        3_TCP_PORT=443
PATH=/usr        sbin:/usr/local/bin:/usr/sb        /bin
NJS_RELEAS        bookworm
```

2 for each service

Your options

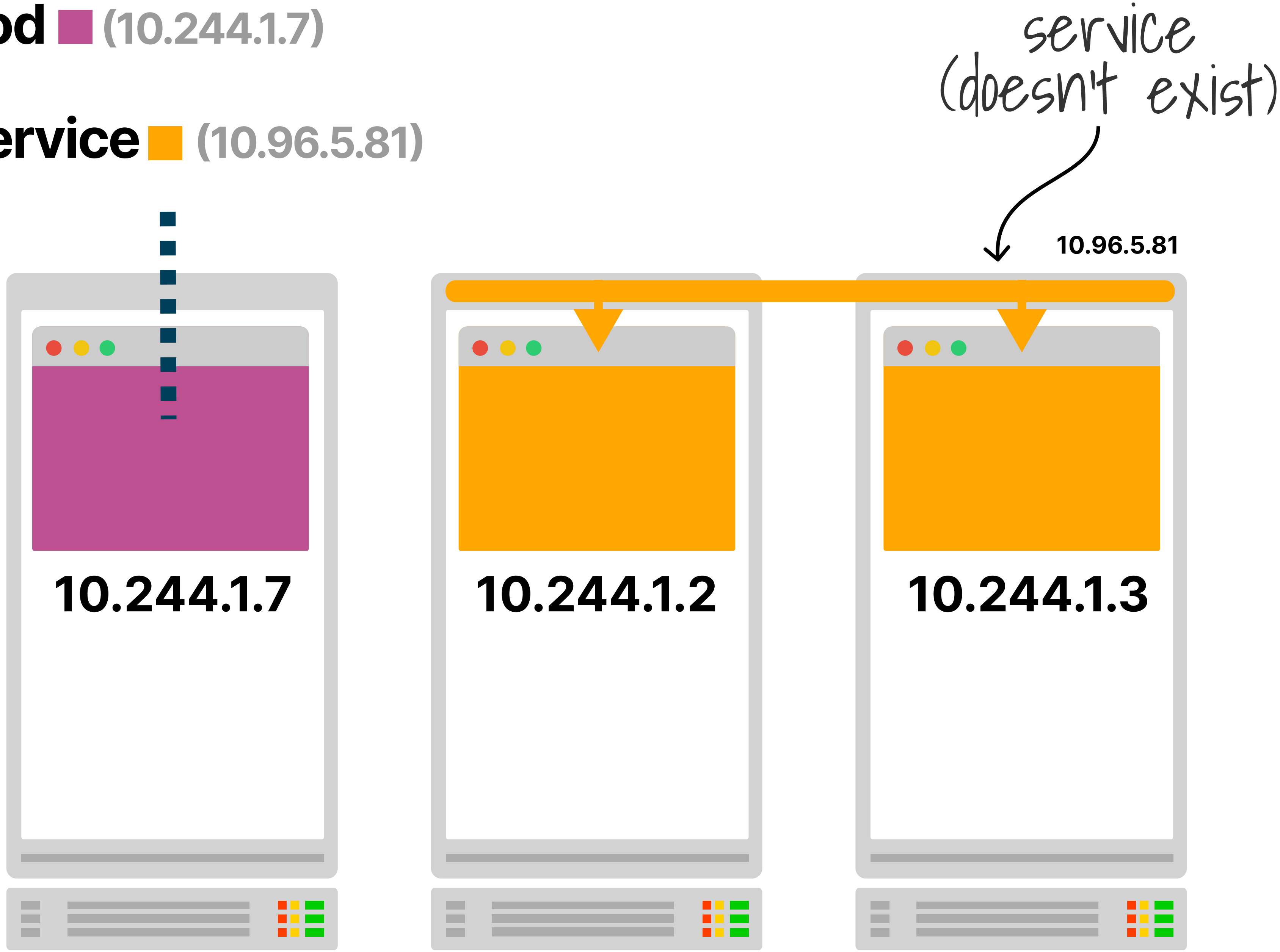
iptables limits

FROM:

Pod 🟪 (10.244.1.7)

TO:

Service 🟧 (10.96.5.81)



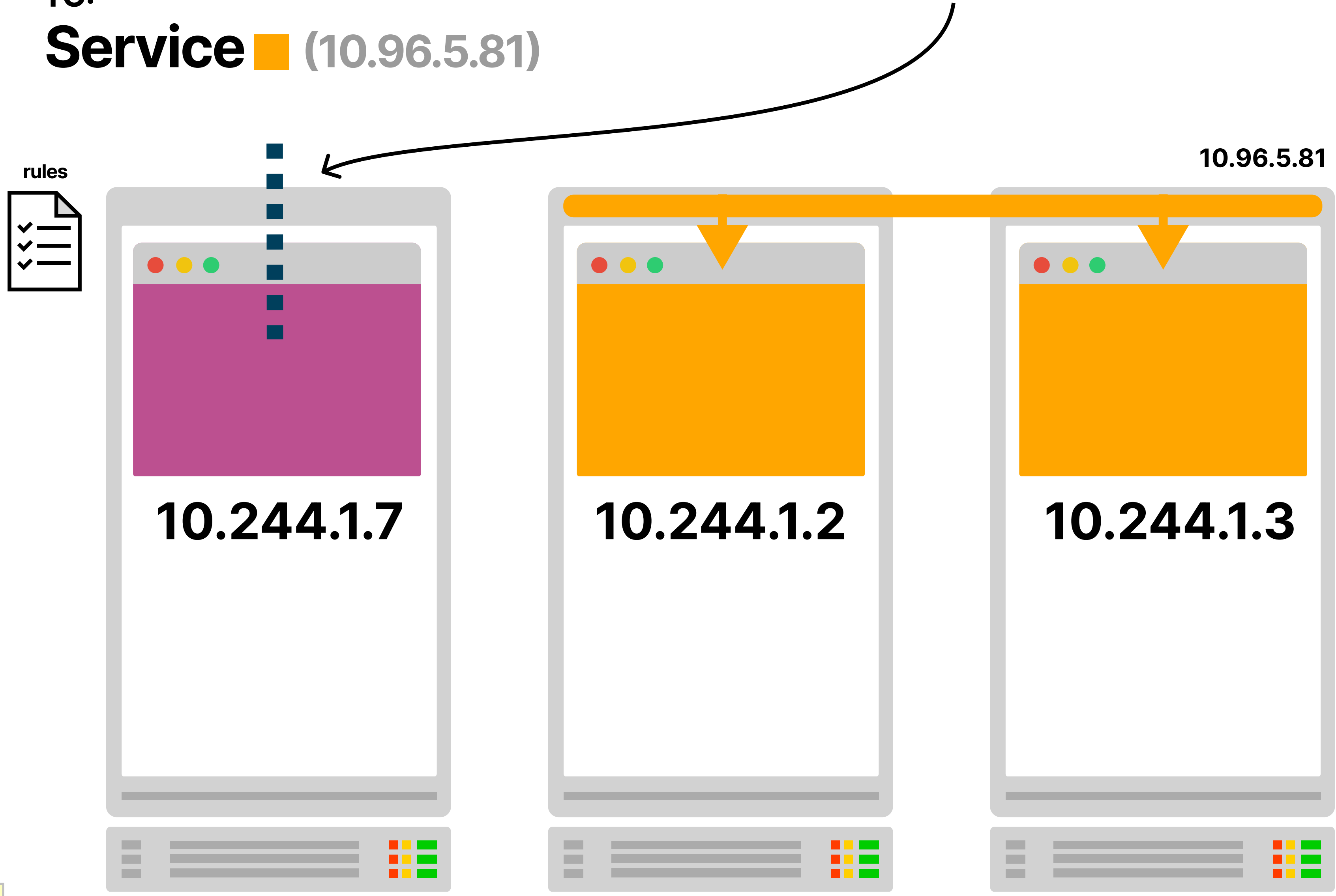
FROM:

Pod 🟪 (10.244.1.7)

TO:

Service 🟧 (10.96.5.81)

the traffic is intercepted



FROM:

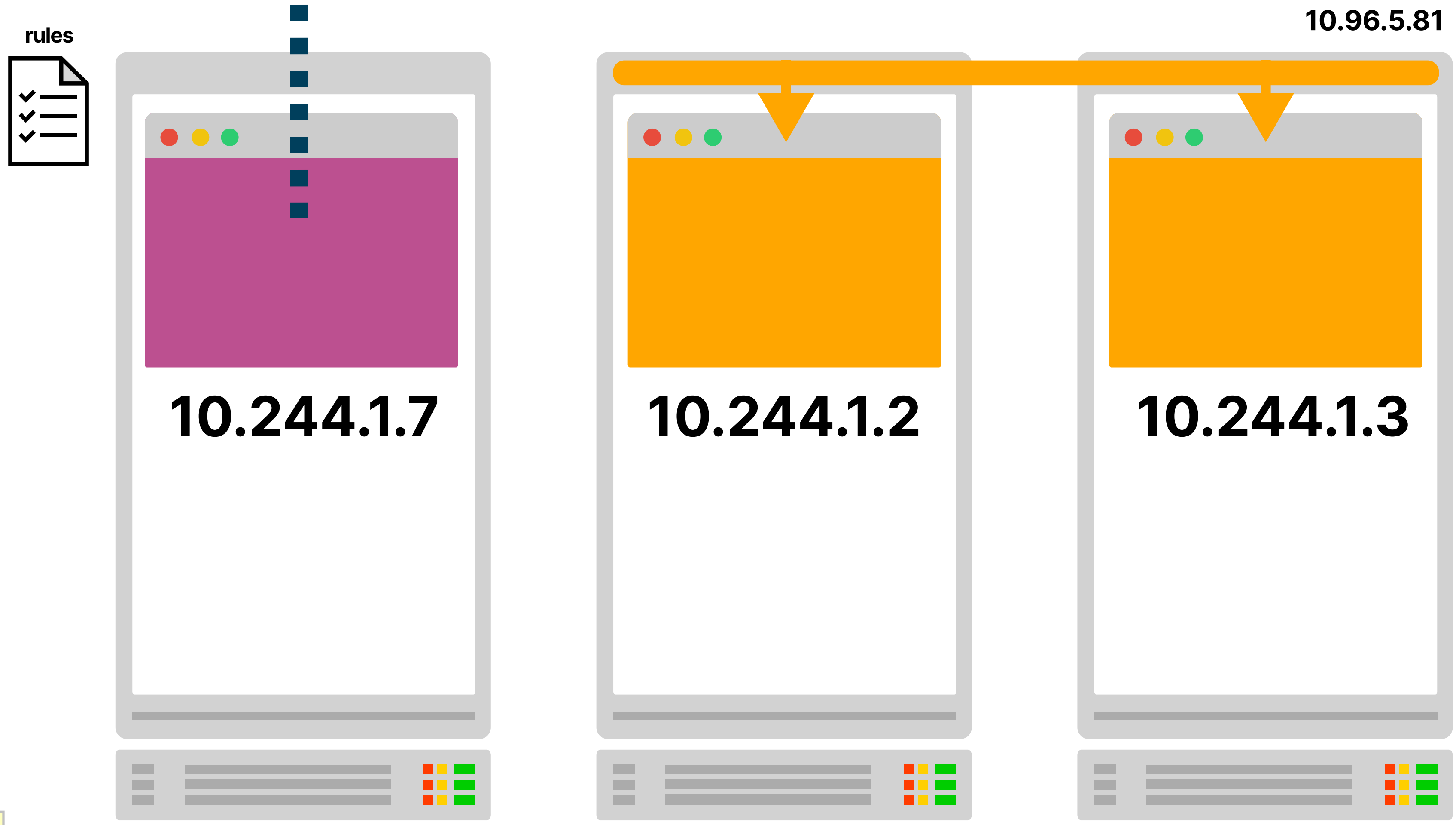
Pod 🟪 (10.244.1.7)

TO:

~~**Service** 🟧 (10.96.5.81)~~

Pod 🟧 (10.244.1.3)

the traffic is rewritten (DNAT)



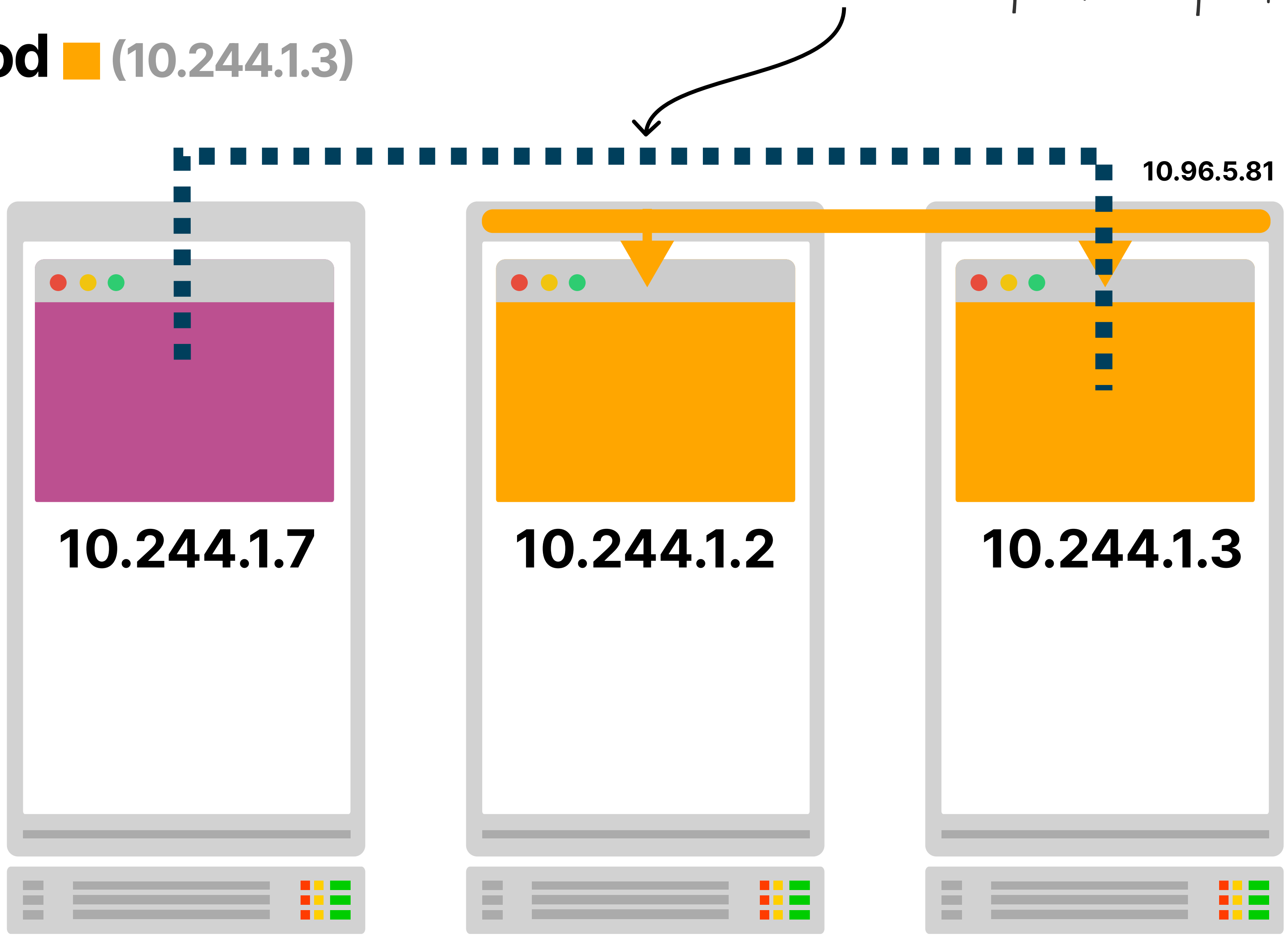
FROM:

Pod 🟪 (10.244.1.7)

TO:

Pod 🟧 (10.244.1.3)

the traffic is now pod to pod



FROM:

Pod 🟪 (10.244.1.7)

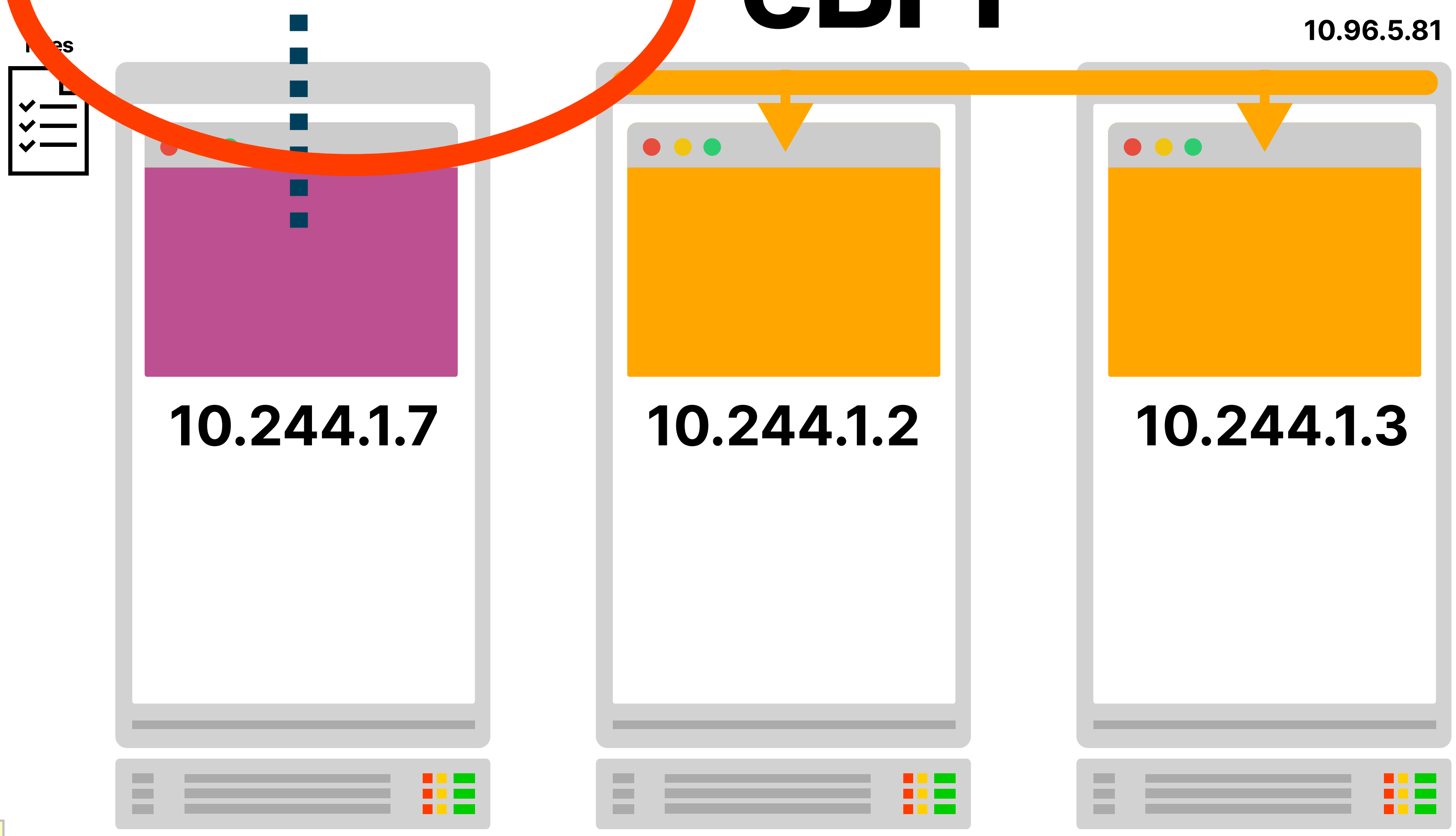
TO:

~~**Service** 🟧 (10.96.5.81)~~

Pod 🟧 (10.244.1.3)

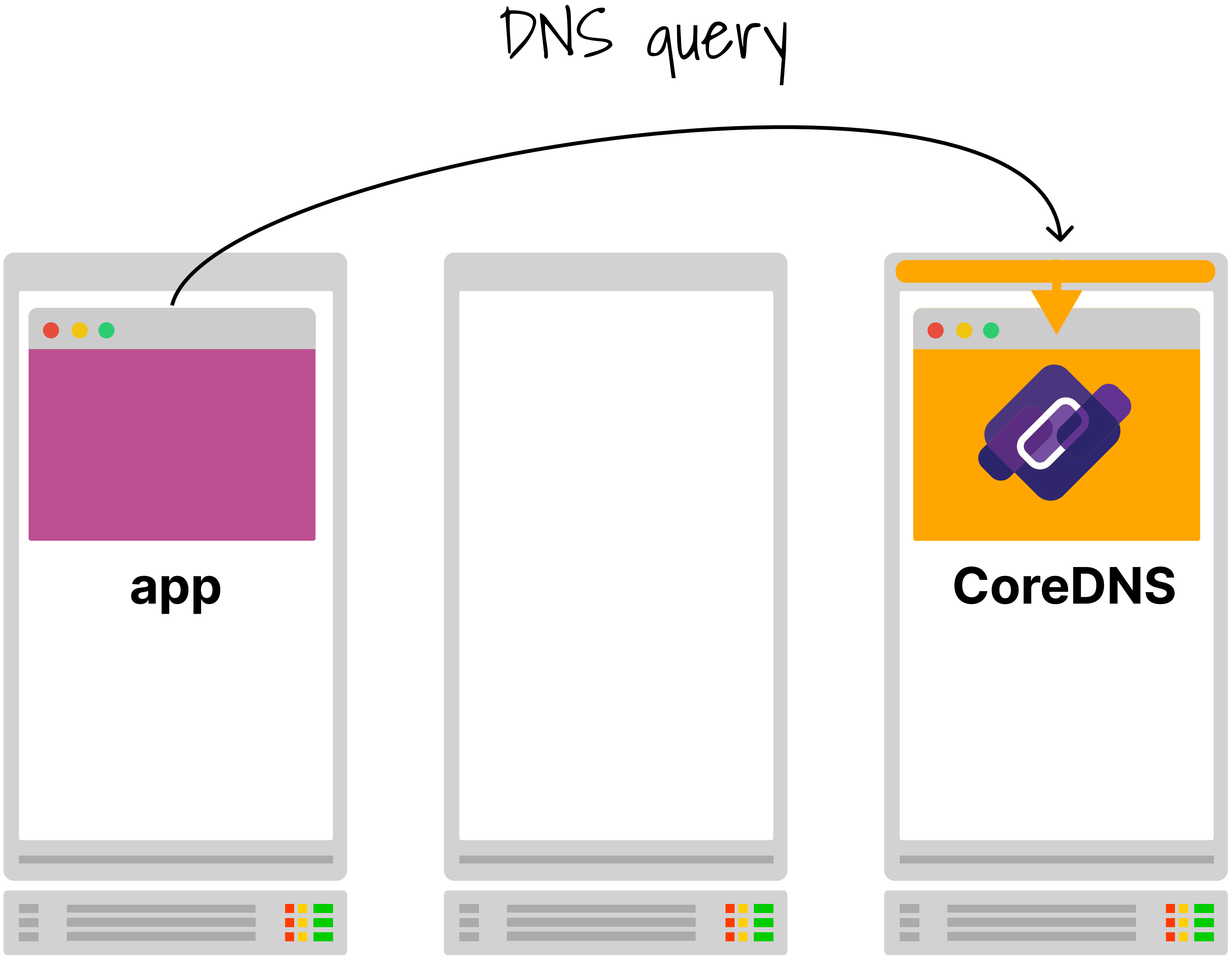
the traffic is rewritten (DNAT)

eBPF



Your options

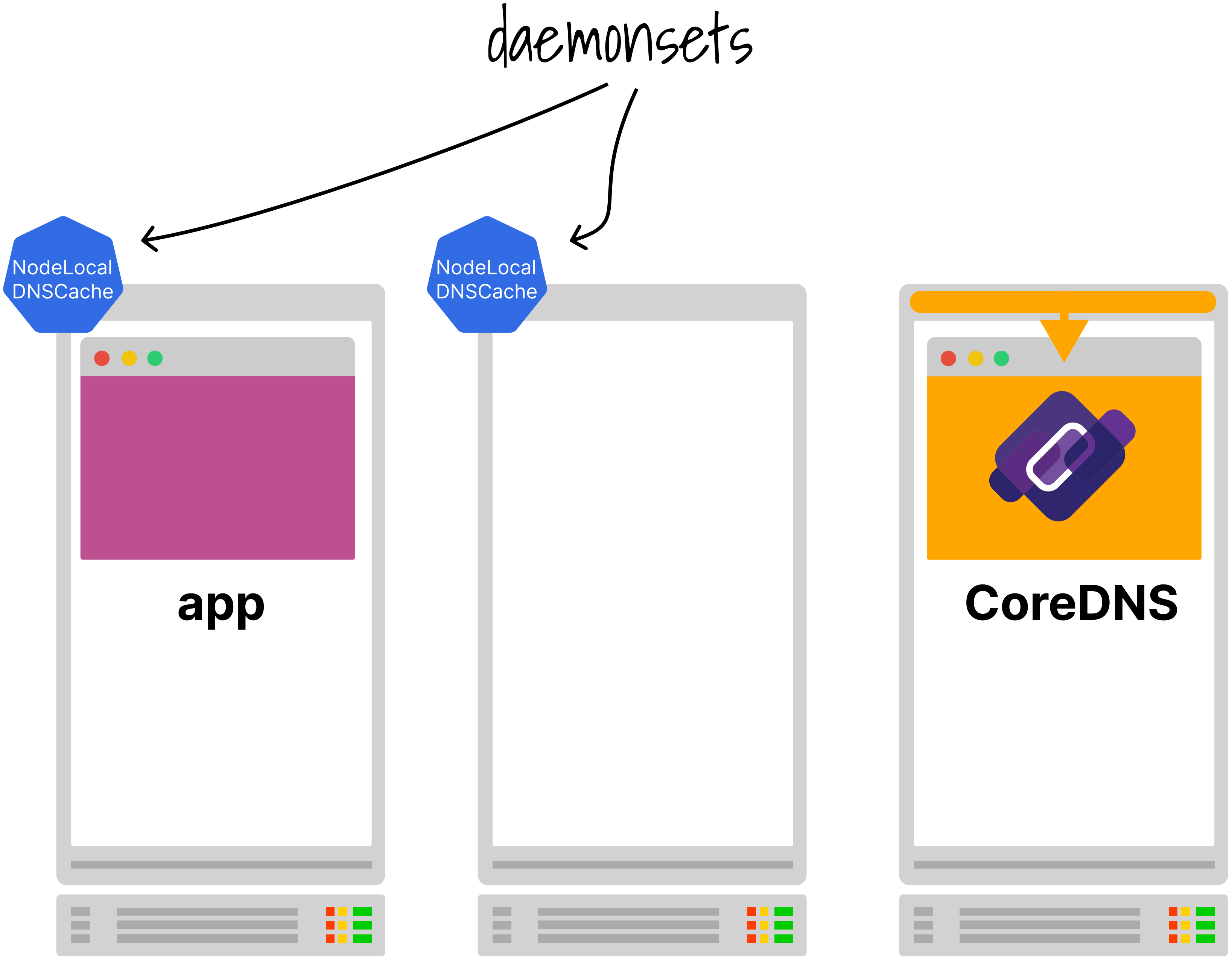
DNS scaling

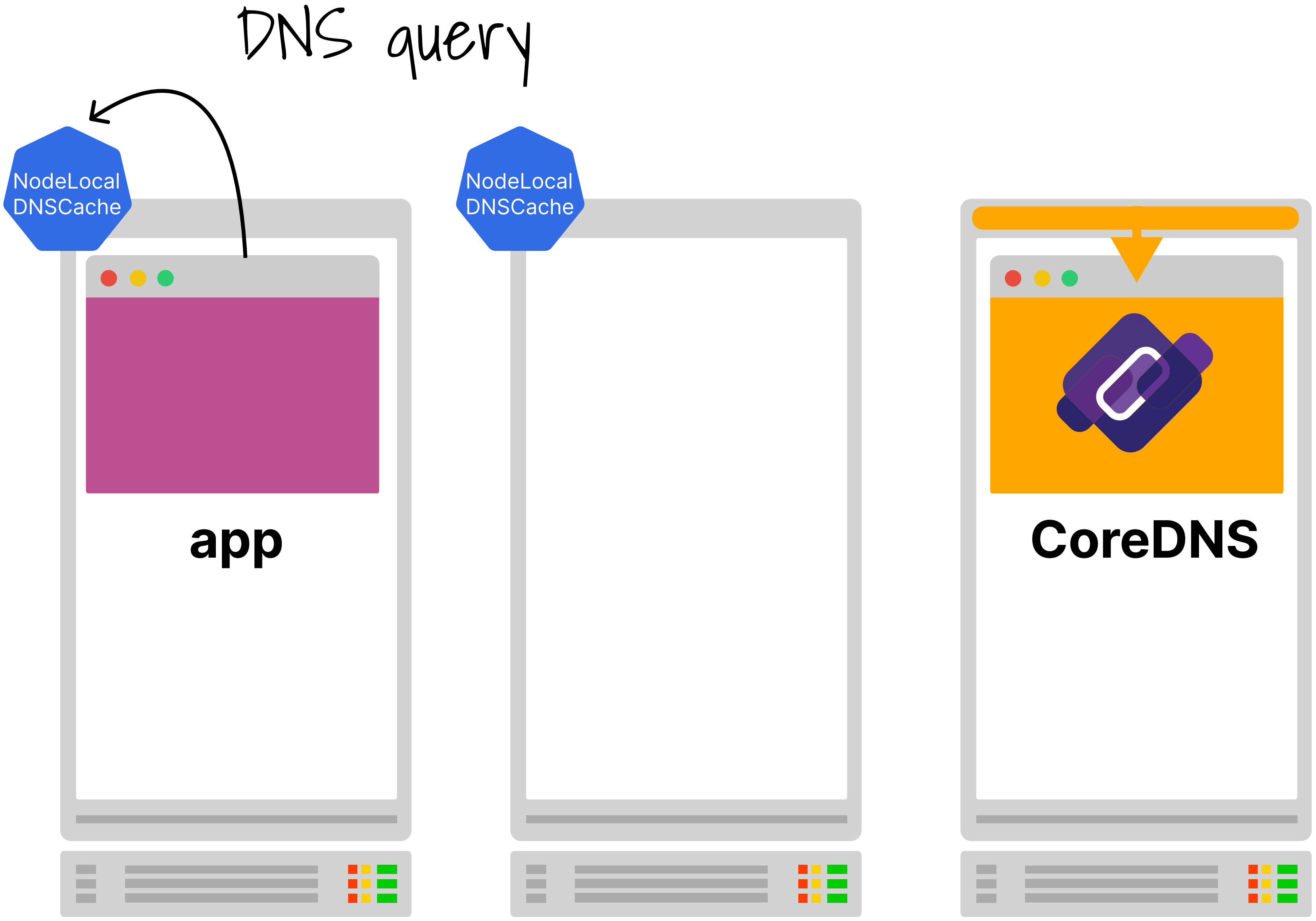


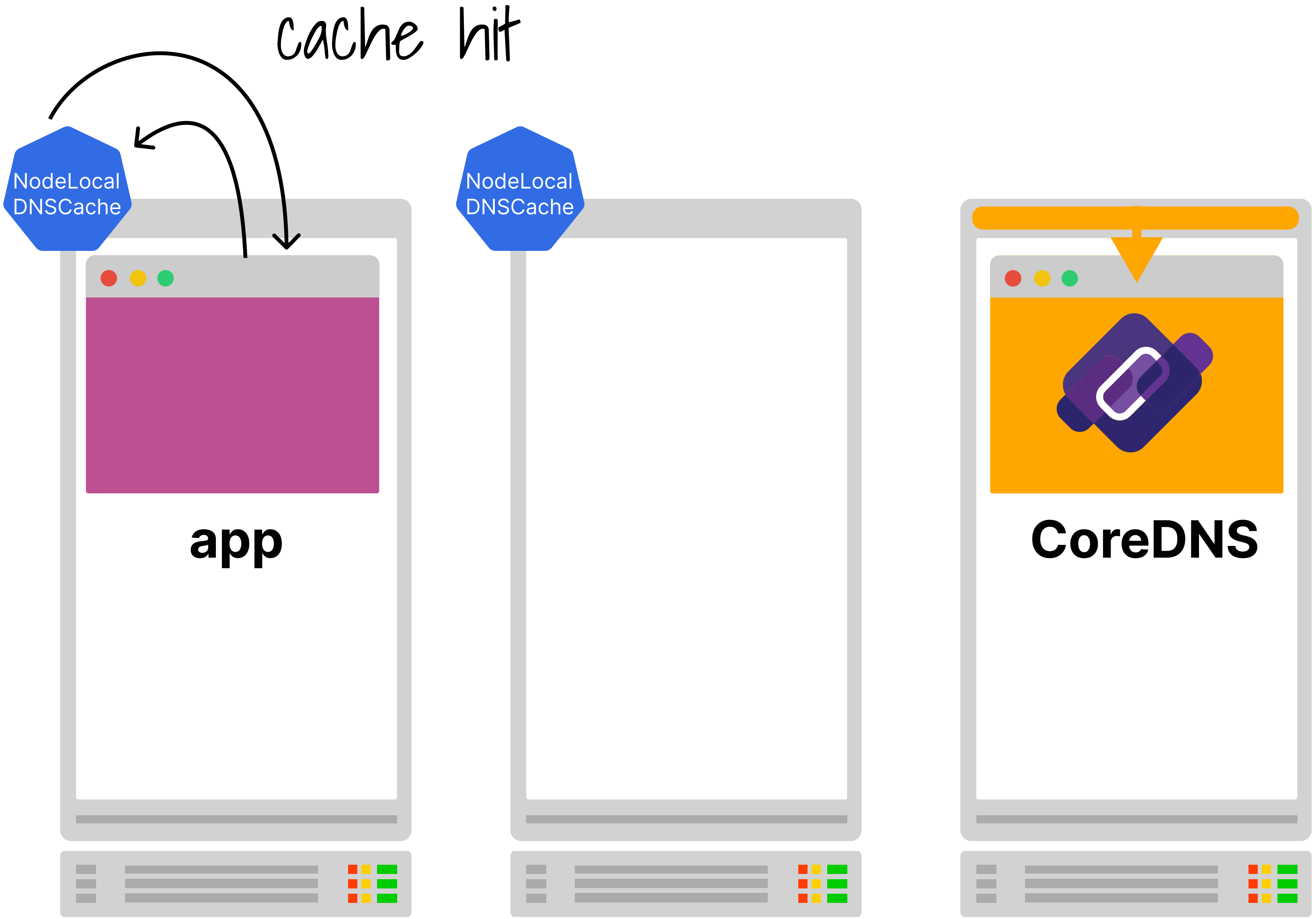
DNS query

app

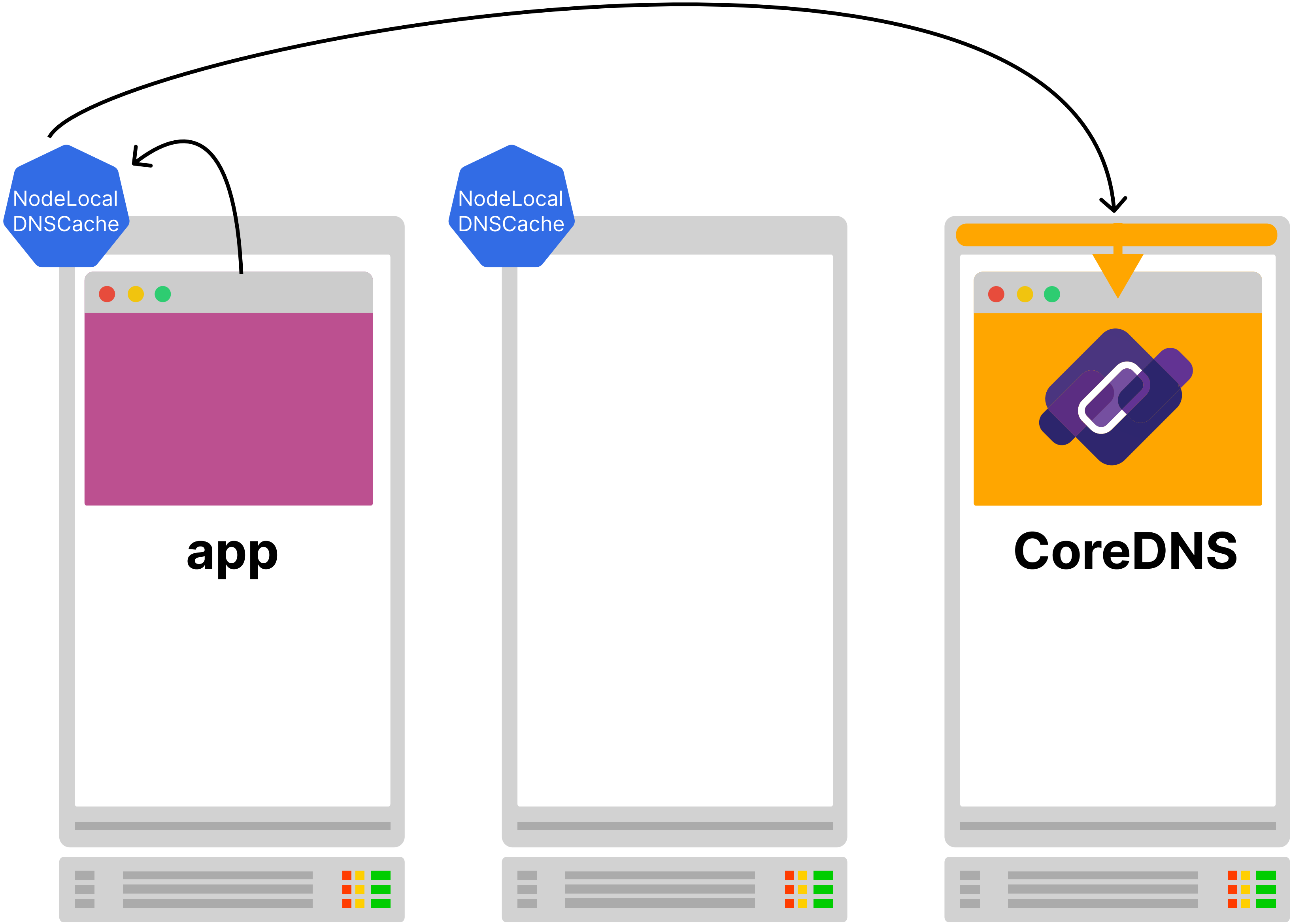
CoreDNS


daemonsets
NodeLocal DNSCache
NodeLocal DNSCache
app
CoreDNS


DNS query
NodeLocal DNSCache
NodeLocal DNSCache
app
CoreDNS


cache hit
NodeLocal DNSCache
NodeLocal DNSCache
app
CoreDNS

cache miss



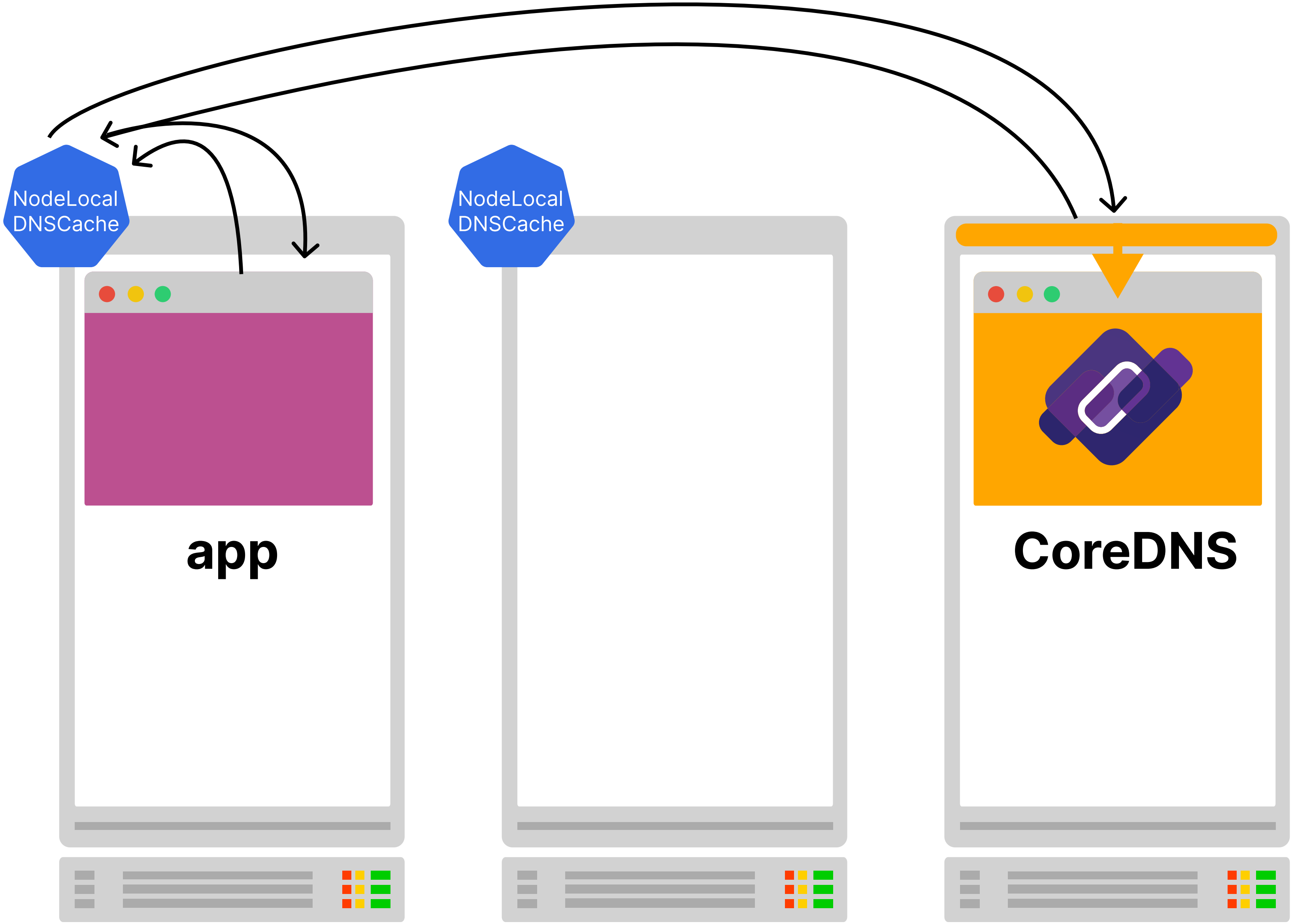
NodeLocal DNSCache

NodeLocal DNSCache

app

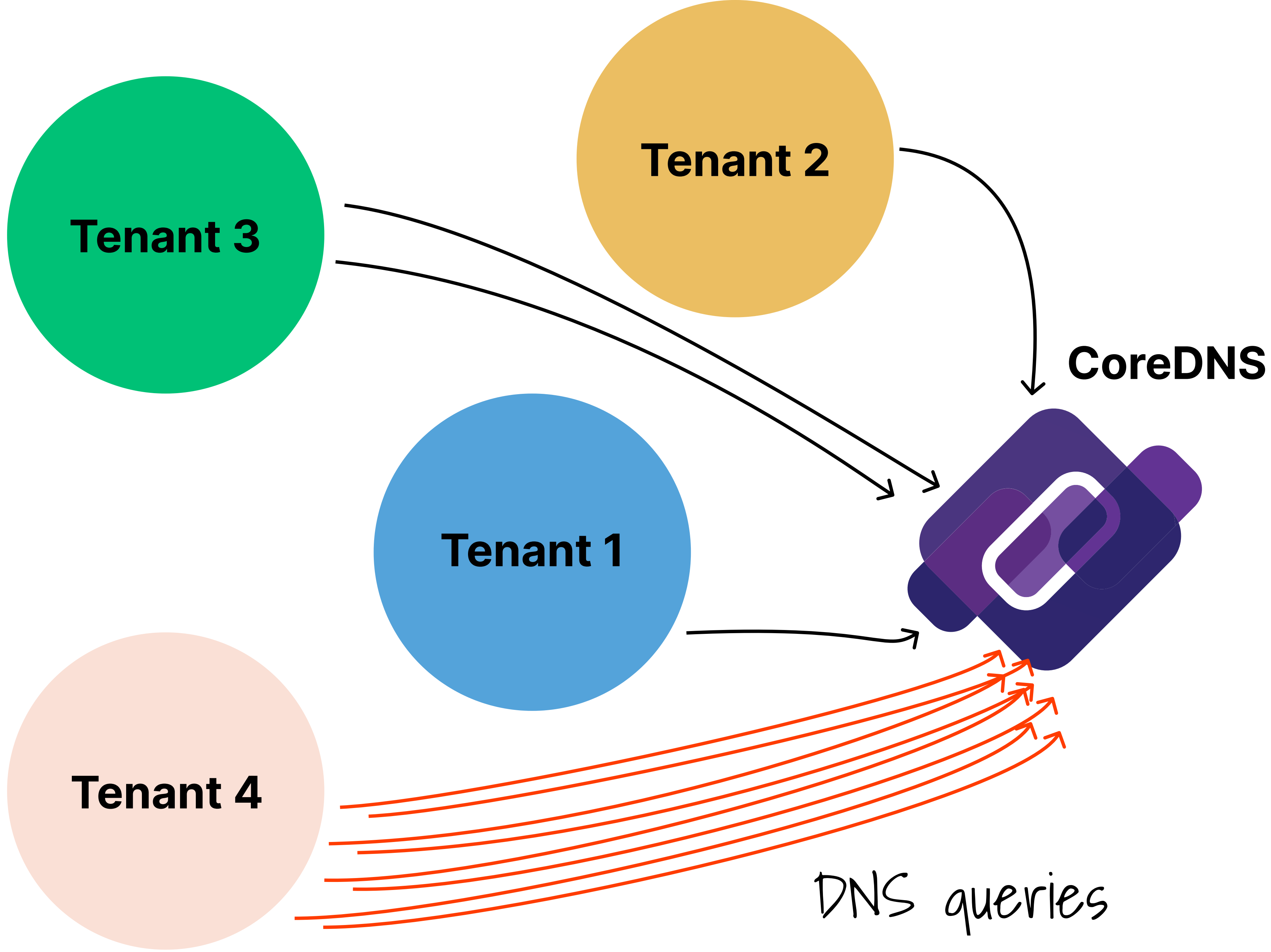
CoreDNS

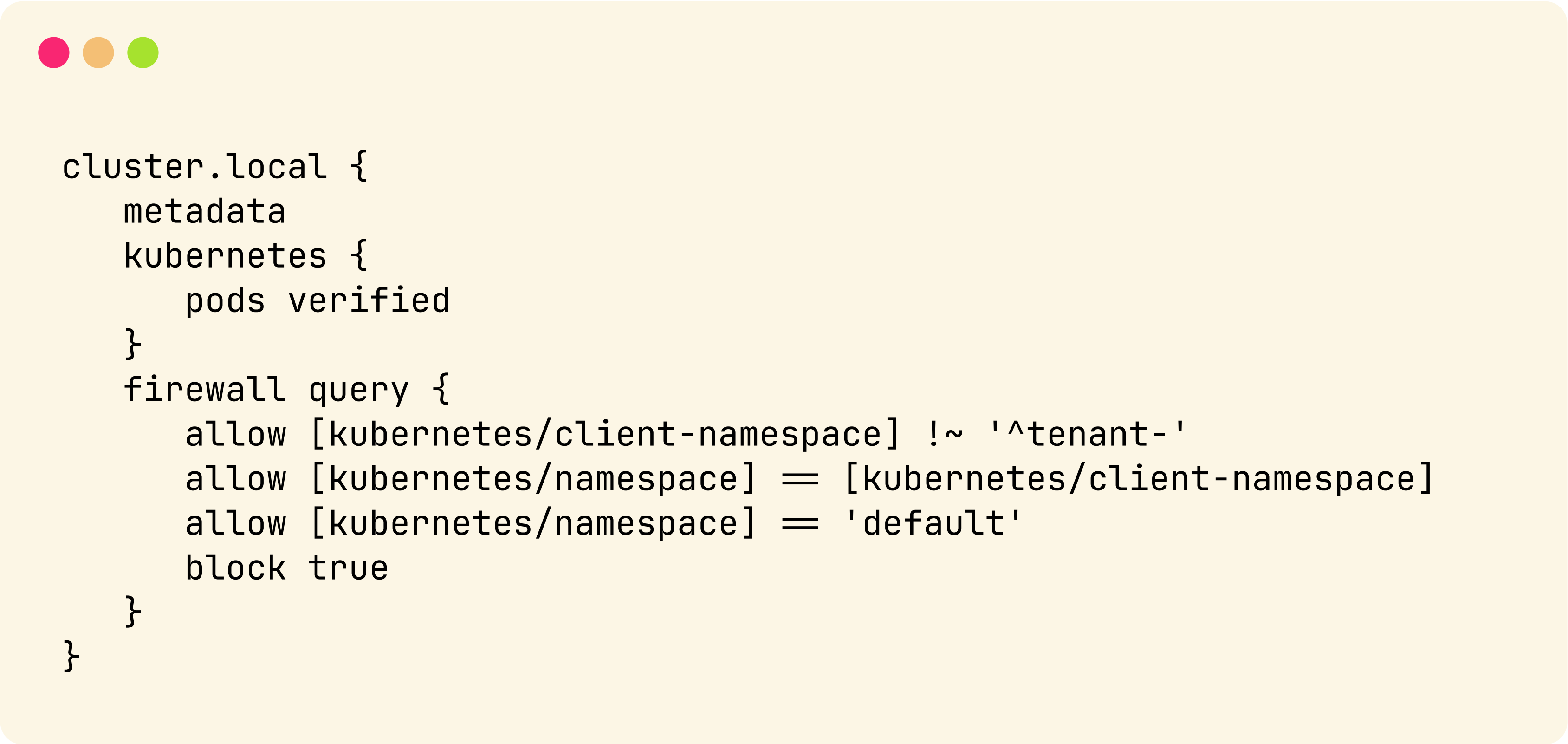
cache miss



Your options

DNS abuse


Tenant 2
Tenant 3
CoreDNS
Tenant 1
Tenant 4
DNS queries

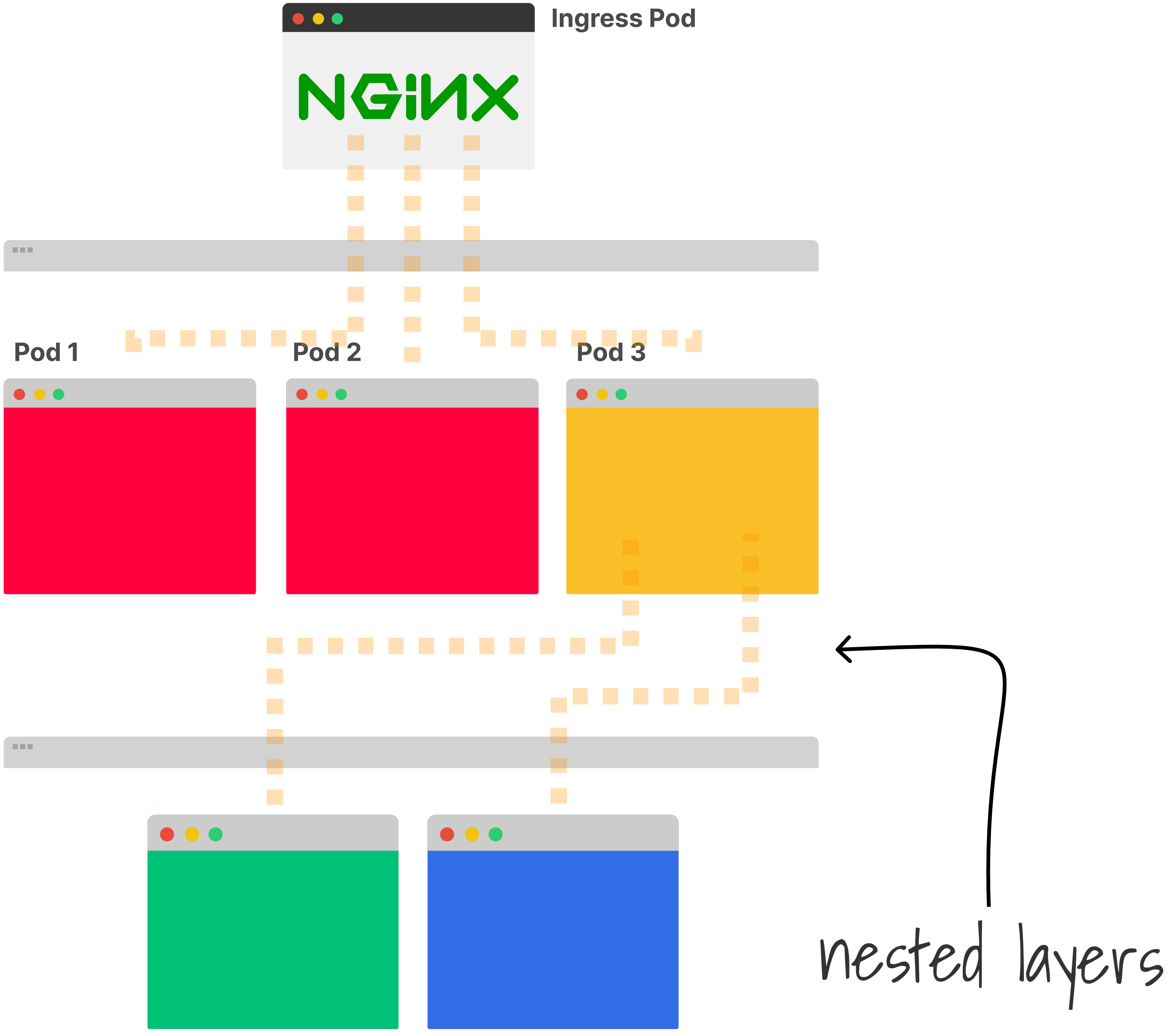


```
cluster.local {
   metadata
   kubernetes {
      pods verified
   }
   firewall query {
      allow [kubernetes/client-namespace] !~ '^tenant-'
      allow [kubernetes/namespace] == [kubernetes/client-namespace]
      allow [kubernetes/namespace] == 'default'
      block true
   }
}
```

coredns/policy

Your options

Service meshes



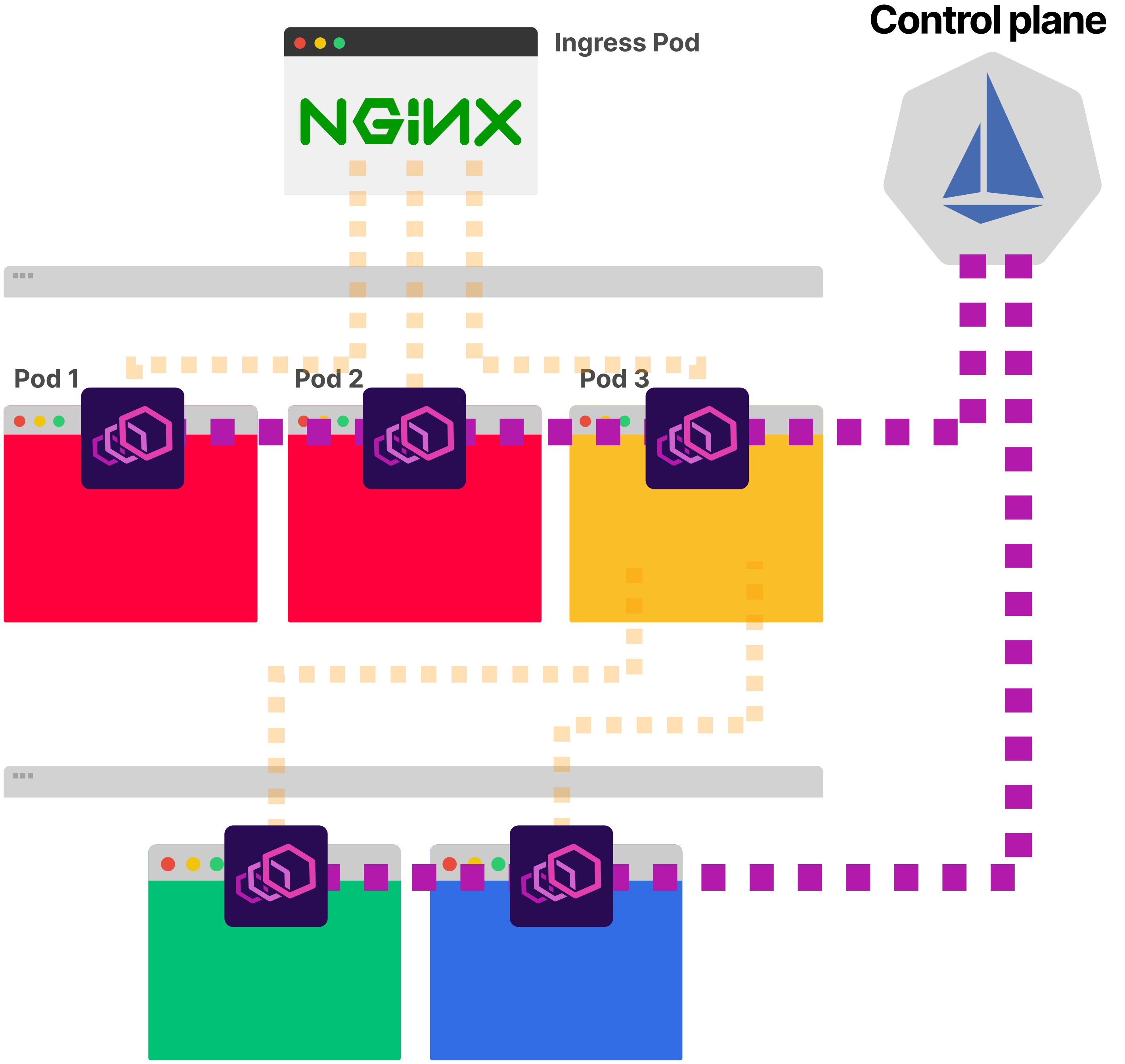
Ingress Pod

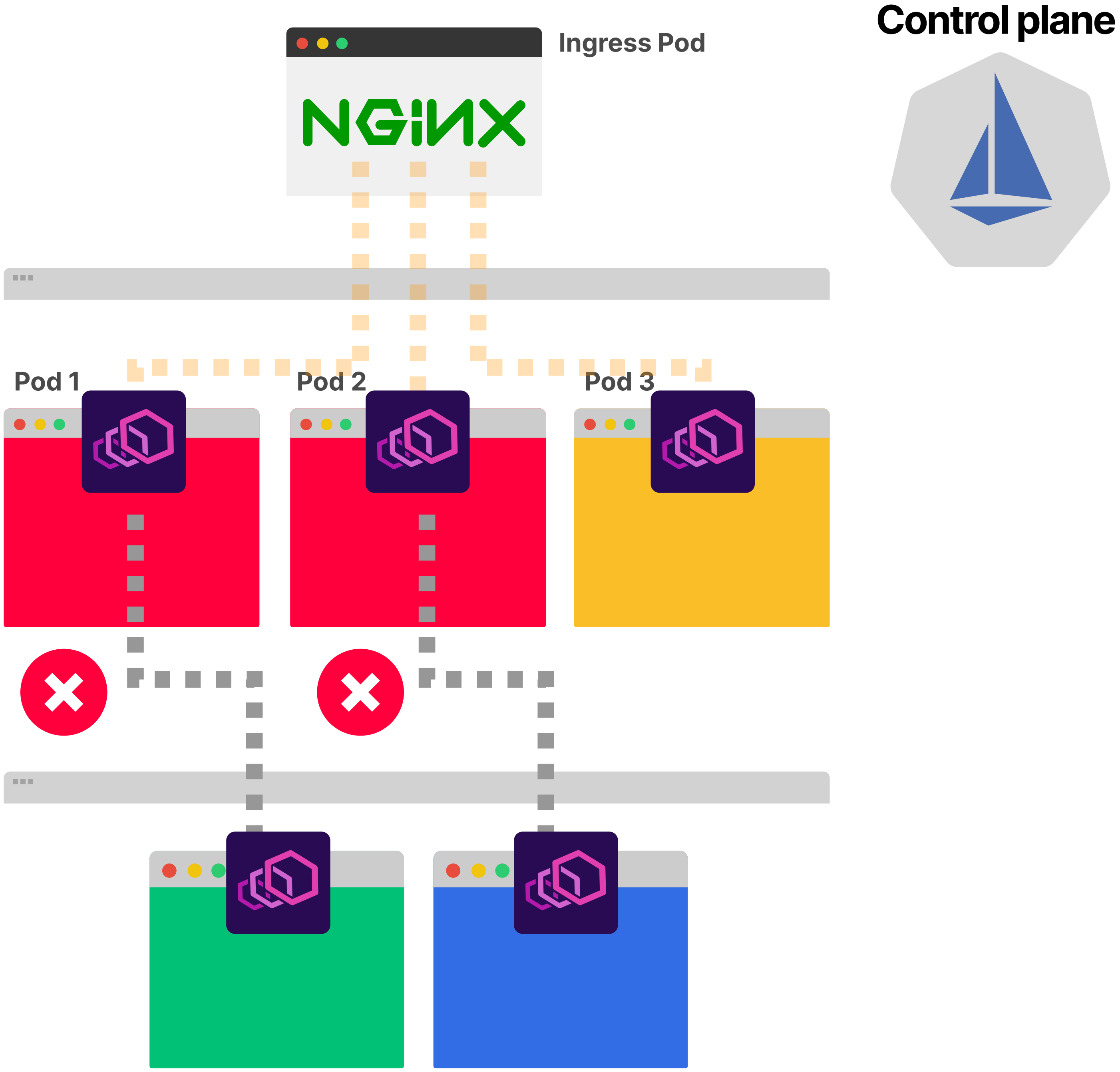
Pod 1

Pod 2

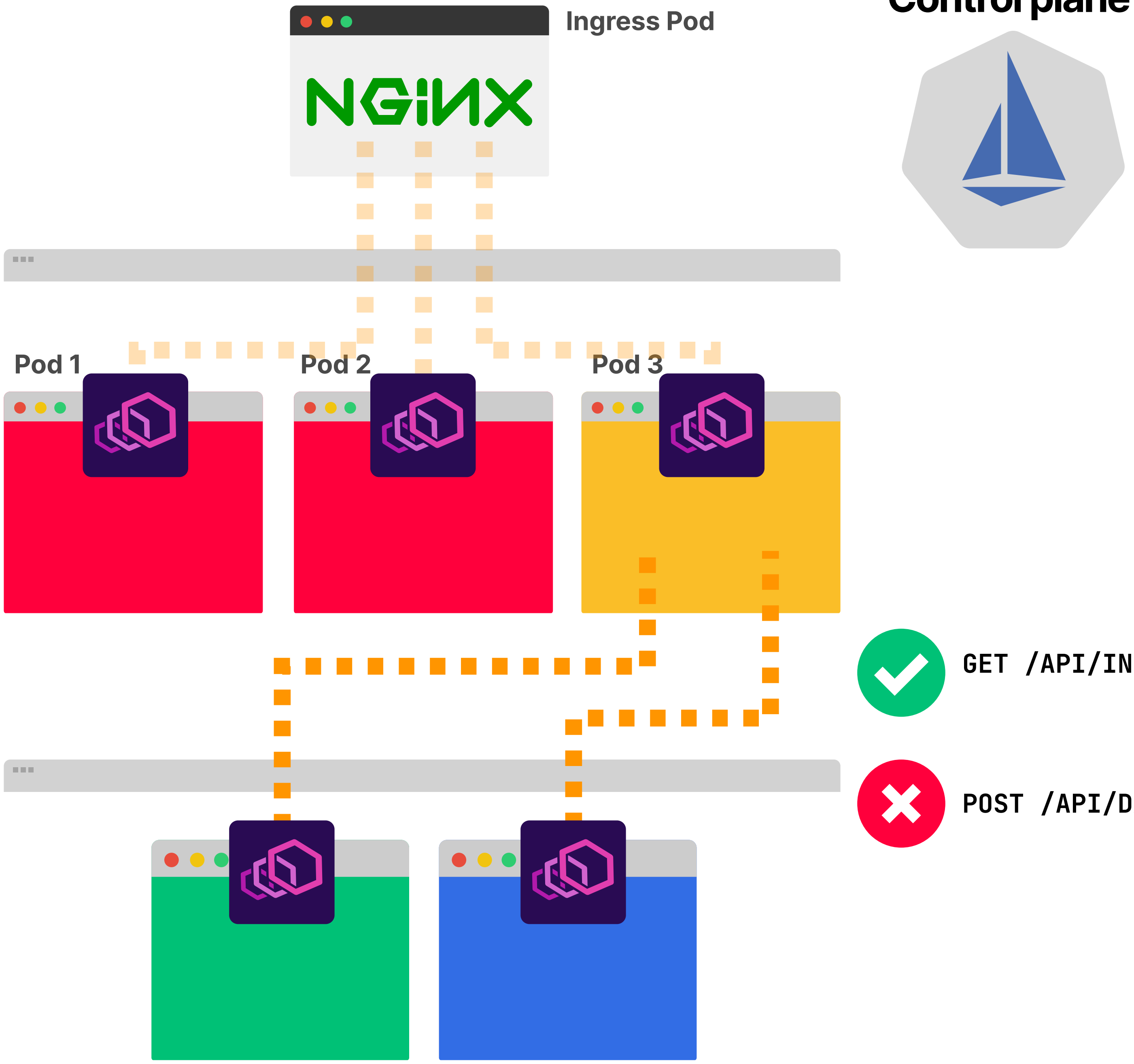
Pod 3

nested layers


Control plane
Ingress Pod
NGINX
Pod 1
Pod 2
Pod 3


Control plane
Ingress Pod
NGINX
Pod 1
Pod 2
Pod 3

Control plane



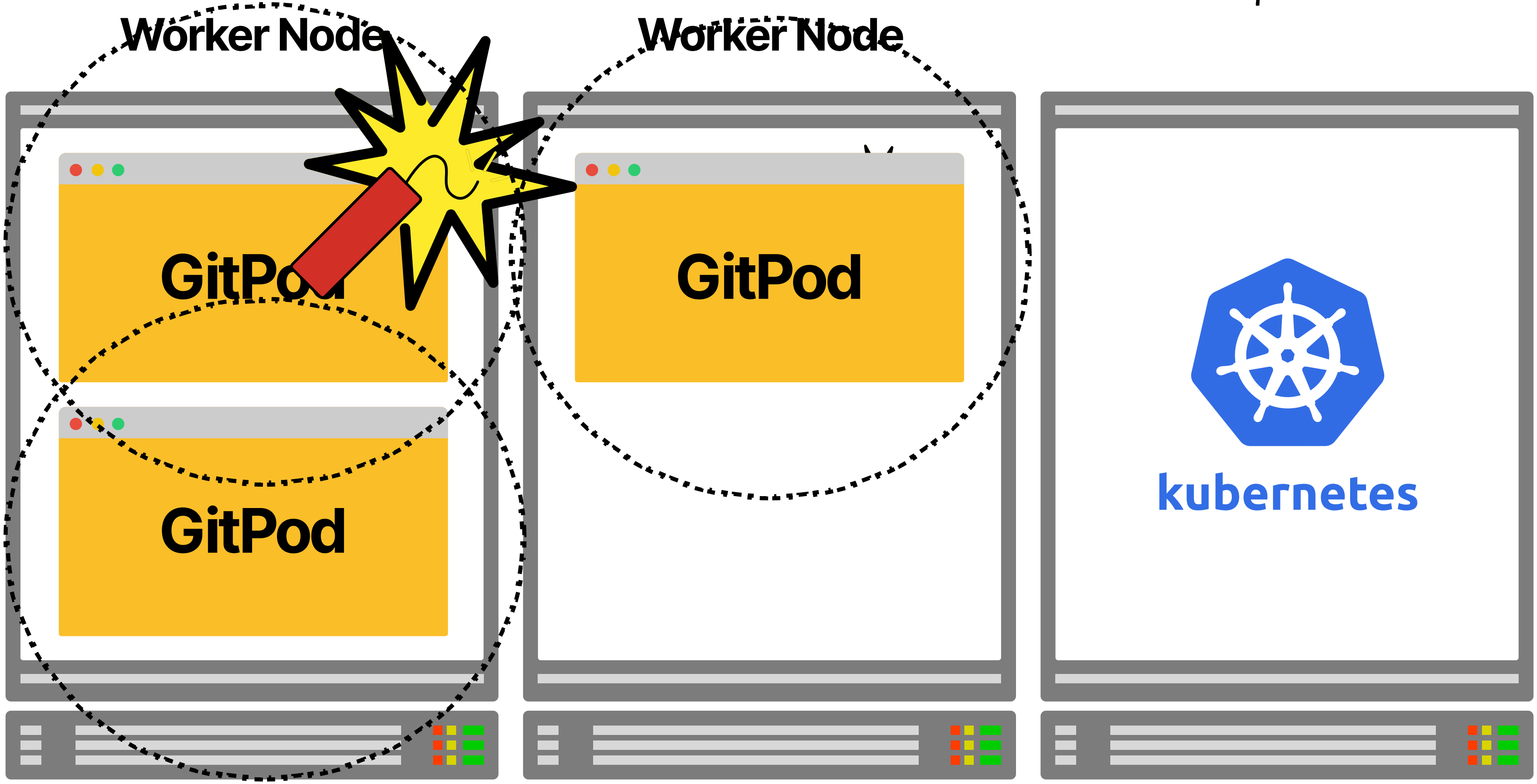
✓ GET /API/INFO

✗ POST /API/DATA

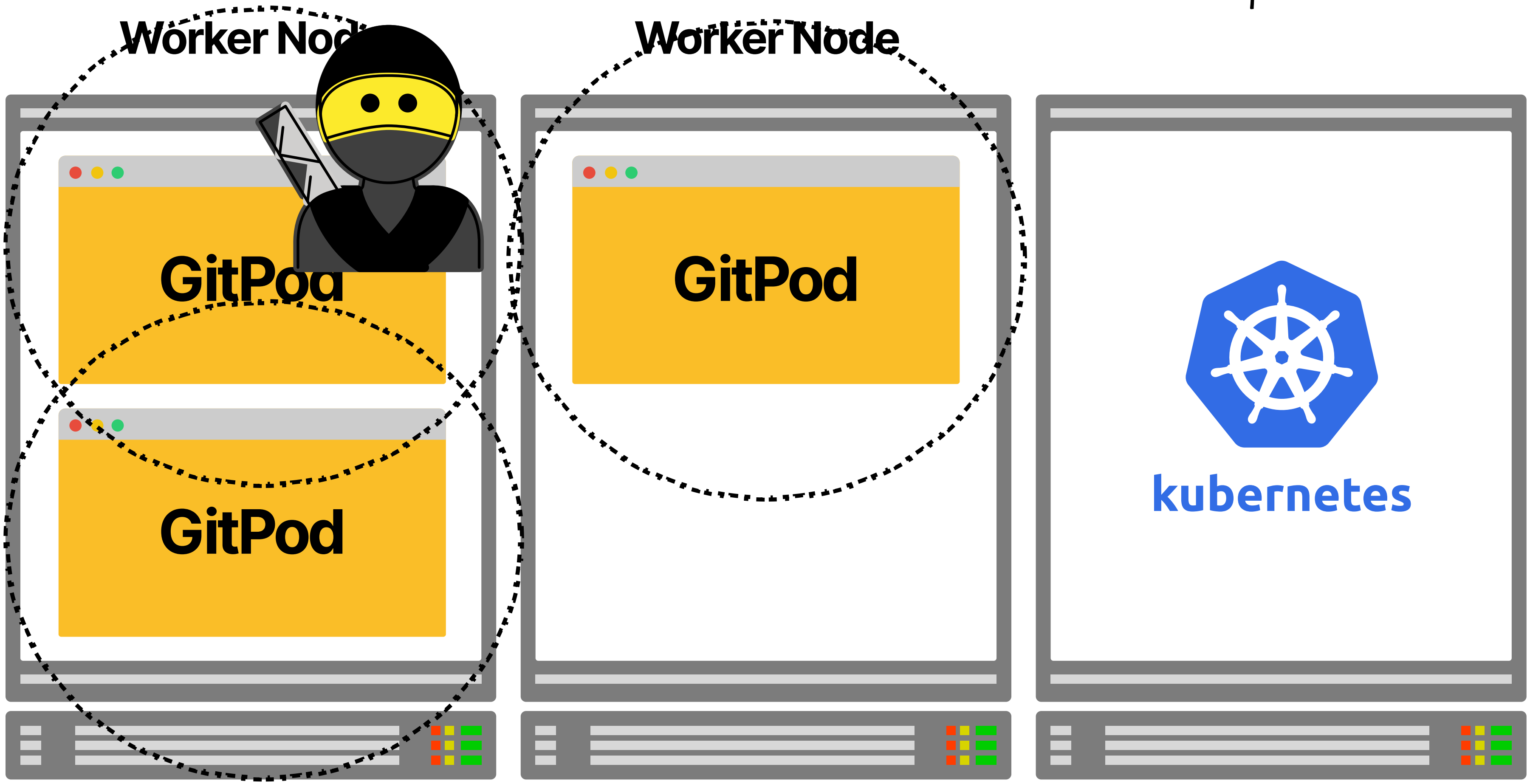
Workload isolation

Challenge 3

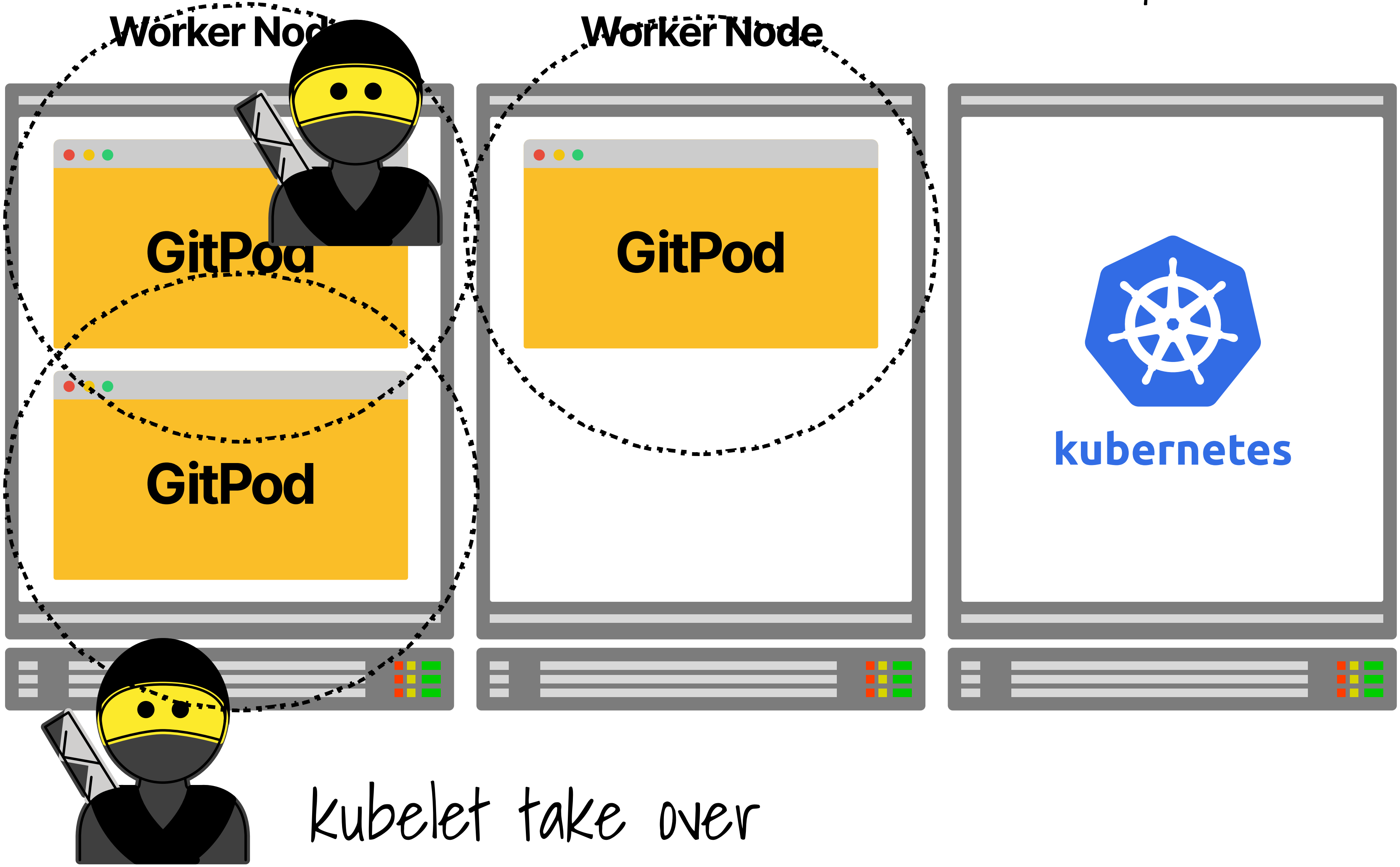
isolated workspace



isolated workspace

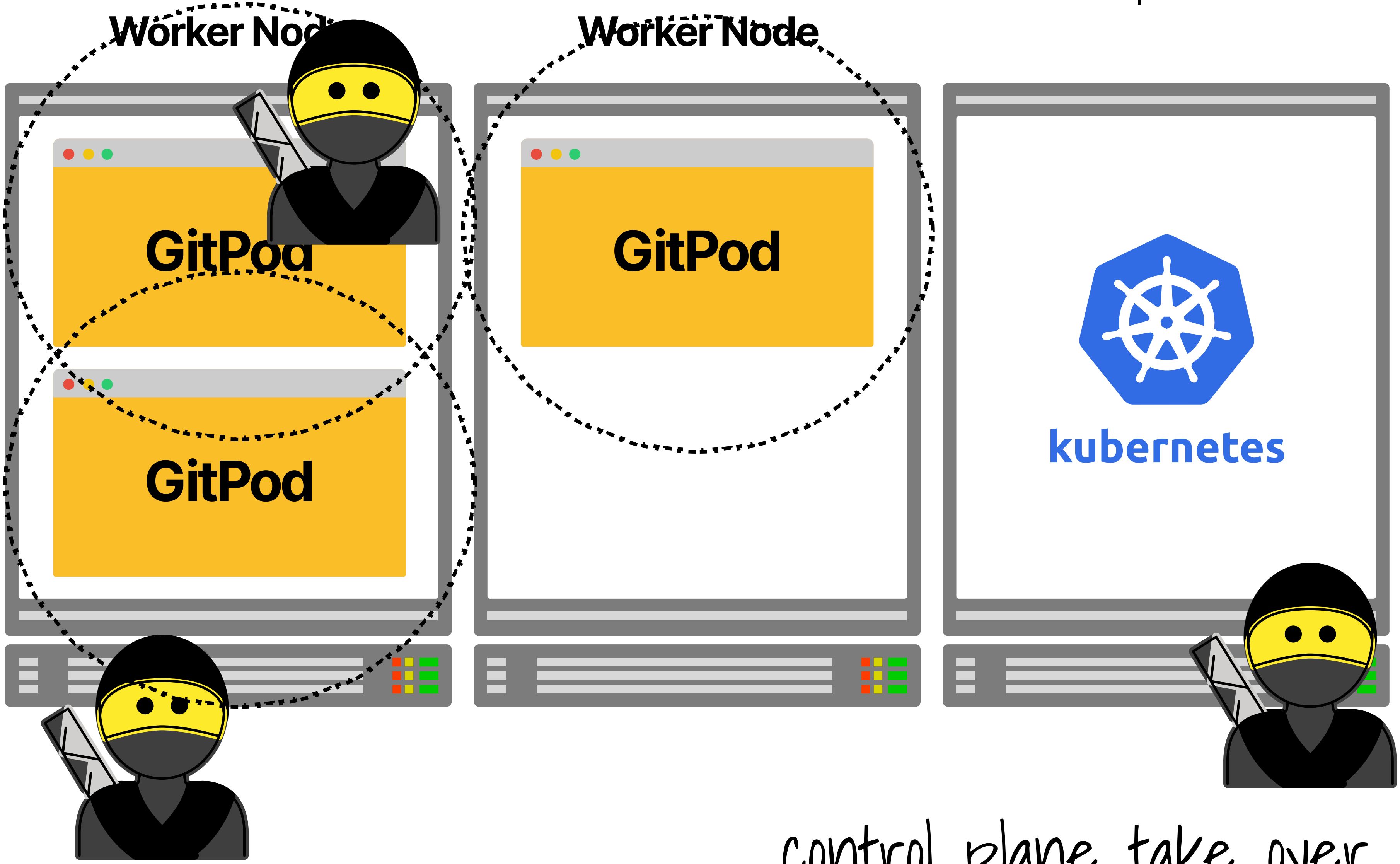


isolated workspace



kubelet take over

isolated workspace



control plane take over

Workload isolation

User namespaces*

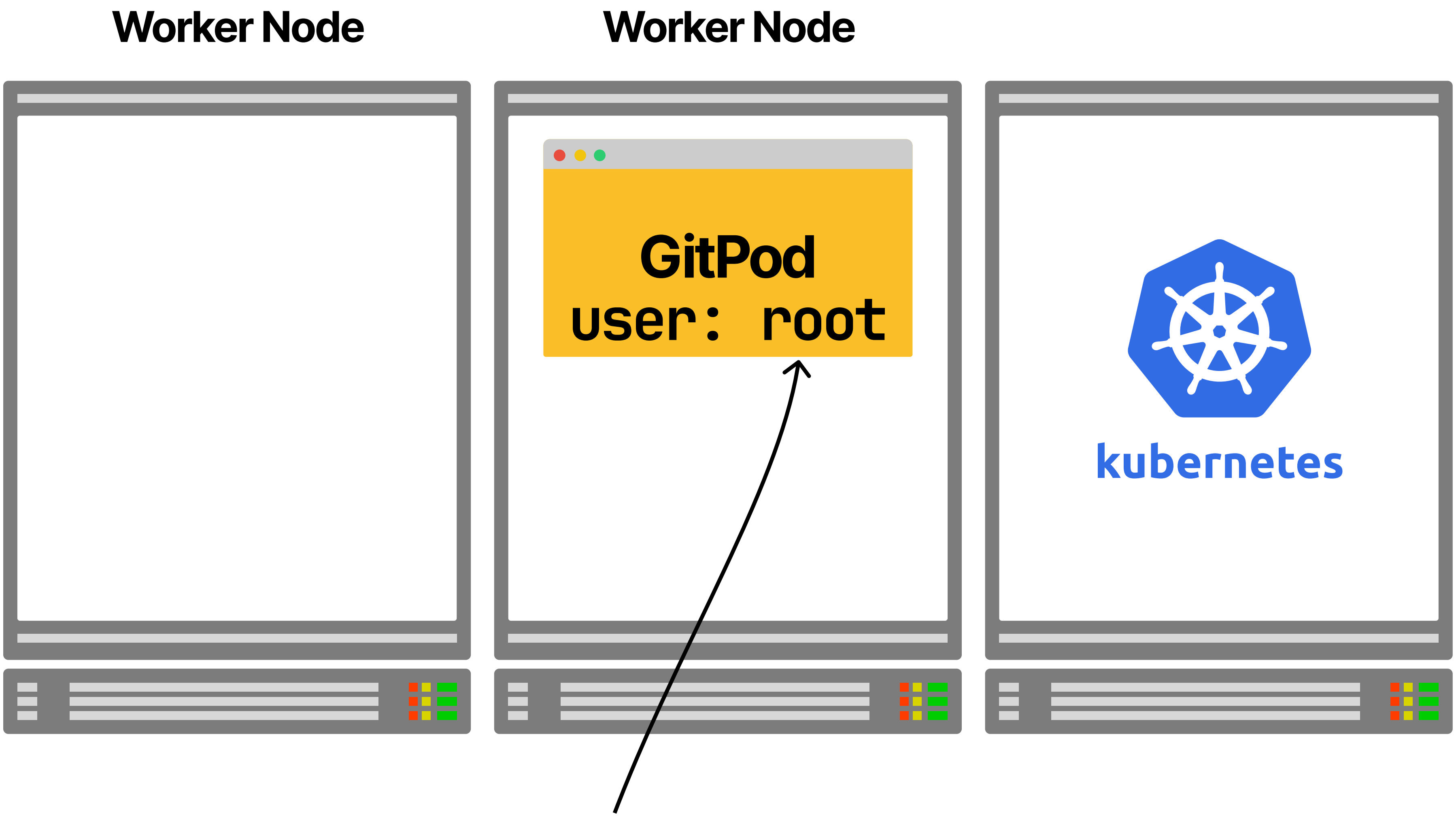
Sandboxed runtimes

Workload isolation

User namespaces*

Sandboxed runtimes

User namespaces



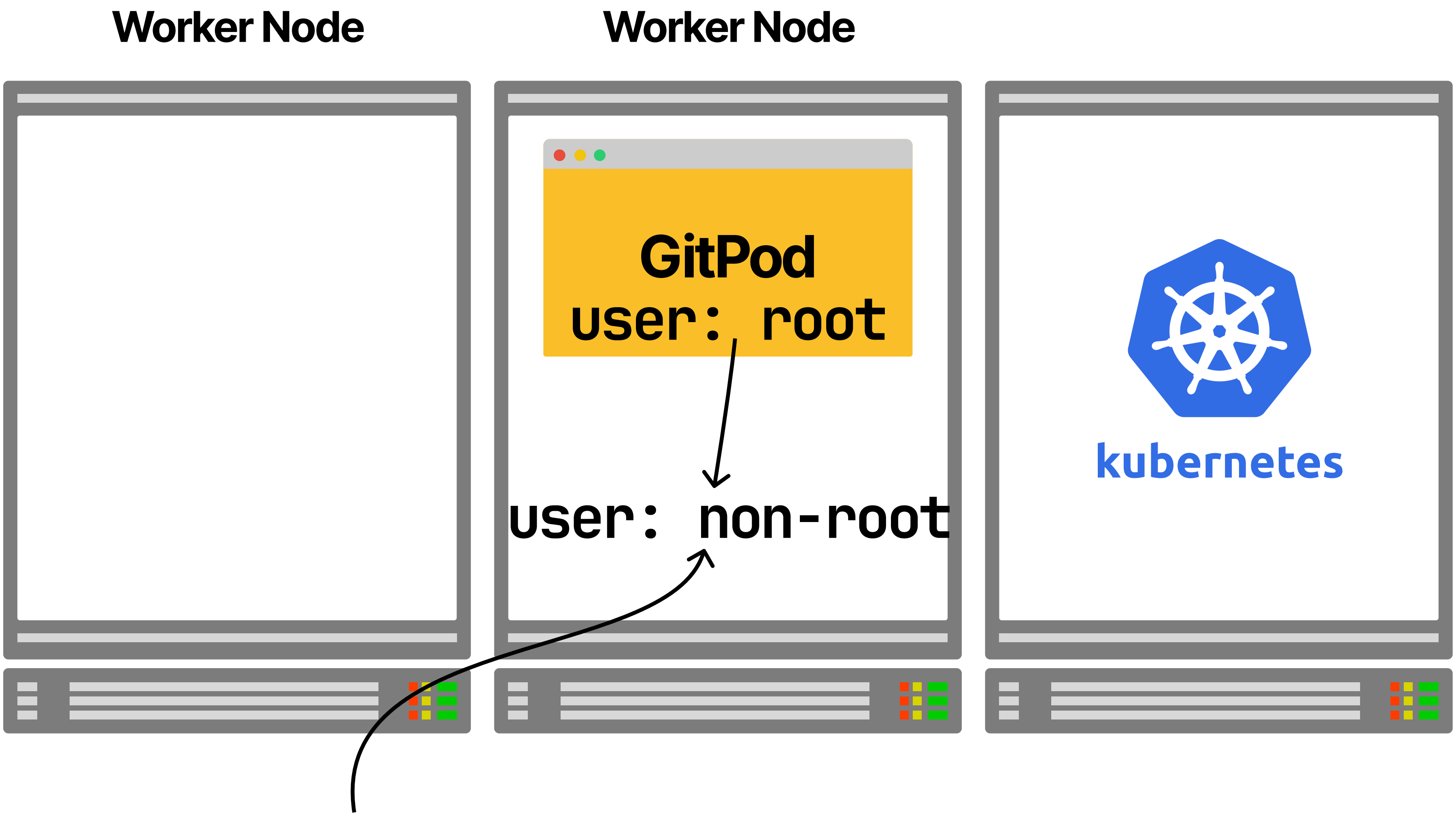
Worker Node

Worker Node

GitPod
user: root

kubernetes

the process in the container
runs as root



Worker Node

Worker Node

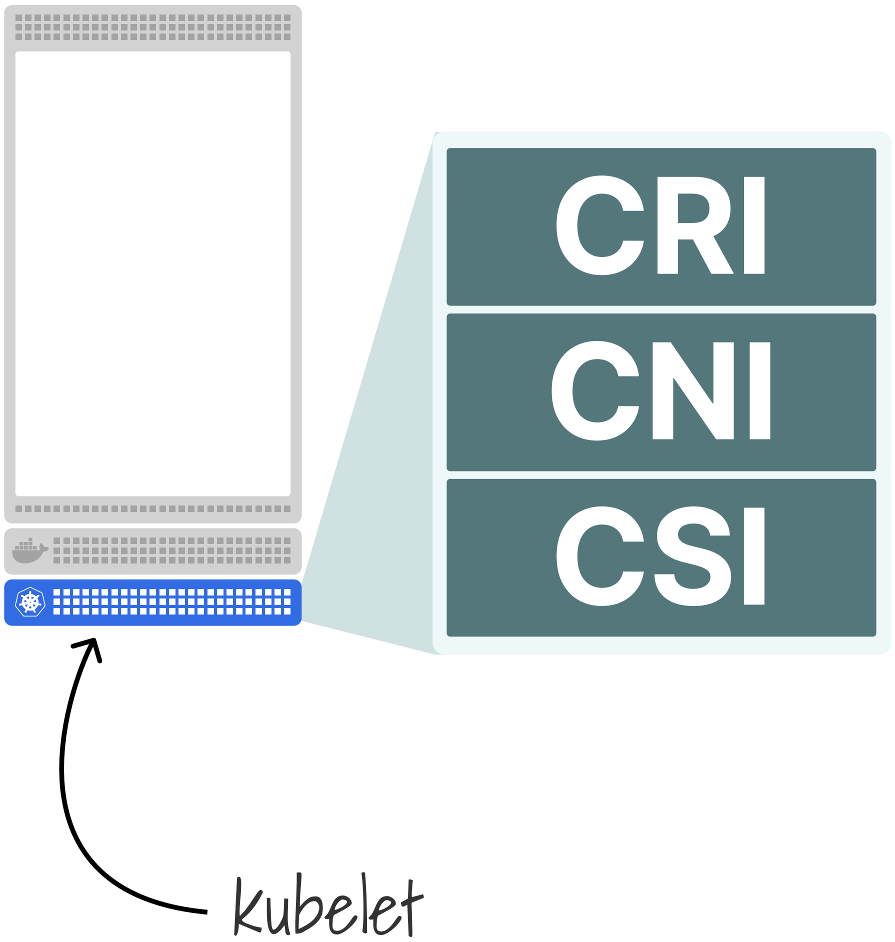
GitPod
user: root

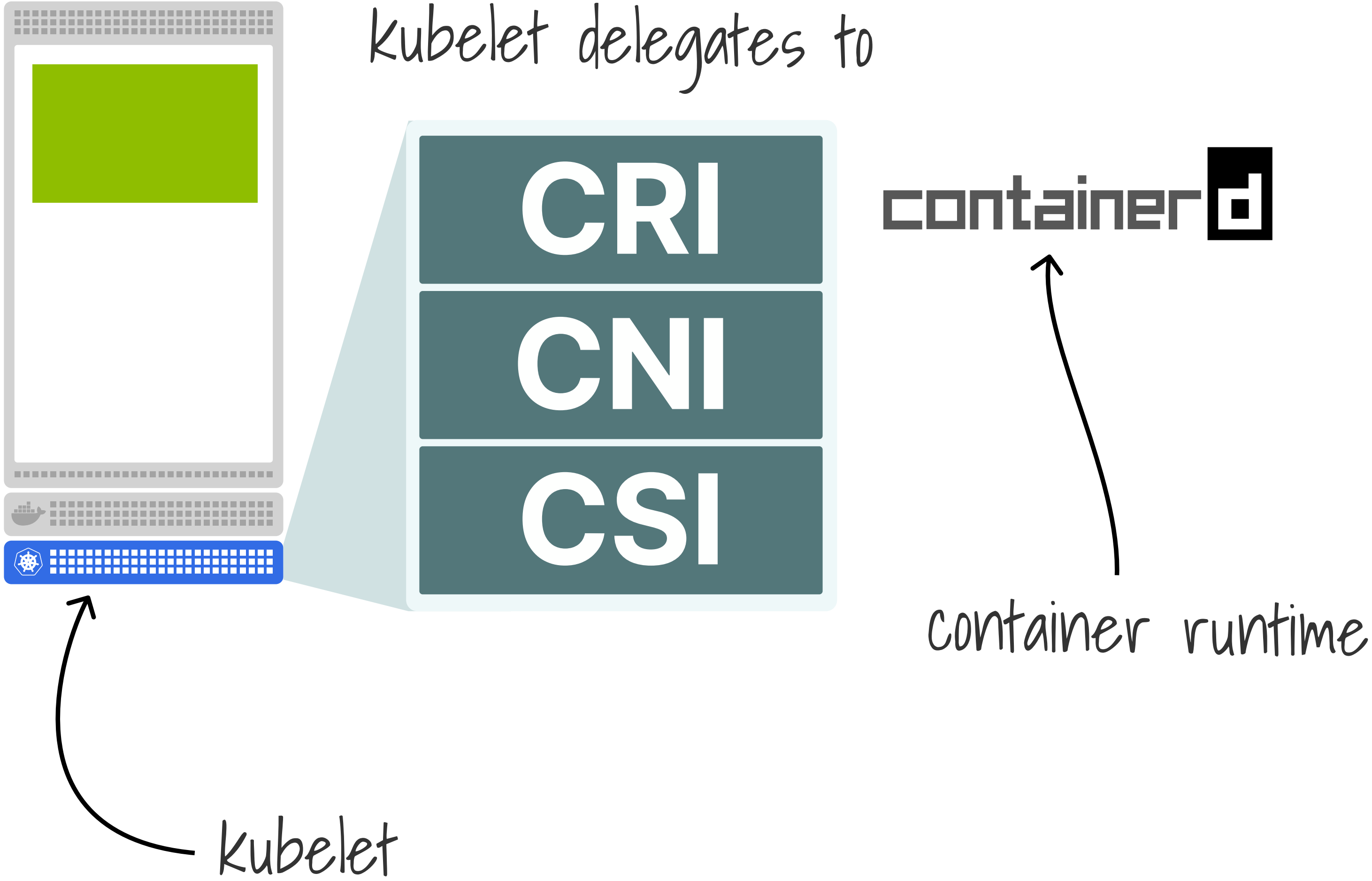
user: non-root

kubernetes

the process is mapped to a
non-root user in the host

Sandbox runtimes


CRI
CNI
CSI
kubelet



kubelet delegates to

CRI

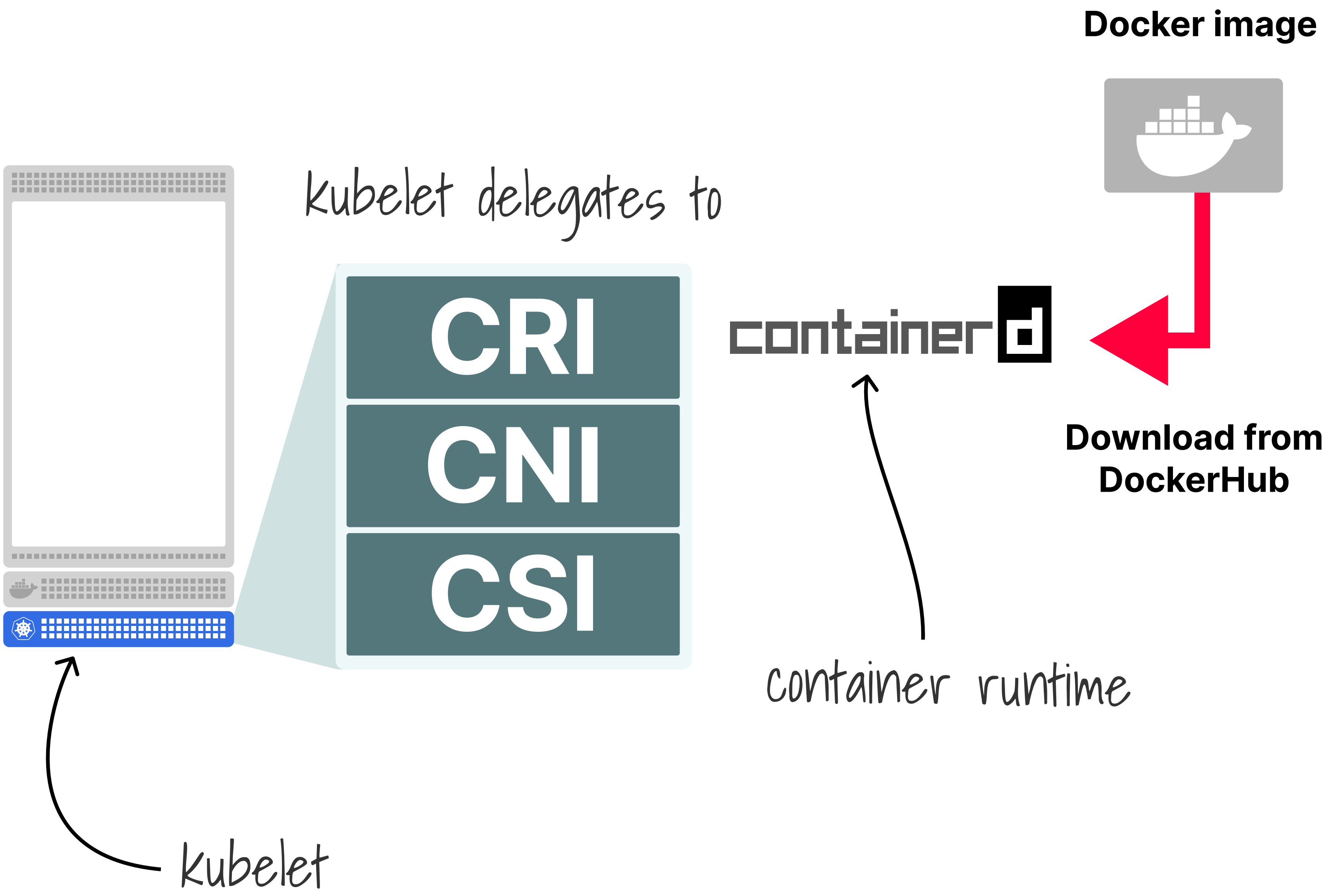
CNI

CSI

containerd

container runtime

kubelet



Docker image

kubelet delegates to

CRI

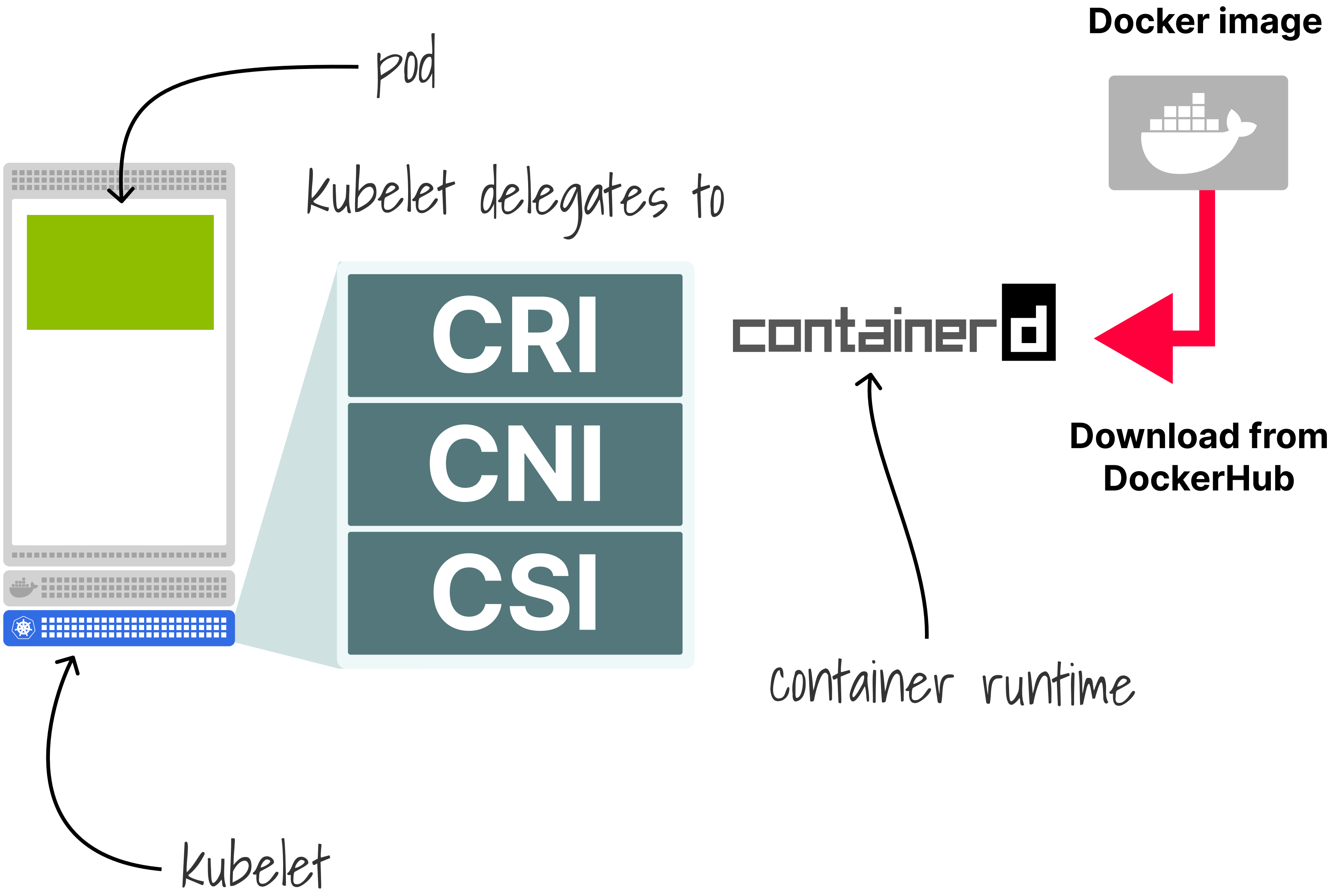
CNI

CSI

Download from DockerHub

container runtime

kubelet



pod

kubelet delegates to

CRI

CNI

CSI

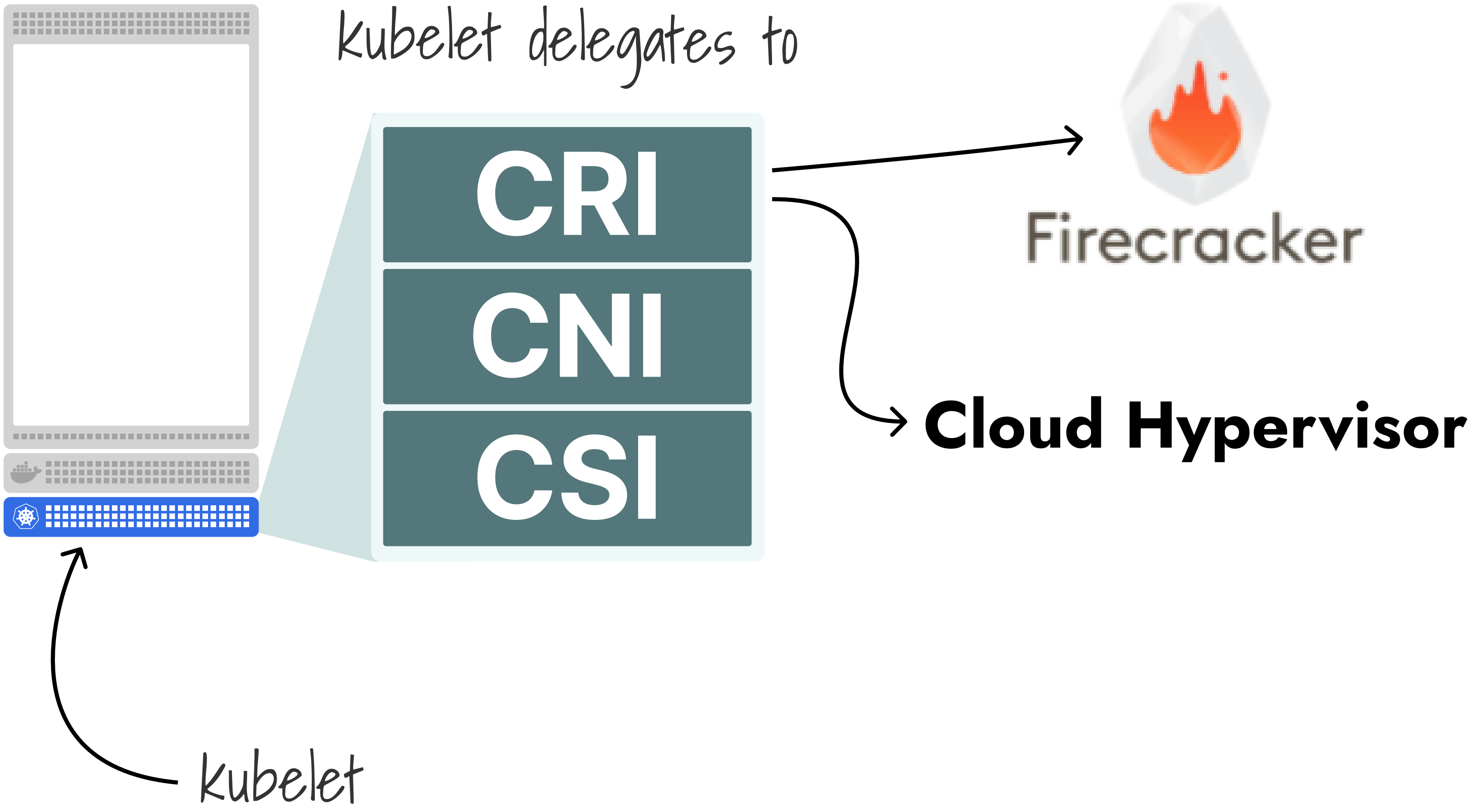
containerd

container runtime

Docker image

Download from DockerHub

kubelet



kubelet delegates to

CRI

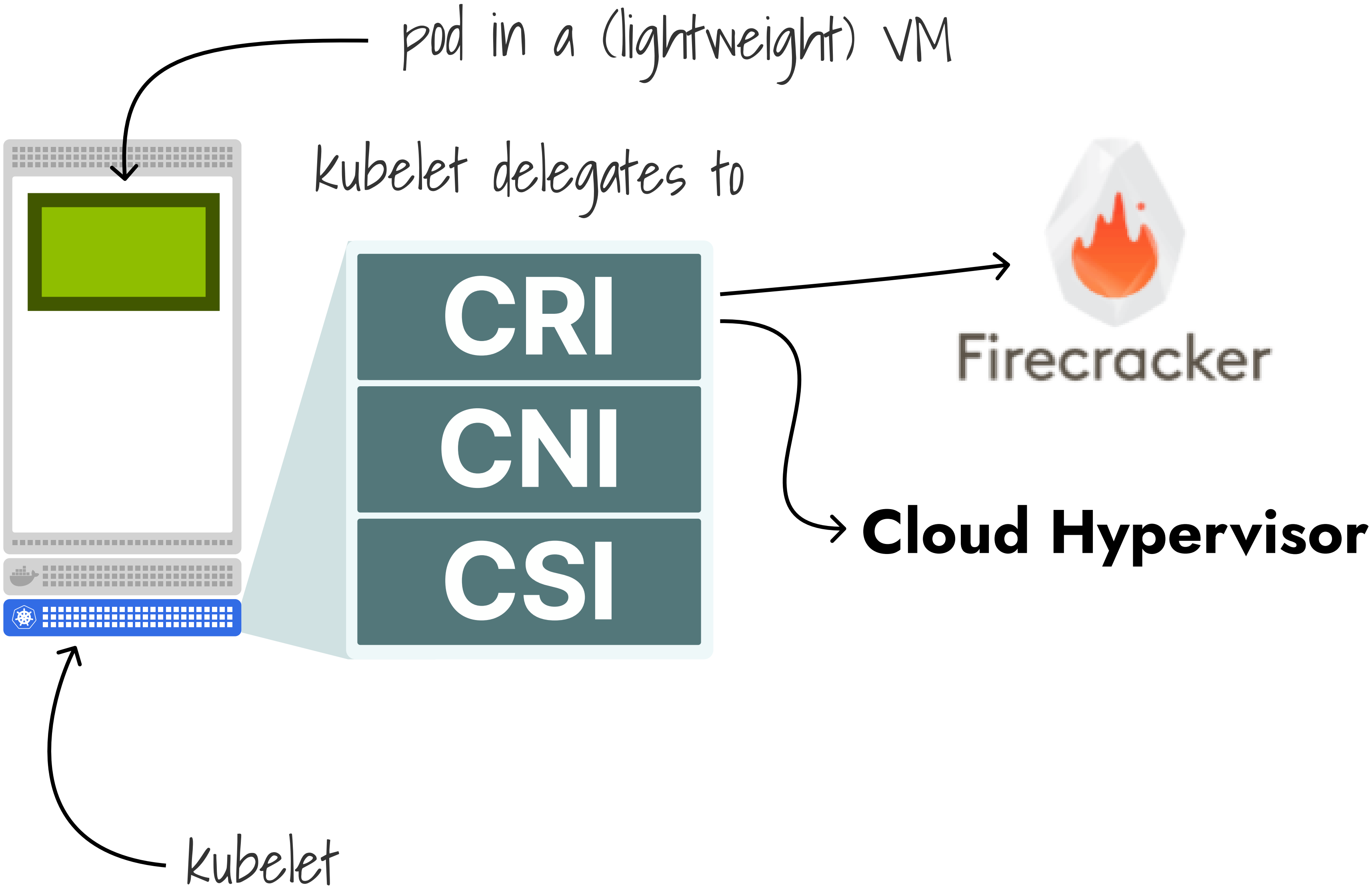
CNI

CSI

Firecracker

Cloud Hypervisor

kubelet



pod in a (lightweight) VM

kubelet delegates to

CRI

CNI

CSI

Firecracker

Cloud Hypervisor

kubelet

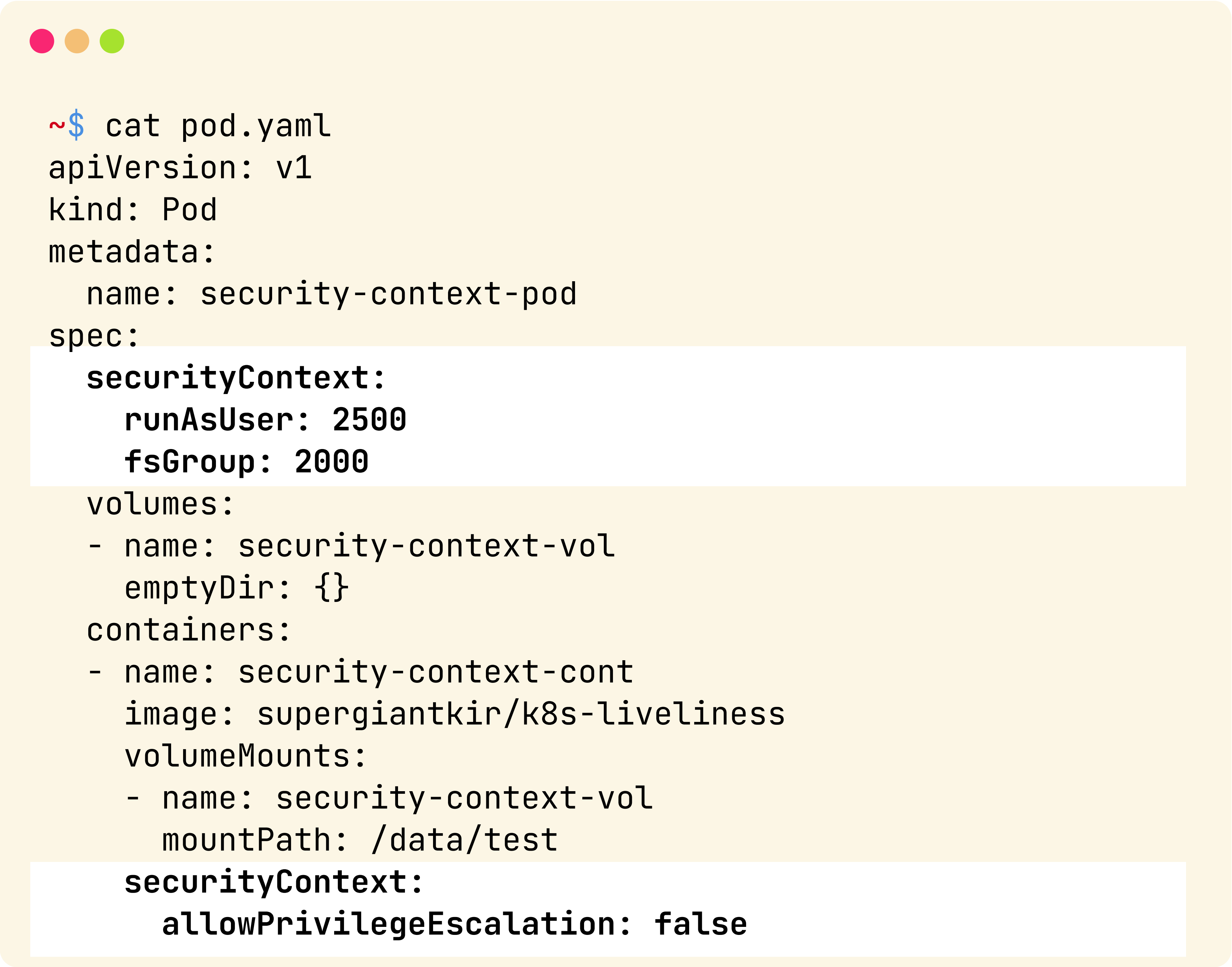
Your options

Workload isolation

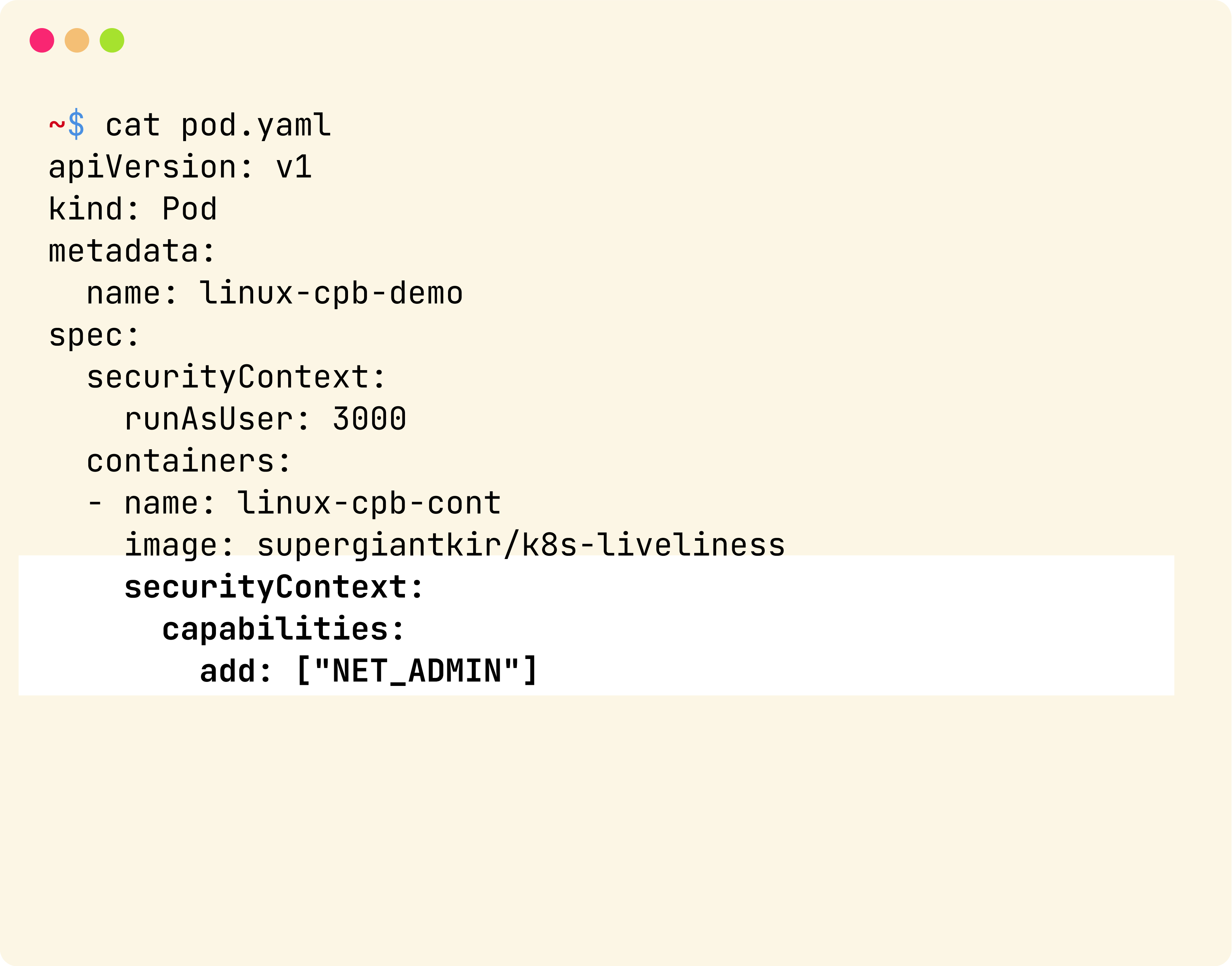
SecurityContext

gVisor

Sandboxed runtimes



```
~$ cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: security-context-pod
spec:
  securityContext:
    runAsUser: 2500
    fsGroup: 2000
  volumes:
  - name: security-context-vol
    emptyDir: {}
  containers:
  - name: security-context-cont
    image: supergiantkir/k8s-liveliness
    volumeMounts:
    - name: security-context-vol
      mountPath: /data/test
    securityContext:
      allowPrivilegeEscalation: false
```



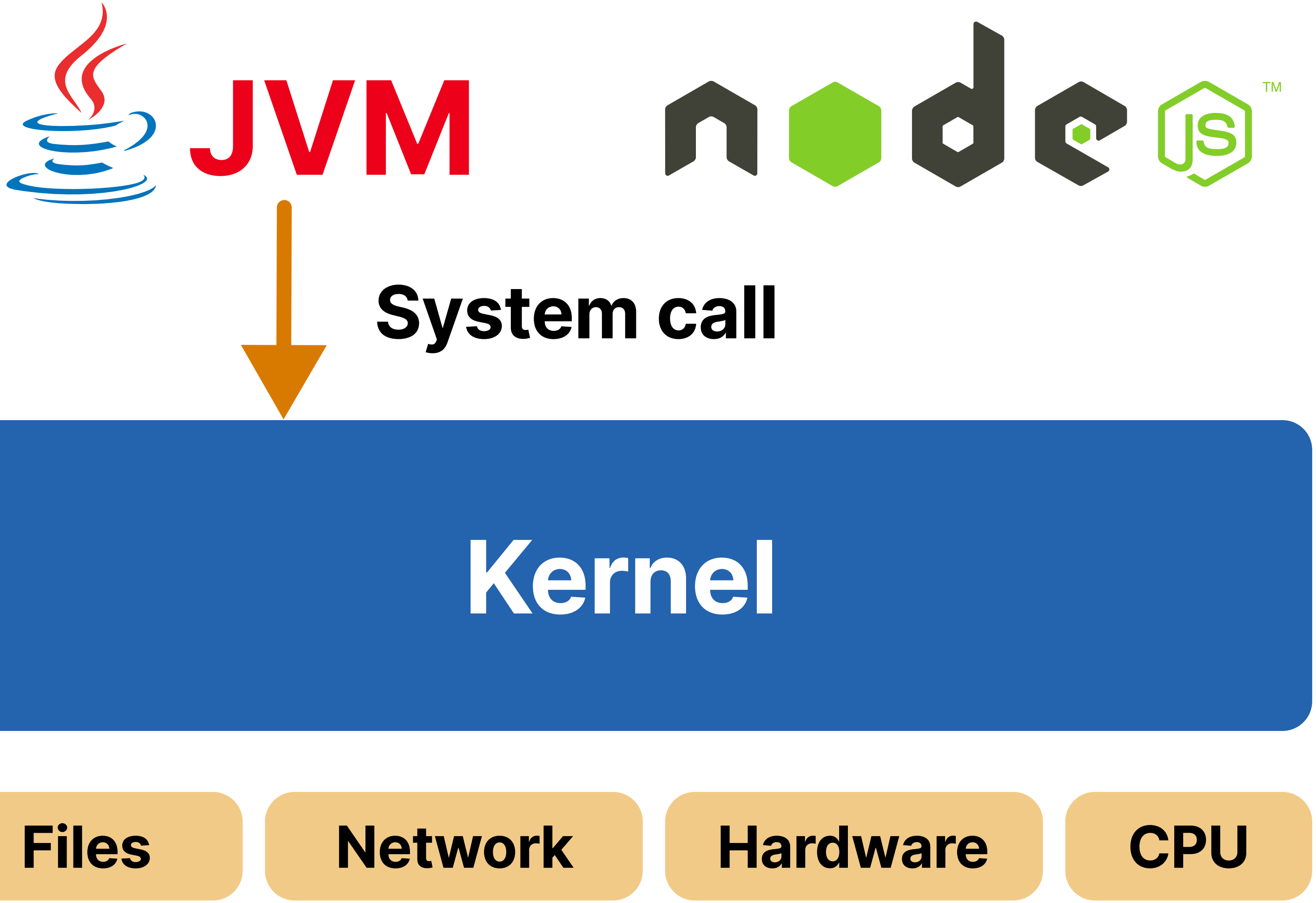
```
~$ cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: linux-cpb-demo
spec:
  securityContext:
    runAsUser: 3000
  containers:
  - name: linux-cpb-cont
    image: supergiantkir/k8s-liveliness
    securityContext:
      capabilities:
        add: ["NET_ADMIN"]
```

Workload isolation

SecurityContext

gVisor

Sandboxed runtimes



JVM

node JS

System call

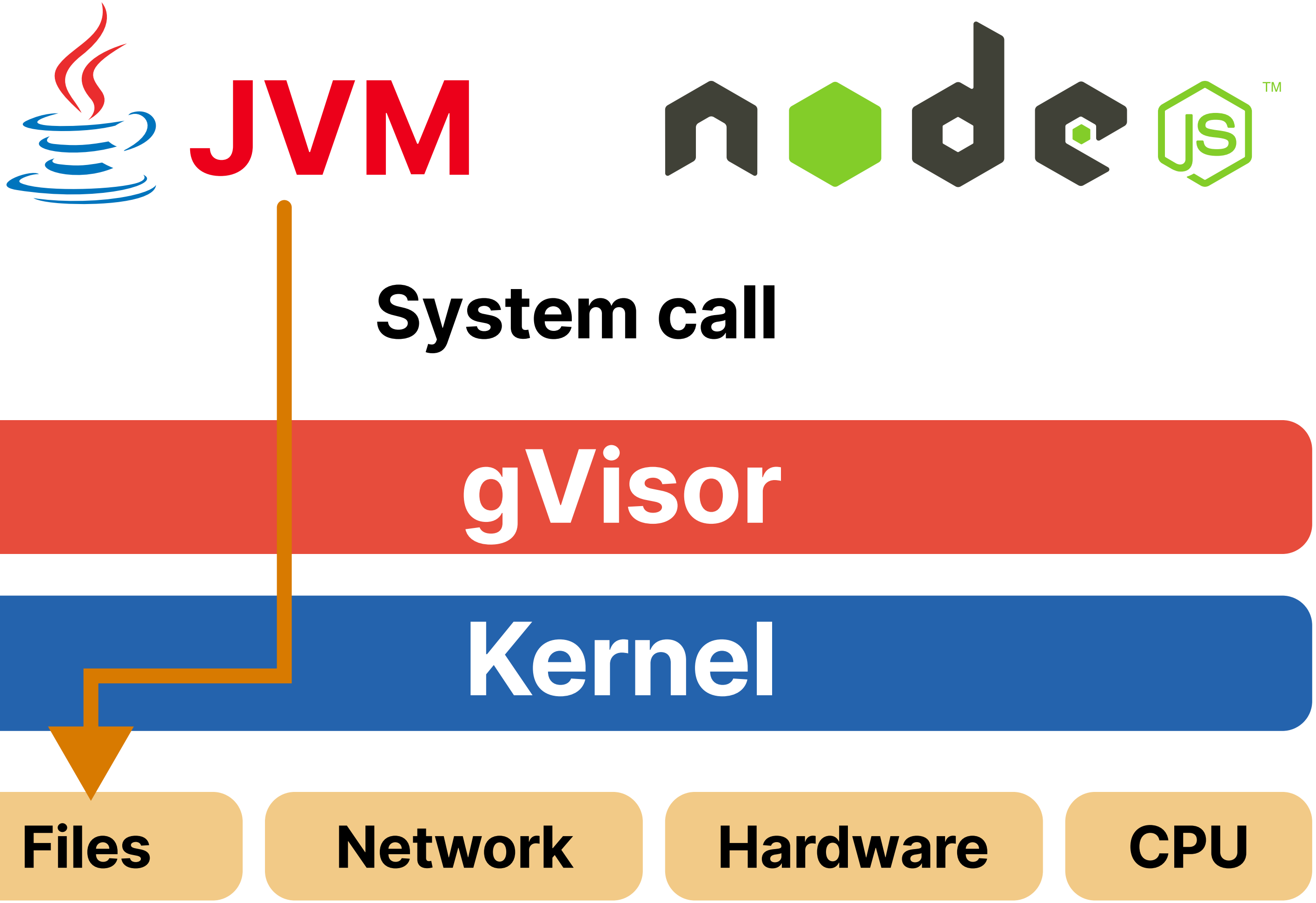
Kernel

Files

Network

Hardware

CPU



JVM

node JS

System call

gVisor

Kernel

Files

Network

Hardware

CPU

Workload isolation

SecurityContext

gVisor

Sandboxed runtimes

Firecracker

Cloud Hypervisor

katacontainers

EDERA

WA

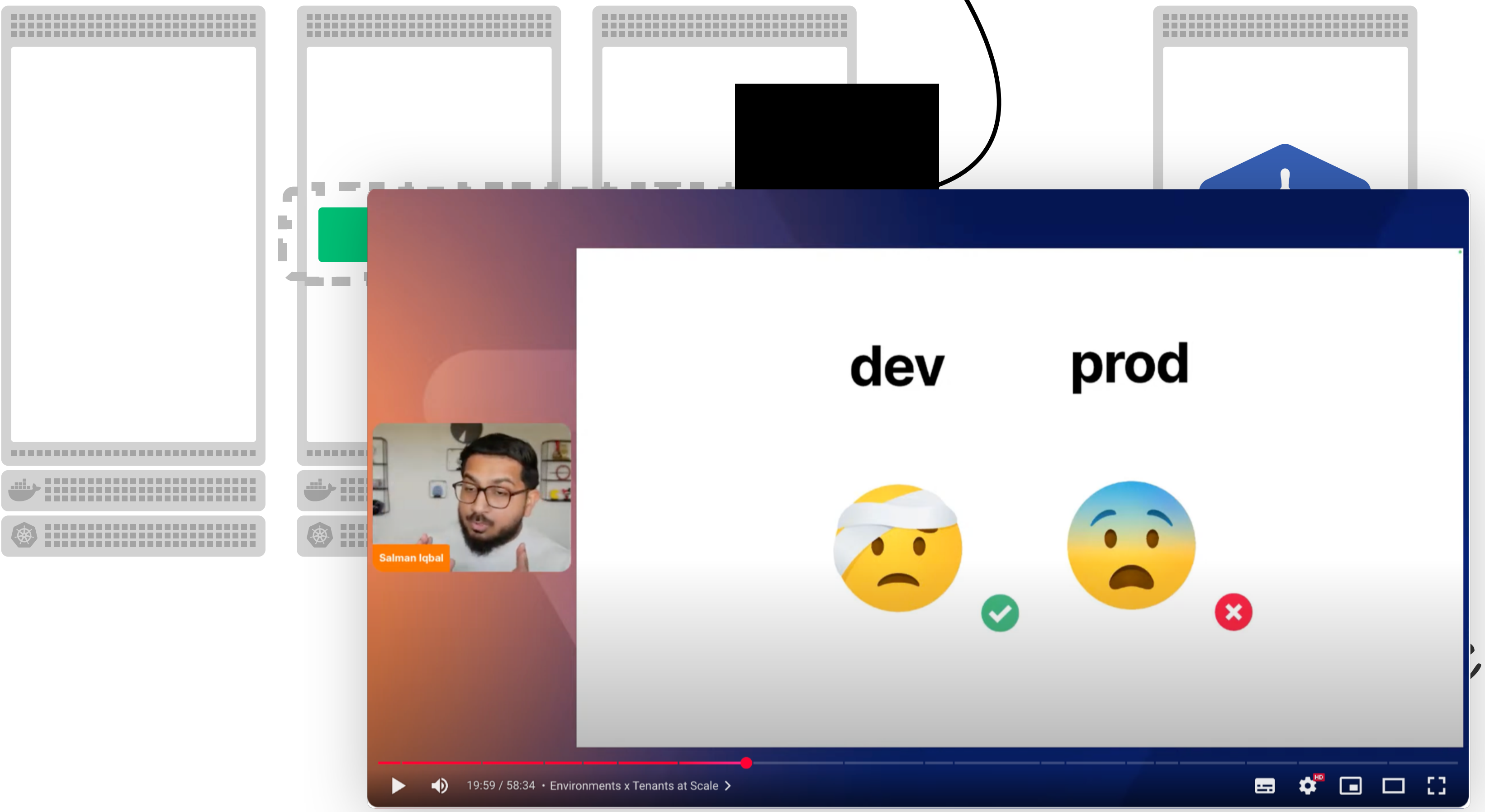
varnish/tinykvm*

container escape

POOL 1

CONTROL PLANE

[redacted]



dev

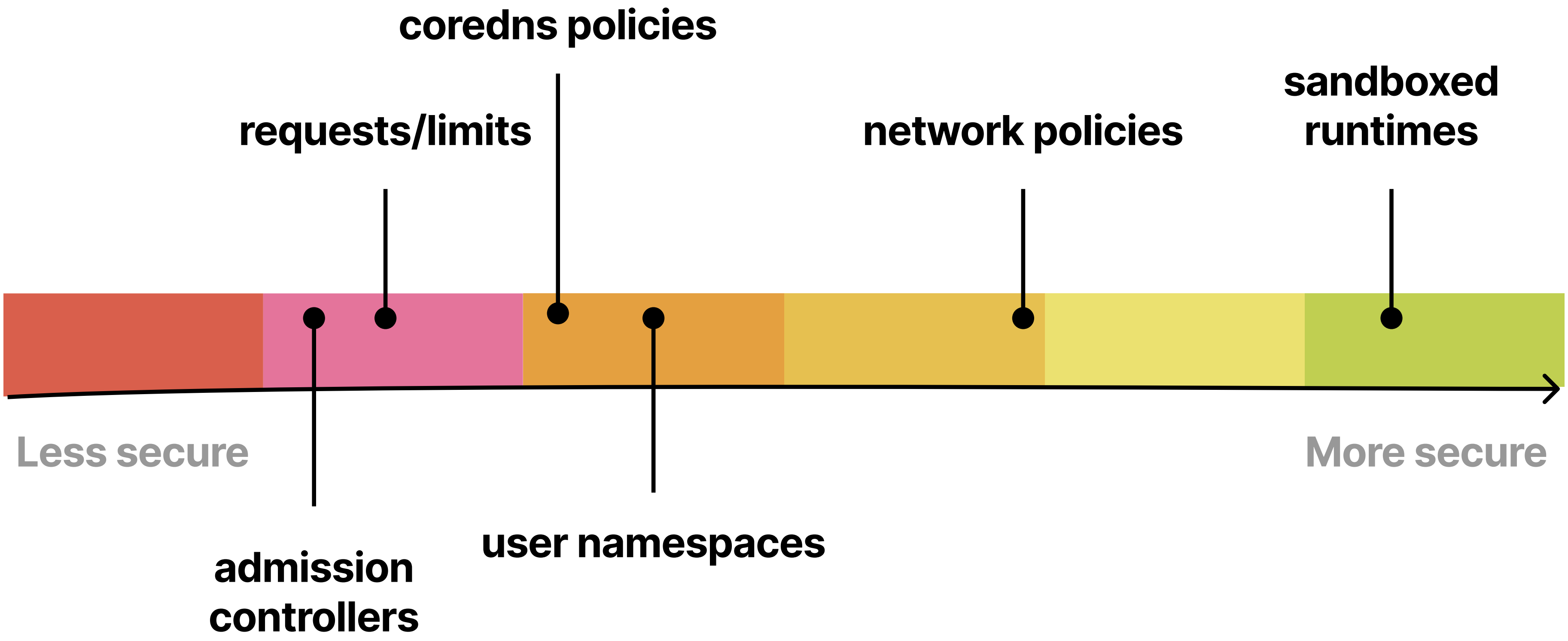
prod

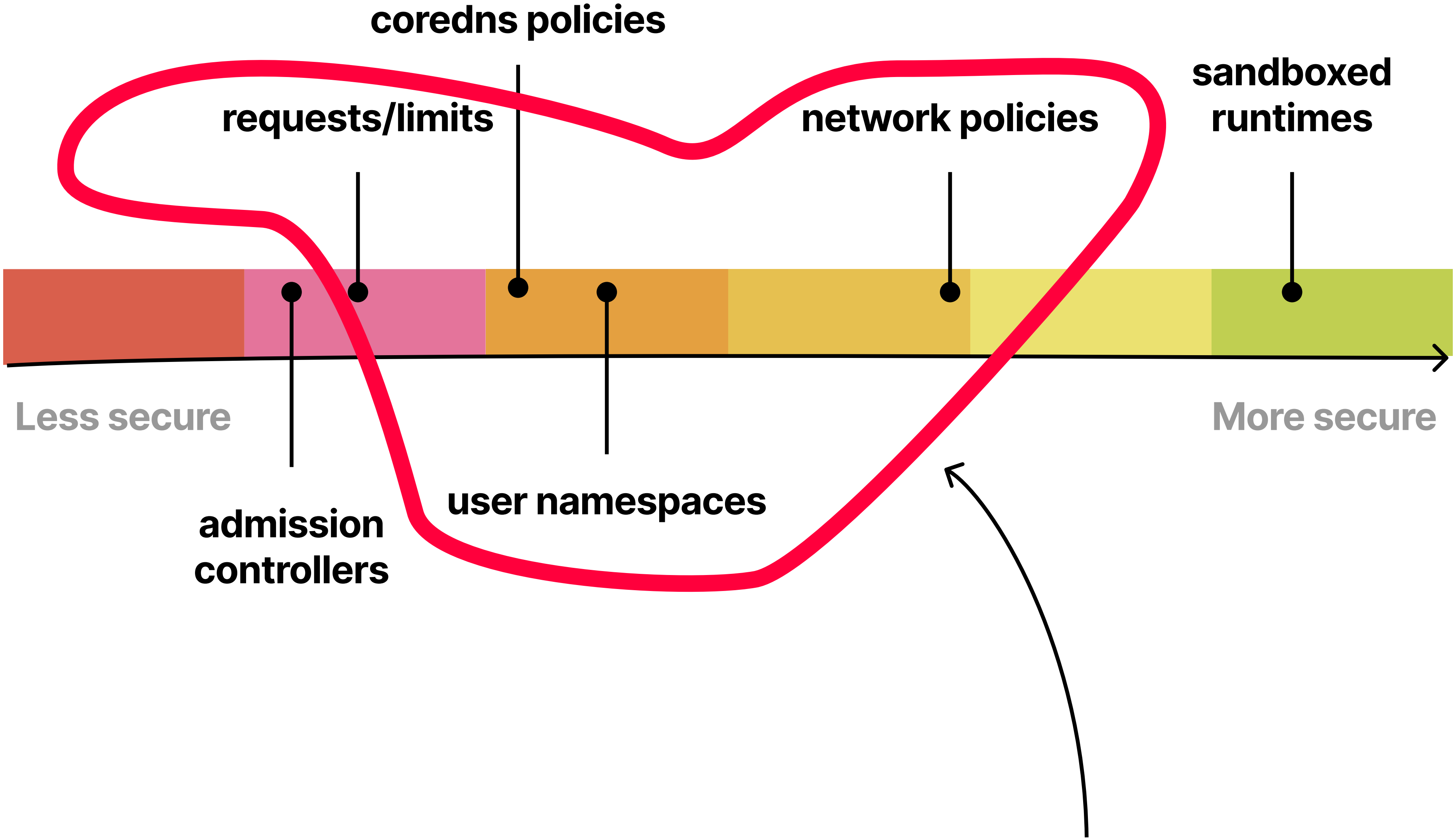
19:59 / 58:34 • Environments x Tenants at Scale

Security as a spectrum

Less secure

More secure


coredns policies
requests/limits
network policies
sandboxed runtimes
Less secure
More secure
admission controllers
user namespaces



coredns policies

requests/limits

network policies

sandboxed
runtimes

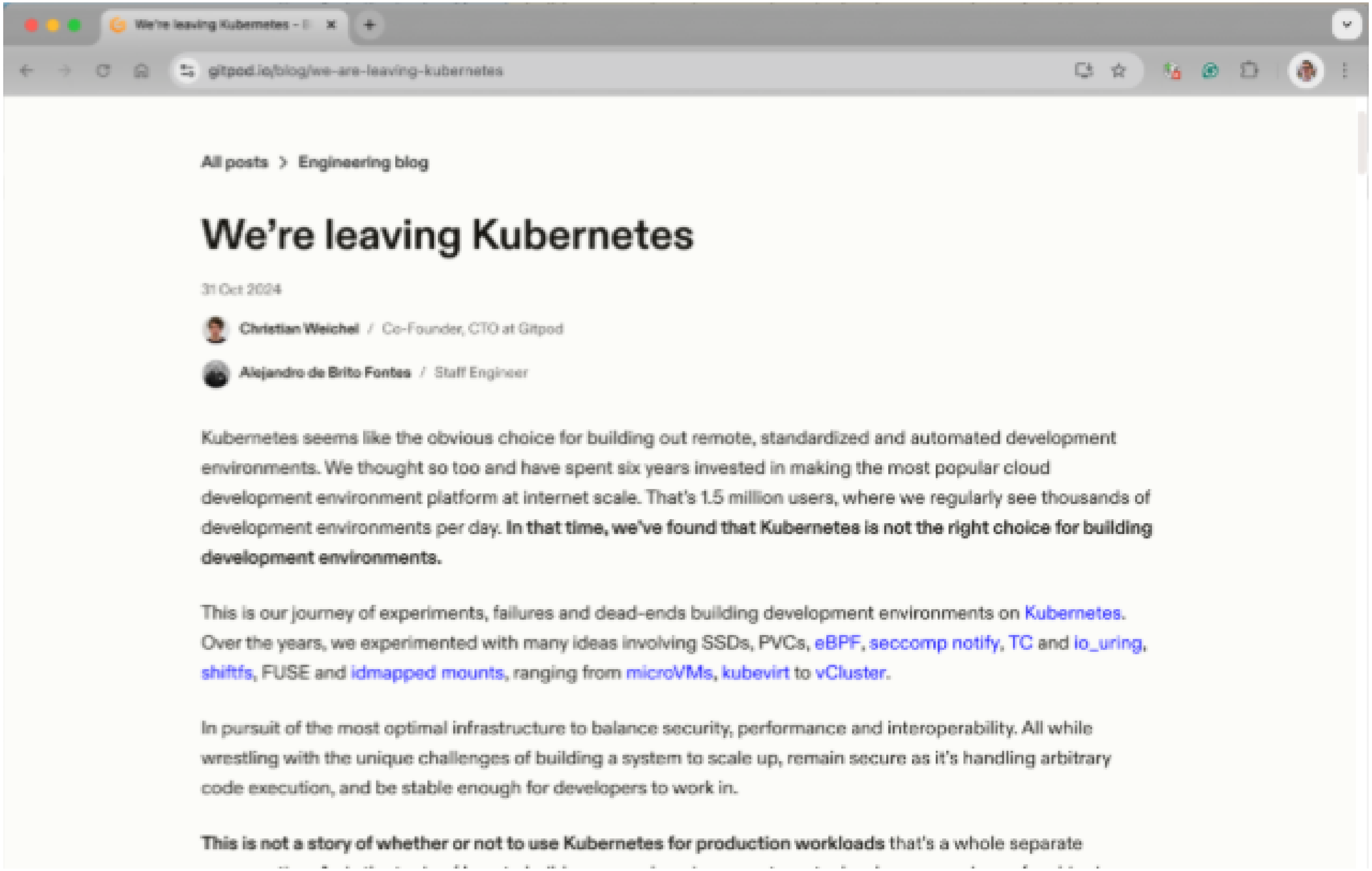
Less secure

More secure

admission
controllers

user namespaces

mix and match


We're leaving Kubernetes - ...
gitpod.io/blog/we-are-leaving-kubernetes

All posts > Engineering blog

We're leaving Kubernetes

31 Oct 2024

Christian Weichel / Co-Founder, CTO at Gitpod

Alejandro de Brito Fontes / Staff Engineer

Kubernetes seems like the obvious choice for building out remote, standardized and automated development environments. We thought so too and have spent six years invested in making the most popular cloud development environment platform at internet scale. That's 1.5 million users, where we regularly see thousands of development environments per day. **In that time, we've found that Kubernetes is not the right choice for building development environments.**

This is our journey of experiments, failures and dead-ends building development environments on Kubernetes. Over the years, we experimented with many ideas involving SSDs, PVCs, eBPF, seccomp notify, TC and io_uring, shiftfs, FUSE and idmapped mounts, ranging from microVMs, kubevirt to vCluster.

In pursuit of the most optimal infrastructure to balance security, performance and interoperability. All while wrestling with the unique challenges of building a system to scale up, remain secure as it's handling arbitrary code execution, and be stable enough for developers to work in.

This is not a story of whether or not to use Kubernetes for production workloads that's a whole separate

31 Oct 2024

Security as a spectrum

Understand constraints

Define goals

Tooling

Security as a spectrum

Understand constraints

Define goals

Tooling

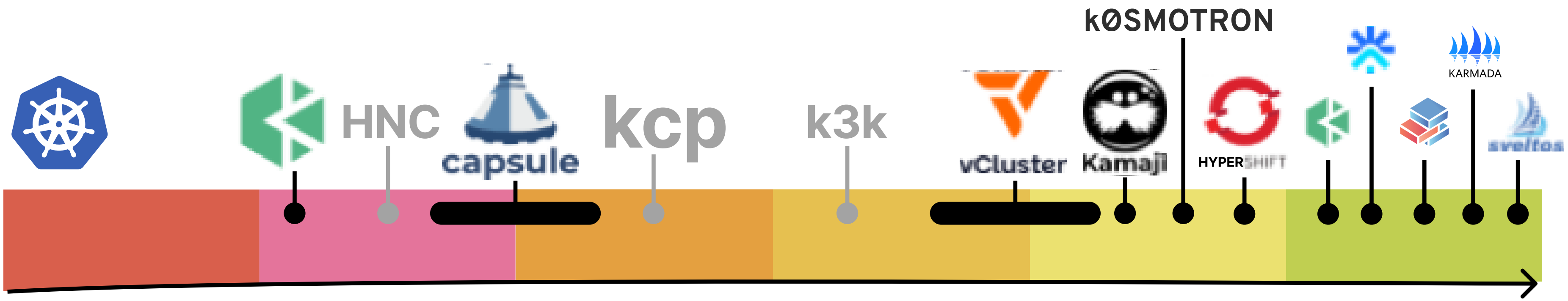
Security as a spectrum

Understand constraints

Define goals

Tooling

Multi-tenancy spectrum



🟥 Namespaces

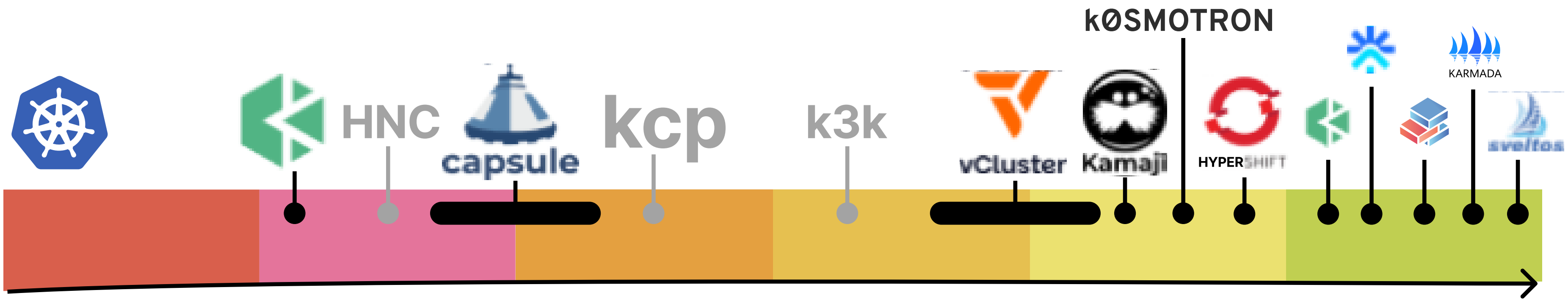
🟪 Namespaces as a Service

🟧 Kubernetes API as a Service

🟧 Control plane as a service (internal)

🟨 Control plane as a service (external)

🟩 Dedicated clusters



🟥 Namespaces

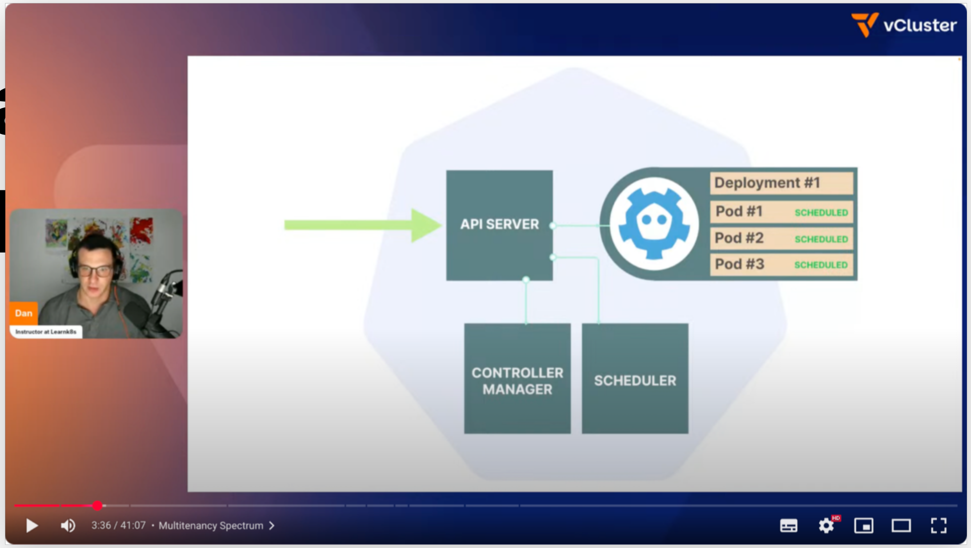
🟪 Namespaces a

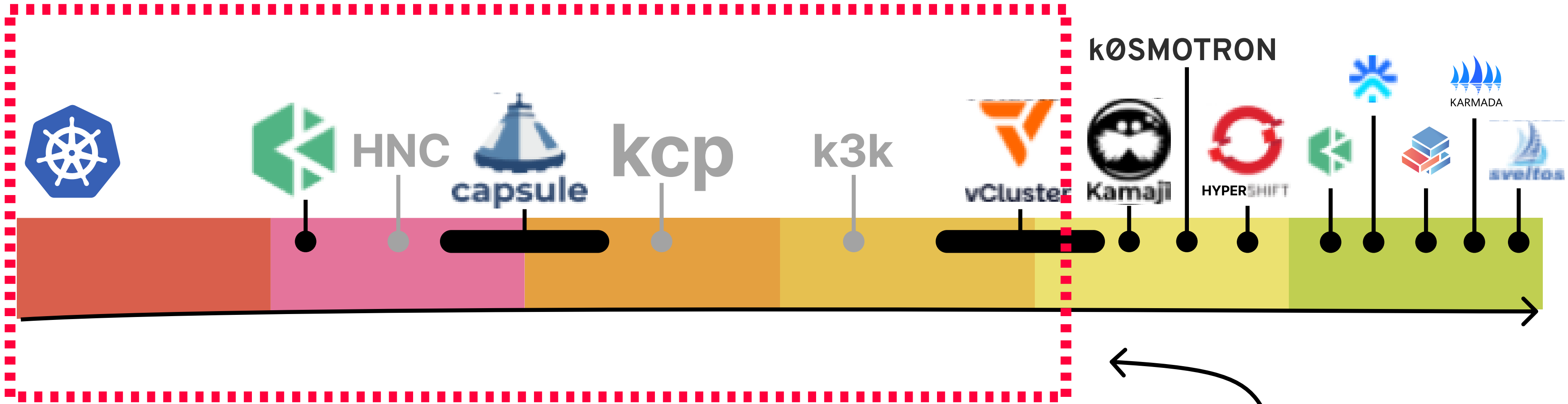
🟧 Kubernetes AP

🟨 Control plane

🟨 Control plane

🟩 Dedicated clusters





lighter options

🟥 **Namespaces**

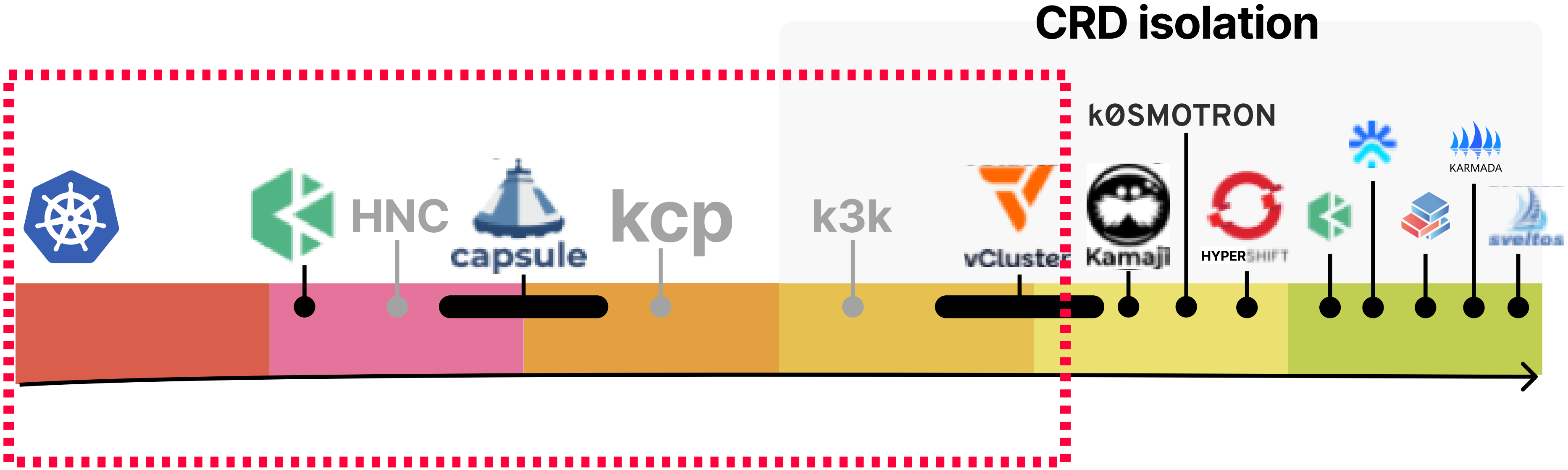
🟪 **Namespaces as a Service**

🟧 **Kubernetes API as a Service**

🟨 **Control plane as a service (internal)**

🟨 **Control plane as a service (external)**

🟩 **Dedicated clusters**



🟥 Namespaces

🟪 Namespaces as a Service

🟧 Kubernetes API as a Service

🟨 **Control plane as a service (internal)**

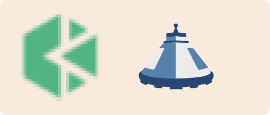
CRD isolaiton

🟨 Control plane as a service (external)

🟩 Dedicated clusters

SINGLE CLUSTER

SHARED CONTROL PLANE



DEDICATED CONTROL PLANE

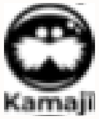
SINGLE CLUSTER

SHARED CONTROL PLANE

DEDICATED CONTROL PLANE

CLUSTER MANAGER

kØSMOTRON



HYPE SHIFT

R

SINGLE CLUSTER

SHARED CONTROL PLANE

DEDICATED CONTROL PLANE

vCluster

CLUSTER MANAGER

kØSMOTRON

Kamaji

HYPE SHIFT

R

DEDICATED CLUSTER

KARMADA

sveltos

SINGLE CLUSTER

SHARED CONTROL PLANE

DEDICATED CONTROL PLANE

vCluster

ADD ONS

- Network policies
- Sandboxed runtime
- CoreDNS Policies
- Service meshes
- Kube-proxy alternative

CLUSTER MANAGER

kØSMOTRON

Kamaji

HYPE SHIFT

R

ADD ONS

- Sandboxed runtime
- Service meshes

DEDICATED CLUSTER

KARMADA

sveltos

ADD ONS

- Service meshes

Takeaways

Recap

Recap

1. Kubernetes is for sharing
2. Sharing resources is hard
3. Sharing network is hard
4. Securing shared workload is hard
5. Security as spectrum

Recap

1. Kubernetes is for sharing
2. **Sharing resources is hard**
3. Sharing network is hard
4. Securing shared workload is hard
5. Security as spectrum

Recap

1. Kubernetes is for sharing
2. Sharing resources is hard
3. **Sharing network is hard**
4. Securing shared workload is hard
5. Security as spectrum

Recap

1. Kubernetes is for sharing

2. Sharing resources is hard

3. Sharing network is hard

4. **Securing shared workload is hard**

5. Security as spectrum

Recap

1. Kubernetes is for sharing
2. Sharing resources is hard
3. Sharing network is hard
4. Securing shared workload is hard
5. **Security as spectrum**

vCluster

Thank you!

Thank you!

Chris Nesbitt-Smith | cns.me