Kubernetes and Cloud Native

NCKU SA/NA

HungWei Chiu/Open Networking Foundation

Outline

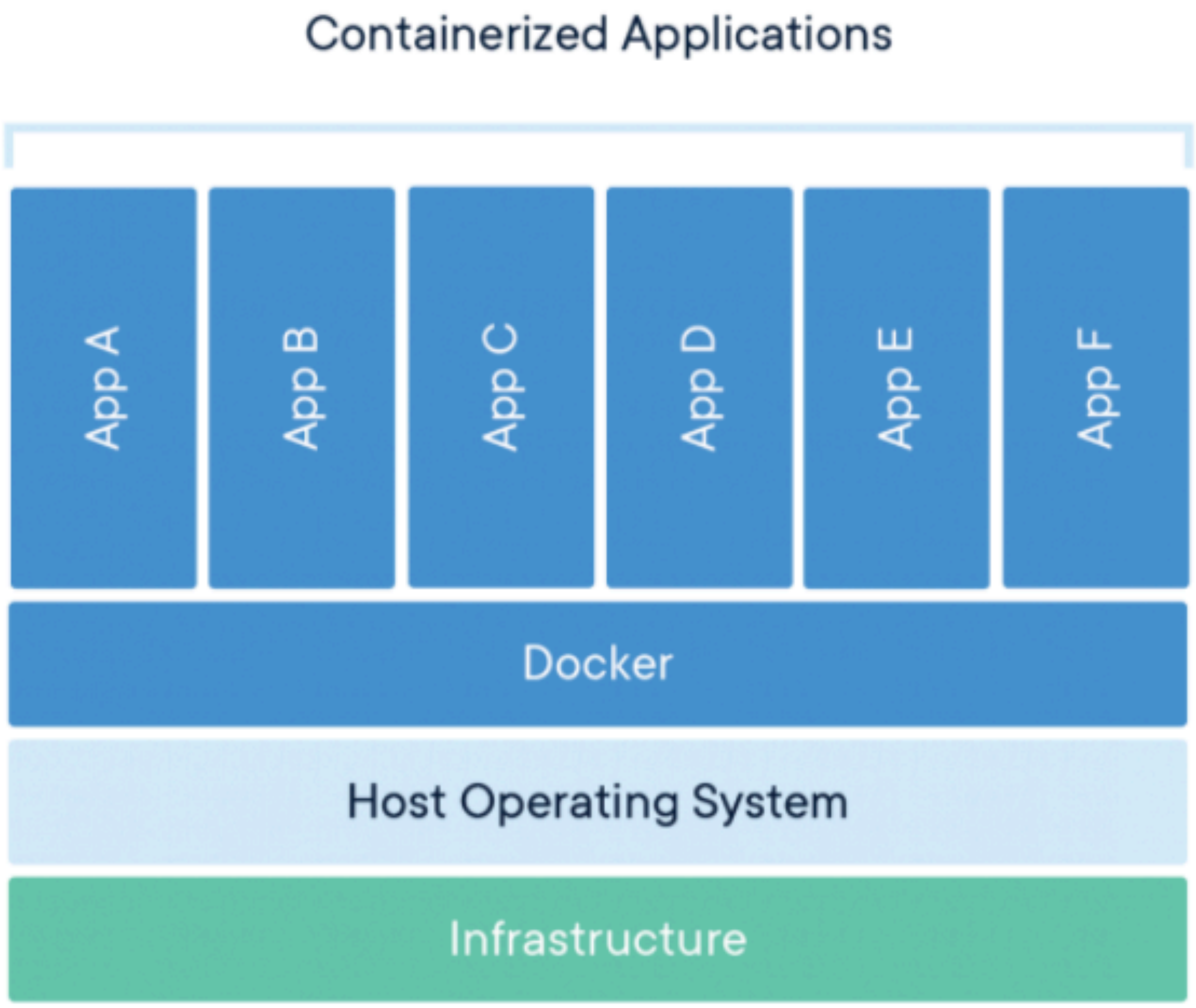
- Container
- Kubernetes
    - Why We Need
- Cloud Native World
- Experience

Who I Am

- HungWei Chiu (hwchiu)
- Open Networking Foundation
  - SDN
- Focus on
  - Networking, Linux, CloudNative, Container/Kubernetes
- From NCTU-CSCC
  - NA/SA

Container

- Virtualization
  - Virtual Machine, Containers
- Docker



https://www.docker.com/resources/what-container

Other Implementations

- Rkt
- LXC
- Containerd
- CRI-O
- SIngularity
- Podman
- ... etc
-

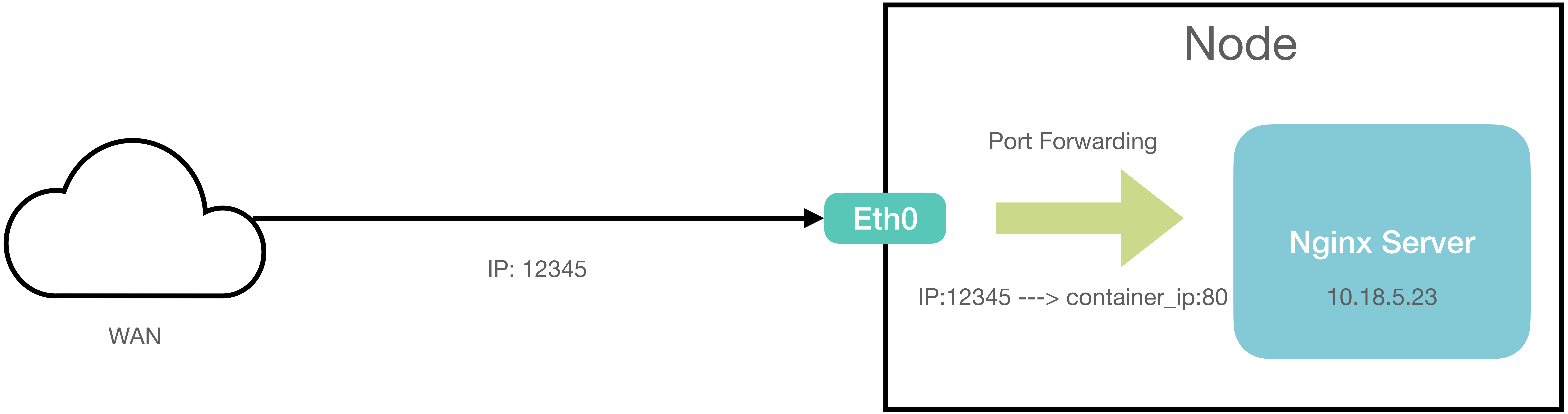
https://www.docker.com/resources/what-container

How We Use Containers

- Single application
  - Prepare the docker image
  - Use the docker run to run the container
    - Volume(Option)
    - Port-forwarding (Option)

https://www.docker.com/resources/what-container

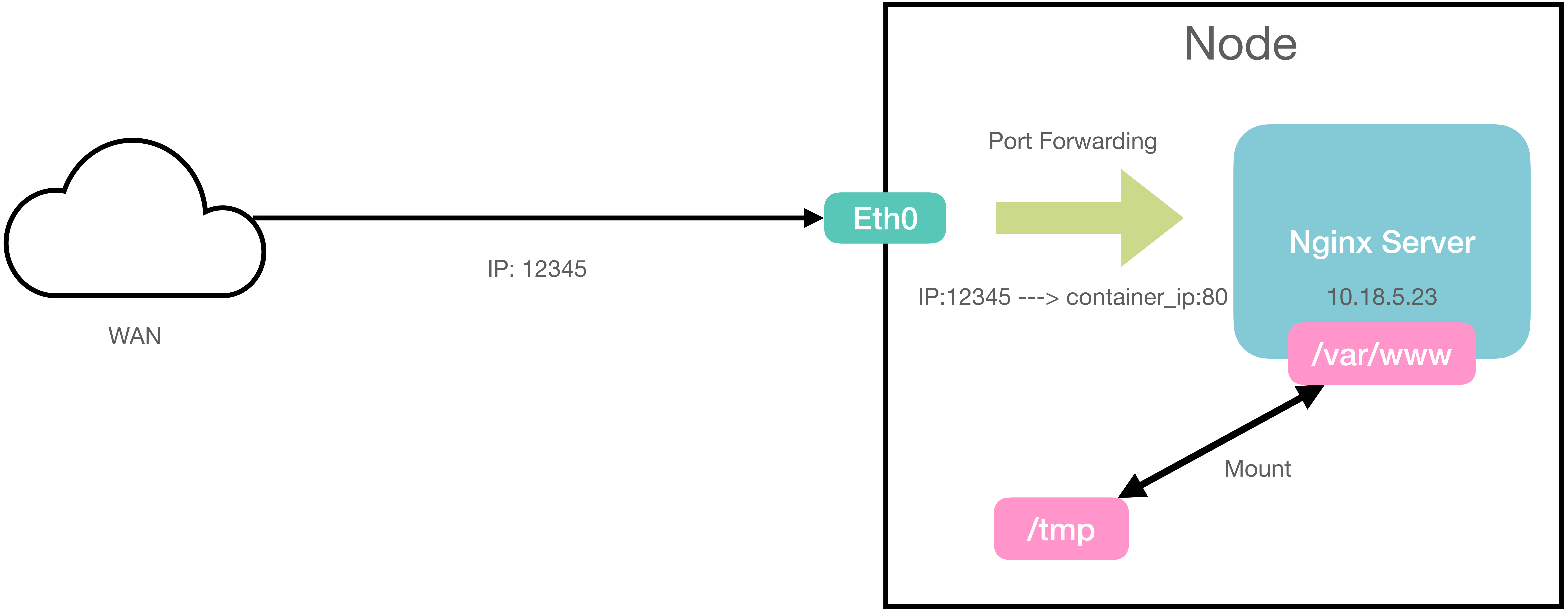
Single Application



```
$docker run -p 12345:80 nginx
```

https://www.docker.com/resources/what-container

Single Application



$docker run -p 12345:80 -v /tmp:/var/www nginx

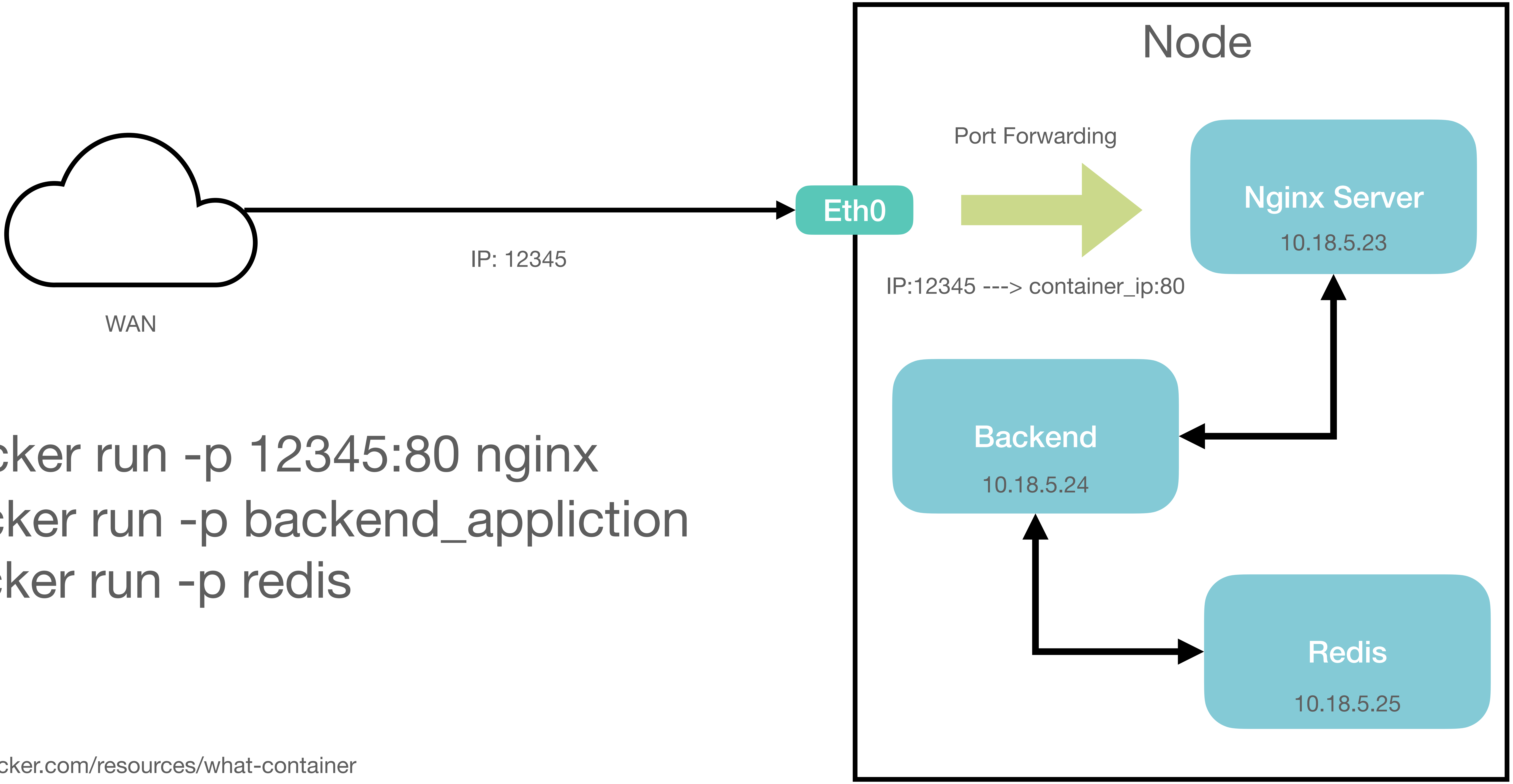
https://www.docker.com/resources/what-container

How We Use Containers

- Multiple applications
  - Front/Back ends
  - Database, Cache...etc
- Network Topology
  - Front-end has to access the back-end
  - Back-end has to access the database

https://www.docker.com/resources/what-container

Multiple Applications



```
$docker run -p 12345:80 nginx
$docker run -p backend_appliction
$docker run -p redis
```

https://www.docker.com/resources/what-container

Multiple Applications

- How to access other container?
  - Container's IP isn't fixed, and will be changed when you restart it.
  - How does client aware of the IP change?
- How to manage containers life-cycles?
  - Start/Stop all containers.
  - Start/Stop part of containers.

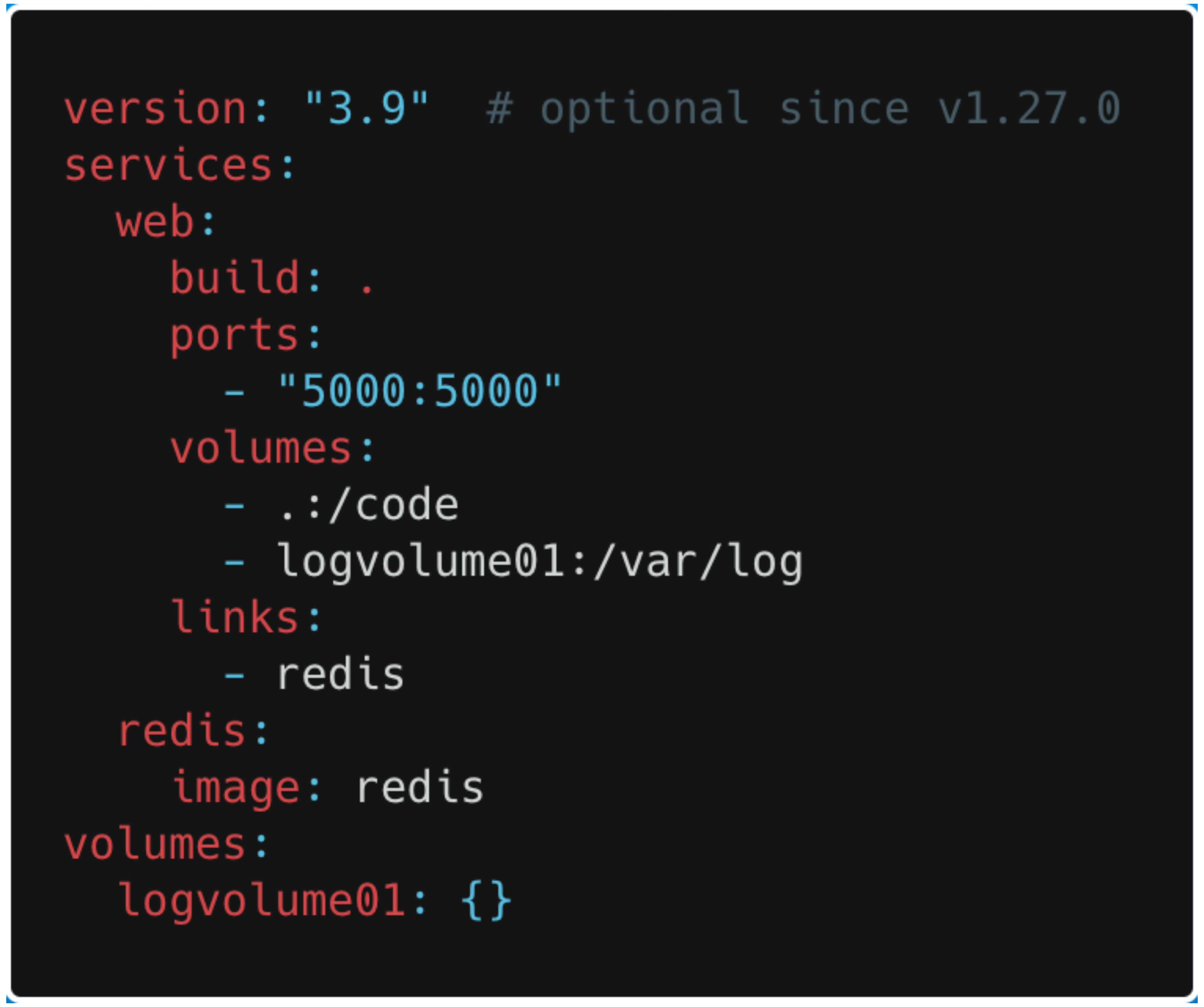
https://www.docker.com/resources/what-container

How We Use Containers

- Docker supports a mechanism called docker-compose to address there issues.
- Compose is a tool for defining and running multi-container applications.
    - You use a YAML file to configure your application's service.
    - Manage all services with a single command.

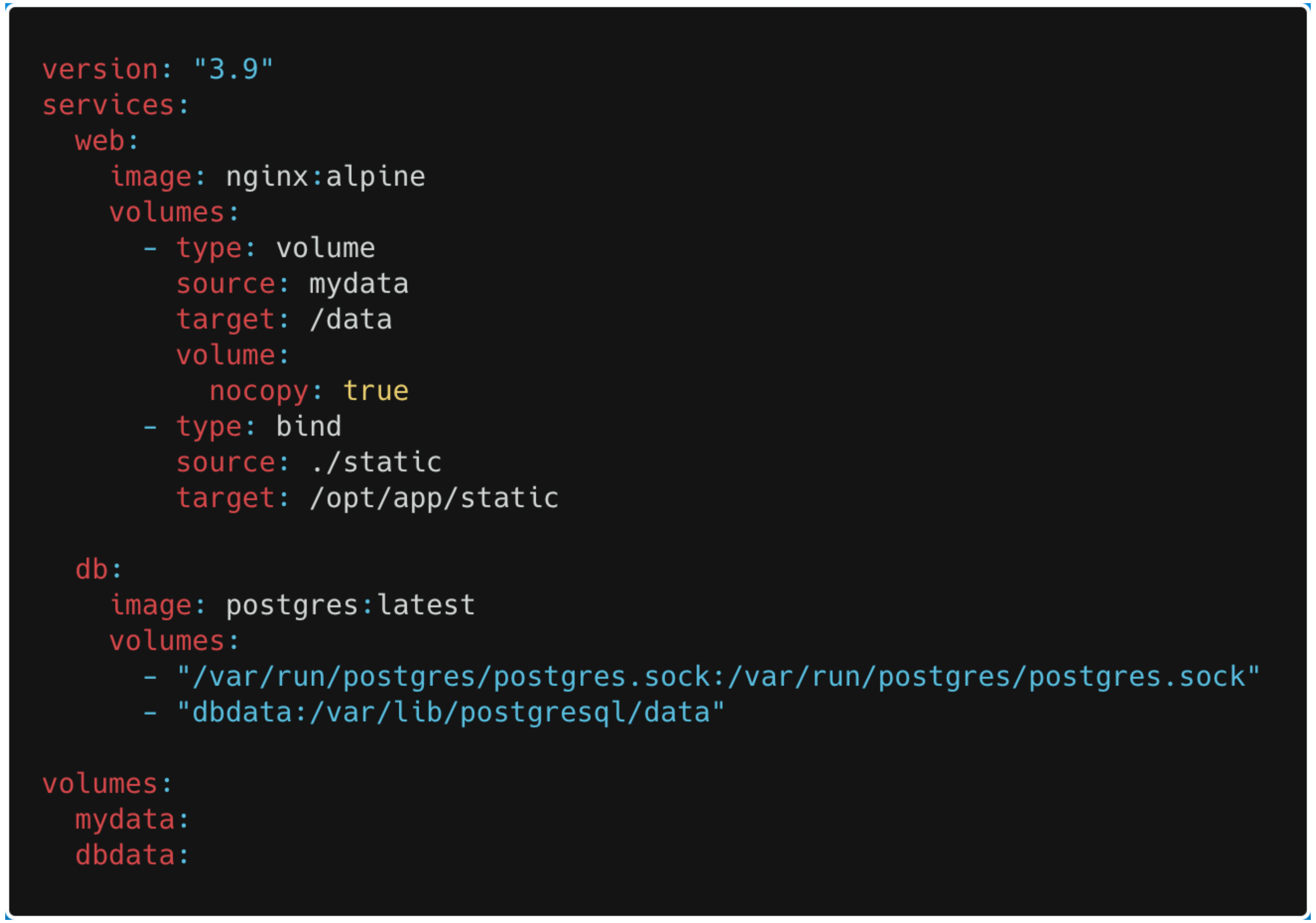
https://www.docker.com/resources/what-container

How We Use Containers



```
version: "3.9"  # optional since v1.27.0
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Use DNS to access container instead of IP address



```
version: "3.9"
services:
  web:
    image: nginx:alpine
    volumes:
      - type: volume
        source: mydata
        target: /data
        volume:
          nocopy: true
      - type: bind
        source: ./static
        target: /opt/app/static

  db:
    image: postgres:latest
    volumes:
      - "/var/run/postgres/postgres.sock:/var/run/postgres/postgres.sock"
      - "dbdata:/var/lib/postgresql/data"

volumes:
  mydata:
  dbdata:
```

Share volumes within containers

Multiple Applications

Node
WAN
IP: 12345
Eth0
Port Forwarding
IP:12345 ---> container_ip:80
backend:1234
Nginx Server
10.18.5.23
Backend
10.18.5.24
Redis
10.18.5.25
redis:2345

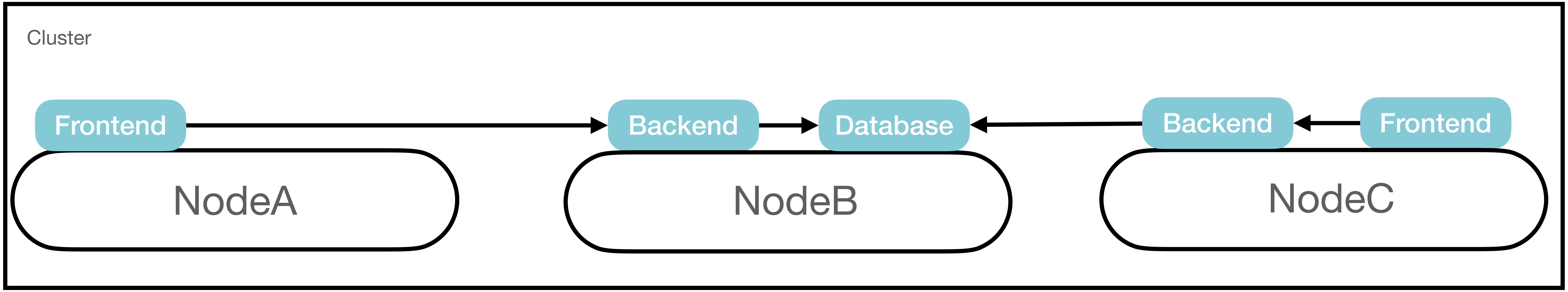
```
$ docker-compose up -d
$ docker-compose down
```

Issue?

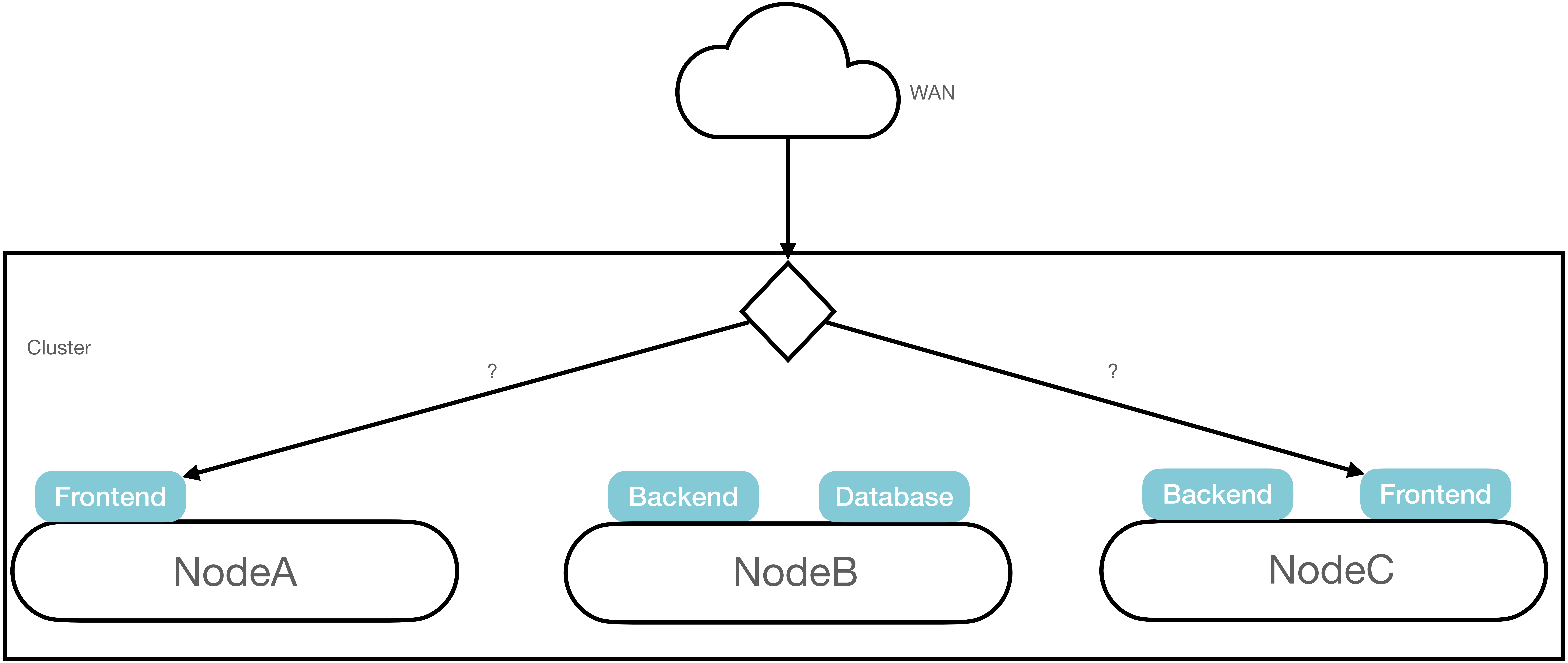
- Single Node
  - Single application
  - Multiple applications
- How about multiple nodes?
  - Compute
  - Network
  - Storage

Multiple Nodes(Cluster)

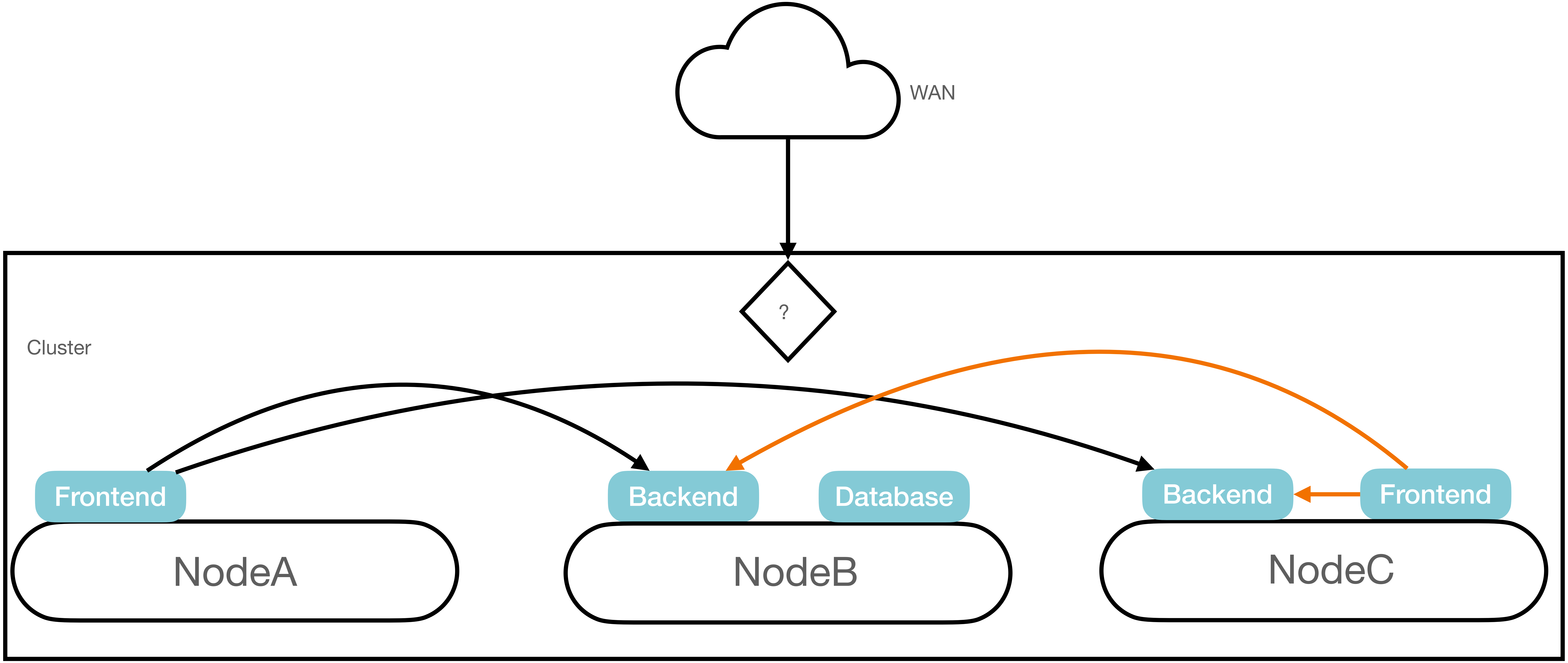
- High Availability
- Load Balancing
- No single point failure



Multiple Nodes(Cluster)

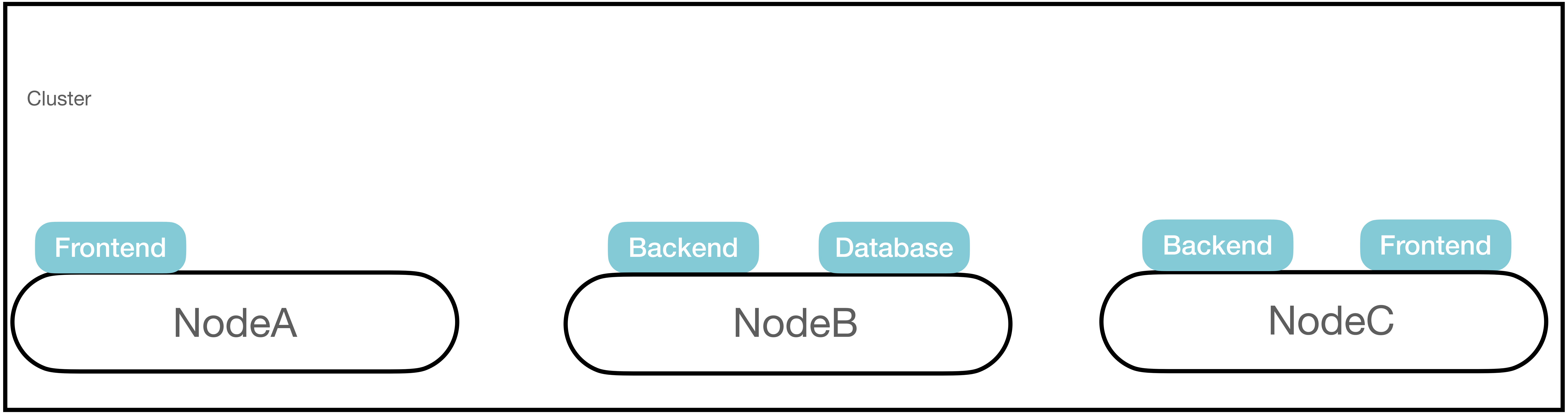


Network Access



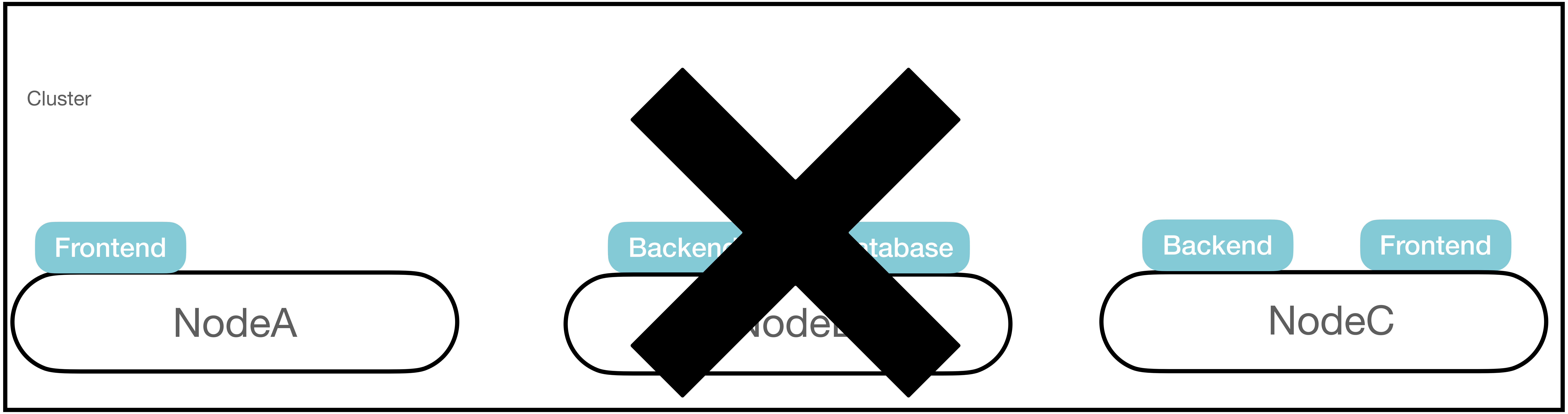
Container Reschedule

- Reschedule running containers to other nodes when node is offline



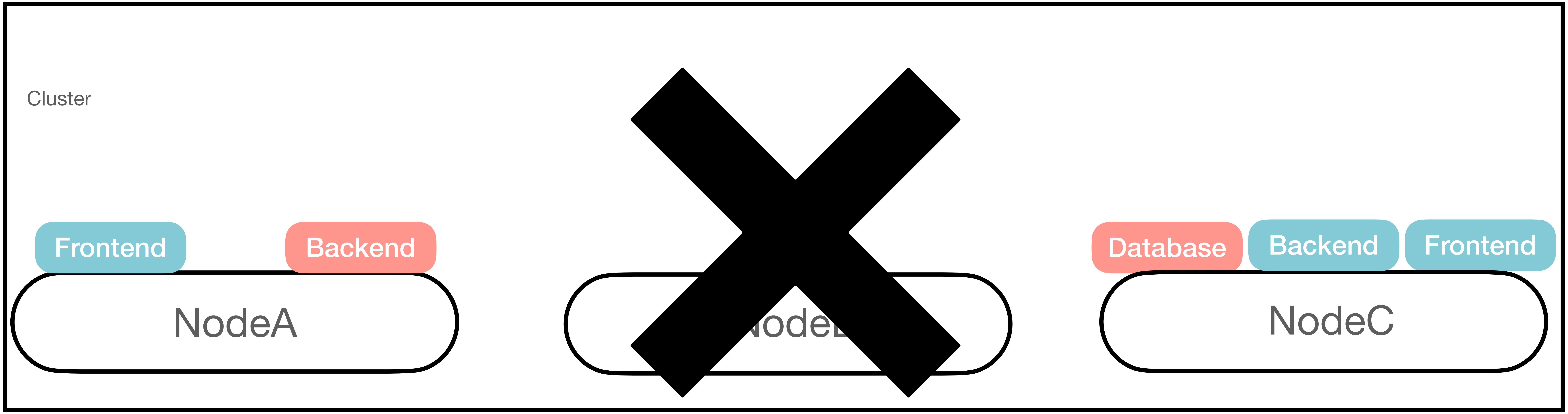
Container Reschedule

- Reschedule running containers to other nodes when node is offline



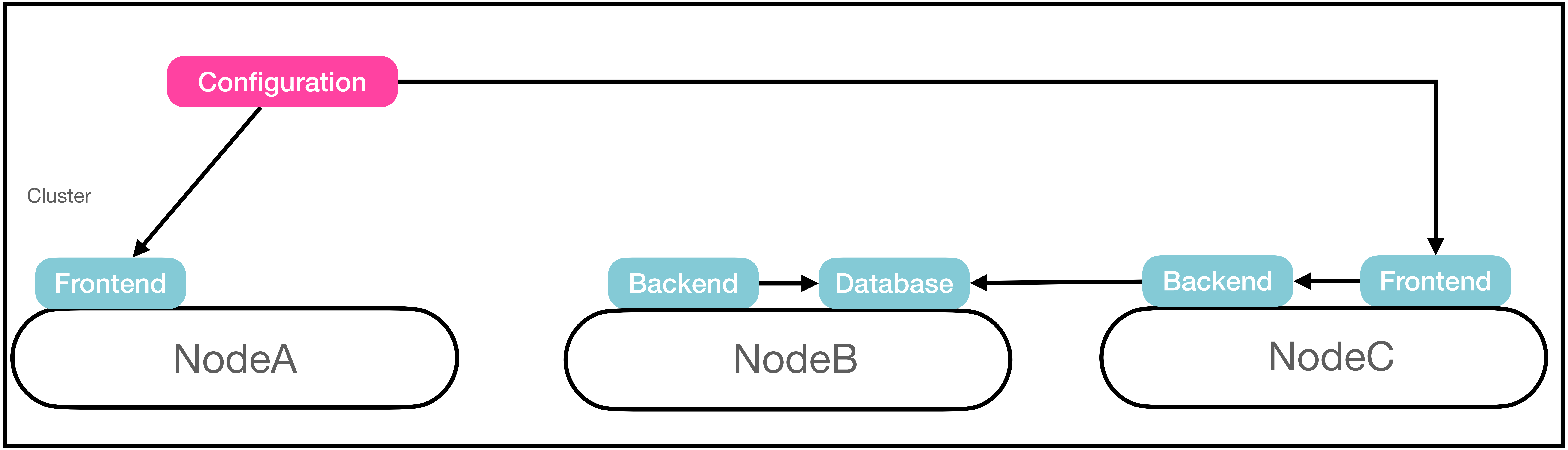
Container Reschedule

- Reschedule running containers to other nodes when node is offline



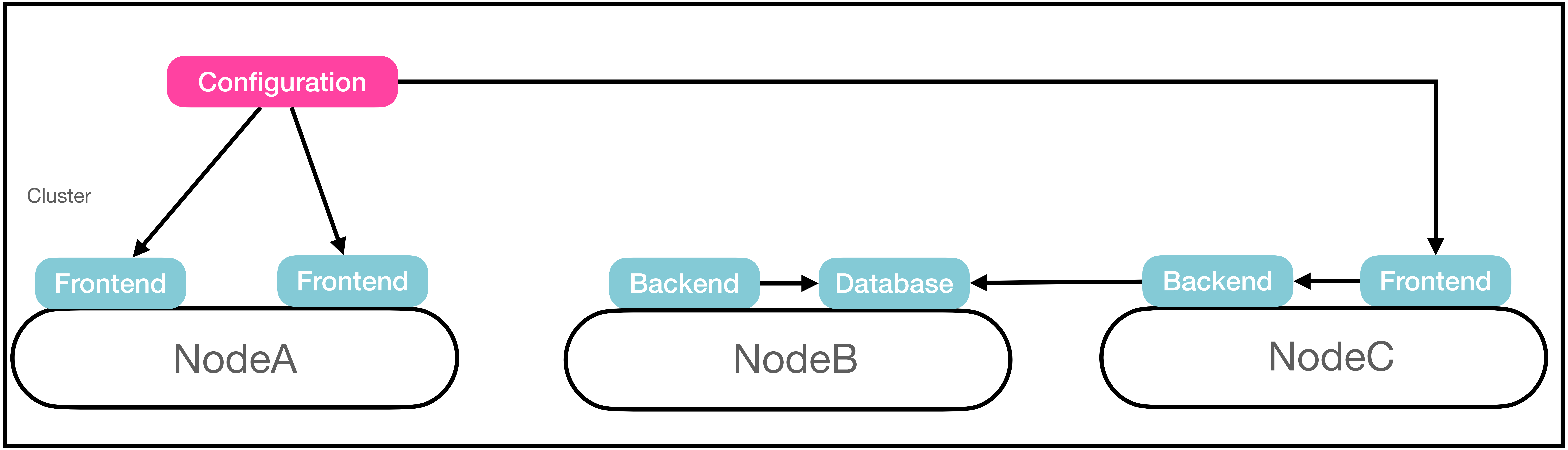
Storage/Configuration

- Configuration for same containers



Storage/Configuration

- Configuration for same containers



Solutions

- DIY with docker and docker-compose
- docker-swam
- Kubernetes
- ...
- etc

kubernetes

https://kubernetes.io/docs/home/

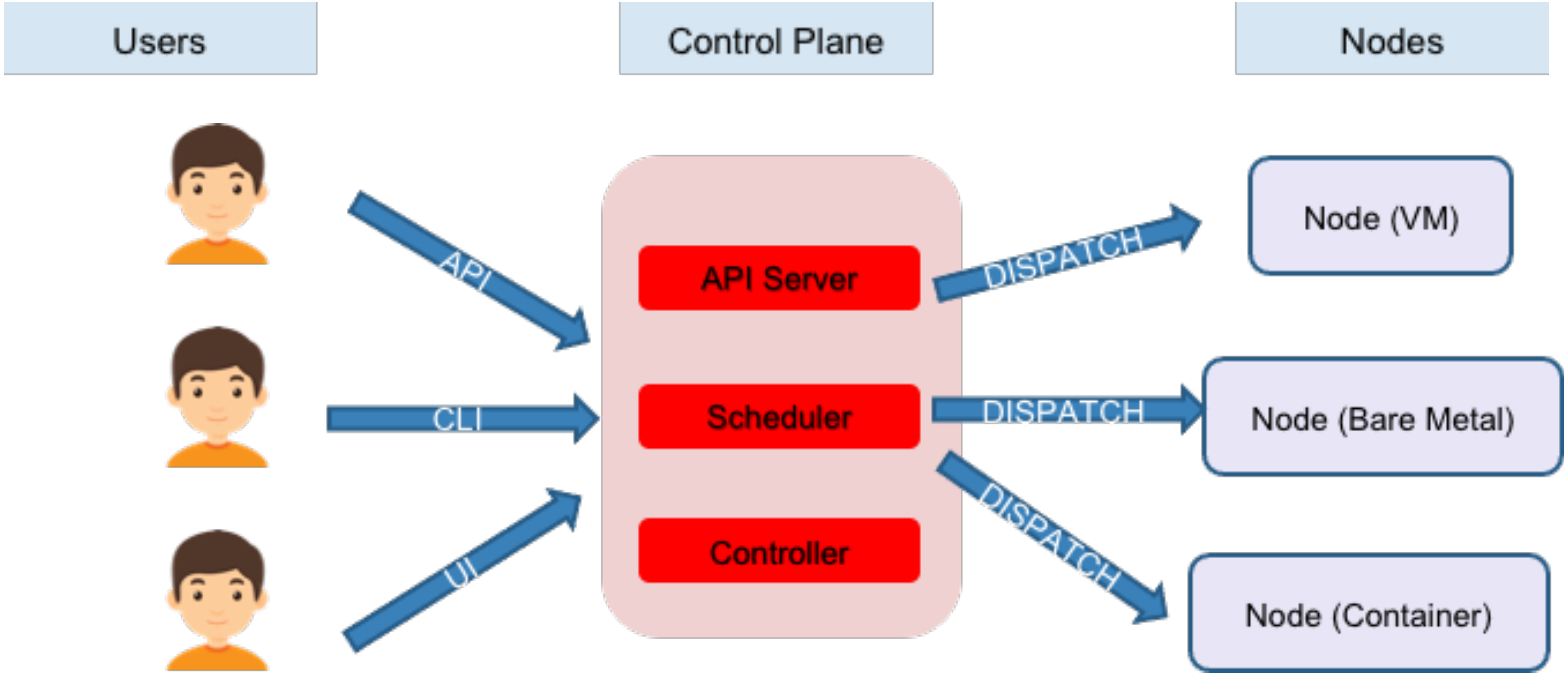
Kubernetes is becoming the Linux of the cloud

Jim Zemlin, Linux Foundation

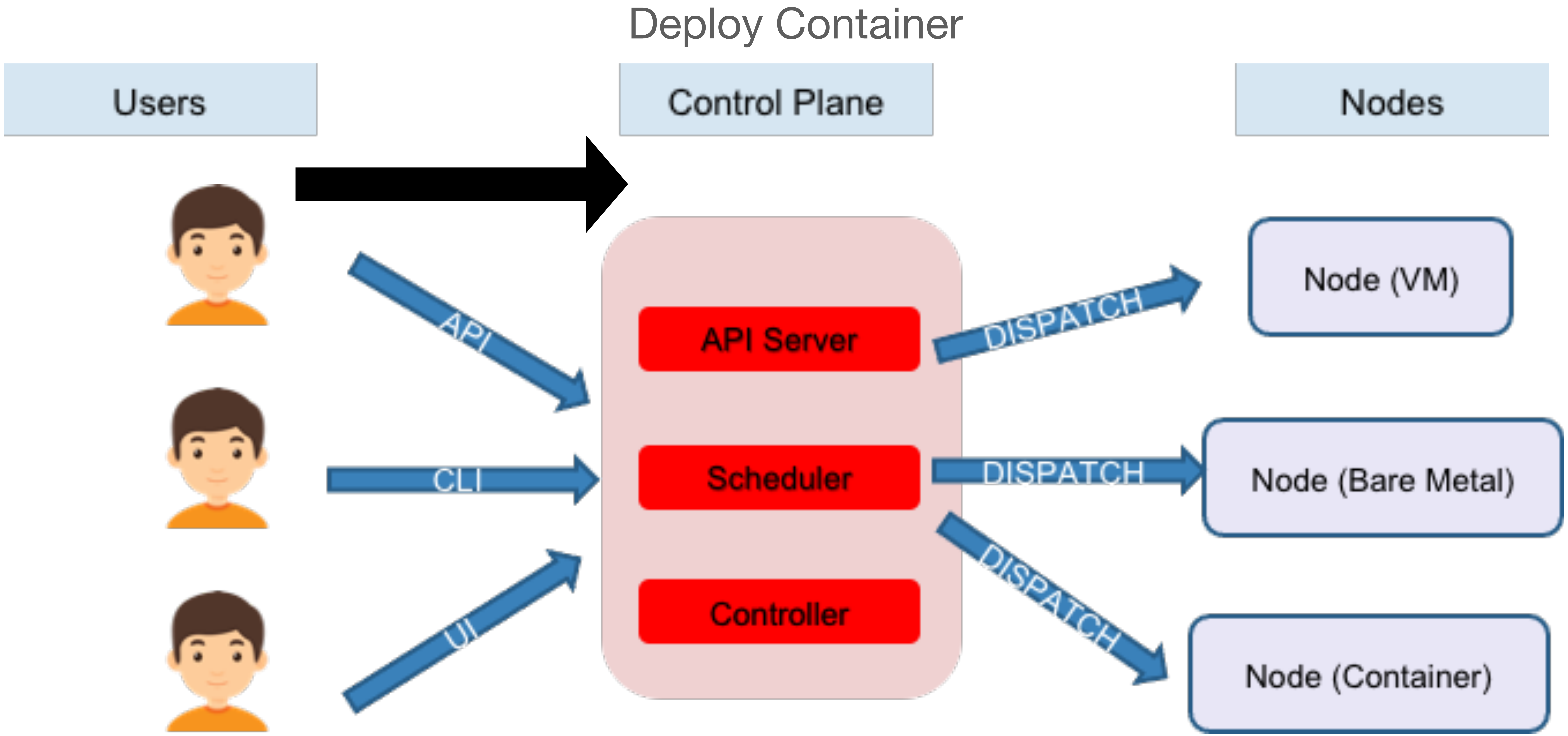
Before Kubernetes

- Google has been running containerized workloads in production
  - Virtually everything runs as a Container
- Borg: One of the predecessors to Kubernetes
  - Internal container-oriented cluster
  - Concept
    - Services/Pod/Label

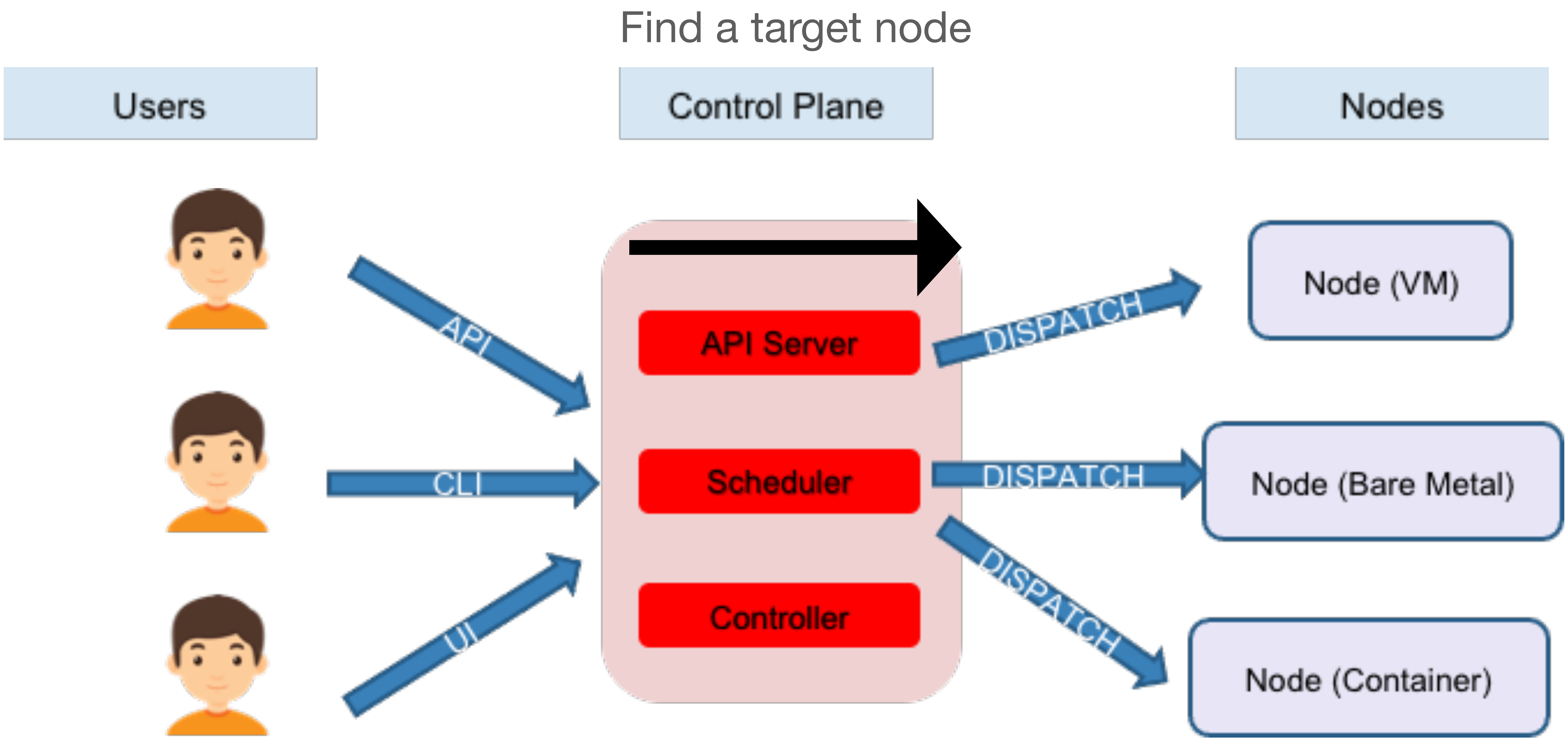
Kubernetes Architecture



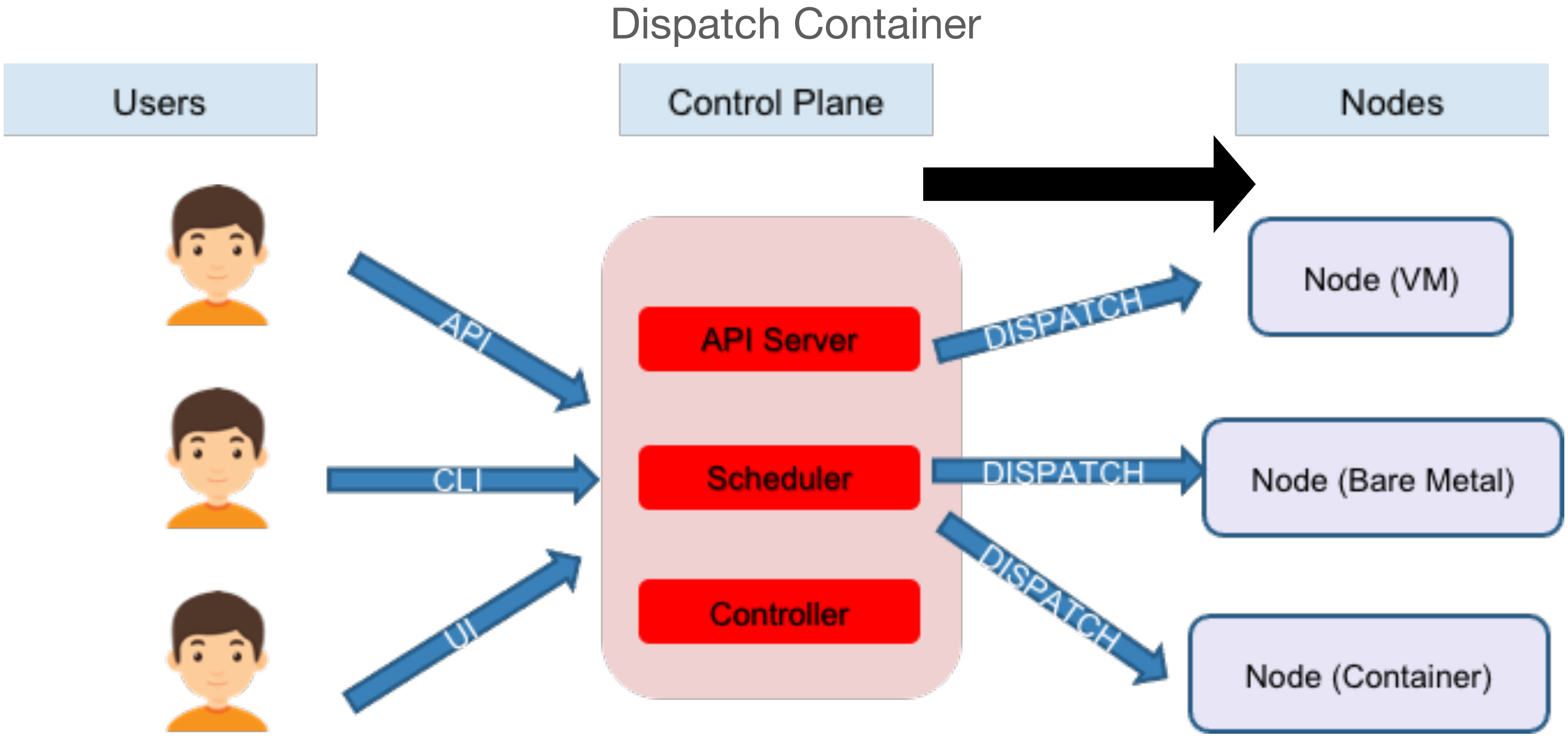
Kubernetes Architecture



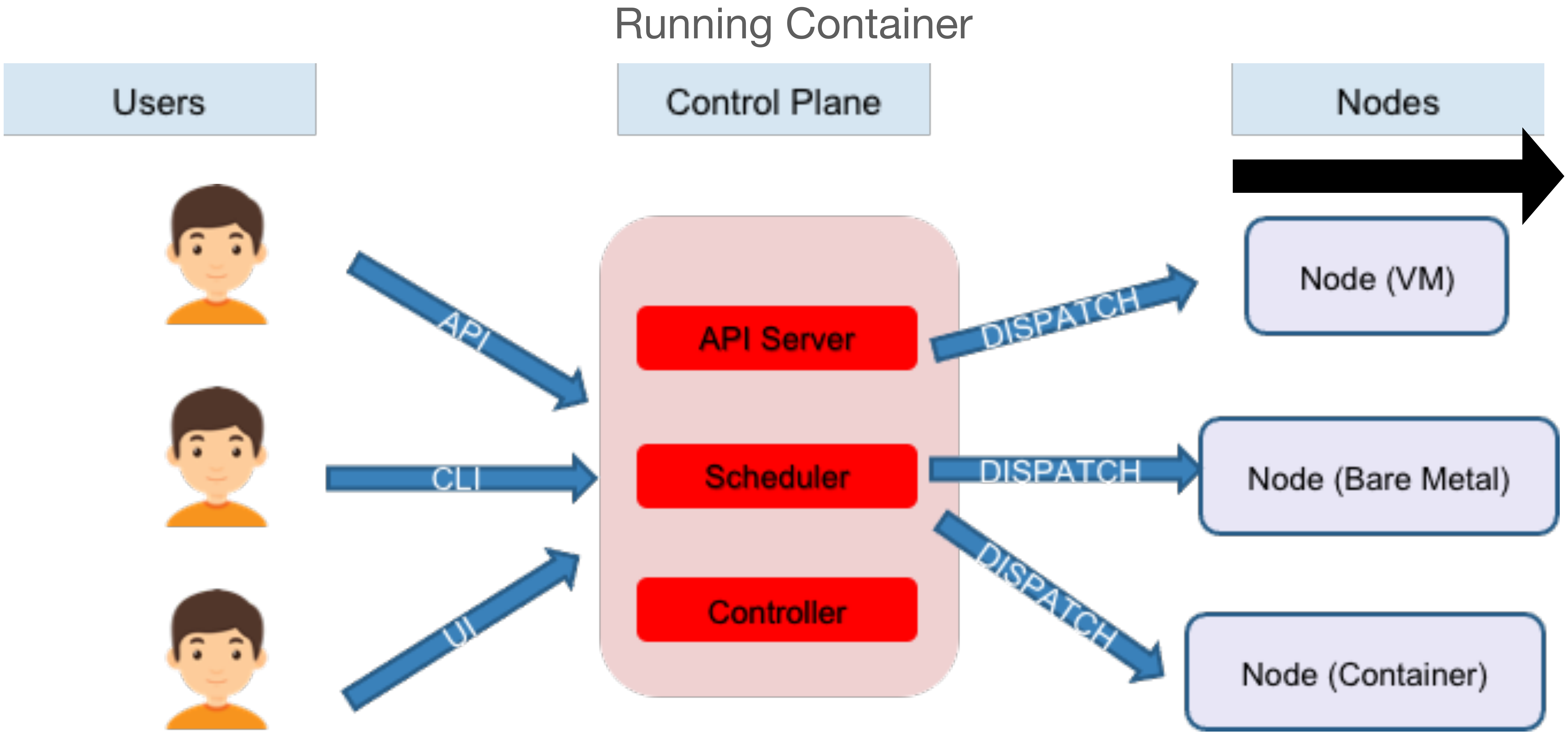
Kubernetes Architecture



Kubernetes Architecture



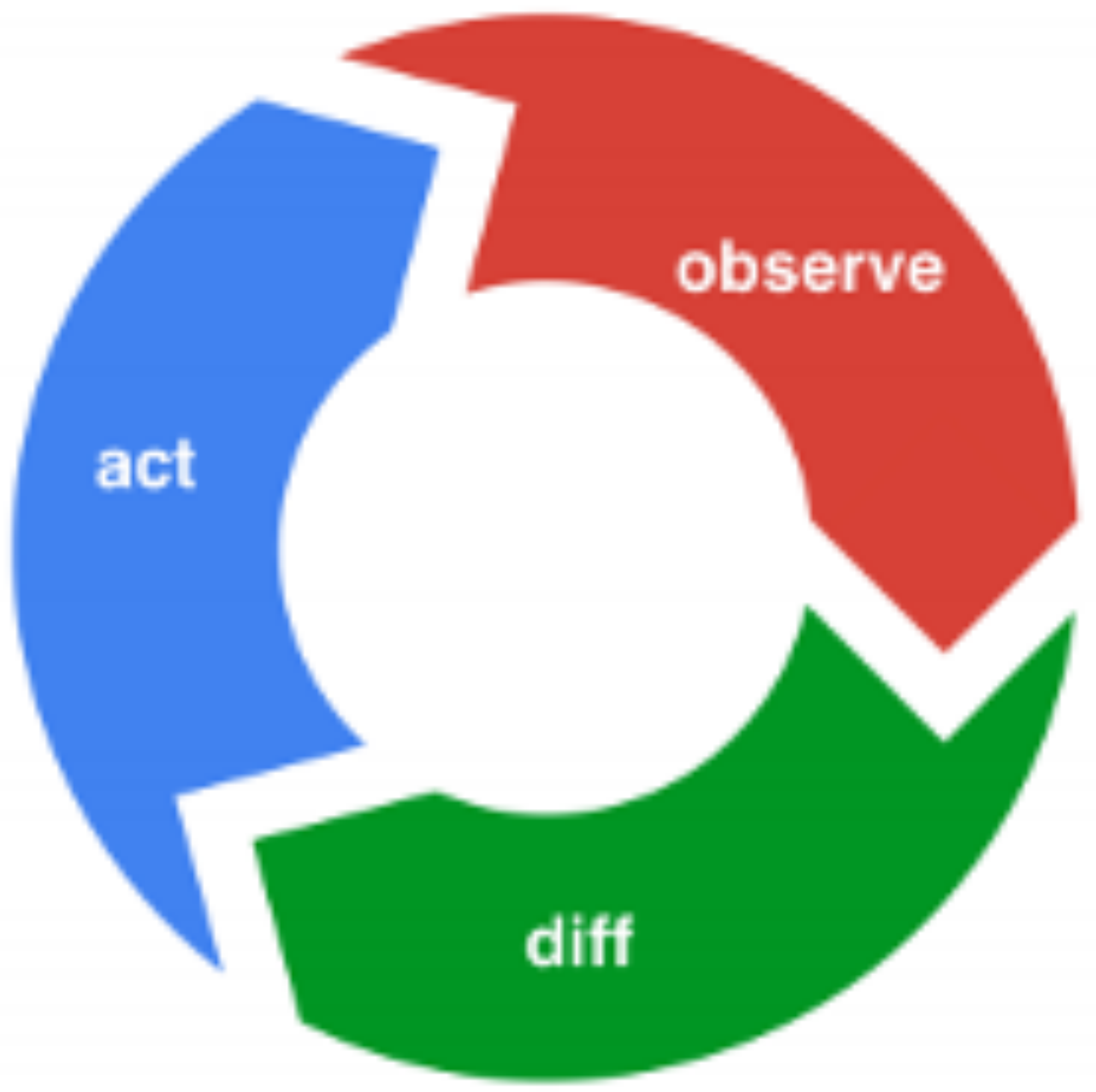
Kubernetes Architecture



Control Plane
Like Docker Daemon

Control Plane

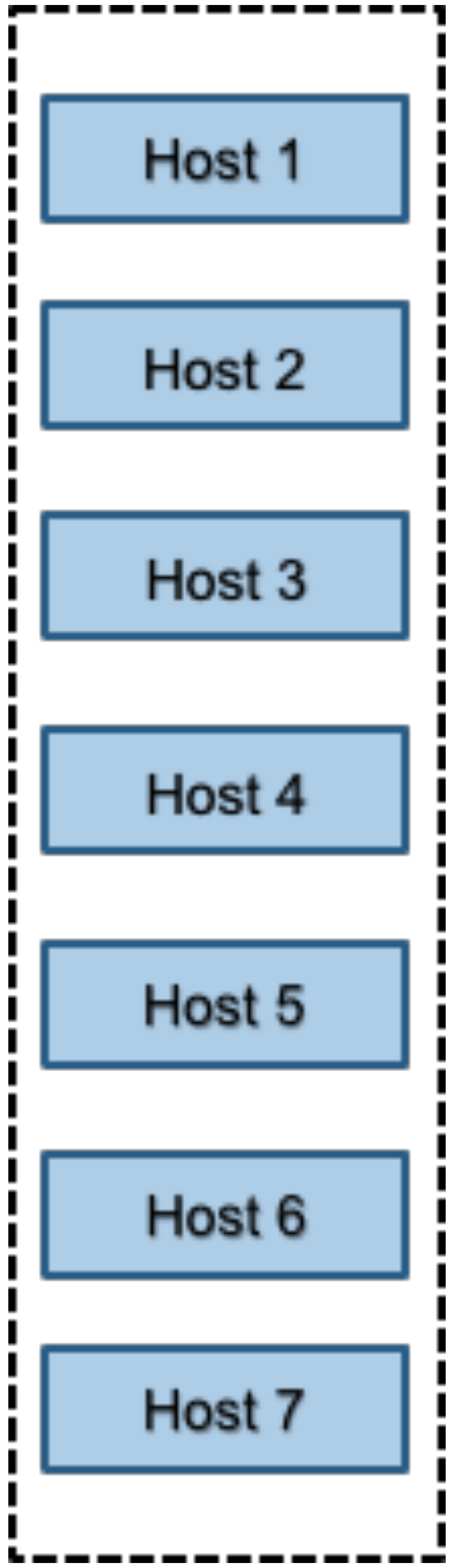
- API Server
  - Validates and configures data for the API object
  - Services REST operations
- Controller
  - Control loop that watches the shared state of the cluster
  - Make changes attempting to move the current state



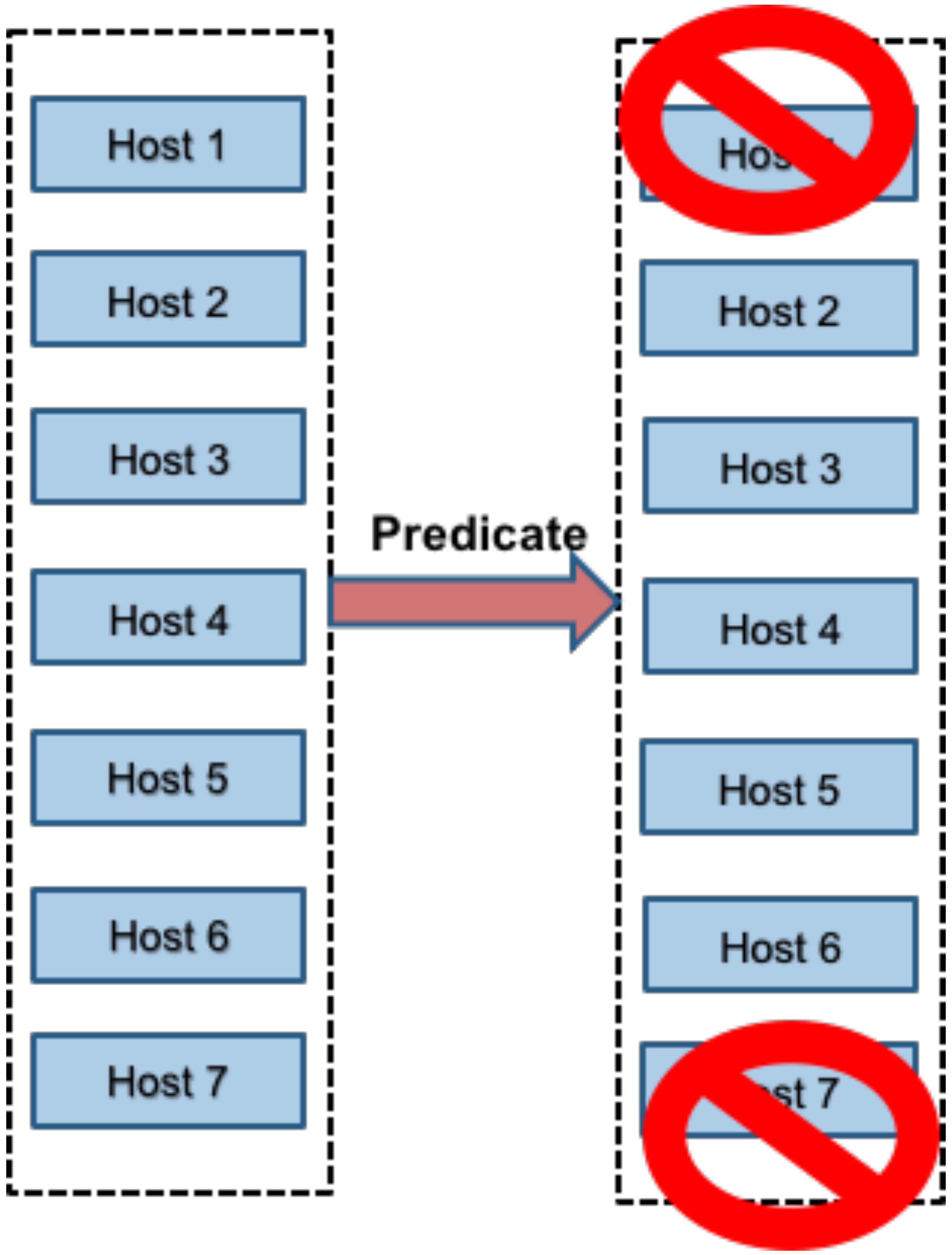
Control Plane

- Scheduler
  - Watches newly created pods and select a node to run on
- Decisions
  - Resource requirements
  - Hardware/Software/Policy constraints
  - Affinity/Anti-Affinity

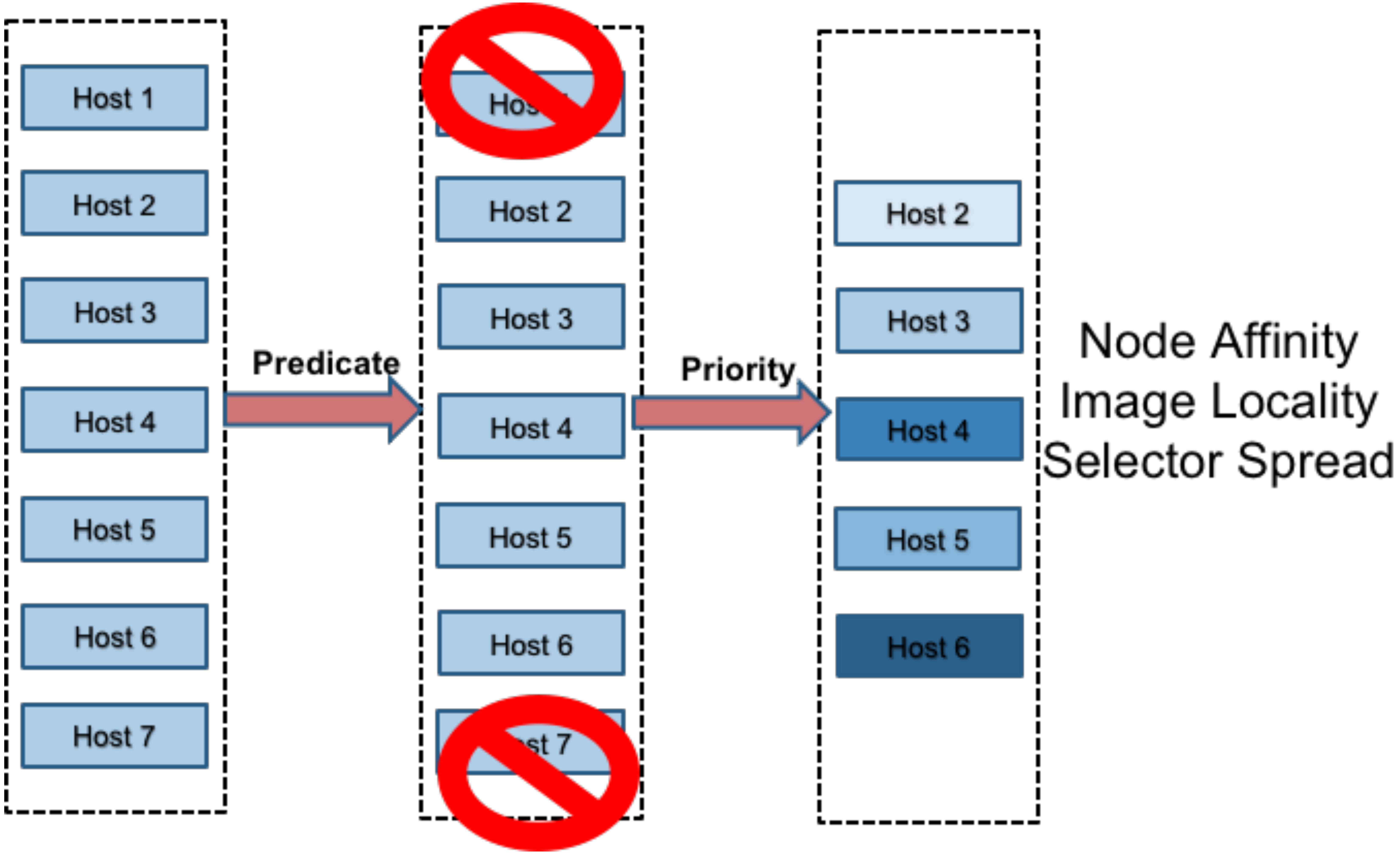
Kubernetes Scheduler



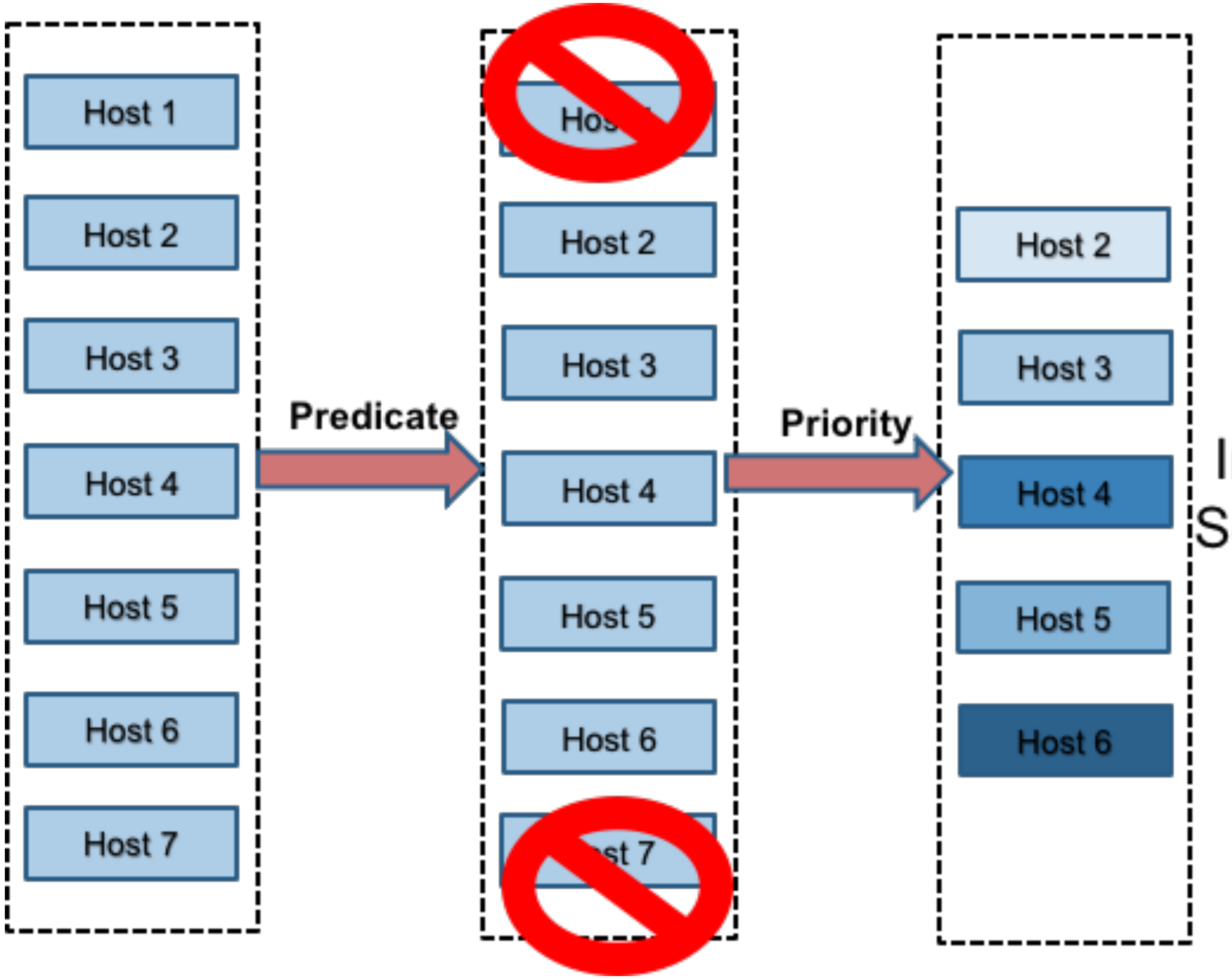
Kubernetes Scheduler



Kubernetes Scheduler



Kubernetes Scheduler



Host 1
Host 2
Host 3
Host 4
Host 5
Host 6
Host 7

Predicate

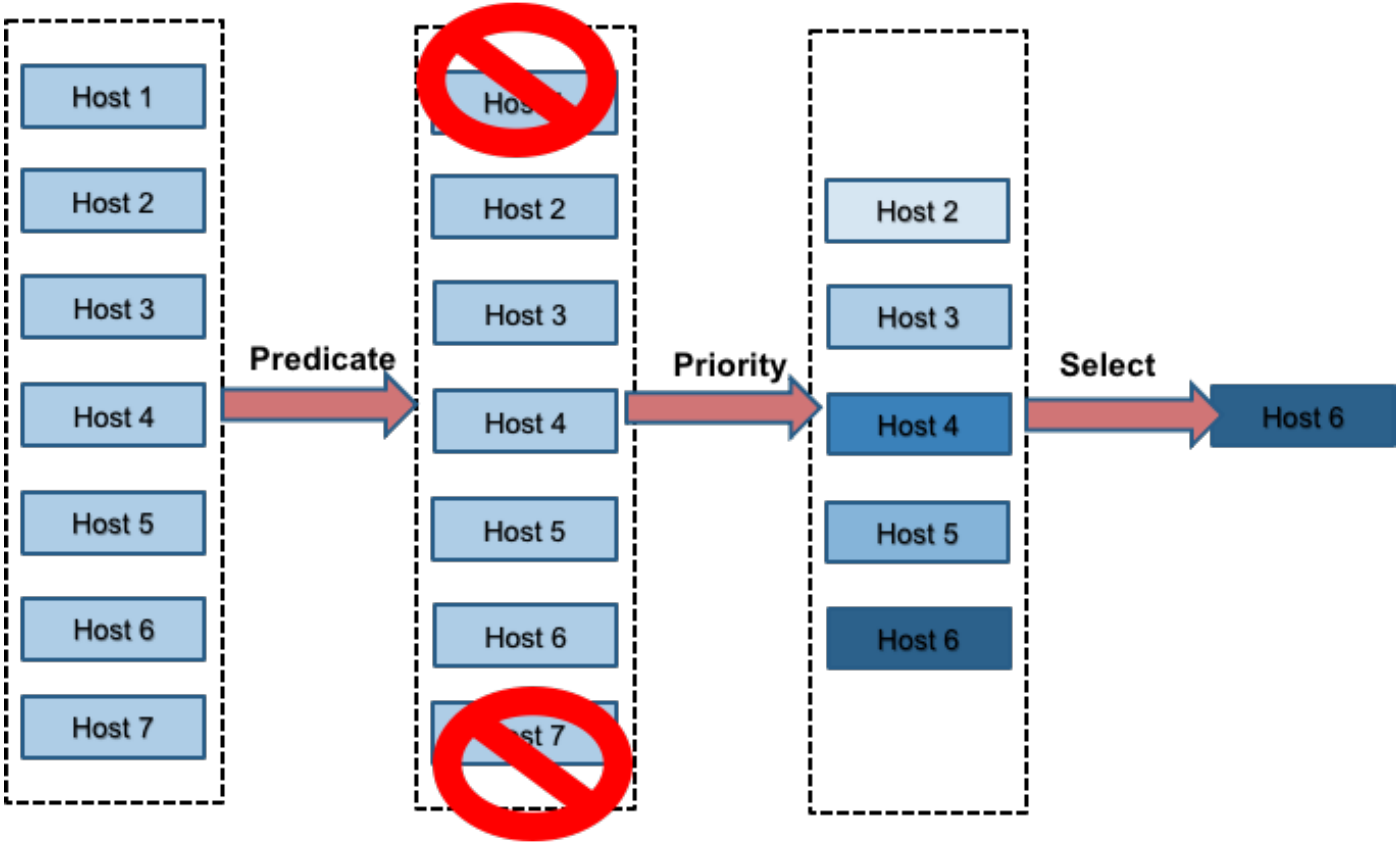
Hos
Host 2
Host 3
Host 4
Host 5
Host 6
st 7

Priority

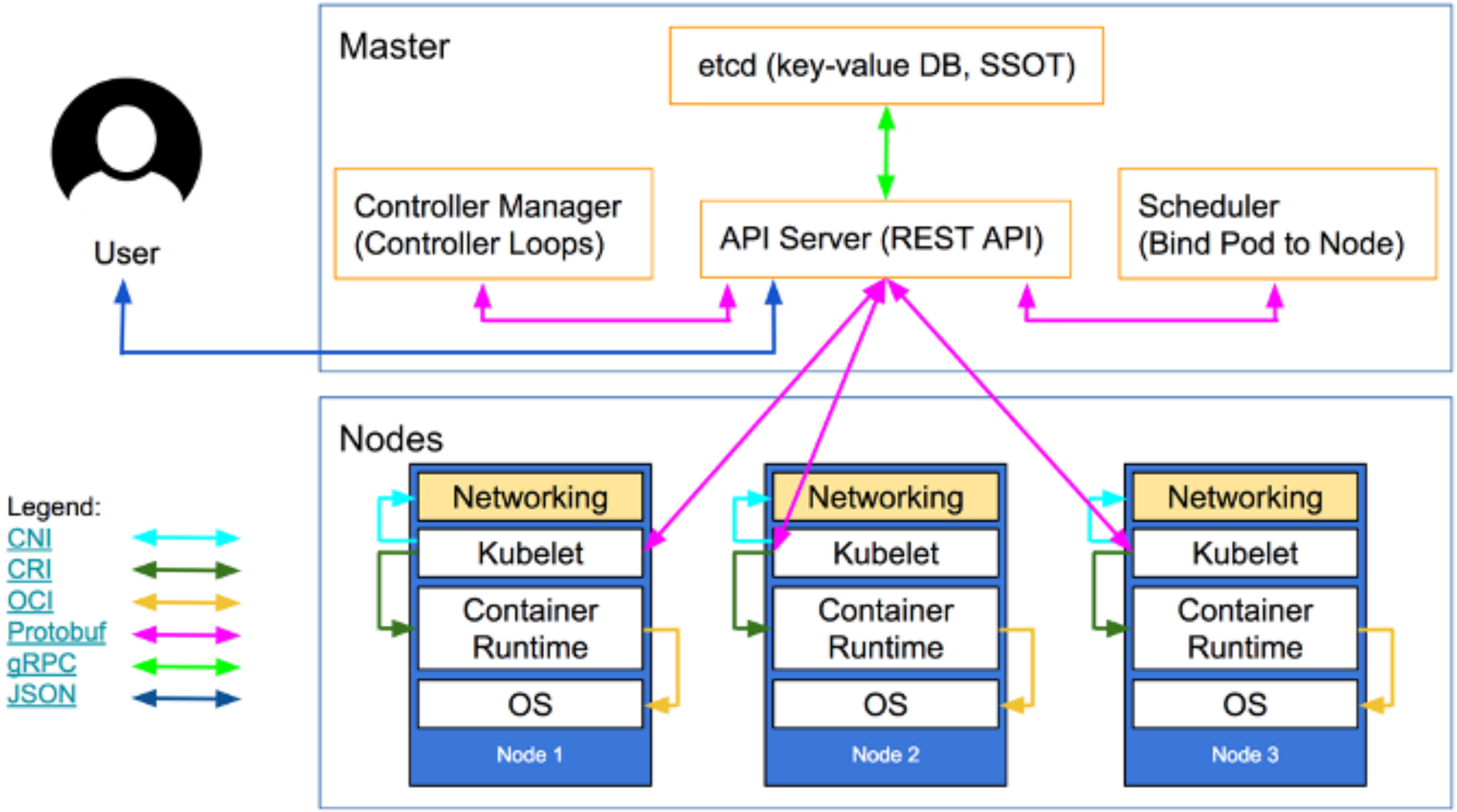
Host 2
Host 3
Host 4
Host 5
Host 6

Node Affinity
Image Locality
Selector Spread

Kubernetes Scheduler



Kubernetes Architecture



User

Master

etcd (key-value DB, SSOT)

Controller Manager
(Controller Loops)

API Server (REST API)

Scheduler
(Bind Pod to Node)

Nodes

Networking
Kubelet
Container Runtime
OS
Node 1

Networking
Kubelet
Container Runtime
OS
Node 2

Networking
Kubelet
Container Runtime
OS
Node 3

Legend:

CNI ⟷
CRI ⟷
OCI ⟷
Protobuf ⟷
gRPC ⟷
JSON ⟷

https://docs.google.com/presentation/d/1Gp-2blk5WExl_QR59EUZdwfO2BWLJqa626mK2ej-huo/edit#slide=id.g1e639c415b_0_56

Installation

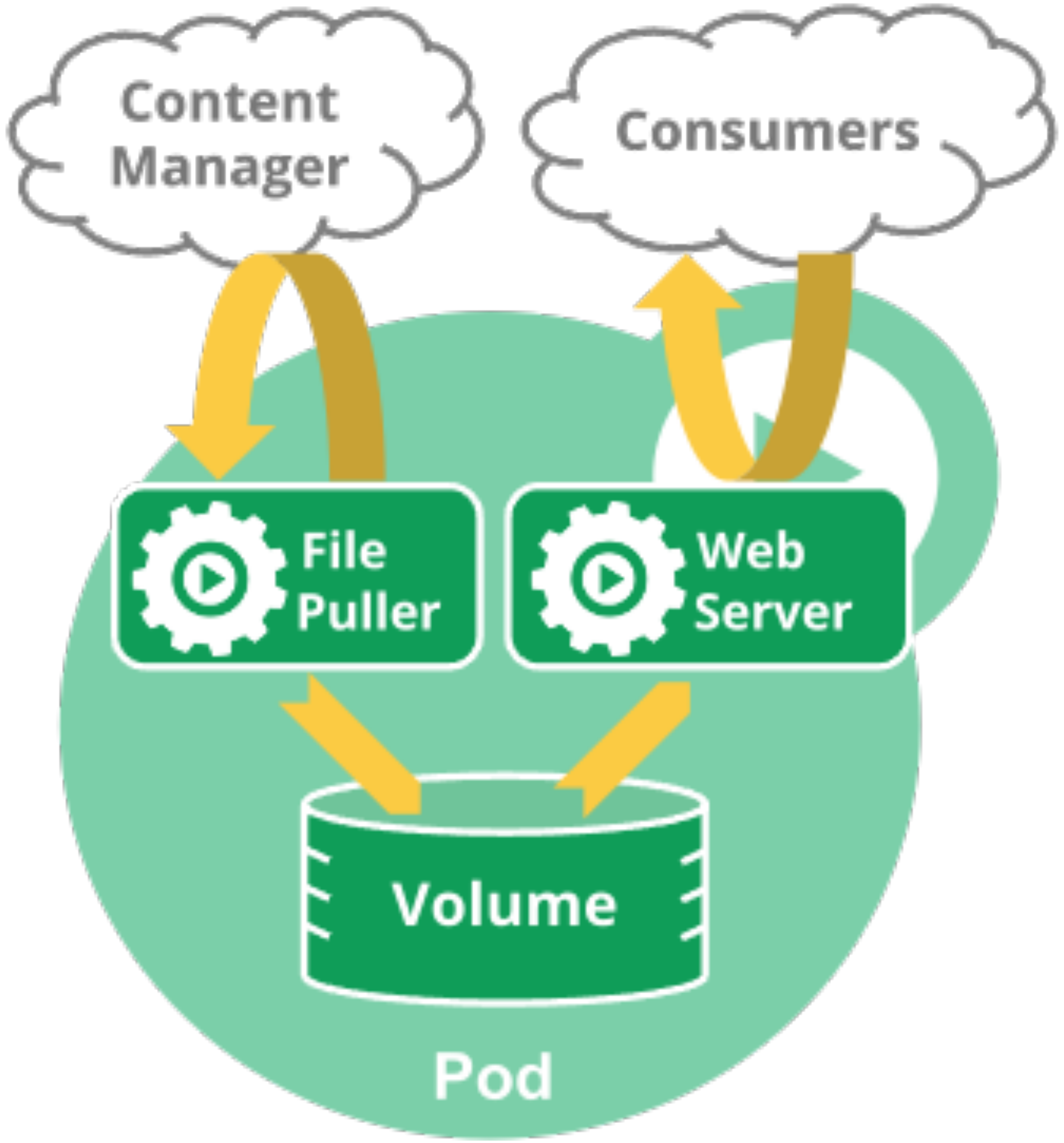
- Kubeadm (Demo)
- Minikube (Demo)
    - Can Setup the VM for you
- KIND/K3D (Demo)
    - Kubernetes in Docker
- Kubernetes Managed Service (Production)
    - AKS(Azure), GKE(Google), EKS(Amazon)
- Rancher (Production)
    - RKE, K3S

Core Primitives

- Workloads
  - Pod, Deployment, DaemonSet, Job, CronJob, Statefulset
- Network
  - Service, Ingress, NetworkPolicy, Endpoints
- Storage
  - StorageClass, PersistentVolume, PersistentVolumeClaim
- Configuration
  - ConfigMap, Secret
- Others
  - Label, CRD...etc

Pod

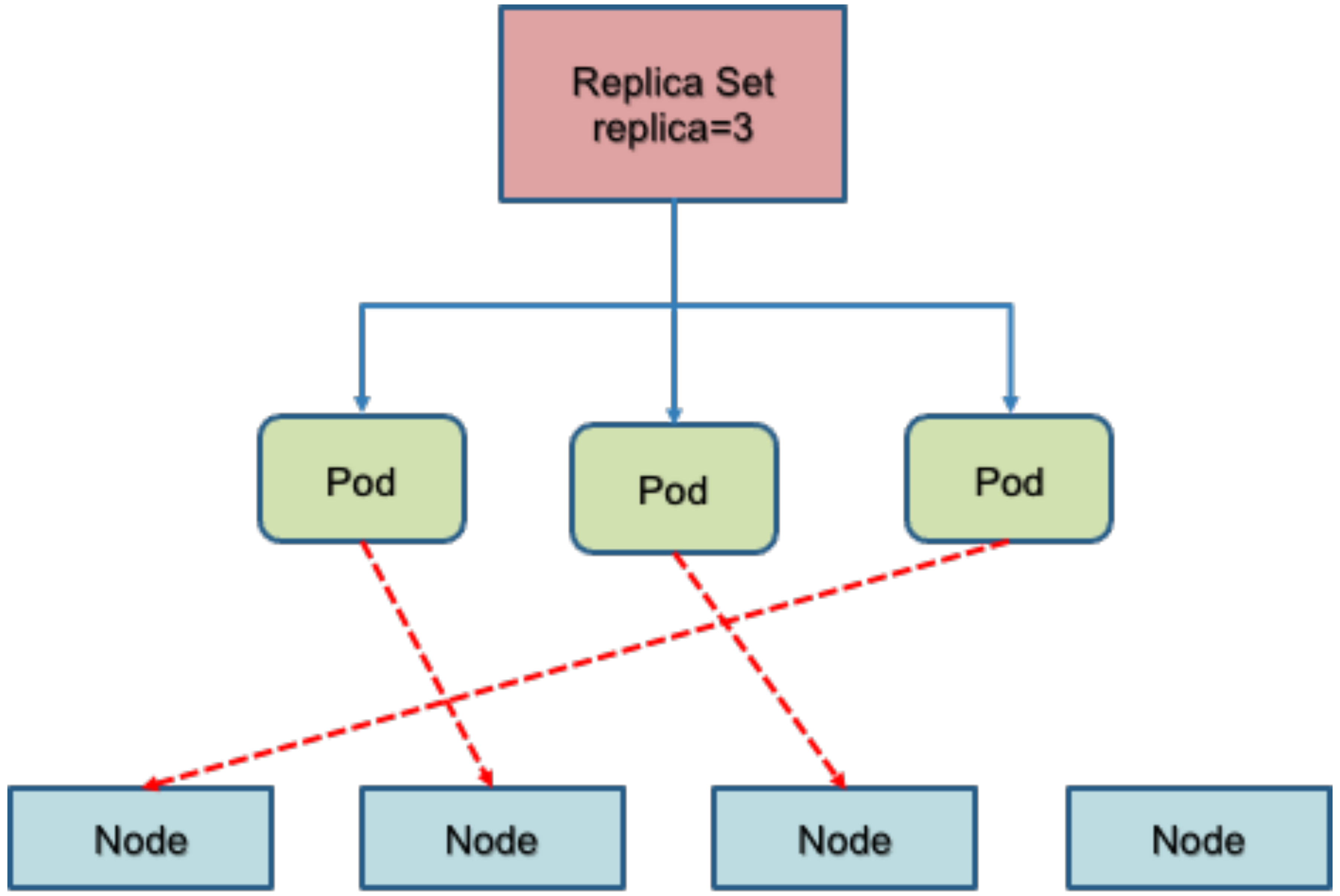
- A single instance of application in Kubernetes
- Group of containers
- Those containers shares
  - IP address
  - File system
  - Network namespace



https://kubernetes.io/docs/concepts/workloads/pods/pod/

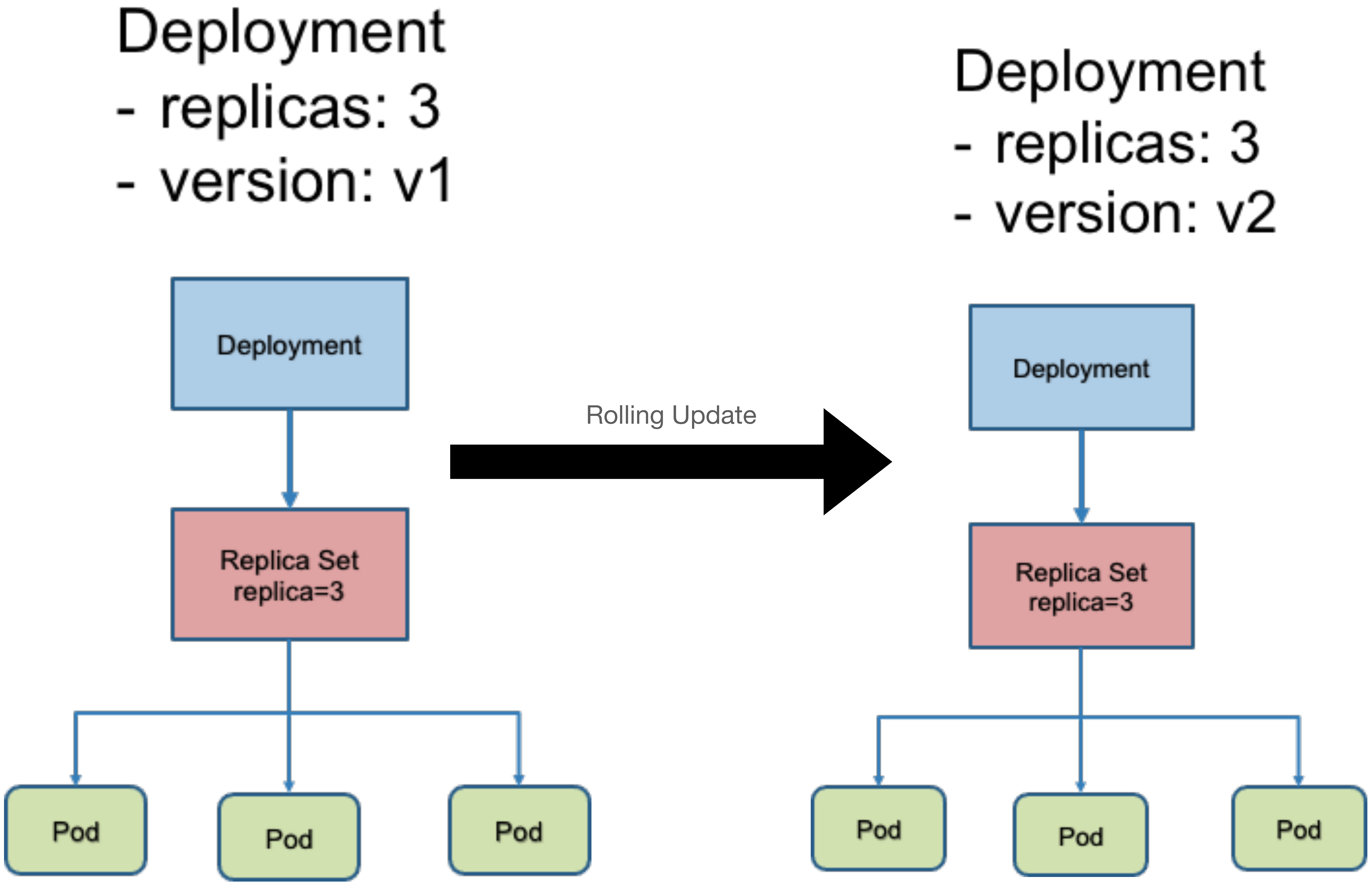
Replica Set

- Maintain a stable set of replica Pods running at any give time
- Guarantee the availability of a specified number of identical Pods



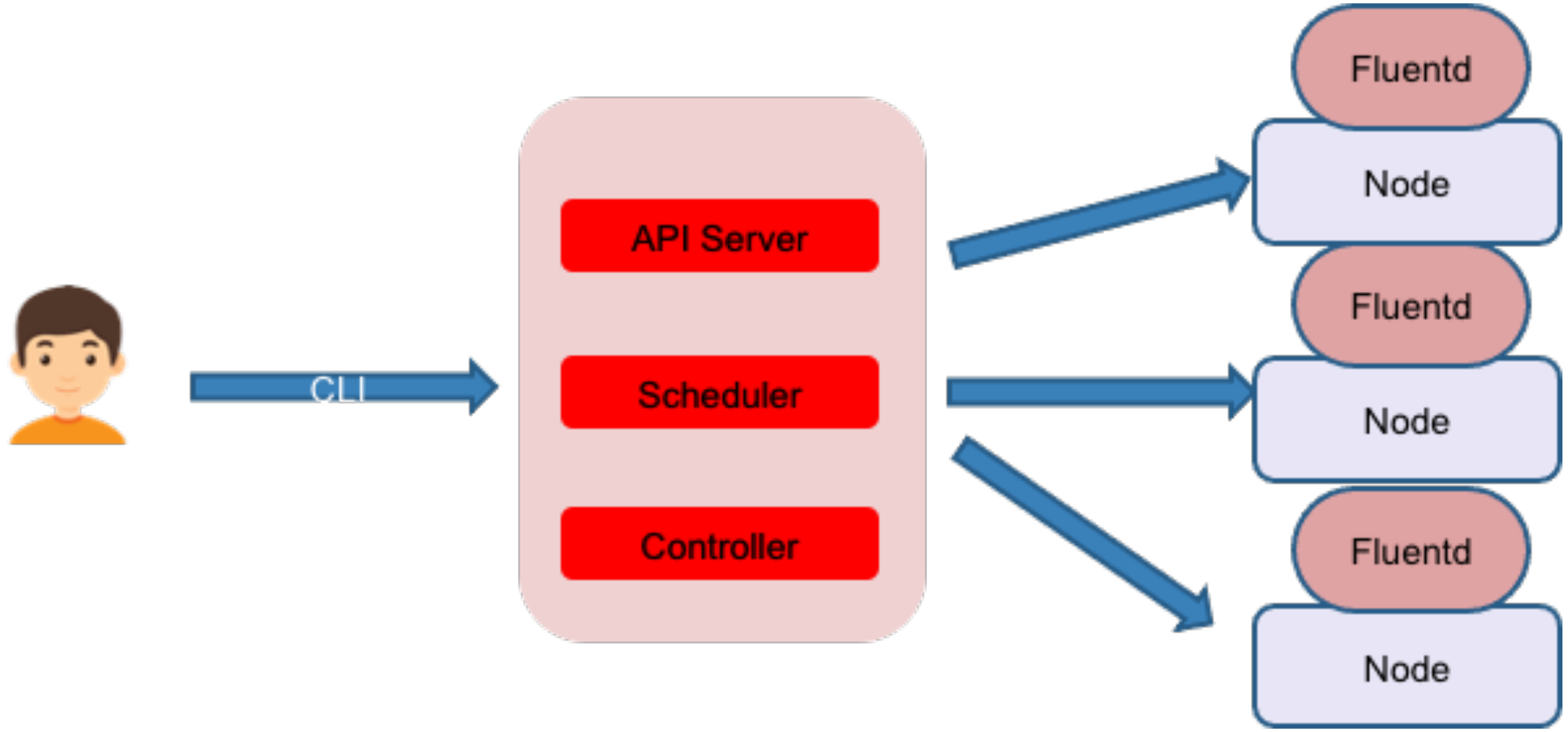
Deployment

- Rollouts as a Service
- Update
  - Rolling update
  - Recreate
- Manage ReplicaSet



DaemonSet

- Ensure that all Nodes run a copy of a Pod
- Pods are added to node once it is added to the cluster
- Usage
    - Storage daemon
    - Log collection daemon
    - Monitoring daemon



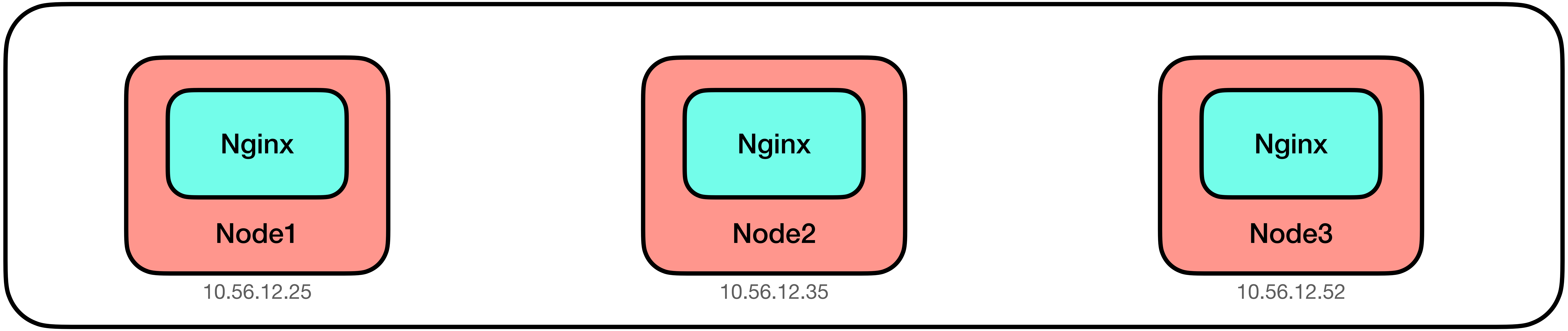
Network

- Network Connectivity
  - Container to Container (Same Node)
  - Container to Container (Across Node)
- Service
  - WAN to Container
  - L3/L4 based LB
- Ingress
  - WAN to Container
  - L7 based LB

https://kubernetes.io/docs/concepts/workloads/pods/pod/

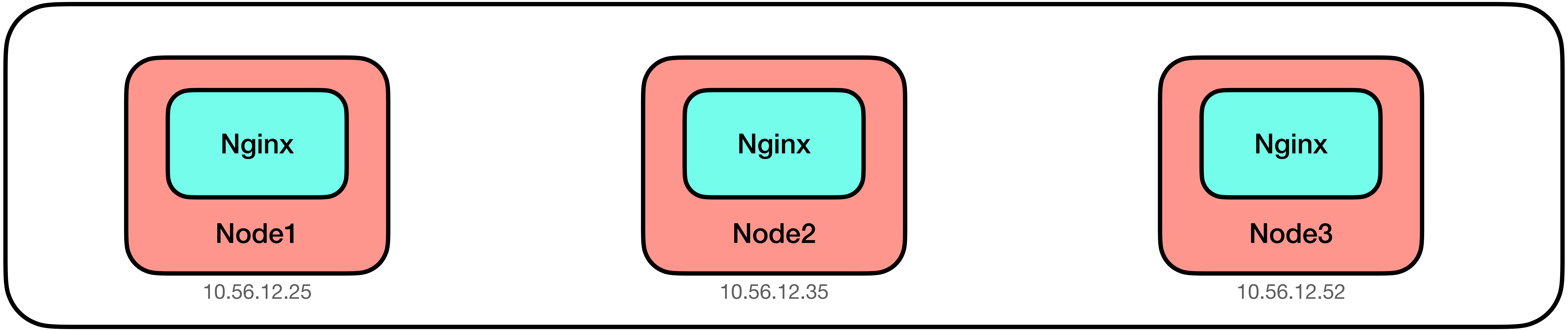
Service

- Why Service?
- Deployment
  - Nginx
  - Replica: 3



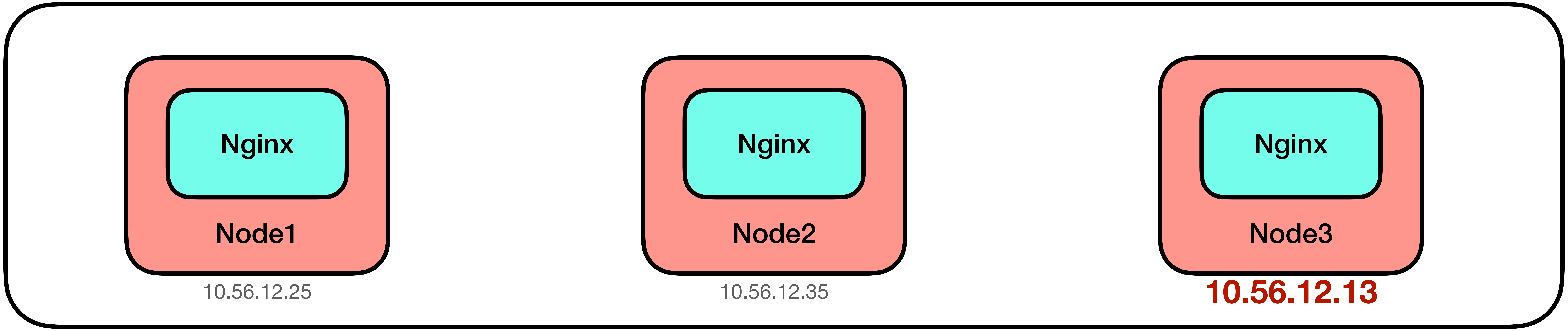
Service

- How does your application access Nginx?
- IP addresses
  - 10.56.12.25:80
  - 10.56.12.35:80
  - 10.56.12.52:80



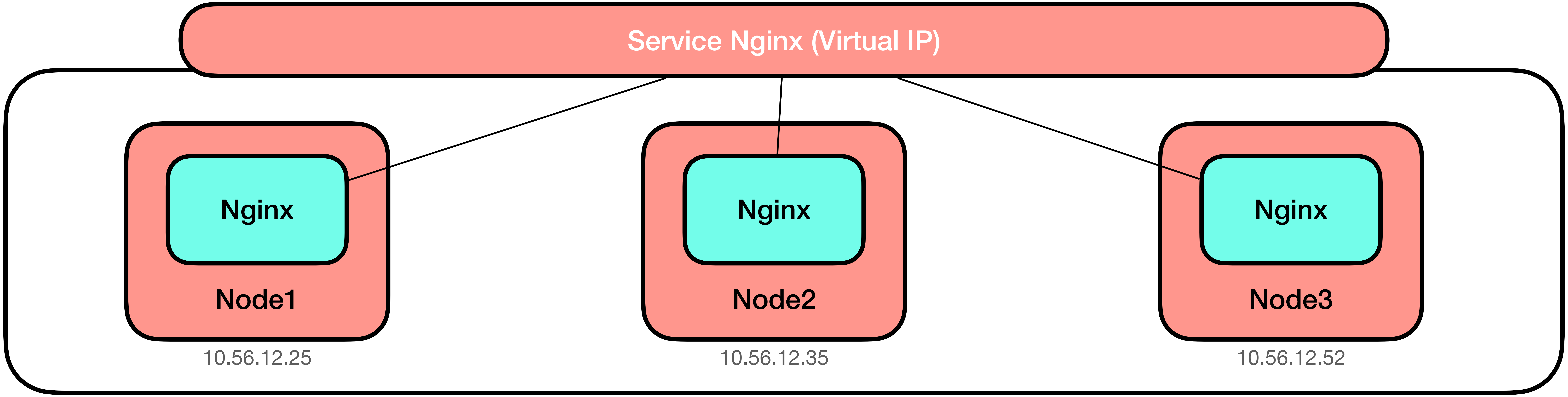
Service

- How does your application access Nginx?
- IP addresses
  - 10.56.12.25:80
  - 10.56.12.35:80
  - ~~10.56.12.52:80~~
  - 10.56.12.13:80
- Inform the apps to use the new ip address?



Service

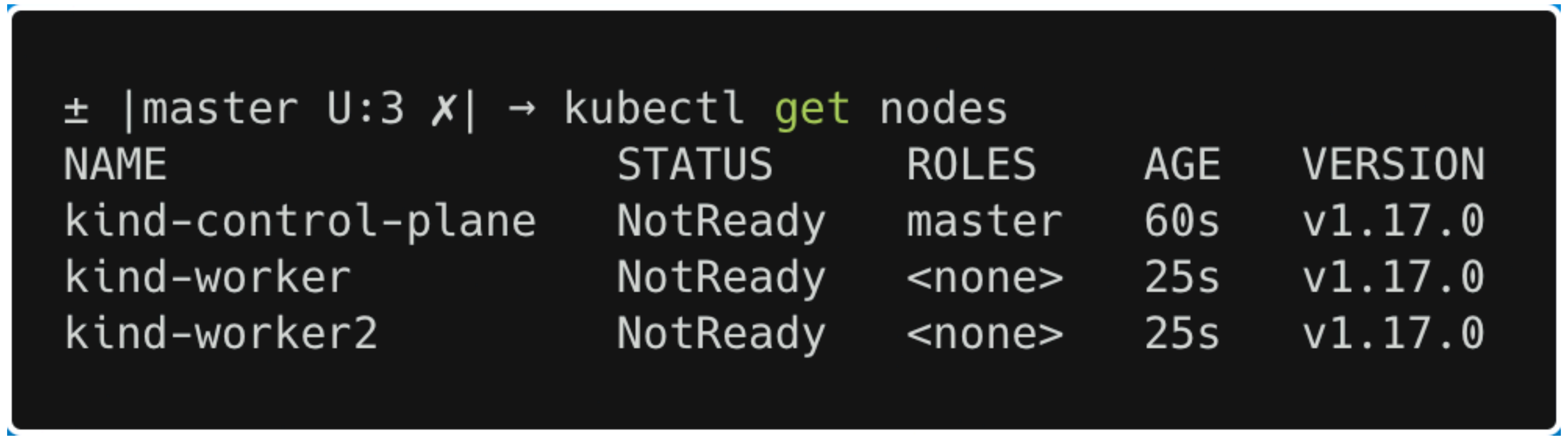
- Service provides a virtual IP address to solve this problem.
- Application point to the virtual IP
- Kubernetes forward your packet to one of the endpoints



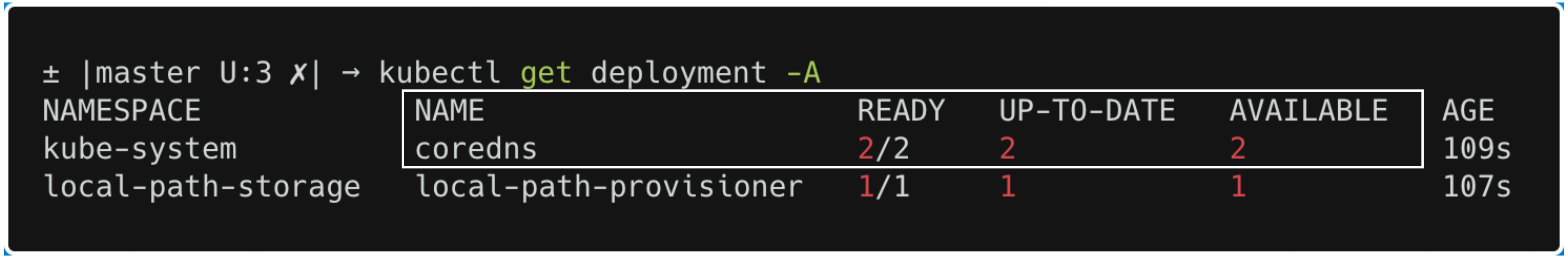
Kubectl

- Kubectl (command CLI tool)
- kubectl get ...
- kubectl apply ...
- kubectl logs ...
- kubectl exec ...
- kubectl delete ...

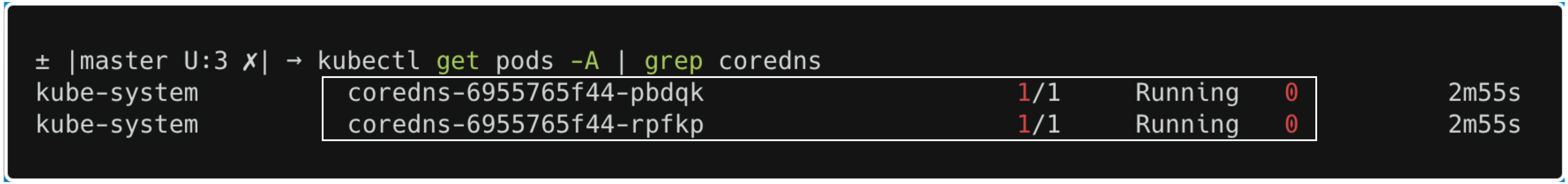
Get Information



```
± |master U:3 ✗| → kubectl get nodes
NAME                 STATUS     ROLES    AGE   VERSION
kind-control-plane   NotReady   master   60s   v1.17.0
kind-worker          NotReady   <none>   25s   v1.17.0
kind-worker2         NotReady   <none>   25s   v1.17.0
```



```
± |master U:3 ✗| → kubectl get deployment -A
NAMESPACE            NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
kube-system          coredns                  2/2     2            2           109s
local-path-storage   local-path-provisioner   1/1     1            1           107s
```



```
± |master U:3 ✗| → kubectl get pods -A | grep coredns
kube-system          coredns-6955765f44-pbdqk           1/1     Running   0          2m55s
kube-system          coredns-6955765f44-rpfkp           1/1     Running   0          2m55s
```

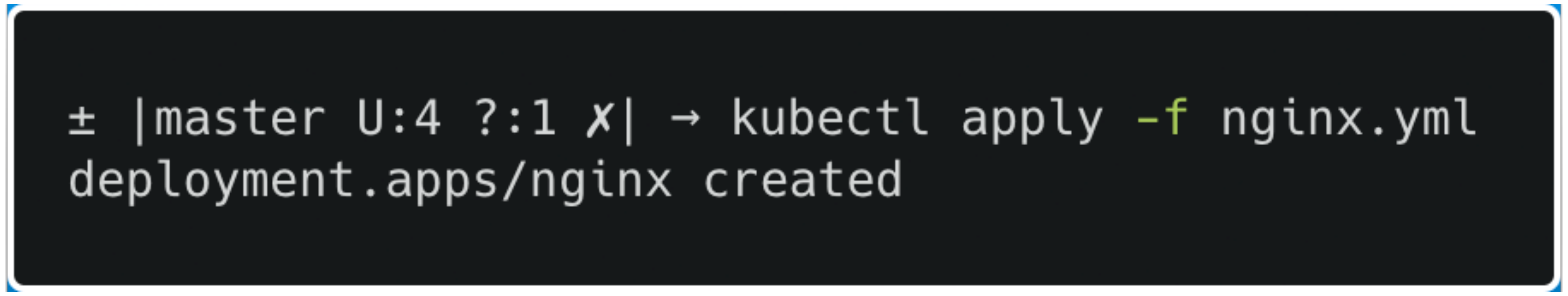
PodName

Number of Containers

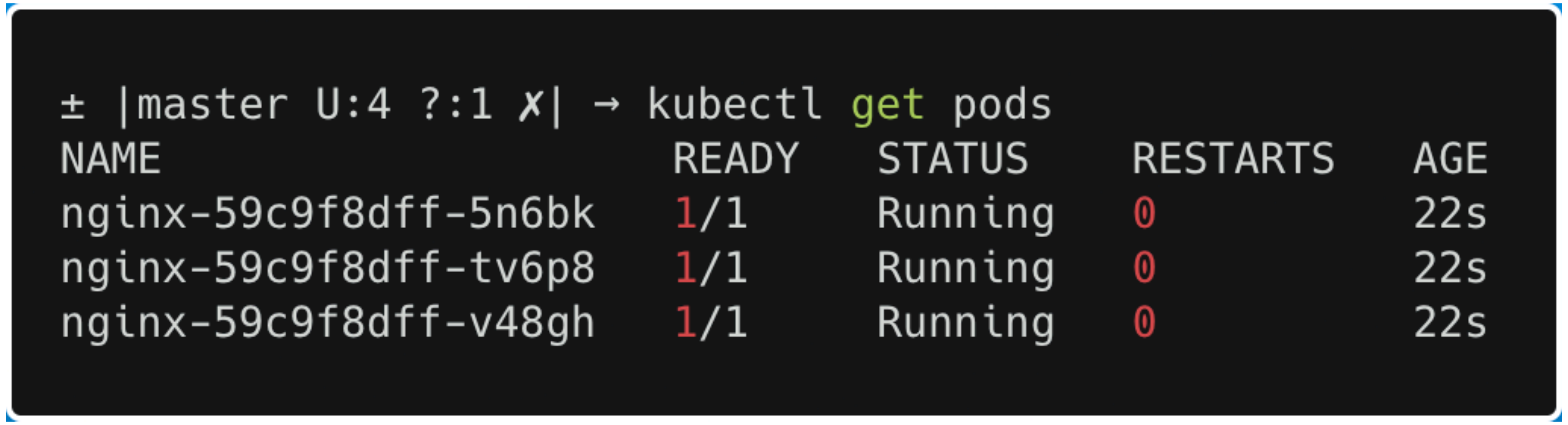
Pod Status

Apply Resources

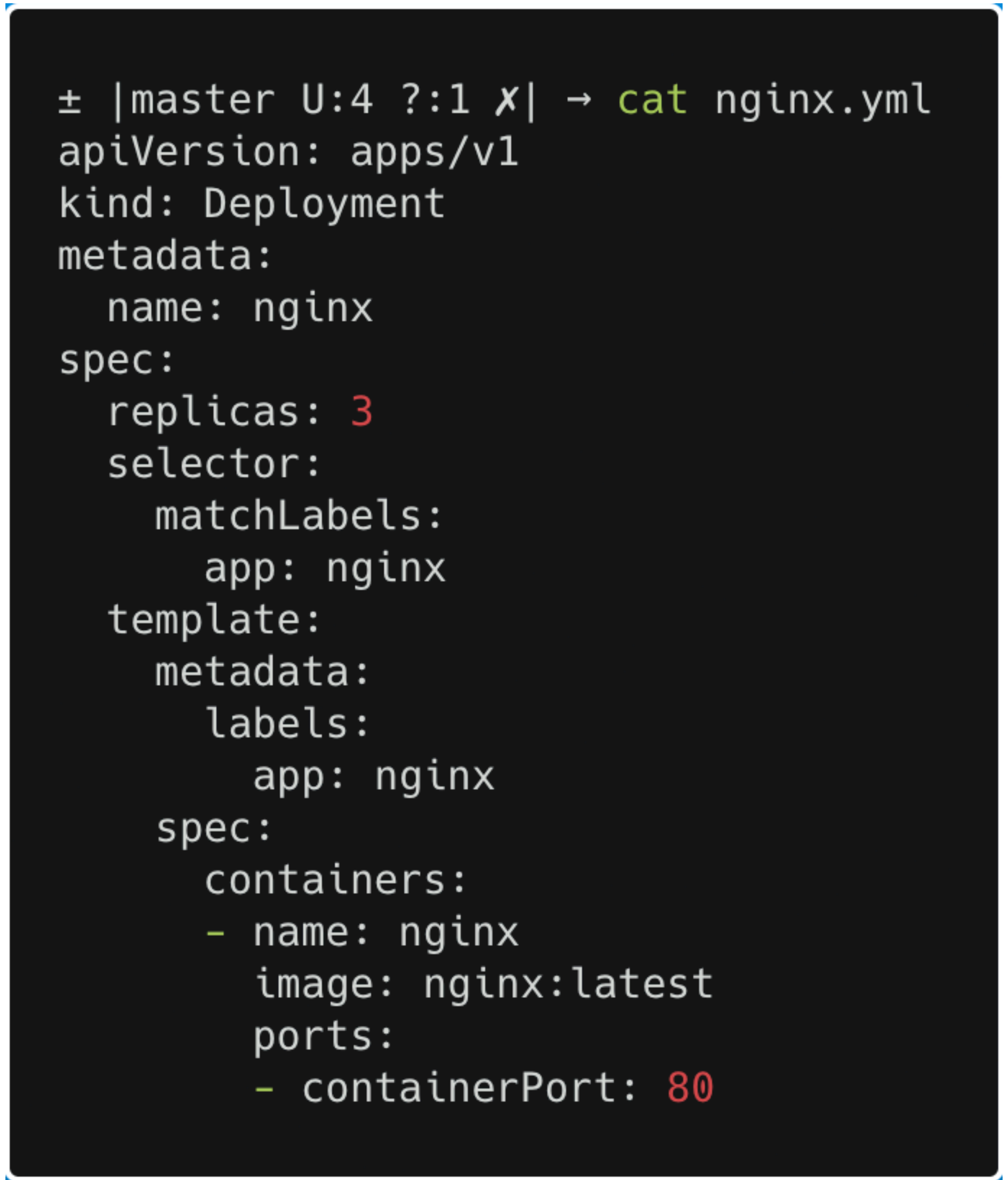
v.s (docker run)



```
± |master U:4 ?:1 ✗| → kubectl apply -f nginx.yml
deployment.apps/nginx created
```



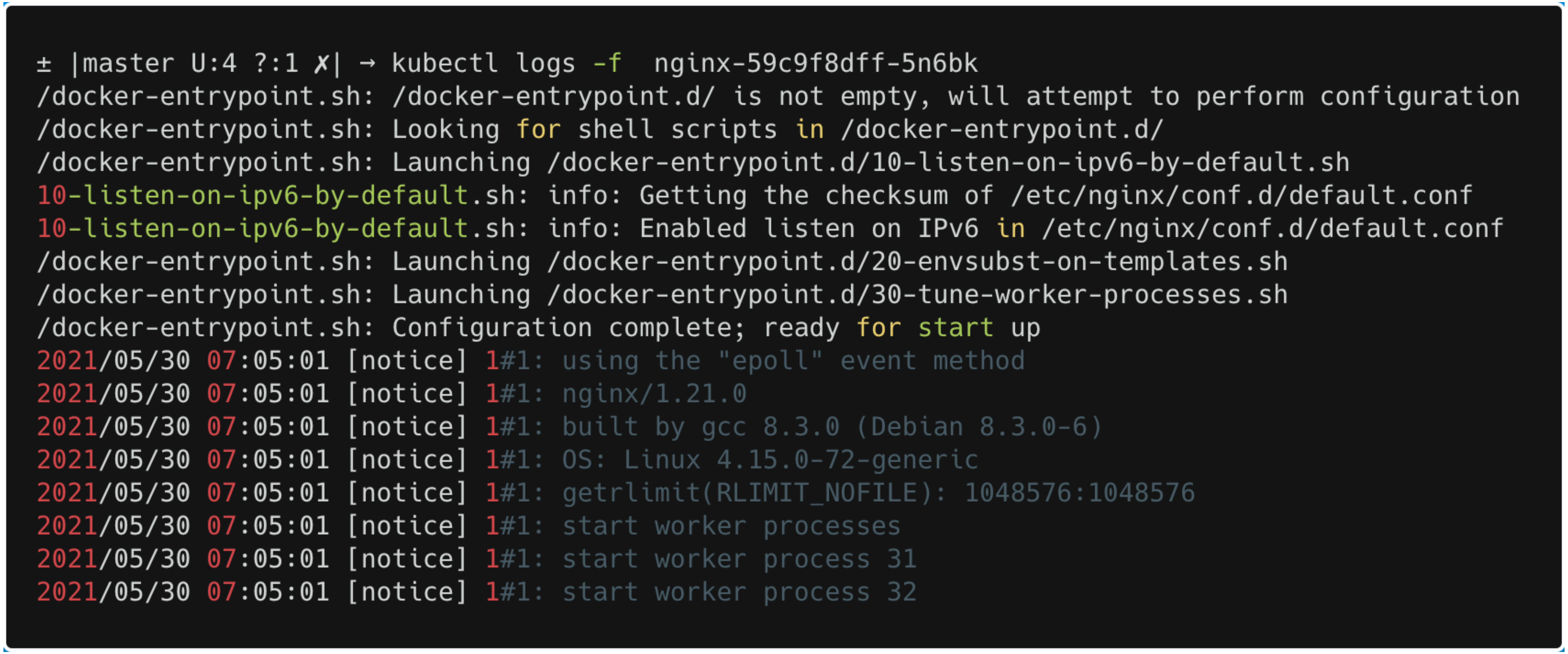
```
± |master U:4 ?:1 ✗| → kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
nginx-59c9f8dff-5n6bk   1/1     Running   0          22s
nginx-59c9f8dff-tv6p8   1/1     Running   0          22s
nginx-59c9f8dff-v48gh   1/1     Running   0          22s
```



```
± |master U:4 ?:1 ✗| → cat nginx.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

Log

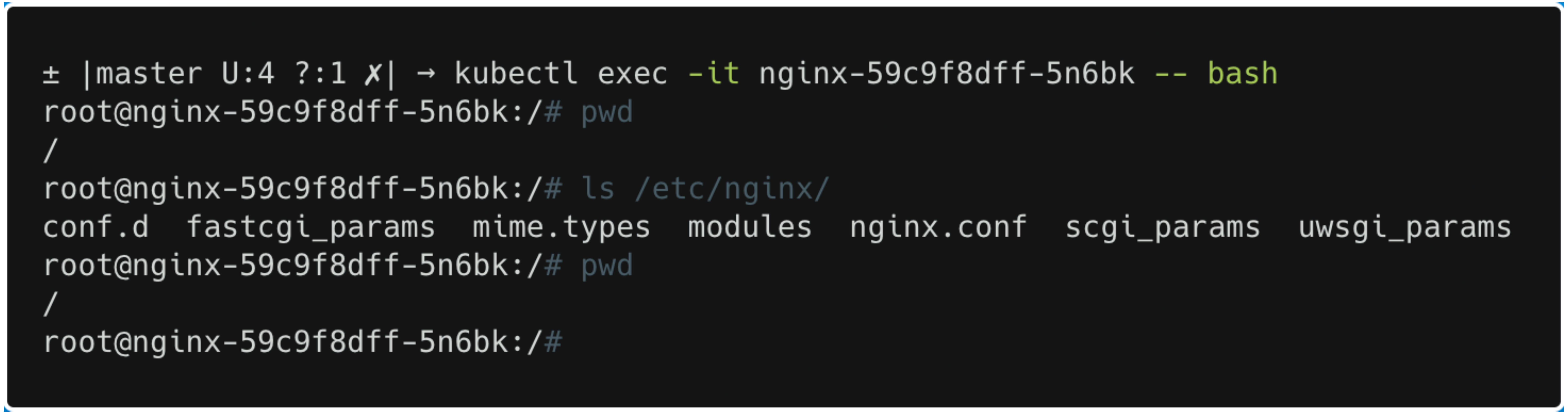
v.s (docker logs)



```
± |master U:4 ?:1 ✗| → kubectl logs -f  nginx-59c9f8dff-5n6bk
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2021/05/30 07:05:01 [notice] 1#1: using the "epoll" event method
2021/05/30 07:05:01 [notice] 1#1: nginx/1.21.0
2021/05/30 07:05:01 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)
2021/05/30 07:05:01 [notice] 1#1: OS: Linux 4.15.0-72-generic
2021/05/30 07:05:01 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2021/05/30 07:05:01 [notice] 1#1: start worker processes
2021/05/30 07:05:01 [notice] 1#1: start worker process 31
2021/05/30 07:05:01 [notice] 1#1: start worker process 32
```

Exec

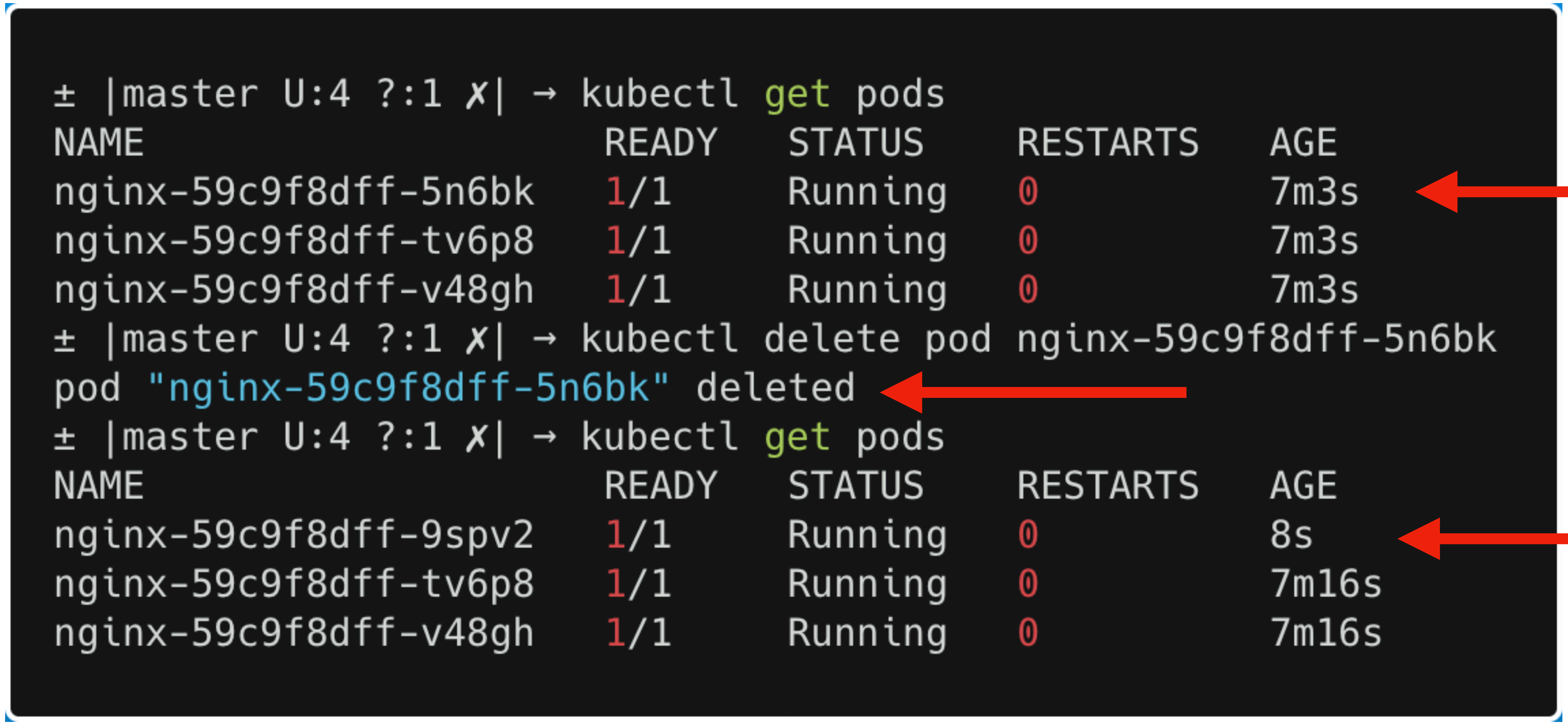
v.s (docker exec)



```
± |master U:4 ?:1 ✗| → kubectl exec -it nginx-59c9f8dff-5n6bk -- bash
root@nginx-59c9f8dff-5n6bk:/# pwd
/
root@nginx-59c9f8dff-5n6bk:/# ls /etc/nginx/
conf.d  fastcgi_params  mime.types  modules  nginx.conf  scgi_params  uwsgi_params
root@nginx-59c9f8dff-5n6bk:/# pwd
/
root@nginx-59c9f8dff-5n6bk:/#
```

Delete

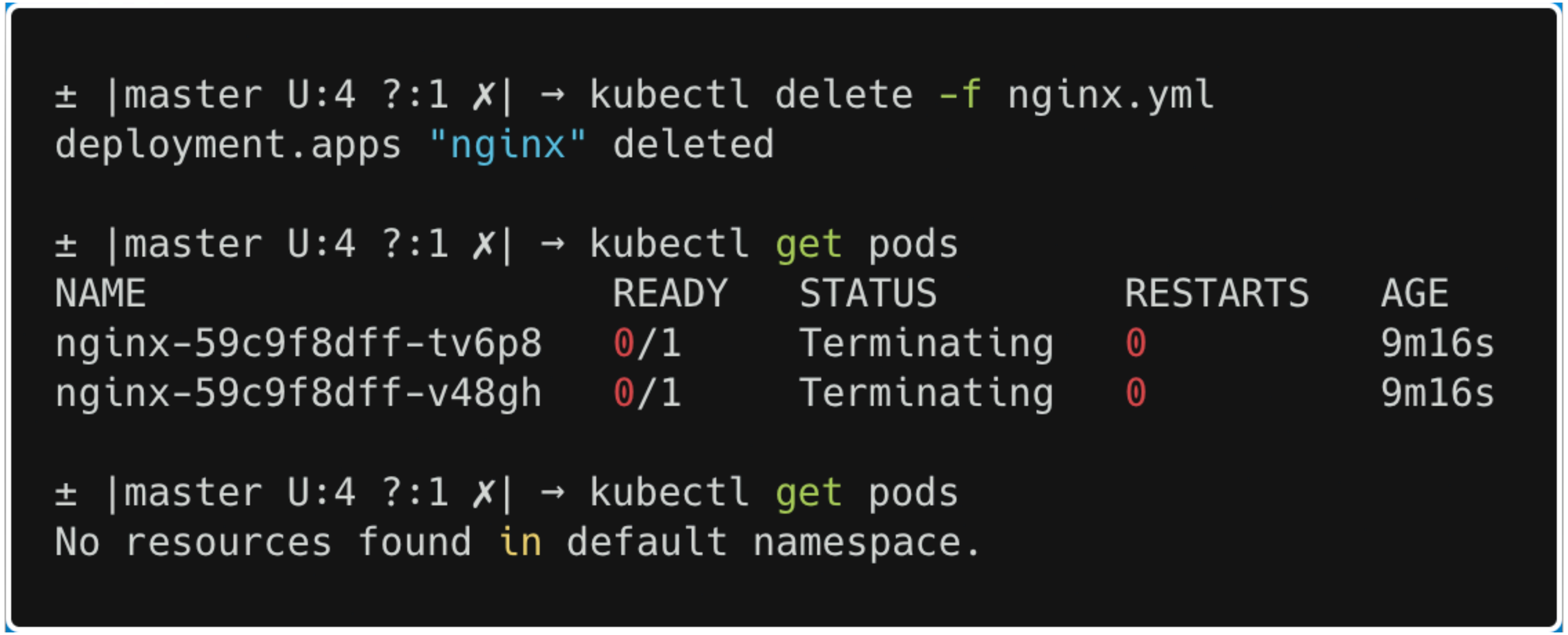
v.s (docker rm)



```
± |master U:4 ?:1 ✗| → kubectl get pods
NAME                       READY     STATUS     RESTARTS    AGE
nginx-59c9f8dff-5n6bk      1/1       Running    0           7m3s
nginx-59c9f8dff-tv6p8      1/1       Running    0           7m3s
nginx-59c9f8dff-v48gh      1/1       Running    0           7m3s
± |master U:4 ?:1 ✗| → kubectl delete pod nginx-59c9f8dff-5n6bk
pod "nginx-59c9f8dff-5n6bk" deleted
± |master U:4 ?:1 ✗| → kubectl get pods
NAME                       READY     STATUS     RESTARTS    AGE
nginx-59c9f8dff-9spv2      1/1       Running    0           8s
nginx-59c9f8dff-tv6p8      1/1       Running    0           7m16s
nginx-59c9f8dff-v48gh      1/1       Running    0           7m16s
```

Delete

v.s (docker rm)

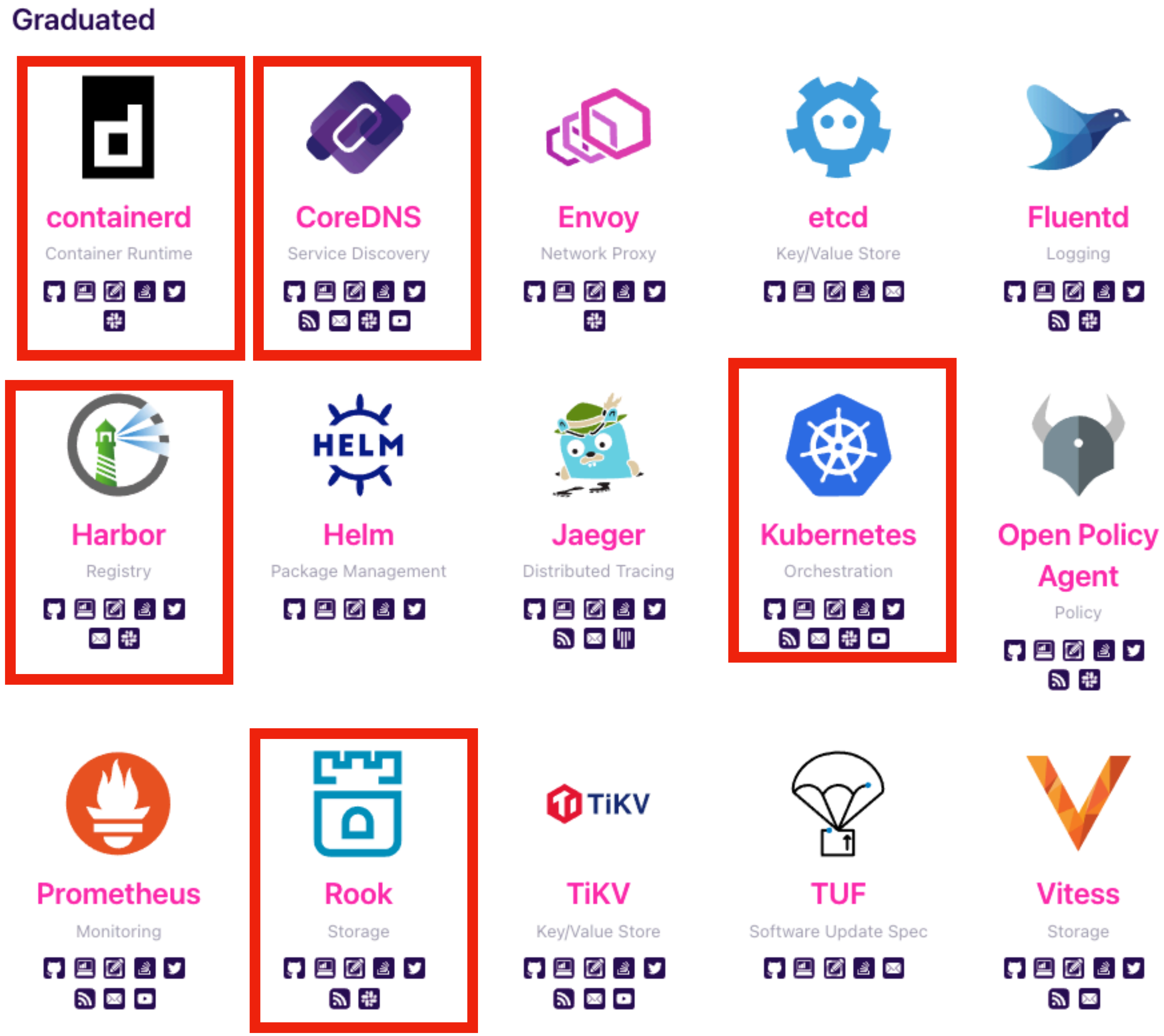


```
± |master U:4 ?:1 ✗| → kubectl delete -f nginx.yml
deployment.apps "nginx" deleted

± |master U:4 ?:1 ✗| → kubectl get pods
NAME                     READY   STATUS        RESTARTS   AGE
nginx-59c9f8dff-tv6p8    0/1     Terminating   0          9m16s
nginx-59c9f8dff-v48gh    0/1     Terminating   0          9m16s

± |master U:4 ?:1 ✗| → kubectl get pods
No resources found in default namespace.
```

CNCF

- Cloud Native Computing Foundation
- Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.
- Projects
    - Graduated
    - Incubating
    - Sandbox

CNCF (Graduated)



Graduated

containerd
Container Runtime

CoreDNS
Service Discovery

Envoy
Network Proxy

etcd
Key/Value Store

Fluentd
Logging

Harbor
Registry

Helm
Package Management

Jaeger
Distributed Tracing

Kubernetes
Orchestration

Open Policy Agent
Policy

Prometheus
Monitoring

Rook
Storage

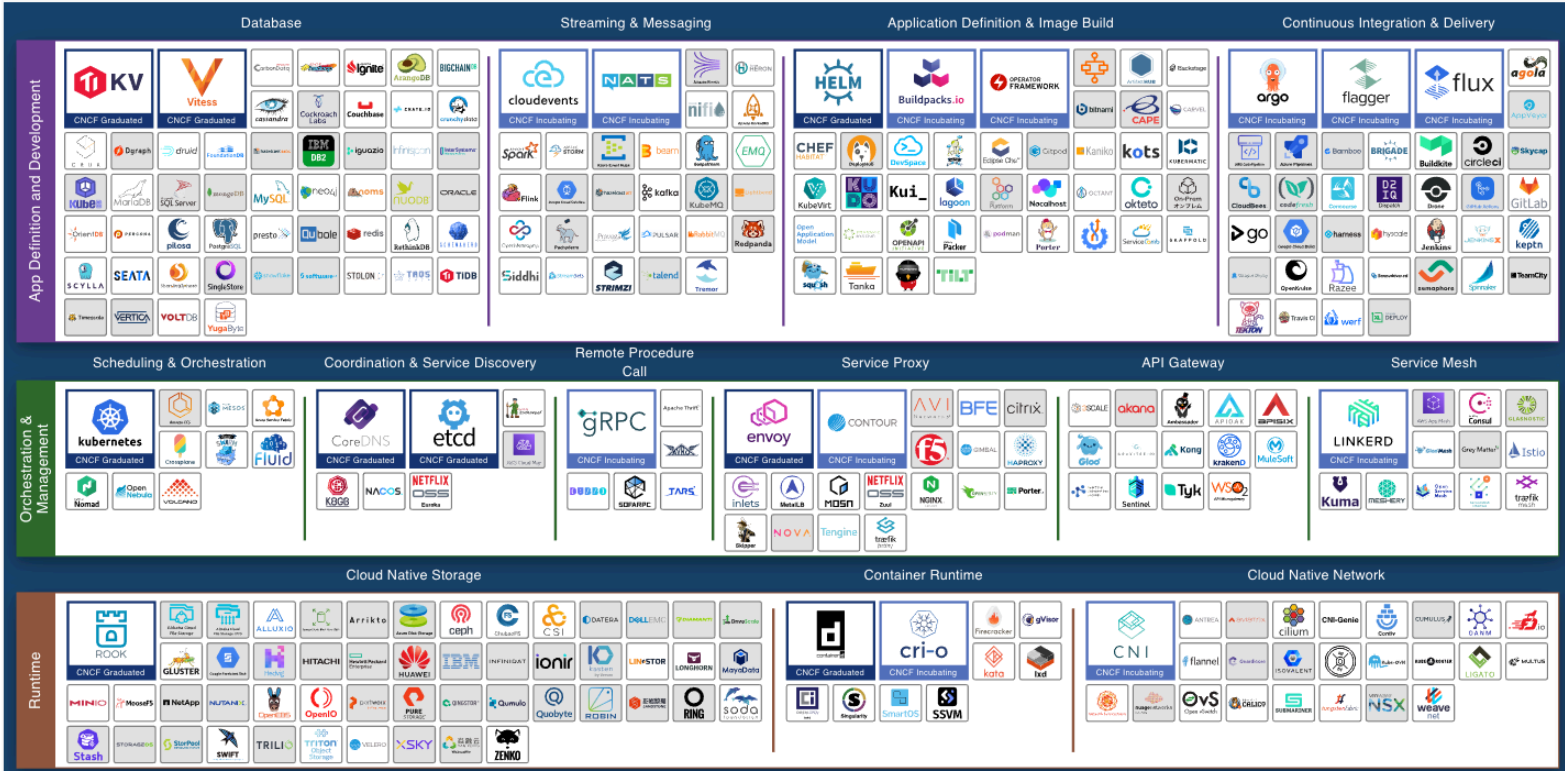
TiKV
Key/Value Store

TUF
Software Update Spec

Vitess
Storage

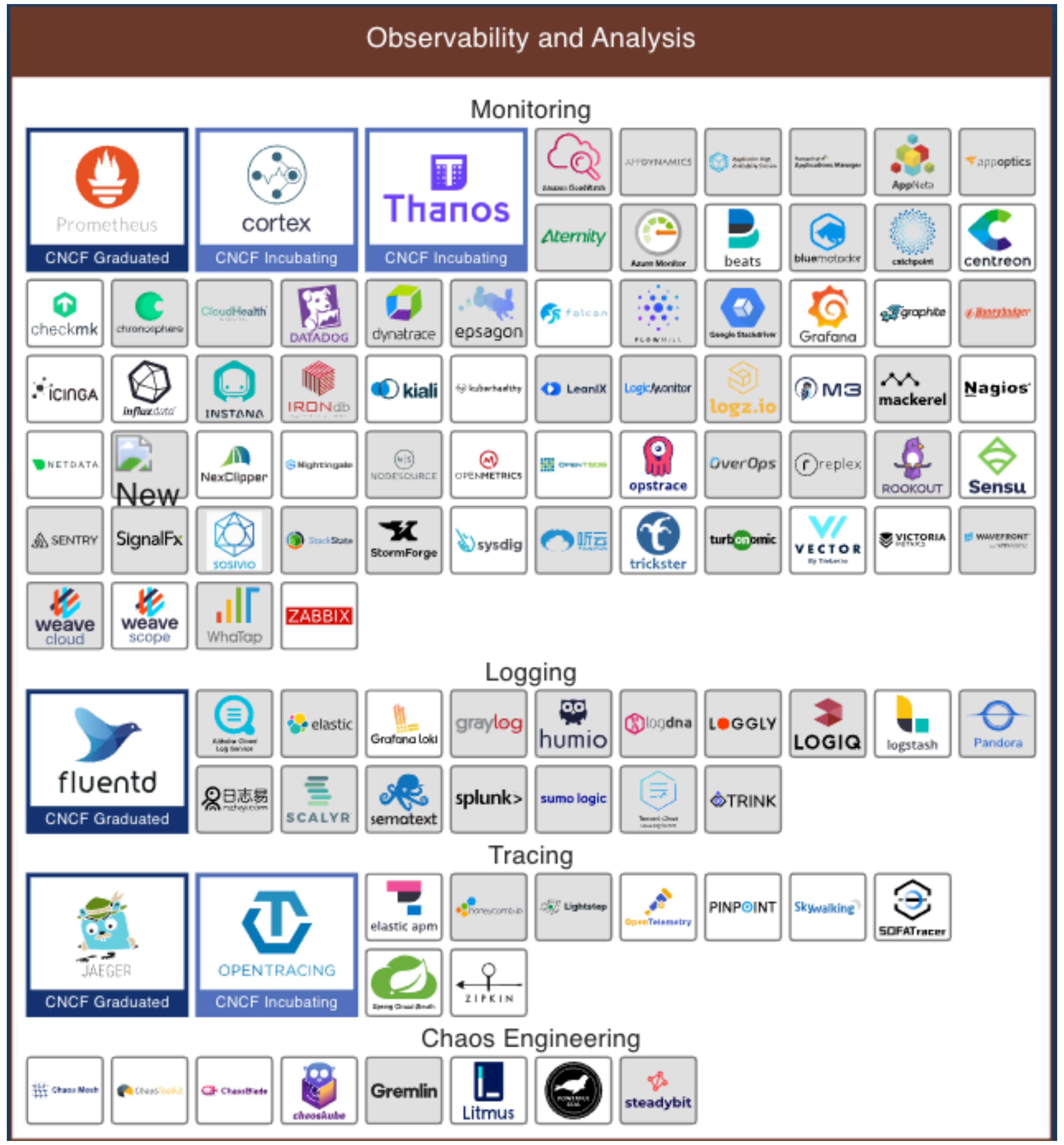
https://www.cncf.io/projects/

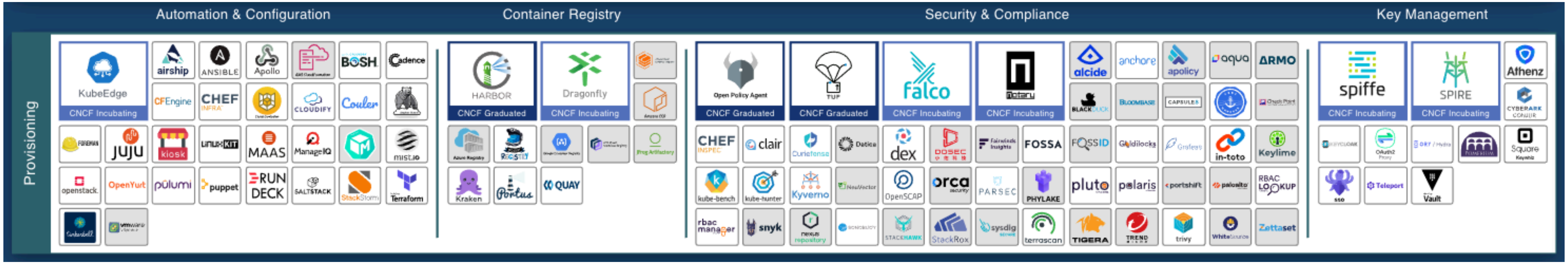
CNCF Landscape



https://landscape.cncf.io/

CNCF Landscape





https://landscape.cncf.io/

CNCF

- Too many projects
- Too many fields
    - Compute, Network, Storage, Database, Steaming, Configuration, CI/CD, Observability ...etc
- How to learn?
    - Learn from the system and core concept, not the application.

CNCF

- More and more projects are friendly to users.
    - Deploy and manage via YAML files
    - Debugging is the difficult part

Experience

- Share Job description from Silicon Vally
- Have to be aggressive
- "How to learn new technology" is the most important skill you have to learn

Hey Hung-Wei,

Checking in to see if you're interested in this role again :)

If you're interested in the platform/DevOps side of the house, we're looking for a hands-on Staff Software Engineer to join our platform engineering team. You would be responsible for architecting and building the next generation commerce platform for our merchants.

In addition, you’d take part in maintaining our cloud-based, large-scale distributed systems sustainably and brainstorm ways to automate. And/or focus on our cloud infra and Kubernetes container eco-system. Exciting stuff.

If you're interested...any chance you're free to talk this week or next? Feel free to send me an email/resume and I can send you a calendar link.

Hewlett Packard Enterprise/Aruba Lead Cloud Development (Permanent) Opportunity in Santa Clara, CA (remote for the foreseeable future)

Hope you and your friends/family are well. I am not sure if you are looking at this time, but I quickly wanted to let you know about a new Lead Cloud Development role that might interest you.

The work for these positions deals with our private and public cloud platform/infrastructure and development of applications on top of the platforms to enhance efficiency for internal engineering teams. More detail below. Let me know if you are interested and hope to hear from you soon!

- You will have multi-year experience in Developing and implementing cloud applications.

- You will have a proven record of developing and releasing cloud applications in production environment.

- You will have knowledge of Cloud Platform - CoreOS, Kubernetes and Docker containers.

- You will have experience with messaging systems like RabbitMQ and distributed streaming platforms like Kafka.

- You will have multi-year experience with Python, Golang, and Java

Senior or Lead SRE role -- Hiring Now -- Fully-Remote or On-Site

Good afternoon,

One of my top clients is looking to bring on a Senior or Lead Site Reliability Engineer (SRE). An ideal candidate has a background in DevOps Engineering or SRE that is experienced in the following:

- AWS
- Terraform or Cloudformation
- Docker or Kubernetes
- Python

Based on my initial review of your profile, I feel that you would be qualified for this position.

My client is a well-funded and technically innovative start-up that is offering the SRE role to candidates who are eager to work with a really broad and modern tech. stack in a heavy hands-on environment, be a part of a collaborative engineering team with the opportunity for mentorship, and develop CI/CD solutions centered around their core product that involves AI.

The first round of interviews will begin later this week. I would love to get you into consideration before that interview period closes.

In addition to a competitive salary, the client is offering an extraordinary benefits package including a 401k, Medical/Dental/Vision, a performance-based bonus structure, and potentially Equity.

When are you available to speak on the phone this afternoon or tomorrow morning?

PLEASE NOTE: This client is not open to employing Visa-Holders at this time.

Best Regards,

Hung-Wei Chiu - Staff SRE - PayPal

Hi Hung-Wei,

I know that with everything going on right now, your inbox is far from your highest priority, so I promise I'll be quick! We're looking for collaborative DevOps engineers with a passion for containerization to join our team. We have a Staff level role open on the team that owns our Kubernetes and AWS infrastructure that I thought might catch your eye!

We're open to hiring someone fully remote for this role. Every time you pay for an Uber ride, book a stay through Airbnb or pay with PayPal when you check out online, you're using our product. We're looking for folks who will help us make sure that the process continues to run smoothly. Across our org, we're language agnostic and value pair-programming and test-driven development.

Senior Platform Engineering Opportunity - Sunnyvale, California

Hi Hung-Wei,

I hope you and your family are keeping well!

I'm currently partnered with a well-known Media company who are looking to add an experienced Engineer to join their growing team.

Requirements:

6+ years of experience in one or more of the following: Java, Python, Scala, Go

6+ years of experience with algorithms, data structures, complexity analysis, and software design

4+ years experience with troubleshooting Unix/Linux

Experience developing Micro service based solutions

Strong hands-on experience on Kubernetes and Ingress/Egress/Multi-tenancy use cases

Strong hands-on experience with containers, and container orchestration, Kubernetes

Strong hands-on experience implementing cloud architecture with AWS, Docker, etc.

Does this sound like something you're interested in? If so, feel free to respond and we'll schedule a call. Also if you can drop your number, email and resume that would help.

Hi Hung-Wei,

I am recruiting for 4 Remote Senior DevOps Engineers on behalf of an Artificial Intelligence organisation.

- AWS, Kubernetes, Terraform
- $150k-$180k + equity
- Remote WFH

Could this one be of interest? If not currently, let's connect. You never know down the line!

Title: Azure Dev Ops Engineer

Location: Sunnyvale, CA

Summary

Need a certified Azure dev ops engineer who has a minimum of 4 years of experience on Cloud Technologies. Should have in-depth knowledge on Azure Architecture, CI/CD Tools & Kubernetes. Should have extensive knowledge on CI/CD pipeline creation and containerization.

Preferred Qualifications

Resource should have minimum 4+ years on cloud technologies.

Should be an Azure Certified resource or should be holding any certification on Linux OS

Should have in-depth knowledge on Azure architecture.

Should have hands-on experience with various CI/CD tool sets (GitHub, Jenkins, Puppet, Docker)

Should have extensive knowledge on CI/CD pipeline creation and containerization

Experience deploying Kubernetes ecosystem into Azure Container Service

Automation experience using Python/Ruby/Go – any scripting

Required Skills

Azure Kubernetes Cloud CI/CD

Hi Hung-Wei,

Hope you are doing well. I wanted to reach out and share a post I recently shared as I am looking for SRE/DevOps Engineers.

You can learn more in the post.

https://www.linkedin.com/posts/christina-finster-33499224_hiring-openposition-recruiting-activity-6684231580124282880-6VF8

Looking over your background the work that you were doing at Thunder Token and assuming now what you are doing at Open Networking Foundation I think you would be a great candidate for the Terraform or Kuberentes Engineer opening at my client.

My client is the worlds largest Crypto Trading Platform in the world trading billions in crypto products a day.

They are actively scaling their SRE and DevOps Team and I would love to see if you would be interested in chatting about the opportunity.

Please let me know if you'd like to schedule sometime to chat: https://calendly.com/christina-finster-1

*And if you're not interested maybe you could think of someone to share my post with or tag in the comments.

Zuora - Subscription Economy - Principal/Senior SRE??

Hope you had a good weekend. During this tough covid year and a crazy election night we have to keep moving forward. Zuora is still hiring bc we are growing/scaling and in need of help. I'm reading about you, thinking to myself, "I wish you worked with us at Zuora." Just what i think.. especially since strong Principal/Senior SRE's are few and far between.I know you get hit up everyday with generic messages, and are probably tired of it.You have no reason to answer, and i'm sure if you want a new job you could find one. But still, i have a job to do, and try to do it well :) Give it 20 seconds, and tell me why not. I would think 3 days a week in the office, but we could work with you.

Principal or Senior SRE/DevOps - have a seat at the table and change the culture/environment. What we are technically is below.

Zuora is a Microservice Environment

Zuora has adopted AWS

Zuora is using Kubernetes

Zuora is building automation

Zuora uses Terraform

- Kubernetes and Docker
    - Docker container is a basic skill, market assumes everyone already know it.
- Integration other tools to build whole system
    - Cloud providers (AWS/GCP/Azure)
    - CI/CD tools (Github, Git, Gitlab, Jenkins, Drone...etc)

Q&A