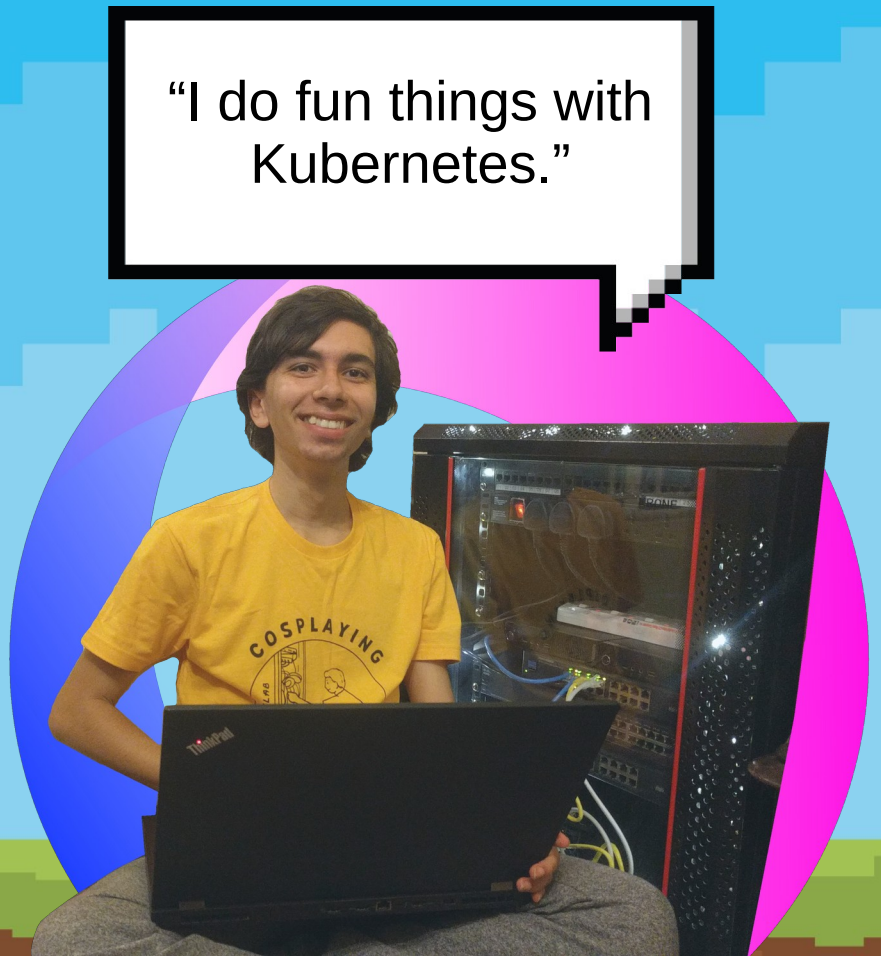
The Future Of Digital Infrastructure

By Alex "not a developer" Bissessur

alex.yaml

```yaml
---
apiVersion: v1
kind: Person
metadata:
 name: Alex Bissessur
spec:
  work:
    company: La Sentinelle
    role: Kubernetes Person
    location: Mauritius
  contact:
    website: alexbissessur.dev
    mastodon: moris.social/@AlexB
    github: github.com/xelab04
  interests:
    - Kubernetes
    - Linux
    - Free & Open Source Software
  hobbies:
    - Playing kubectl with Homelab
```



Glossary

- Virtual Machines
- Containers
- Cloud Native
- Kubernetes
- Microservices
- Infrastructure as Code

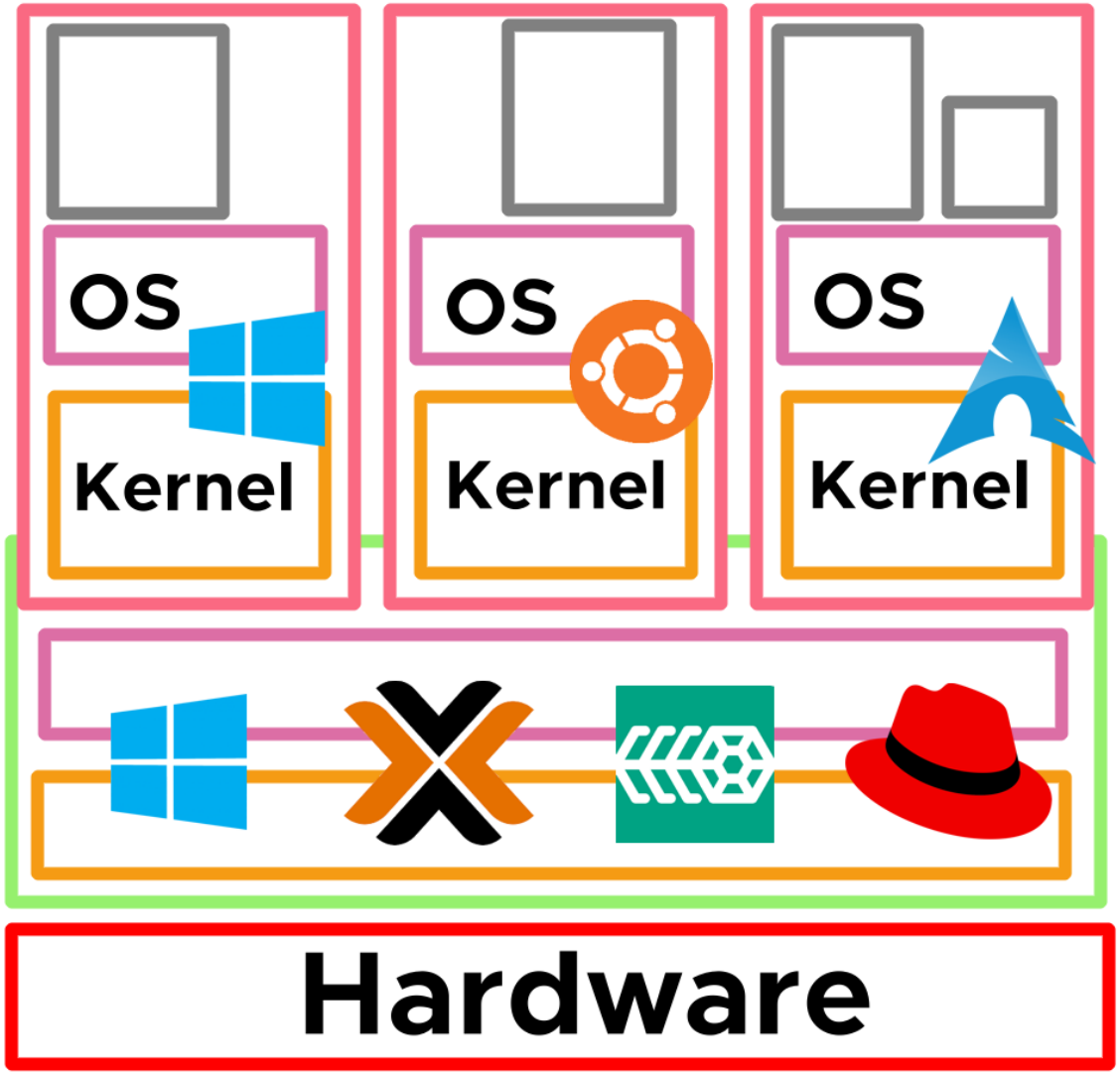
What is the future of IT Infra?

What is the future of IT Infra?

- Laziness. Being incredibly lazy.

Virtual Machines

- A complete system within a system
- Runs on a hypervisor or host OS
- Emulates hardware (hdd, cpu, ram)
- Runs entire operating system
- Runs the guest OS kernel


OS
Kernel
OS
Kernel
OS
Kernel
Hardware

Issues with VMs

- Heavy – requires hardware emulation and running a whole OS
  - 10 Vms → 10 kernels
  - OS uses resources even when idle
- Large – not portable or easily moved between developers/admins
- Inconvenient – setting up VMs at scale is a chore

Why do We Use VMs?

- Isolation – separating applications which have different uses
- Security – isolating applications and their data in case of breach
- Dependency isolation – different applications may have different requirements – eg PHP 6 vs 8
- Splitting a host – host resources can be split across VMs

History of Containers

- chroot originated with Bill Joy in BSD in 1982
  - lets you create a directory to virtualise the file system
  - merged while drinking, or smoking, or both
- chroot developed into Jails for BSD
  - still in use today
- Jails was targeted at running binaries in an isolated environment
  - no access to main filesystem
  - no ability to make changes to host OS

What is A Container

- It is a Linux process.

What is A Container

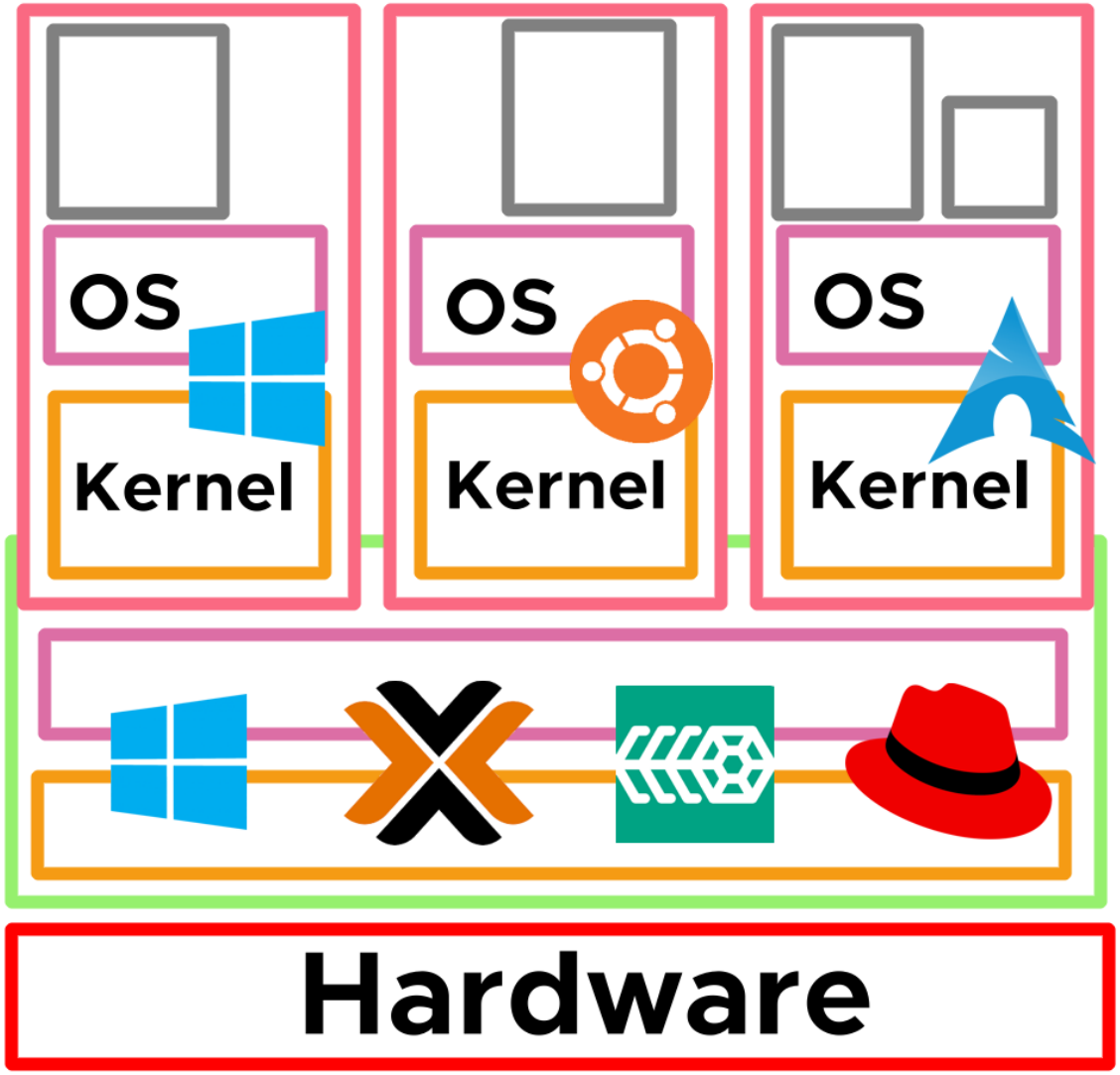
- It is a Linux process.
- It provides isolation.
- It is replicable.
- Typically lightweight.
- And runs basically anywhere*.

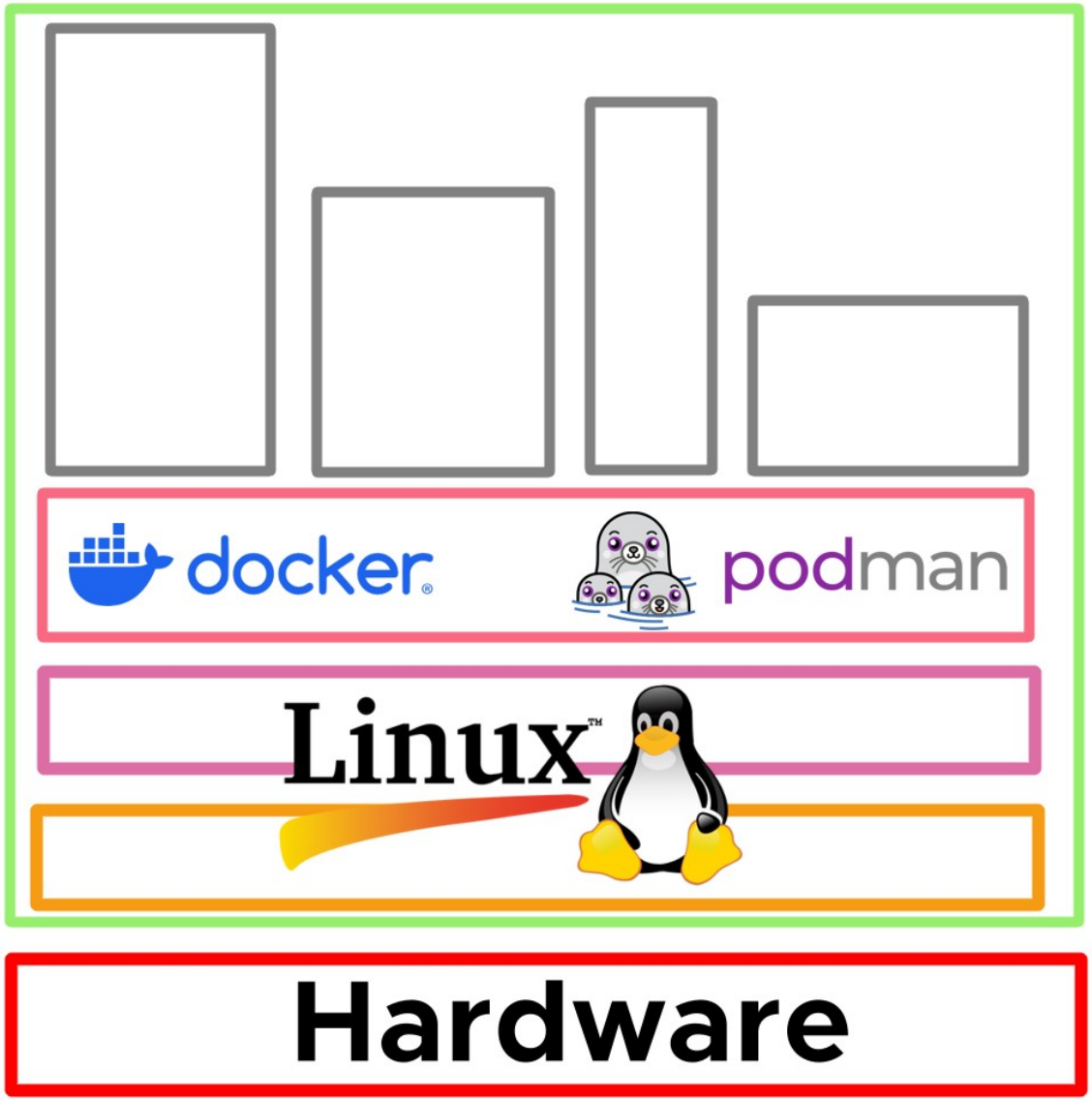
Container Definition

```
FROM registry.suse.com/bci/nodejs:22  AS BUILDER

WORKDIR /app

RUN npm install -g pnpm@latest-10

COPY . /app

RUN pnpm install

RUN pnpm run nuxt generate

FROM registry.suse.com/suse/nginx:1.21 AS PRODUCTION

COPY --from=BUILDER /app/packages/frontendmu-nuxt/dist /srv/www/htdocs/
```

Why Are Containers Cool?

- Lightweight on resources – no hardware emulation, no kernel to run, no OS bloat
- Lightweight on storage – container images can be as small as 10MB
- Portable – they run the same on dev laptop and prod server
- Isolated – secure, and avoids conflicts in applications


OS
Kernel
OS
Kernel
OS
Kernel
Hardware


docker
podman
Linux™
Hardware

Scaling

- Modern day has new demands for internet services.
- A single server is not enough for high load, plus physical/hardware limits
- Having many servers allows large scale services
- Supercomputers are built out of many smaller computers

Cloud Native

- The practice of building software and infrastructure prioritising modern demands for internet services.
- An ecosystem of tools and platforms to build infrastructure in a scalable way, often using containers.

App Definition and Development

Application Definition & Image Build

- Helm — CNCF GRADUATED
- Artifact Hub — CNCF INCUBATING
- Backstage — CNCF INCUBATING
- Buildpacks.io — CNCF INCUBATING
- dapr — CNCF INCUBATING
- KubeVela — CNCF INCUBATING
- KubeVirt — CNCF INCUBATING
- Operator Framework — CNCF INCUBATING

Database

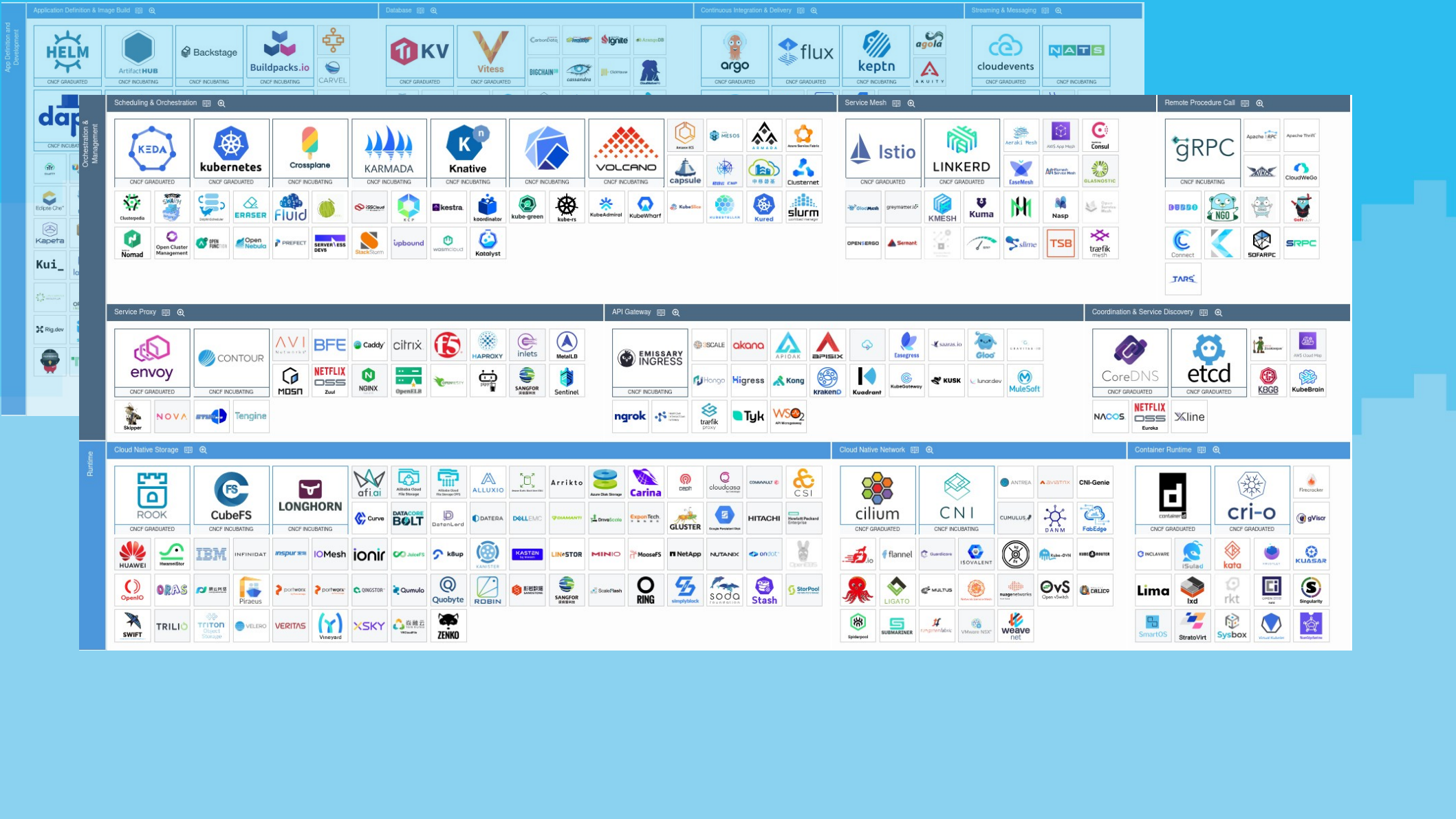
- TiKV — CNCF GRADUATED
- Vitess — CNCF GRADUATED

Continuous Integration & Delivery

- argo — CNCF GRADUATED
- flux — CNCF GRADUATED
- keptn — CNCF INCUBATING
- OpenKruise — CNCF INCUBATING

Streaming & Messaging

- cloudevents — CNCF GRADUATED
- NATS — CNCF INCUBATING
- Strimzi — CNCF INCUBATING


App Definition and Development
Application Definition & Image Build
HELM
CNCF GRADUATED
ArtifactHUB
CNCF INCUBATING
Backstage
CNCF INCUBATING
Buildpacks.io
CNCF INCUBATING
CARVEL
dapr
CNCF INCUBATING
Database
KV
CNCF GRADUATED
Vitess
CNCF GRADUATED
Ignite
Continuous Integration & Delivery
argo
CNCF GRADUATED
flux
CNCF GRADUATED
keptn
CNCF INCUBATING
agola
AKUITY
Streaming & Messaging
cloudevents
CNCF GRADUATED
NATS
CNCF INCUBATING
Orchestration & Management
Scheduling & Orchestration
KEDA
CNCF GRADUATED
kubernetes
CNCF GRADUATED
Crossplane
CNCF INCUBATING
KARMADA
CNCF INCUBATING
Knative
CNCF INCUBATING
VOLCANO
CNCF INCUBATING
capsule
Clusternet
slurm
Service Mesh
Istio
CNCF GRADUATED
LINKERD
CNCF GRADUATED
Consul
EaseMesh
GLASNOSTIC
KMESH
Kuma
Nasp
TSB
traefik mesh
Remote Procedure Call
gRPC
CNCF INCUBATING
Apache Thrift
CloudWeGo
NGO
Connect
SOFARPC
SRPC
TARS
Service Proxy
envoy
CNCF GRADUATED
CONTOUR
CNCF INCUBATING
BFE
Caddy
citrix
f5
HAPROXY
inlets
MetalLB
NETFLIX OSS Zuul
NGINX
OpenELB
SANGFOR
Sentinel
Skipper
NOVA
Tengine
API Gateway
EMISSARY INGRESS
CNCF INCUBATING
SCALE
akana
APIOAK
APISIX
Easegress
Gloo
Higress
Kong
krakenD
Kuadrant
KubeGateway
KUSK
MuleSoft
ngrok
træfik proxy
Tyk
WSO2
Coordination & Service Discovery
CoreDNS
CNCF GRADUATED
etcd
CNCF GRADUATED
AWS Cloud Map
KBGB
KubeBrain
NACOS
NETFLIX OSS Eureka
Xline
Runtime
Cloud Native Storage
ROOK
CNCF GRADUATED
CubeFS
CNCF INCUBATING
LONGHORN
CNCF INCUBATING
afi.ai
Arrikto
Carina
ceph
cloudcasa
CSI
Curve
DATACORE BOLT
DATERA
DELL EMC
GLUSTER
HITACHI
HUAWEI
IBM
INFINIDAT
IOMesh
ionir
kBup
KASTEN
LINESTOR
MINIO
MooseFS
NetApp
NUTANIX
OpenIO
ORAS
Piraeus
portworx
QINGSTOR
Qumulo
Quobyte
ROBIN
SANGFOR
soda foundation
Stash
StorPool
SWIFT
TRILIO
VELERO
VERITAS
Vineyard
XSKY
ZENKO
Cloud Native Network
cilium
CNCF GRADUATED
CNI
CNCF INCUBATING
ANTREA
AVISTRIX
CNI-Genie
CUMULUS
DANM
FabEdge
flannel
LIGATO
MULTUS
OvS
Open vSwitch
Container Runtime
container d
CNCF GRADUATED
cri-o
CNCF GRADUATED
Firecracker
gVisor
INCLAVARE
iSulad
kata
KUASAR
Lima
lxd
rkt
Singularity
SmartOS
StratoVirt
Sysbox


App Definition and Development
Application Definition & Image Build
HELM
Artifact HUB
Backstage
Buildpacks.io
CARVEL
dapr
Database
KV
Vitess
Ignite
Continuous Integration & Delivery
argo
flux
keptn
agola
Streaming & Messaging
cloudevents
NATS
Orchestration & Management
Scheduling & Orchestration
KEDA
kubernetes
Crossplane
KARMADA
Knative
VOLCANO
capsule
Service Mesh
Istio
LINKERD
Consul
EaseMesh
GLASNOSTIC
Remote Procedure Call
gRPC
Apache Thrift
CloudWeGo
Provisioning
Security & Compliance
Falco
Open Policy Agent
TUF
in-toto
KEYCLOAK
Kyverno
Notary
Automation & Configuration
Cloud Custodian
KubeEdge
airship
ANSIBLE
Apollo
Container Registry
HARBOR
Dragonfly
Key Management
spiffe
SPIRE
Runtime
Service Proxy
envoy
CONTOUR
Cloud Native Storage
ROOK
CubeFS
Observability and Analysis
Observability
fluentd
JAEGER
Prometheus
cortex
OpenTelemetry
Thanos
Continuous Optimization
Chaos Engineering
Chaos Mesh
Litmus
Feature Flagging
OpenFeature
flagsmith
Flipt
split

Why Cluster Computing?

- Single computer has limited hardware upgradability.
- Expand sideways, not downwards.
- Netflix serving content from a single server is nuts.
- Redundancy from having several servers.
- Use of clusters ≡ microservices architecture.
- Better upgradability, availability and portability

Demo Time

Thank You!

alexbissessur.dev

t.me/alexbissessur

moris.social/@AlexB