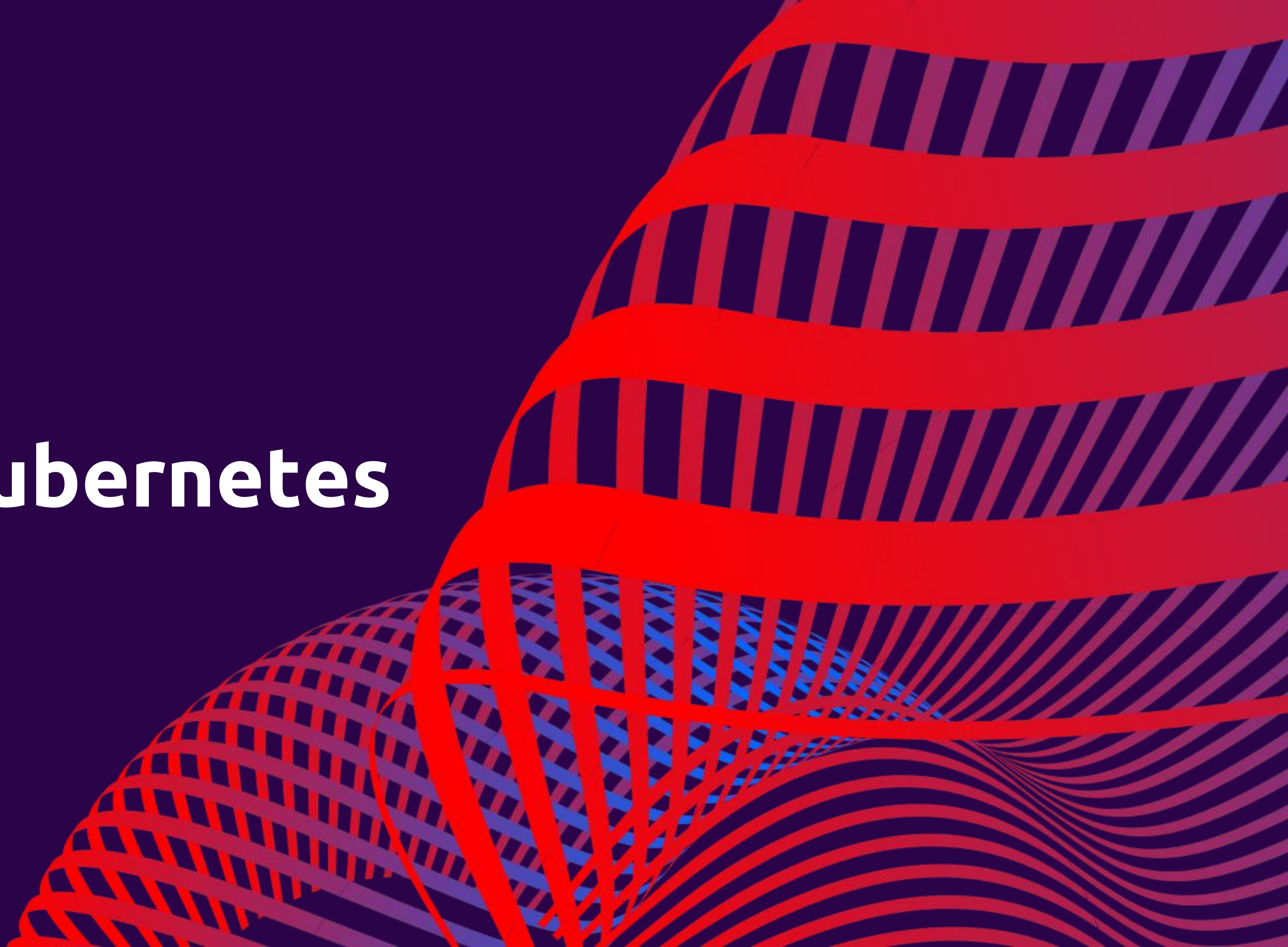
kubernetes



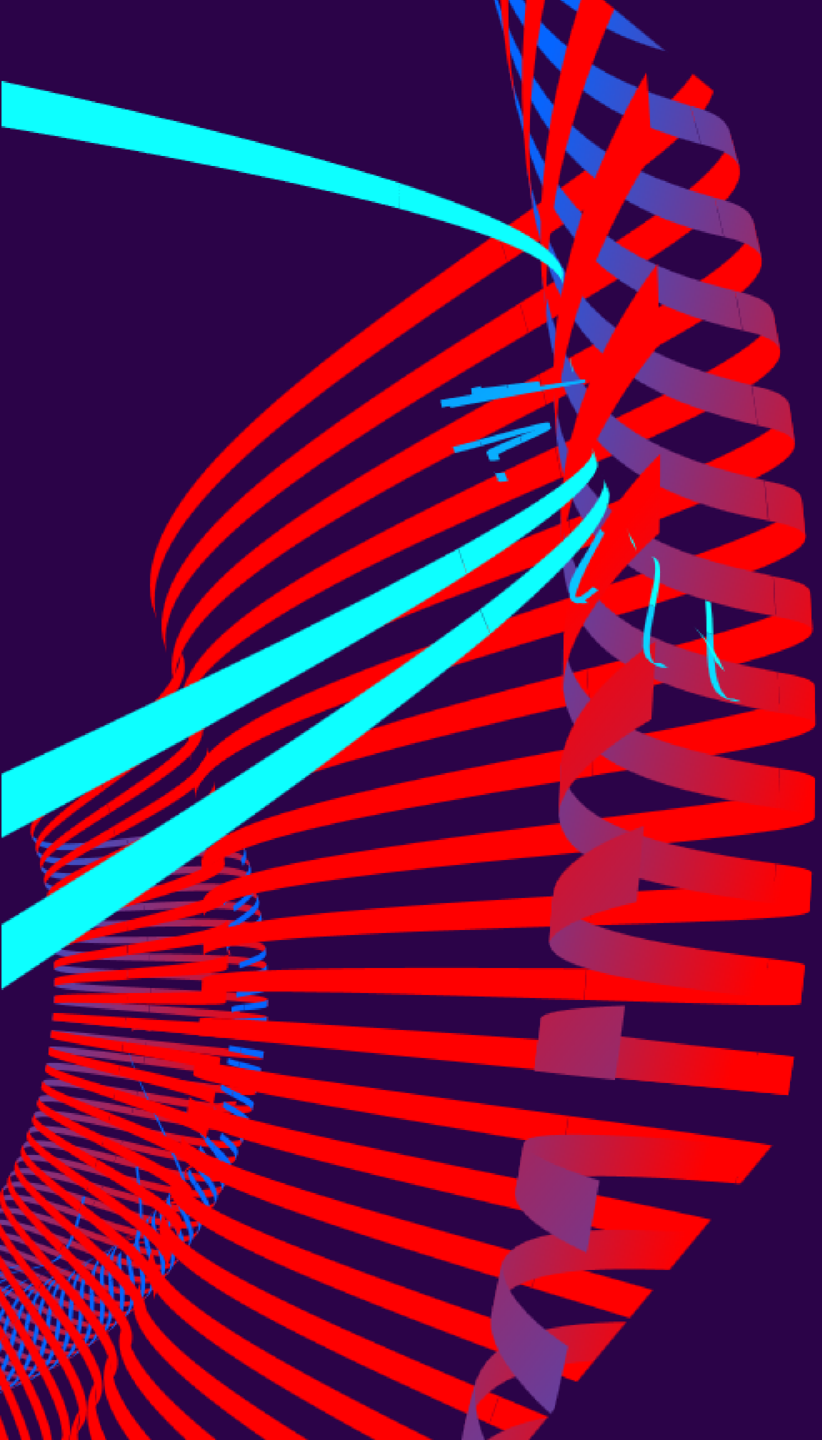
Agenda

I. Why the need for Kubernetes
II. Kubernetes competitors
III. How to work with Kubernetes
IV. What is the Kubernetes architecture
V. How to setup a Kubernetes cluster

κυβερνήτης

Kubernetes

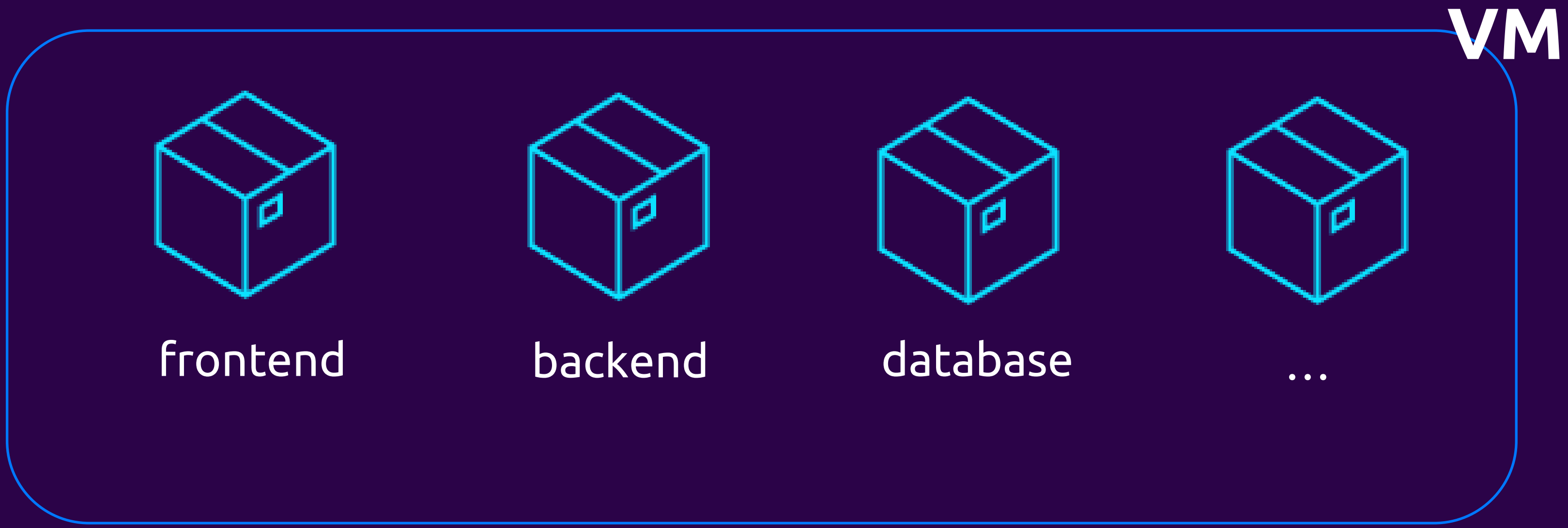
K8s



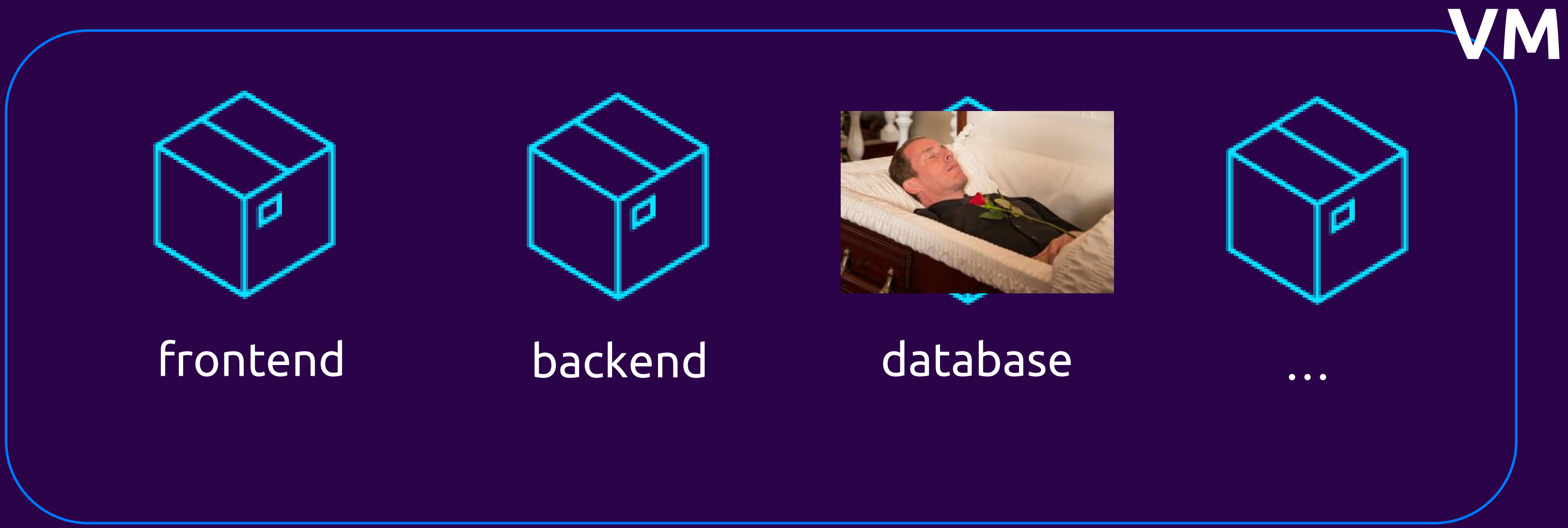
I. Why the need for K8s?

C

Small Application



Small Application

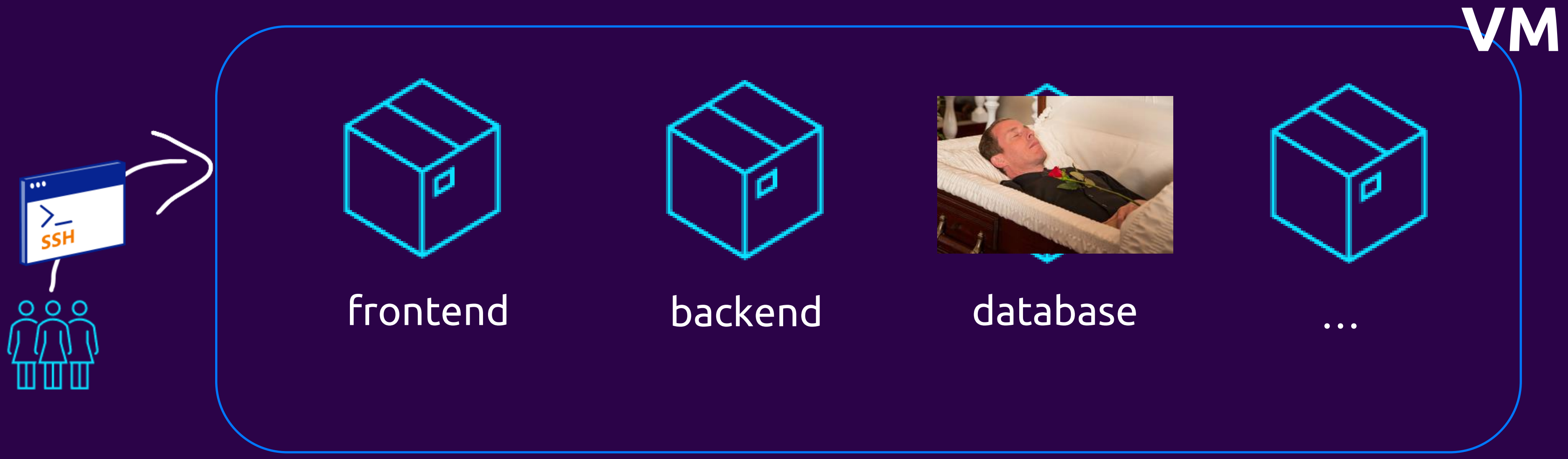


S

VM

fr

Small Application



Sma n

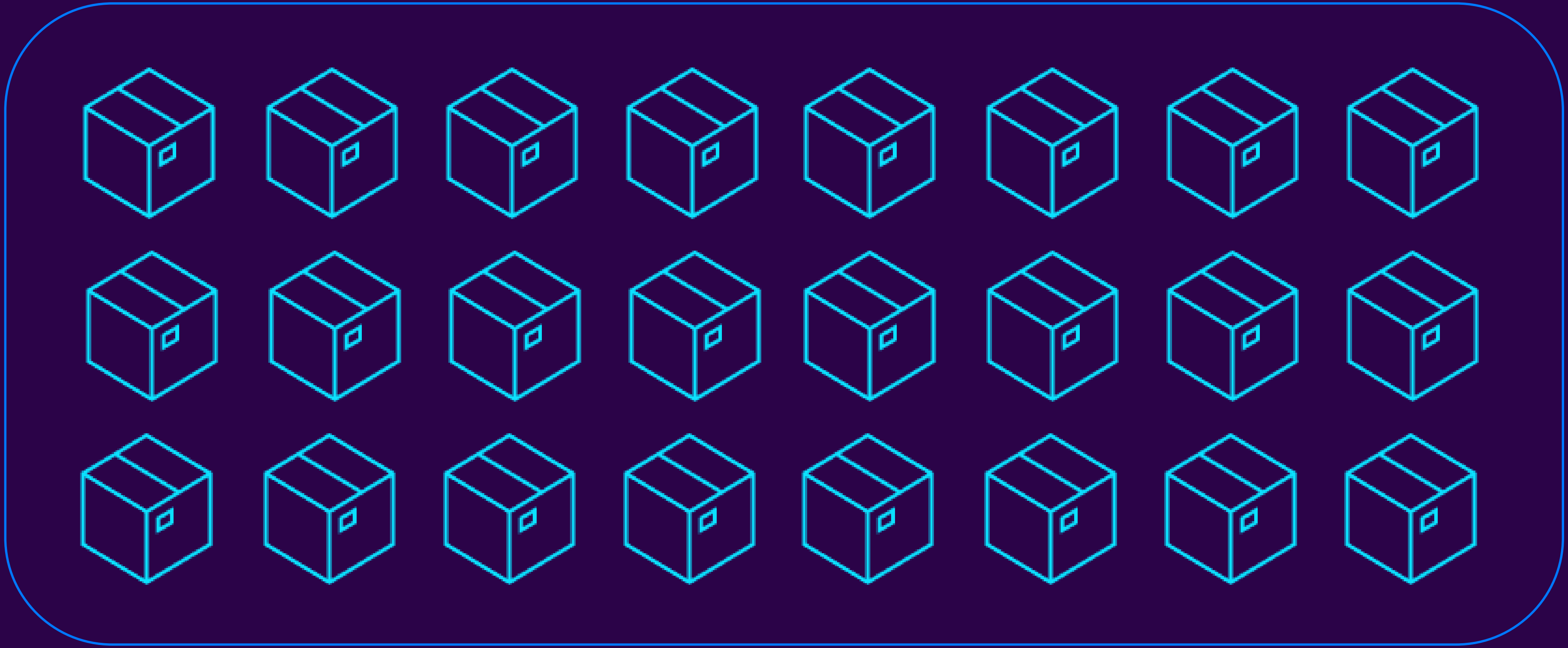
VM
SSH
frontend
...

Small Application

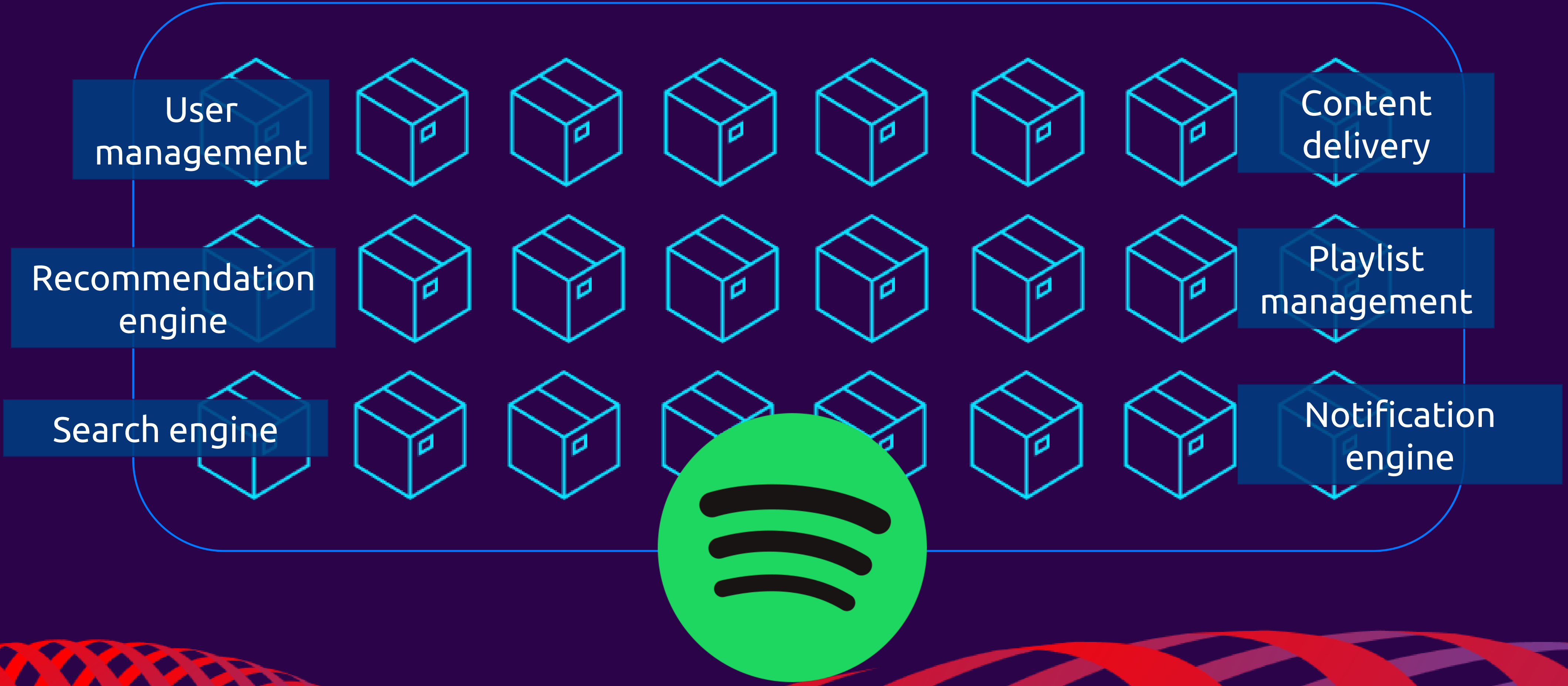
What to use for smaller applications?

- Docker compose
- VM
- VPS
- nothing, just bare-metal machine

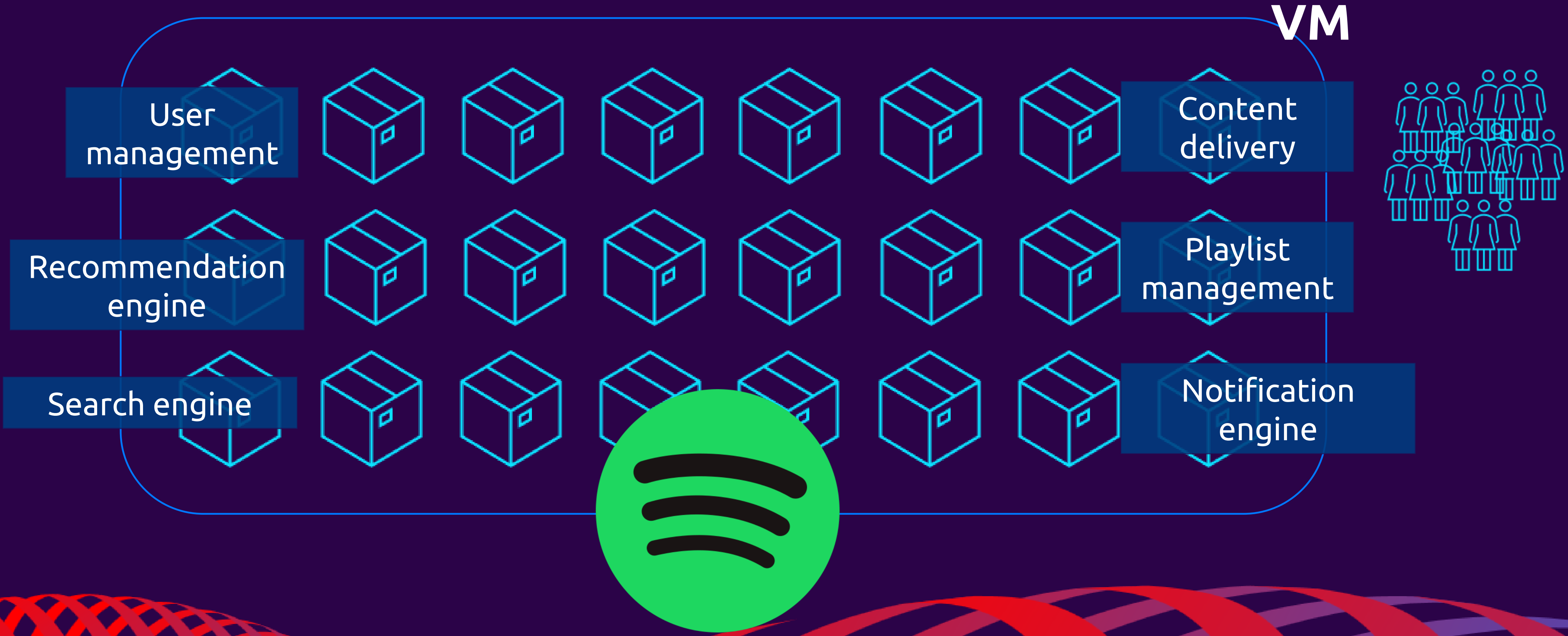
Enterprise Application



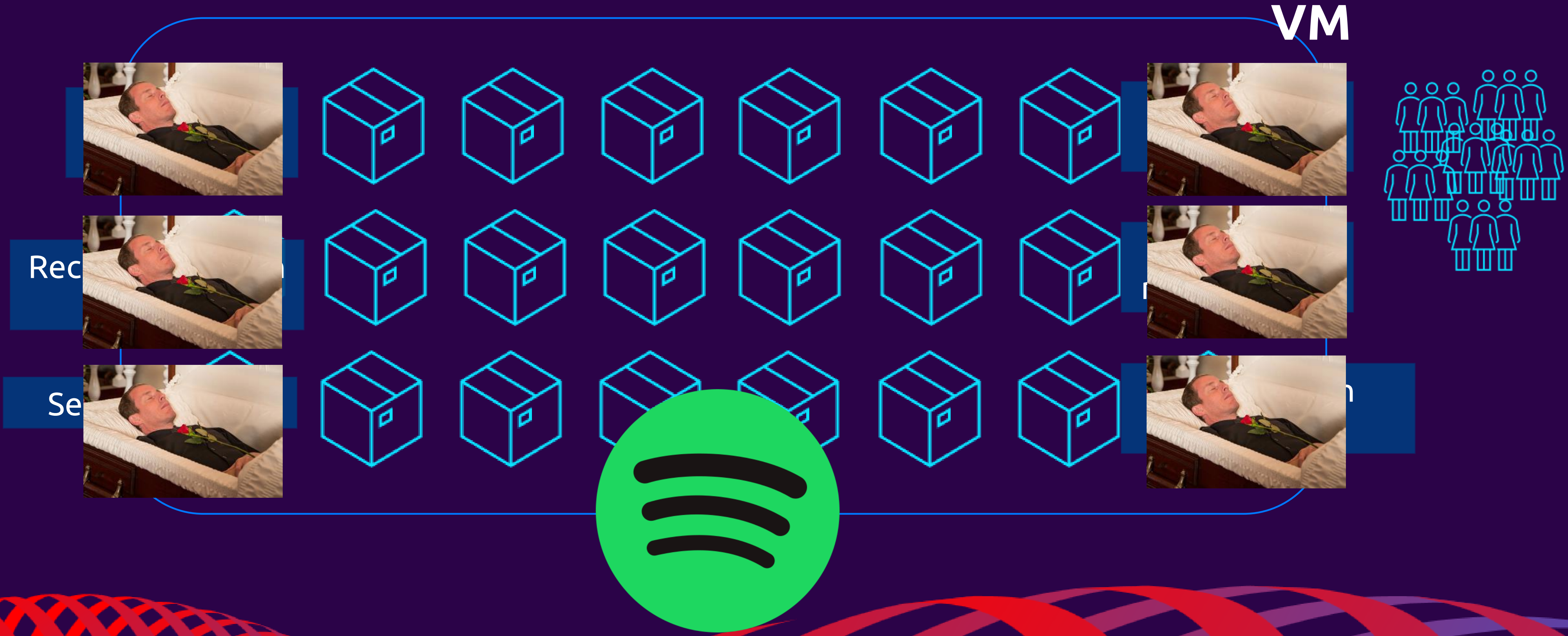
Enterprise Application



Enterprise Application



Enterprise Application



Ent ion

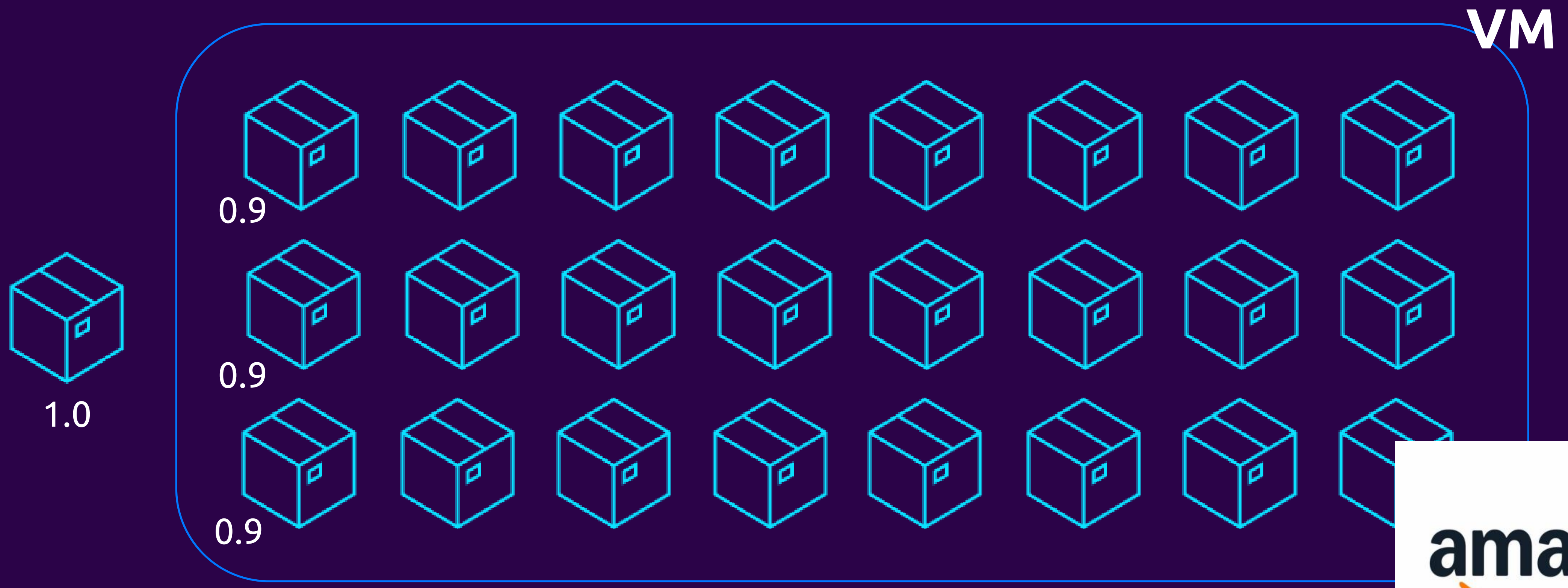
VM

Rec

Se

How can you make sure the application is up and healthy all the time, or with minimal intervention?

Enterprise Application



amazon

How can you make sure all containers are updated? Do it manually?

Who takes care of

resource management?

security?

scalability?

high availability?

fault tolerance?

load balancing?

or

K8s managed service

Amazon EKS

Google Kubernetes Engine

Azure Kubernetes Service

Amazon Elastic Kubernetes Service

IBM Cloud Kubernetes Service

RED HAT® OPENSHIFT Container Platform

or

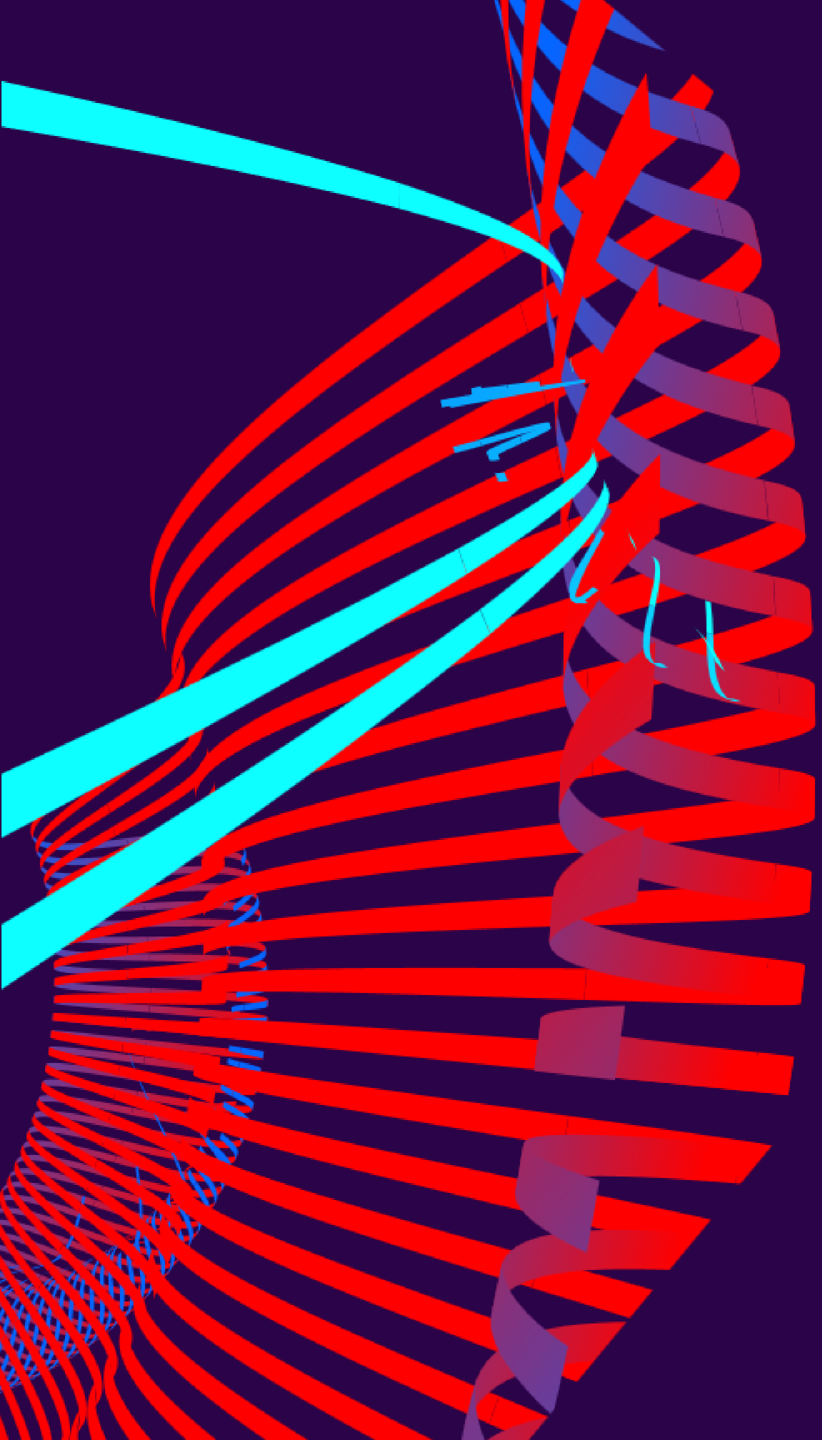
Container orchestration system

kubernetes

HashiCorp Nomad

Apache MESOS™

RANCHER® BY SUSE

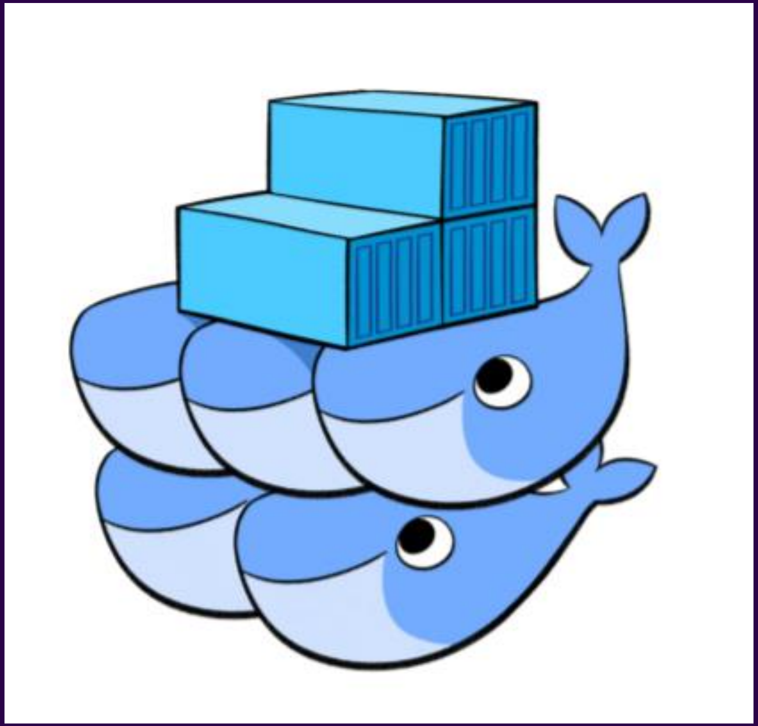


II. K8s competitors?

kubernetes

- Open source, developed by Google released in 2014
- Manage applications in different environments
- Improves reliability
- Reduces the time and resources of DevOps
- Scheduling scaling
- Managing health checks
- declarative model
- Flexibility in multi-cloud and hybrid cloud environments
- Challenges when managing and scaling containers between cloud providers

Docker Swarm



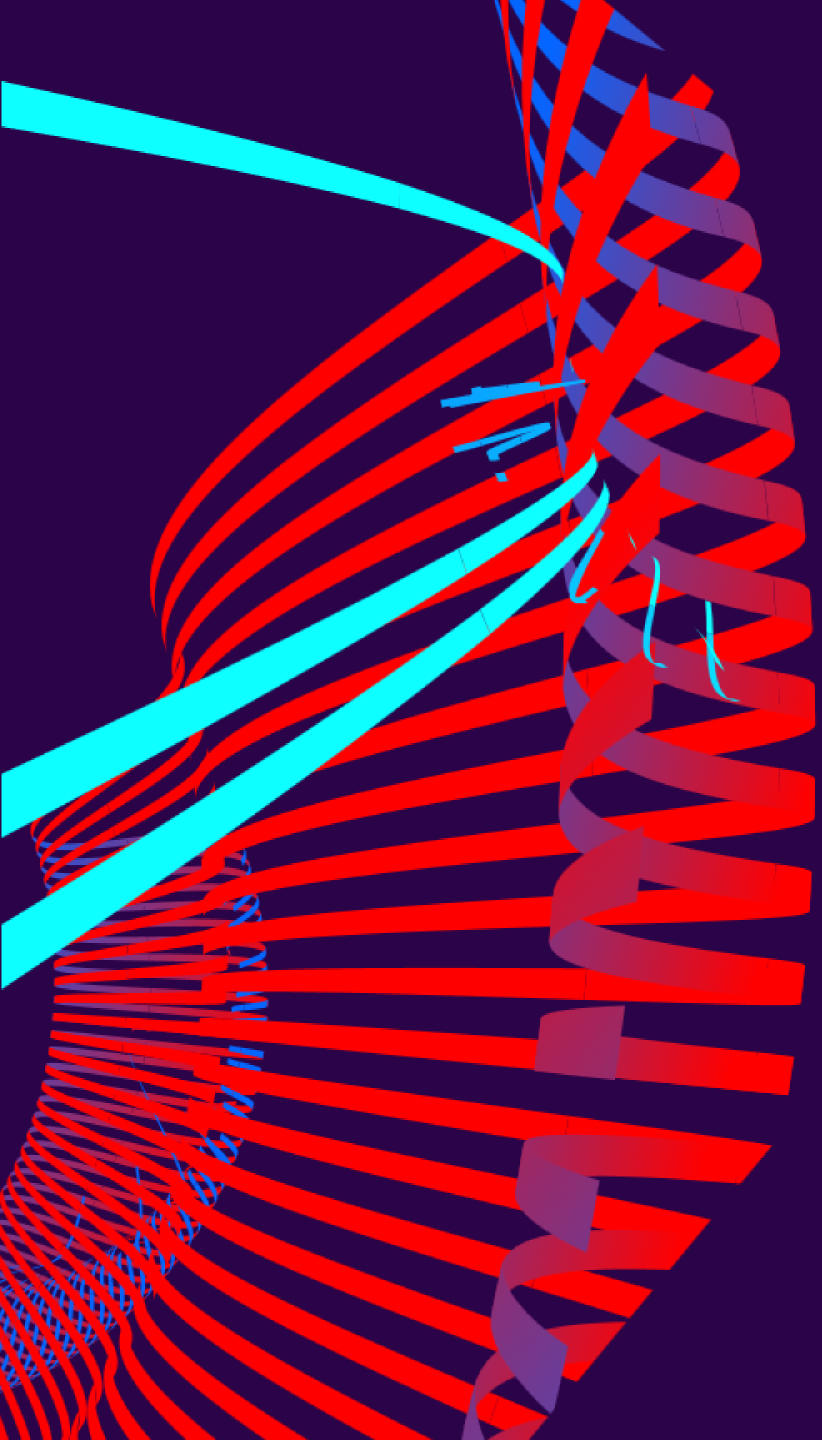
- b

- simplicity with fast deployments
- Docker Engine provides the option of integration with Kubernetes
- You do not need to re-architect your app to adapt to other container orchestrators
- Ideal for smaller deployments
- Automated load balancing within the Docker containers
- No option to connect containers to storage, less user-friendly when it comes to storage-related issues
- Less robust automation capabilities

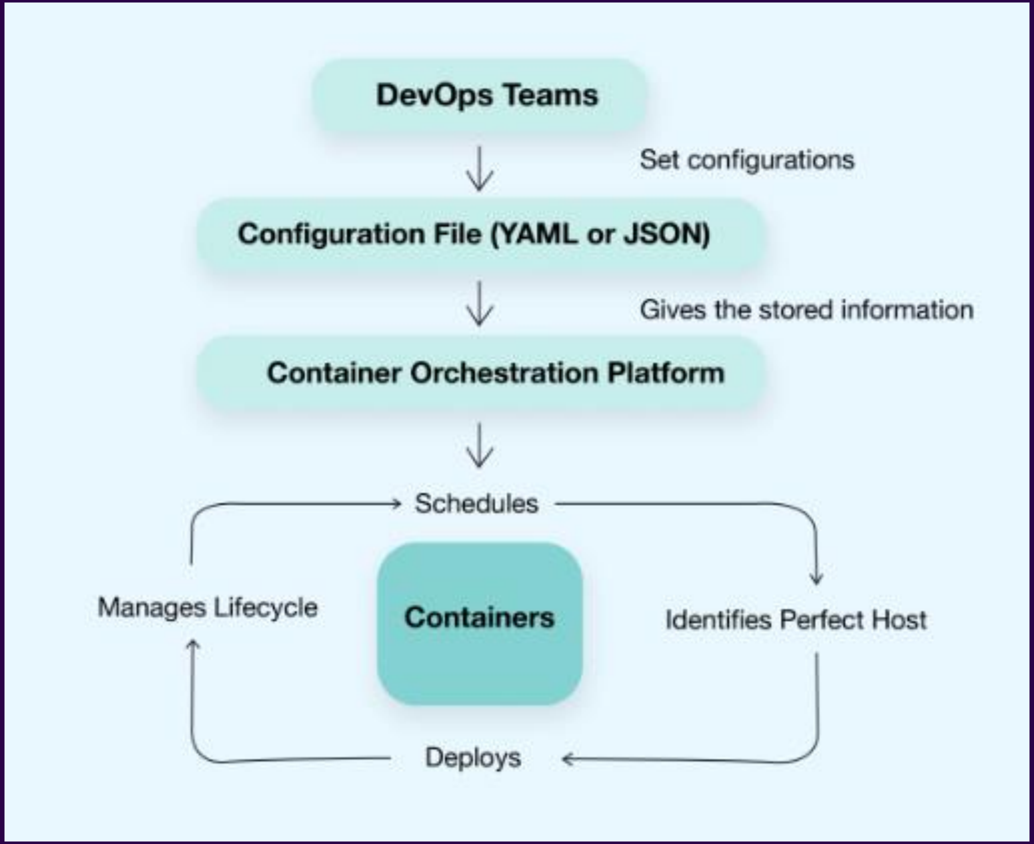
Apache Mesos

- b

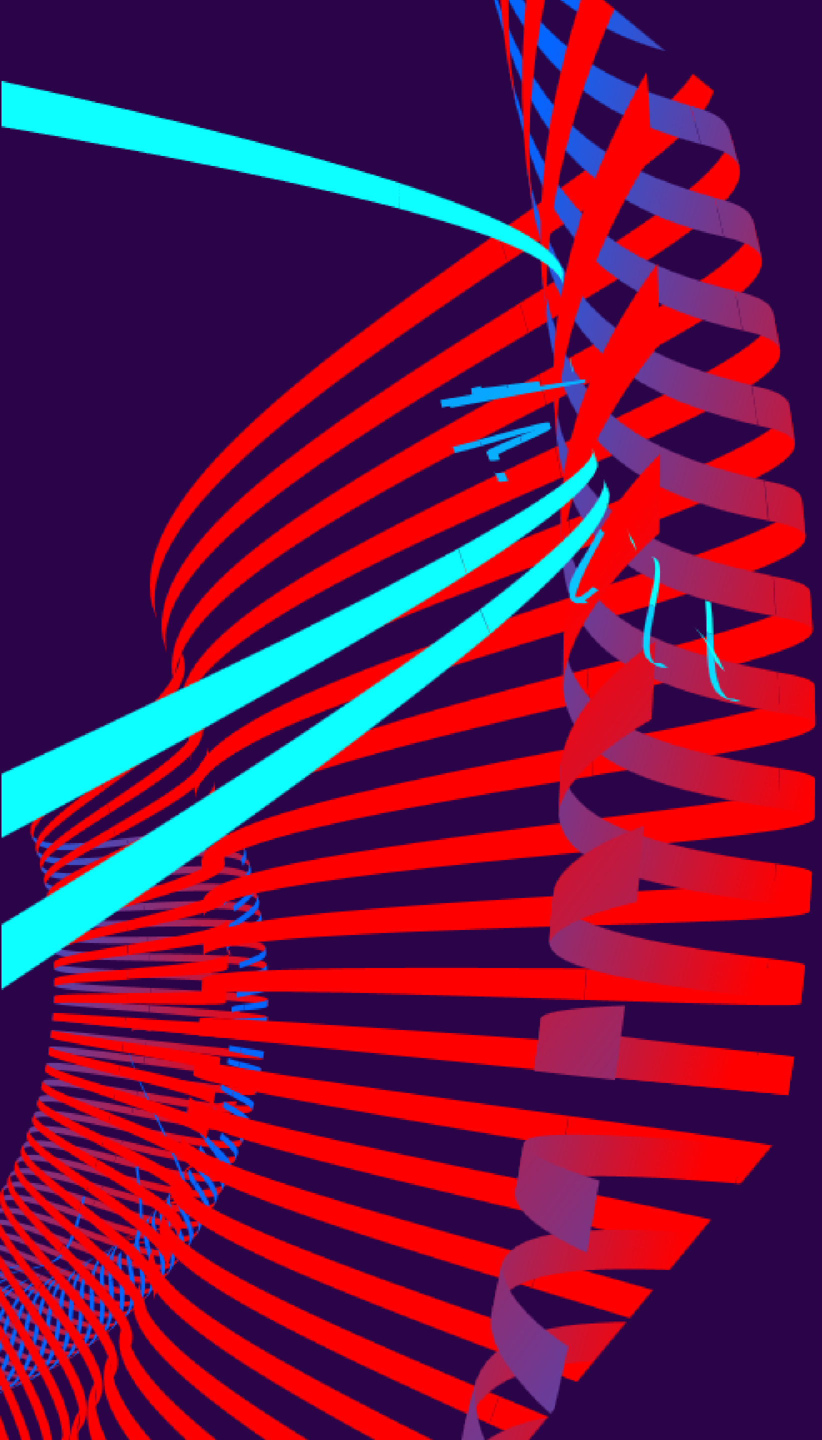
- older than Kubernetes
- open-source cluster
- Lightweight interface
- APIs support popular languages like C++, Java, and Python
- only provides management of the cluster -> it is **not a container orchestration system**, therefore, many frameworks have been built on top
- has a high learning curve entails



III. How to work with K8s?



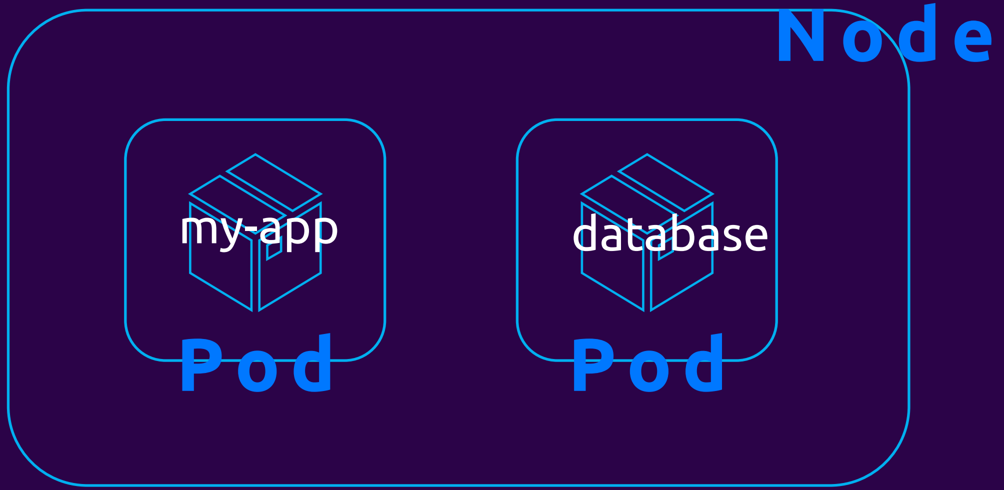
1. Requirement for a new container for cluster deployment,
2. K8 schedules an event & identifies the appropriate host based on specifications
3. K8s will manage its lifecycle based on the specifications defined
4. Vertical or horizontal scaling to spread the load across host infrastructure
   - Load balancing
   - Moving containers from one host to another if a host dies



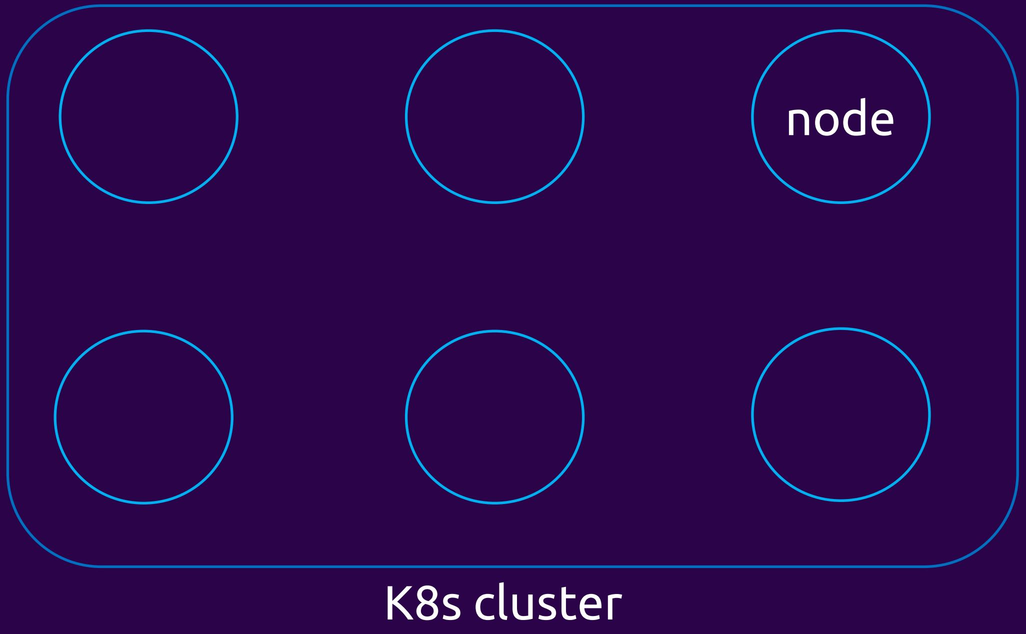
IV. What is the K8s architecture?

Node

A server (physical or virtual)



Nodes



Master (control plane)

- Controlls the cluster

Worker

- Does the work
- Amount depends on work to be done
- Has a „kubelet“ process running on it, makes it possible for the cluster to talk to each other

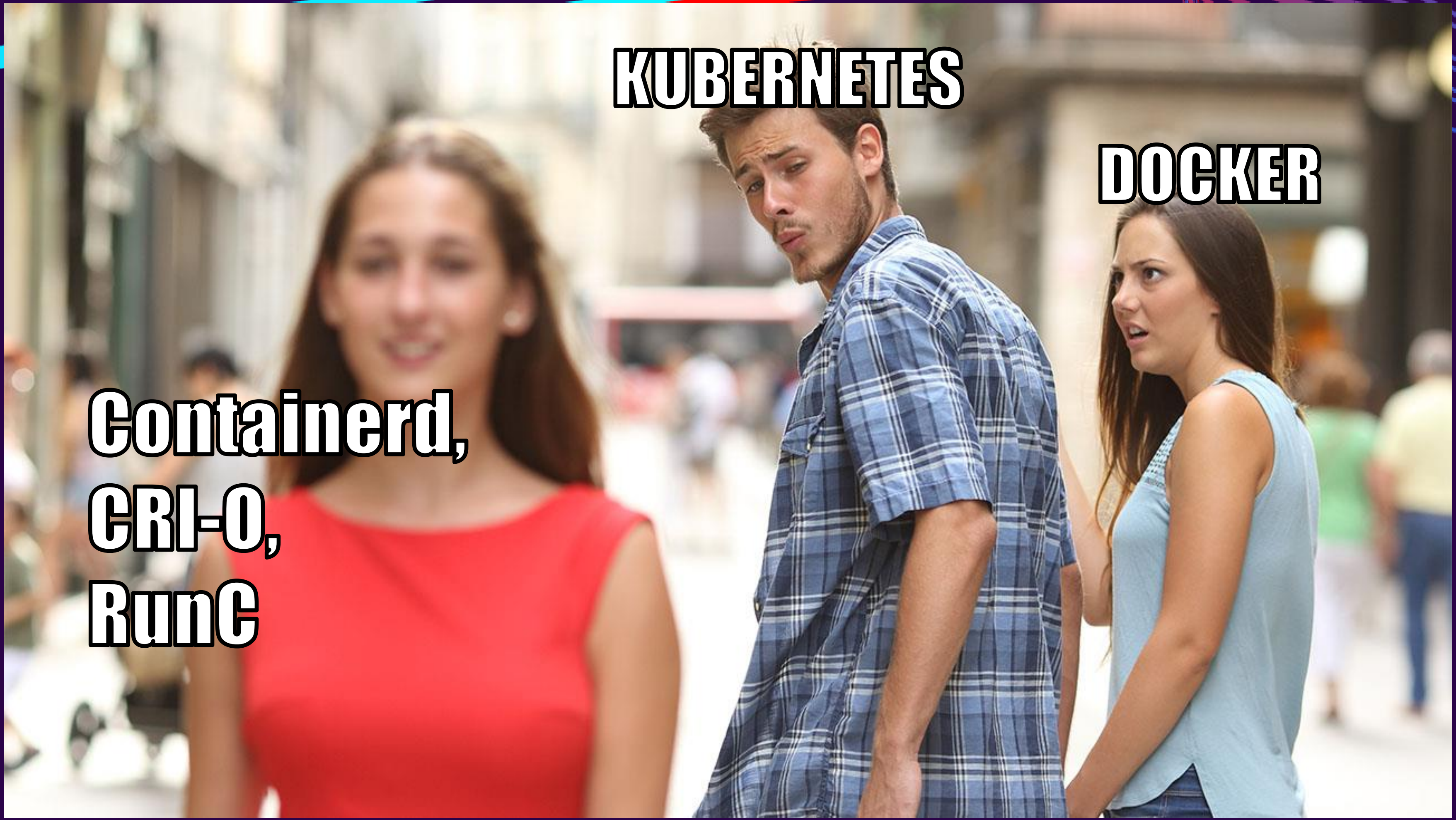
What exactly is running on master node?

- **API Server** (a container): the only **entrypoint** to cluster
- **Controller manager**: restarts a container if it dies, keep track of what is happening in the cluster
- **Scheduler**: scheduling containers on different nodes based on workload
- **Etcd**: key value storage (holds the current state of the cluster at any time, has all the config data, and all status data of each node and container)
- **Virtual network**: enables communication between master and worker, turns all the nodes inside of a cluster into one powerful machine that has the sum of all the resources of nodes)

Is one master node enough?

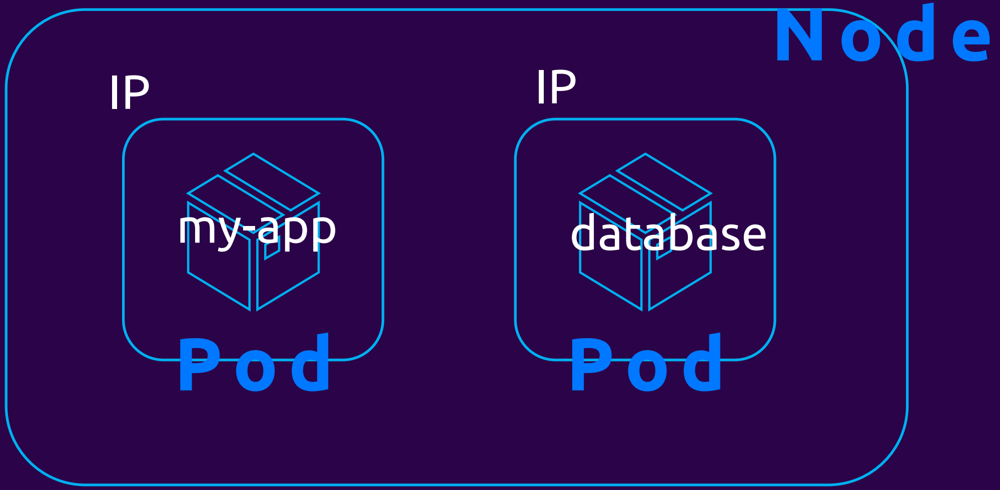
Pod

- Smallest unit in K8s,
- Abstraction over container
- Meant to run 1 application container inside of it


KUBERNETES
DOCKER
Containerd,
CRI-O,
RunC

How do they communicate with each other?

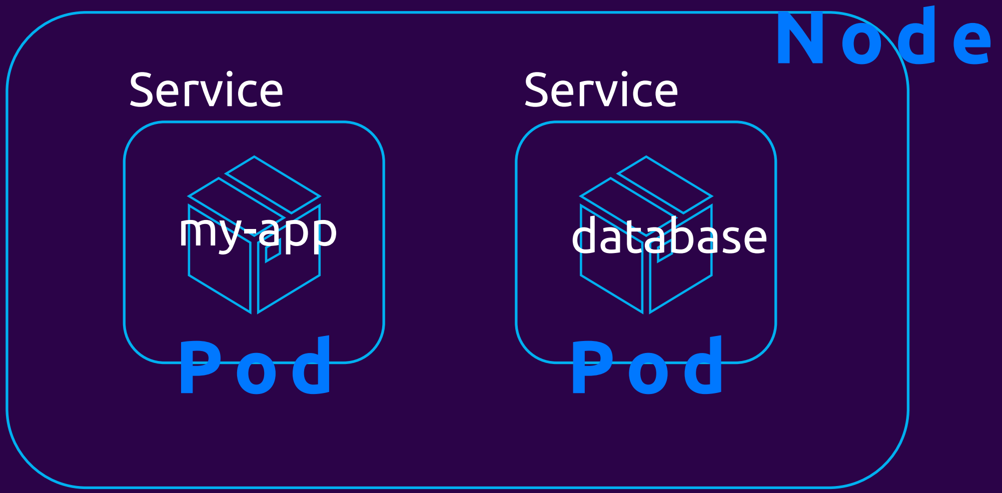
- Virtual network
- Each pod gets their own IP (not the container!)
- Pods can die pretty easily ->a new pod gets created and is assigned a new IP address



But if you are communicating with the database pod using the IP address you need to adjust it every time pod restarts...

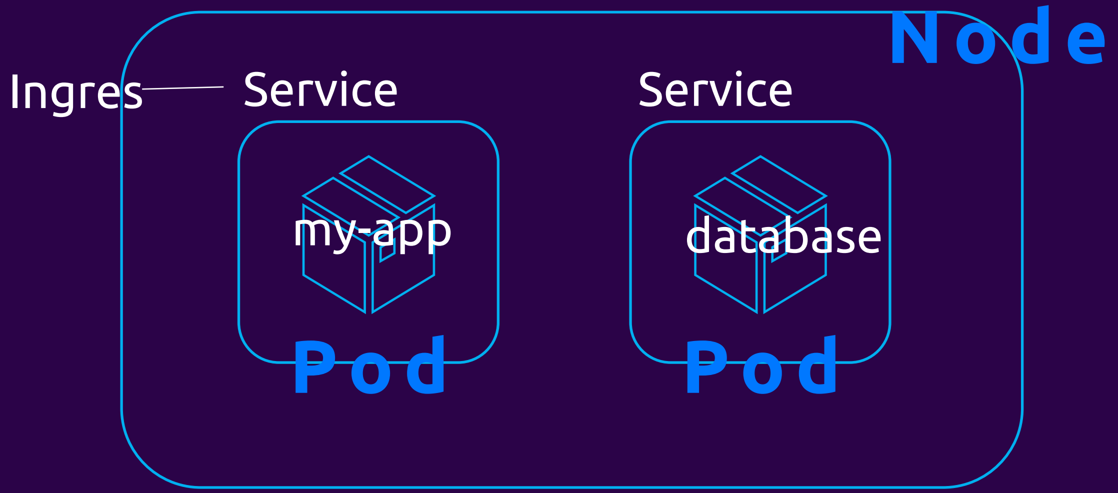
Service

- A permanent IP that can be attached to a pod
- Lifecycles of service and pod are not connected
- App accesible through web broweser? External service



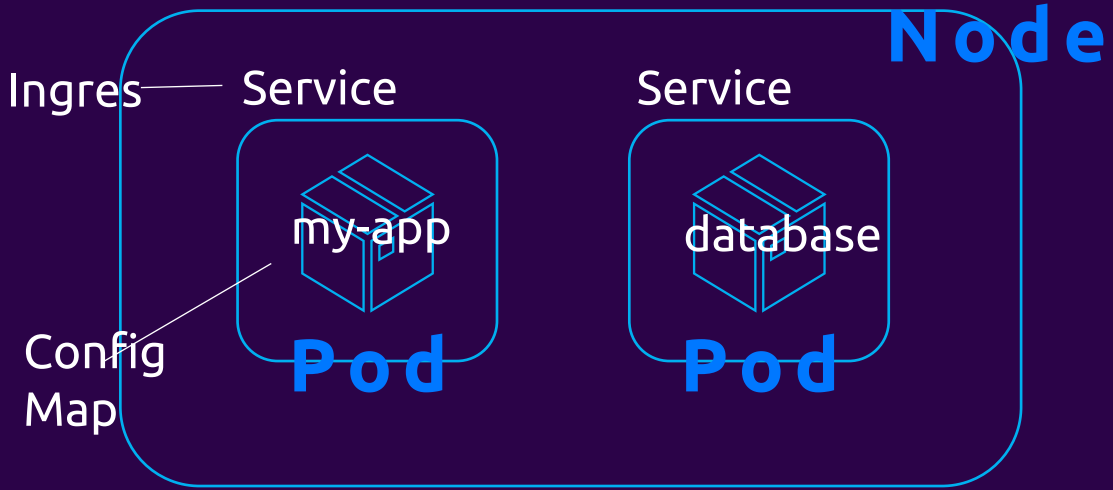
Ingress

- Gives URL a secure protocol and a domain name
- The outside request goes to ingres and the ingres forwards to the service



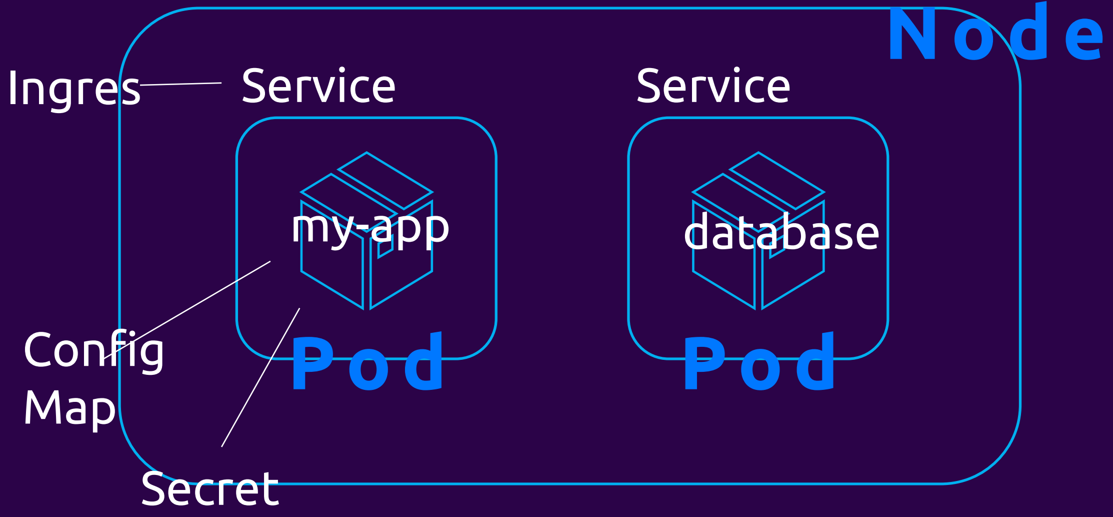
ConfigMap

- Pods communicate with other other using service
- The communication endpoints are configured inside of the image
- If the endpoint changes you have to rebuild image, push, pull in your pod and restart the whole thing
- configMap: contains config data like endpoints and you just connect it to the pod



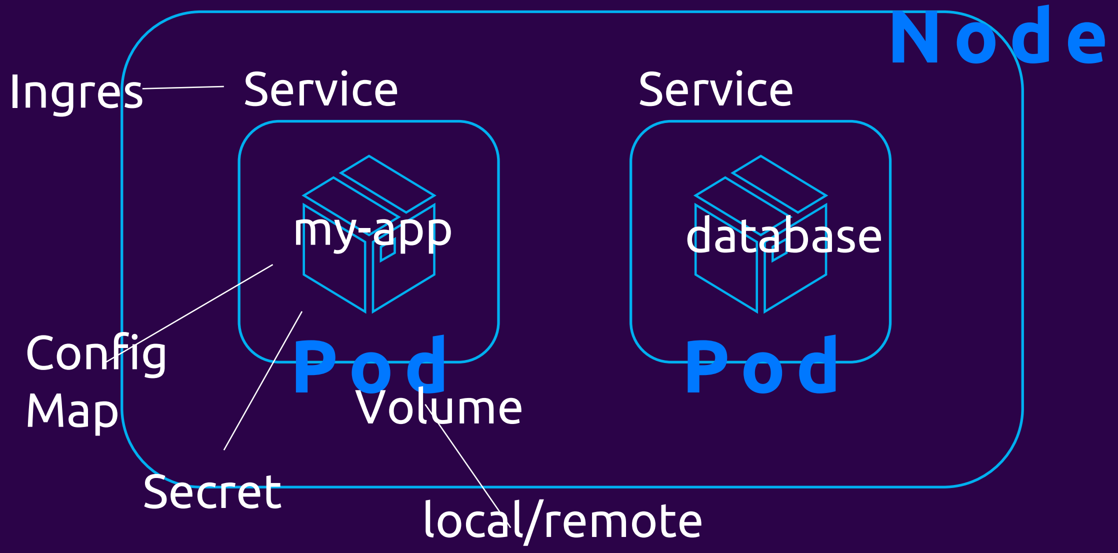
Secret

- like configmap but stores secret data & stored in base64 encoded format
- BUT these secrets are stored **unencrypted** in the API server‘s **etcd**, anyone with API access and etcd can access themYou need 3rd party tools or cloud provider tools to encrypt



Volumes

- K8 does not manage data persistance
- Vol attaches a physical storage on a hard drive to your pod

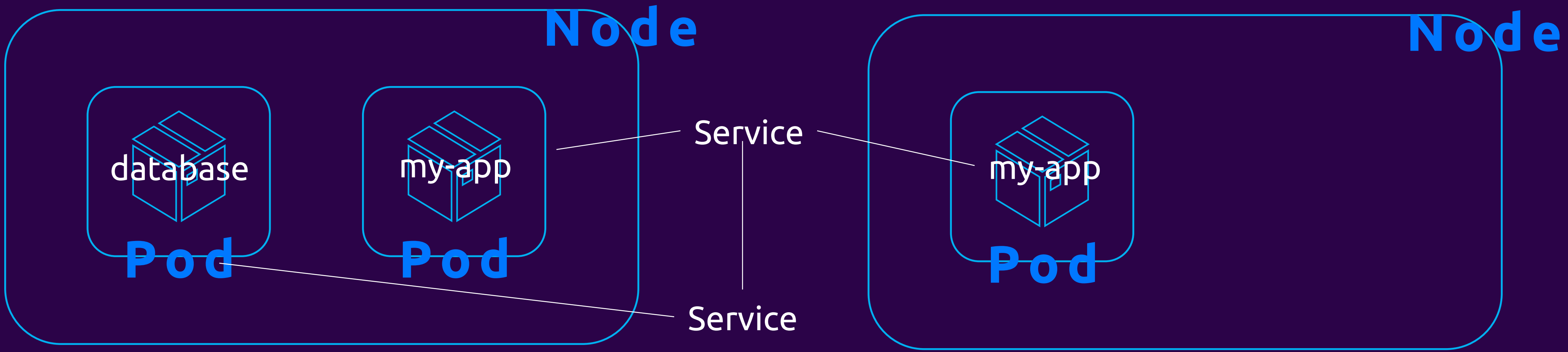


What happens if my application pod dies? Crashes or i have to restart it bc there is a new image

Downtime

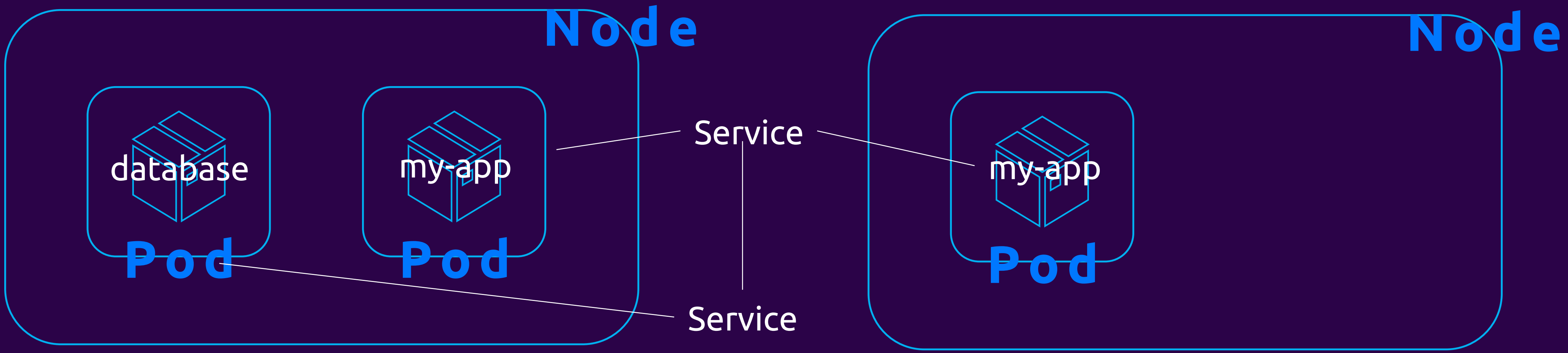
Replicas

- Replicate everything on multiple servers (nodes)
- Replica is connected to the same service
- Service is also a load balancer, it will catch the request and forward it to whichever pod is leat busy

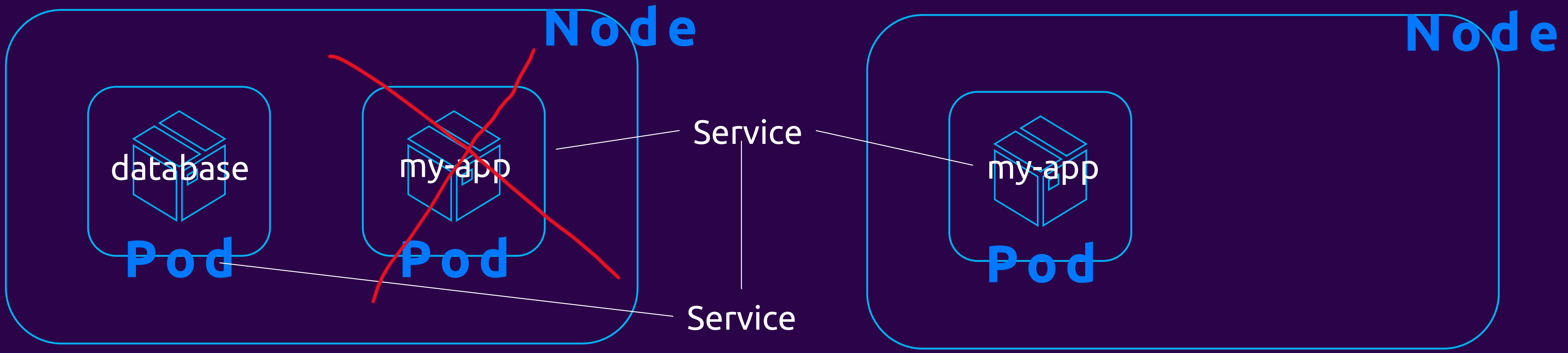


Deployment

- You will not be creating pods, you will be creating deployments
- Abstraction of pods making it more convenient to replicate them and interact with them



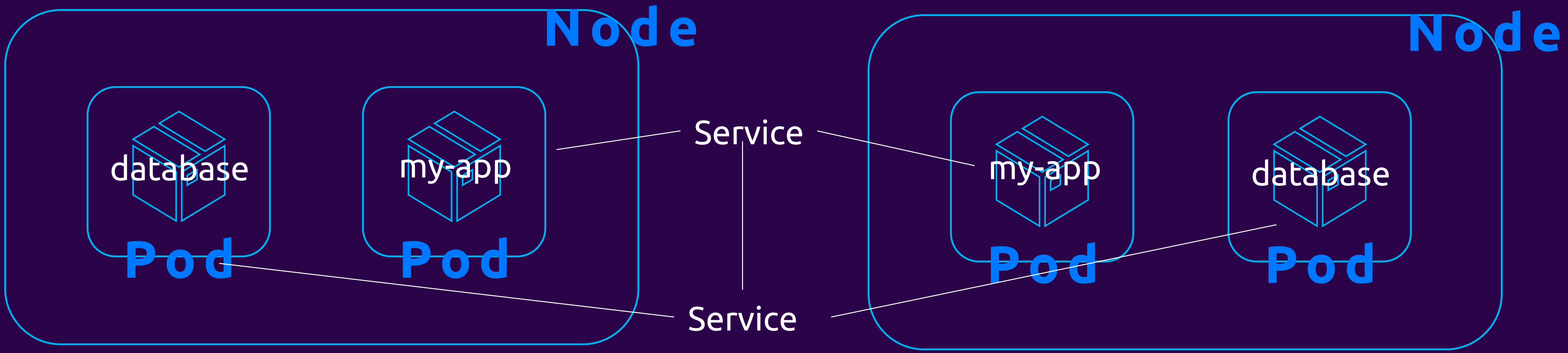
So what would happen now if one of the replicas dies?

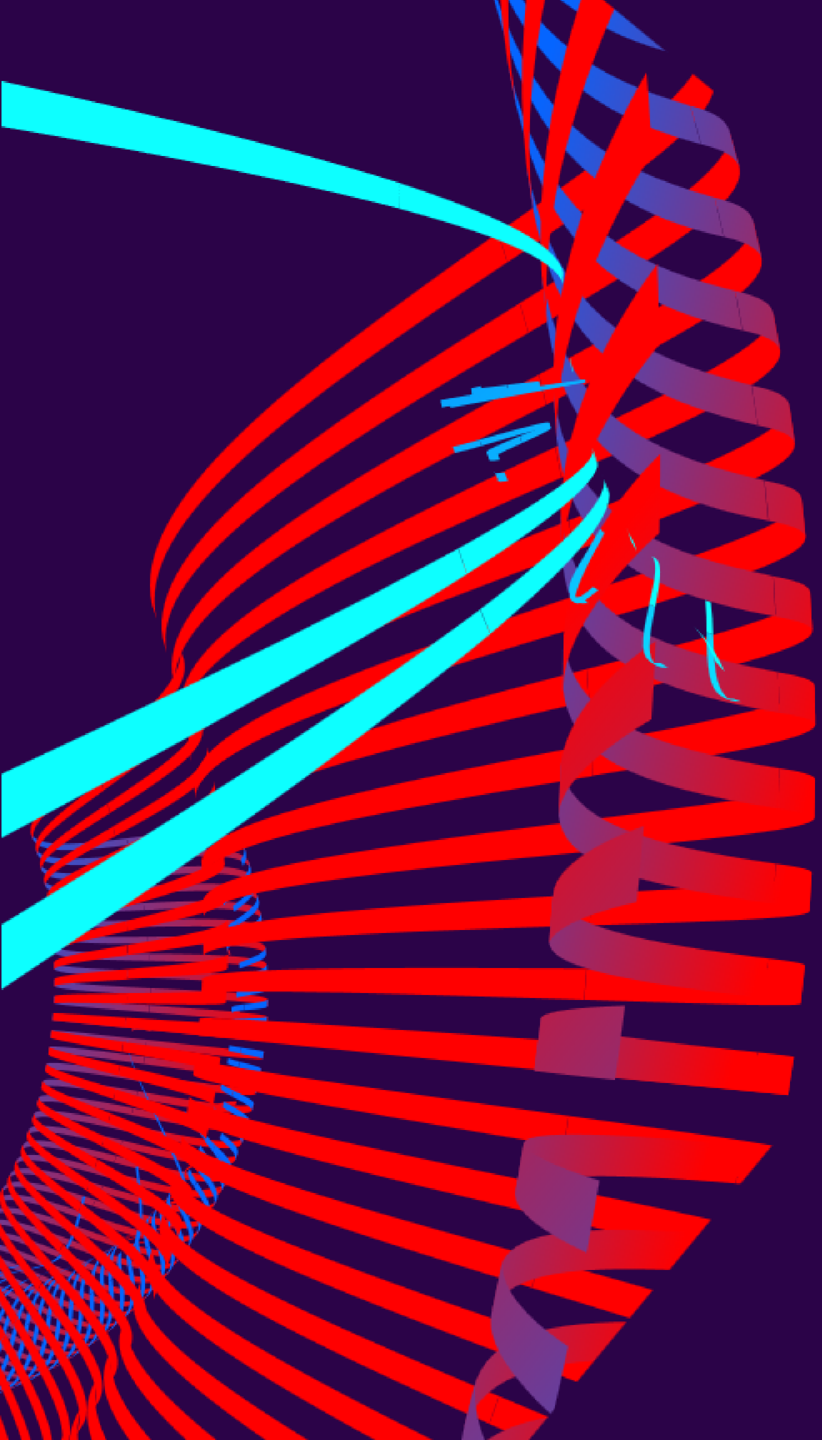

Node
database
Pod
my-app
Pod
Service
Service
Node
my-app
Pod

What if the database dies?

StatefulSet

- The database cannot be replicated using the deployment bc the db has state
- StatefulSet: clones access the same shared data storage & avoid data inconsistencies
- Host the database outside K8 cluster and have a stateless app communicating with it





V. How to setup a K8s cluster?

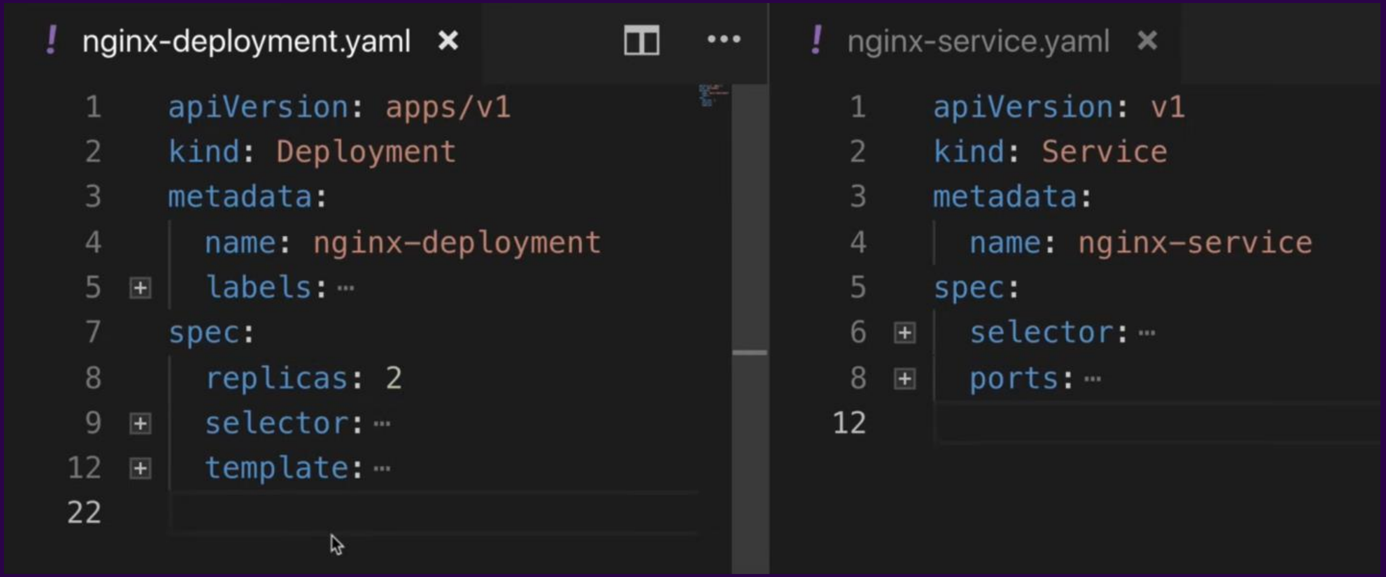
- All the configuration in K8 cluster goes through a master node with the process API server where config requests (deployment, service) are sent to.
- To talk to the API server, you can use
    - UI (K8 dashboard),
    - API (script, curl command),
    - CLI like kubectl they all talk to the API server and send their config requests.

K8s configuration file

1) Metadata
where the metadata of that component that you are creating resides

2) Specification
configs you want to apply to that component

3) Status
generated by K8s, K8s will compare what is the desired state in the specification and what is the actual state, K8s updates the state continuously



nginx-deployment.yaml

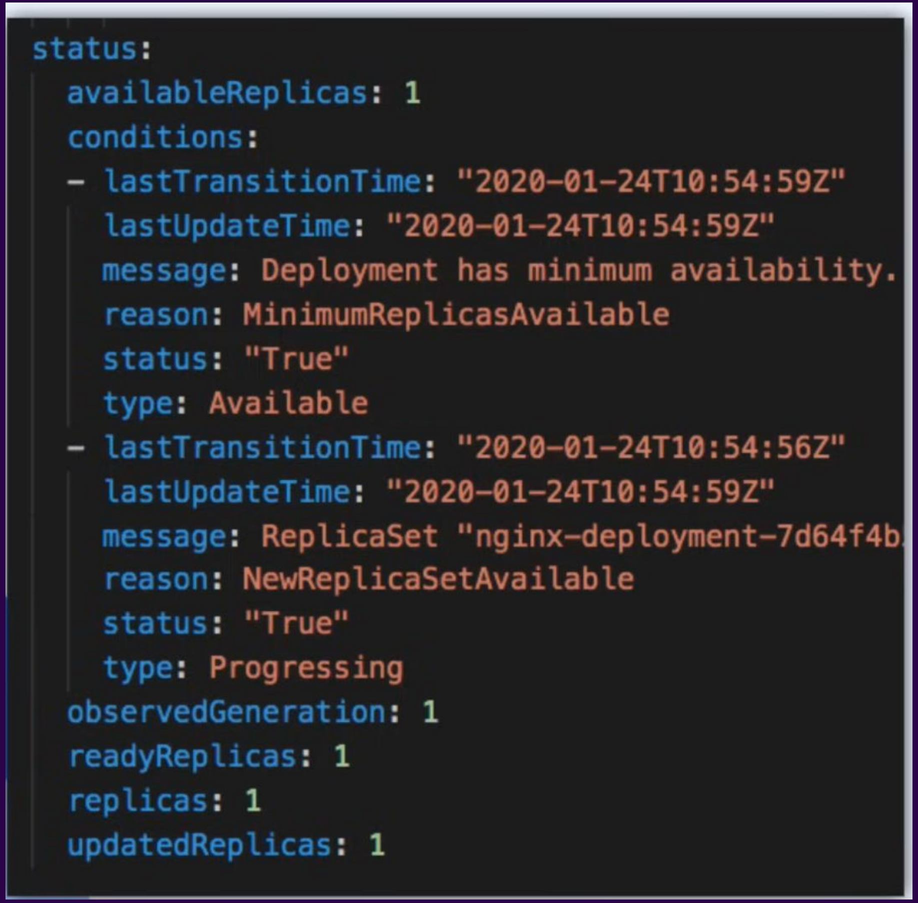
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels: ⋯
spec:
  replicas: 2
  selector: ⋯
  template: ⋯
```

nginx-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector: ⋯
  ports: ⋯
```

Where does k8 get this status data?

- from the etcd, the master process that stores the cluster data
- etcd holds at any time the current status of any K8s component & this is where the status info comes form



```
status:
  availableReplicas: 1
  conditions:
  - lastTransitionTime: "2020-01-24T10:54:59Z"
    lastUpdateTime: "2020-01-24T10:54:59Z"
    message: Deployment has minimum availability.
    reason: MinimumReplicasAvailable
    status: "True"
    type: Available
  - lastTransitionTime: "2020-01-24T10:54:56Z"
    lastUpdateTime: "2020-01-24T10:54:59Z"
    message: ReplicaSet "nginx-deployment-7d64f4b
    reason: NewReplicaSetAvailable
    status: "True"
    type: Progressing
  observedGeneration: 1
  readyReplicas: 1
  replicas: 1
  updatedReplicas: 1
```

Who can do the setup?

1. You from hand
2. Tool-supported, e.g., kops (kubernetes operations) with automations
3. Managed cluster – you do no setting up of the cluster, you are using someone elses, e.g., cloud providers
    - You pay for the worker & control plane nodes

THANK YOU FOR YOUR ATTENTION

REPO:

GITHUB.COM/ZEZL7/ESD-2024-KUBERNETES