



Devoxx France 2023



Kubernetes, dépassionné et pour les ultra débutants

Deep Dive



Speakers



**Sébastien
Blanc**

DevRel
Aiven
@sebi2706



**Horacio
Gonzalez**

DevRel
OVHCloud
@LostInBrittany



**Sun
Tan**

Senior Software Engineer
Red Hat
@__sunix__



Devoxx France 2023

AGENDA



INTRODUCTION

Why Kubernetes
Containers
What is Kubernetes?

1 - DIVE INTO K8S BUILDING BLOCKS

Playing with kubectl
YAML

2 - BEING A GOOD CLOUD NATIVE CITIZEN

Requests and limits
Health probes
ConfigMap and Secrets

3 - ADVANCED K8S

Persistent Volumes
jKube
Tolerance and taints
Operators

Quizz Kaoot : Le livre d'Aurélie



Devoxx France 2023

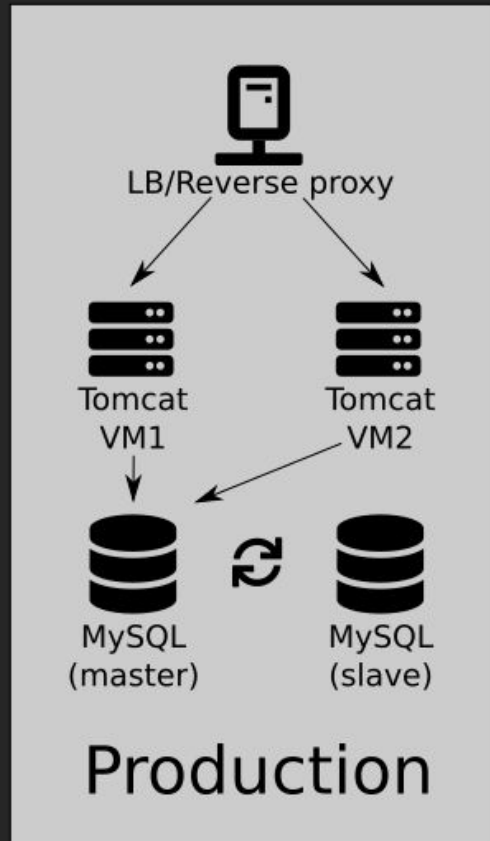
Introduction

WHY KUBERNETES?

CONTAINERS

WHAT IS KUBERNETES?

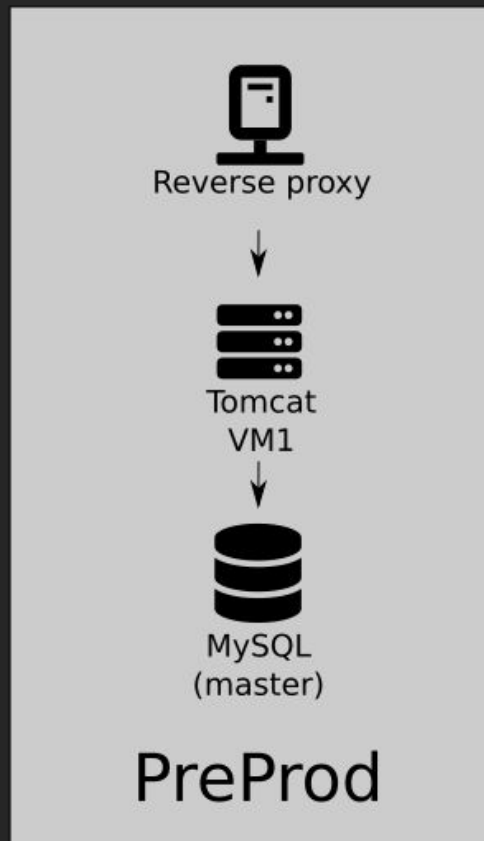
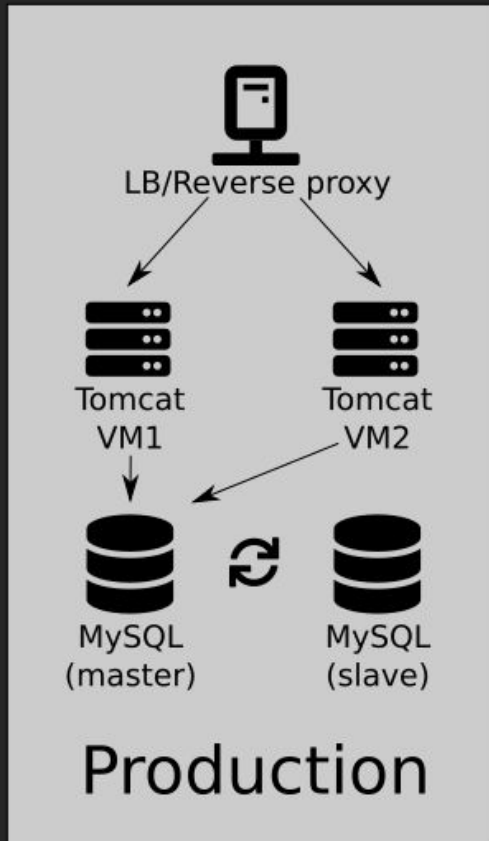
Why k8s?



A typical Java application

Based on real life experiences

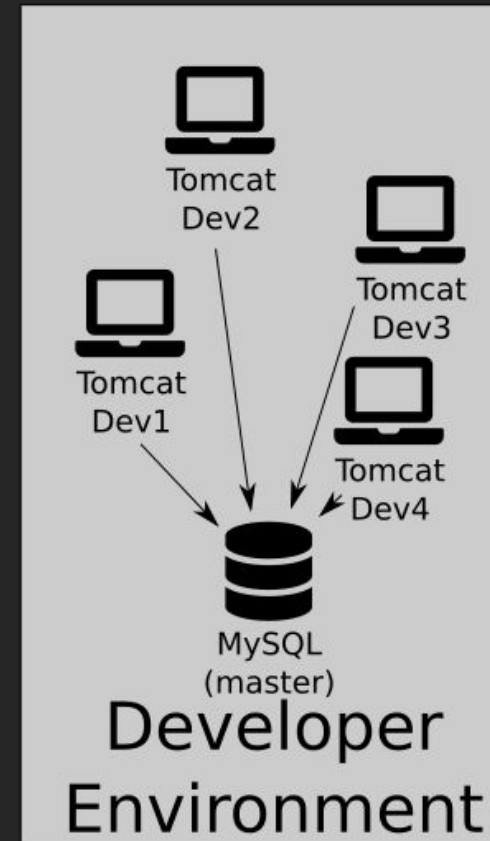
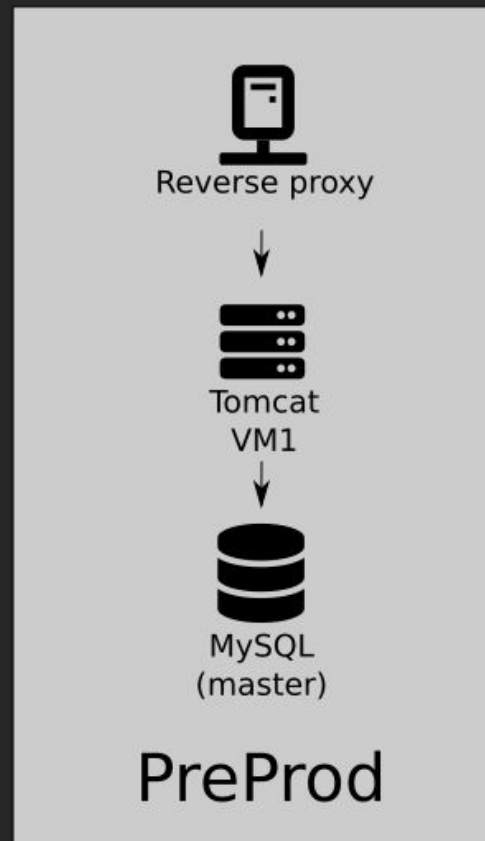
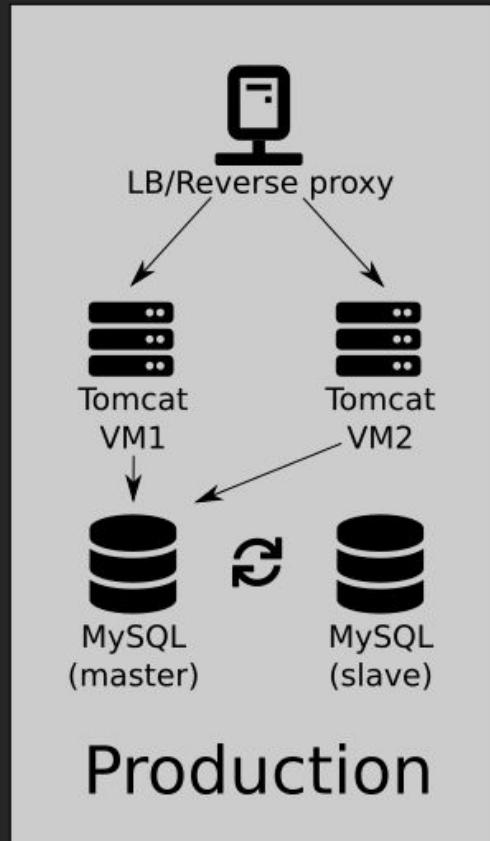
Why k8s?



A typical Java application

Based on real life experiences

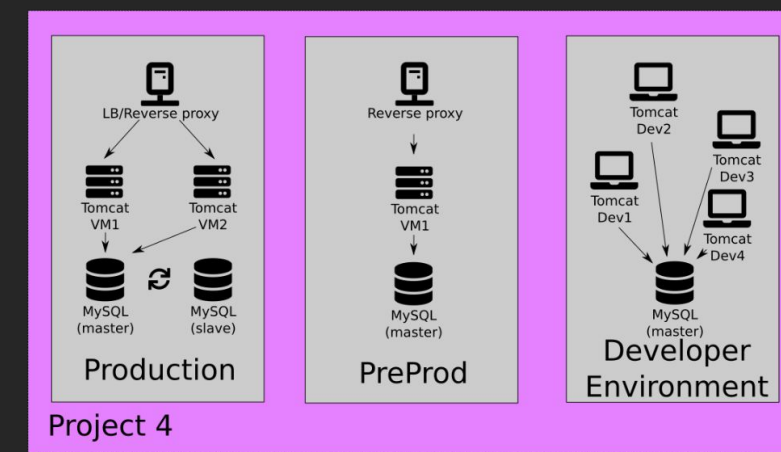
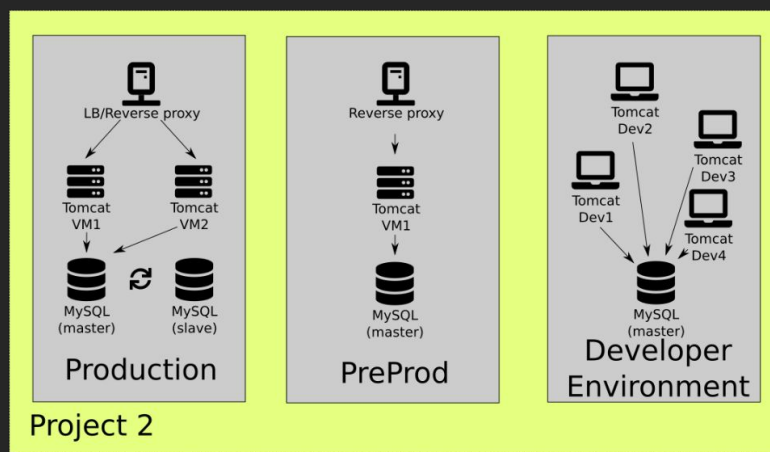
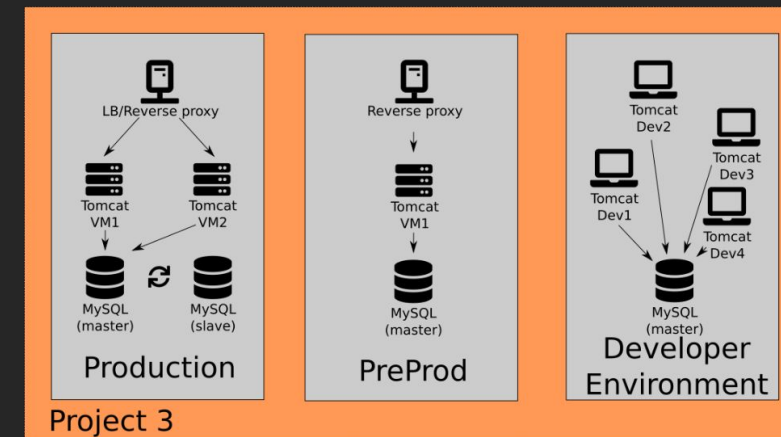
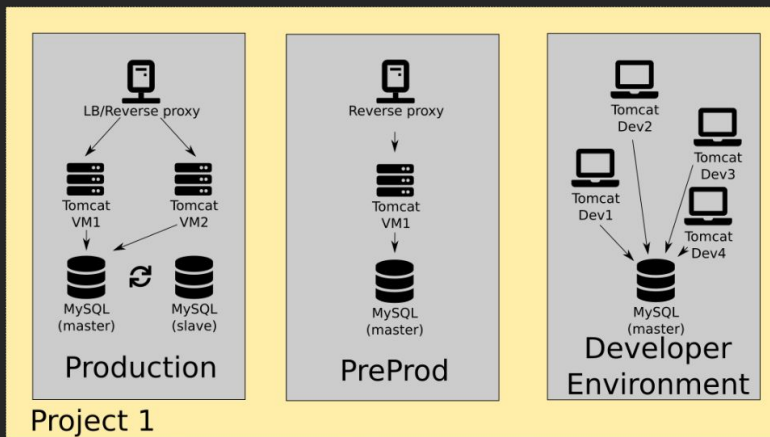
Why k8s?



A typical Java application

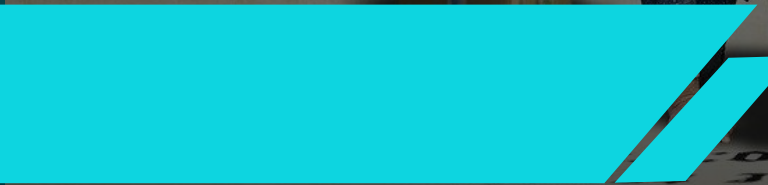
Based on real life experiences

Why k8s?



A typical Java application

Based on real life experiences



Pain point #1
MANUAL
DEPLOYMENTS



Pain point #2
SCALING



Pain point #3
DEVELOPER
ENVIRONMENT

The background is a dark blue gradient with a complex, glowing white circuit board pattern. A light blue cloud icon is centered in the upper half of the image. On the left side, there is a bright cyan horizontal bar with a diagonal cut-off on its right end.

Kubernetes

To the rescue!

Why k8s?



Think Big,
Start Small,
Scale Learn Fast

—
Jim Carroll

Why k8s?

Start small with Containers



- Containers are used in Kubernetes
- Containers could be used without Kubernetes



Devoxx France 2023

Introduction

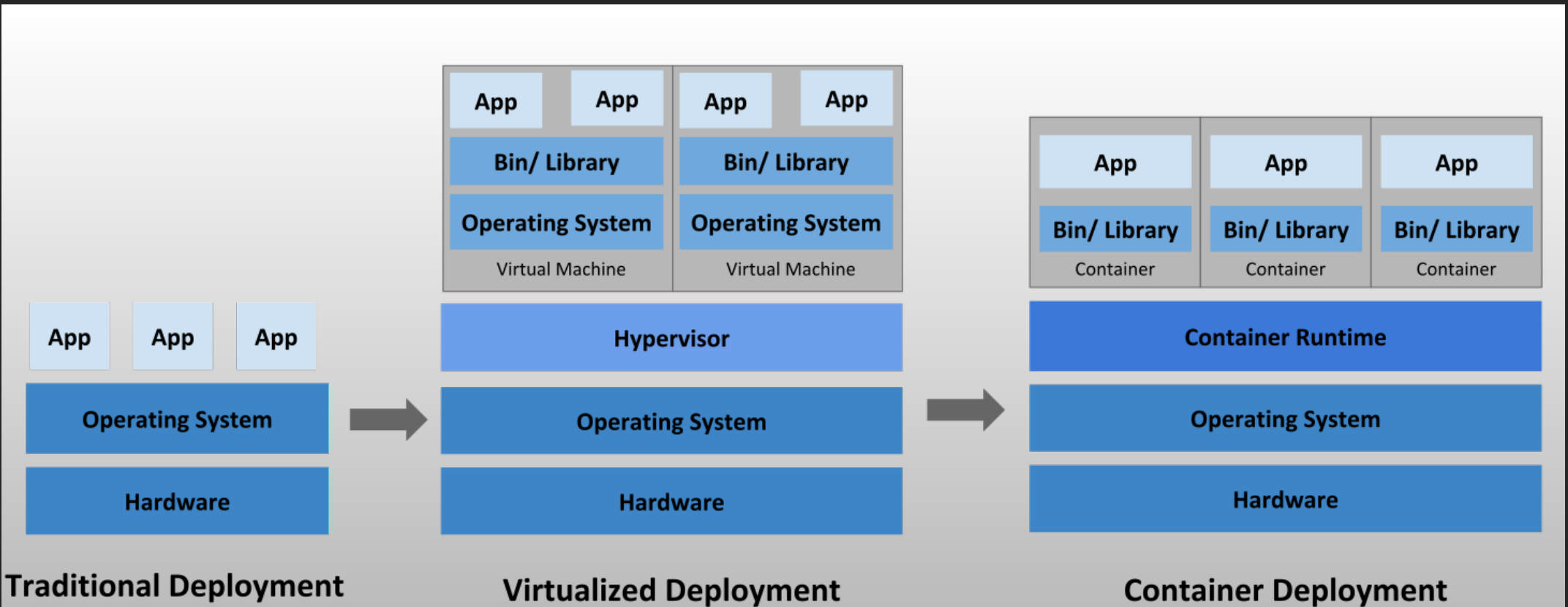
WHY KUBERNETES?

CONTAINERS

WHAT IS KUBERNETES?

Containers

Container evolution



Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Container tools



docker

Docker

The most popular



podman

Podman

Daemon less

Pods/containers

Run an existing container

1. Select one from a Container Registry
 - Docker.io <https://hub.docker.com/> /httpd
 - Quay.io
2. Run the container:

```
$ docker run -p 8080:80 httpd:alpine
```
3. Access to the container <http://localhost:8080>

Inside the container

1. Execute a command in the container:

```
$ docker ps
```

```
$ docker exec [container-id] pwd
```

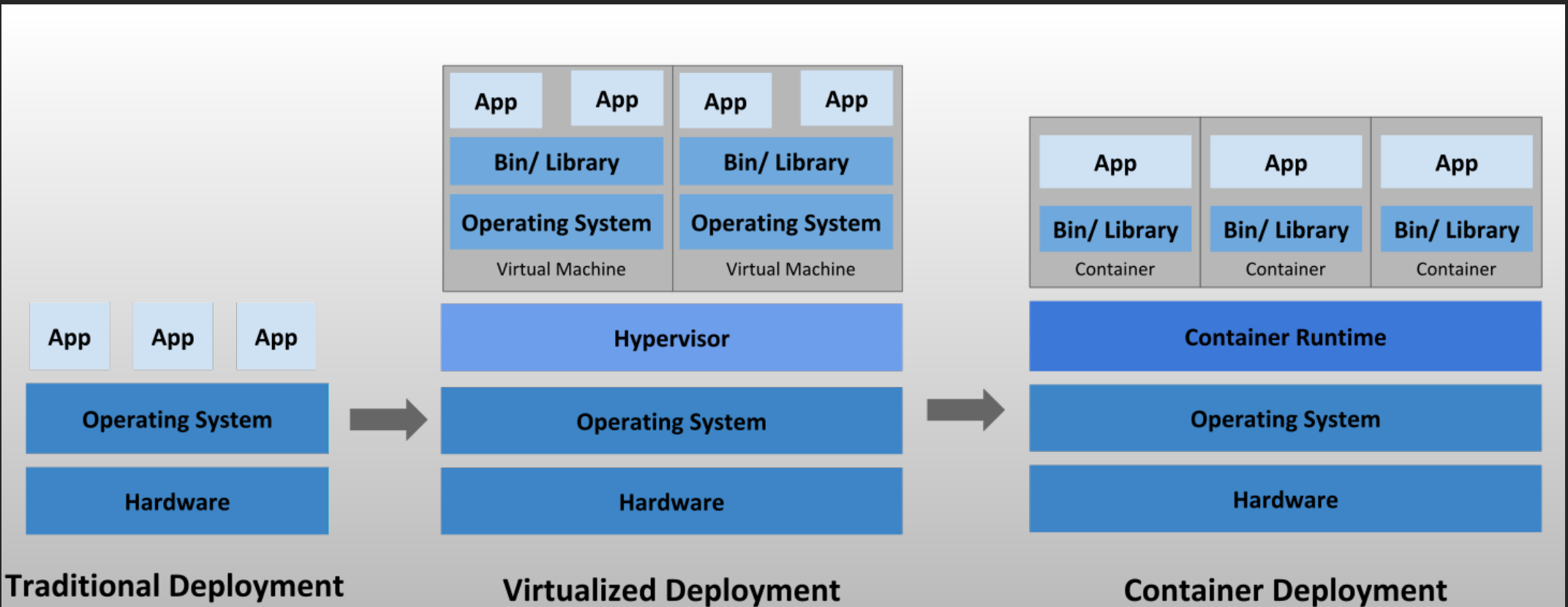
2. Get an **interactive terminal(tty)** inside the container

```
$ docker exec -it [container-id] /bin/sh
```

3. Look at the **filesystem** and the **processes**

Containers

Containers are isolated processes of the host



Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Create your own container image

1. Create a **Dockerfile**

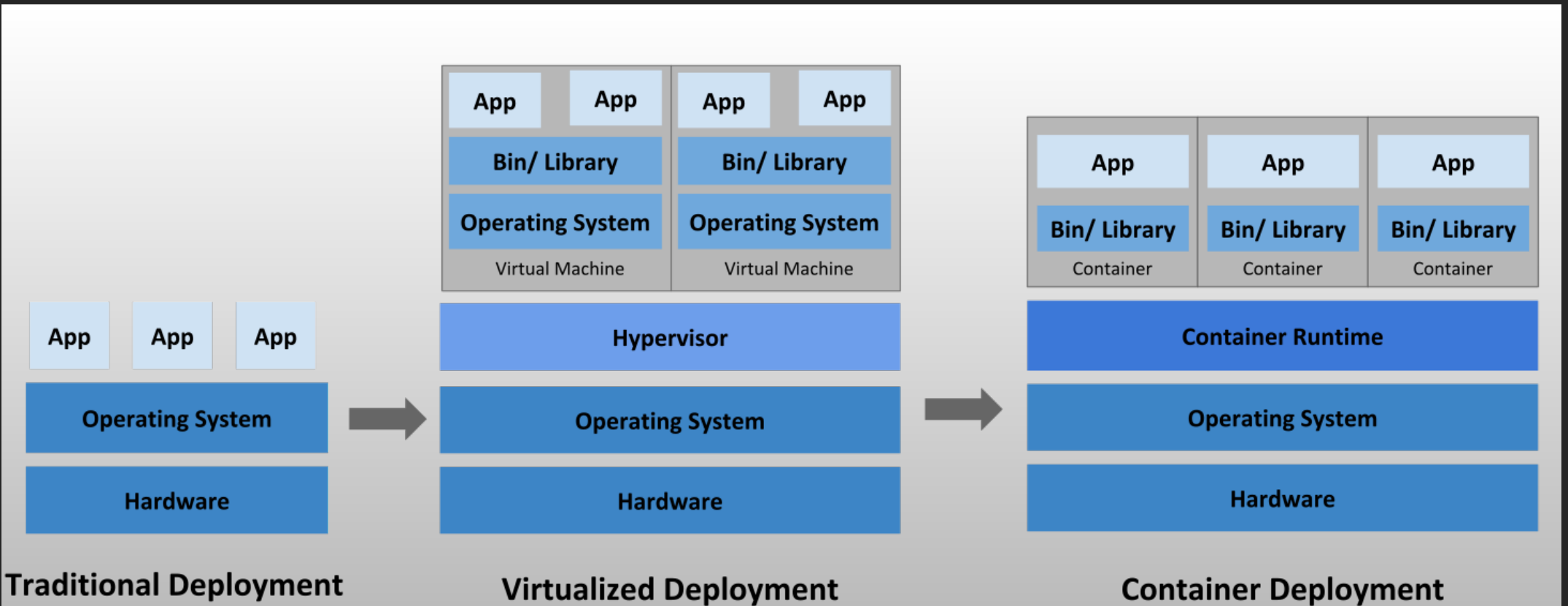
```
FROM docker.io/tomcat:10.1-jdk17-openjdk
COPY target/*.war /usr/local/tomcat/webapps/ROOT.war
```

2. Build

```
$ mvn clean install
$ docker build -f src/main/docker/Dockerfile -t
quay.io/sunix/hello-tomcat .
```

Containers

Containers were there for a while



Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Containers

Create your own container image

1. Create a **Dockerfile**
2. Build
3. Push

```
$ docker login quay.io
```

```
$ docker push quay.io/sunix/hello-tomcat
```

4. Run

```
$ docker run -p 8888:8080 quay.io/sunix/hello-tomcat
```

Containers

Container image

From Zero to production with IaC



Source code

Basically a Dockerfile



Build

Using `Docker or Podman



Push/Pull

Optionally to a container image registry like dockerhub or quay.io



Run anywhere

Any linux host that support container technology should be able to run it.

Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Containers

versus a Java application

From Zero to production



Source code

Basically Java files



Build

Using Maven, or Gradle



Push/Pull

Optionally to a Maven repo
like Nexus or Artifactory



Run anywhere

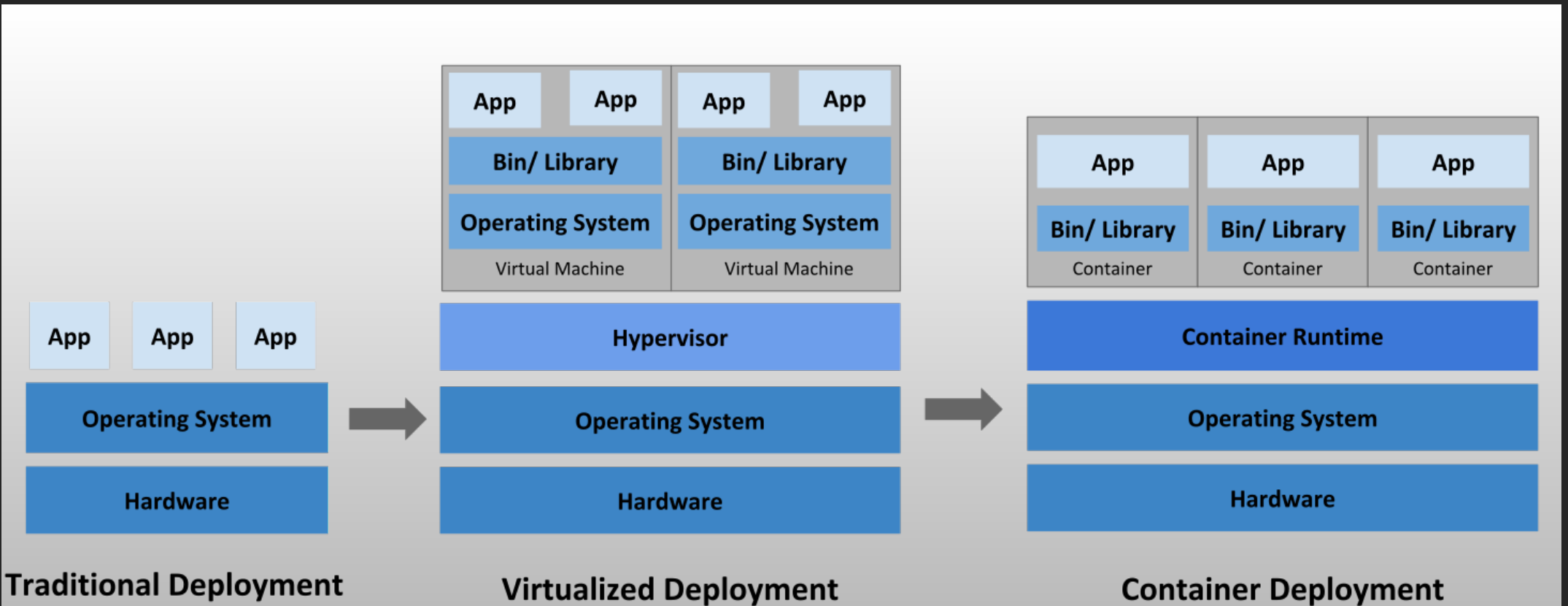
Any OS host that support
JVM technology should be
able to run it.

Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Containers

Containers limitations

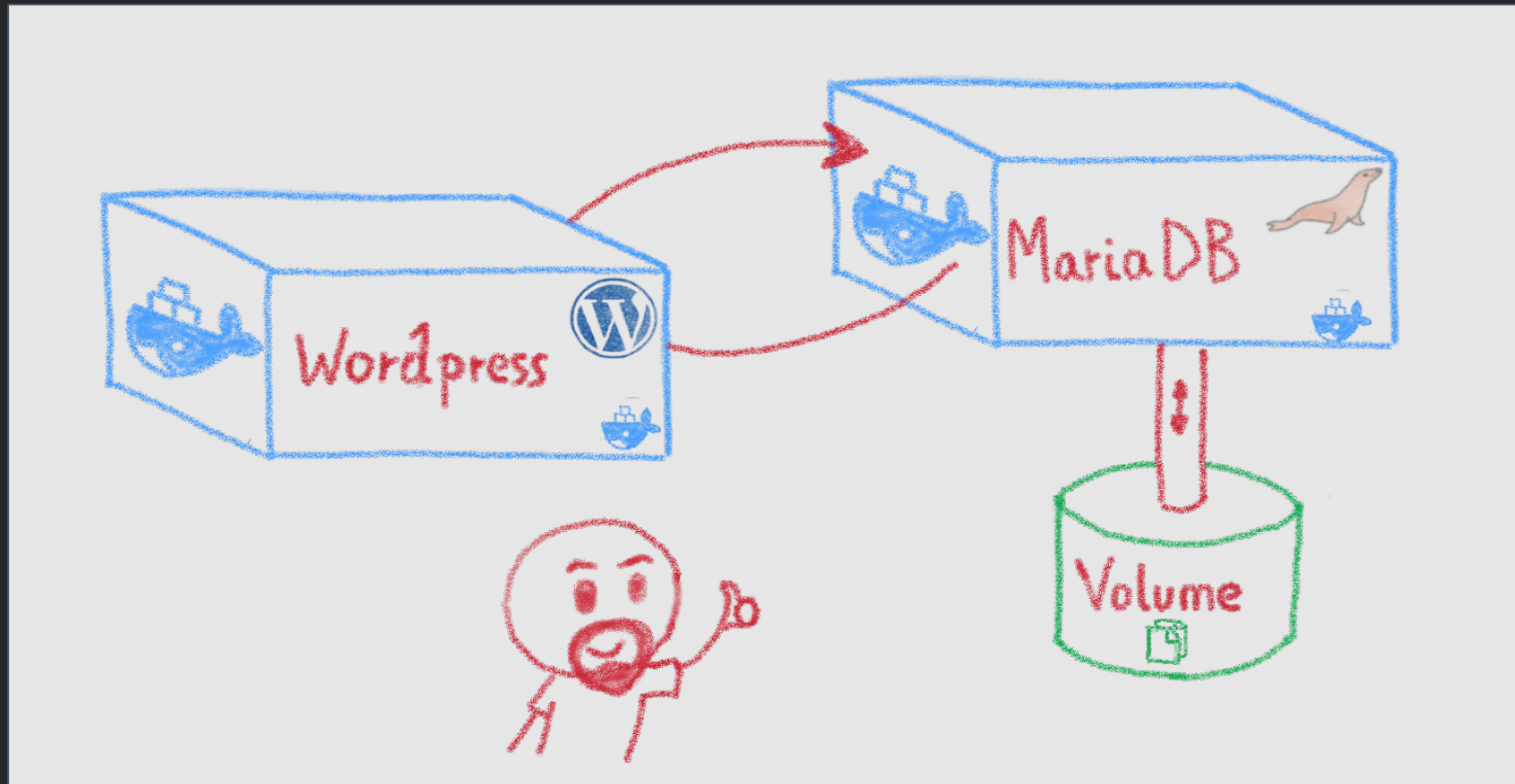


Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Containers

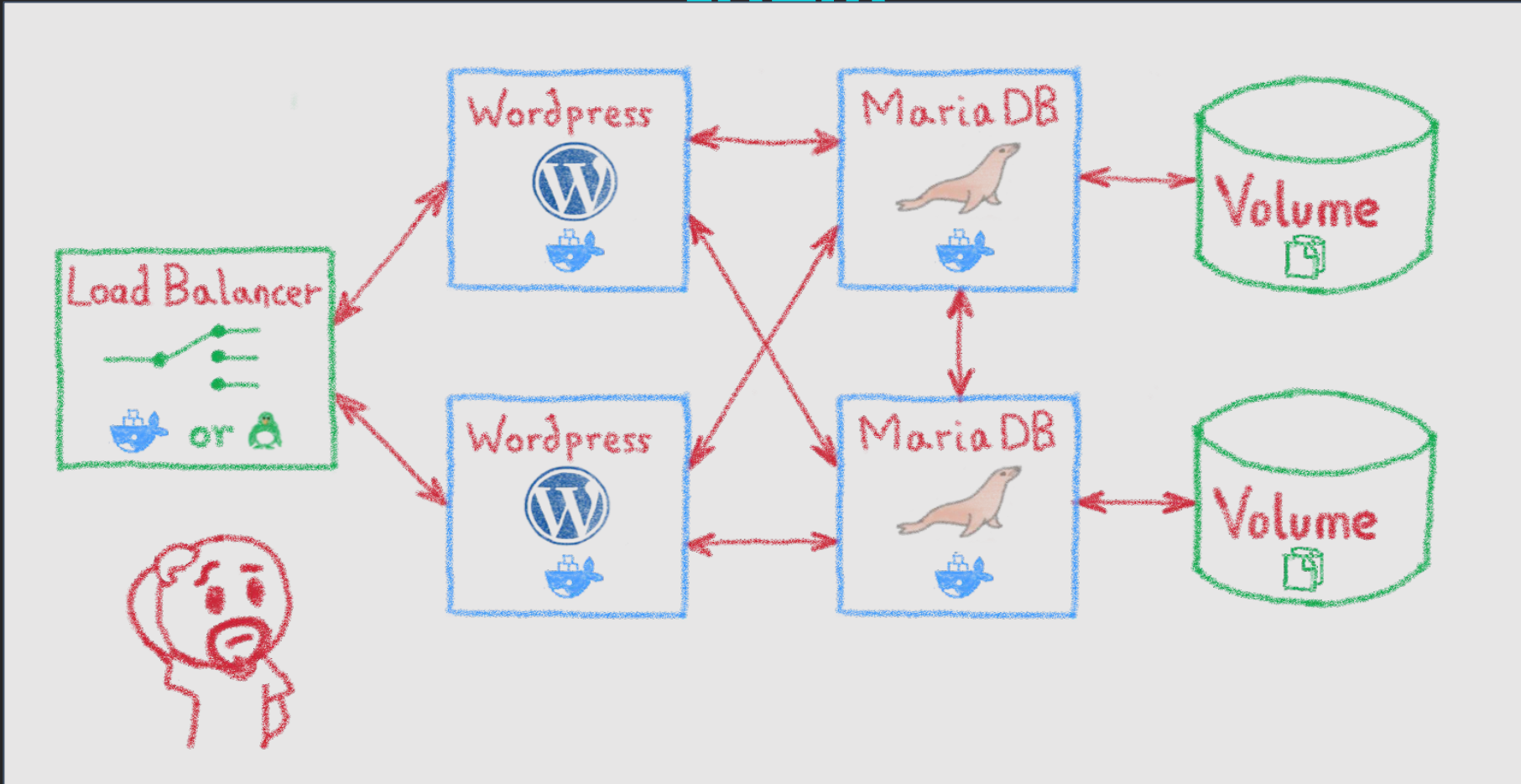
Containers are easy for Devs



No more "It works on my computer" problem!

Containers

Less simple if you must operate them



Like in a production context...

Containers

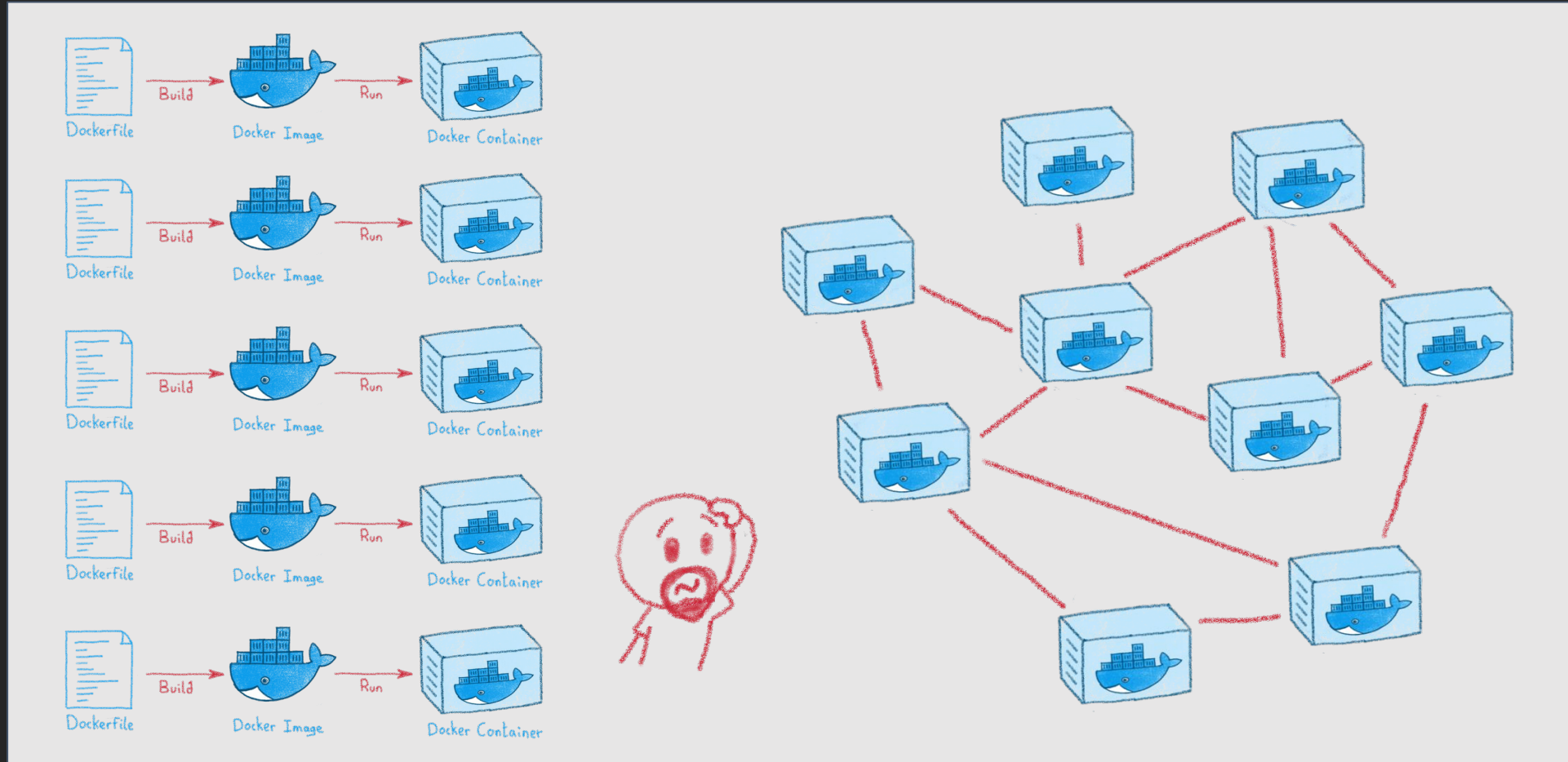
Containers opened the doors to microservices



Are you sure you want to operate them by hand?

Containers

Containers opened the doors to microservices

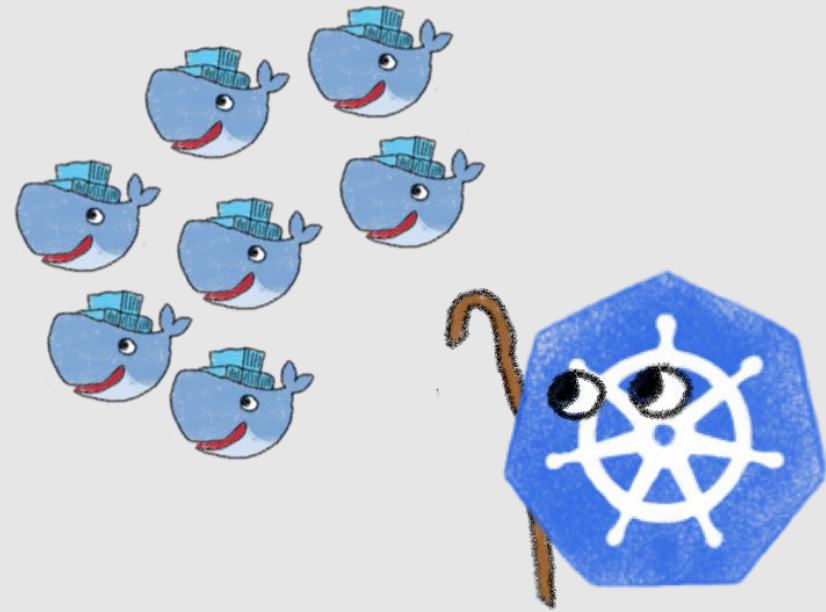


Are you sure you want to operate them by hand?

Kubernetes is a full orchestrator

Takes care of:

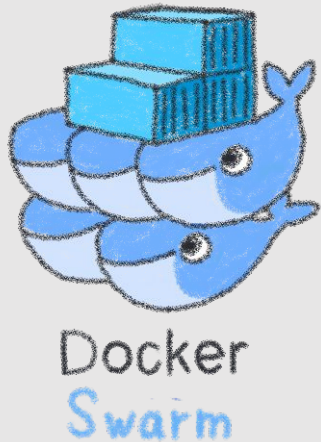
- Deployment
- Scaling
- Monitoring
- Repairing
- Securing
- ...



A shepherd for your containers

Containers

A sweet spot between features and complexity



And marketing, foundations and money



Devoxx France 2023

Introduction

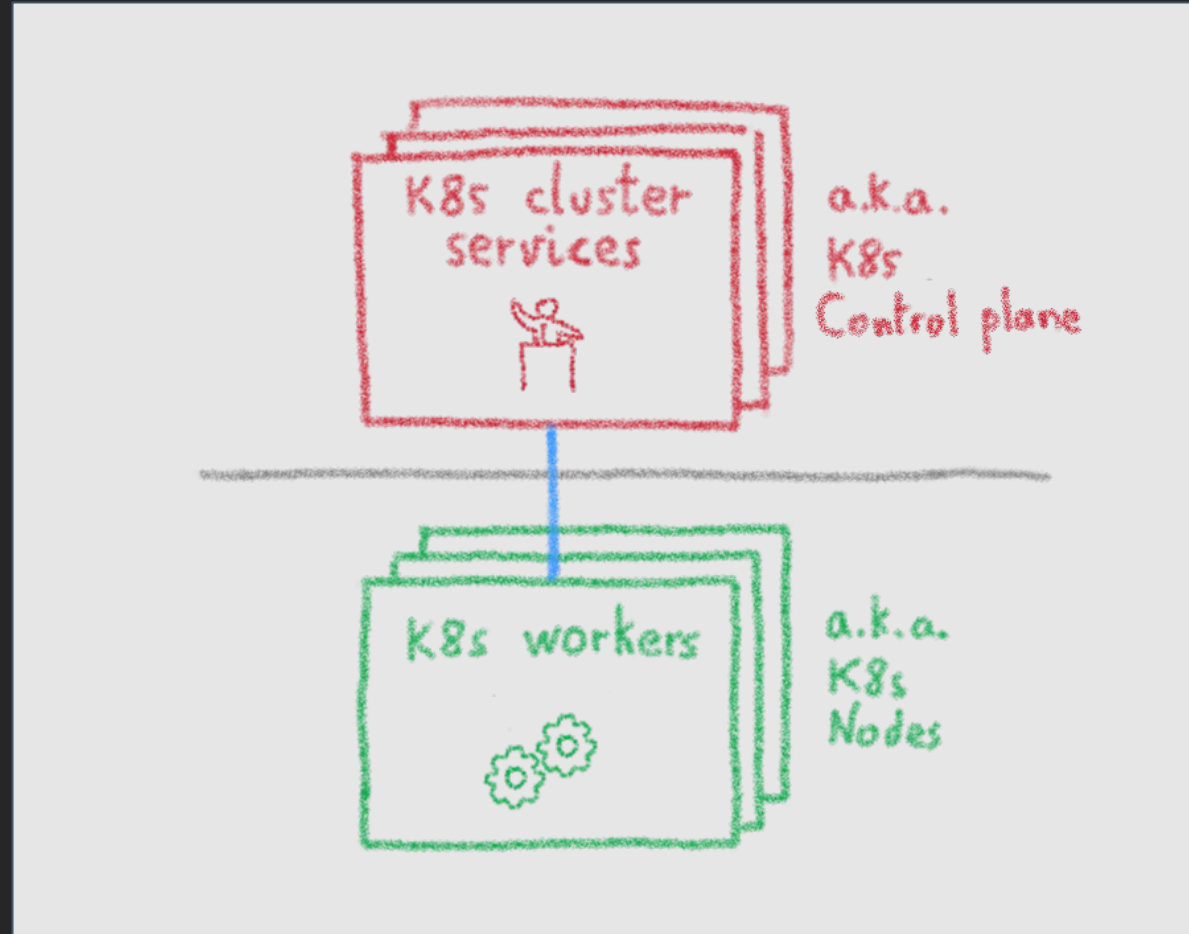
WHY KUBERNETES?

CONTAINERS

WHAT IS KUBERNETES?

Kubernetes

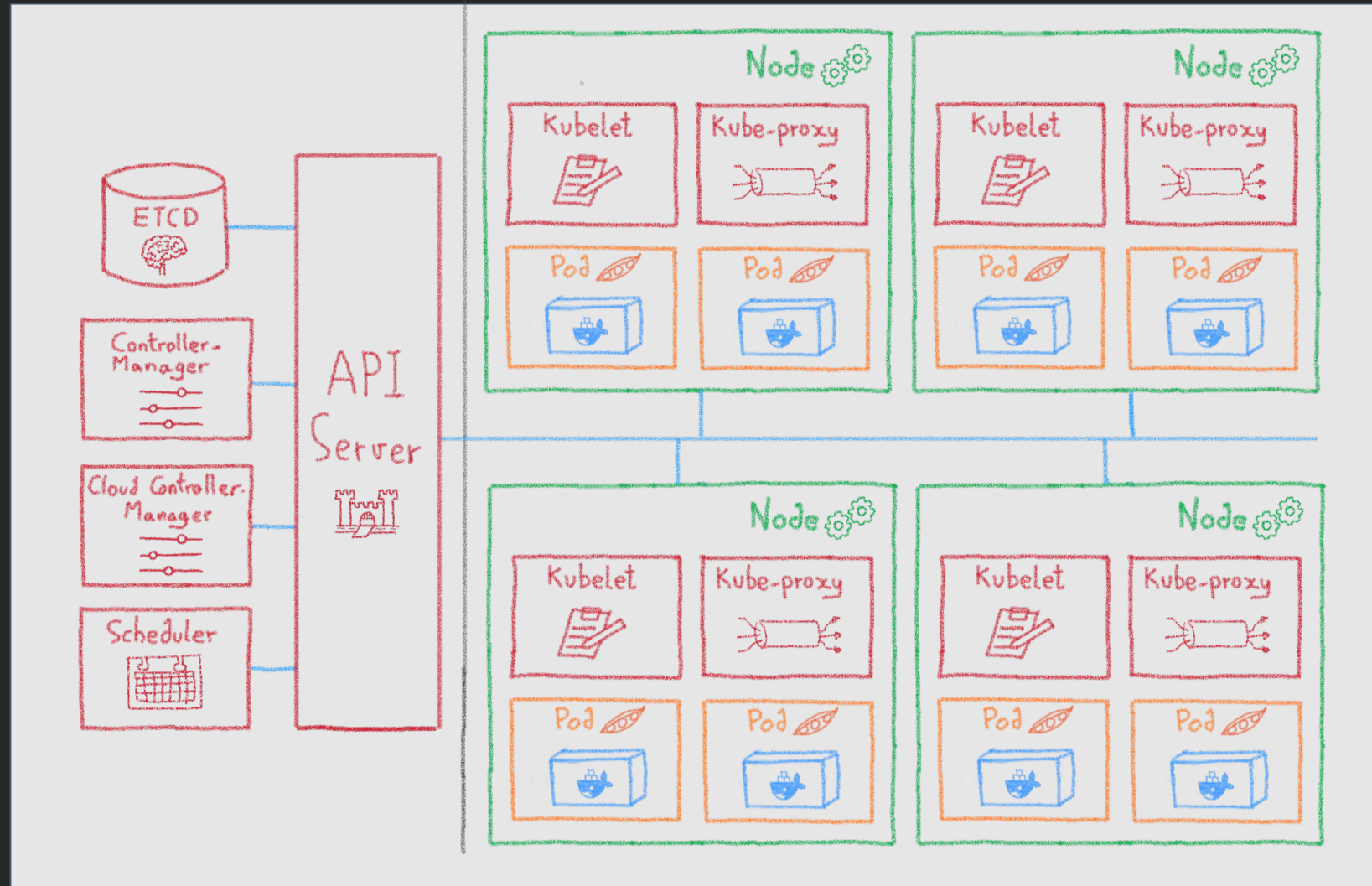
An open-source container orchestration system



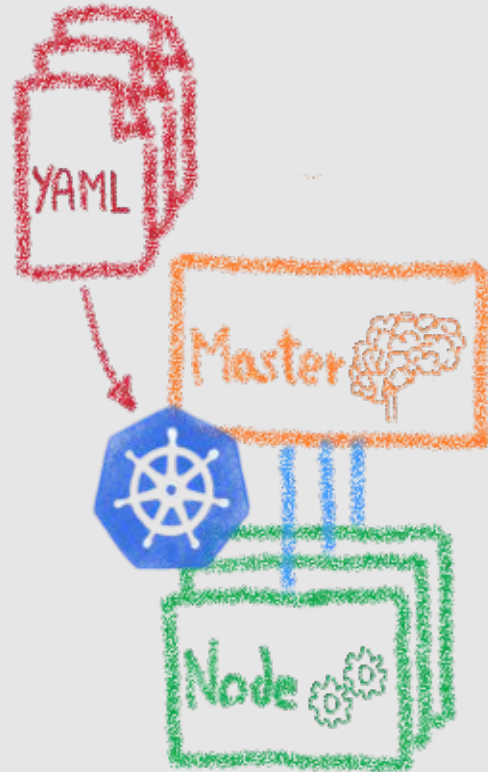
A cluster of instances

Kubernetes

Kubernetes cluster: more details



Desired State management



Manifest files:

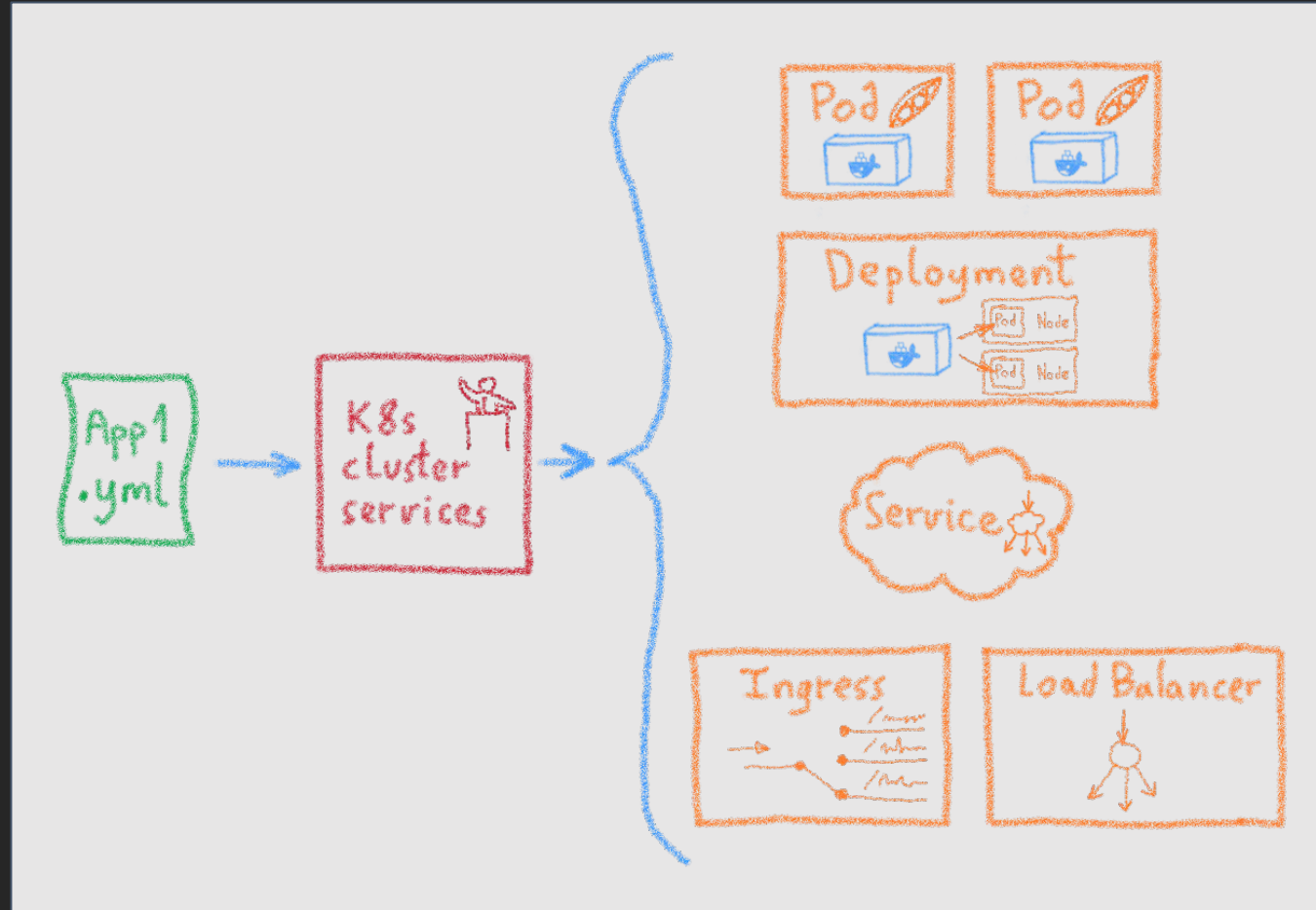
Text files in YAML format

High-level description of
the target architecture

Declarative infrastructure

Kubernetes

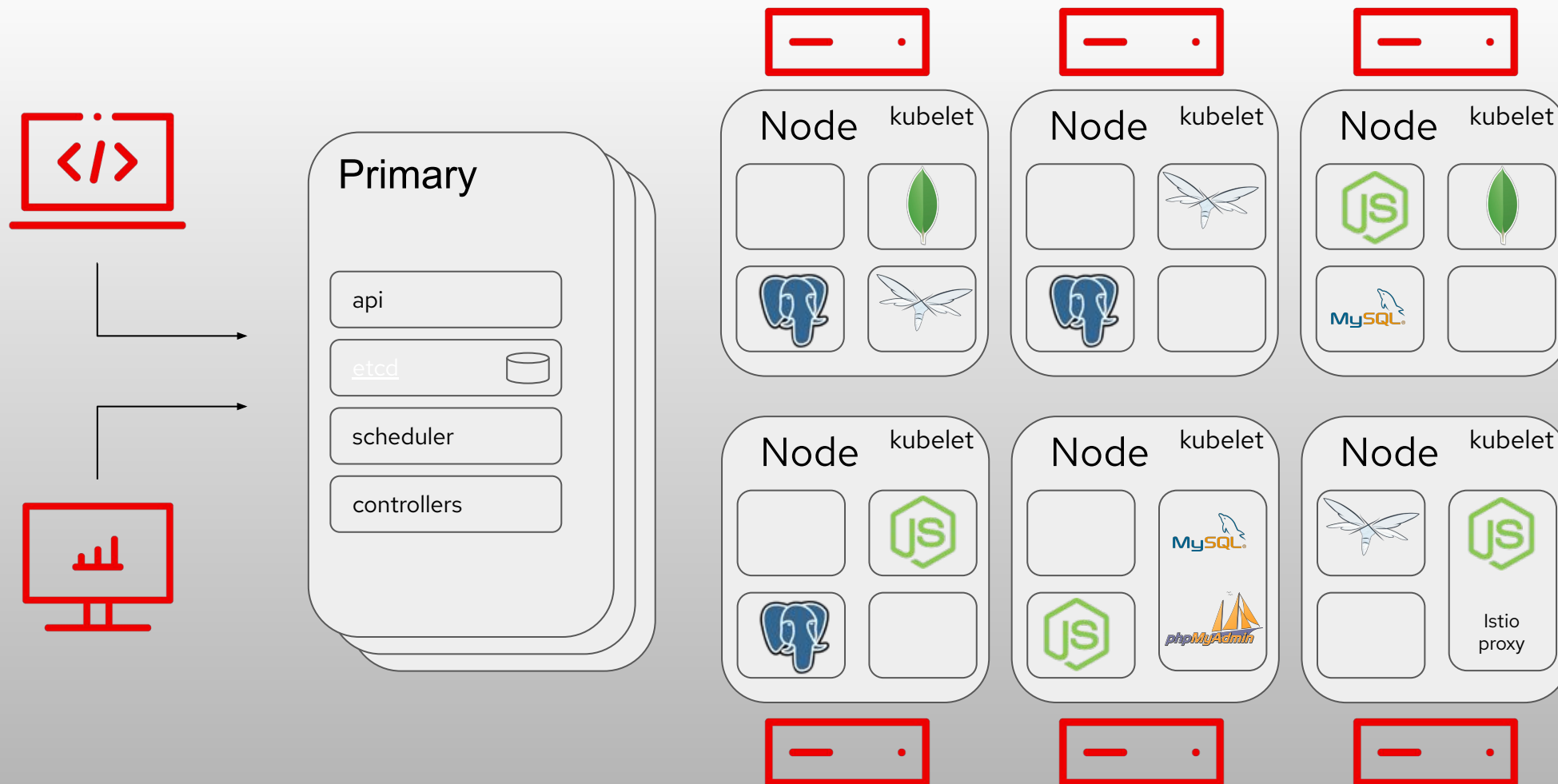
Desired State management



Let's begin with five objects

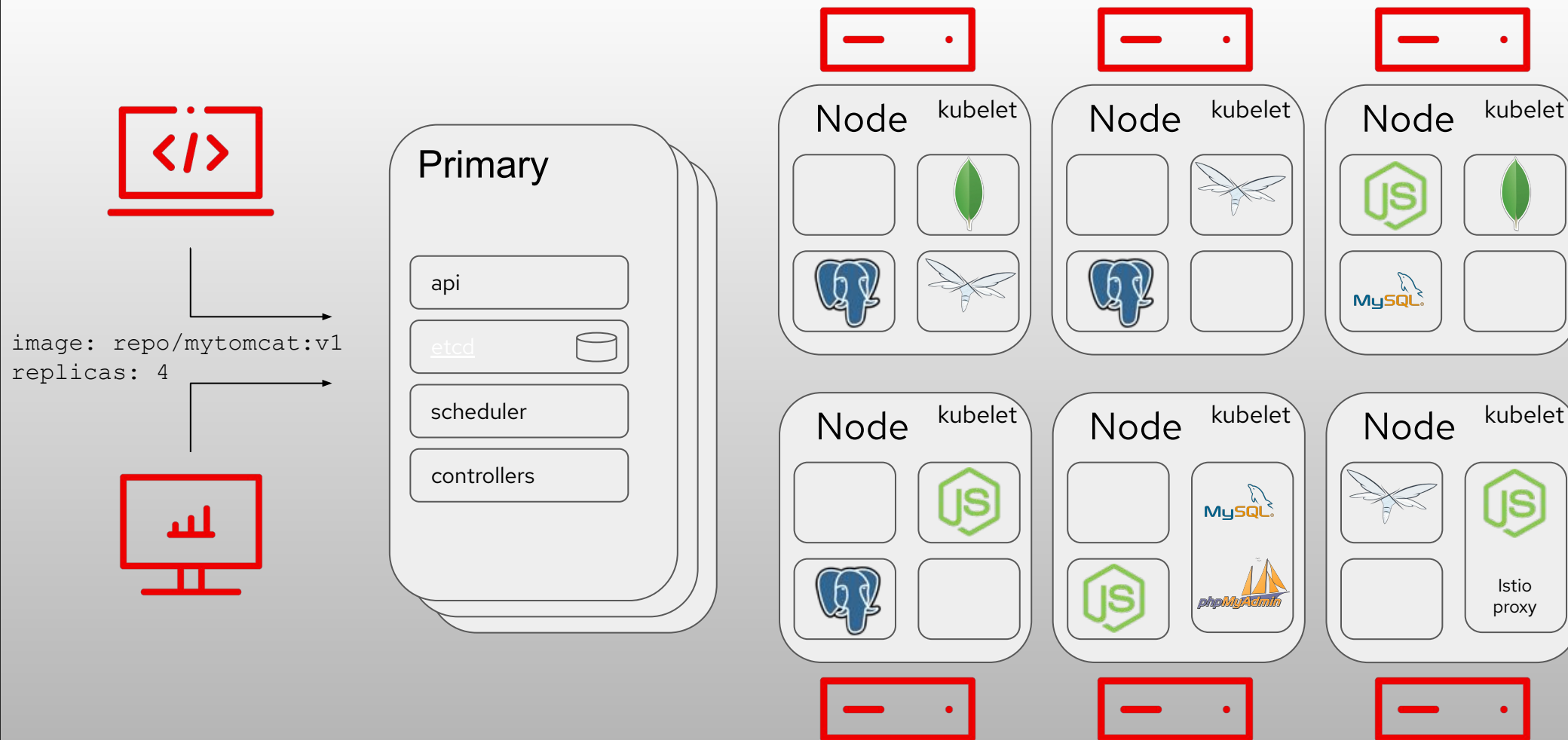
What is Kubernetes?

Kubernetes Cluster - Nodes



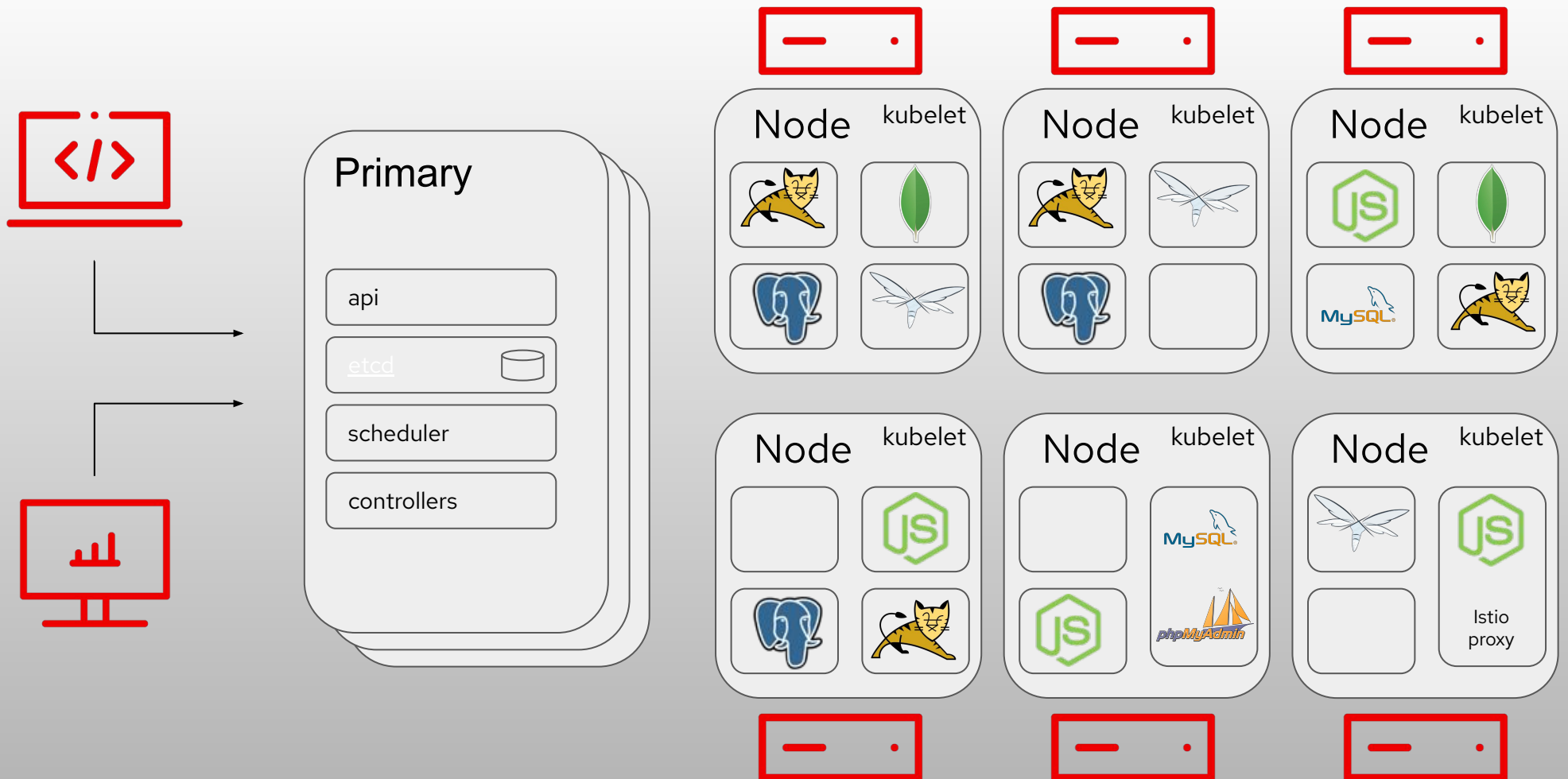
What is Kubernetes?

Kubernetes Cluster - Declarative API



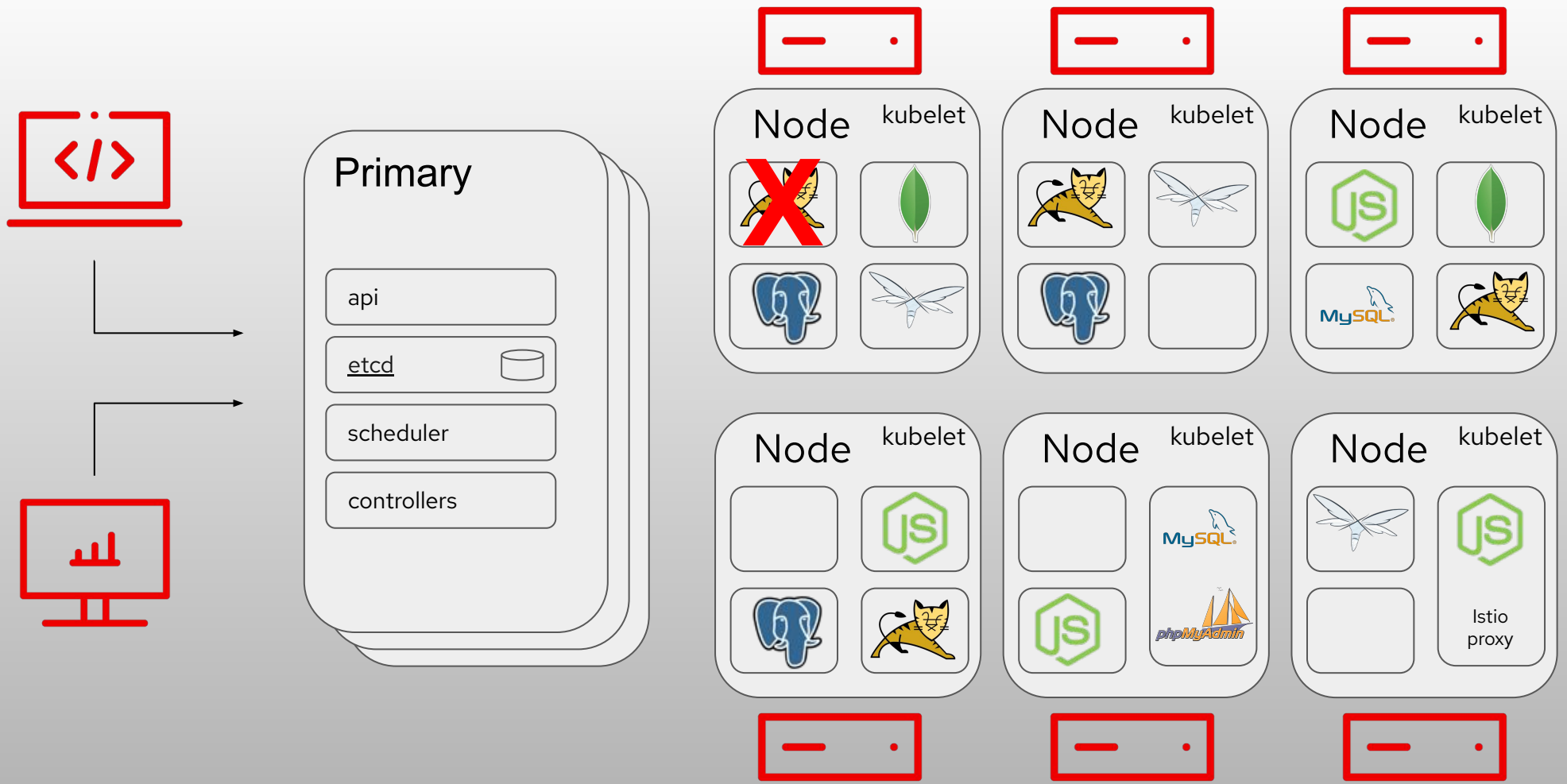
What is Kubernetes?

K8s Cluster - 4 Tomcat Instances



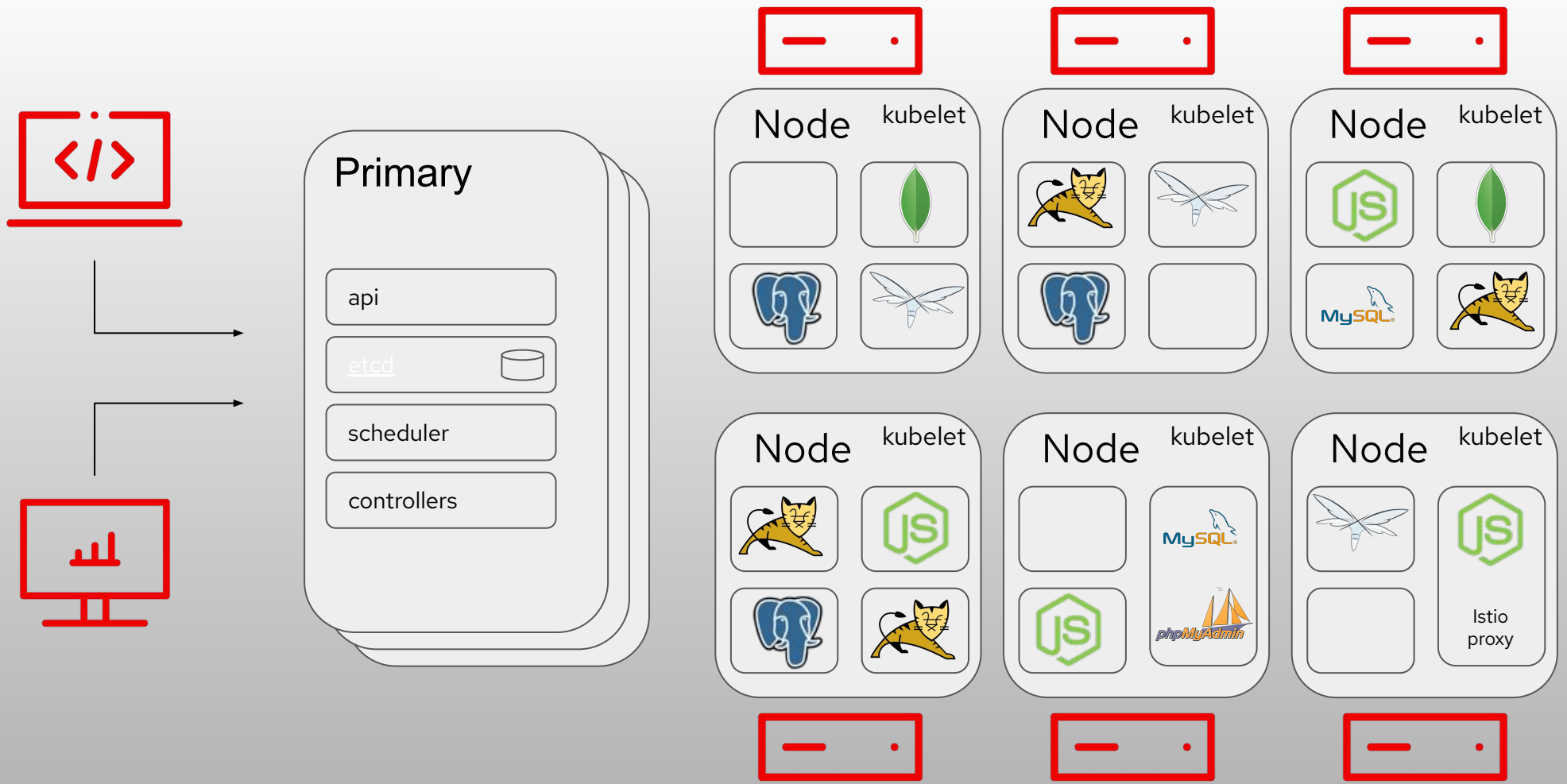
What is Kubernetes?

Kubernetes Cluster - Pod Failure



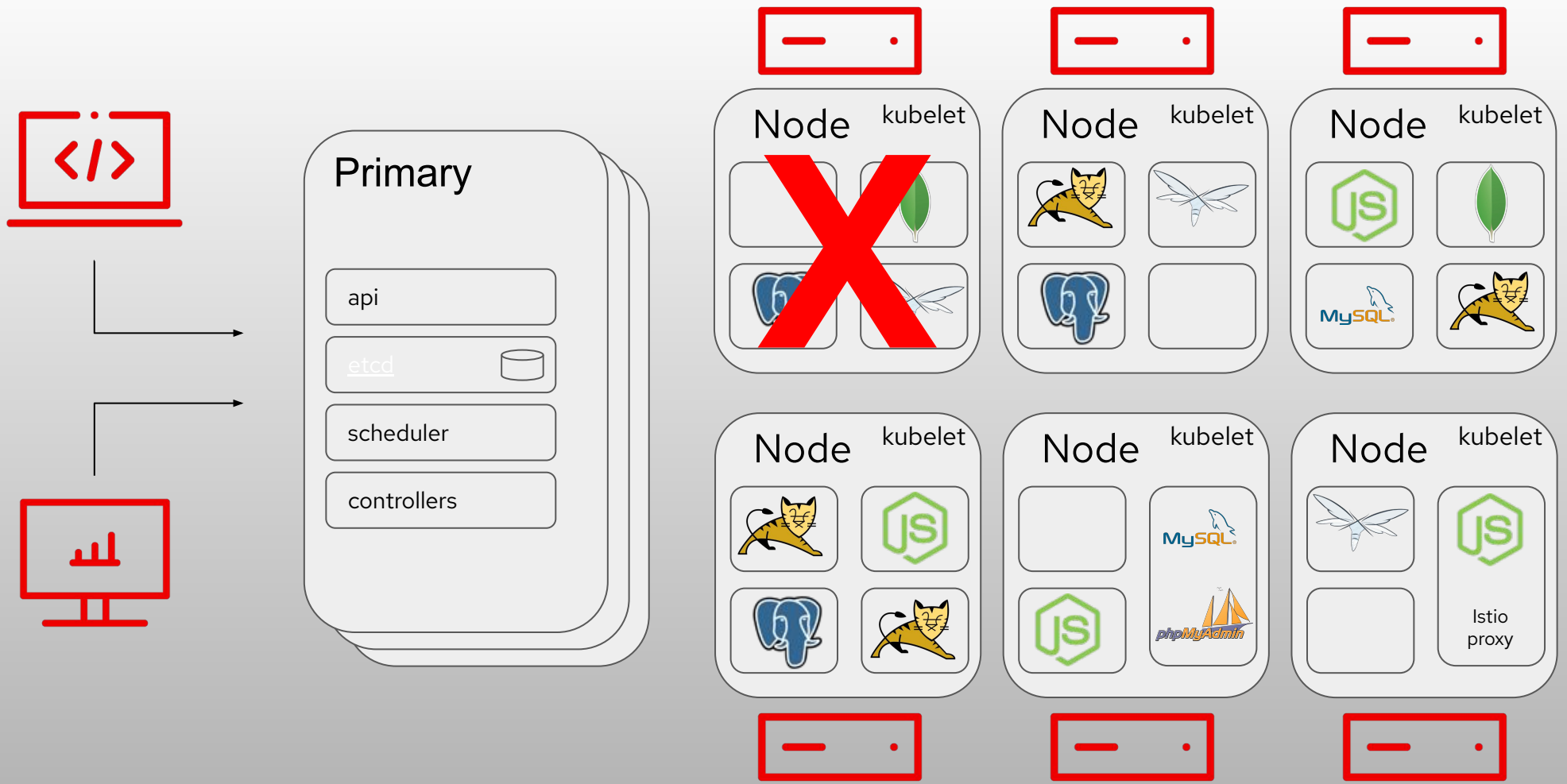
What is Kubernetes?

Kubernetes Cluster - Recovery



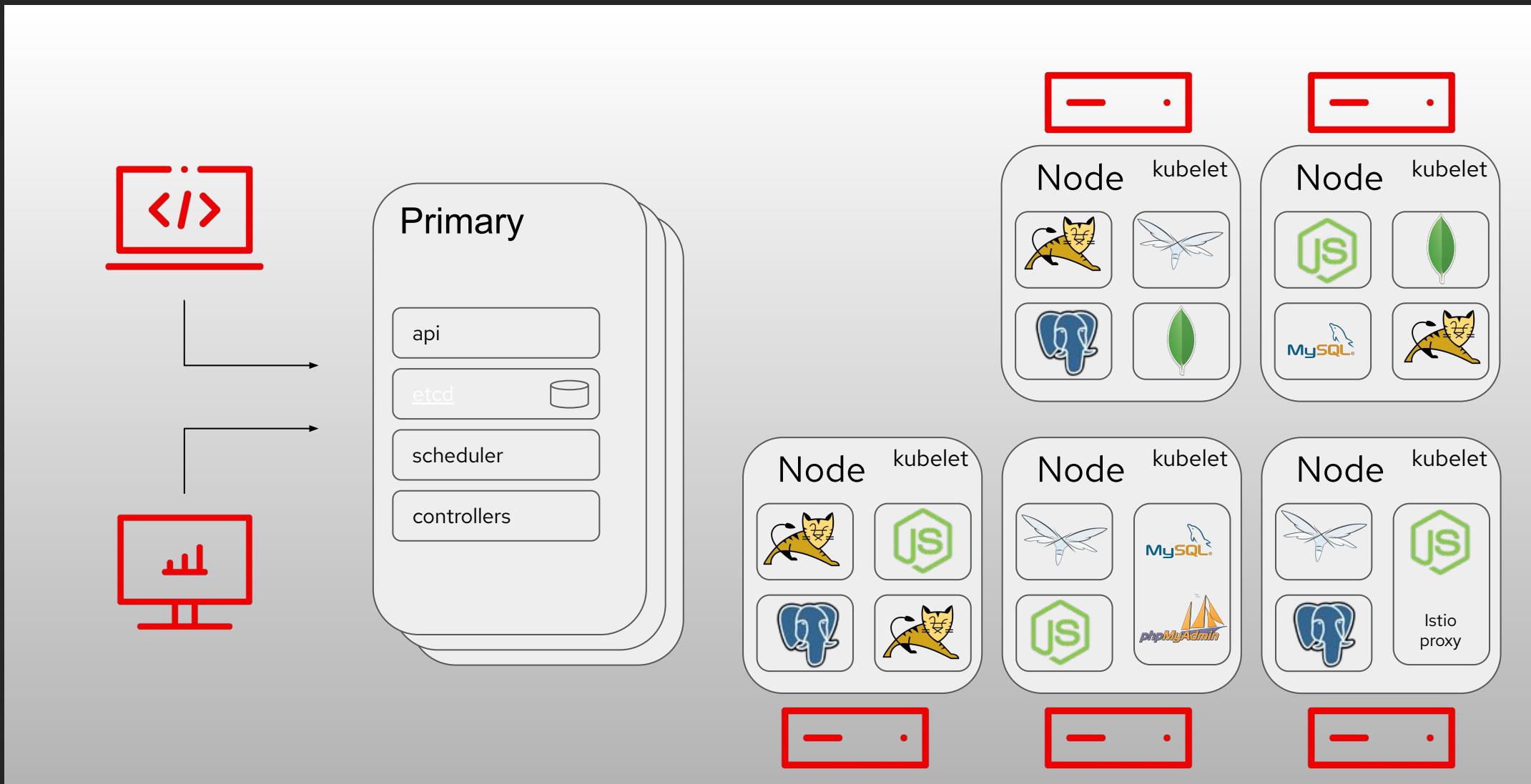
What is Kubernetes?

Kubernetes Cluster - Node Failure



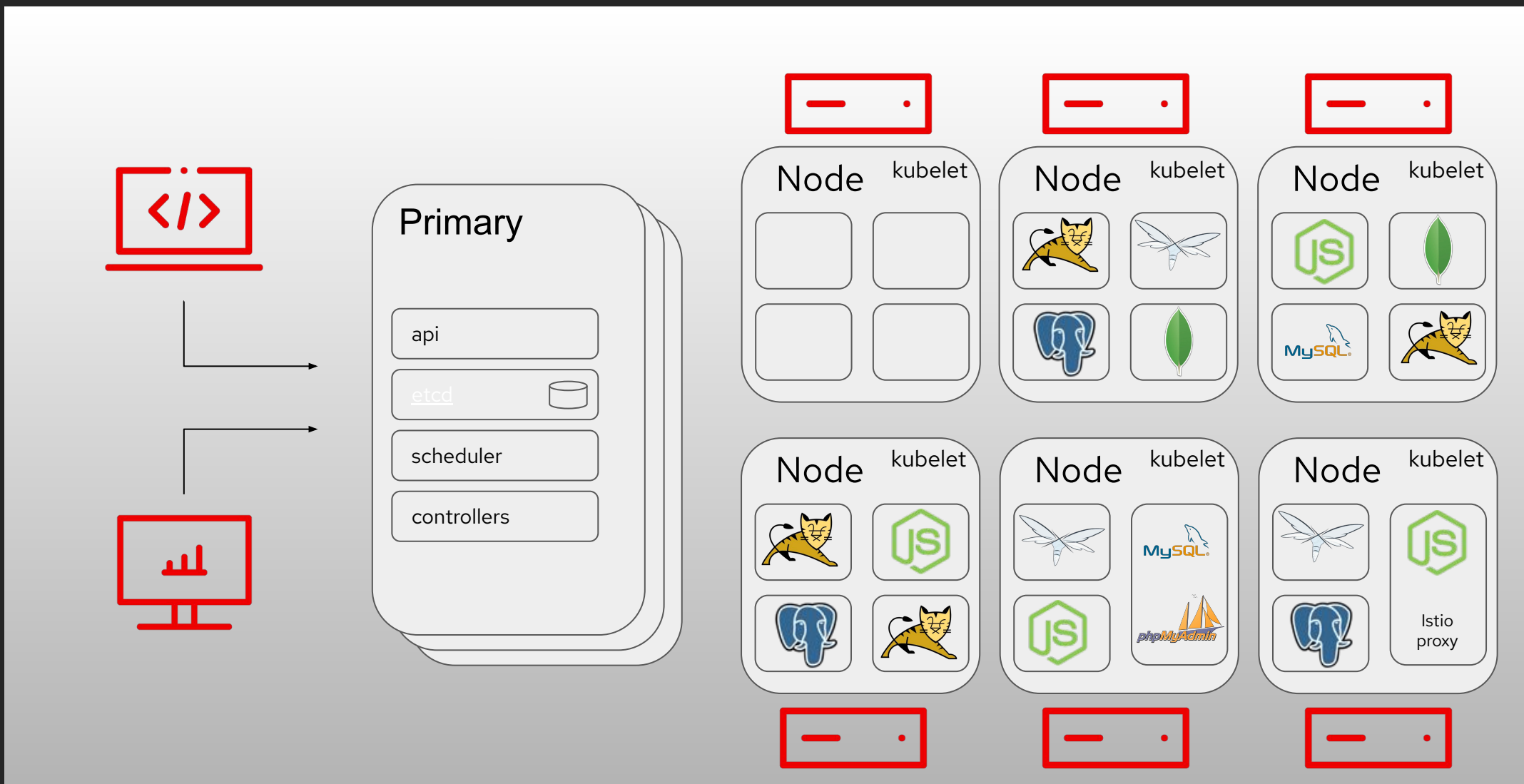
What is Kubernetes?

Kubernetes Cluster - Pods Replaced



What is Kubernetes?

Kubernetes Cluster - New Node





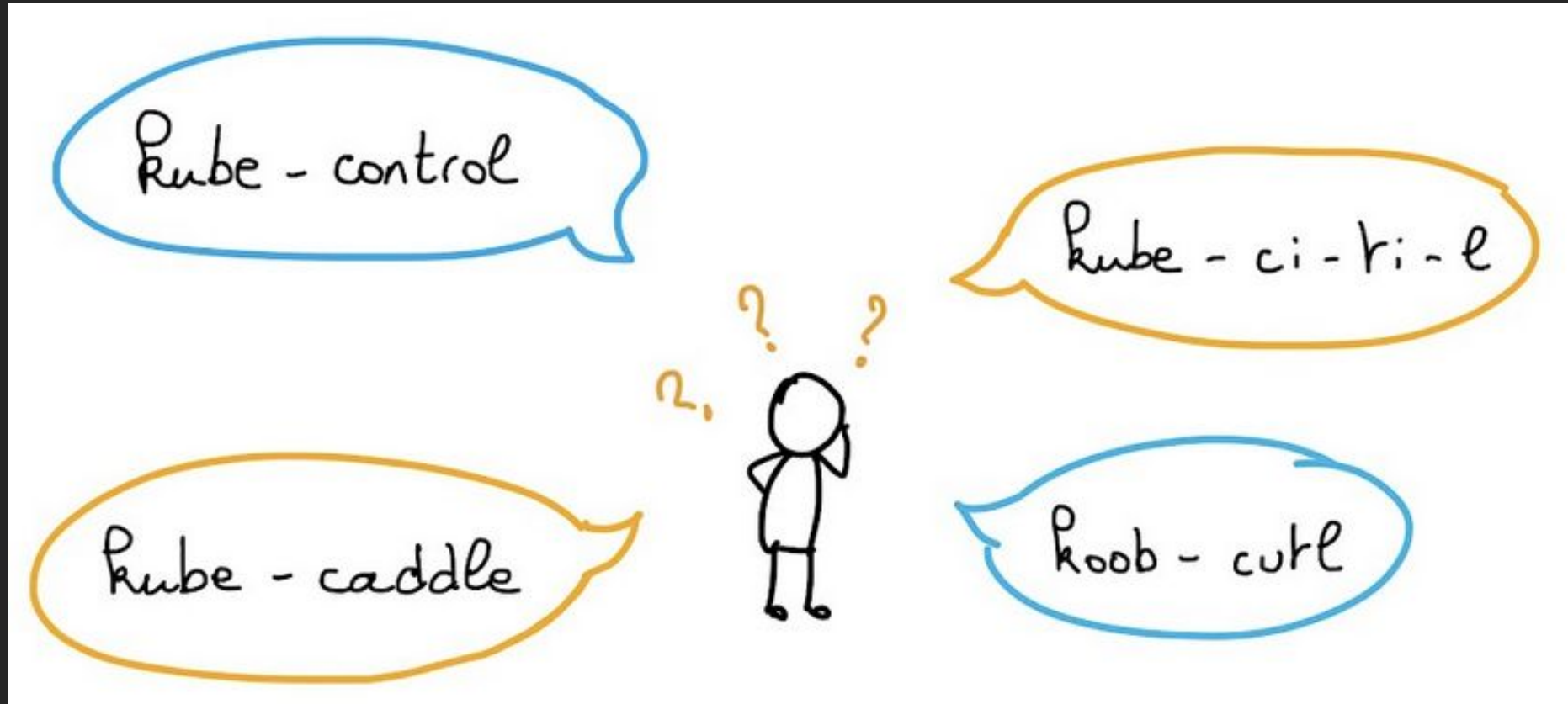
Devoxx France 2023

1 - Dive Into K8S Building Blocks

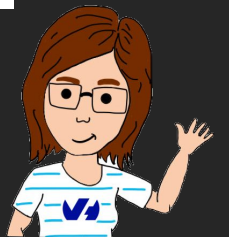
PLAYING WITH KUBECTL

YAML

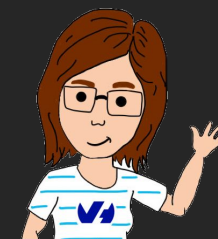
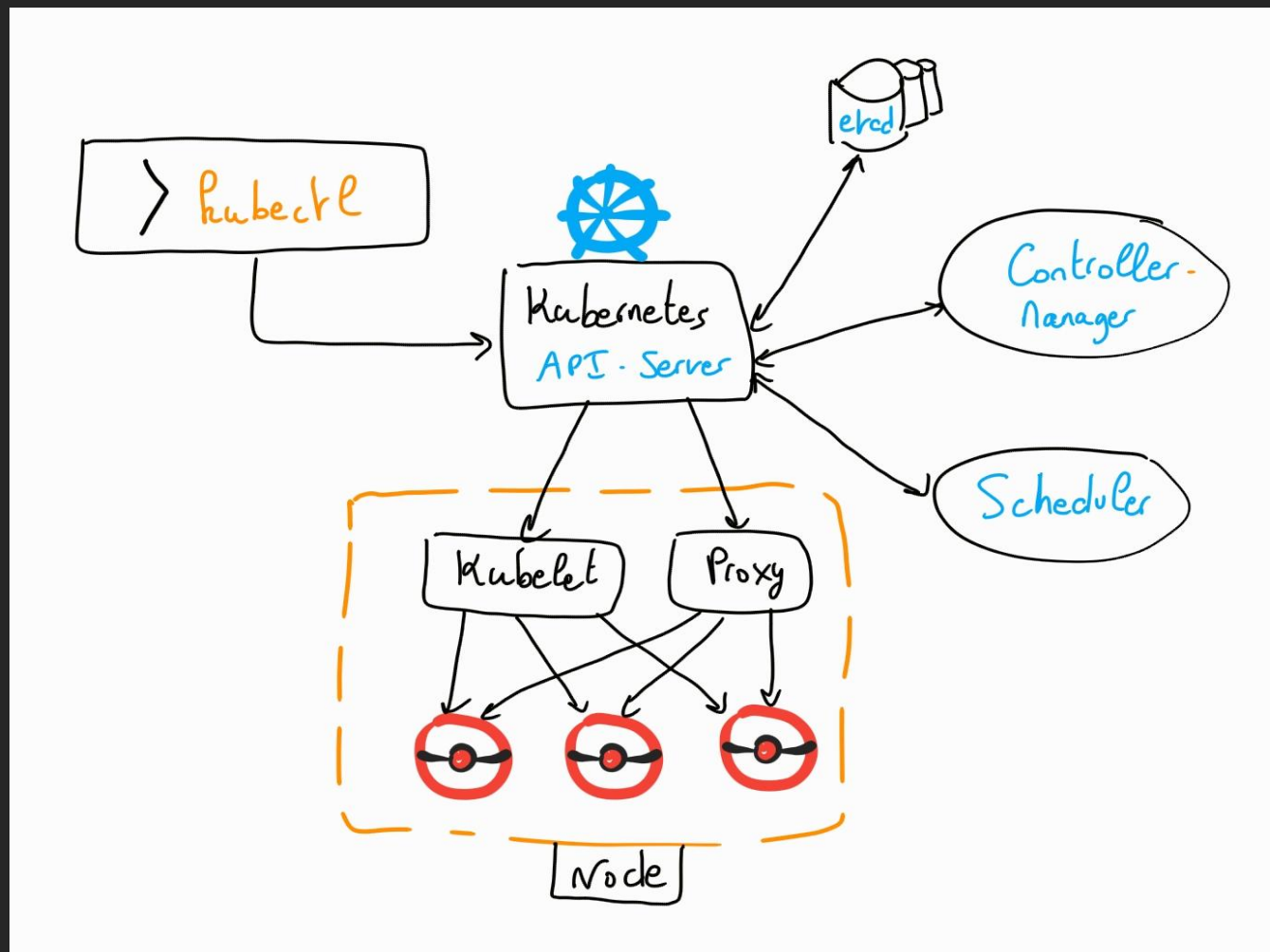
Kubectl > pronunciation fight



Pronounce "kubectl" as you want :-)



Kubectl > kubernetes tool/cli





Devoxx France 2023

1 - Dive Into K8S Building Blocks

PLAYING WITH KUBECTL

YAML

Kubernetes

- A distributed and structured YAML database
 - CRUD, **structured** and **typed** objects: **Resources**
 - Resources live in **Namespaces**

Create a Resource object

- speaker.yaml

```
apiVersion: "stable.world.com/v1"  
kind: Speaker  
metadata:  
  name: horacio  
spec:  
  name: "Horacio"  
  title: "DevRel at OVH Cloud"  
  action: "speak"
```

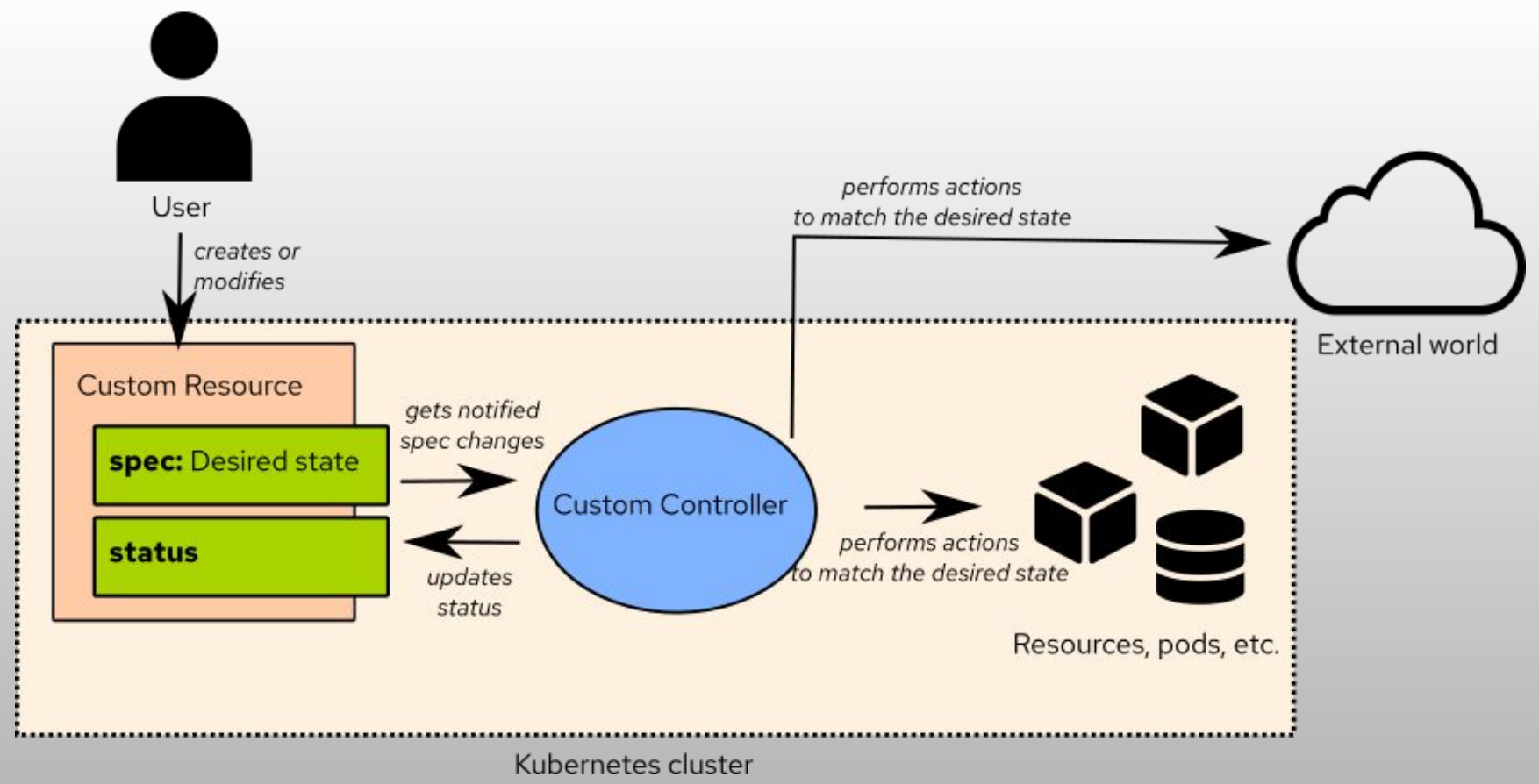
- Execute

```
$ kubectl apply -f speaker.yaml  
$ kubectl get Speaker
```

Kubernetes

- A distributed and structured YAML database
- **Controllers** that do the job
 - **Listening** to Resources Create/Update/Delete events: the **user requirements**
 - Perform to **match** the user requirements

Kubernetes



Source:

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Kubernetes

- A distributed and structured YAML database
- Controllers that do the job
 - Listening to Resources Create/Update/Delete events: the user requirements
 - Perform to match the user requirements
- By default, a set of Resources and Controllers to manage a cluster of machines

Pod

1-n containers sharing network addressing/volumes, etc.

- mypod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: hellotomcat-container
    image: quay.io/sunix/hello-tomcat
    ports:
    - containerPort: 8080
  imagePullPolicy: IfNotPresent
```

<https://asciinema.org/a/EeeNkoQ2eJ76Twx2S0sCybTzz>

Deployment

Deploy and manage pods that looks similar: Scaling

- `hello-tomcat-deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hellotomcat
  labels:
    app: hellotomcat
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hellotomcat
  template:
    metadata:
      labels:
        app: hellotomcat
    spec:
      containers:
        - name: hellotomcat-container
          image: quay.io/sunix/hello-tomcat
          ports:
            - containerPort: 8080
          imagePullPolicy: IfNotPresent
```

Service

Load balancer

Let the pods communicates in the cluster or outside

- myservice.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: hellotomcat-service
spec:
  type: NodePort
  selector:
    app: hellotomcat
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 8080
```

Ingress

Manage the paths and domain name redirections Services

- `myingress.yaml`

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: hellotomcat-ingress
  labels:
    app: hellotomcat
spec:
  rules:
  - host: 192.168.49.2.nip.io
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: hellotomcat-service
            port:
              number: 8080
```



Devoxx France 2023

2 - BEING A GOOD CLOUD NATIVE CITIZEN

Requests and limits

Health probes

ConfigMap and Secrets

Resource management

Resource management

- **Requests:** how many resources it needs
- **Limits:** how many resources it can use

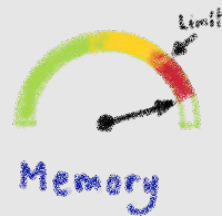
Resources types and units

- **Memory:** $1\text{ T} = 1000\text{ G} = 10^6\text{ M} = 10^9\text{ K}$
 $1\text{ Ti} = 2^{10}\text{ Gi} = 2^{20}\text{ Mi} = 2^{30}\text{ Ki}$
- **CPU:** $1\text{ vCore} = 1\text{ CPU} = 1000\text{ mCPU}$

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image: images.my-company.example/app
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

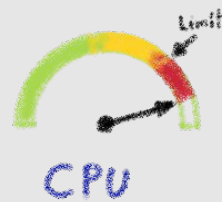
What if a pod uses too many resources?

Memory: if Pod tries to over allocate, it generates an OOM



Kernel OOM subsystem kills the Pod

CPU: if Pod tries to over use, kernel waits before allowing it to continue



At next time slice, kernel will make the Pod wait

Resource quota

Resource quota

limits.cpu	requests.cpu
limits.memory	requests.memory
limits.storage	requests.storage

A global quota per namespace

```
kind: ResourceQuota
metadata:
  name: compute-resources
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
    requests.nvidia.com/gpu: 4
```

Limit the total sum of compute resources that can be requested in a given namespace

Limit range

Limit ranges

- Min and max resources
- Default request/limits per pod in a namespace

```
apiVersion: v1
kind: LimitRange
metadata:
  name: cpu-resource-constraint
spec:
  limits:
    - default: # this section defines default limits
      cpu: 500m
      defaultRequest: # this section defines default requests
        cpu: 500m
      max: # max and min define the limit range
        cpu: "1"
      min:
        cpu: 100m
    type: Container
```

Default, minimum and maximum resources usage per pod in a namespace

Verifying resource usage

```
% kubectl top pods
```

NAME	CPU(cores)	MEMORY(bytes)
hello-world-deployment-bc4fd6b9-dgspd	3m	2Mi
hello-world-deployment-bc4fd6b9-f85mf	3m	2Mi
hello-world-deployment-bc4fd6b9-hh7xs	4m	2Mi
hello-world-deployment-bc4fd6b9-lz494	5m	2Mi

```
% kubectl top pods --containers
```

POD	NAME	CPU(cores)	MEMORY(bytes)
hello-world-deployment-bc4fd6b9-dgspd	hello-world	0m	2Mi
hello-world-deployment-bc4fd6b9-f85mf	hello-world	1m	2Mi
hello-world-deployment-bc4fd6b9-hh7xs	hello-world	1m	2Mi
hello-world-deployment-bc4fd6b9-lz494	hello-world	0m	2Mi

```
% kubectl top nodes
```

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
nodepool-ce18c6cd-1291-4a6e-83-node-5c283f	110m	5%	1214Mi	23%
nodepool-ce18c6cd-1291-4a6e-83-node-85b011	104m	5%	1576Mi	30%
nodepool-ce18c6cd-1291-4a6e-83-node-c3cfcf	121m	6%	1142Mi	22%



Devoxx France 2023

2 - BEING A GOOD CLOUD NATIVE CITIZEN

Requests and limits

Health probes

ConfigMap and Secrets

Health Probe

Tell people you're ready



Tell people you're ready

```
readinessProbe:  
  httpGet:  
    path: /health  
    port: 8080  
  initialDelaySeconds: 10  
  periodSeconds: 3
```

Tell people you're alive



STAYING ALIVE?

Health Probe

Tell people you're alive

```
livenessProbe:  
  httpGet:  
    port: 8080  
    path: /alive  
  initialDelaySeconds: 10  
  periodSeconds: 5  
  timeoutSeconds: 2
```



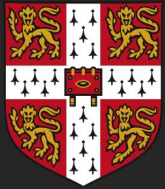
Devoxx France 2023

2 - BEING A GOOD CLOUD NATIVE CITIZEN

Requests and limits

Health probes

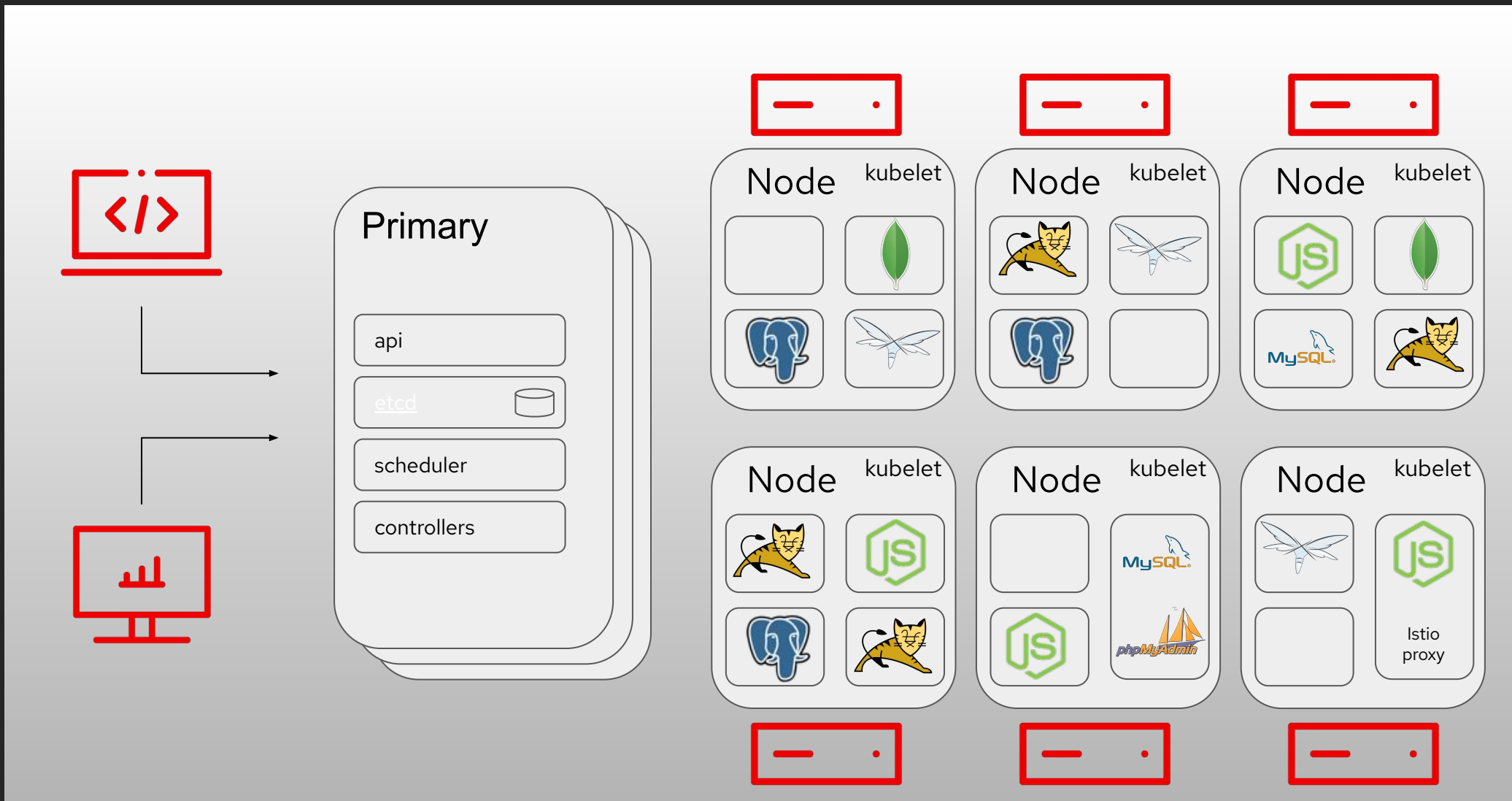
ConfigMap and Secrets



Cambridge
Dictionary

Secret: a piece of information that is only known by one person or a few people and should not be told to others.

Kubernetes Cluster - Nodes



ConfigMap & Secrets

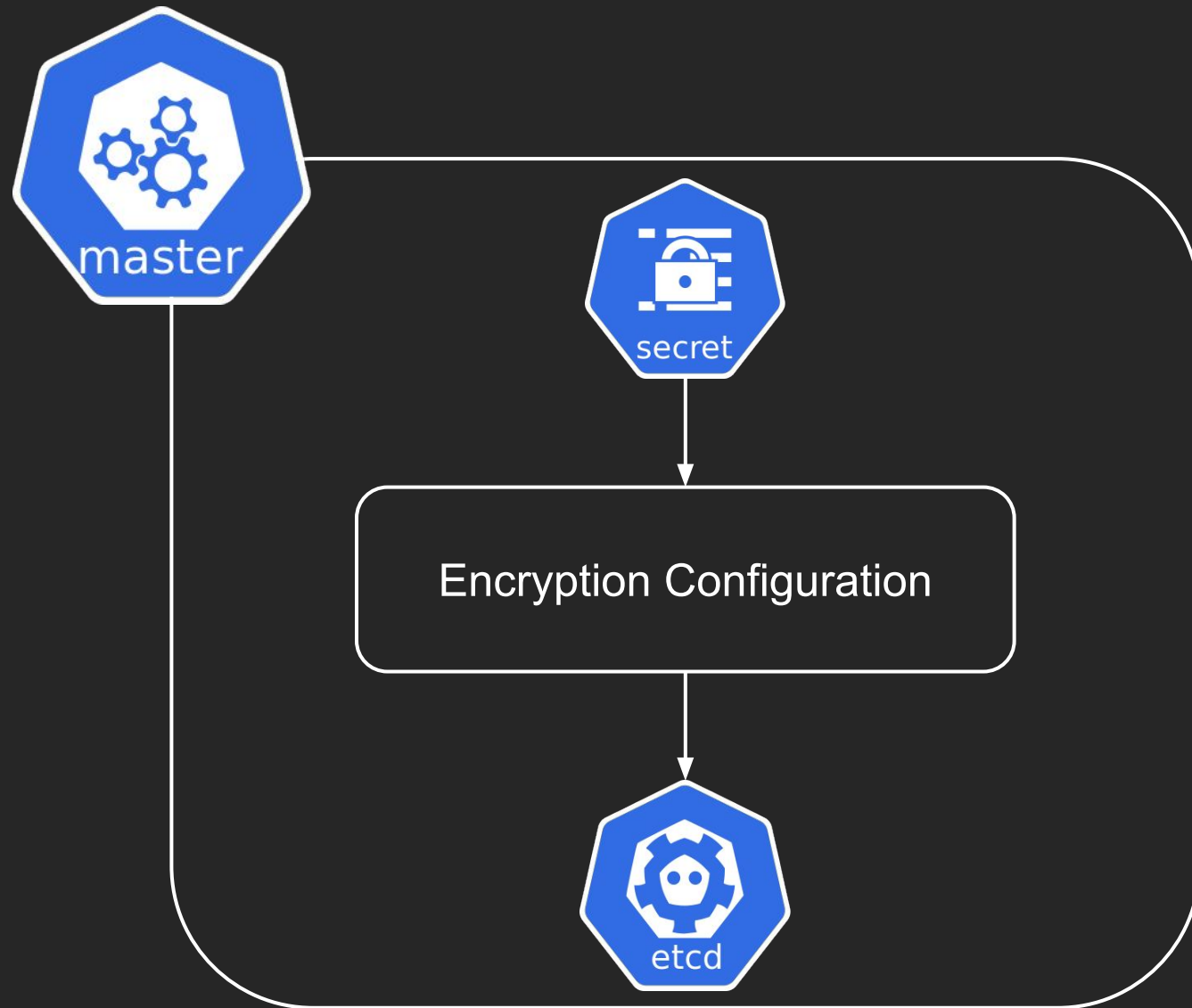


- Kubernetes object that contains a small amount of sensitive data.
- Injected as volume or environment variable.

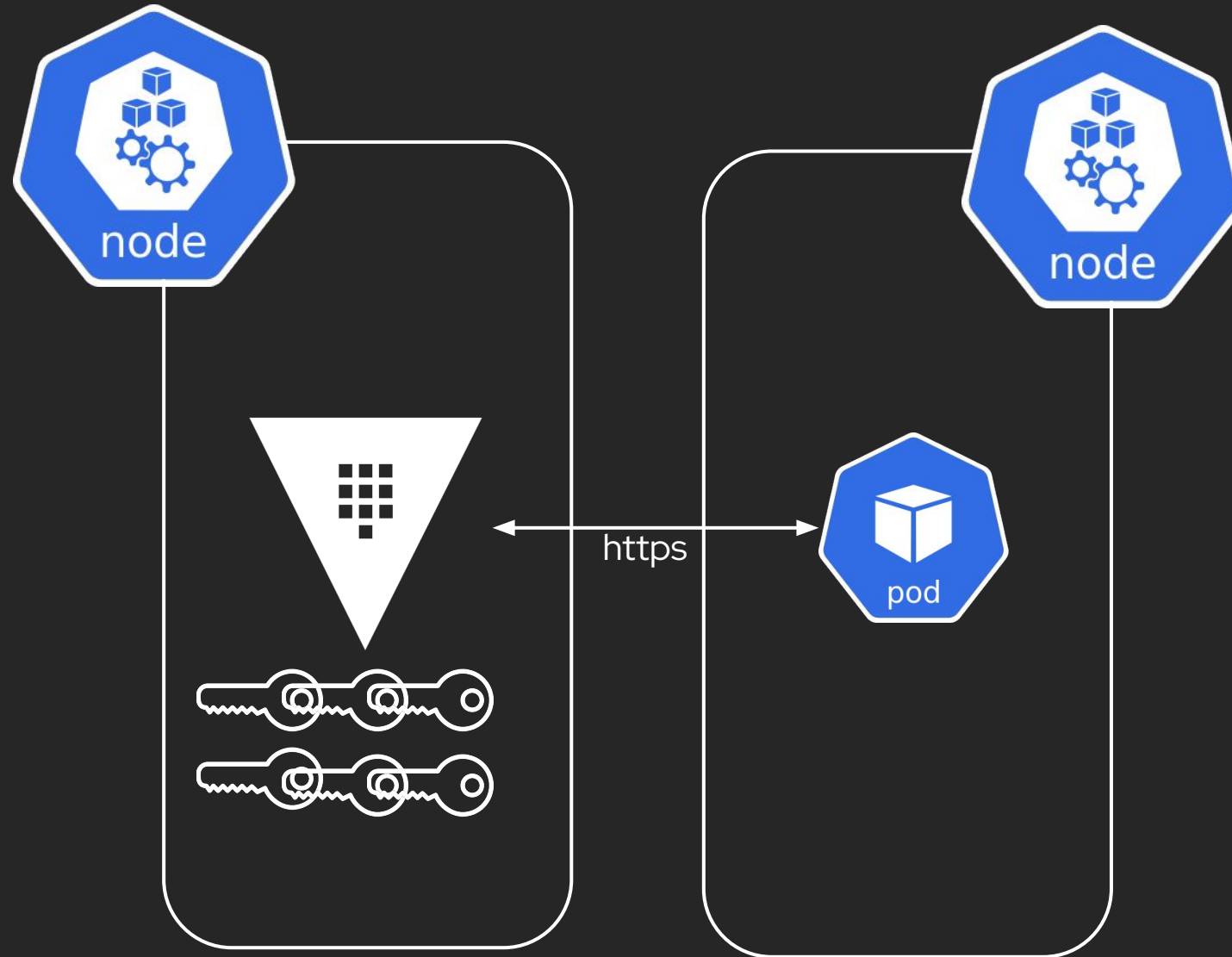
All YAMLS and base64



ConfigMap & Secrets



Vaults provide full encryption





Devoxx France 2023

3 - ADVANCED K8S

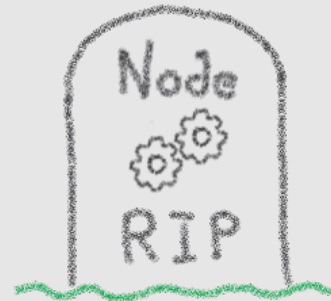
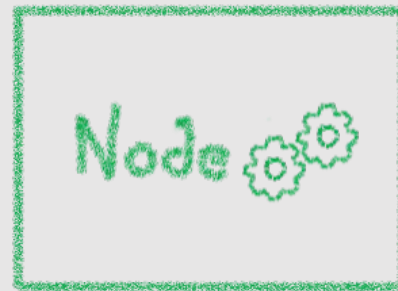
Persistent Volumes

StatefulSets & Jobs (ou JKube)

Tolerance and taints

Operators

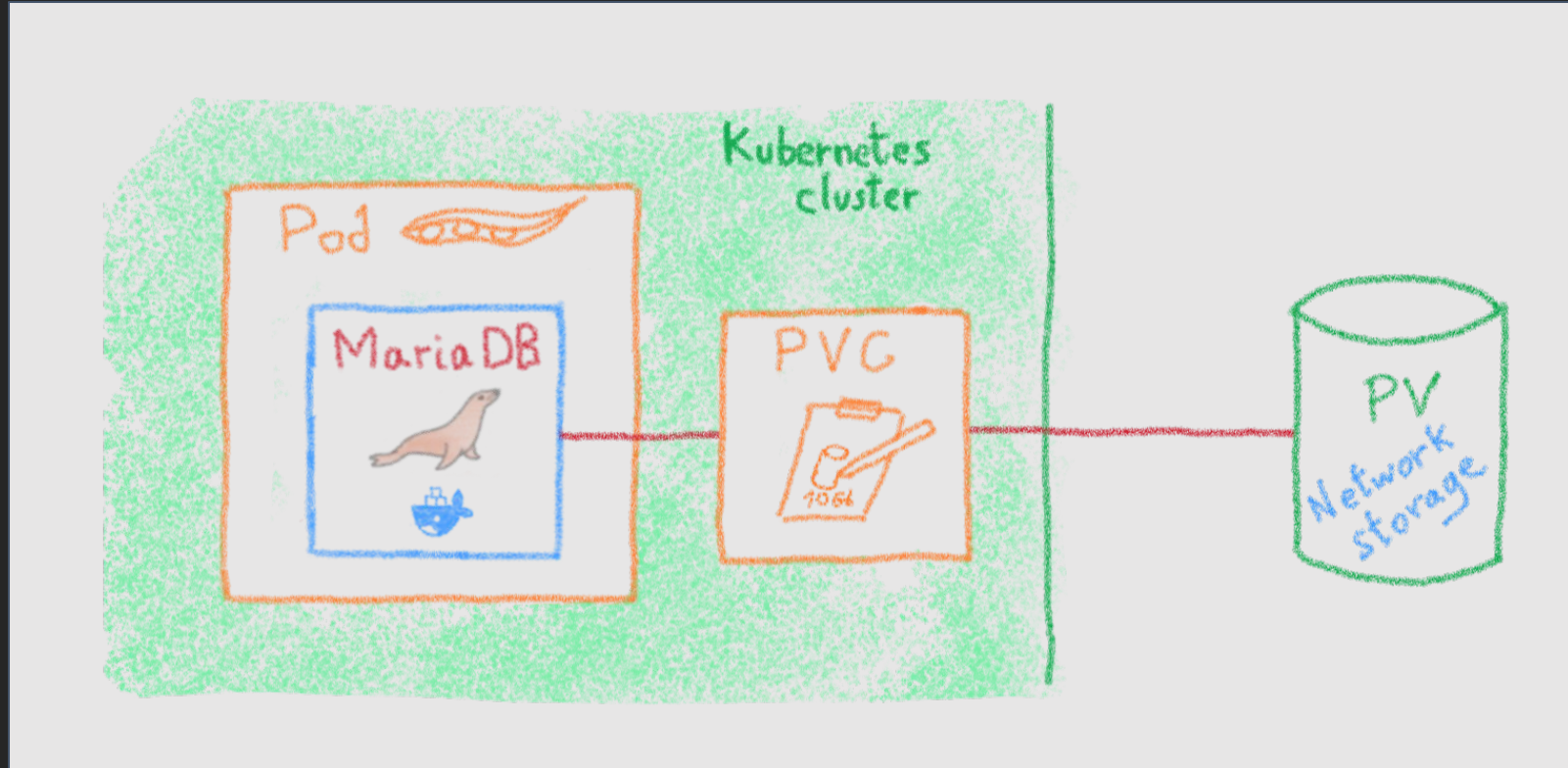
Local storage is a BAD idea



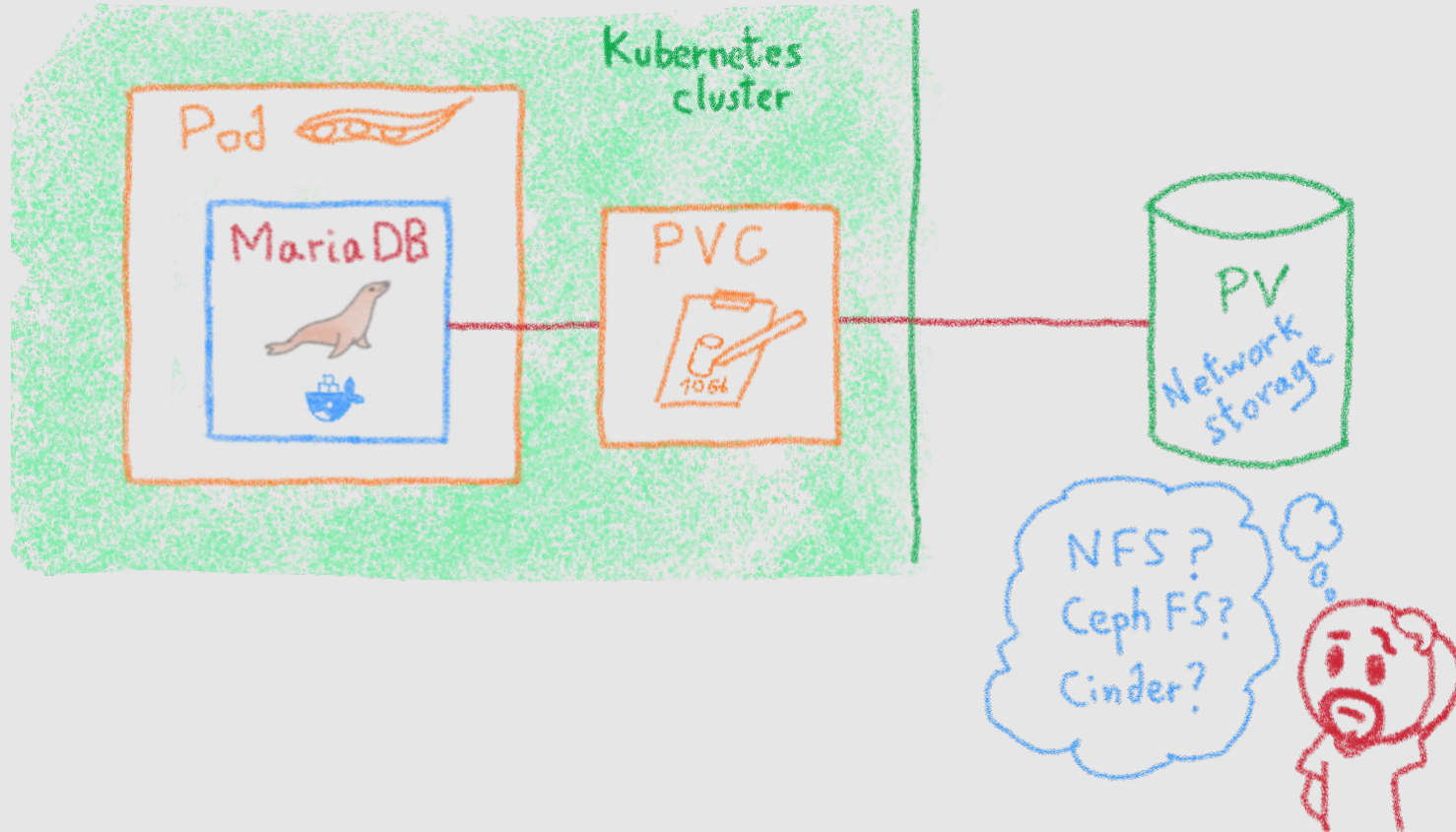
Pods & Nodes are transient, they can and will die

Persistent volumes

Persistent Volumes



The storage dilemma





Devoxx France 2023

3 - ADVANCED K8S

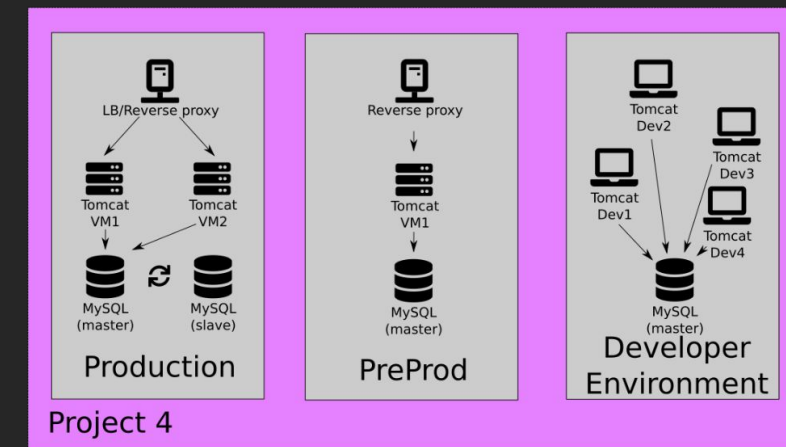
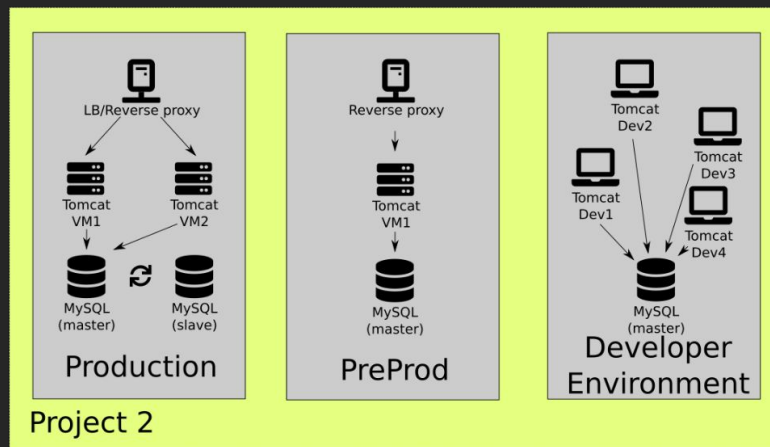
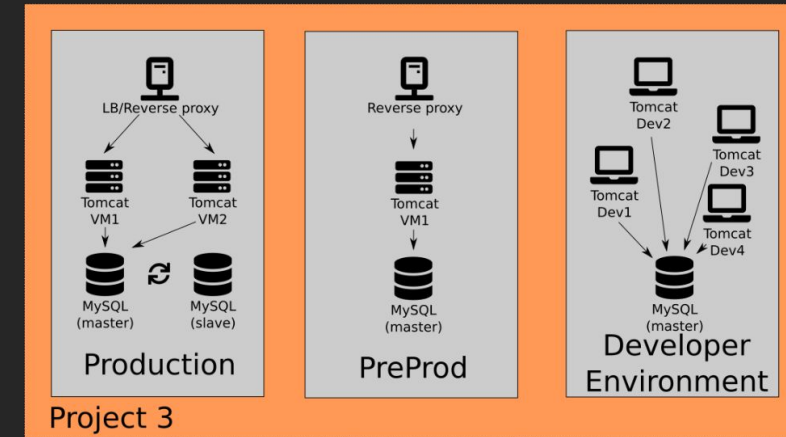
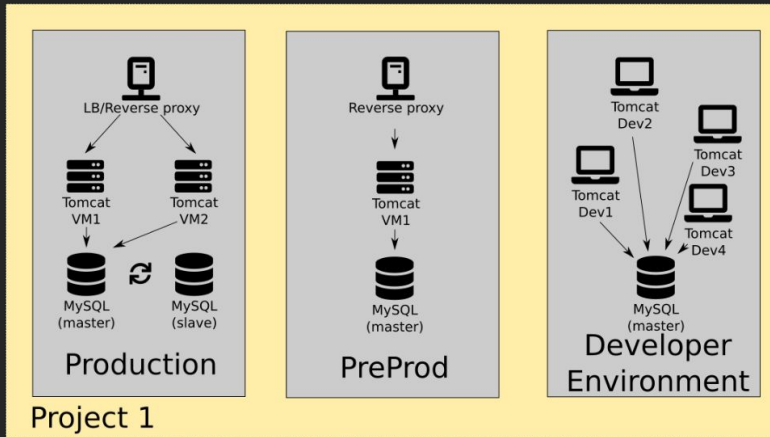
Persistent Volumes

Jkube

Tolerance and taints

Operators

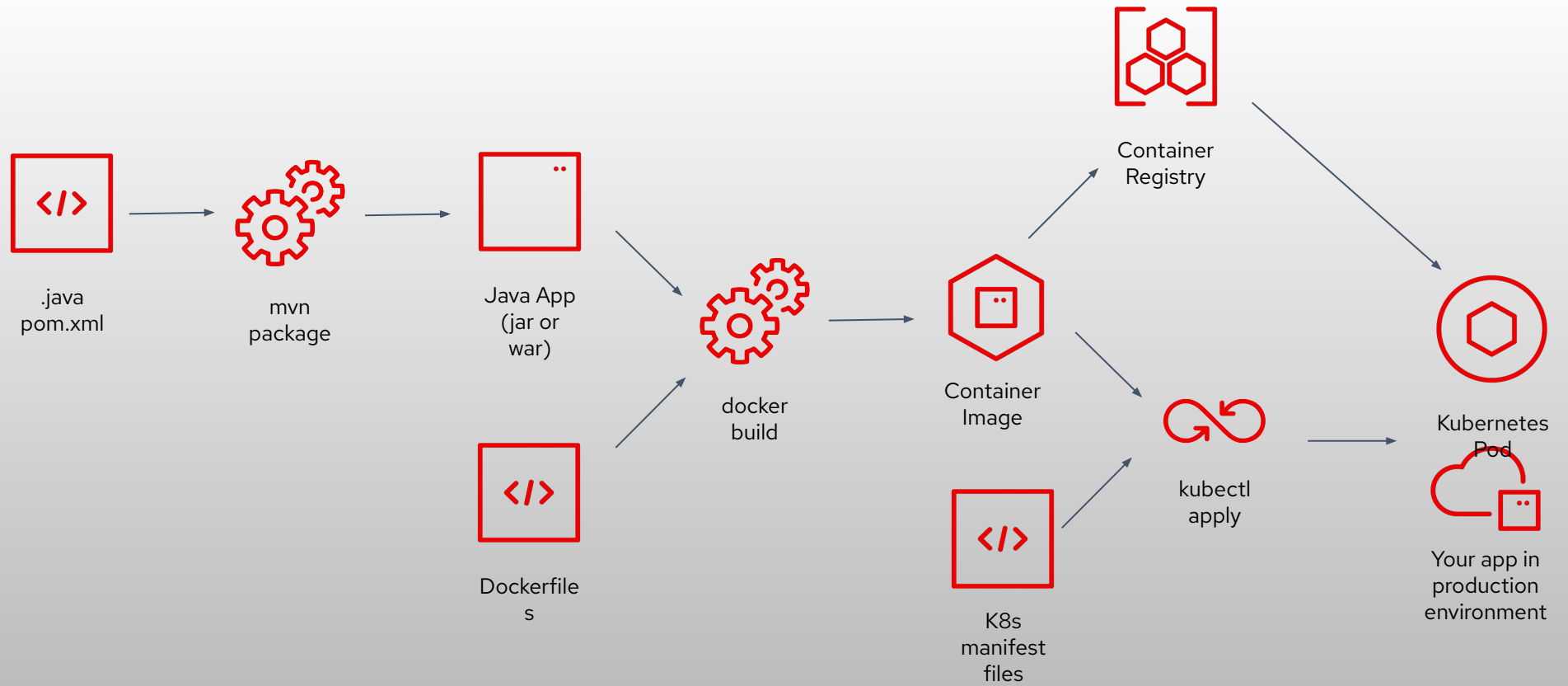
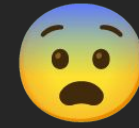
JKUBE



A typical Java application

Based on real life experiences

Your Java to Kubernetes



JKUBE

Eclipse JKube



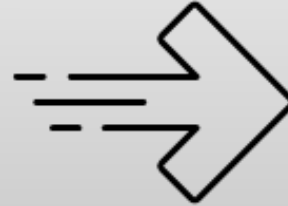
JKube

Eclipse JKube

VERT.x



JKube



Maven



Gradle



OPENSIFT



JAKARTA EE



MICROPROFILE





JKUBE

Eclipse JKube

<https://jkube-website-quickstarts.surge.sh/jkube/>

JKube with Ingress and 5 replicats

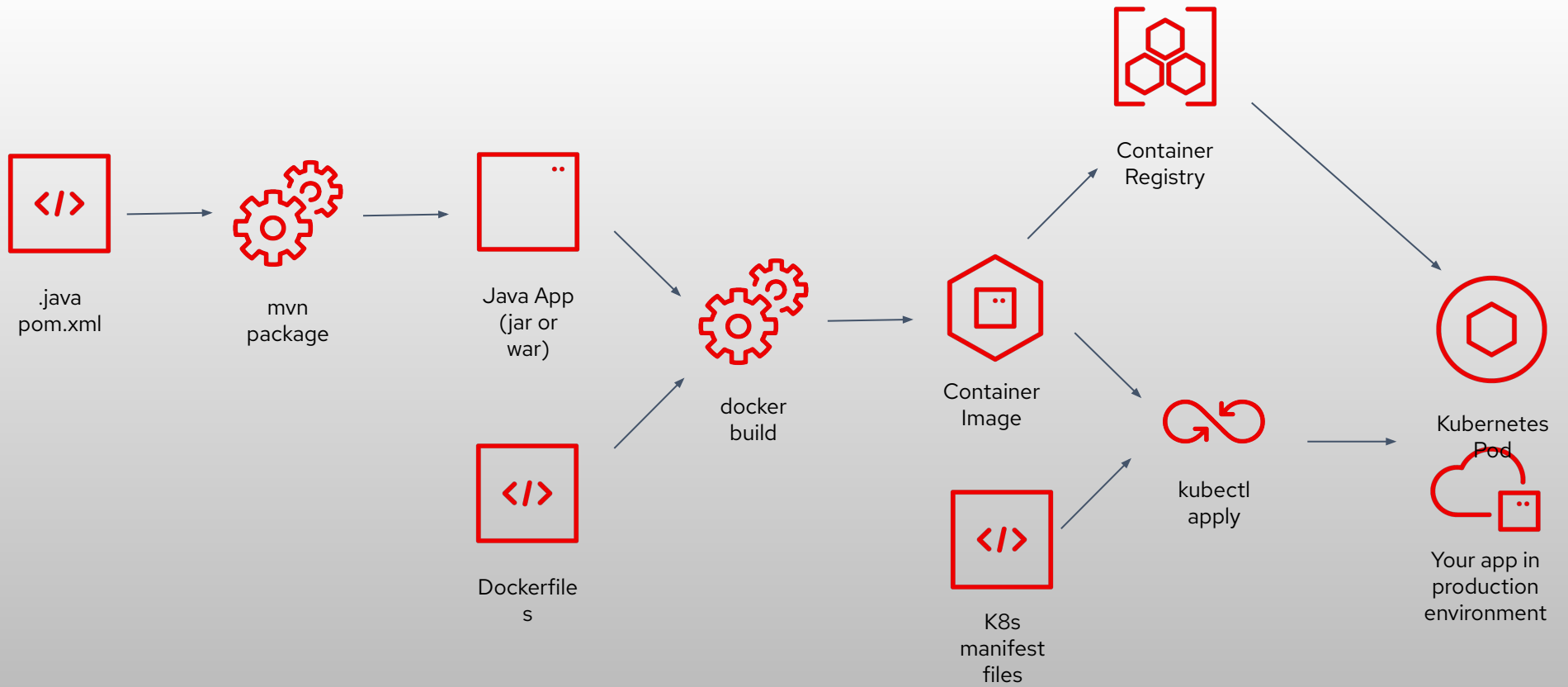
- pom.xml

```
<plugin>
  <groupId>org.eclipse.jkube</groupId>
  <artifactId>kubernetes-maven-plugin</artifactId>
  <version>1.10.1</version>
</plugin>
```

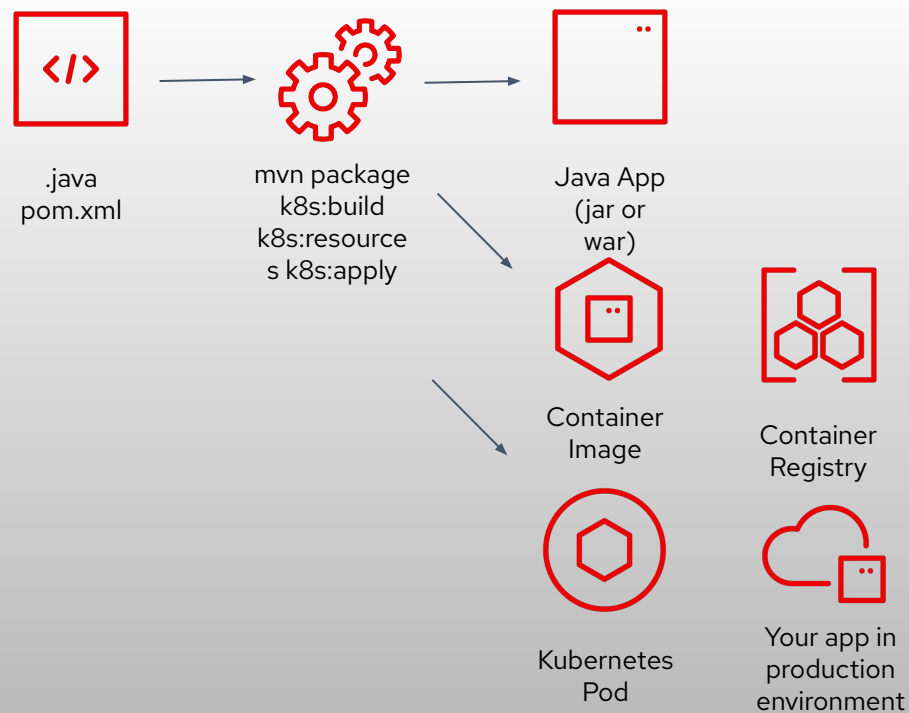
```
<jkube.createExternalUrls>>true</jkube.createExternalUrls>
<jkube.replicas>5</jkube.replicas>
```

```
mvn clean install k8s:build \
      k8s:resource \
      k8s:apply \
      -Djkube.domain=$(minikube ip).nip.io
```

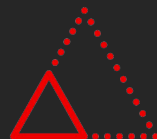
Your Java to Kubernetes



Your Java to Kubernetes with JKube



Eclipse JKube benefits



Minimize the Dockerfile or YAML files to maintain

It generates container images and kubernetes manifests based on defaults. Of course these defaults could be overridden or enriched.



Use your CI

You can use JKube without docker or kubectl. It perfectly fits to any CI



One tool to deploy them all !

It doesn't require Docker or kubectl. JIB or S2i can be used for build. Kubernetes communication is done with the Java Fabric8 Kubernetes client.





DEVOXX FRANCE

11ème édition | Palais des Congrès, Paris



Neuilly 252 AB

Mercredi 12 avril | 18:30 - 19h

JKube remote-dev :

Coder avec tous vos micro-services
du cloud... en local !

Sun Tan





Devoxx France 2023

3 - ADVANCED K8S

Persistent Volumes

StatefulSets & Jobs (JKube)

Tolerance and taints

Operators

Taints & Tolerances

Taint - applied to a Kubernetes Node that signals the scheduler to avoid or not schedule certain Pods.

Toleration - applied to a Pod definition and provides an exception to the taint.

Affinity & Anti-affinity

Node Affinity - rules that ban the node where pods can run, but also to favor where they should run

Pod Affinity - indicate a group of pods should always be deployed together on the same node (because of network communication, shared storage, etc.)



Devoxx France 2023

3 - ADVANCED K8S

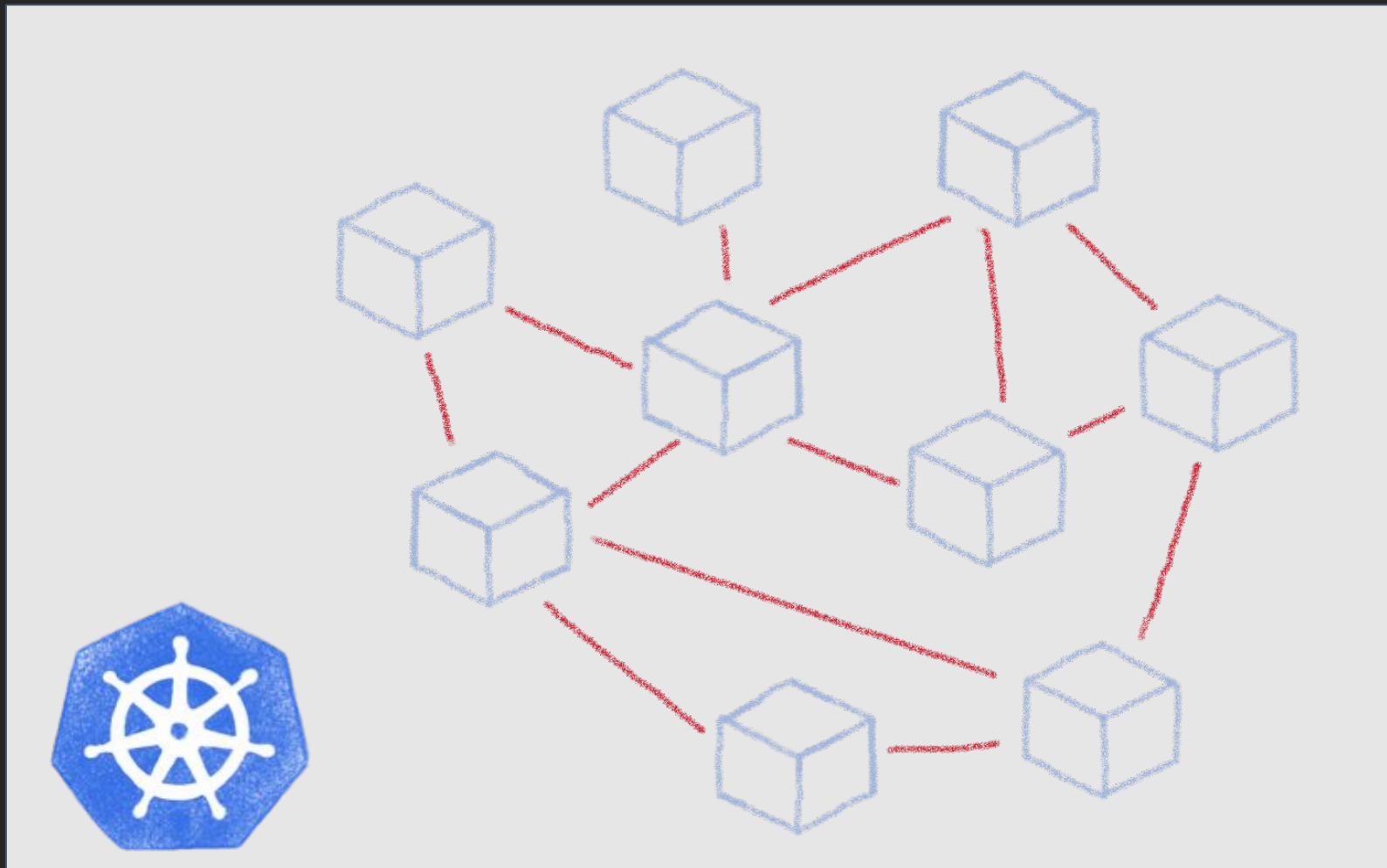
Persistent Volumes

StatefulSets & Jobs

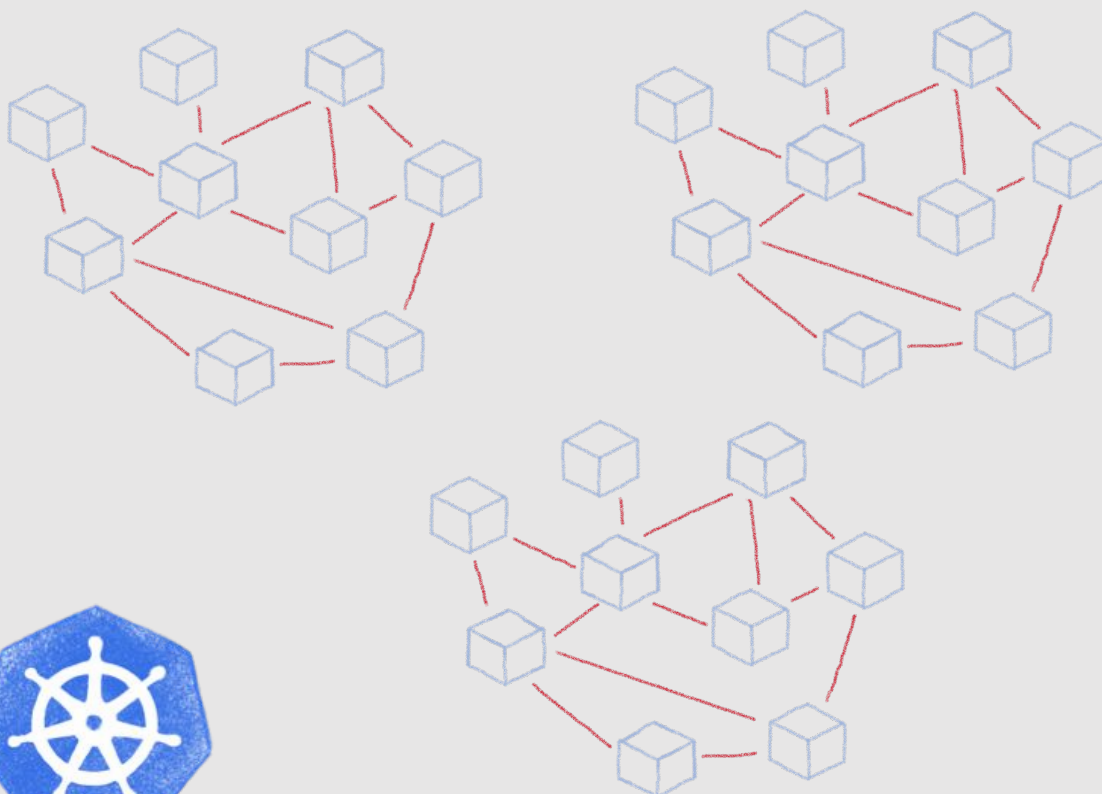
Tolerance and taints

Operators

Taming microservices with Kubernetes



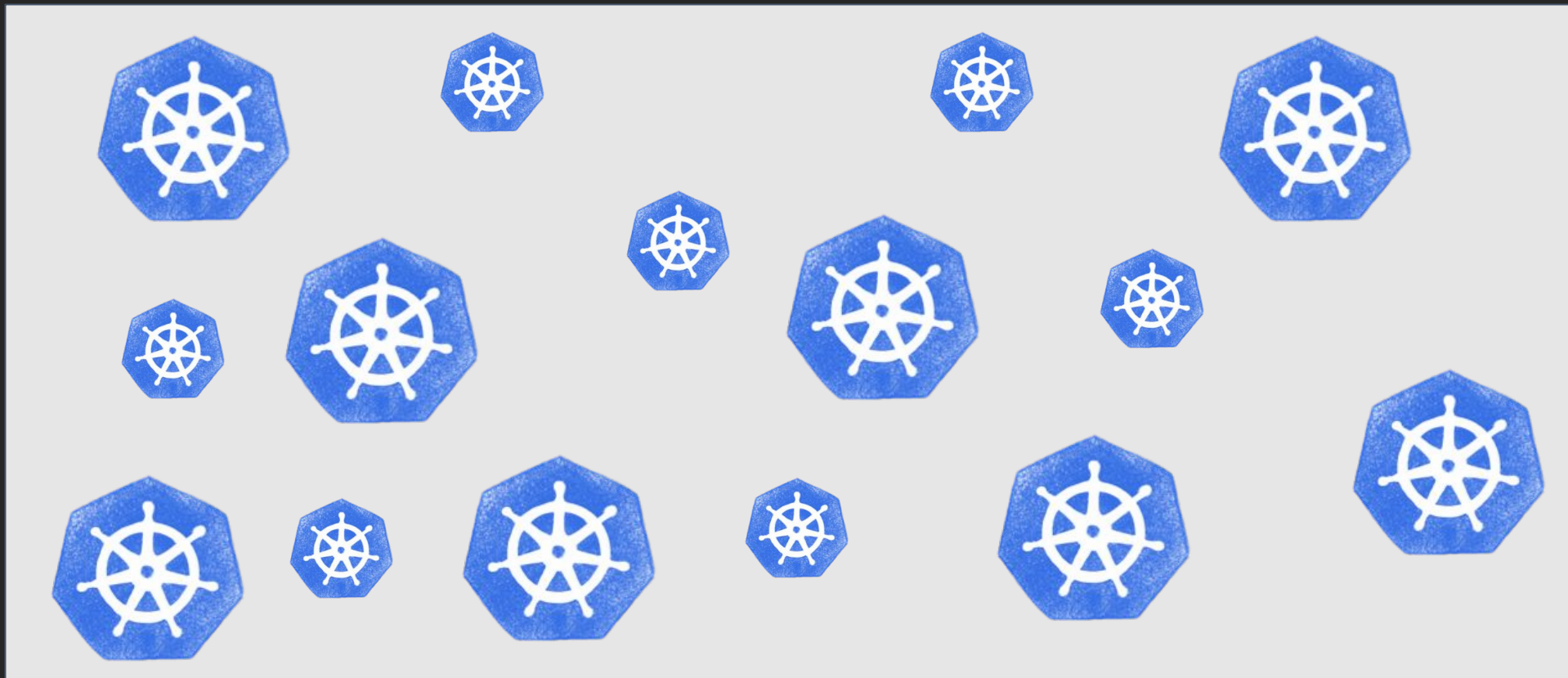
What about complex deployments?



The diagram illustrates a complex deployment architecture consisting of three interconnected clusters of pods. Each cluster is represented by a group of blue cube-shaped nodes connected by red lines. The clusters are arranged in a triangular pattern, with each cluster connected to the other two. In the bottom left corner of the diagram area is the Kubernetes logo (a blue shield with a white ship's wheel). In the bottom right corner is a red hand-drawn cartoon character with a confused expression, one hand on its chin, and a question mark above its head.

- Ingress
- Services
- Deployments
- Pods
- Sidecars
- Replica Sets
- Stateful Sets

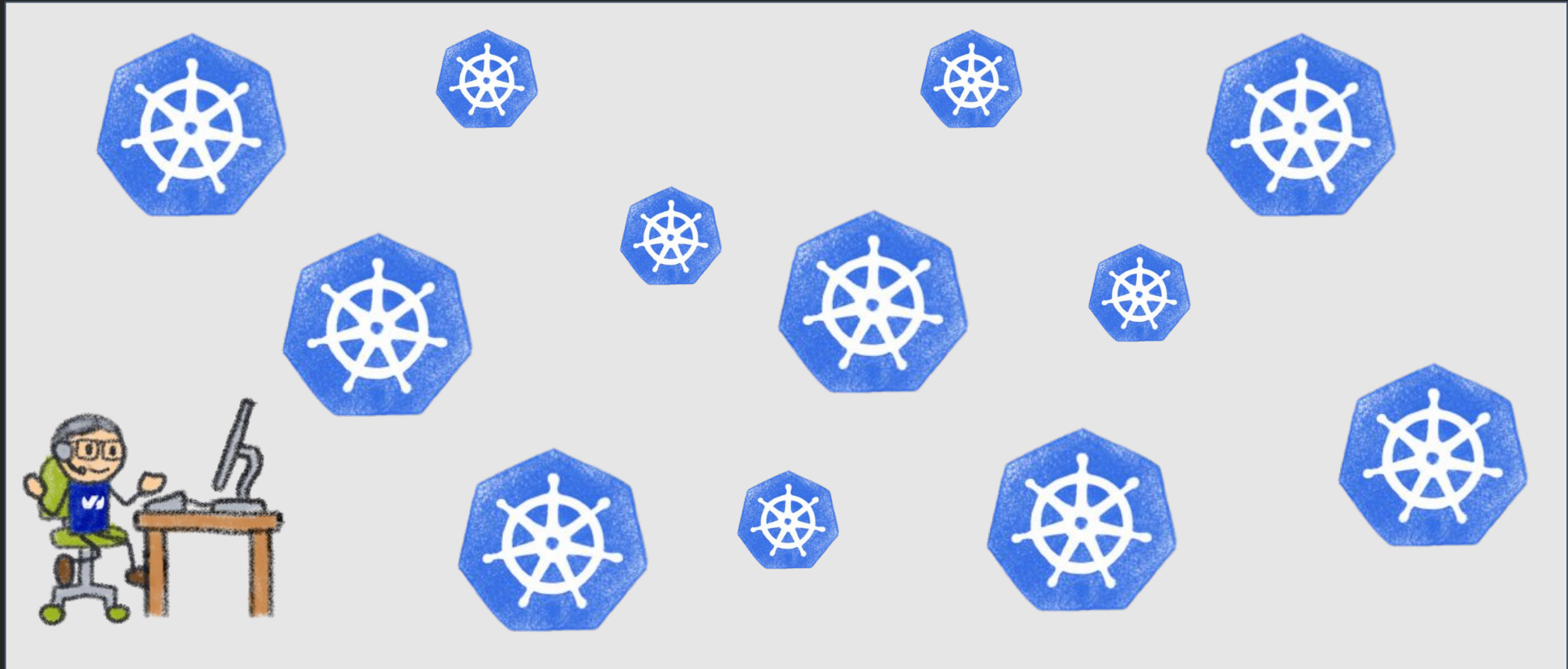
Specially at scale



Lots of clusters with lots and lots of deployments

Operators

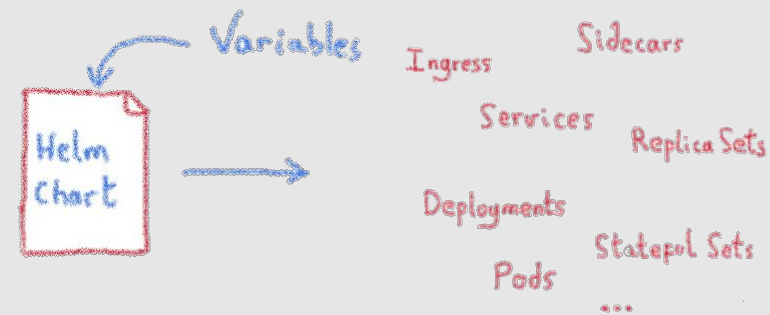
We need to tame the complexity



Making it easier to operate

Taming the complexity

A package manager for Kubernetes



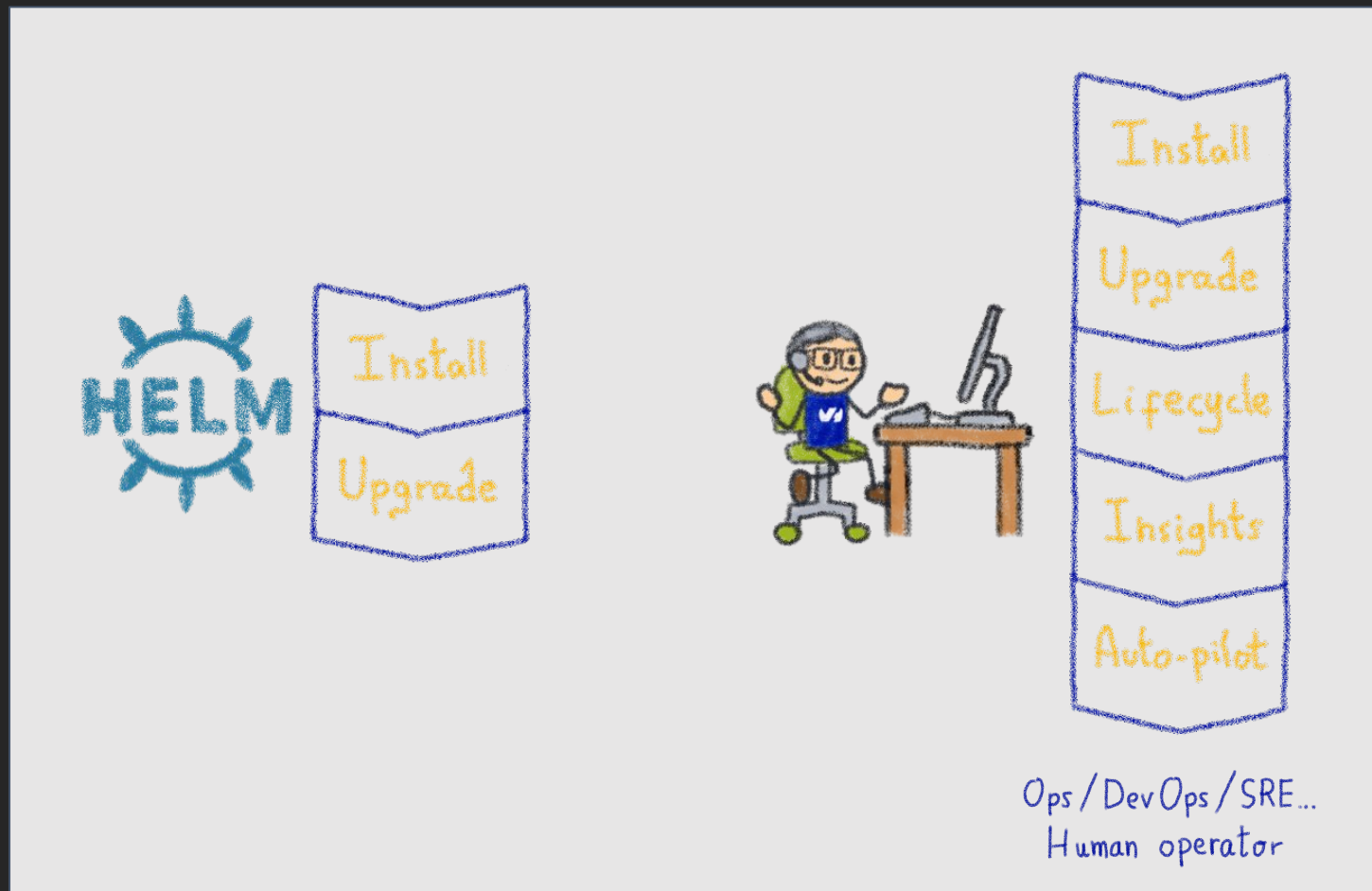
- Manage complexity 

- Easy upgrades 

- Simple sharing 

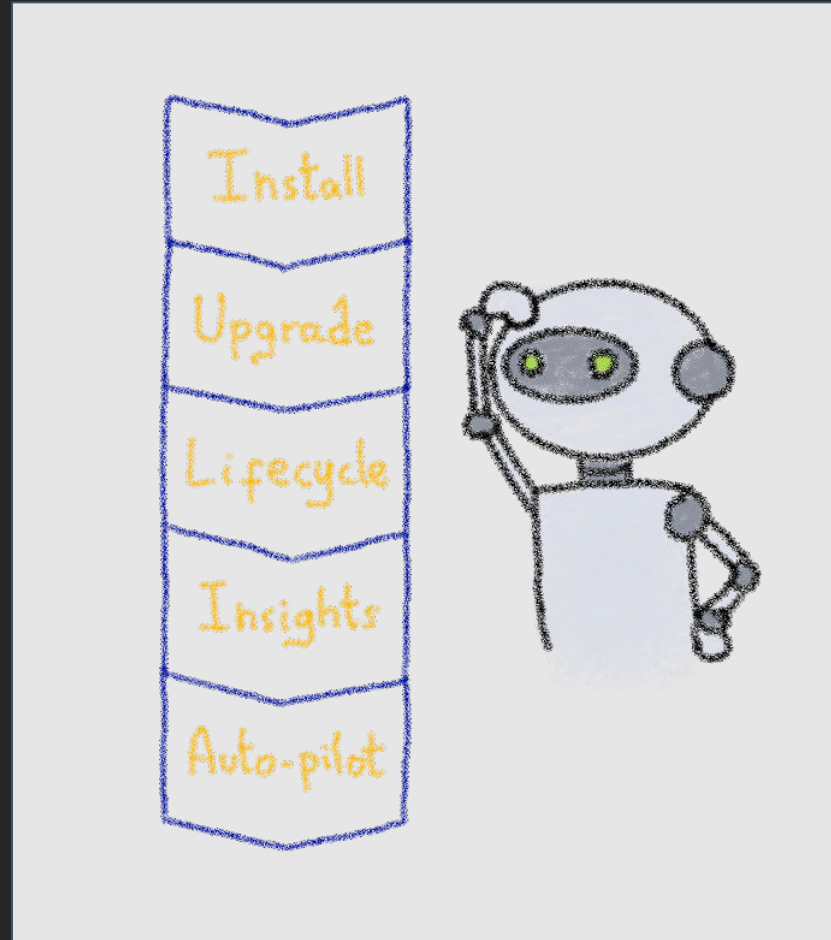
- Easy rollbacks 

Helm Charts are configuration



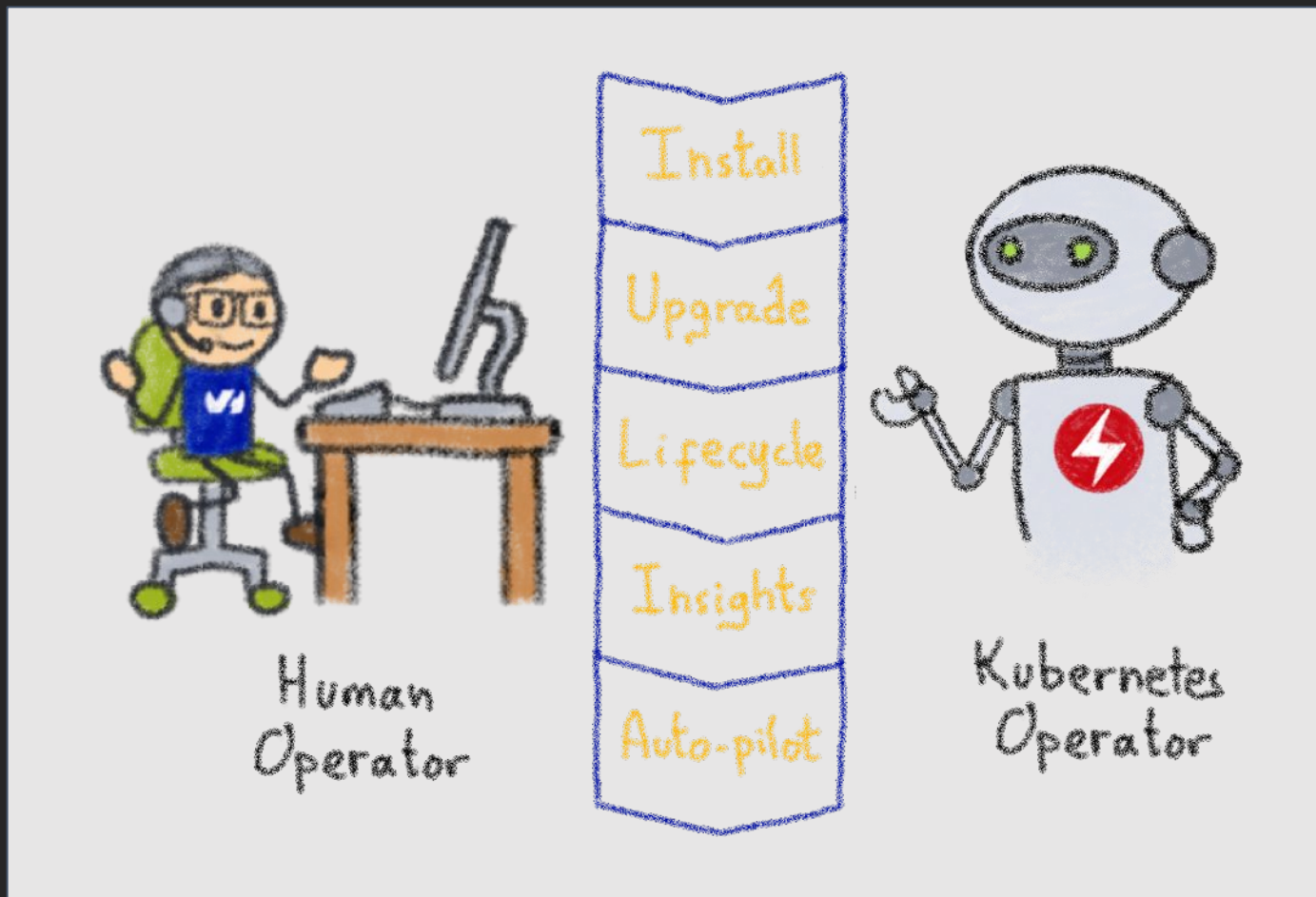
Operating is more than installs & upgrades

Kubernetes is about automation



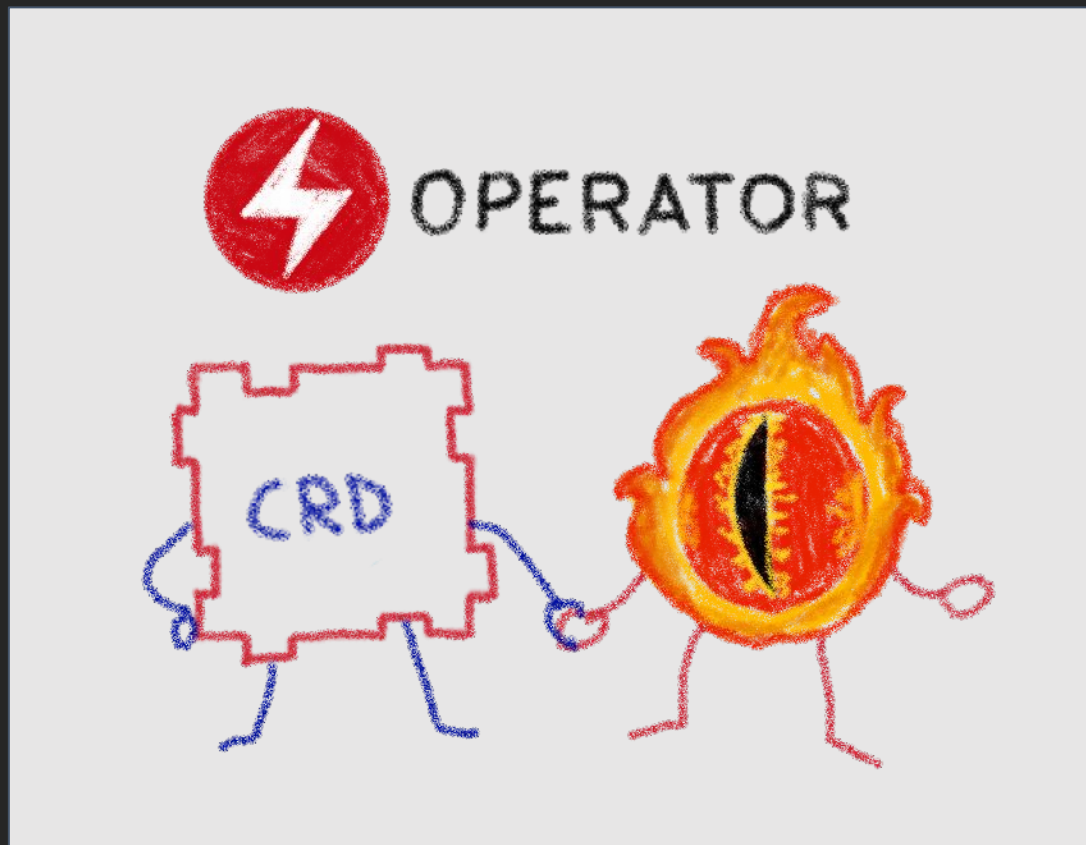
How about automating human operators?

Kubernetes Operators



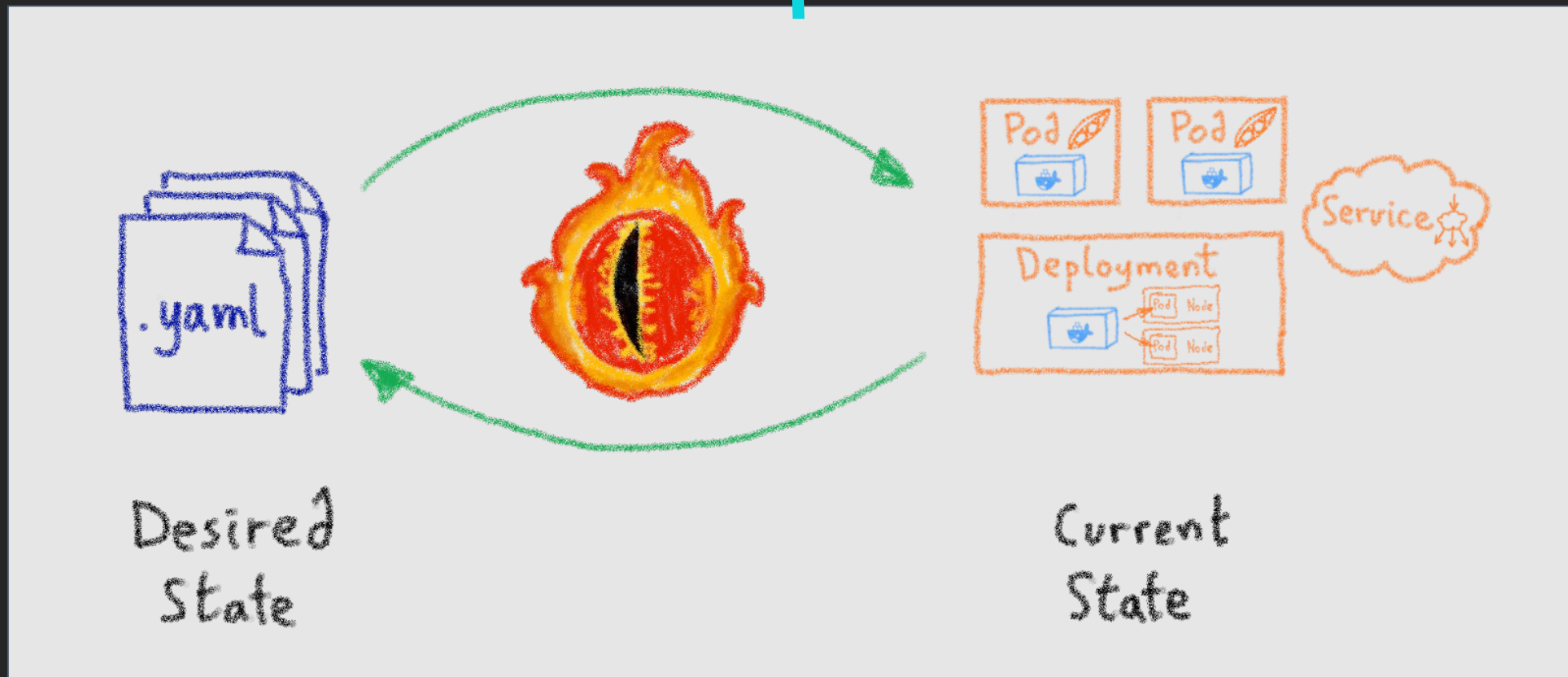
A Kubernetes version of the human operator

Building operators



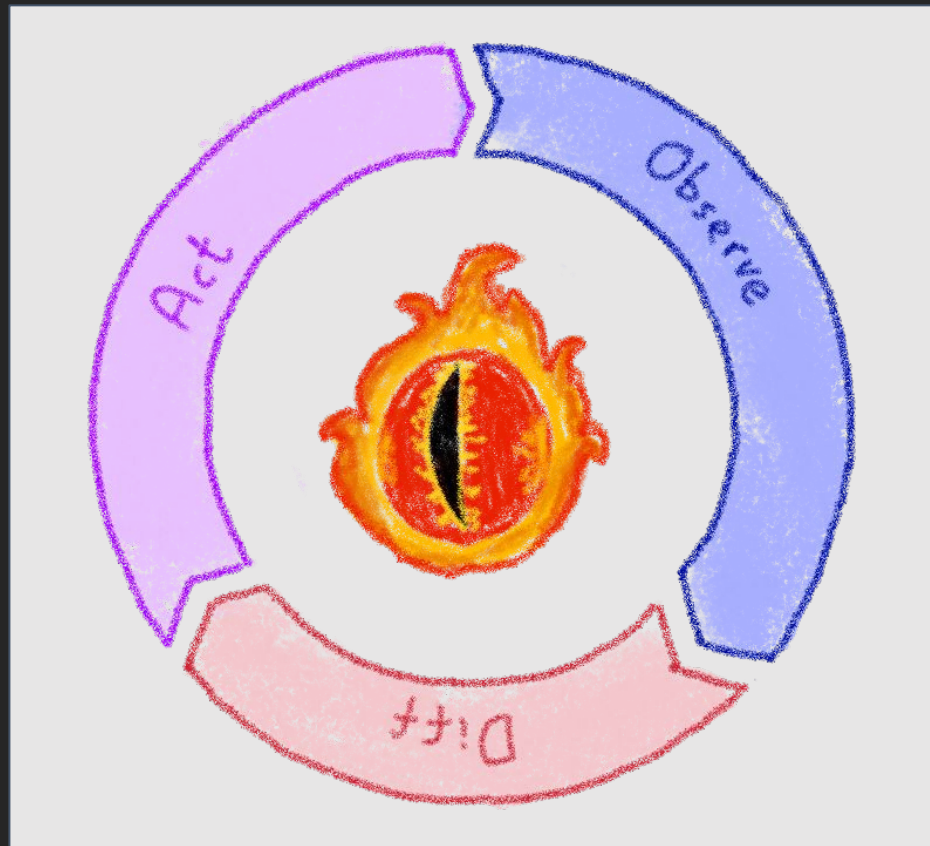
Basic K8s elements: Custom Resources & Controllers

Kubernetes Controllers: control loops



They watch the state of the cluster,
and make or request changes where needed

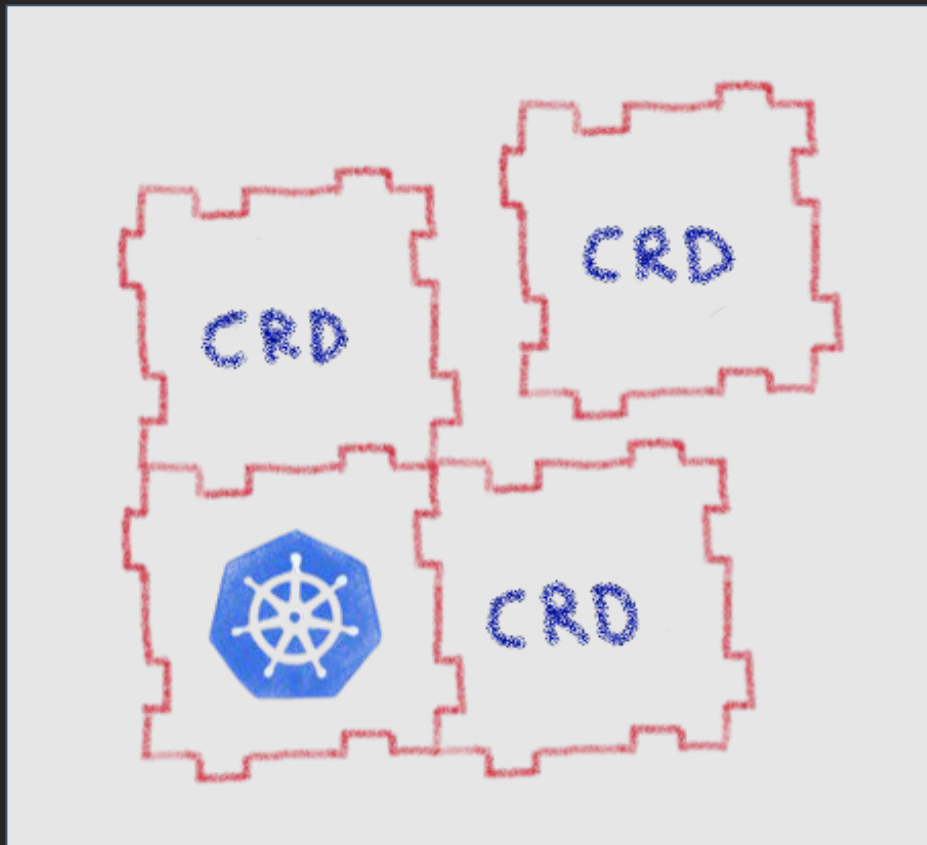
KBs controllers: a reconcile loop



Strives to reconcile current state and desired state

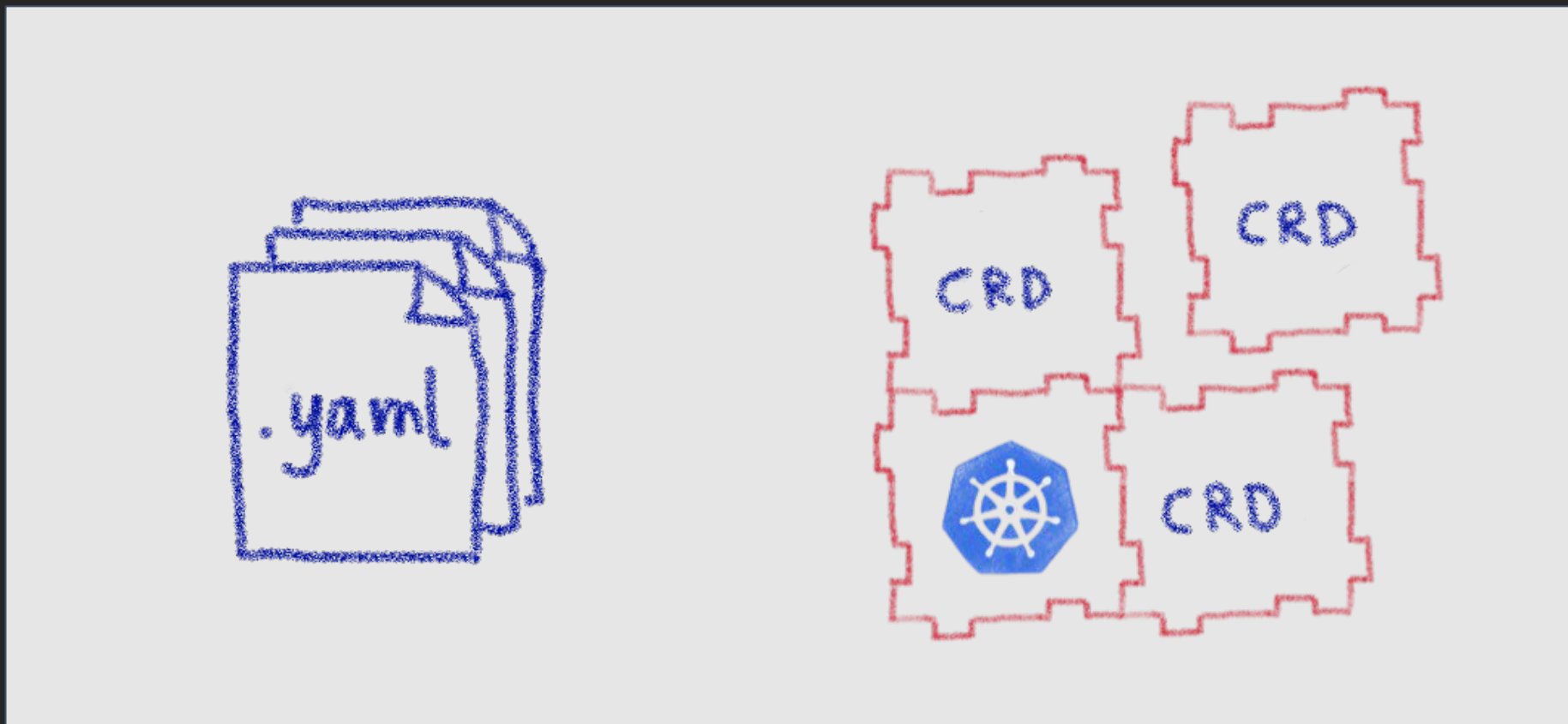
Custom Resource Definitions

Operators



Extending Kubernetes API

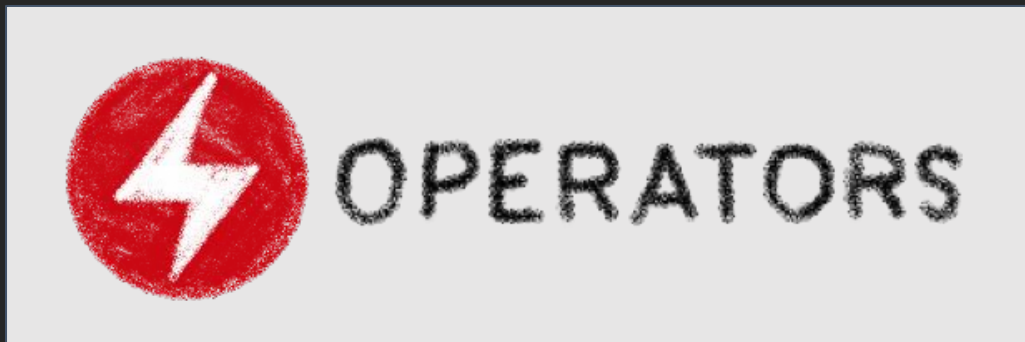
Extending Kubernetes API



By defining new types of resources

Operators

Kubernetes Operators

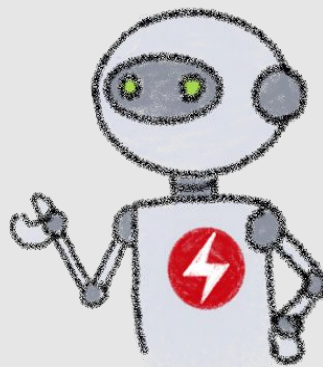
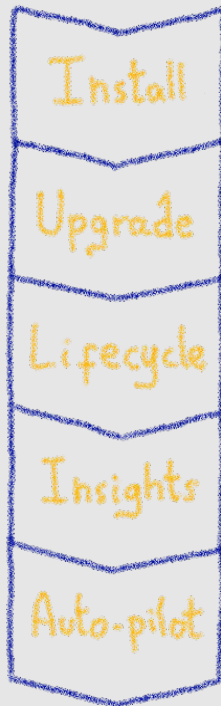


Automating operations

What's a Kubernetes Operator?



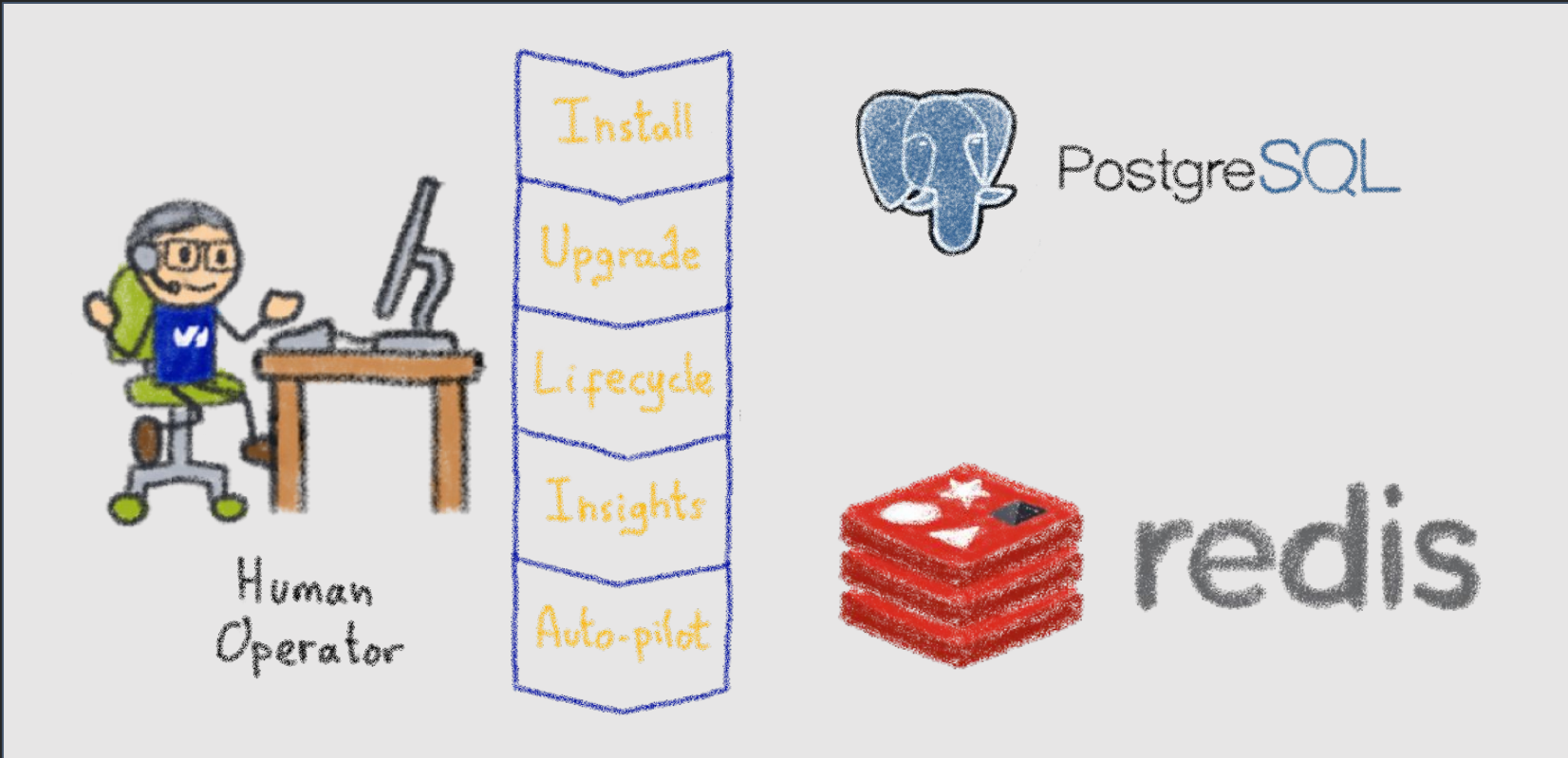
Human Operator



Kubernetes Operator

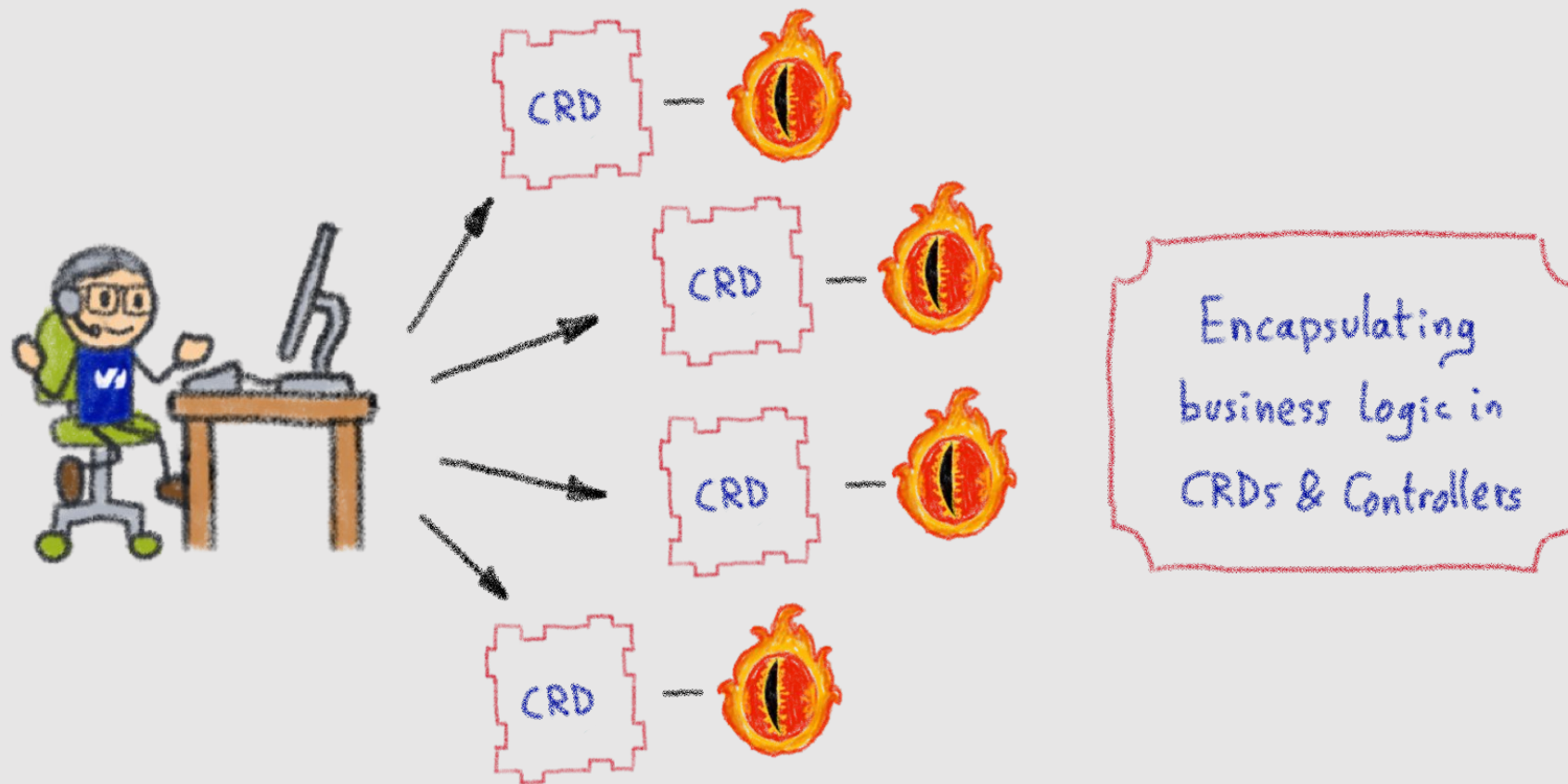
An Operator represents human operational knowledge in software to reliably manage an application

Example: databases

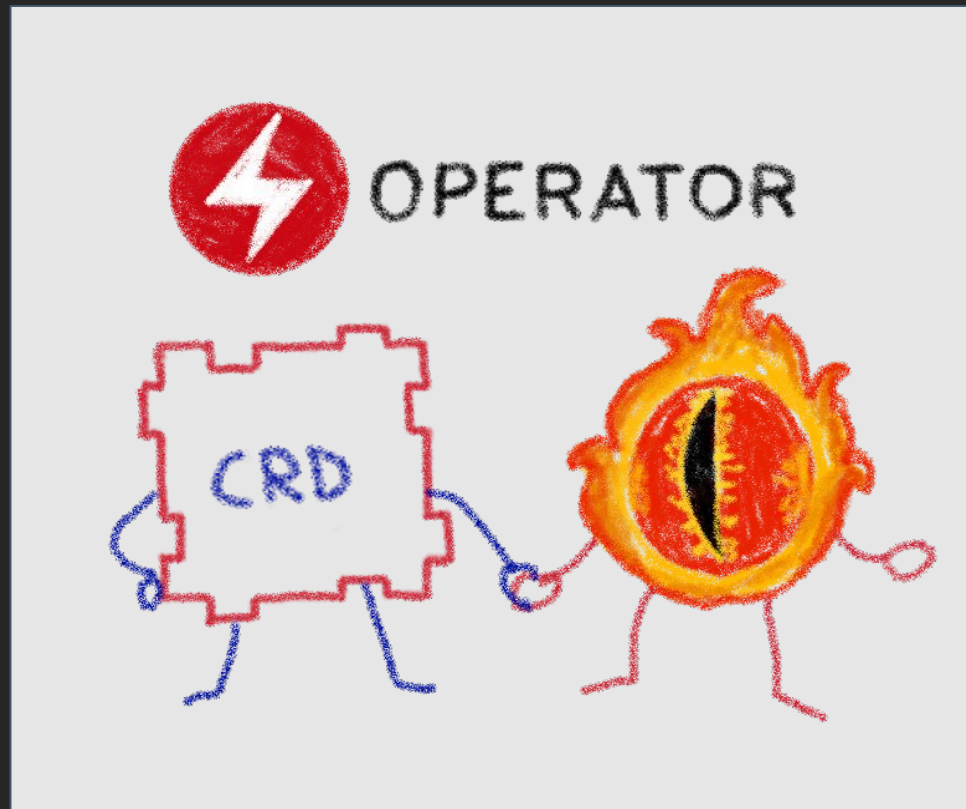


Things like adding an instance to a pool,
doing a backup, sharding...

Knowledge encoded in CRDs and Controllers

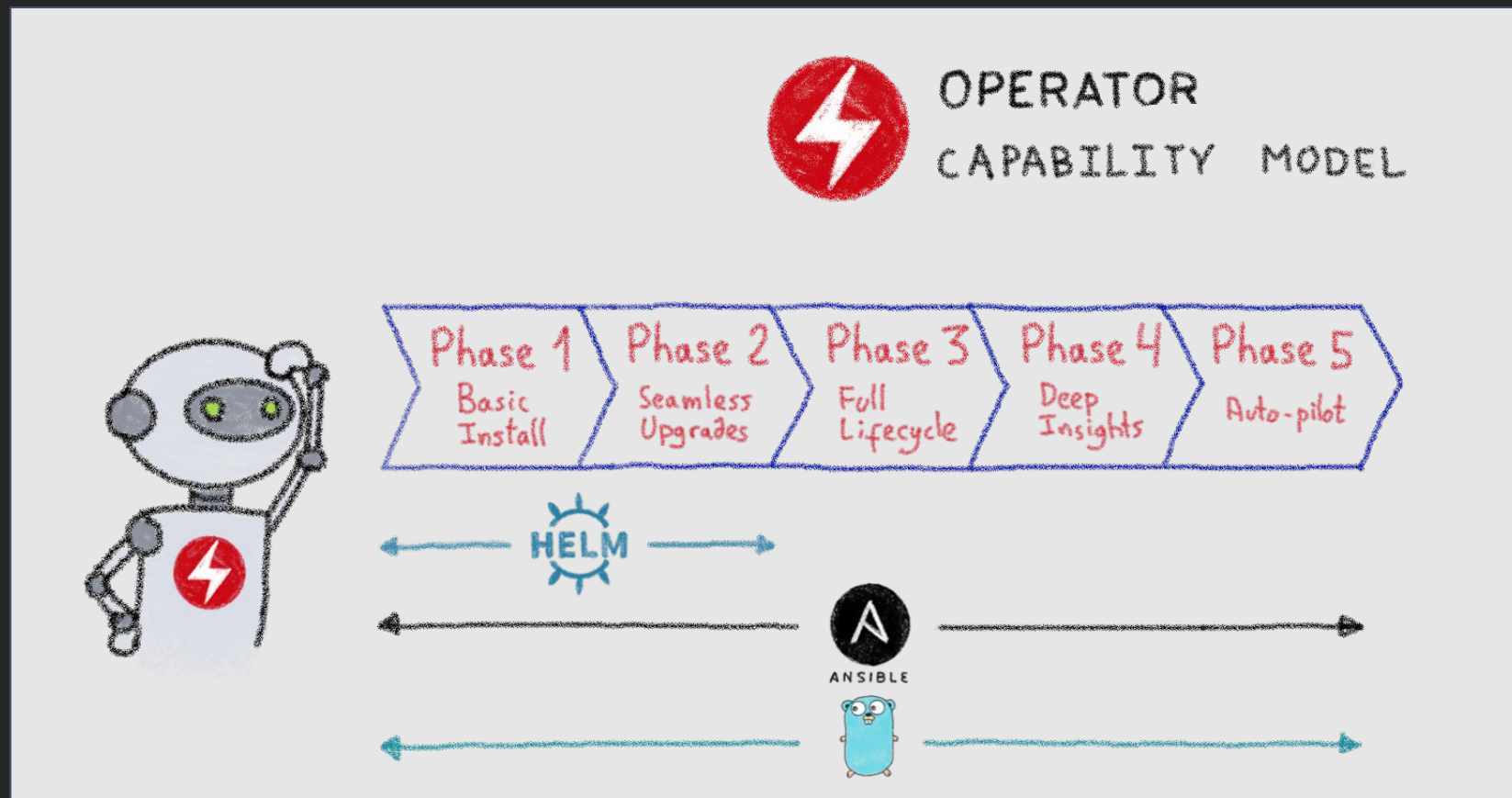


Custom Controllers for Custom Resources



Operators implement and manage Custom Resources using custom reconciliation logic

Operator Capability Model



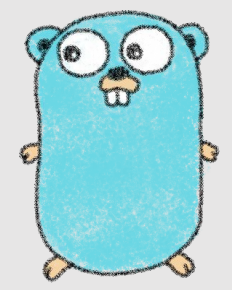
Gauging the operator maturity

The Operator SDK

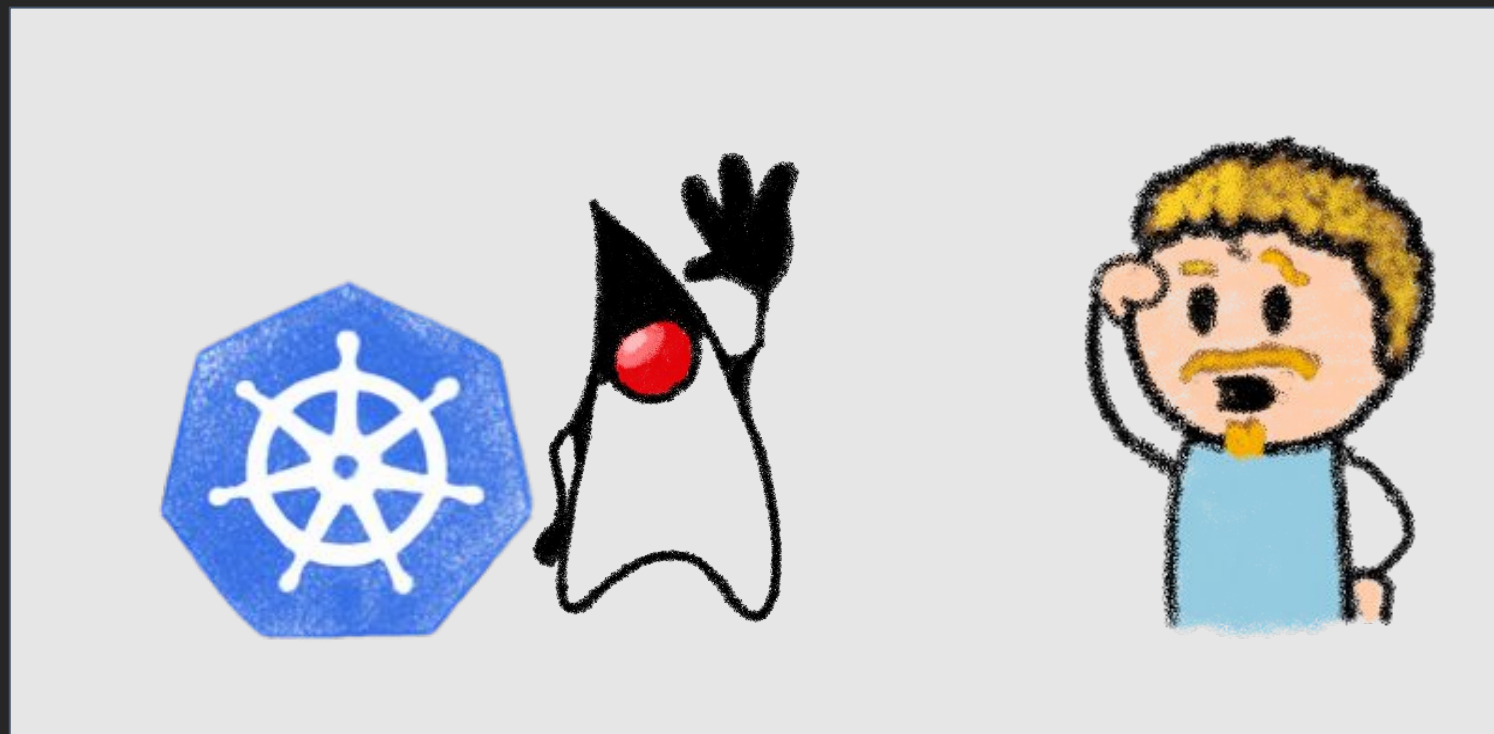


OPERATOR
SDK

BUILD
TEST
ITERATE



But I'm a Java developer!



Can I code Kubernetes Operators in Java?
Easily?

Operators

Operators in Java



The screenshot shows the homepage of the Java Operator SDK. The page has an orange header with a navigation menu containing links for HOME, DOCS, CODE OF CONDUCT, and RELEASES, along with icons for Discord and GitHub. Below the header is a large orange section featuring the Java logo, the text "[JAVA OPERATOR SDK]", and three buttons: "GET STARTED", "CONTRIBUTE" (with a GitHub icon), and "DISCORD CHANNEL" (with a Discord icon). The bottom section is white and contains the text "Sponsored by:" followed by the logos for Container Solutions and Red Hat.

[JAVA OPERATOR SDK]

HOME DOCS CODE OF CONDUCT RELEASES

[JAVA OPERATOR SDK]

GET STARTED CONTRIBUTE DISCORD CHANNEL

Sponsored by:

Container Solutions & Red Hat

Quizz Kahoot

Le livre d'Avrilie





THANKS
FOR WATCHING