



KUBERNETES

DEPLOYMENT FODÁSTICO

DENNY R S VRIESMAN // JUNHO/2018

AGENDA

INTRODUÇÃO
CONCEITOS K8S
GKE & SETUP ENV
MEU PRIMEIRO DEPLOYMENT
INGRESS
HELM
TÓPICOS ESPECIAIS

INTRODUÇÃO



Escopo

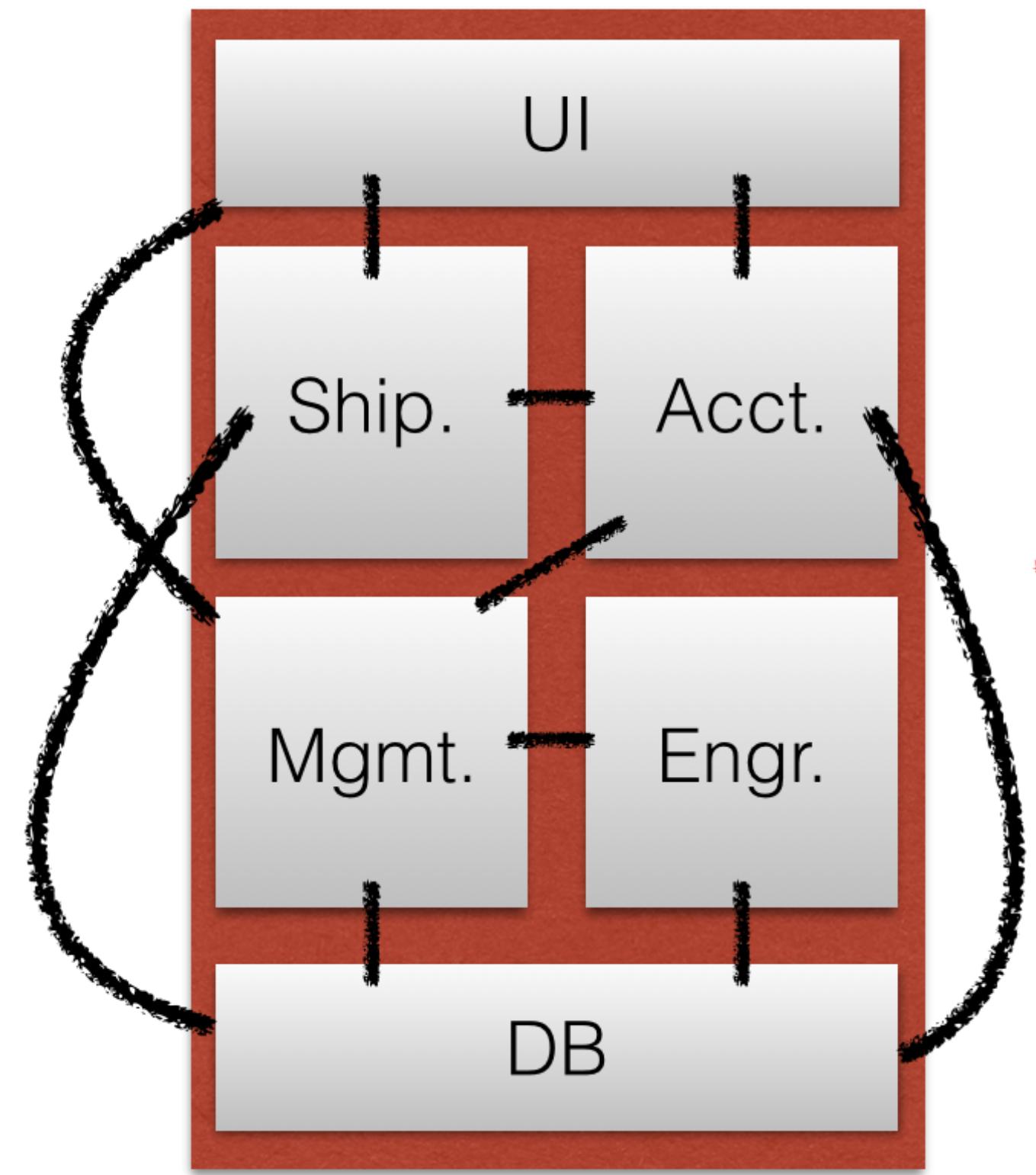


Fora do escopo

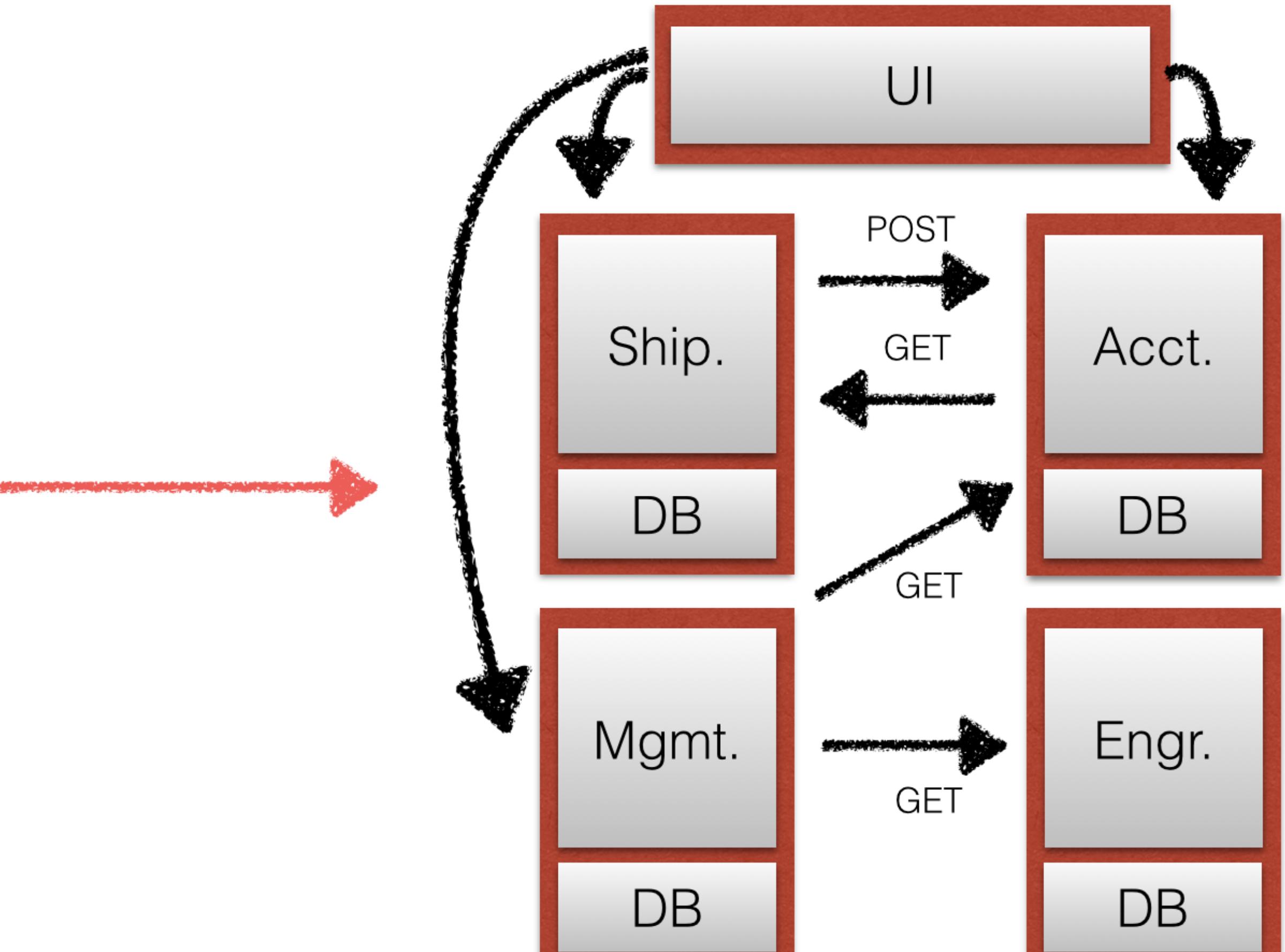


INFRAM

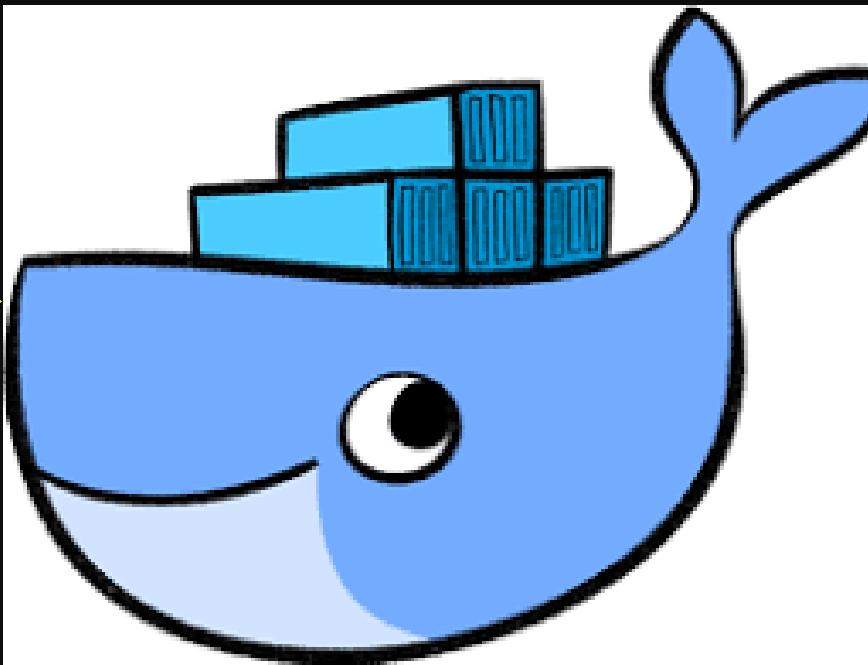
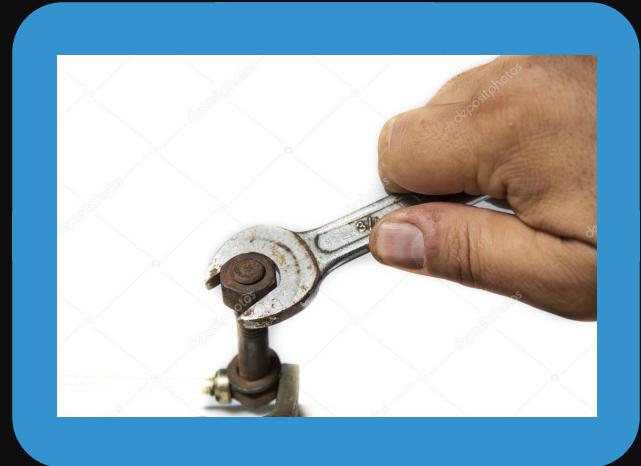
Monolithic



Microservices



historinha - parte 1



0



historinha - parte 2



FATOS



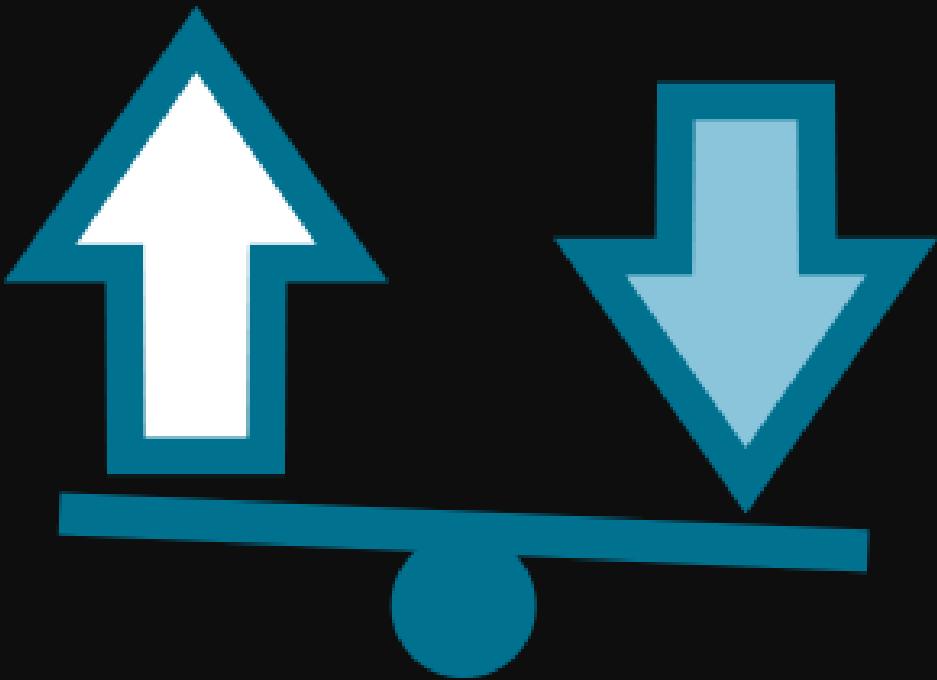
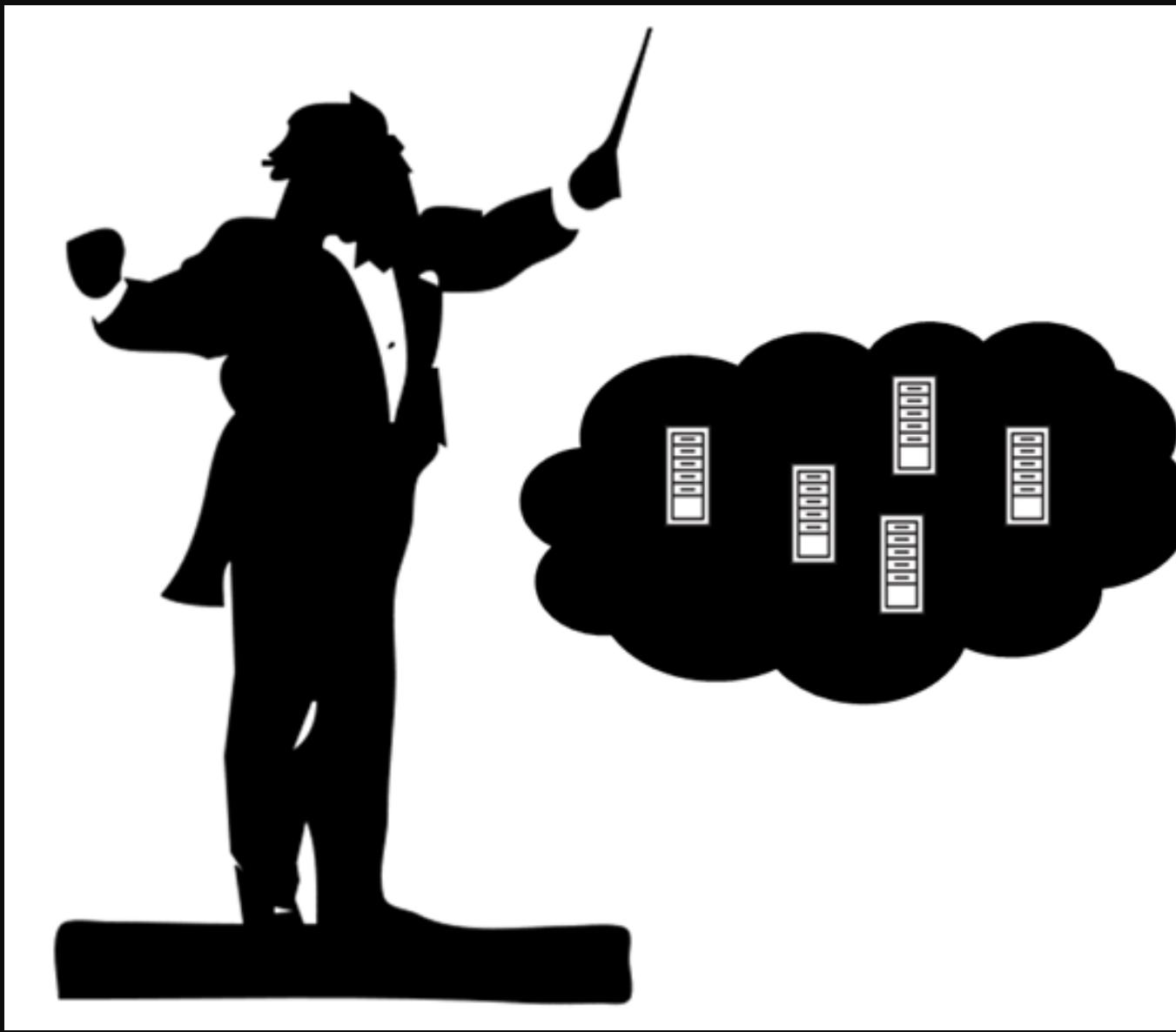




CONCEITOS K8S



do grego Kuvernetes



KIND: POD

Menor unidade de escalonamento



1 OU EXCEPTIONALMENTE MAIS CONTAINERS

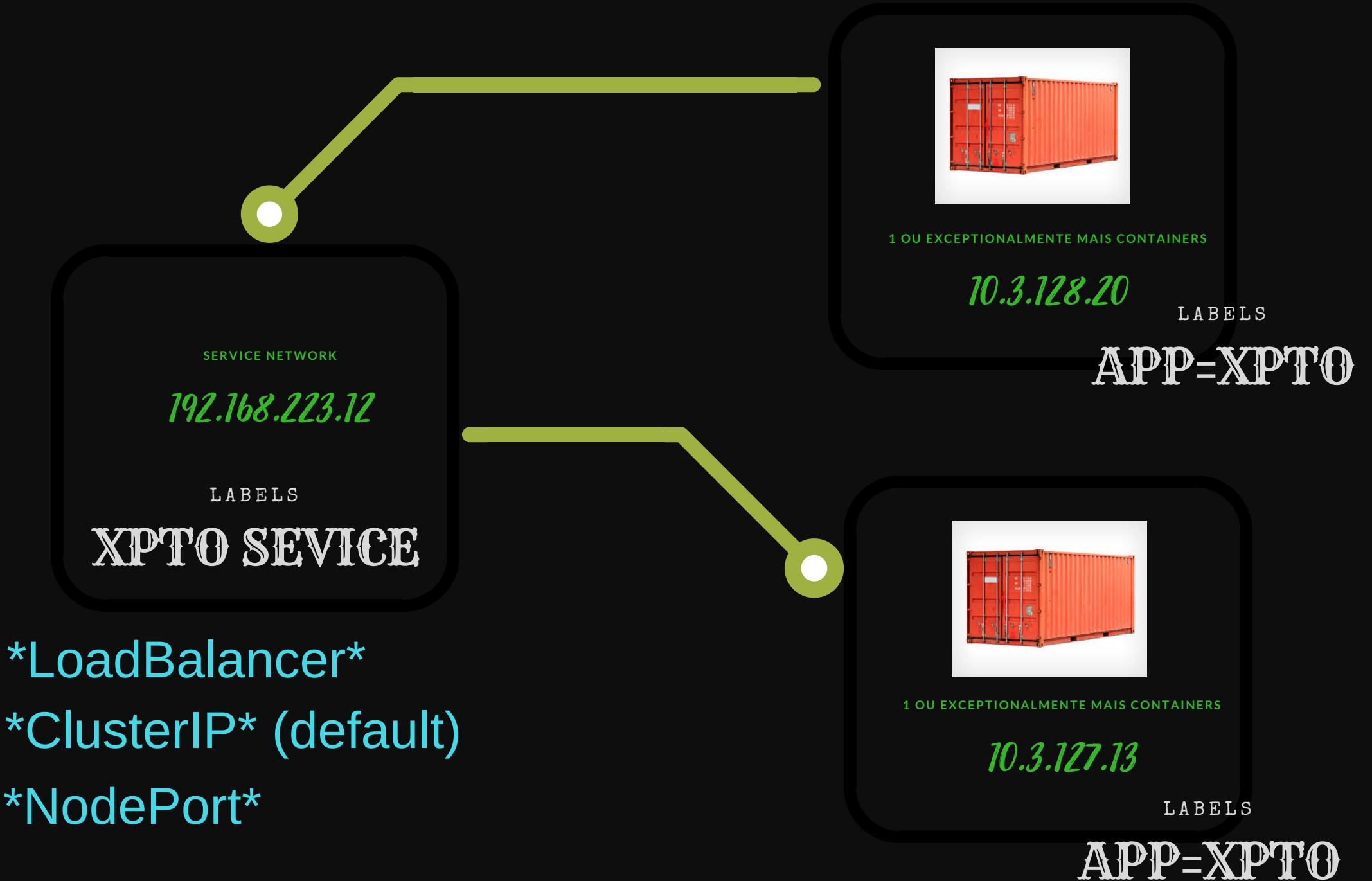
10.3.128.20

KEY = VALUE

LABELS

KIND: SERVICE

Camada de acesso



ReplicaSet = 2

KIND: DEPLOYMENT

YAML FILE



deployment.yaml

```
1  apiVersion: extensions/v1beta1
2  kind: Deployment
3  metadata:
4    name: hello-express
5  spec:
6    replicas: 1
7    template:
8      metadata:
9        labels:
10       application: hello-express
11    spec:
12      containers:
13      - name: hello-express
14        image: hello-express:latest
15        imagePullPolicy: IfNotPresent
16      resources:
17        requests:
18          cpu: 100m
19          memory: 100Mi
20      ports:
21      - containerPort: 8081
```

KIND: SERVICE

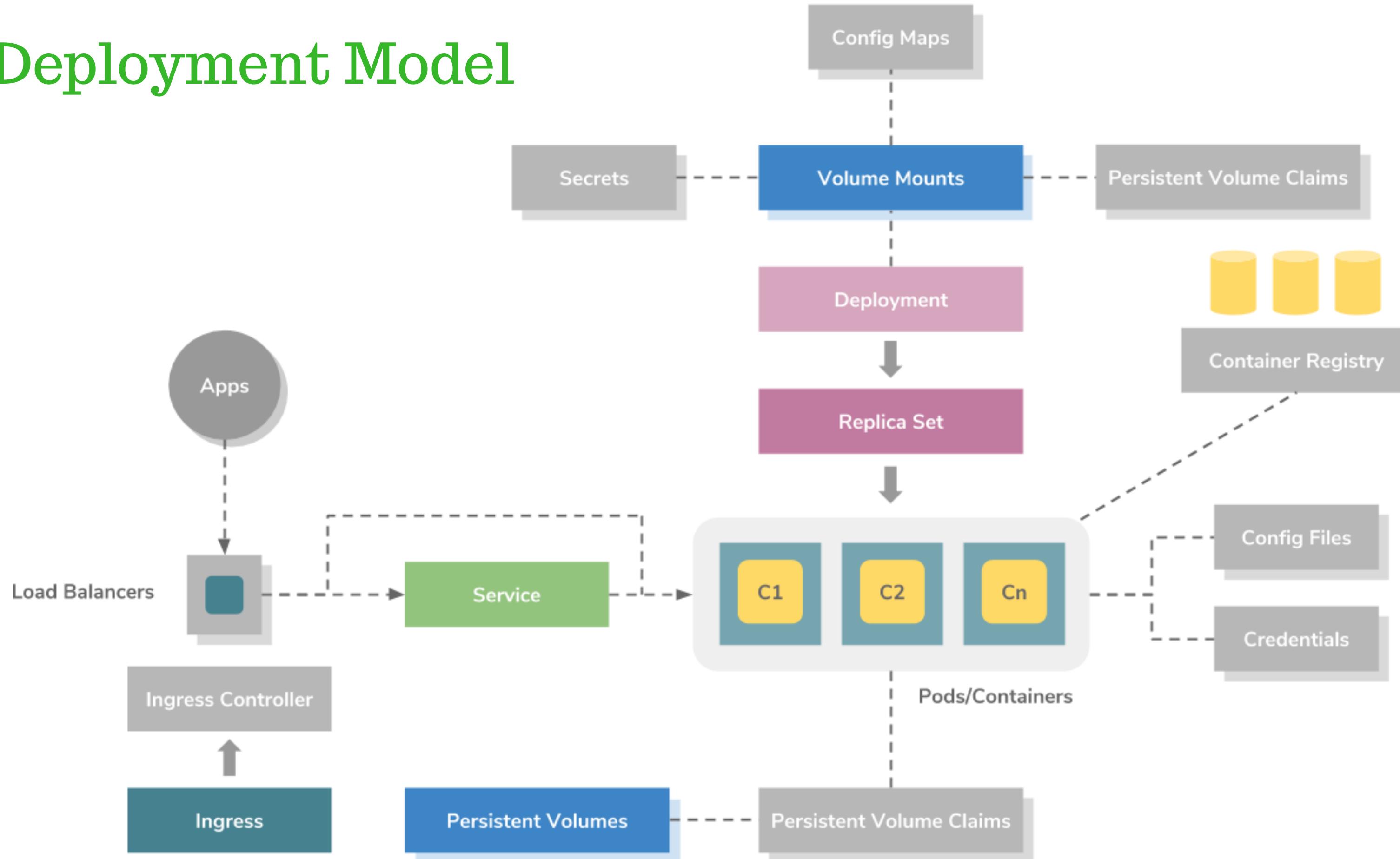
YAML FILE



service.yaml

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: hello-express
5   labels:
6     application: hello-express
7 spec:
8   ports:
9     - port: 80
10    protocol: TCP
11    targetPort: 8081
12   selector:
13     application: hello-express
14   type: LoadBalancer
```

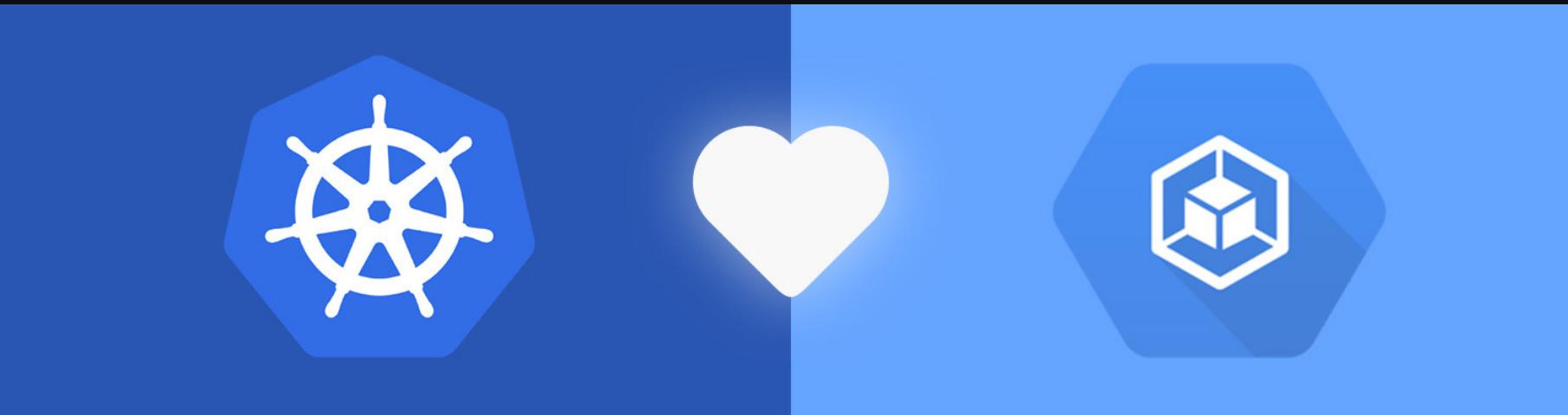
Deployment Model



GKE

Google Kubernetes Engine

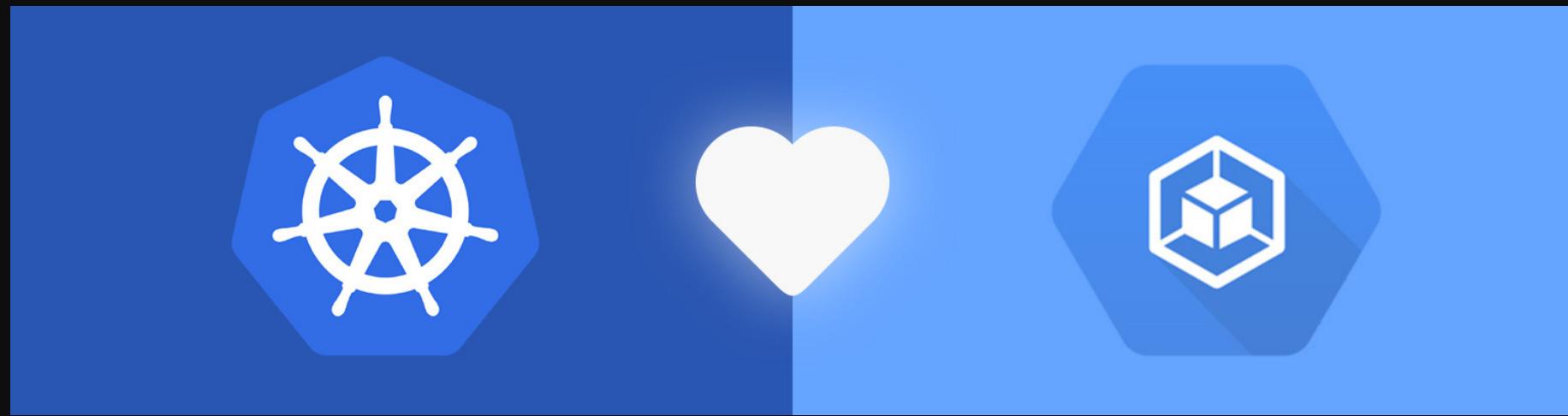




Executa containers **Docker** no Google Cloud Platform,
sobre o **Kubernetes (k8s)**.

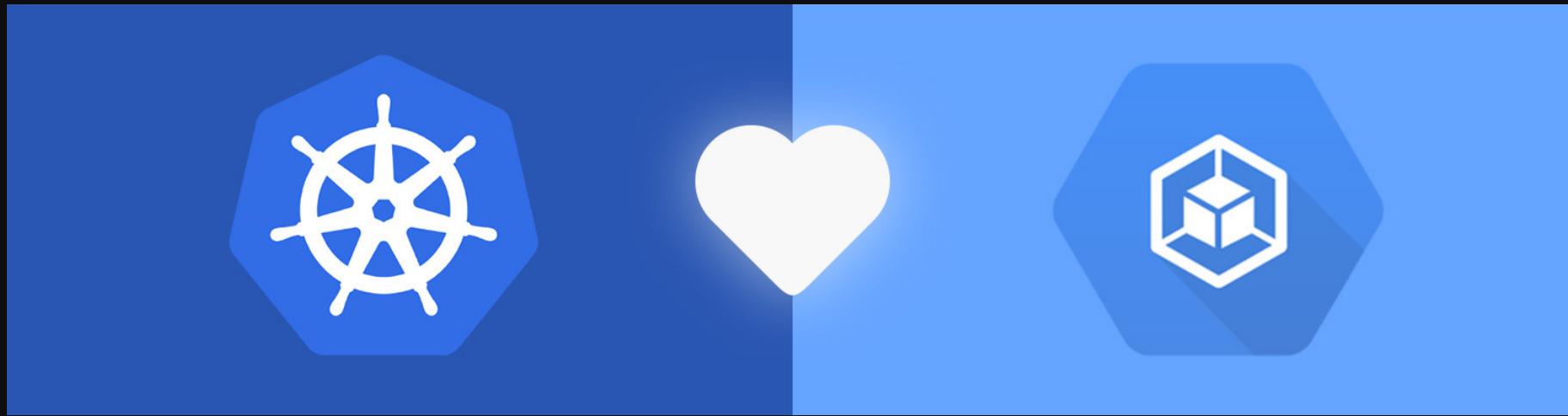
Container Engine (GKE) cuida do provisionamento e
manutenção da máquina virtual **container cluster**,
escalando sua aplicação e logística operacional como
logging, monitoring, and health management.

Setup Env



- 1) Download && install
<https://cloud.google.com/sdk/downloads>
- 2) gcloud components install kubectl
- 3) gcloud auth login
- 4) gcloud container clusters get-credentials cluster-1 --zone us-east1-b --project treinamentokubernetes

Namespace



`kubectl create namespace [my-name-space]`

`kubectl get namespace`



SOURCE <(KUBECTL COMPLETION BASH)

MEU PRIMEIRO DEPLOYMENT

- Microservices + Mongo
- YAML
- kubectl apply -f
- kubectl get pods
- kubectl get services

yaml



- kubectl apply -f
- kubectl get pods
- kubectl describe pod
- kubectl log

```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   name: mysql
5   labels:
6     app: mysql
7 spec:
8   replicas: 1
9   template:
10    metadata:
11      labels:
12        app: mysql
13   spec:
14     containers:
15       - name: mysql
16         image: mysql:5.6
17         env:
18           - name: MYSQL_ROOT_PASSWORD
19             value: minhasenha
20         ports:
21           - name: mysql-port
22             containerPort: 3306
23
24
25
```

yaml



kubernetes

- kubectl apply -f
- kubectl get services
- kubectl describe service

```
1 apiVersion: v1
2 kind: Service
3
4 metadata:
5   name: mysql-service
6   labels:
7     tier: service
8
9 spec:
10  type: LoadBalancer
11  ports:
12    - port: 3306
13      targetPort: mysql-port
14  selector:
15    app: mysql
16
17
```

YAML



kubernetes

- kubectl apply -f
- kubectl get services
- kubectl describe service

```
1 apiVersion: extensions/v1beta1
2 kind: Deployment
3 metadata:
4   name: wordpress
5   labels:
6     app: wordpress
7 spec:
8   replicas: 1
9   template:
10    metadata:
11      labels:
12        app: wordpress
13    spec:
14      containers:
15        - name: wordpress
16          image: wordpress:4.8-apache
17        env:
18          - name: WORDPRESS_DB_HOST
19            value: mysql-service.amarelo:3306
20        env:
21          - name: WORDPRESS_DB_PASSWORD
22            value: minhasenha
23        ports:
24          - name: http-port
25            containerPort: 80
26
27
```



- kubectl apply -f
- kubectl get services
- kubectl describe service

```
1 apiVersion: v1
2 kind: Service
3
4 metadata:
5   name: wordpress-service
6   labels:
7     tier: service
8
9 spec:
10  type: LoadBalancer
11  ports:
12    - port: 80
13      targetPort: http-port
14  selector:
15    app: wordpress
```

YOUR

SECRET

KUBECTL GET SECRETS

```
kubectl create secret generic mysql-pass --from-literal=password=YOUR_PASSWORD
```

valueFrom:

secretKeyRef:

- name:

key:

YOUR

DOCKER REGISTRY

KUBECTL GET SECRETS

```
kubectl create secret docker-registry regcred --docker-server=${S} --docker-username=${U} --docker-password=${P} --docker-email=${E}
```

spec:

imagePullSecrets:

- name: regcred

EXERCÍCIOS



CONTINUA...

:)