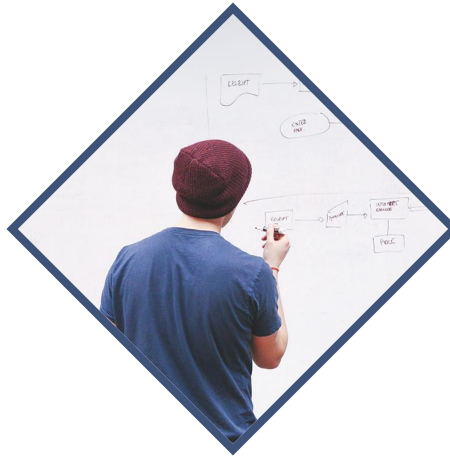


KUBERNETES (K8s)

An Overview



HELLO!

Rodrigo Carvalho
5by5

1

DOCKER - REVIEW



O QUE É DOCKER ?

O Docker é uma ferramenta projetada para facilitar a criação, implementação e execução de aplicativos usando containers. Com o Docker é possível lidar com os containers como se fossem máquinas virtuais modulares, e extremamente leves. Estas qualidades permitem você copiar / migrar um container de um ambiente para o outro de forma extremamente simples.

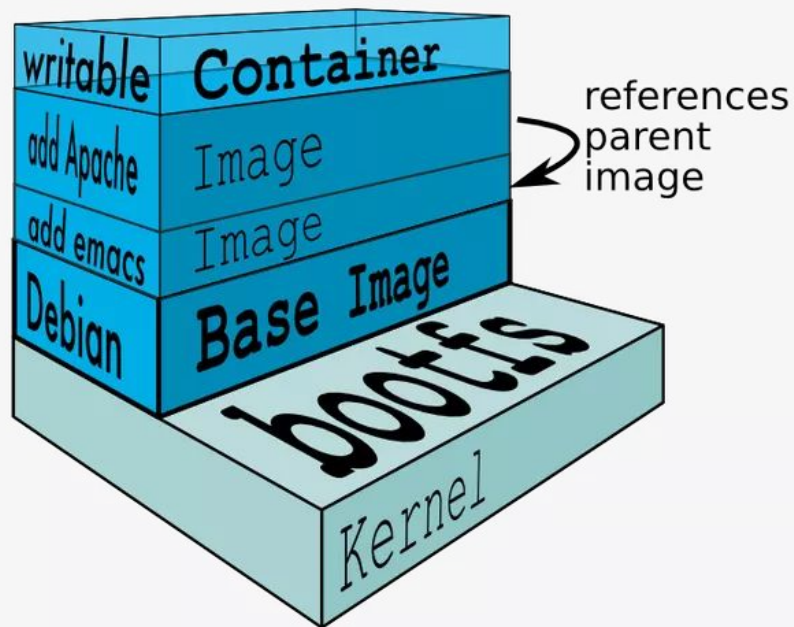


O QUE É UMA IMAGEM DOCKER ?

Uma **Imagem Docker** é um arquivo, composto de várias camadas, usado para executar código em um container do Docker. Uma imagem é essencialmente construída a partir das instruções para uma versão completa e executável de um aplicativo, que se baseia no kernel do sistema operacional host. Quando o usuário do Docker executa uma imagem, ela se torna uma ou várias instâncias desse container.



O QUE É UMA IMAGEM DOCKER ?





O QUE SÃO CONTAINERS ?

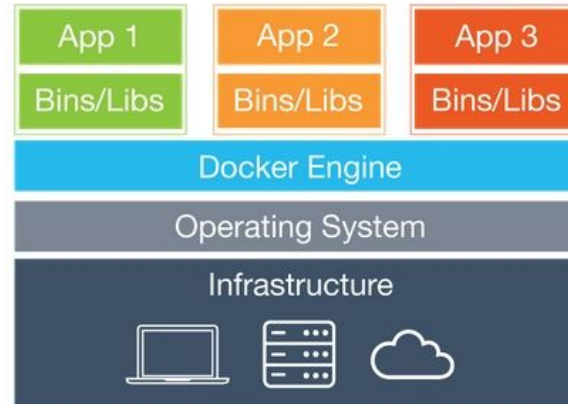
Um **Container** é um conjunto de um ou mais processos organizados isoladamente do sistema. Todos os arquivos necessários à execução de tais processos são fornecidos por uma imagem distinta. Na prática, os containers são portáteis e consistentes durante toda a migração entre os ambientes de desenvolvimento, teste e produção. Essas características os tornam uma opção muito mais rápida do que os pipelines de desenvolvimento, que dependem da replicação dos ambientes de teste tradicionais.



VM vs CONTAINERS



Virtual Machines



Containers



O QUE É UMA IMAGEM DOCKER ?

```
rodrigo@HomeDesktop001:DockeDemo$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
rccarvalho/dockerdemo v3                 dbfe9056bcee       31 minutes ago     265MB
<none>               <none>            6de7507f46c7       31 minutes ago     1.78GB
microsoft/dotnet     2.1.500-sdk        04868b49e01f       3 months ago       1.73GB
microsoft/dotnet     2.1.5-aspnetcore-runtime 1fe6774e5e9e       5 months ago       255MB

rodrigo@HomeDesktop001:DockeDemo$ sudo docker history 1fe6774e5e9e
IMAGE                CREATED             CREATED BY          SIZE
1fe6774e5e9e        5 months ago       /bin/sh -c curl -SL --output aspnetcore.tar... 149MB
<missing>           5 months ago       /bin/sh -c #(nop) ENV ASPNETCORE_VERSION=2... 0B
<missing>           5 months ago       /bin/sh -c apt-get update && apt-get ins... 7.02MB
<missing>           5 months ago       /bin/sh -c #(nop) ENV ASPNETCORE_URLS=http:... 0B
<missing>           5 months ago       /bin/sh -c apt-get update && apt-get ins... 43.7MB
<missing>           5 months ago       /bin/sh -c #(nop) CMD ["bash"]              0B
<missing>           5 months ago       /bin/sh -c #(nop) ADD file:f8f26d117bc4a9289... 55.3MB

rodrigo@HomeDesktop001:DockeDemo$ sudo docker history dbfe9056bcee
IMAGE                CREATED             CREATED BY          SIZE
dbfe9056bcee        31 minutes ago     /bin/sh -c #(nop) EXPOSE 80                  0B
ace7b6b4f8a5        31 minutes ago     /bin/sh -c #(nop) ENTRYPOINT ["dotnet"] "Doc... 0B
92d1a5714f98        31 minutes ago     /bin/sh -c #(nop) COPY dir:f0b0796960cb3523b... 10.8MB
7ebaeb9dd2f9        31 minutes ago     /bin/sh -c #(nop) WORKDIR /app                0B
1fe6774e5e9e        5 months ago       /bin/sh -c curl -SL --output aspnetcore.tar... 149MB
<missing>           5 months ago       /bin/sh -c #(nop) ENV ASPNETCORE_VERSION=2... 0B
<missing>           5 months ago       /bin/sh -c apt-get update && apt-get ins... 7.02MB
<missing>           5 months ago       /bin/sh -c #(nop) ENV ASPNETCORE_URLS=http:... 0B
<missing>           5 months ago       /bin/sh -c apt-get update && apt-get ins... 43.7MB
<missing>           5 months ago       /bin/sh -c #(nop) CMD ["bash"]              0B
<missing>           5 months ago       /bin/sh -c #(nop) ADD file:f8f26d117bc4a9289... 55.3MB

rodrigo@HomeDesktop001:DockeDemo$
```

A person with short dark hair, seen from the back, is looking at a wall covered in various design sketches, photos, and notes. The wall is a collage of creative work, including wireframes, photographs of people and objects, and handwritten text. The person is wearing a grey and black striped sweater. The overall scene suggests a creative or design workspace.

COMO GERENCIAR SEUS CONTAINERS ?

2

Kubernetes - Overview



O QUE É KUBERNETES ?

O **Kubernetes** (K8s) é um sistema de orquestração de containers de código aberto para automatizar a implantação, o dimensionamento e o gerenciamento de containers . Kubernetes foi anunciado em 2014, desenvolvido baseado no BORG, que foi o orquestrador do Google por 15 anos, em 2015 foi doado a Cloud Native Computing Foundation (CNCF).



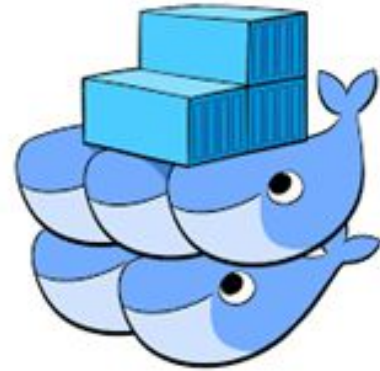
ORQUESTRADORES



KUBERNETES



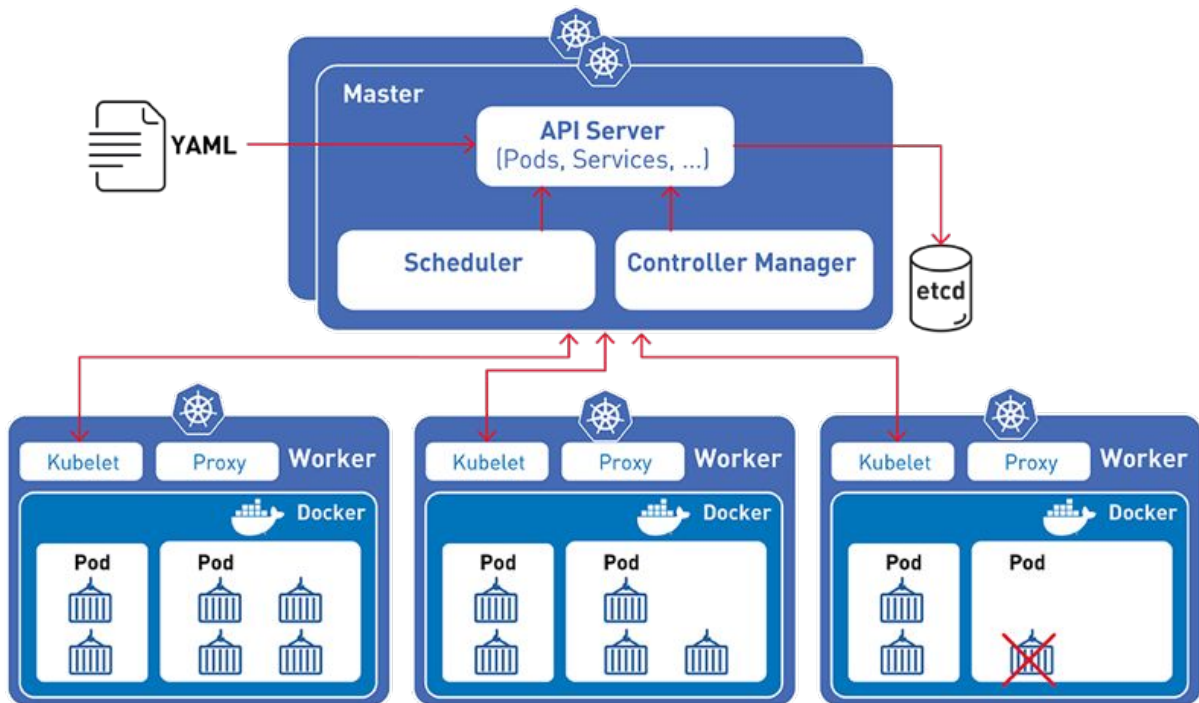
APACHE MESOS



DOCKER SWARM



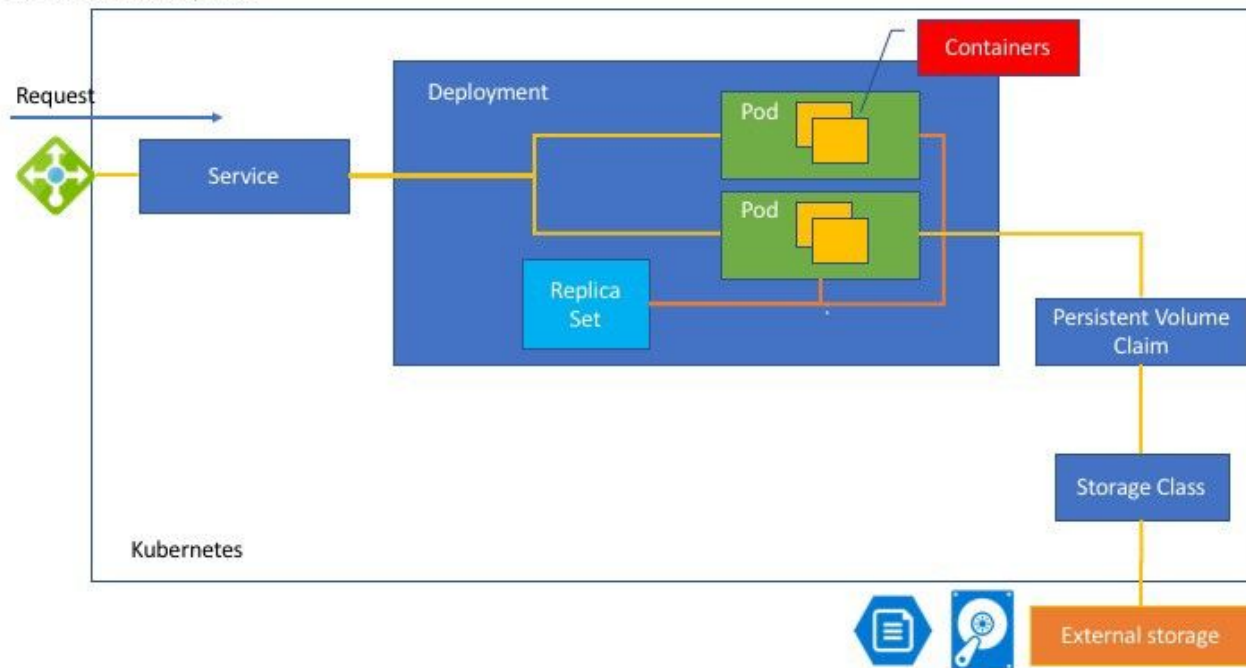
ARQUITETURA DO KUBERNETES





PRINCIPAIS OBJETOS DO KUBERNETES

Kubernetes Objects





K8s DEPLOY YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-dev-redis-dep
spec:
  selector:
    matchLabels:
      app: demo-dev
      tier: persistence
      version: v1
  replicas: 1
  template:
    metadata:
      labels:
        app: demo-dev
        tier: persistence
        version: v1
    spec:
      containers:
        - name: redis
          image: redis:alpine
          resources:
            requests:
              cpu: 25m
              memory: 100Mi
          ports:
            - containerPort: 6379
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-dev-app-dep
  labels:
    app: demo-dev
    tier: app
    version: v1
spec:
  selector:
    matchLabels:
      app: demo-dev
      tier: app
      version: v1
  replicas: 4
  template:
    metadata:
      labels:
        app: demo-dev
        tier: app
        version: v1
    spec:
      containers:
        - name: demoapp
          image: rccarvalho/dockerdemo:v1
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
          livenessProbe:
            httpGet:
              path: /swagger/index.html
              port: 80
              initialDelaySeconds: 15
              periodSeconds: 60
          readinessProbe:
            httpGet:
              path: /swagger/index.html
              port: 80
              initialDelaySeconds: 5
              periodSeconds: 5
              successThreshold: 1
          resources:
            requests:
              cpu: 25m
              memory: 200Mi
          imagePullPolicy: Always
          restartPolicy: Always
      strategy:
        type: RollingUpdate
        rollingUpdate:
          maxUnavailable: 1
          maxSurge: 1
```




K8s SERVICE YAML

```
apiVersion: v1
kind: Service
metadata:
  name: demo-dev-redis-svc
  labels:
    app: demo-dev
    tier: persistence
    version: v1
spec:
  ports:
    - port: 6379
      targetPort: 6379
  selector:
    app: demo-dev
    tier: persistence
    version: v1
```

```
apiVersion: v1
kind: Service
metadata:
  name: demo-dev-app-svc
  labels:
    app: demo-dev
    tier: app
    version: v1
spec:
  #Tipos ClusterIP / NodePort / LoadBalancer
  type: NodePort
  ports:
    - port: 80
  selector:
    app: demo-dev
    tier: app
    version: v1
```



RODANDO KUBERNETES LOCALMENTE

- MiniKube

<https://kubernetes.io/docs/tasks/tools/install-minikube>

- Docker

<https://blog.docker.com/2018/01/docker-windows-desktop-now-kubernetes>

- Play With Kubernetes

<https://labs.play-with-k8s.com>



KUBECTL COMMANDS

Pod & Container Introspection

```
# List the current pods
kubectl get pods
# Describe pod <name>
kubectl describe pod <name>
# List the replication controllers
kubectl get rc
# List the replication controllers in <namespace>
kubectl get rc --namespace=<namespace>
# Describe replication controller <name>
kubectl describe rc <name>
# List the services
kubectl get svc
# Describe service <name>
kubectl describe svc <name>
# Delete pod <name>
kubectl delete pod <name>
# Watch nodes continuously
kubectl get nodes -w
```

Cluster Introspection

```
# Get version information
kubectl version
# Get cluster information
kubectl cluster-info
# Get the configuration
kubectl config view
# Output information about a node
kubectl describe node <node>
```

Debugging

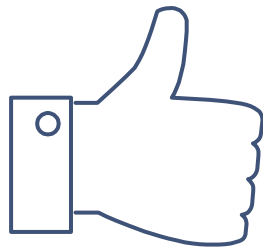
```
# Execute <command> on <service> optionally #
selecting container <$container>
kubectl exec <service> <command> [-c <$container>]
# Get logs from service <name> optionally # selecting
container <$container>
kubectl logs -f <name> [-c <$container>]
# Watch the Kubelet logs
watch -n 2 cat /var/log/kublet.log
# Show metrics for nodes
kubectl top node
# Show metrics for pods
kubectl top pod
```

Quick Commands

```
# Launch a pod called <name>
# using image <image-name>
kubectl run <name> --image=<image-name>
# Create a service described # in <manifest.yaml>
kubectl create -f <manifest.yaml>
# Scale replication controller
# <name> to <count> instances
kubectl scale --replicas=<count> rc <name>
# Map port <external> to # port <internal> on replication
# controller <name>
kubectl expose rc <name> --port=<external> --target-
port=<internal>
# Stop all pods on <n>
kubectl drain <n> --delete-local-data --force --ignore-
daemonsets
# Create namespace <name>
kubectl create namespace <namespace>
# Allow Kubernetes master nodes to run pods
kubectl taint nodes --all node-role.kubernetes.io/master-
```

Objects

```
all
clusterrolebindings
clusterroles
cm = configmaps
controllerrevisions
crd = customresourcedefinition
cronjobs
cs = componentstatuses
csr = certificatesigningrequests
deploy = deployments
ds = daemonsets
ep = endpoints
ev = events
hpa = horizontalpodautoscalers
ing = ingresses
jobs
limits = limitranges
netpol = networkpolicies
no = nodes
ns = namespaces
pdb = poddisruptionbudgets
po = pods
podpreset
podtemplates
psp = podsecuritypolicies
pv = persistentvolumes
pvc = persistentvolumeclaims
quota = resourcequotas
rc = replicationcontrollers
rolebindings
roles
rs = replicasets
sa = serviceaccounts
sc = storageclasses
secrets
sts = statefulsets
```



OBIGADO !

Perguntas?

Rodrigo: rodrigo.carvalho@5by5.com.br



FONTES:

<http://kubernetes.io>

<http://www.docker.com>

<http://www.redhat.com.br>

<http://caylent.com/kubectl-commands-cheat-sheet>