UNIVERSIDADE
FEDERAL DO PIAUÍ

Sistemas Distribuídos

Containers

5º Semana - Aula 09

Sistemas de Informação

prof. Rayner Gomes - rayner@ufpi.edu.br/raynergomes@gmail.com

Aviso: As videoaulas gravadas e disponibilizada aos alunos da UFPI são estritamente reservados aos alunos da UFPI, sendo proibido qualquer divulgação e distribuição. A reprodução só é permitida aos alunos matriculados na disciplina.

Tópicos

- Objetivo
- Definição
- Docker
- Comandos básicos

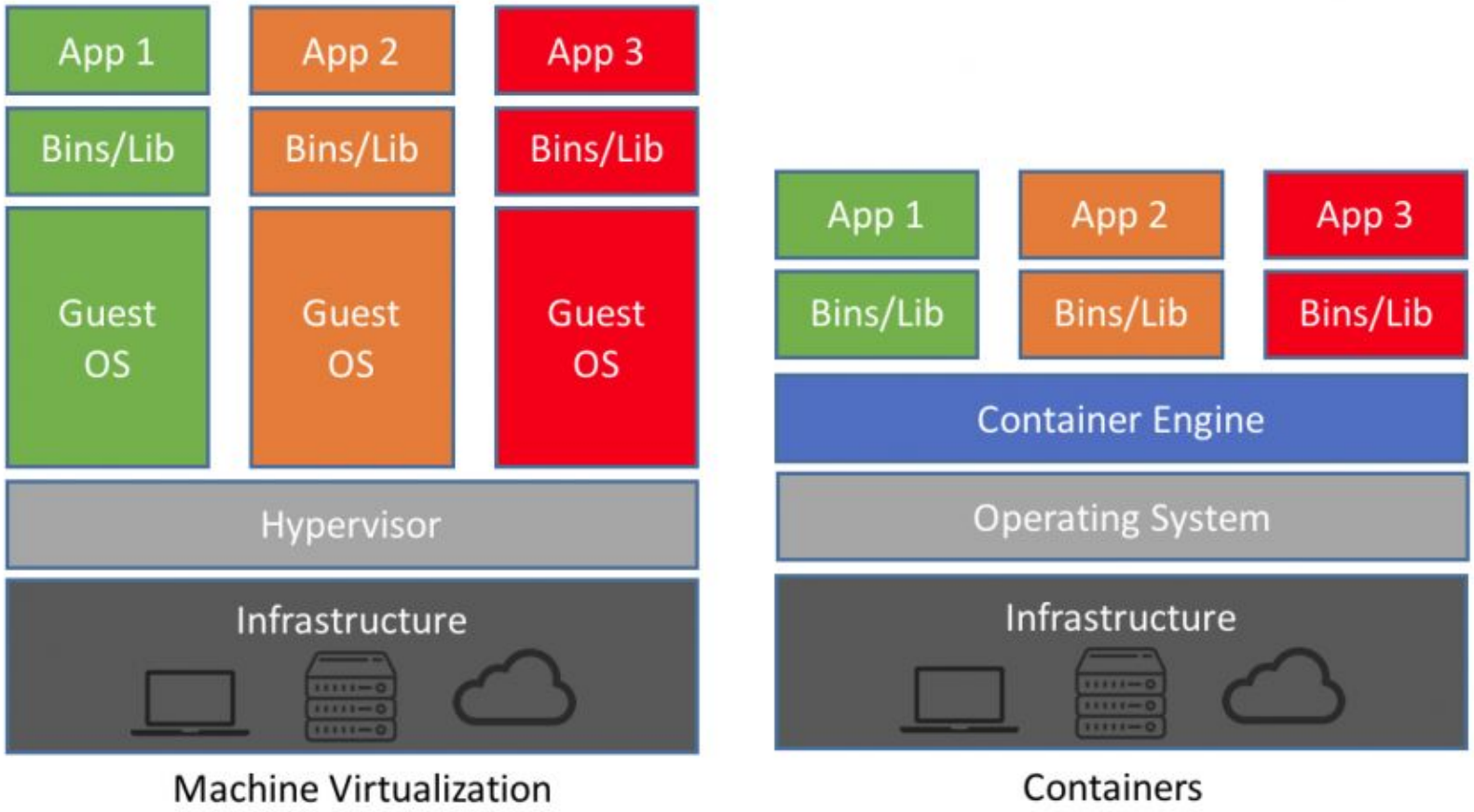
Objetivos

- Vamos usar Docker para simular componentes distribuídos.
- A vantagem do Docker é a que diminuição de sobrecarga para simular hosts.
- Outra vantagem é facilidade do gerenciamento dos containers quando comparado às VM.

Definição

É o agrupamento de uma aplicação junto com suas dependências, compartilha o kernel do sistema operacional do *host* onde está rodando.

VM X Container



https://blog.netapp.com/blogs/containers-vs-vms

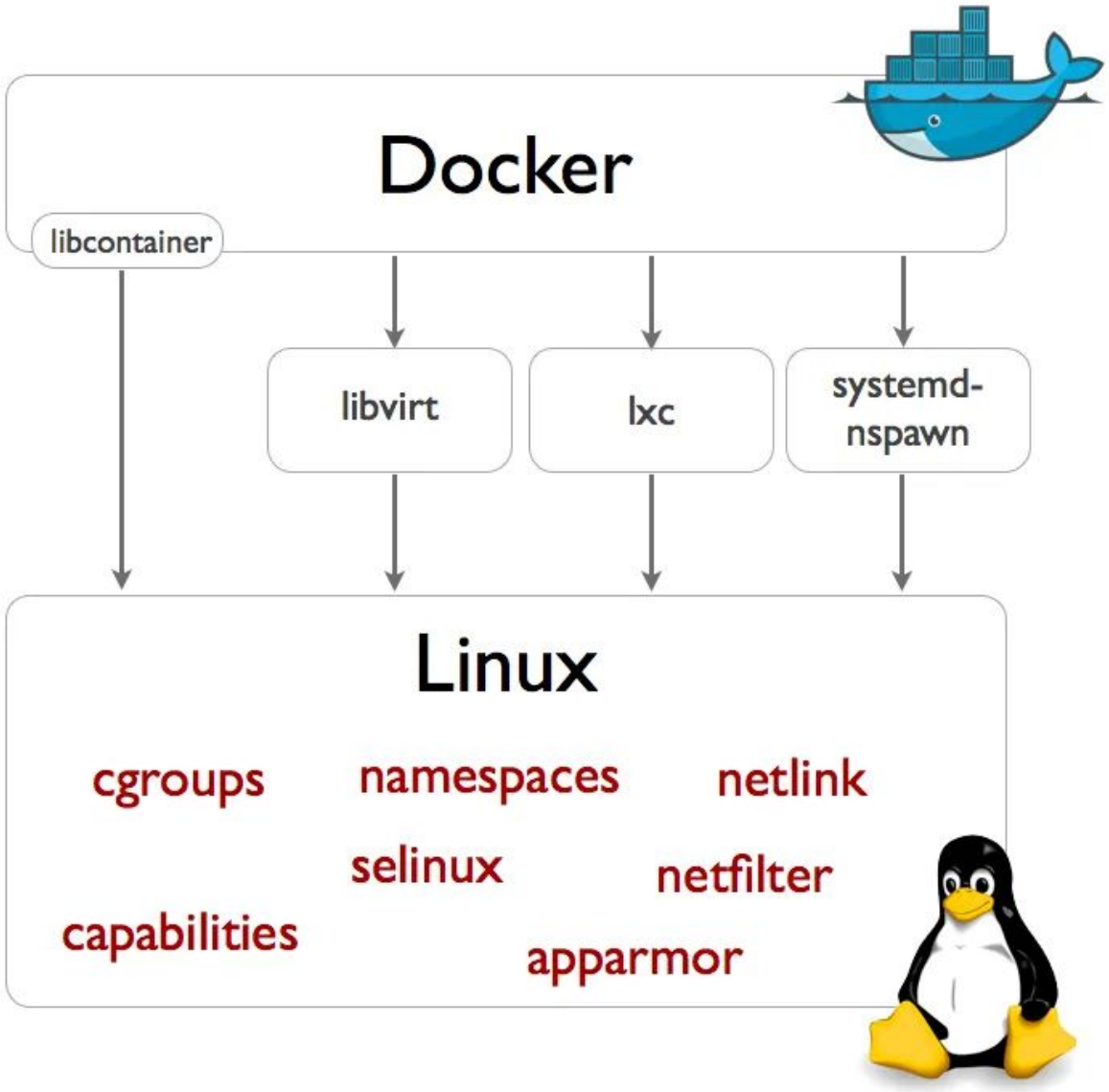
Containers Vantagens

- Portabilidade
- Velocidade

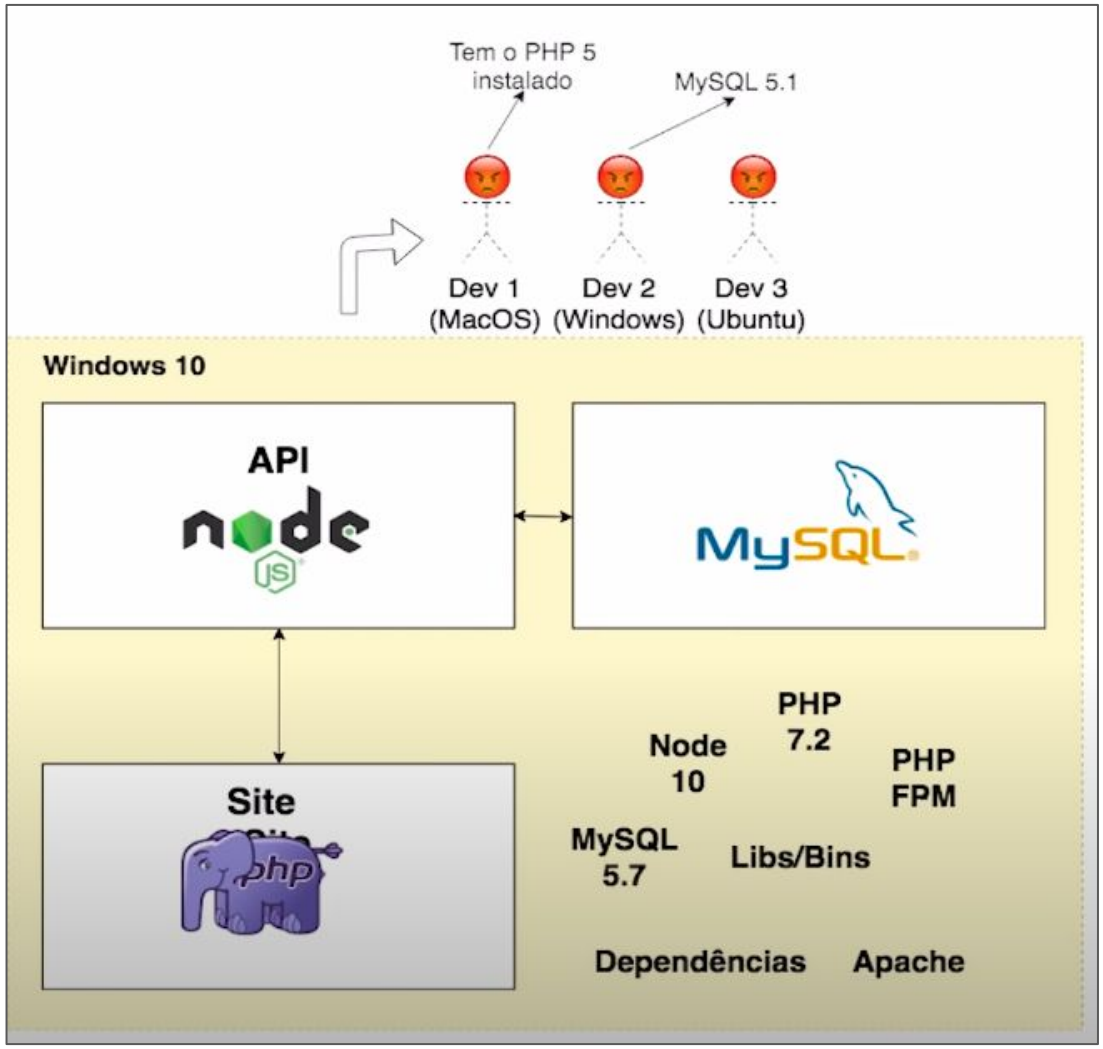
[Containers emula uma aplicação e VM emula um SO]

Breve histórico

- apesar de ter popularizado recentemente a ideia é antiga.
- *chroot*: isolar o sistema.
- *jails*: isolamento do filesystem e dos processos
- *OpenVZ*: ambiente de gerenciamento de containers usados nas VPS (*Virtual Private Server*).
- *Bridge* e *Open VSwitch*: virtualização de switches.
- *namespaces*: isolamento de ambientes de sistema.
- 2013 popularização do Docker.

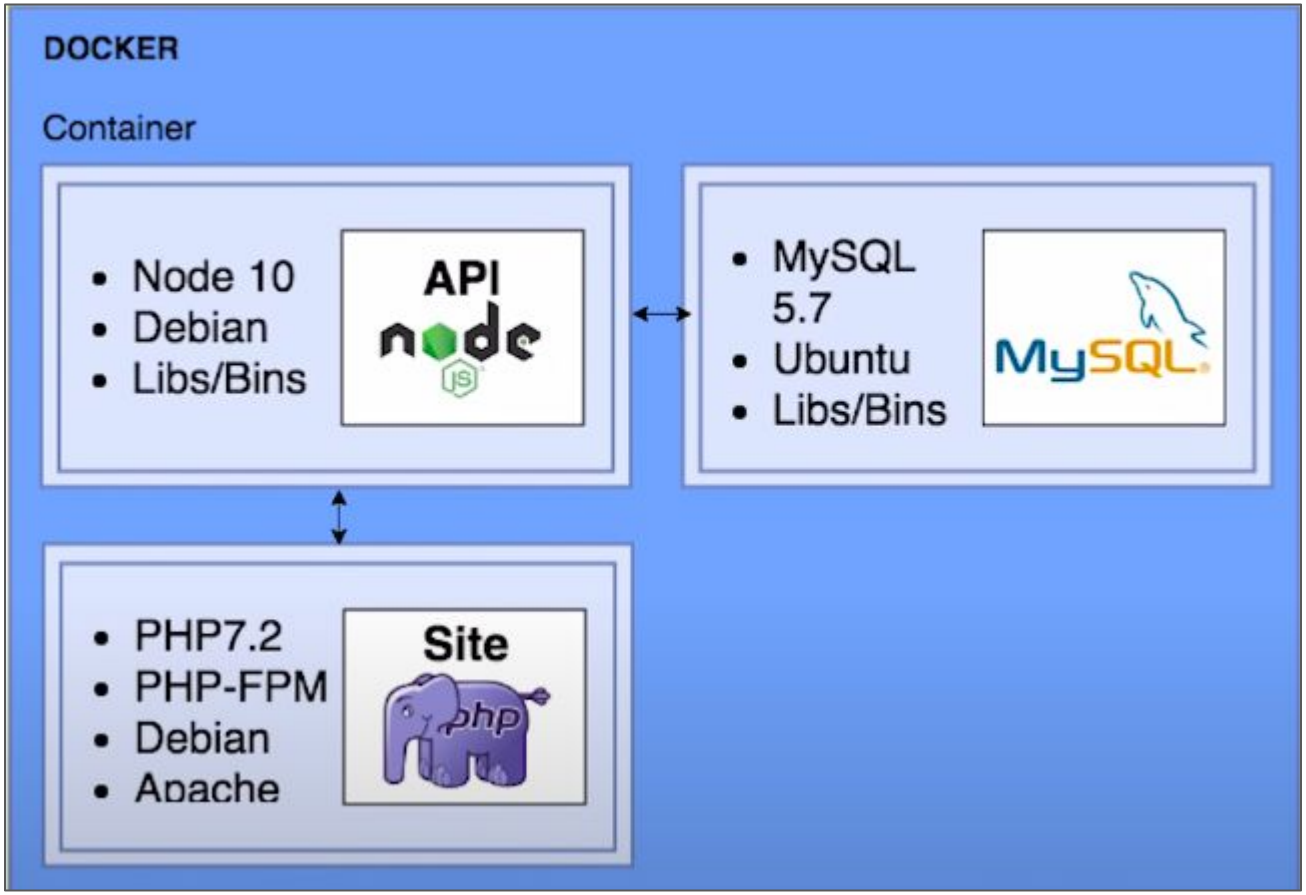

Docker
libcontainer
libvirt
lxc
systemd-
nspawn
Linux
cgroups
namespaces
netlink
selinux
netfilter
capabilities
apparmor

Motivação: “mas na minha máquina funciona!”



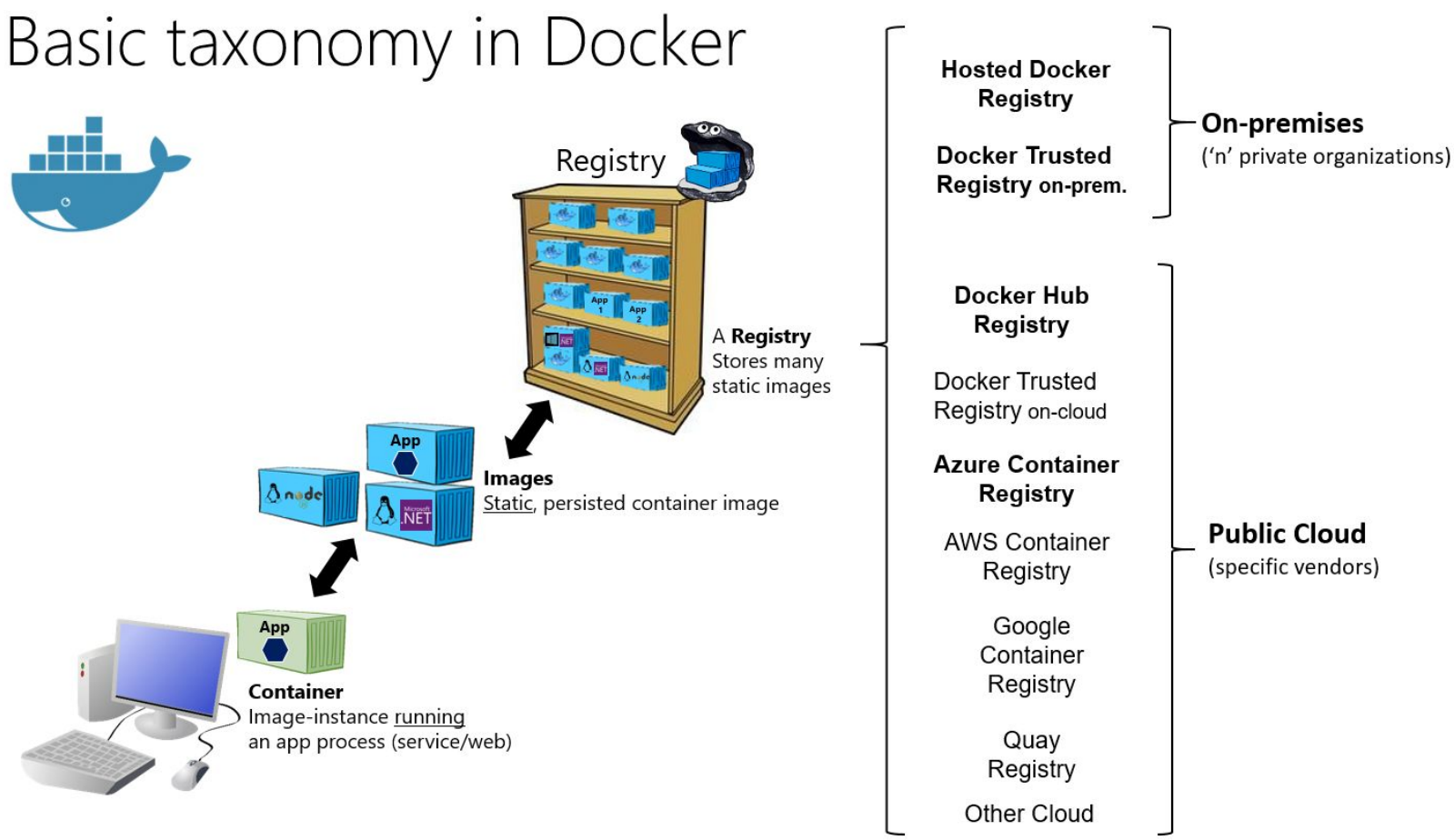
Source: shorturl.at/eqvS3

Motivação: “tudo encapsulado”



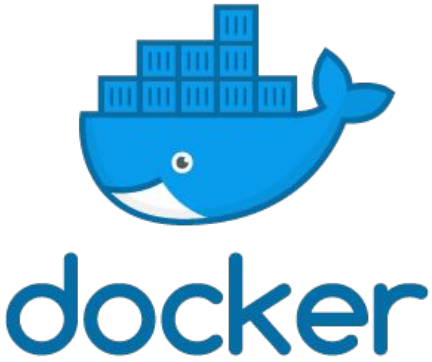
Source: shorturl.at/eqvS3

Nomenclatura Docker



Source: shorturl.at/enxz1

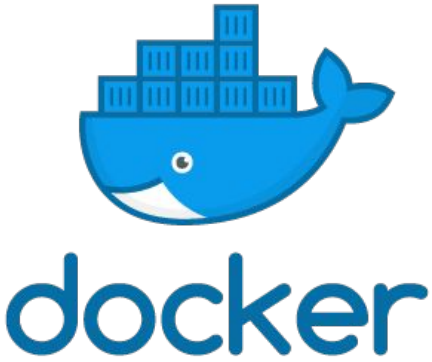
Docker Instalação



Requisitos

- Docker não suporta processadores 32 bits.
- Docker é suportado somente (*stable*) na versão do kernel 3.8 ou superior.
- O kernel deve suportar sistemas de arquivos utilizados pelo Docker, como AUFS, Device Mapper, OverlayFS, etc.
- O kernel deverá ter suporte a cgroups e namespaces.

Docker Instalação



Instalação Linux:

https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-using-the-repository

SET UP THE REPOSITORY

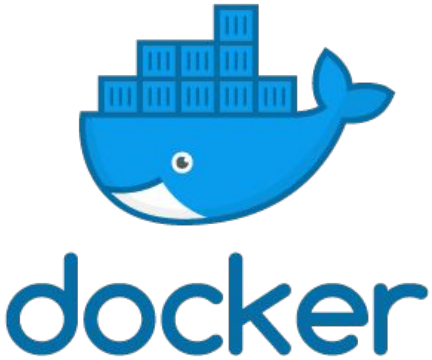
1. Update the `apt` package index:

```
$ sudo apt-get update
```

2. Install packages to allow `apt` to use a repository over HTTPS:

```
$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

Docker Instalação



sudo apt-get install curl

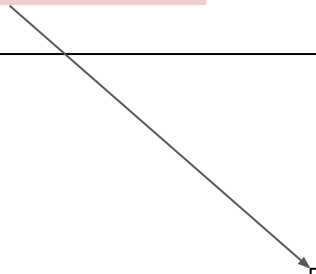
sudo apt install docker.io docker-compose

sudo systemctl enable --now docker docker.socket containerd

Dicas

rayner@Y720:~$ **docker images ls**

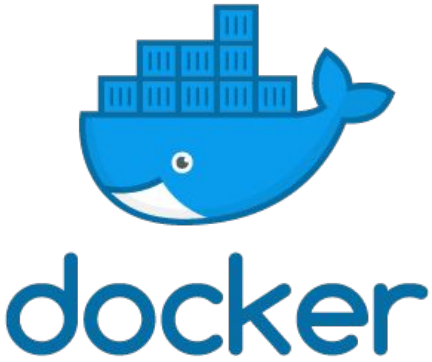
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/json?filters=%7B%22reference%22%3A%7B%22ls%22%3Atrue%7D%7D: dial unix /var/run/docker.sock: connect: permission denied



/etc/group

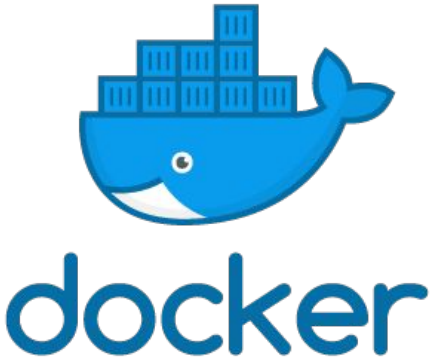
docker:x:998:rayner

Docker: O comando mais importante



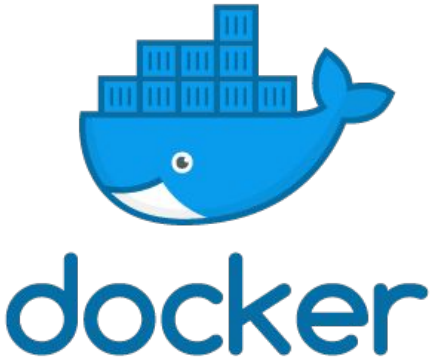
docker help

Gerenciamento Básico



- service docker start
- service docker status
- service docker stop

Docker: Comandos básicos

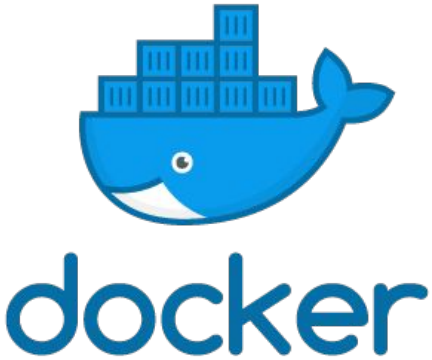


Iniciar um container:

```
root@Lenovo:/home/rayner# docker run alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
bdf0201b3a05: Pull complete
Digest: sha256:28ef97b8686a0b5399129e9b763d5b7e5ff03576aa5580d6f4182a49c5fe1913
Status: Downloaded newer image for alpine:latest
root@Lenovo:/home/rayner#
```

Segunda vez, não faz o download do repositório!

Docker: Comandos básicos



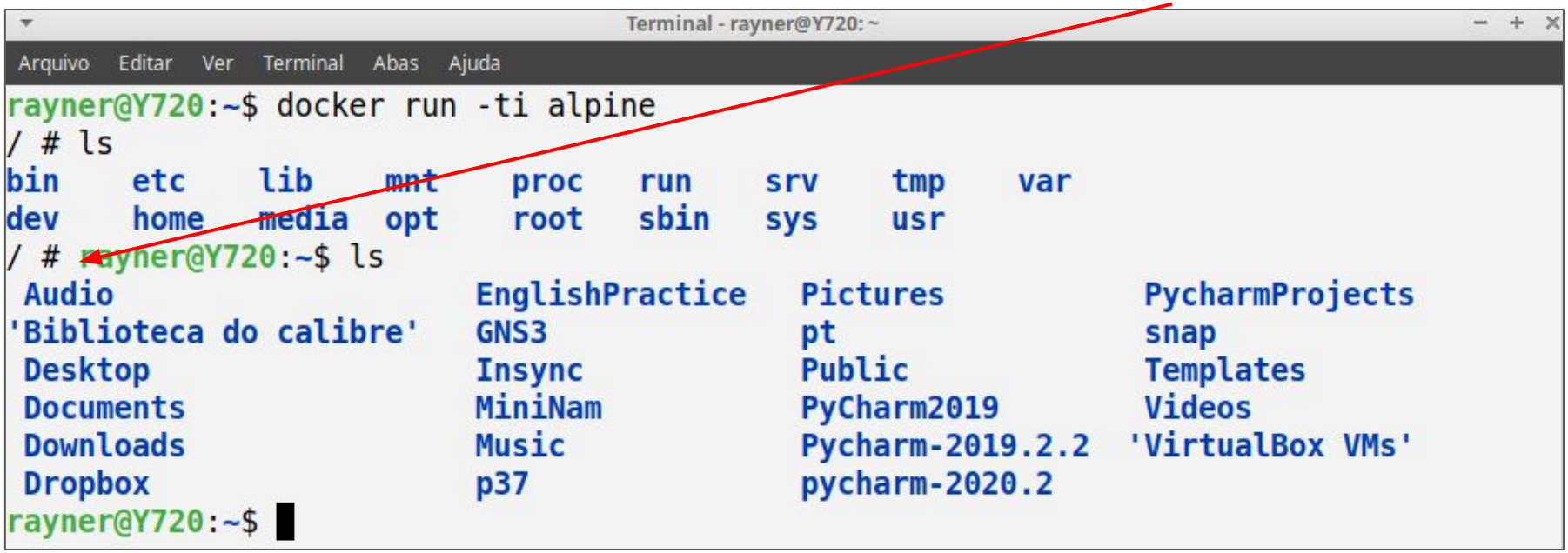
Iniciar um container:

```
root@Y720:~# docker run -ti alpine
/ #
```

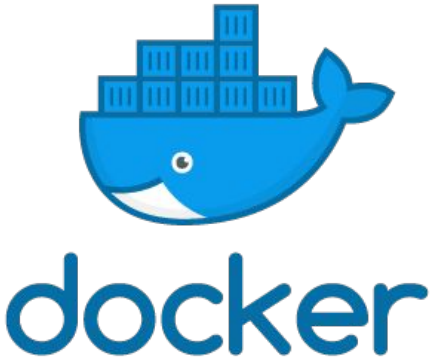
Docker: Comandos básicos

Sai sem encerrar o container:

[GO BACK] Ctrl P + Ctrl Q



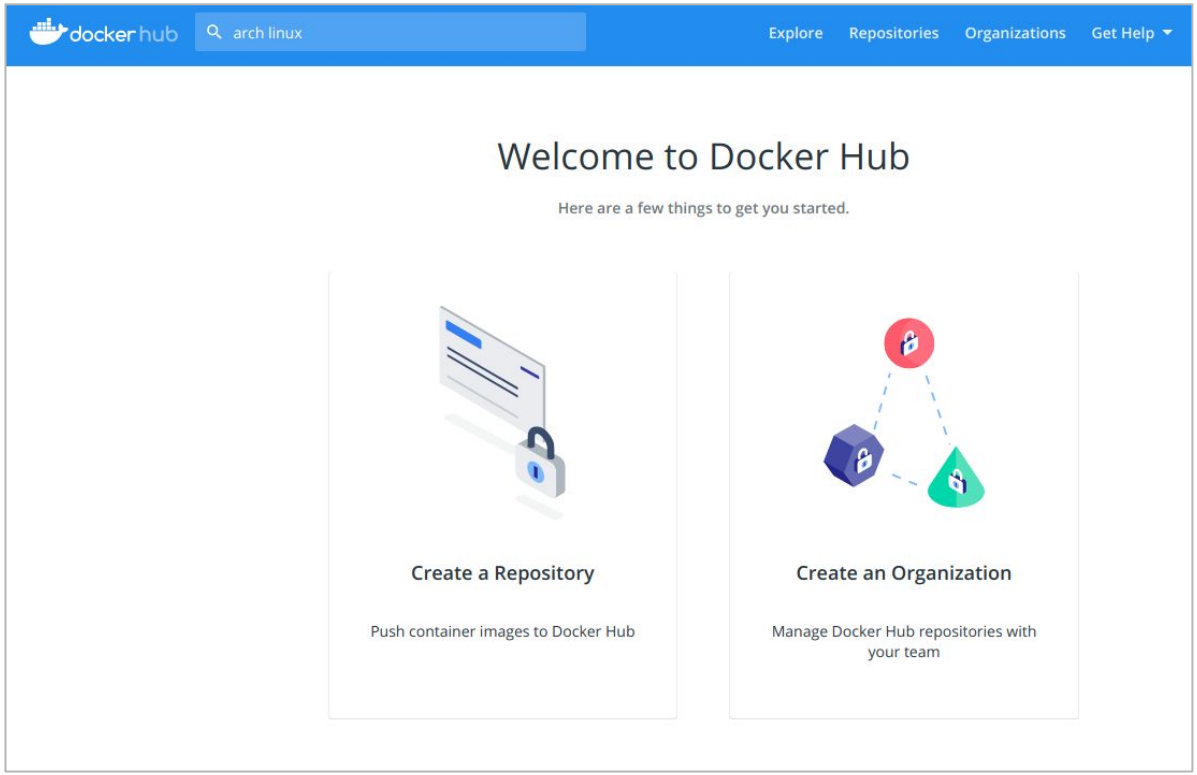
Docker: Comandos básicos



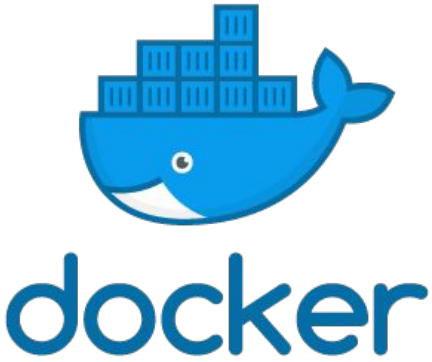
Sai sem encerrar o container:

```
root@Y720:~# docker run -ti alpine
/ # ^C
/ # root@Y720:~# docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
5b0ee89fae20        alpine              "/bin/sh"           3 minutes ago       Up 3 minutes
relaxed_austin
root@Y720:~#
```

Procurar por Repositórios

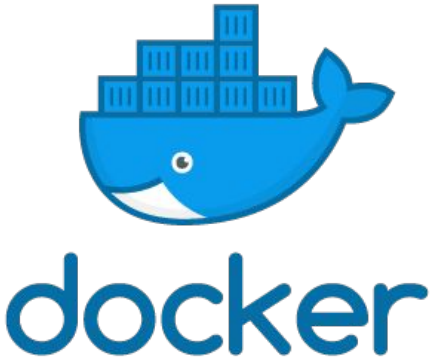


Docker: Comandos Básicos



- docker ps [ou docker container ps]// lista os containers ativos
- docker ps -a [ou docker container ps -a] //lista todos containers
- docker images // lista as imagens
- docker stop CONT_ID
- docker start CONT_ID
- docker attach CONT_ID
- docker rm CONT_ID // remove do disco
- docker inspect IMG_ID // obtém todos as informações do container

Container ID



docker ps mostra os ativos!

```
root@Y720:~# docker ps
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
5b0ee89fae20   alpine   "/bin/sh"   10 minutes ago   Up 10 minutes   name_abx
root@Y720:~#
```

Docker Attach

root@Y720:~# **docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

5b0ee89fae20 alpine "/bin/sh" 10 minutes ago Up 10 minutes relaxed_austin

root@Y720:~# **docker attach 5b0ee89fae20**

/ # ifconfig

eth0 Link encap:Ethernet HWaddr 02:42:AC:11:00:02

inet addr:172.17.0.2 Bcast:172.17.255.255 Mask:255.255.0.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:149 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:27654 (27.0 KiB) TX bytes:0 (0.0 B)

Docker Exemplos

```
root@Lenovo:/home/rayner# docker run -ti nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
27833a3ba0a5: Pull complete
eb51733b5bc0: Pull complete
994d4a01fbe9: Pull complete
Digest: sha256:50174b19828157e94f8273e3991026dc7854ec7dd2bbb33e7d3bd91f0a4b333d
Status: Downloaded newer image for nginx:latest
```

[Container]

[outro host]

Onde os Arquivos do Docker são Armazenados

root@Y720:/var/lib# **ls docker/**

builder buildkit containers image network overlay2 plugins runtimes swarm tmp trust volumes

root@Y720:/var/lib# **du docker/ -h**

225M docker

Onde os Arquivos do Docker são Armazenados

root@Y720:/var/lib# **ls docker/**

builder buildkit containers image network overlay2 plugins runtimes swarm tmp trust volumes

root@Y720:/var/lib# **du docker/ -h**

225M docker

Onde os Arquivos do Docker são Armazenados

root@Y720:/var/lib# **ls docker/**

builder buildkit containers image network overlay2 plugins runtimes swarm tmp trust volumes

root@Y720:/var/lib# **du docker/ -h**

225M docker

Remoção de Imagens

```
rayner@Y720:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              f643c72bc252        7 days ago          72.9MB
nginx               latest              bc9a0695f571        8 days ago          133MB
alpine              latest              d6e46aa2470d        6 weeks ago         5.57MB
hello-world         latest              bf756fb1ae65        11 months ago       13.3kB
```

```
rayner@Y720:~$ docker image rm bf756fb1ae65
```

um Servidor HTTP Python

```
root@83606ef861f9:/# python3 -m http. server 8000

/usr/bin/python3: No module named http.

root@83606ef861f9:/# python3 -m http.server 8000

Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

[em outra máquina]

Saindo e Voltando ao Container

[vamos sair]

rayner@Y720:~$ docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

rayner@Y720:~$ docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

83606ef861f9 ubuntu "bash" 13 minutes ago Exited (0) 44 seconds ago elegant_joliot

rayner@Y720:~$

Saindo e Voltando ao Container

```
rayner@Y720:~$ docker start 83606ef861f9

83606ef861f9

rayner@Y720:~$ docker attach  83606ef861f9

root@83606ef861f9:/#
```

Salvando o Container como um Imagem

```
rayner@Y720:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
83606ef861f9        ubuntu              "bash"              16 minutes ago      Exited (0) 4 seconds ago
rayner@Y720:~$ docker commit  83606ef861f9  ubuntu:python.1
sha256:41bfa3e23e1fe3f45c6101281b9aa8e3ffb6c6d757df580805eae0ddb84cfa46
rayner@Y720:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              python.1            41bfa3e23e1f        6 seconds ago       139MB
rayner@Y720:~$
```

Testando a Nova Imagem

```
rayner@Y720:~$ docker run -ti ubuntu:python.1 bash

root@9b8fd6848d40:/# python3

Python 3.8.5 (default, Jul 28 2020, 12:59:40)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>
```

Inspecionado os Recursos Usados pelo Docker

rayner@Y720:~$ **docker inspect 9b8fd6848d40 | grep -i mem**

```
"Memory": 0,
"CpusetMems": "",
"KernelMemory": 0,
"KernelMemoryTCP": 0,
"MemoryReservation": 0,
"MemorySwap": 0,
"MemorySwappiness": null,
```

Os valores correspondente à memória estão zerados, ou seja, sem nenhum limite estabelecido!

Atribuindo a Qtd de Mem para um Container

rayner@Y720:~$ **docker run -m 512M --name novo_container -ti ubuntu:python.1 bash**

WARNING: Your kernel does not support swap limit capabilities or the cgroup is not mounted. Memory limited without swap.

root@361093ec72c2:/#

rayner@Y720:~$ **docker inspect novo_container | grep -i mem**

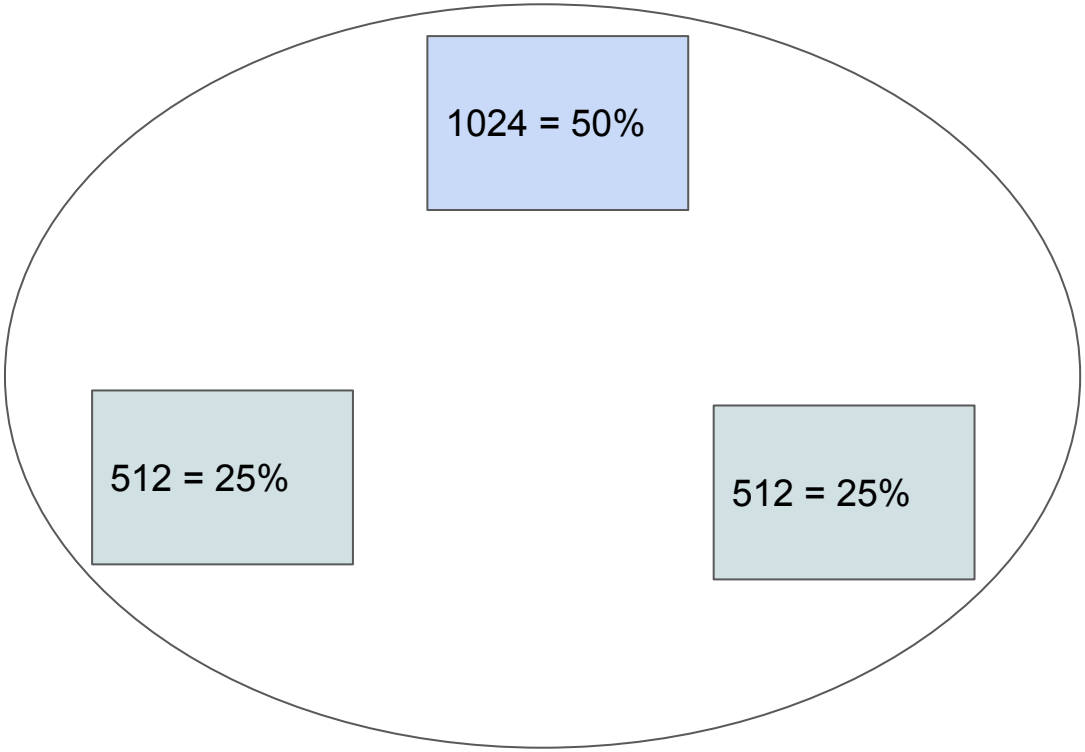
```
        "Memory": 536870912,
        "CpusetMems": "",
        "KernelMemory": 0,
        "KernelMemoryTCP": 0,
        "MemoryReservation": 0,
        "MemorySwap": -1,
        "MemorySwappiness": null,
```

Atribuindo a Qtd de CPU para um Container

rayner@Y720:~$ docker run -ti --cpu-shares 1024 --name teste2 ubuntu

rayner@Y720:~$ docker run -ti --cpu-shares 512 --name teste2 ubuntu

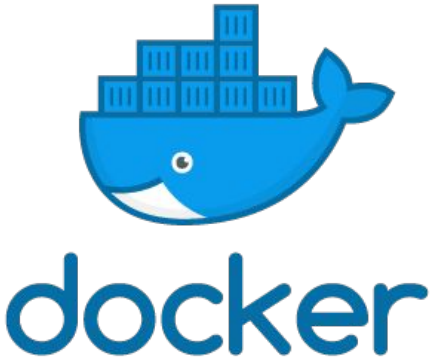
rayner@Y720:~$ docker run -ti --cpu-shares 512 --name teste3 ubuntu



Inspecionando o uso da CPU

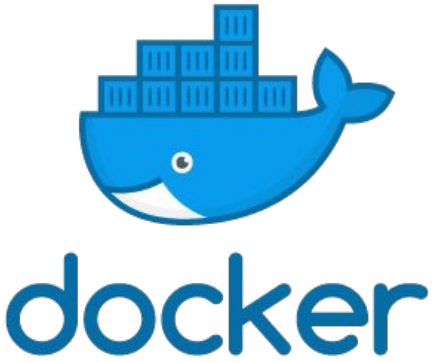
```
rayner@Y720:~$ docker inspect teste1 | grep -i cpu
            "CpuShares": 1024,
            "NanoCpus": 0,
            "CpuPeriod": 0,
            "CpuQuota": 0, (111)

rayner@Y720:~$ docker inspect teste2 | grep -i cpu
            "CpuShares": 512,
            "NanoCpus": 0,
            "CpuPeriod": 0,
            "CpuQuota": 0, (...)

rayner@Y720:~$ docker inspect teste3 | grep -i cpu
            "CpuShares": 512,
            "NanoCpus": 0, (...)
```

Alterando CPU e Memória em Execução




```
rayner@Y720:~$ docker run -ti --cpu-shares 1024 --name teste1 ubuntu

rayner@Y720:~$ docker inspect teste1 | grep -i mem
            "Memory": 0,
            "CpusetMems": "", (...)

rayner@Y720:~$ docker update -m 256m --cpu-shares 512 teste1
teste1

rayner@Y720:~$ docker inspect teste1 | grep -i mem
            "Memory": 268435456,
            "CpusetMems": "",
```

Removendo Containers



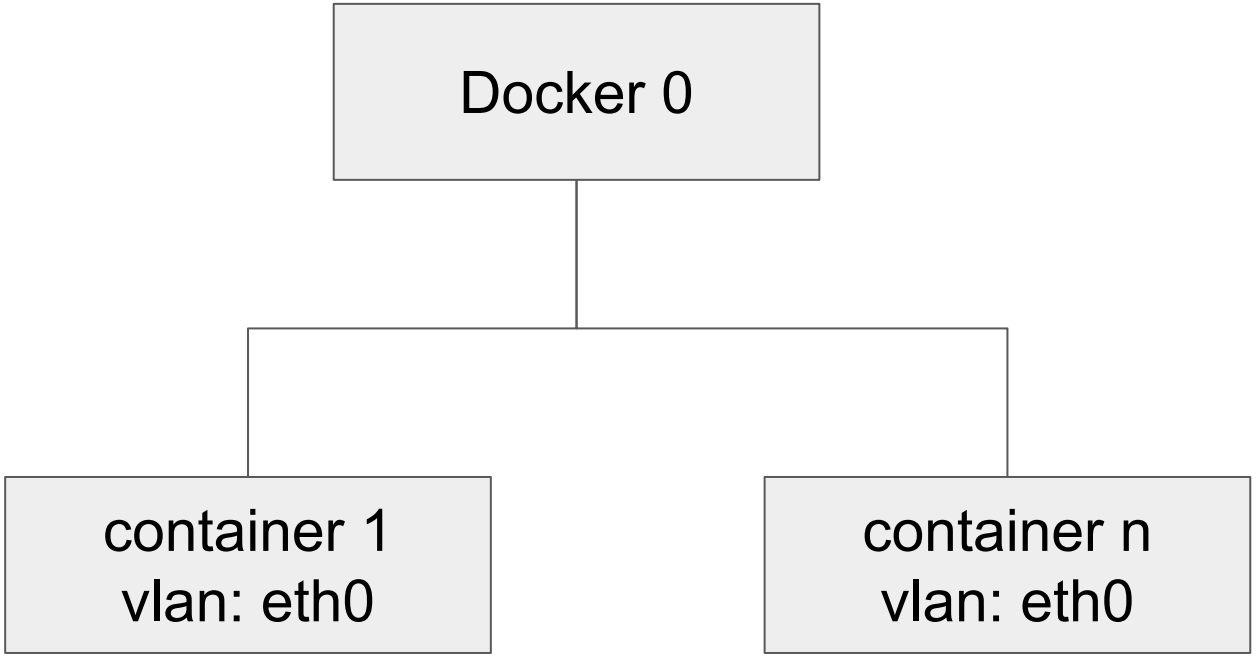
- **docker ps -a**: Mostra todos os containers
- **docker ps**: Mostra os containers ativos

```
rayner@Y720:~$ docker stop teste1

teste1

rayner@Y720:~$ docker rm teste1

teste1
```

Docker e a Rede

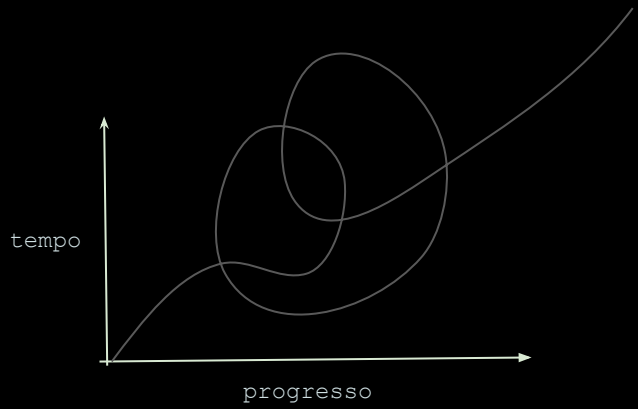
```
rayner@Y720:~$ ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:e6ff:fecf:6691  prefixlen 64  scopeid 0x20<link>
        ether 02:42:e6:cf:66:91  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 258  bytes 48884 (48.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

rayner@Y720:~$ docker run -ti ubuntu:python.1 bash
root@f676a5e61cc6:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 14  bytes 1897 (1.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Docker e a Rede



Tópicos Importantes para Aprender

- Dockerfile
- Volumes
- Gerenciamento Imagens
  - Docker Hub
- Docker e Rede
  - Open VSwitch



"Containers Smash" - Hulk

até a próxima aula.
[be continued]

Image source
https://www.tumgir.com/