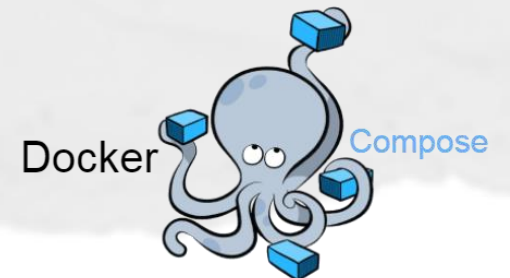
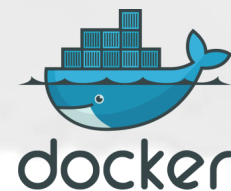
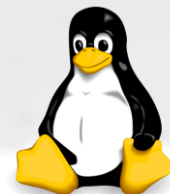


VIII CONFESO

DIREITOS HUMANOS  
E A AGENDA 2030

# Containers utilizando Docker e WSL2

Palestrante: Felipe Augusto



# Felipe Pimentel Augusto

Technical Architecture Manager @ Avanade since 10/2020









# Agenda

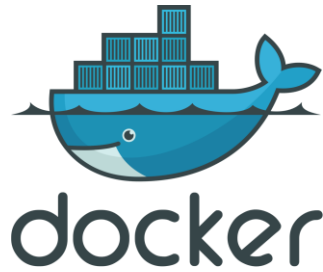
01

Windows Subsystem for Linux 2



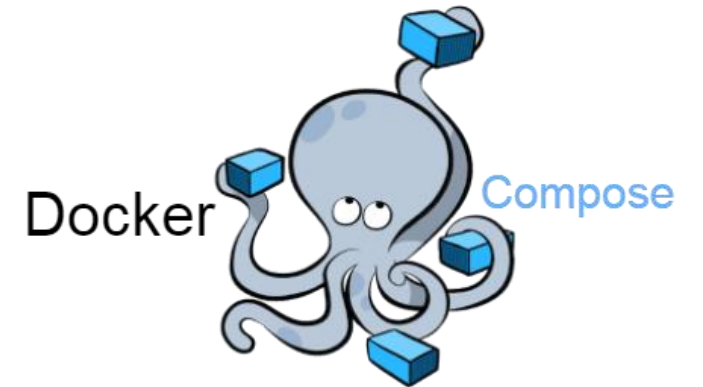
02

Docker



03

Docker Compose



slido



**Eu uso o WSL2 ou Linux!**

ⓘ Start presenting to display the poll results on this slide.

# WSL2



Chegou a hora  
do Linux dentro  
do Windows

- Algumas opções de distro;
- Terminal Bash dentro do Windows;
- Possibilidade de compartilhar o uso de arquivos e pastas.

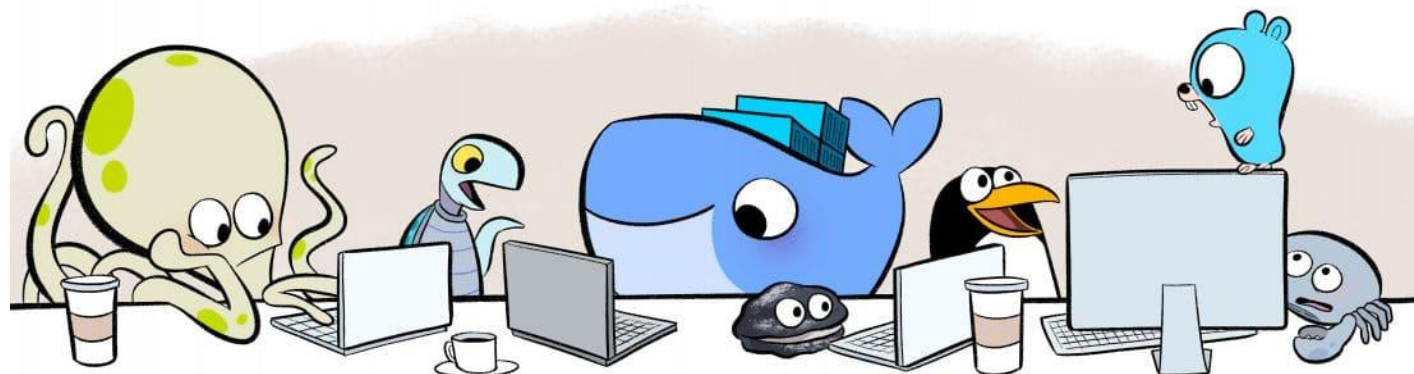




Docker Desktop

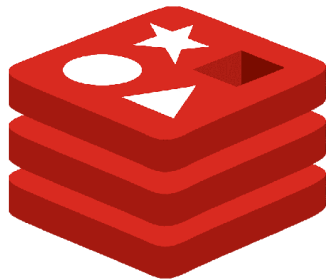
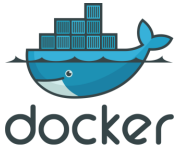
Desktop app

Desde 31 de agosto de 2021 com  
carência até 31 de janeiro de 2022



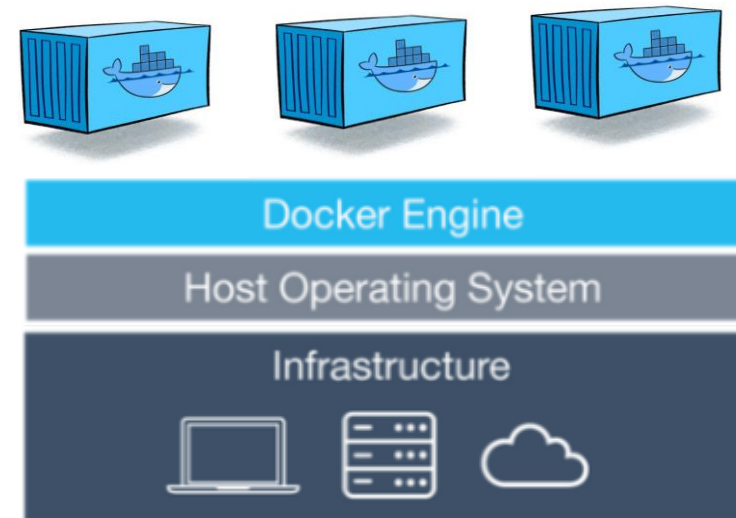
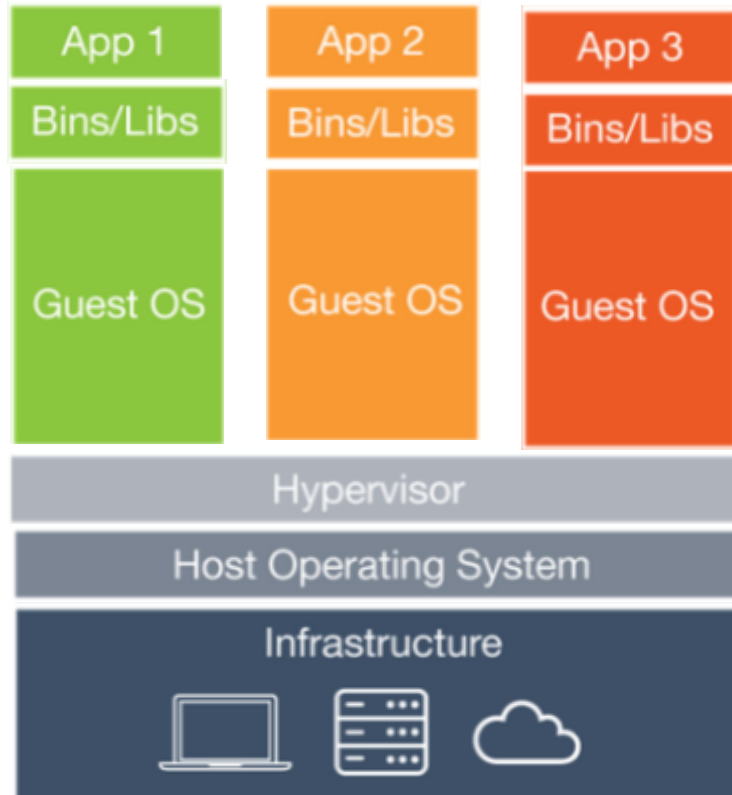
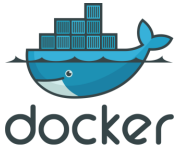
- O Docker Desktop permanece gratuito para pequenas empresas (menos de 250 funcionários e menos de US\$ 10 milhões em receita anual). Requer uma assinatura paga (Pro, Team ou Business), a partir de US\$ 5 por usuário por mês, para uso profissional em empresas maiores.
- Para uso pessoal, educação e projetos de código aberto não comerciais, pode ser utilizado de forma gratuita.

# Alguns softwares que podemos rodar em containers



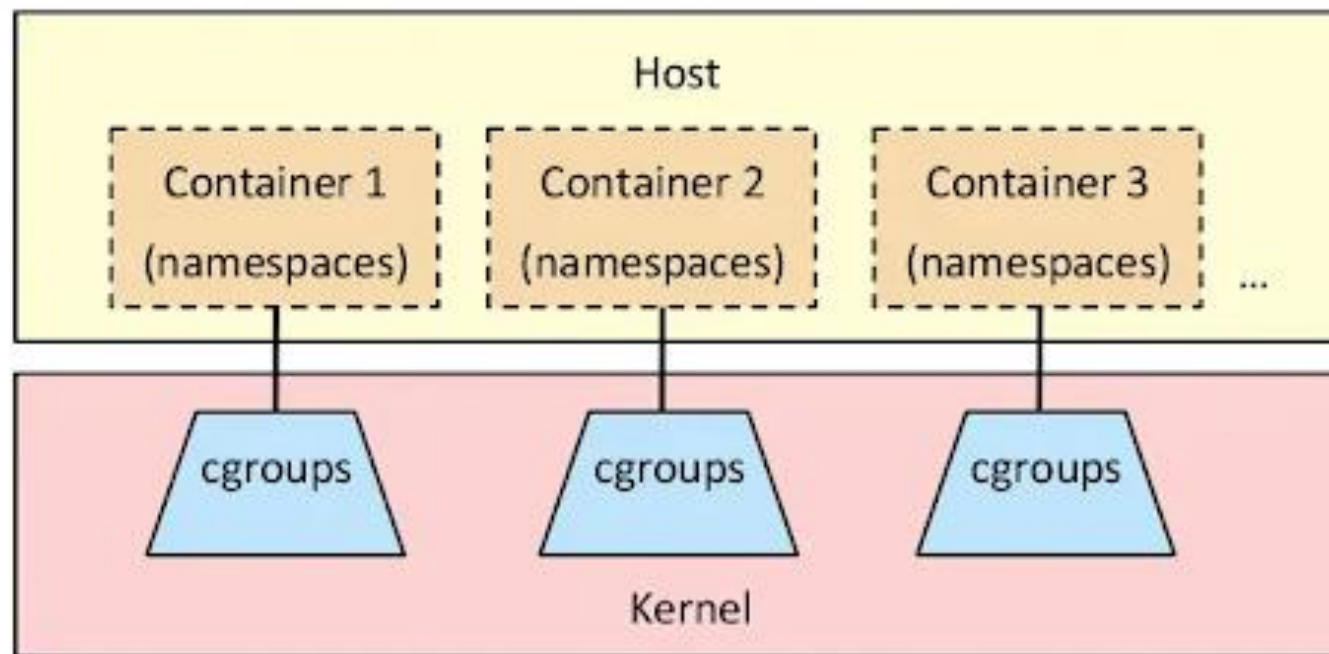


# VM e Docker



# Linux Containers


















Container = combination of namespaces & cgroups



- Namespaces | Capacidade de isolar processos
- Cgroups | Gerenciamento de recursos entre processos (CPU, Memória, Storage, Network)

# Tipos de runtime de containers

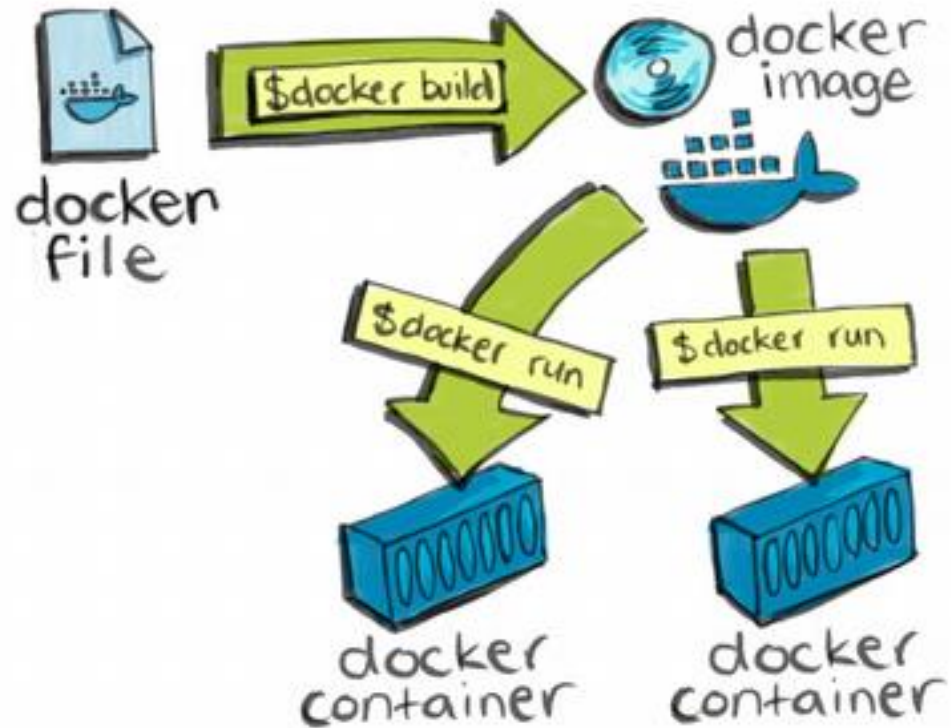


Runtime - Container Runtime (17)				
 <b>containerd</b> ★ 14,857 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>cri-o</b> ★ 4,754 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>Firecracker</b> ★ 22,578 Amazon Web Services	 <b>gVisor</b> ★ 14,172 Google	 <b>INCLAVARE</b> ★ 556 Inclavare Containers Cloud Native Computing Foundation (CNCF) Funding: \$3M
 <b>iSulad</b> ★ 325 OpenAtom Foundation	 <b>kata</b> ★ 3,818 Kata Containers Open Infrastructure Foundation	 <b>KUASAR</b> ★ 894 Kuasar Huawei Cloud	 <b>Lima</b> ★ 12,610 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>lxd</b> ★ 1,083 lxd Canonical Funding: \$12.8M
 <b>rkt</b> ★ 8,838 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 <b>runc</b> ★ 10,757 Open Container Initiative (OCI)	 <b>Singularity</b> ★ 2,459 Singularity Sylebo	 <b>SmartOS</b> ★ 1,519 SmartOS MNX Solutions	 <b>StratoVirt</b> ★ 314 StratoVirt OpenAtom Foundation
 <b>Sysbox</b> ★ 2,195 Neotybox Funding: \$125K	 <b>WasmEdgeRuntime</b> ★ 6,674 Cloud Native Computing Foundation (CNCF) Funding: \$3M			

Crunchbase data is used under license from Crunchbase to CNCF. For more information, please see the [license](#) info.



# Dockerfile



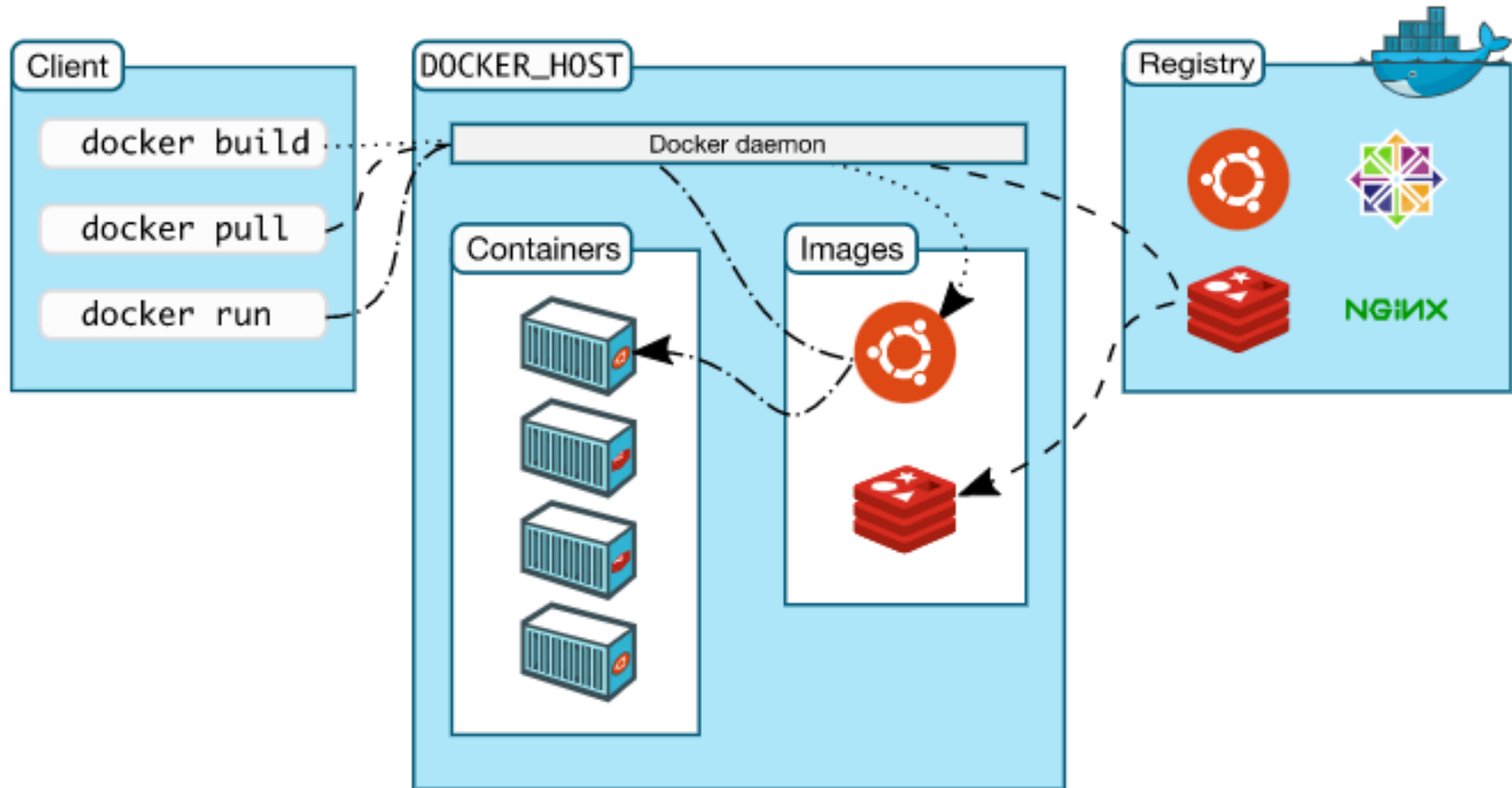
**felipementel** Create nodejs.dockerfile

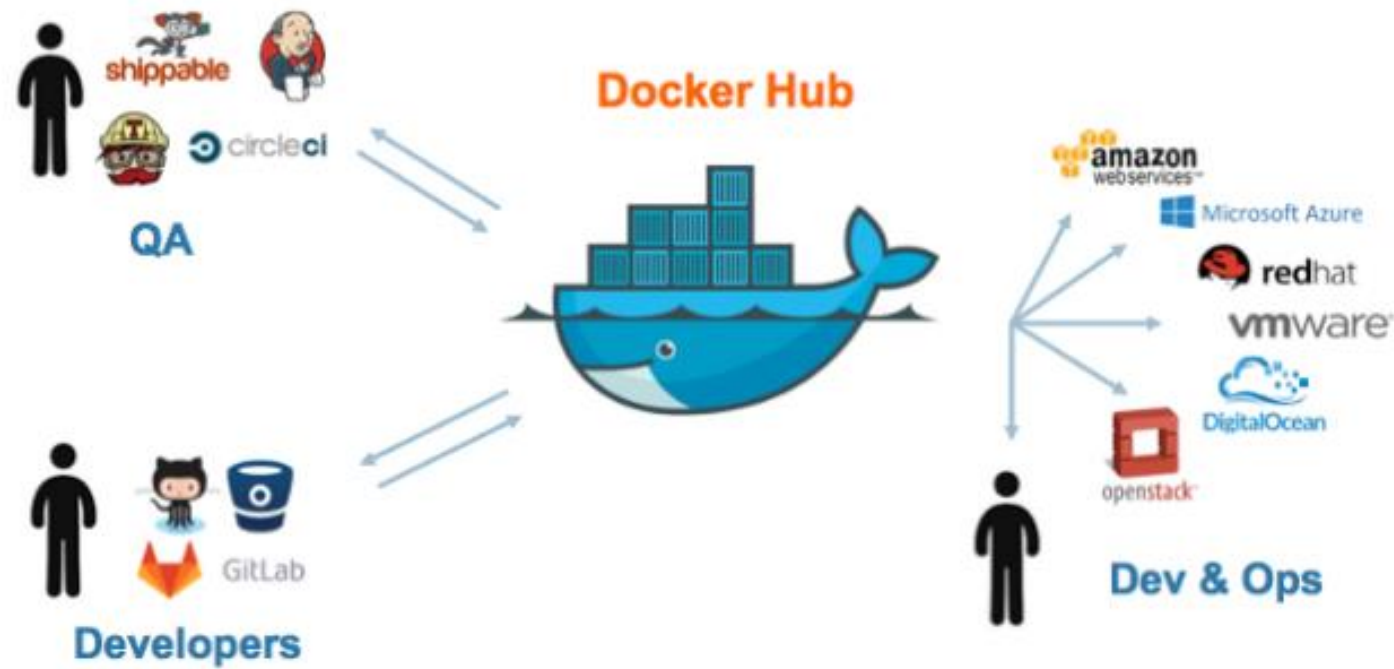
1 contributor

9 lines (9 sloc) | 176 Bytes

```
1 FROM node:17-alpine
2 WORKDIR /usr/src/app
3 COPY package*.json ./
4 COPY --chown=node:node . /usr/src/app
5 RUN npm install
6 COPY . .
7 EXPOSE 8080
8 USER node
9 CMD [ "node", "server.js" ]
```

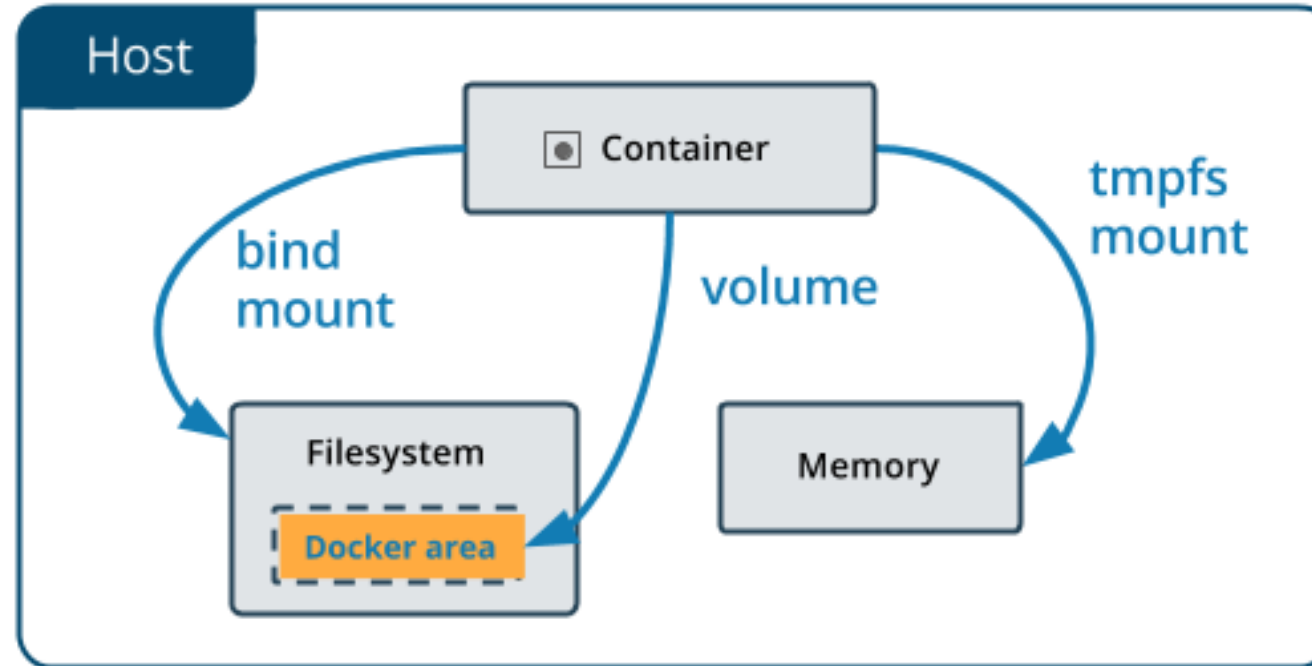
# Como o Docker funciona







# Tipos de volumes



# Tipos de redes

Bridge:: Driver padrão. Utilizando para comunicar containers

Host:: Utilizado entre o container e o host

Overlay:: Para conexão entre docker deamons

Ipvlan:: Para controle total sobre IPv4 e IPv6. Driver VLAN L2 ou L3

Macvlan:: Para atribuir endereço MAC e utiliza-lo para rotear o tráfego

None:: Para desativar todas as redes

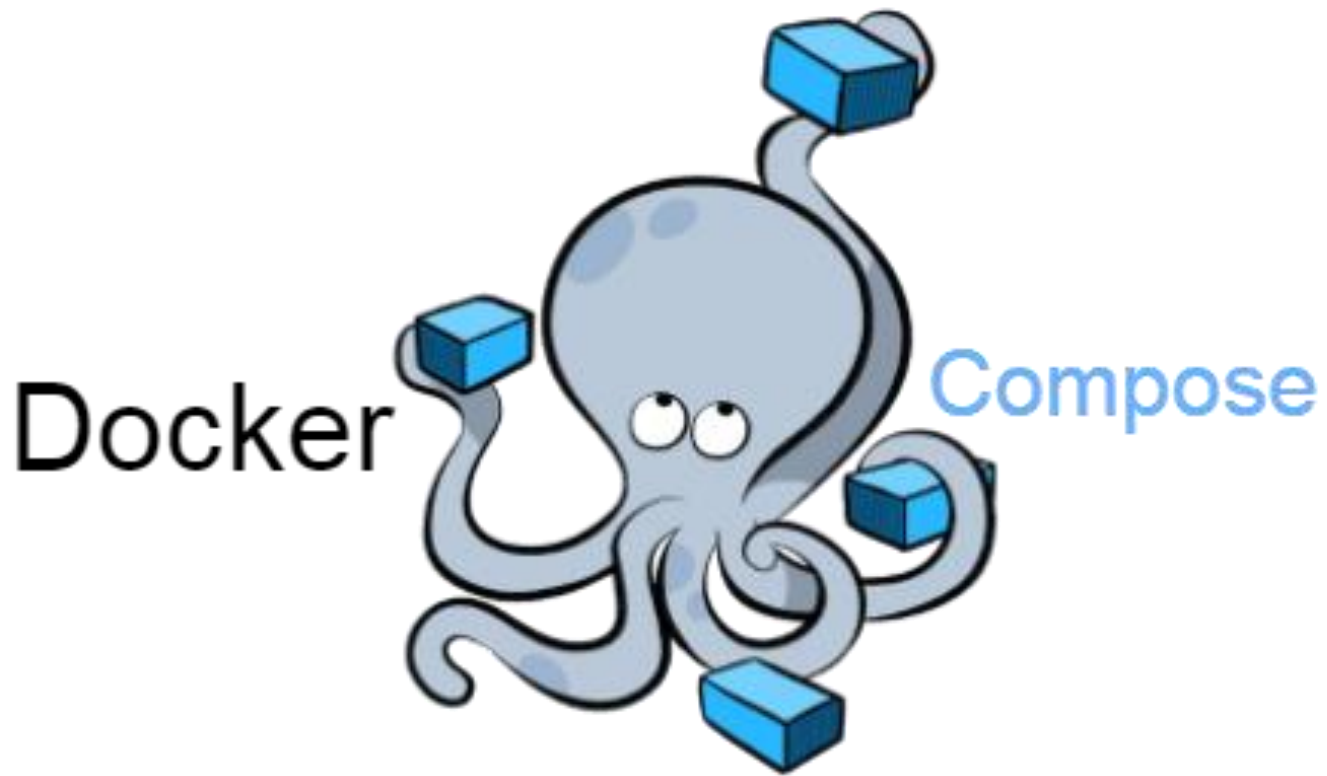
# Alguns comandos



build	push
commit	rm
exec	rmi
images	start
inspect	stop
logs	tag
pull	



# Exemplo de Docker Compose

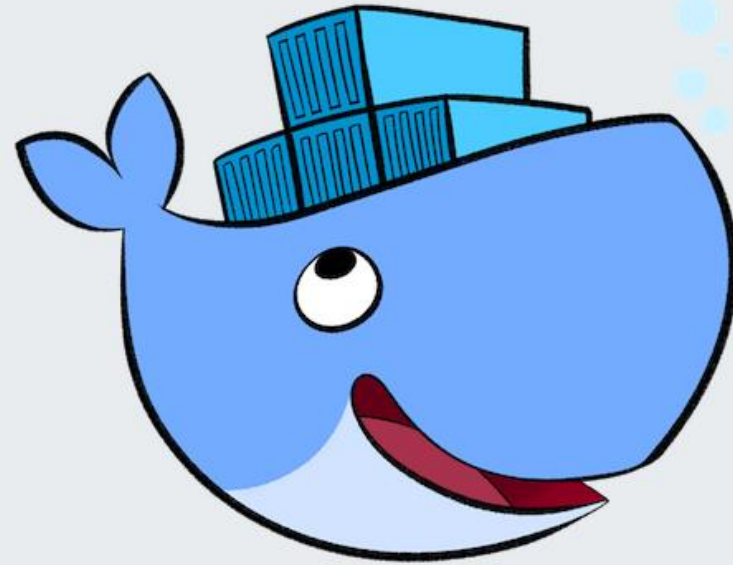


felipementel Create docker-compose-unifeso.yml

Code Blame 36 lines (31 loc) · 854 Bytes

```
1 # docker-compose -f docker-compose-unifeso.yml up -d
2 # docker-compose -f docker-compose-unifeso.yml down --remov
3
4 version: "3.7"
5
6 services:
7   sonar-canal-deploy:
8     image: sonarqube:9.9.2-community
9     ports:
10      - 9044:9000
11
12   rabbit-canal-deploy:
13     image: rabbitmq:3.12.6-management
14     ports:
15      - 15672:15672
16     environment:
17       RABBITMQ_DEFAULT_USER: admin
18       RABBITMQ_DEFAULT_PASS: deploy
19
20   postgres-canal-deploy:
21     image: postgres:16.0
22     environment:
23       POSTGRES_USER: user
24       POSTGRES_PASSWORD: deploy
25     volumes:
26      - ./postgres-data-unifeso:/var/lib/postgresql/data
27
28   pgAdminhn-canal-deploy:
29     image: dpage/pgadmin4:7.7
```

# Uma forma gratuita de utilizar o Docker



## Play with Docker

A simple, interactive and fun playground to learn Docker

Login ▼

# Referencias

- <https://docs.microsoft.com/en-us/windows/wsl/install>
- <https://docs.docker.com/get-started/>
- <https://docs.docker.com/engine/reference/commandline/cli/>
- <https://docs.docker.com/compose/reference/>
- <https://landscape.cncf.io/card-mode?category=container-runtime&grouping=category>
- <https://docs.docker.com/get-started/overview/#the-docker-daemon>
- <https://www.docker.com/blog/updating-product-subscriptions/>
- <https://docs.docker.com/storage/volumes/>
- <https://docs.docker.com/network/>
- <https://github.com/felipementel/Presentation.UNIFESO.Containers>
- <https://labs.play-with-docker.com/>

# Obrigado



<https://forms.office.com/r/Ldf91j71zC>



Espero ter contribuido

