



docker



kubernetes

- 01. Plataforma Docker
- 02. Objetos Docker
- 03. Arquitetura
- 04. Kubernetes
- 05. Na prática
- 06. Indicações de leitura



PLATAFORMA DOCKER

"**Docker** é uma plataforma **Open Source** para desenvolvimento, envio e execução de aplicativos que permite separar seus aplicativos de sua infraestrutura para que você possa entregar software de uma forma rápida. Com o **Docker**, você pode gerenciar sua infraestrutura da mesma forma que gerencia seus aplicativos."

<https://docs.docker.com/get-started/overview/>



OBJETOS

DOCKER

Imagens (*images*)

É um template apenas de leitura onde estão contidas as instruções para criar um contêiner **Docker**.

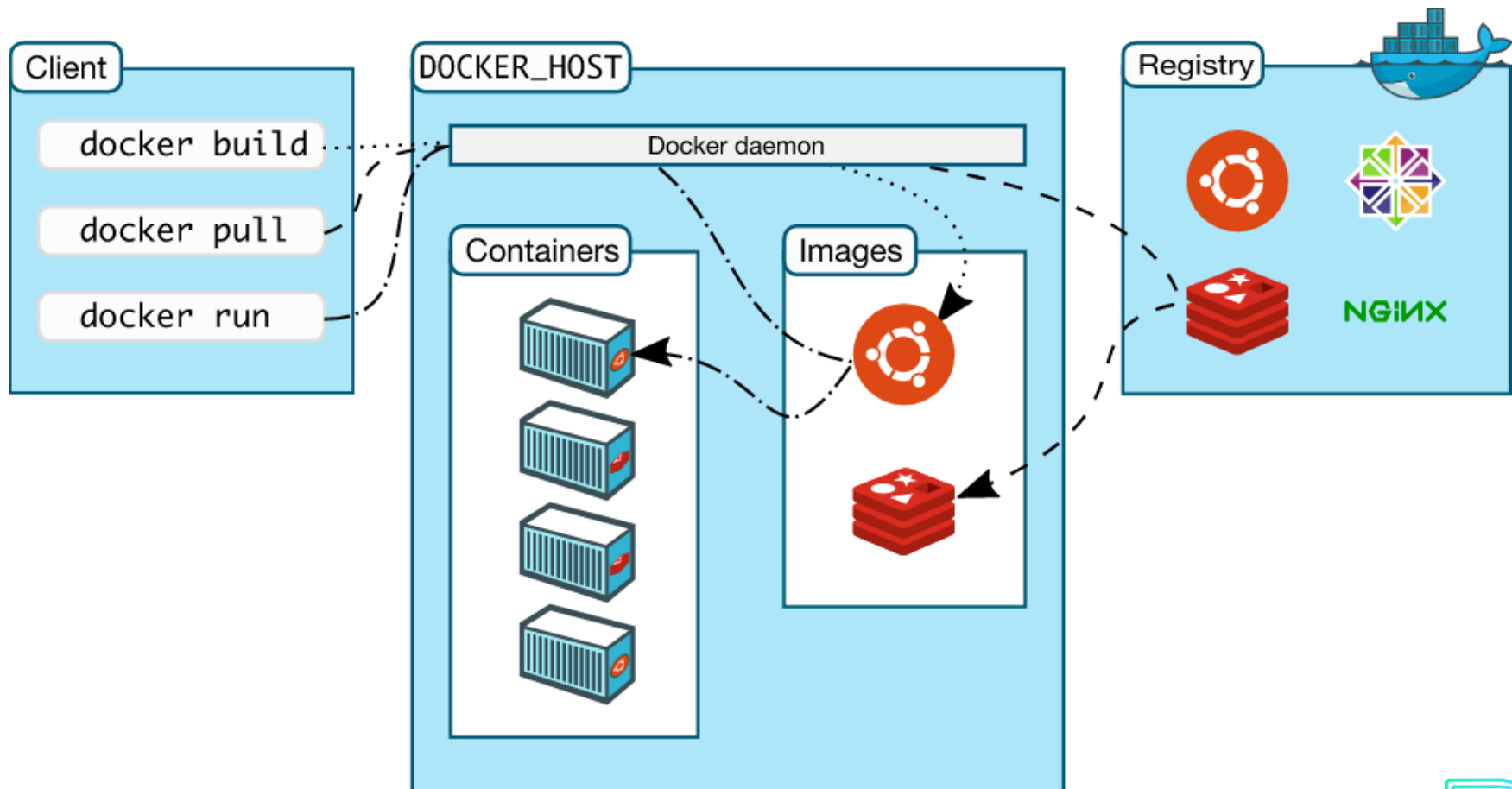
Contêiner (*container*)

Ambiente isolado e que pode ser executado a partir das definições de uma imagem.

<https://docs.docker.com/get-started/overview/>



ARQUITETURA DOCKER



<https://docs.docker.com/get-started/overview/>



KUBERNETES

Trata-se da plataforma de orquestração de contêineres, podendo ser complementar à containerização oferecida pela plataforma **Docker**.

"Com a orquestração do **Kubernetes**, é possível criar serviços de aplicações que abrangem múltiplos containers, programar o uso deles no cluster, escalá-los e gerenciar a integridade deles com o passar do tempo. Com o **Kubernetes**, você toma medidas reais para aprimorar a segurança da TI."

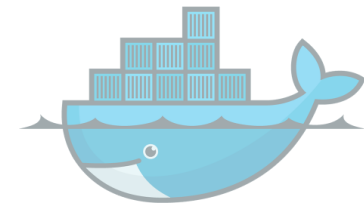
<https://www.redhat.com/pt-br/topics/containers/what-is-kubernetes>



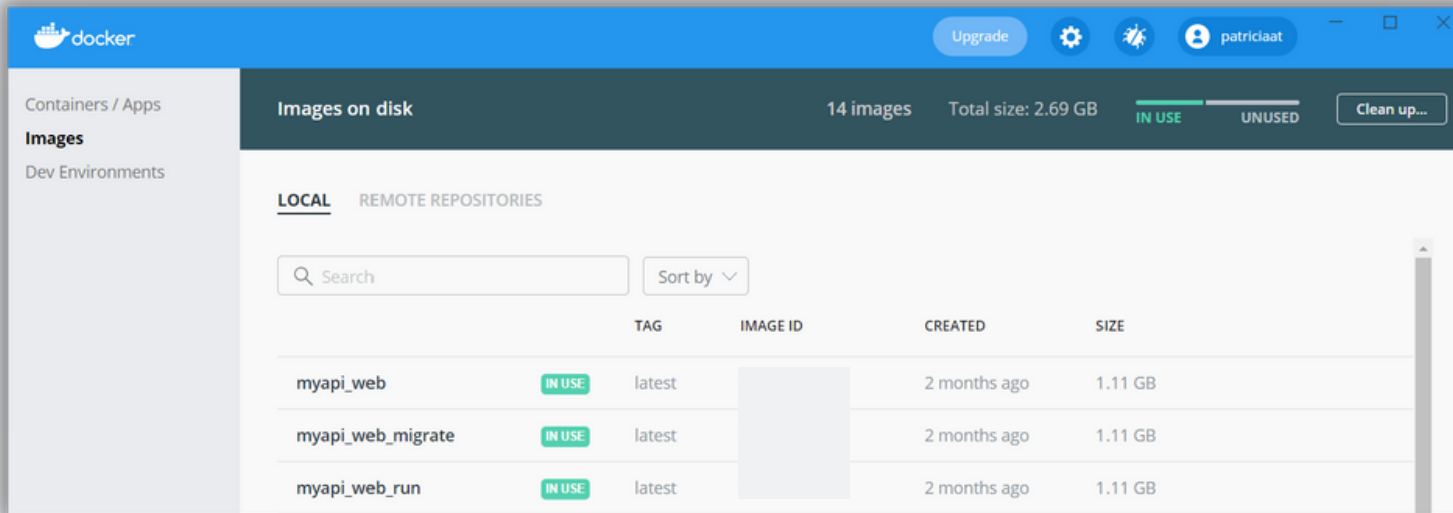
N A P R Á T I C A

```
docker-compose.yml
1  version: "3.9"
2
3  services:
4    web:
5      build: .
6      environment:
7        MYENV: venv
8      volumes:
9        - ./code
10   web_migrate:
11     extends:
12       service: web
13     command: python manage.py migrate
14   web_run:
15     extends:
16       service: web
17     command: python manage.py runserver 0.0.0.0:8080
18     ports:
19       - "8080:8080"
```

```
Dockerfile
1  FROM python:3.9.1
2
3  ENV PYTHONUNBUFFERED 1
4  RUN mkdir /store
5  WORKDIR /store
6  COPY requirements.txt /store/
7  RUN pip install -r requirements.txt
8  COPY . /store/
```

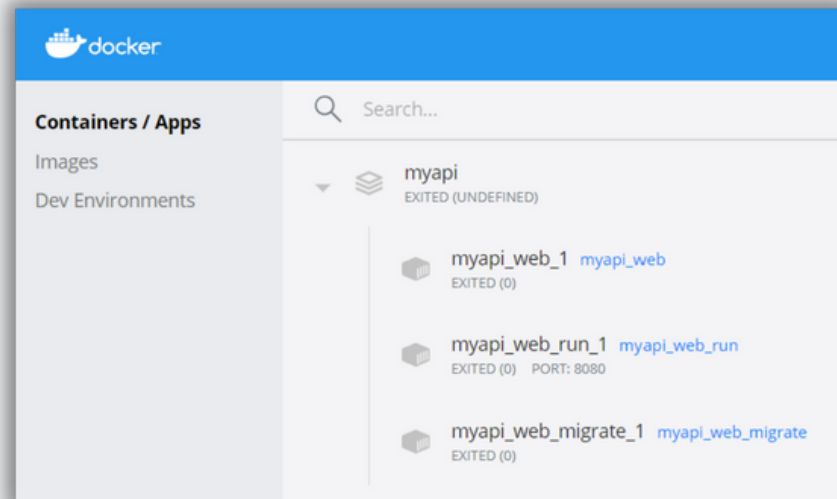


N A P R Á T I C A



The screenshot shows the Docker Desktop interface with the 'Images on disk' tab selected. The left sidebar shows 'Containers / Apps', 'Images', and 'Dev Environments'. The main area displays a table of local images. At the top, it indicates '14 images' and 'Total size: 2.69 GB'. A progress bar shows 'IN USE' (green) and 'UNUSED' (grey) status. A 'Clean up...' button is visible. The table has columns for TAG, IMAGE ID, CREATED, and SIZE. Three images are listed, all with the 'latest' tag and a size of 1.11 GB, all marked as 'IN USE'.

TAG	IMAGE ID	CREATED	SIZE	
myapi_web	IN USE	latest	2 months ago	1.11 GB
myapi_web_migrate	IN USE	latest	2 months ago	1.11 GB
myapi_web_run	IN USE	latest	2 months ago	1.11 GB



The screenshot shows the Docker Desktop interface with the 'Containers / Apps' tab selected. The left sidebar shows 'Containers / Apps', 'Images', and 'Dev Environments'. The main area displays a list of containers. At the top, there is a search bar. The list shows three containers, all with the status 'EXITED (0)'. Each container is associated with a specific image.

Container Name	Image	Status	Port
myapi_web_1	myapi_web	EXITED (0)	
myapi_web_run_1	myapi_web_run	EXITED (0)	PORT: 8080
myapi_web_migrate_1	myapi_web_migrate	EXITED (0)	



INDICAÇÕES DE LEITURA

<https://www.supero.com.br/blog/kubernetes-vs-docker/>

<https://blog.geekhunter.com.br/docker-na-pratica-como-construir-uma-aplicacao/>

<https://www.meupositivo.com.br/panoramapositivo/container-docker/>

