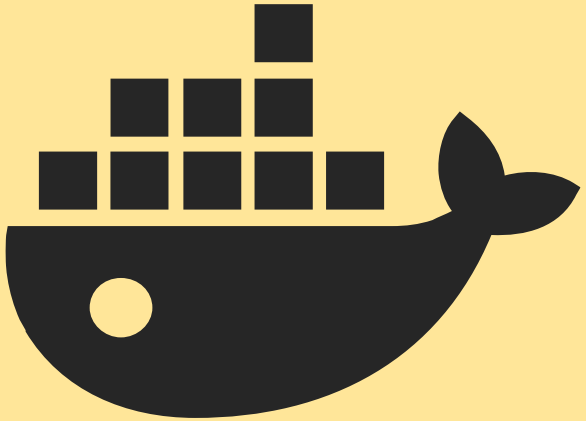
Iai qual o plano ?

Plano de Estudos

Roteiro

1. O mínimo necessário sobre Docker
   Uma "revisão introdutória"
2. Docker
   Conhecendo a plataforma
3. Primeiros passos em Docker
   Botando o container para rodar
4. VsCode
   Perfil de Desenvolvimento
5. O mínimo necessário sobre Python
   Uma "Revisão introdutória"
6. Python
   Data Types, Repetição, Módulo, Classe...
7. O mínimo necessário sobre Django
   Uma "revisão introdutória"
8. Django
   Conhecendo a framework
9. Primeiros passos em Django
   Botando o Django para rodar
11. Frameworks
    Conhecendo DataTables, Bootstrap, Echarts
12. Mkdocs
    Documentação é sempre bom
13. Criação do Projeto
    Mão na massa.



Uma "revisão introdutória"

Docker

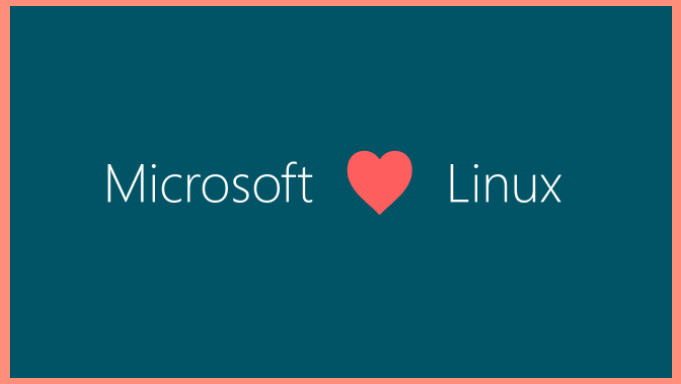
Introdução ao Docker

O que é Docker?

O Docker é uma plataforma de virtualização de software que permite criar e executar aplicativos em contêineres.

Ele foi criado em 2013 pela empresa Docker, Inc. e é baseado em tecnologias de virtualização de nível de sistema operacional, que permitem que um aplicativo e suas dependências sejam empacotados em um contêiner isolado e portátil.

Inicialmente o Docker rodava somente no Linux, mas devido ao seu sucesso gerou uma parceria com a Microsoft que trouxe os contêineres Docker e sua funcionalidade para o Windows a partir das versões Windows Server 2016 e do Windows 10 Anniversary Update.

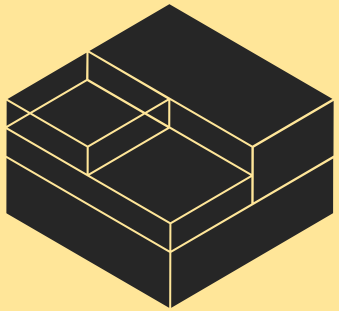


Introdução ao Docker

O que é um contêiner?

Containers (ou contêineres) são um tipo de tecnologia de virtualização que permite que um aplicativo ou serviço seja executado em um ambiente, sem a necessidade de uma máquina virtual completa, resolvendo a dor de cabeça “funciona na minha máquina”.

Hoje para milhões de desenvolvedores, o Docker é o padrão de fato para criar e compartilhar aplicativos em contêineres, mas existem outras plataformas, como:



LXC

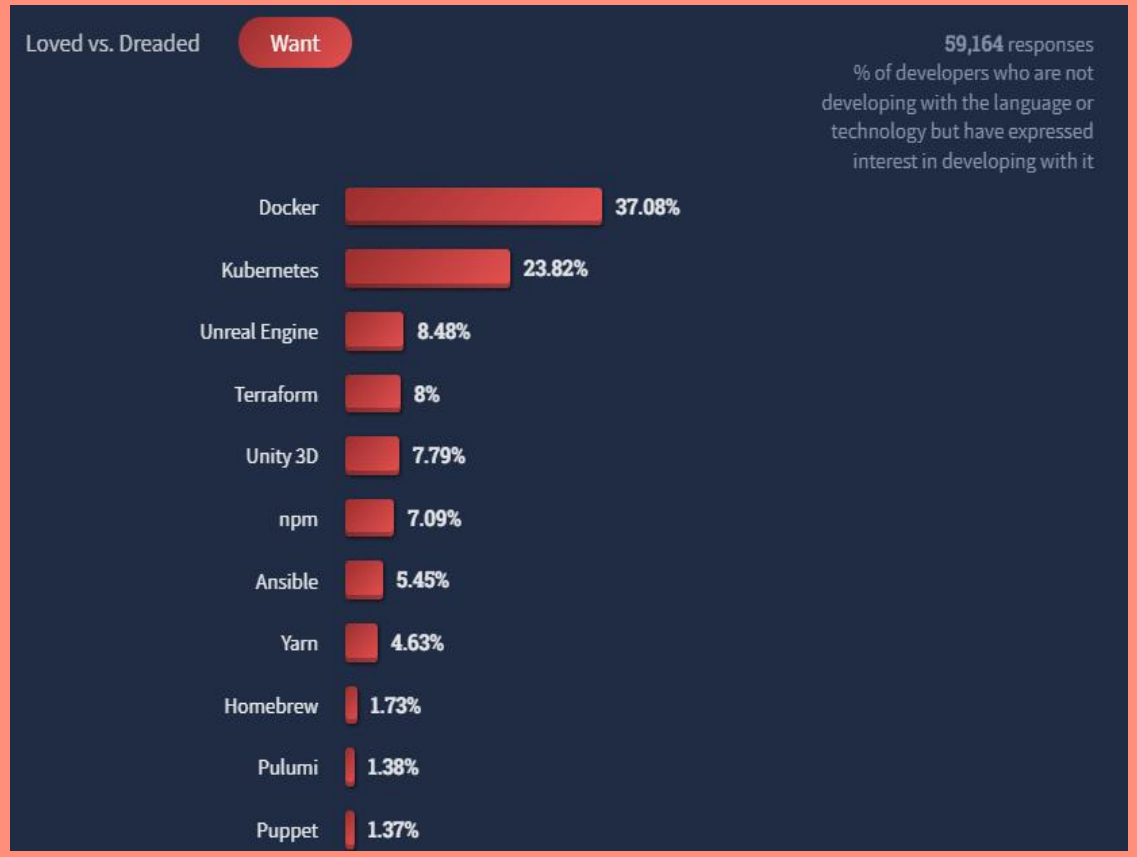
Kubernetes

Introdução ao Docker

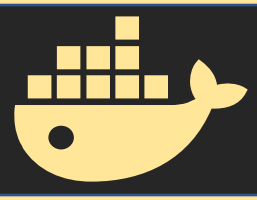
[Pesquisa StackOverflow 2022]

Docker é a ferramenta de desenvolvimento número 1 mais amada e continua sendo a ferramenta número 1 mais ferramenta desejada.

Docker teve um aumento, passou de 30% no ano passado para 37% este ano como ferramenta mais desejada.



Introdução ao Docker

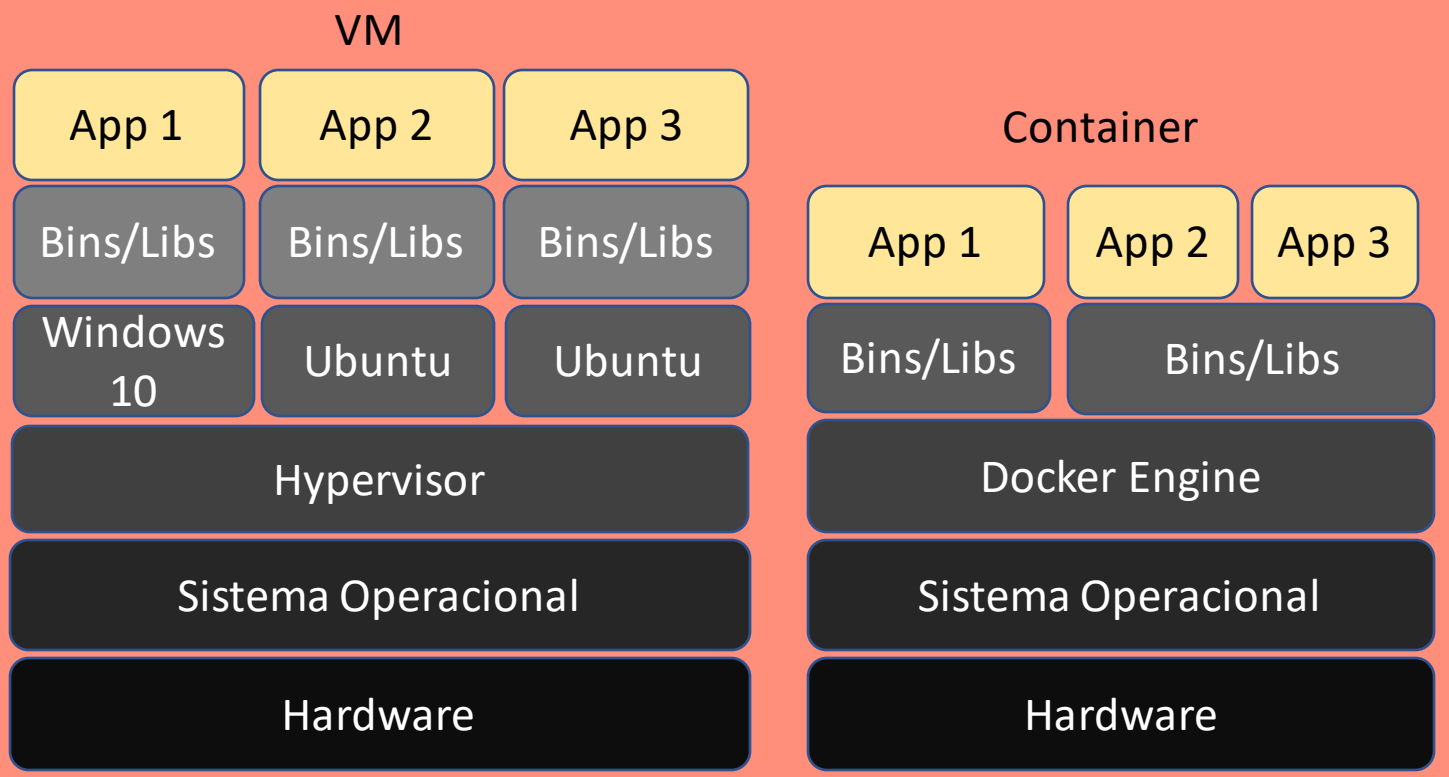


Quais são as vantagens do uso do Docker?

- **Portabilidade:** O Docker permite que os aplicativos sejam empacotados em um formato padrão que pode ser executado em qualquer sistema operacional ou ambiente de hospedagem.
- **Eficiência:** Os contêineres são muito mais leves do que as máquinas virtuais, pois não precisam de emular um sistema operacional completo para serem executados.
- **Escalabilidade:** O Docker permite que os aplicativos sejam facilmente escalados horizontalmente, adicionando ou removendo instâncias conforme necessário.

Introdução ao Docker

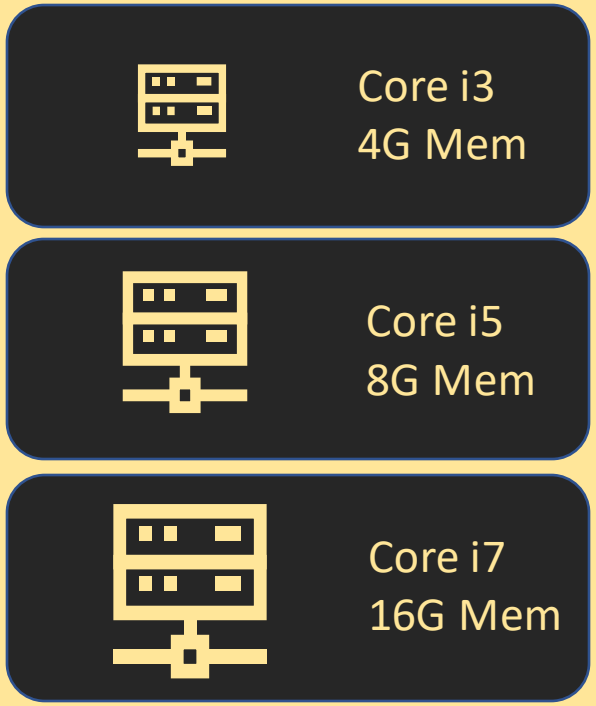
Comparando contêineres e máquinas virtuais



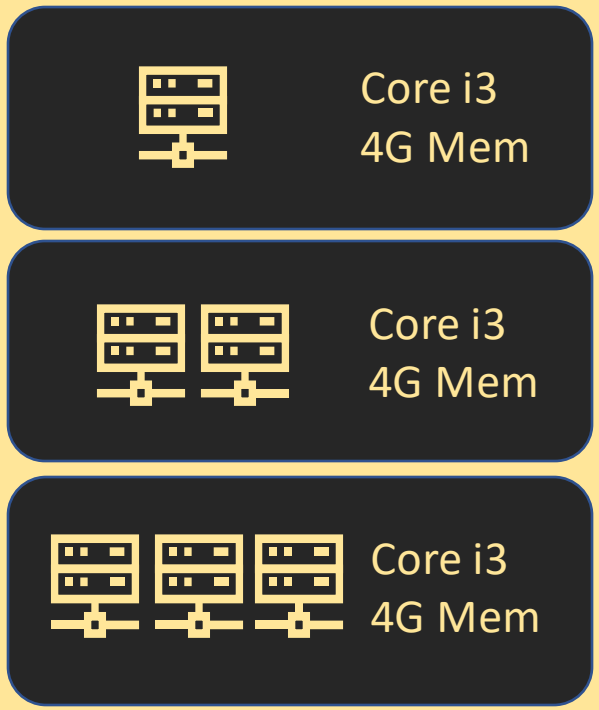
Introdução ao Docker

Tipos de escalonamento

Vertical
Scale up



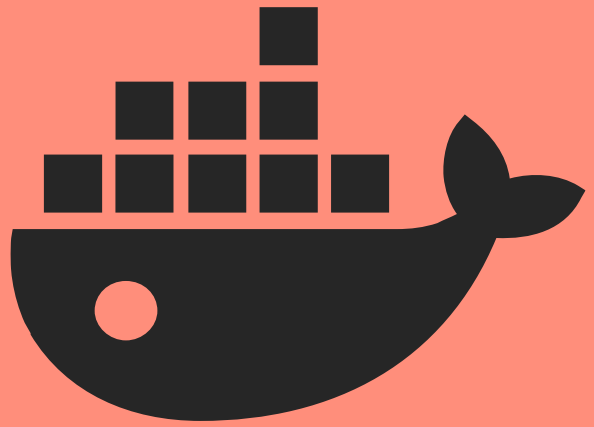
Horizontal
Scale out



Facilitação

O Docker fornece ferramentas para gerenciar facilmente vários contêineres em várias máquinas.

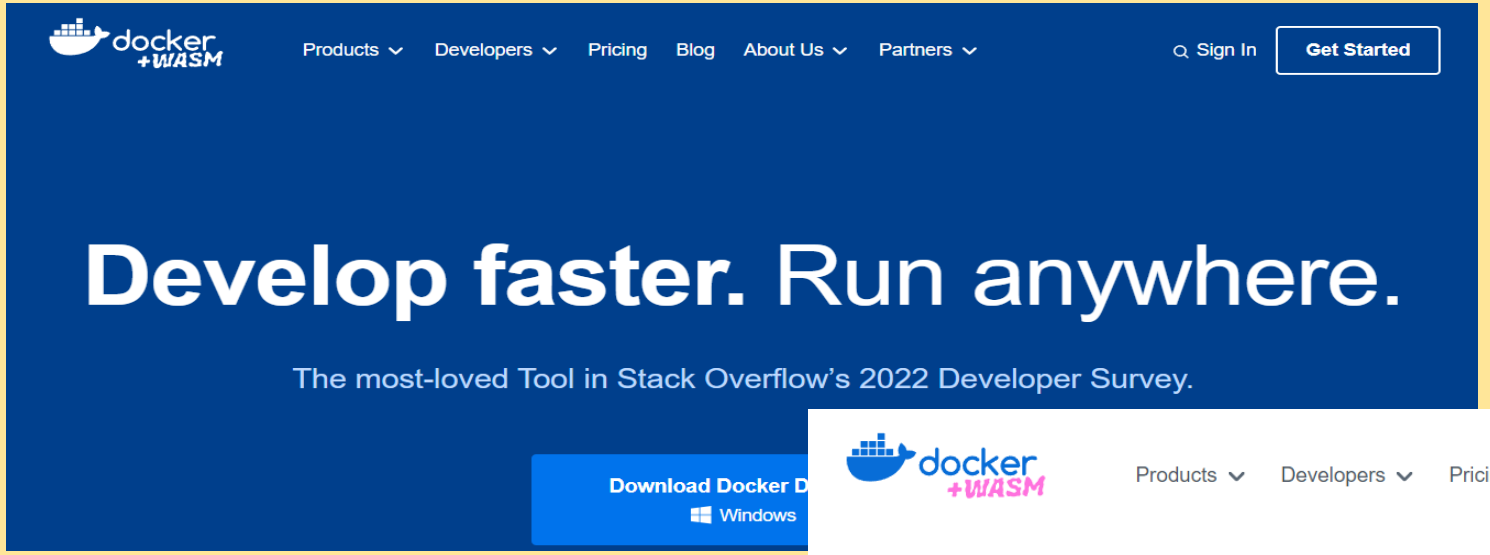
O Docker Swarm é uma ferramenta que pode ser usada para gerenciar um cluster de servidores em que vários contêineres estão sendo executados.

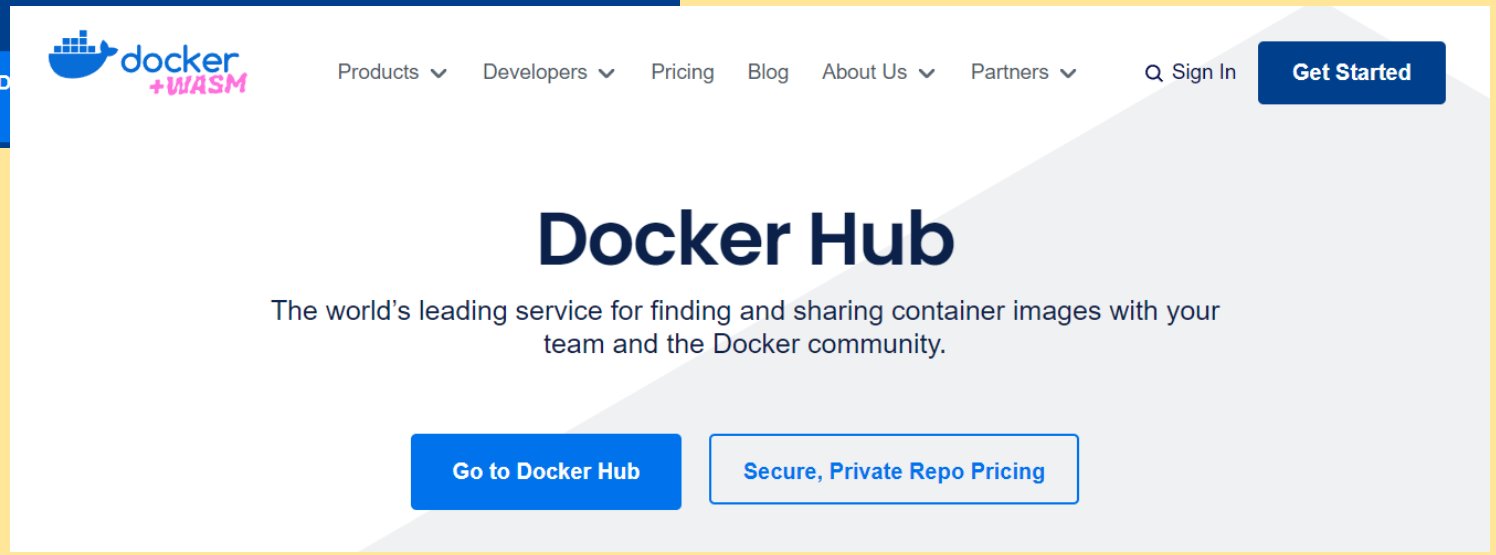


Conhecendo
a plataforma

Docker

Conhecendo a Plataforma





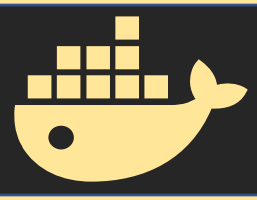
Conhecendo a Plataforma

Ritual do Hello World

```
#Baixando a imagem do Docker hub
docker pull hello-world

#Verificando as imagens
docker images

#Criando o container
docker run hello-world

#Verificando os containers em execução
docker ps

#Verificando todos os containers
docker ps -a
```

```
#Iniciando o container
docker start ID ou NAME

#Parando o container
docker stop ID ou NAME

#Removendo a imagem
docker rmi ID ou NAME

#Removendo o container
docker rm ID ou NAME
```

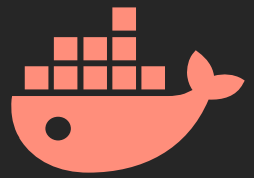
Conhecendo a Plataforma



Outros comandos

```
#Verificar informações de um container
docker inspect ID ou NAME

#Verificando logs de um container
docker container logs ID ou NAME

#Verificando processos do container
docker container top ID ou NAME

#Verificando recursos em uso
docker stats ID ou NAME

#Update de recursos
docker update ID ou NAME –m 256M --cpus 0.2
```

Conhecendo a Plataforma



Como criar uma imagem ?

Existem várias formas de criar uma imagem Docker, aqui estão algumas delas:

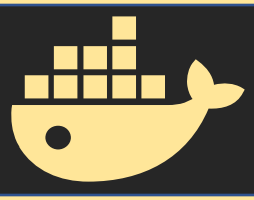
1. **A partir de um Dockerfile:** Um Dockerfile é um arquivo de texto que contém as instruções para criar uma imagem. Você pode criar um Dockerfile e executar o comando **`docker build`** para criar uma imagem a partir dele. O Dockerfile descreve o ambiente em que o aplicativo será executado e inclui todas as dependências necessárias.

2. **A partir de um contêiner em execução:** Se você tiver um contêiner em execução que esteja configurado exatamente como deseja, poderá criar uma imagem a partir dele usando o comando **`docker commit`**. Isso criará uma nova imagem a partir do estado atual do contêiner.

3. **A partir de um arquivo tar:** Você também pode criar uma imagem a partir de um arquivo tar que contenha o sistema de arquivos do contêiner. Você pode criar o arquivo tar usando a opção **`docker export`** e, em seguida, criar uma nova imagem a partir dele usando a opção **`docker import`**.

4. **Usando uma ferramenta de terceiros:** Há também várias ferramentas de terceiros que podem ajudá-lo a criar imagens Docker, como o Docker Compose, que permite criar e gerenciar aplicativos multi-container.

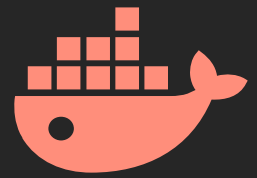
Conhecendo a Plataforma



Criando uma imagem com docker commit

```
#Criar o container com todos os requisitos para rodar a aplicação.
docker run -dti -p 8080:80 nginx

#Criar a imagem a partir do container existente.
docker commit ID ou NAME USERNAME/minha-primeira-imagem:1.0

#Inicializar um novo container a partir na imagem criada.
docker run -dti -p 8080:80 USERNAME/minha-primeira-imagem:1.0
```

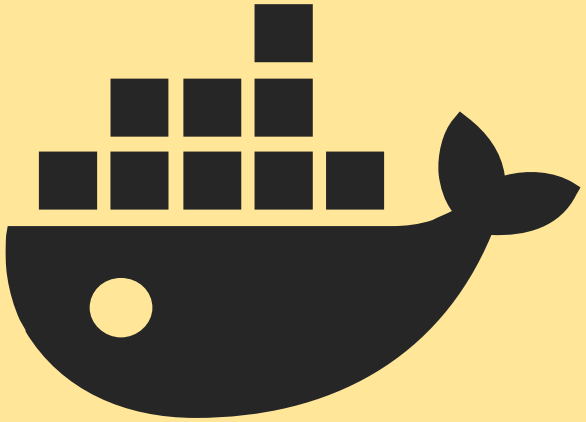
Conhecendo a Plataforma



Como compartilhar minha imagem?

Para compartilhar uma imagem Docker, você pode fazer o upload da imagem para um registro de contêineres (container registry) acessível pela Internet, como o Docker Hub.

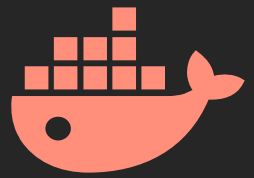
```
#Faça login.
docker login --username=USERNAME

#Execute o push da imagem.
docker push USERNAME/minha-primeira-image:v1.0
```



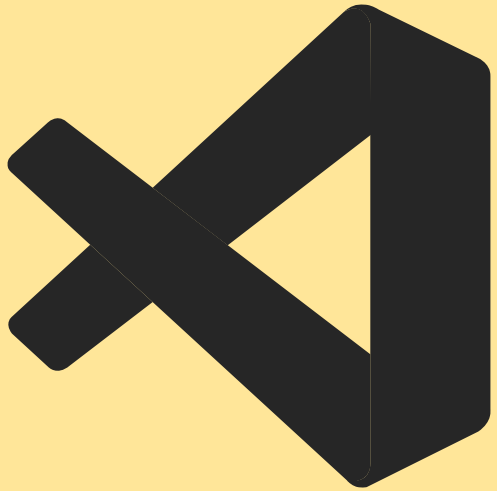
Botando o container para rodar

Docker

Container UP



```
# Criando nosso container
docker run -dti --name Ubuntu-Estudo -p 8000:8000 -p 8001:8001 -m 8G --cpus 0.5 ubuntu

Parâmetros

    -d: Inicia o contêiner em segundo plano (modo daemon).

    -t: Aloca um pseudo-TTY (terminal) para o contêiner.

    -i: Mantém STDIN aberto, mesmo que não esteja conectado.

    --name: Define um nome personalizado para o contêiner.

    -p 8000:8000: Mapeia a porta 8000 do host para a porta 8000 do contêiner.

    -p 8001:8001: Mapeia a porta 8001 do host para a porta 8001 do contêiner.

    -m 8G: Limita a memória RAM que o contêiner pode usar a 8GB.

    --cpus 0.5: Limita o número de CPUs que o contêiner pode usar a 0.5 (50% de uma CPU).

    ubuntu: Especifica a imagem que será usada para criar o contêiner. Neste caso, a imagem é "ubuntu".

# Acessando  o container
docker exec -ti Ubuntu-Estudo /bin/bash
```



S2 S2

VsCode

Profile

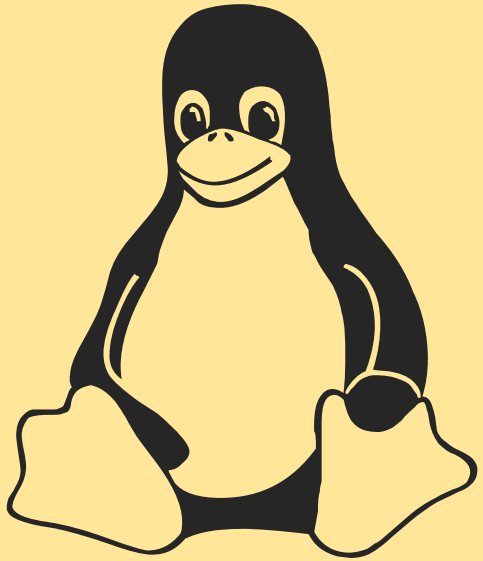
Gerenciando perfis

Alternar entre perfis ajuda em não instalar extensões desnecessárias em host remotos.

1. Criar o profile.

2. Instalar as extensões:

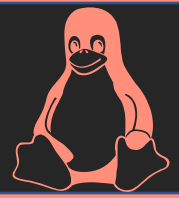
    - Dev Containers
    - Django
    - Python
    - Isort
    - Autopep8



Gestão de pacotes

Linux

Bash



Comandos Linux mais comuns de gerenciamento de repositórios:

- **apt update**: Atualiza o índice de pacotes disponíveis nos repositórios configurados no sistema. Verifica se há atualizações disponíveis.
- **apt upgrade**: Atualiza os pacotes já instalados no sistema para suas versões mais recentes.
- **apt install**: Instala um pacote especifico no sistema.
- **add-apt-repository**: Adicionar repositórios mantidos por terceiros que não fazem parte dos repositórios oficiais das distribuições Linux.

Atualização e instalação pacotes necessários:

```
apt update -y
apt upgrade -y
apt install git -y
apt install build-essential checkinstall libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev -y
add-apt-repository ppa:deadsnakes/ppa
apt install python3.11 python3.11-venv -y
```