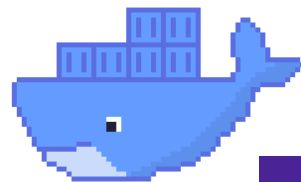




trybe

Introdução ao Docker

Bloco 19 - Aula 19.1



Combinados



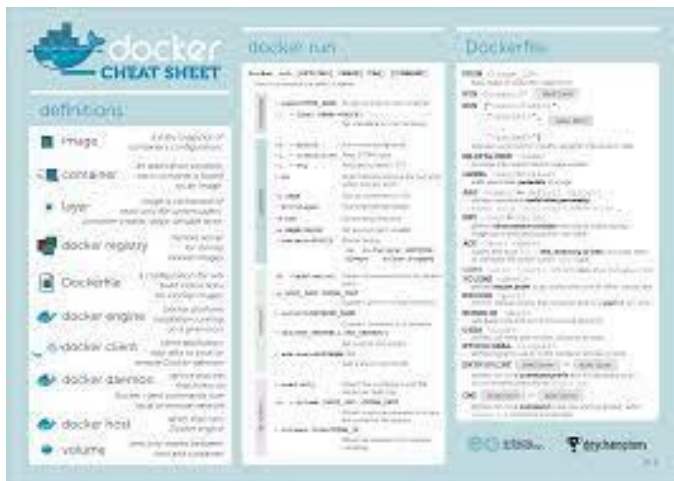
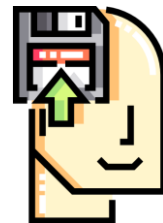
- Chat durante a aula
- Slido e Mãos Levantadas
- Microfones Mutados
- Reações
- Câmeras ligadas
- Dúvidas sobre instalação do Docker devem ser levadas para o plantão



Finalizamos a entrega...



- Conceitos que abrangem etapas durante o desenvolvimento de software ou após o processo de desenvolvimento de software
- **Mas são muitos comandos, qual técnica devo usar para decorar?**



Etapas de Produção de um Software

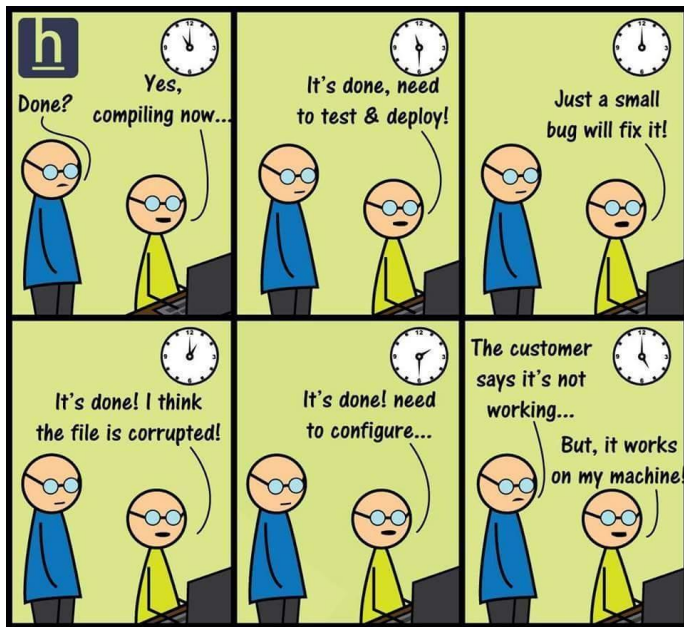


- Desenvolvimento
- Testes
- **Produção**





- Não se torna prático/escalável para a equipe que eu envie o código para as outras pessoas do time de forma simplificada, visto que precisaríamos enviar também as instruções detalhadas para que a aplicação consiga rodar



Ambiente de Produção



- Configuração do Sistema Operacional
- Banco de Dados
- Softwares
- Bibliotecas
- Configurações



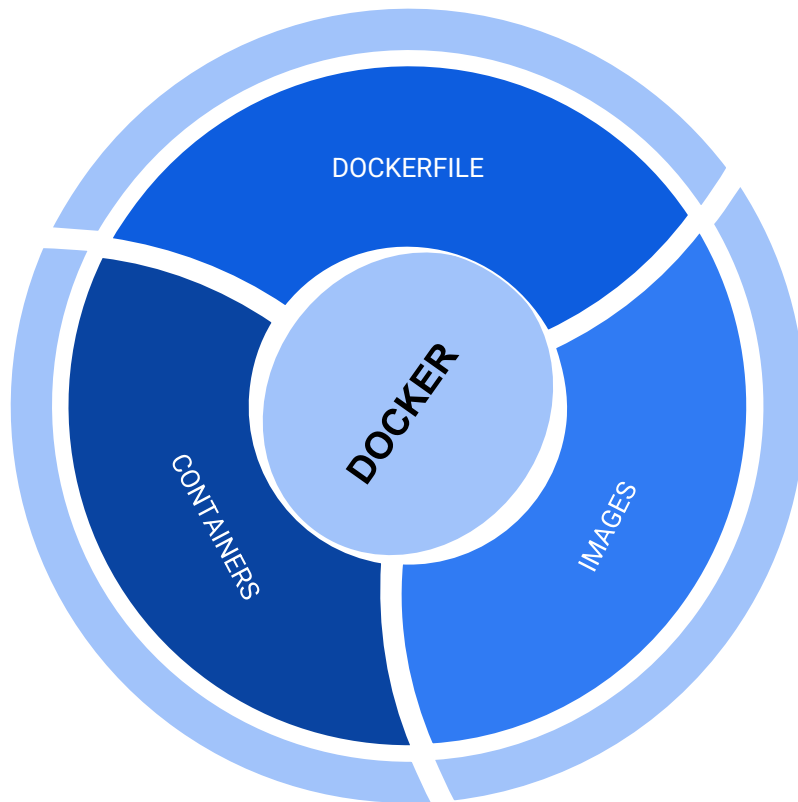
Ambiente de Produção



- É por isso que se torna inviável ter que subir e configurar um sistema toda vez que precisarmos rodar um aplicativo



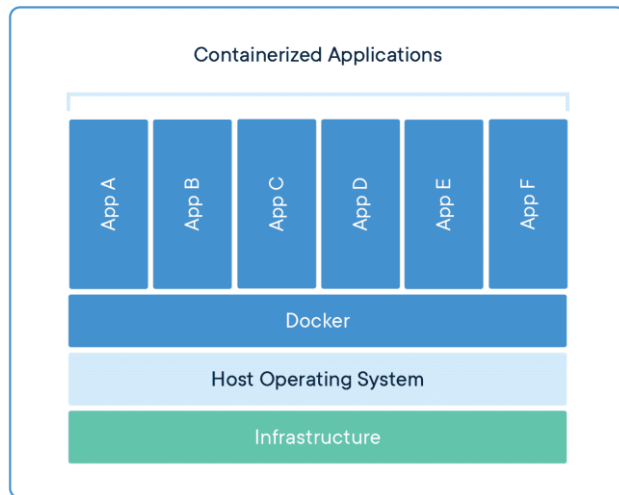
Docker



Docker



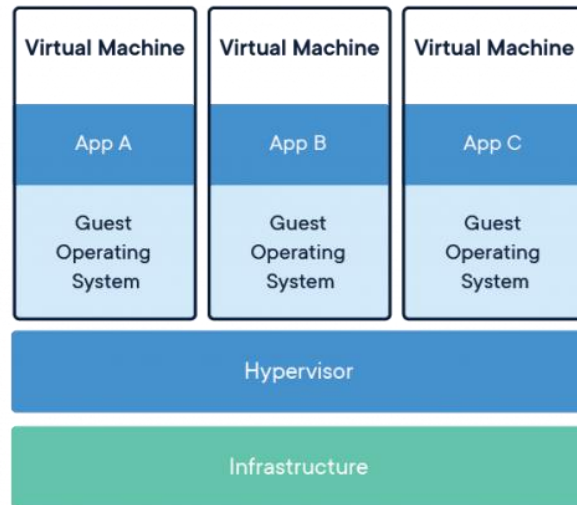
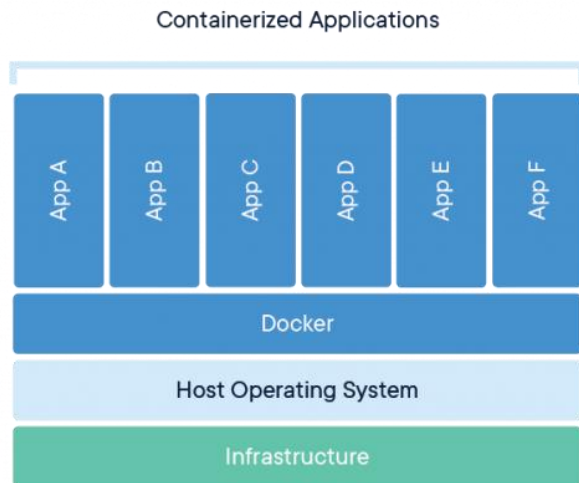
- Docker chega para resolver tais dilemas
- Tudo que é instalado em nossa máquina pode ser **containerizado** utilizando a tecnologia do Docker
- **Docker é uma tecnologia de virtualização que possibilita o empacotamento de uma aplicação ou ambiente inteiro dentro de um contêiner**



Docker - Containers



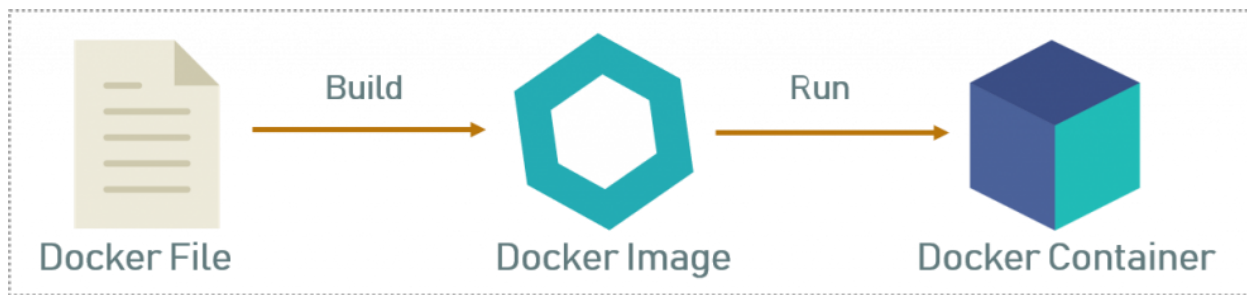
- Um contêiner é um ambiente isolado dentro de um servidor;
- Imagine um trem de carga com diversos contêineres de mercadorias. Se em um dos containers a mercadoria estragar, isso não vai afetar os outros containers, pois cada um deles está isolado e protegido



Docker - Imagens



- É um modelo de um sistema de arquivos (somente leitura) usado para criar contêiner;
- As imagens são criadas através de um processo de build, com um descritor (DockerFile) com um passo a passo e os detalhes para a geração da imagem;

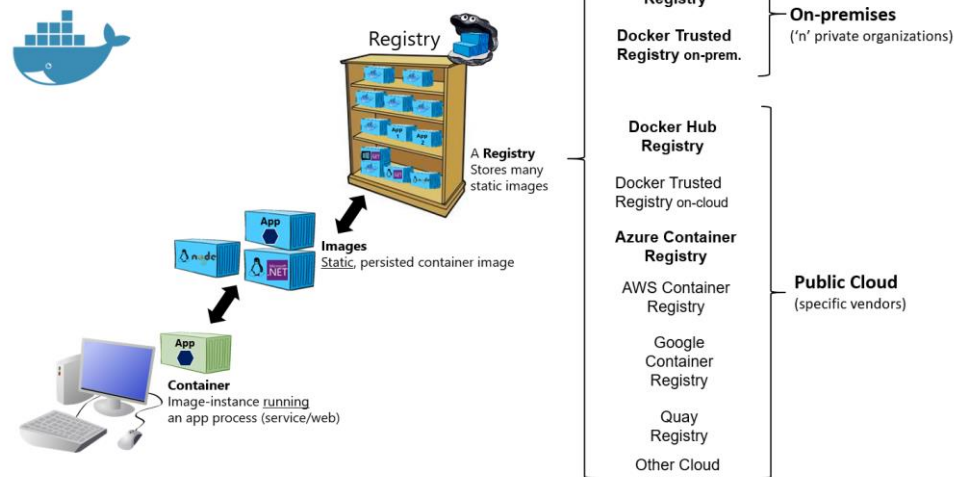


Docker - Imagens



- Uma imagem é apenas um template com instruções para criar um container;
- Podemos criar a nossa própria imagem usando o Dockerfile e reaproveitar outras imagens para adaptar em nosso projeto.

Basic taxonomy in Docker



Docker - DockerFile



- O Dockerfile nada mais é do que um meio que utilizamos para criar nossas próprias imagens;
- Em outras palavras, ele serve como a receita para construir um container, permitindo definir um ambiente personalizado e próprio para meu projeto pessoal ou empresarial;



Docker - DockerFile



- Exemplo de DockerFile;

```
FROM node:alpine

RUN mkdir -p /app

WORKDIR /app

COPY . /app

RUN npm install

EXPOSE 3000

CMD [ "node", "index.js" ]
```

Docker - Container e Comunicação



- Cada container fica isolado, podendo se comunicar com outros containers usando a rede (network), dando acesso às portas de cada aplicação;
- Inclusive o redirecionamento de portas pode ser feito de um jeito bem simples;

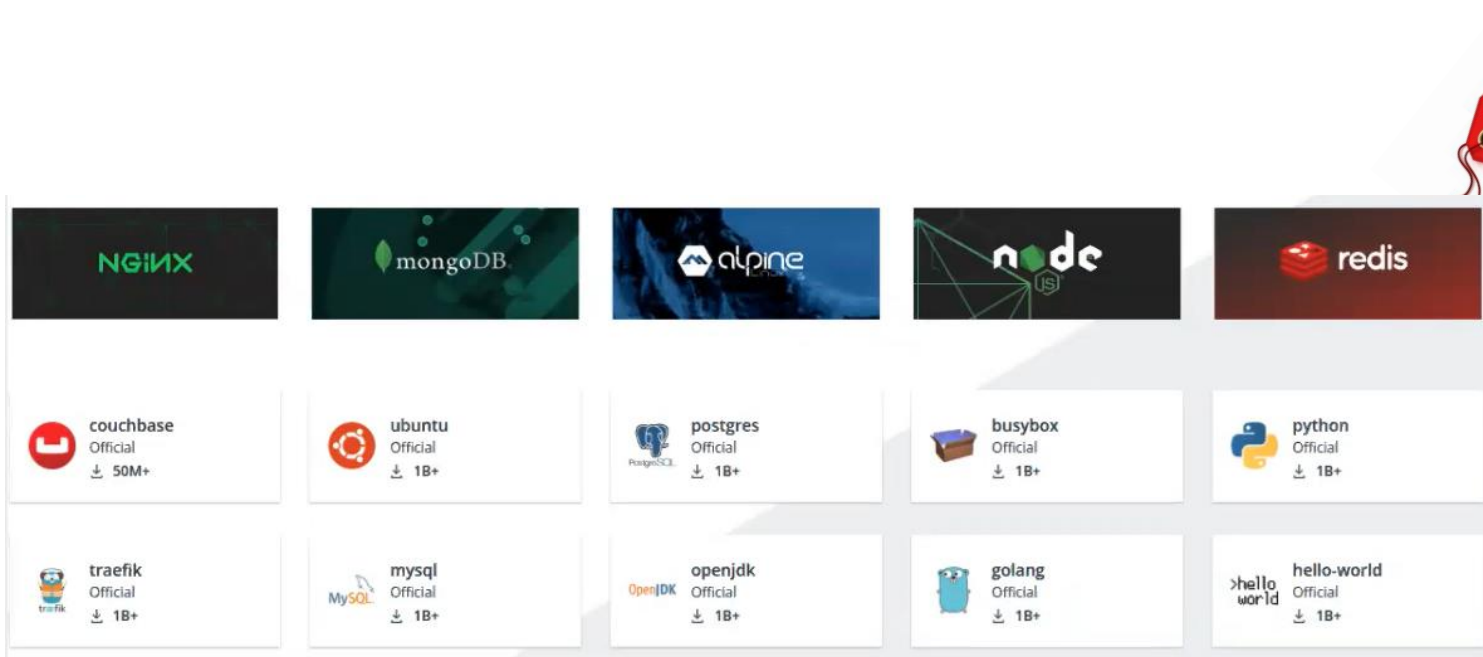
```
docker run --name db_postgres -e POSTGRES_PASSWORD=docker -  
p 5432:5432 -d postgres
```



Docker - Hub



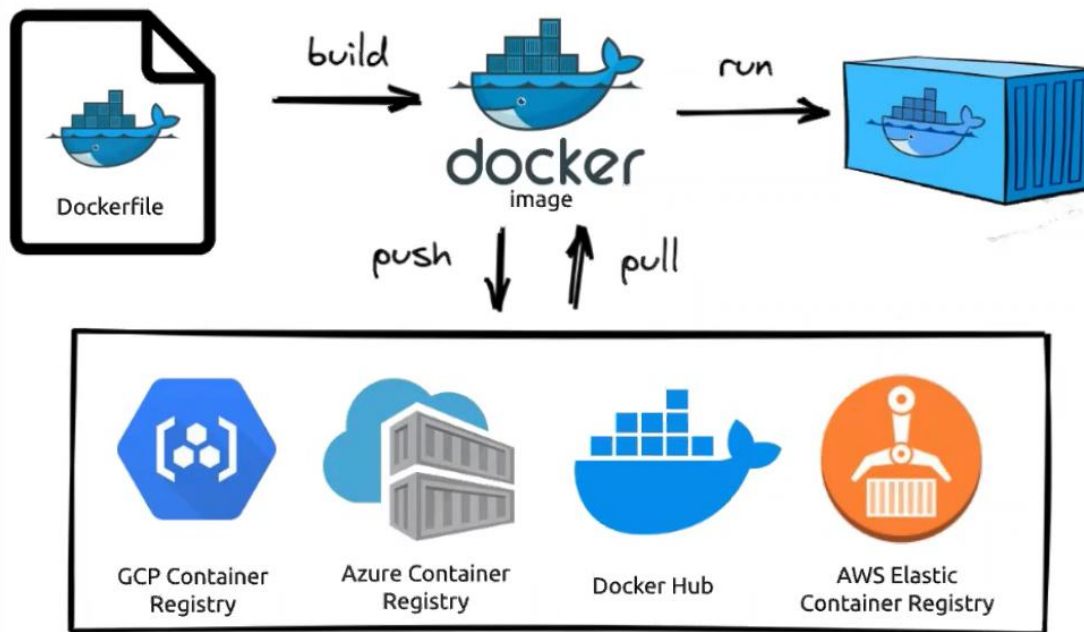
- Docker Hub é o lugar onde ficam armazenados várias imagens para utilizarmos;



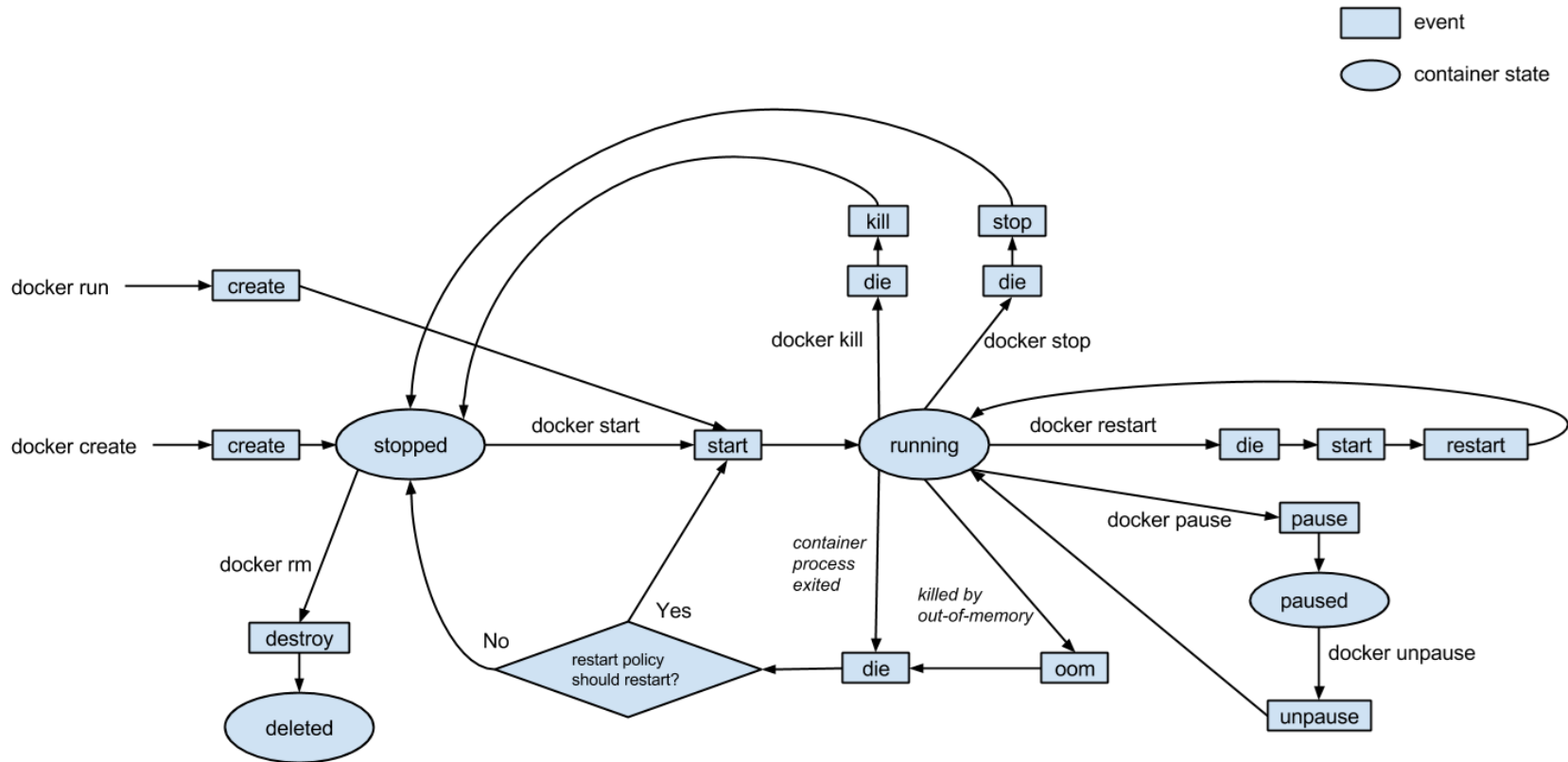
Docker - Fluxo padrão



- Registry é o serviço que hospeda suas imagens de maneira que é possível fazer operações de push e pull



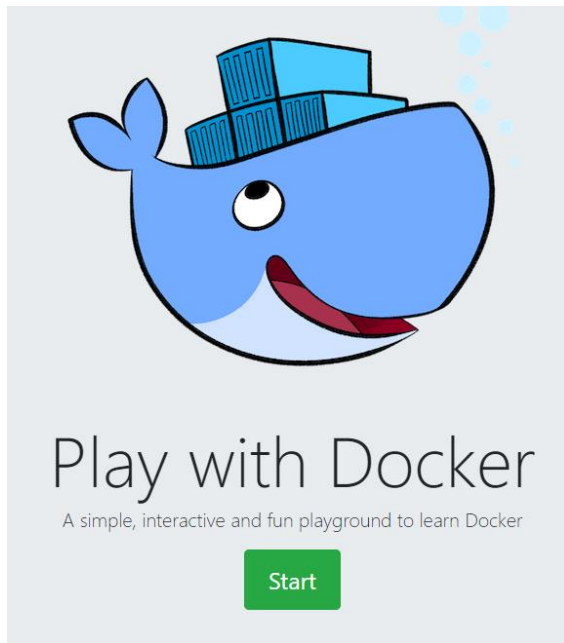
Docker - Ciclo de Vida - Container



Docker - Testando



- **Testando containers na nuvem**
 - Cadastro - <https://hub.docker.com/>
 - Playground - <https://labs.play-with-docker.com/>



Docker - Testando



- **Vamos verificar os containers**
 - `docker ps -a //versão antiga`
 - `docker container ls -a //versão nova`
 - `docker container ps -a //versão nova`
- **Criando um container**
 - `docker container run`
 - `docker container run hello-world`



Docker - Testando



- **Vamos verificar os containers**
 - `docker container run -it`
 - `docker container run -it ubuntu bash`
- **Vamos criar um container com ubuntu**



Docker - Testando



- **Comandos em um container**
 - docker container create ubuntu
 - docker container ps/ls
 - docker container start ID



Docker - Testando



- **Comandos em um container**
 - `docker container run -dit ubuntu`
 - `docker container ls -a`
 - `docker container ls`
 - `docker container stop ID`
 - `docker container ls`
 - `docker container start ID`
 - `docker container ls`
 - `docker container attach ID`



Docker - Testando



- **Comandos em um container**
 - docker container restart ID
 - docker container ls
 - docker container pause ID
 - docker container ls
 - docker container unpause ID
 - docker container ls



Docker - Testando



- **Removendo um container**
 - `docker container rm <hash>`
 - `docker container rm -f <hash>`
 - `docker container prune`



Docker - Atividade



- https://docs.google.com/document/d/1a4X39rwxioZTS_IdTbXbfjQrt0JamW7aE1VvqVZec-o/edit?usp=sharing
- <https://labs.play-with-docker.com/>



Docker - Atividade



- **Faça a criação de um container a partir da imagem ubuntu com o comando run e os parâmetros dit**
 - `docker container run -dit ubuntu`
- **Verifique se o container está rodando**
 - `docker ps`
- **Dê um stop no container**
 - `docker container stop ID`
- **Verifique se foi dado o stop no container**
 - `docker ps -a`



Docker - Atividade



- **Dê o start no container**
 - `docker container start ID`
- **Verifique se o container está rodando**
 - `docker ps`
- **Acesse o container com o attach**
 - `docker container attach ID`
- **Rode o comando ls para verificar as pastas**
 - `ls`



Docker - Atividade



- **Entre na pasta home**
 - `cd home`
- **Crie um arquivo chamado atividade.txt com o texto Hello World**
 - `echo "Hello World" > atividade.txt`
- **Rode ls para ver se criou**
 - `docker container attach ID`
- **Dê um exit do terminal do ubuntu**
 - `exit`



Docker - Atividade



- **Entre novamente no terminal com o comando start e parâmetro i**
 - `docker container start -i`
- **Entre na pasta home**
 - `cd home`
- **Verifique se o arquivo continua lá**
 - `ls`
- **Dê um exit do terminal do ubuntu**
 - `exit`



Docker - Atividade



- **Remova este container em específico**
 - `docker container rm -f <hash>`
- **Elimine todos containers**
 - `docker container prune`
- **Liste todas imagens - EXTRA**
 - `docker images`
- **Remova a imagem do Ubuntu - EXTRA**
 - `docker rmi -f ID`

