



Universidade Federal do Piauí

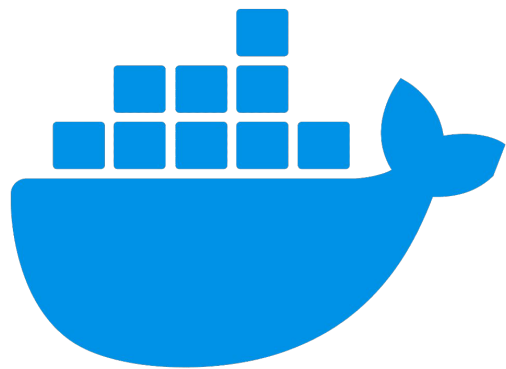
Curso de Sistemas de Informação

Disciplina: Programação Orientada a Objetos II

Apresentador: **Luís Clício**

luisclicio@ufpi.edu.br

github.com/DarkTechLC



docker[®]

Pré-requisitos

É necessário ter o **Docker instalado** na máquina.

Etapas de instalação:

- Windows:
 - <https://docs.docker.com/desktop/install/windows-install/>
- MacOS:
 - <https://docs.docker.com/desktop/install/mac-install/>
- Linux (Ubuntu based):
 - `sudo apt install docker.io docker-compose`
 - Usar sem `sudo`:
<https://docs.docker.com/engine/install/linux-postinstall/>

O que vamos aprender?

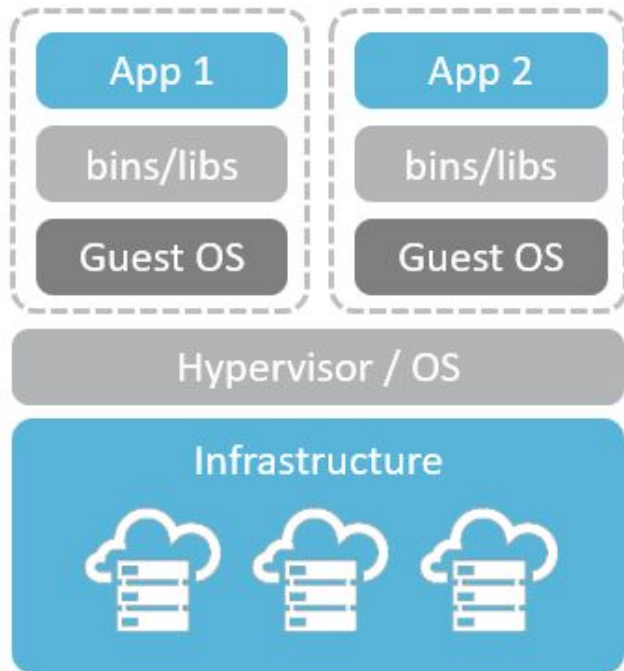
- O que são e como gerenciar imagens
- O que são e como gerenciar containers
- Como persistir os dados dos containers no host
- Como realizar a comunicação entre containers
- Como orquestrar vários containers com Docker Compose

Por que estudar Docker?

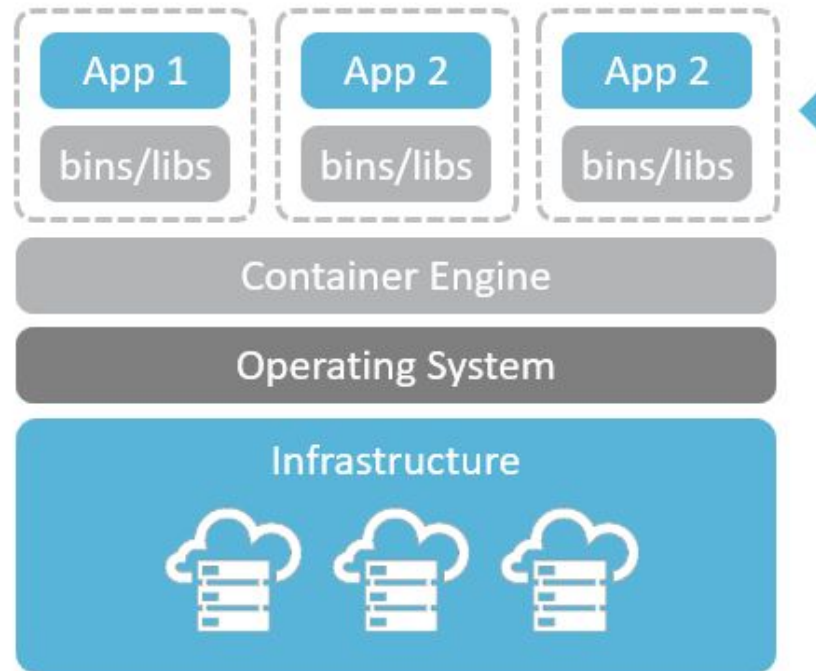
- Maior isolamento entre os componentes da aplicação
- Redução do consumo de recursos em comparação com VMs
- Padronização entre os ambientes de trabalho
- Facilidade para testar, implantar e escalar as aplicações
- **Amplamente utilizado no mercado de trabalho**

Máquinas virtuais X Containers

Virtual Machine Model



Containerized Application Deployment Model



Conceito de imagem e container

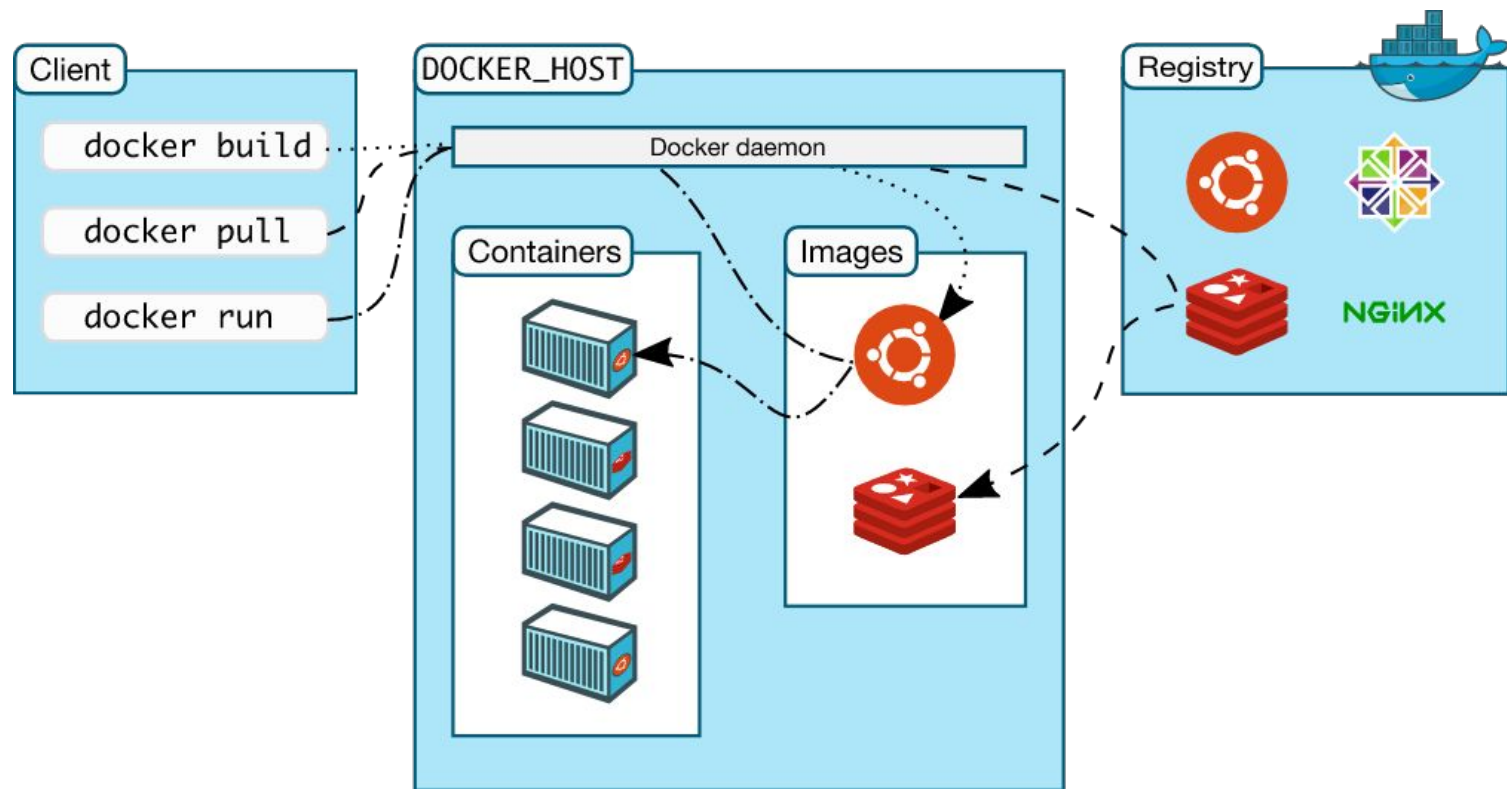
Imagem:

- Conjunto estático de instruções
- Permite a criação de containers
- Customizar uma imagem para criar outra
- Camadas

Container:

- Execução de uma imagem
- Processo isolado no SO
- Vários containers a partir de uma imagem
- Possui acesso a recursos

Arquitetura do Docker



Gerenciando containers

- Execução
 - `docker run [opções] <imagem> [comandos]`
 - `docker start [opções] <container>`
 - `docker stop [opções] <container>`
 - `docker restart [opções] <container>`
- Mapeamento de portas
 - `docker run -p <porta_host>:<porta_container> <imagem>`
- Persistindo dados
 - `docker volume create <nome_volume>`
 - `docker run -v <nome_volume/dir_host>:<dir_container> <imagem>`
- Comunicação
 - `docker network create [opções] <nome_net>`
 - `docker run --network <nome_net> <imagem>`

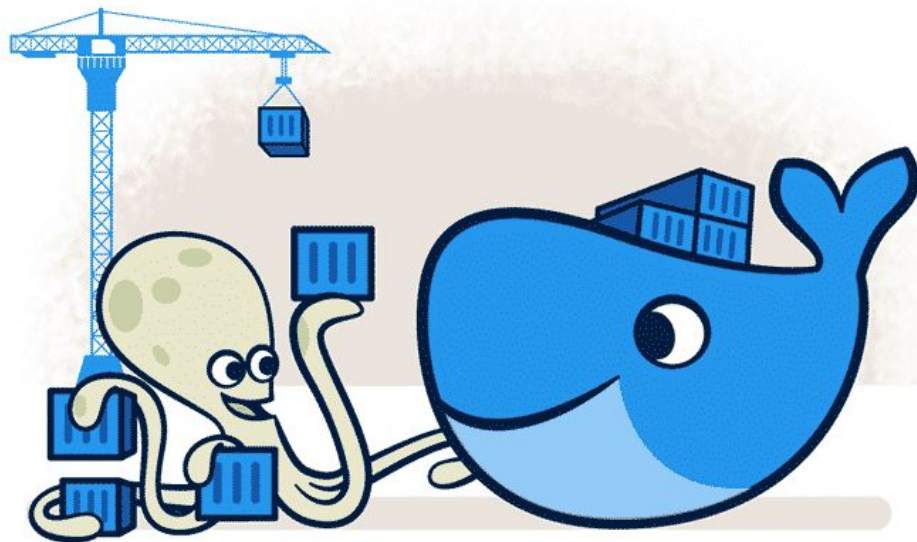
Gerenciando containers

- Listagem
 - `docker ps [opções]`
 - `docker container ls [opções]`
- Logs e consumo de recursos
 - `docker logs [opções] <container>`
 - `docker stats [opções] [<container>]`
- Atualização
 - `docker update [opções] <container>`
- Remoção
 - `docker rm [opções] <container>`
 - `docker container rm [opções] <container>`

Gerenciando imagens

- Obtenção
 - `docker pull [opções] <imagem[:tag]>`
 - Automaticamente ao iniciar um container
- Listagem
 - `docker images [opções]`
 - `docker image ls [opções]`
- Geração
 - `docker build [-t <imagem[:tag]>] [opções] <caminho>`
- Remoção
 - `docker rmi [opções] <images>`
 - `docker image rm [opções] <images>`
 - `docker image prune [opções]`

Vamos brincar...



Docker Compose

O que é e para que serve o Docker Compose?

- Orquestrador de containers
- Permite definir e compartilhar aplicações com múltiplos containers
- Gerenciar e implantar infraestruturas a partir de instruções em arquivos
- Automatiza a geração de imagens e criação de redes e volumes
- Simplifica a configuração de ambientes

Vejamos o uso na prática...

Tarefa

Substitua a conexão do seu aplicativo de banco para que ele utilize um container do MySQL.

Etapas:

- Execute um container a partir da imagem do MySQL:
 - `docker run -d --name mysql-db --restart unless-stopped -p 3300:3306 -e MYSQL_ROOT_PASSWORD=senharoot -e MYSQL_USER=usuario -e MYSQL_PASSWORD=senha -e MYSQL_DATABASE=banco mysql:latest`
- Substitua as variáveis de conexão no seu código.

Fonte: https://hub.docker.com/_/mysql

Dúvidas?

