

Escola de Inverno em Computação

Inscrição gratuita



25 a 29 de julho de 2022

19h às 21h30min | Online




Setrem Faculdade

Automação com contêineres usando Docker





Prof. Dinei A. Rockenbach



Apresentação

Formação

- Técnico em Informática **Setrem**
- Bacharel em Sistemas de Informação **Setrem**
- Mestre em Ciência da Computação 

Experiência

- WT Agência 
- Migrate Company 
- Setrem 
- Tribunal Regional do Trabalho da 4ª Região 

Laboratórios de Pesquisa:  e 



/dineiar



@dineiar26



/darockenbach

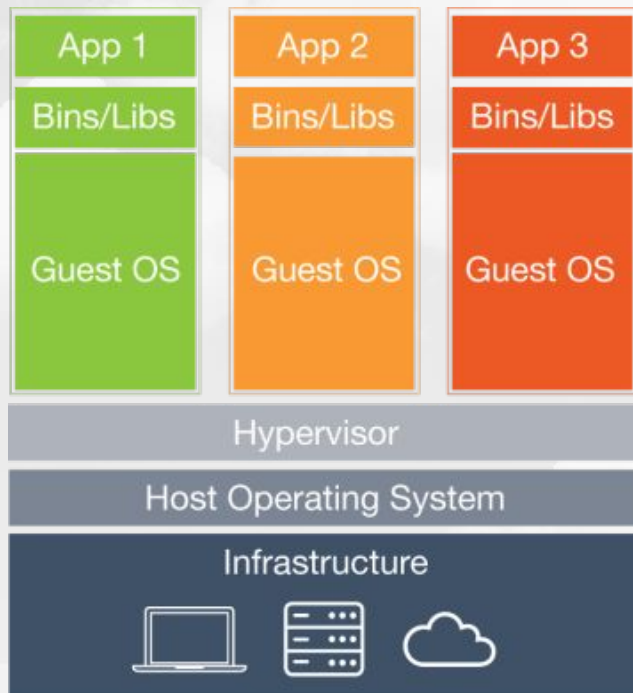
Uma breve introdução à Virtualização

- Como fazíamos as coisas?
 - Pessoas
 - Empresas
- Como a virtualização com máquinas virtuais ajuda?
 - Isolamento de ambientes

Então as máquinas virtuais resolvem todos os problemas do mundo?

- A virtualização oferece isolamento, mas tem certo custo envolvido (em relação ao desempenho)
 - E os pesquisadores?
- Cada máquina virtual possui seu próprio sistema operacional completo, rodando sobre o *hypervisor* (software de virtualização)
- The Twelve-Factor App - <https://12factor.net/>

Máquinas Virtuais

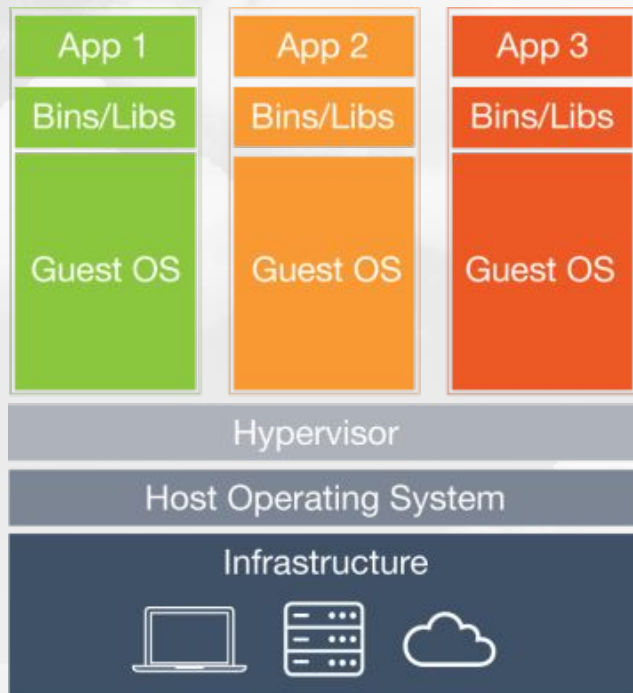


E o que é o Docker?

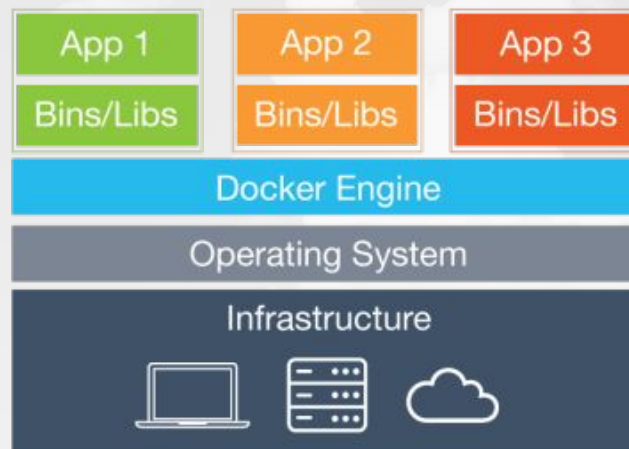
- Virtualização a nível de sistema operacional: containers.
- Isolamento é fornecido pelo kernel Linux
- Como não precisam carregar um sistema operacional, são muito mais leves, rápidos, e versáteis

Leia mais: <https://www.ibm.com/cloud/learn/docker>

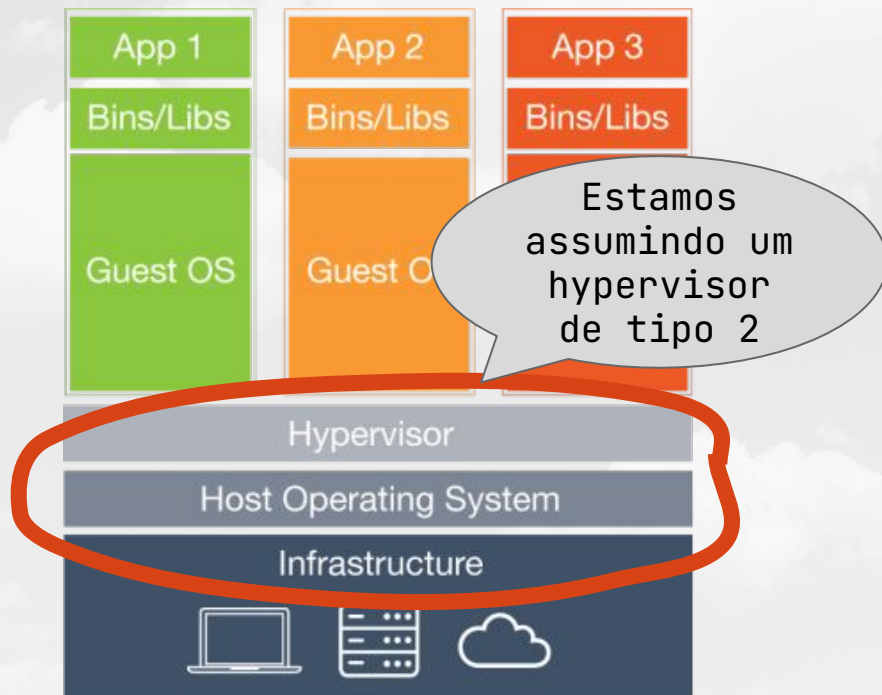
Máquinas Virtuais



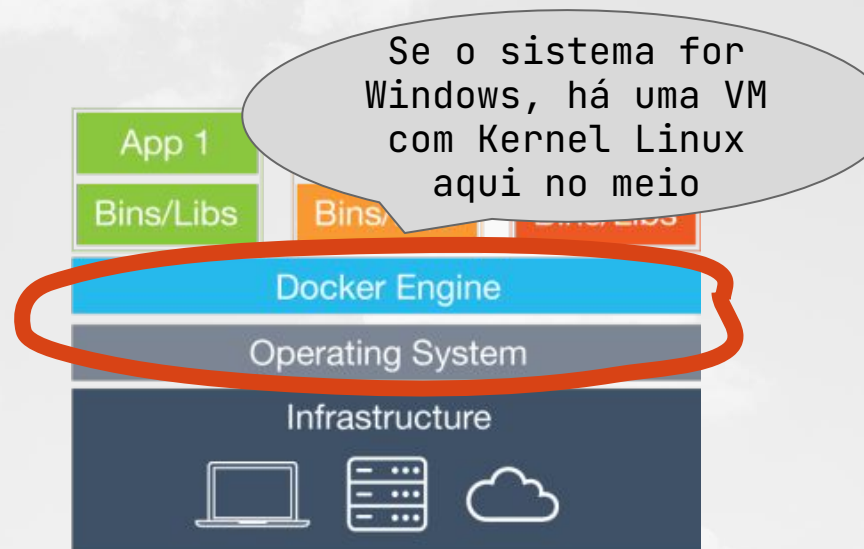
Containers



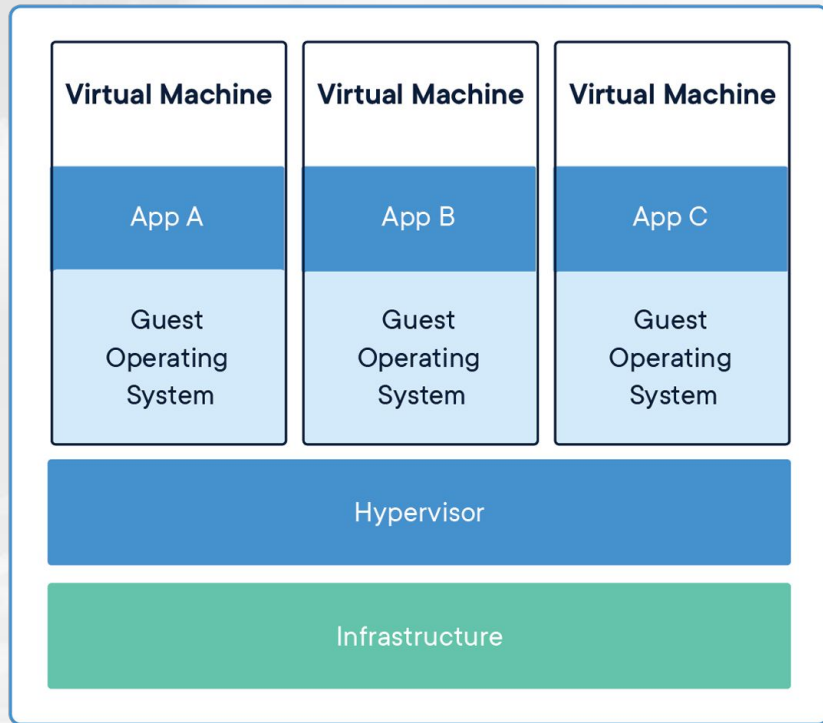
Máquinas Virtuais



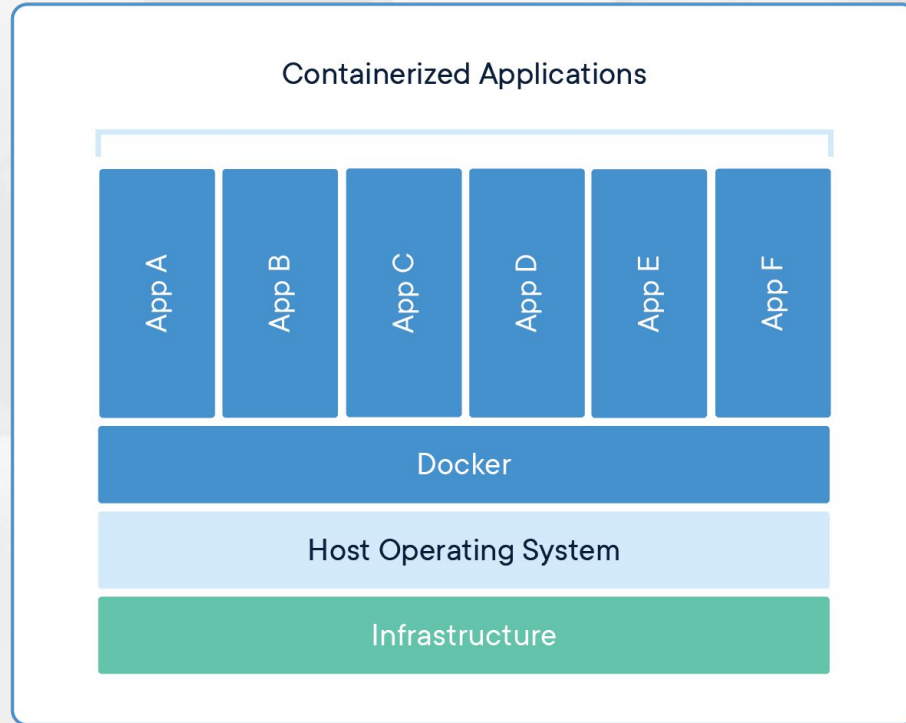
Containers



Máquinas Virtuais



Containers

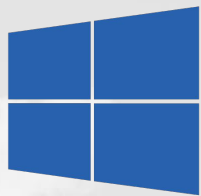


Instalação

Linux

- Ubuntu 18+ 64bits / Debian / CentOS / Fedora
- Instalação:
 - a. Definir repositório e chave GPG (ver orientações específicas)
 - b. `sudo apt install docker-ce docker-ce-cli containerd.io`
- Orientações específicas:
 - a. <https://docs.docker.com/engine/install/ubuntu/>
 - b. <https://docs.docker.com/engine/install/debian/>
 - c. <https://docs.docker.com/engine/install/centos/>
 - d. <https://docs.docker.com/engine/install/fedora/>

Windows



- www.docker.com/products/docker-desktop
- Recomendação: habilitar WSL 2
 - *Windows Subsystem for Linux (WSL) 2 introduces a significant architectural change as it is a full Linux kernel built by Microsoft, allowing Linux containers to run natively without emulation.*
<https://docs.docker.com/docker-for-windows/wsl/>

Containers / Apps

Images

Volumes

Dev Environments PREVIEW


docker-larcc-escola-inverno

C:\Users\dinei\Projects\docker-larcc-escola-inverno

Open in Visual Studio Code



CONTAINERS



larcc-ei-db sha256:5c62e45...

RUNNING PORT: 3306



larcc-ei-app sha256:cebc80a...

RUNNING PORT: 80

```
larcc-ei-db | 2021-07-13 04:29:29+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL
Server 8.0.25-1debian10 started.
larcc-ei-db | 2021-07-13 04:29:29+00:00 [Note] [Entrypoint]: Switching to dedicated user
'mysql'
larcc-ei-db | 2021-07-13 04:29:29+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL
Server 8.0.25-1debian10 started.
larcc-ei-db | 2021-07-13T04:29:29.764978Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld
(mysqld 8.0.25) starting as process 1
larcc-ei-db | 2021-07-13T04:29:29.780274Z 1 [System] [MY-013576] [InnoDB] InnoDB
initialization has started.
larcc-ei-db | 2021-07-13T04:29:30.014211Z 1 [System] [MY-013577] [InnoDB] InnoDB
initialization has ended.
larcc-ei-db | 2021-07-13T04:29:30.151626Z 0 [System] [MY-011323] [Server] X Plugin ready
for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock
larcc-ei-db | 2021-07-13T04:29:30.313381Z 0 [Warning] [MY-010068] [Server] CA certificate
ca.pem is self signed.
larcc-ei-db | 2021-07-13T04:29:30.313617Z 0 [System] [MY-013602] [Server] Channel
mysql_main configured to support TLS. Encrypted connections are now supported for this
channel.
larcc-ei-db | 2021-07-13T04:29:30.324545Z 0 [Warning] [MY-011810] [Server] Insecure
configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all
OS users. Consider choosing a different directory.
larcc-ei-db | 2021-07-13T04:29:30.350301Z 0 [System] [MY-010931] [Server]
/usr/sbin/mysqld: ready for connections. Version: '8.0.25' socket:
'/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
```



Search...



Stick to bottom



Criando containers e usando imagens Docker

Docker Hub

- hub.docker.com
- Várias imagens pré-construídas já disponíveis
- Completamente grátis para usuários “comuns”
- Facilmente alimentado pela própria comunidade quando as organizações não fornecem as imagens

Imagem:Tag

`ubuntu:20.10`

Nome da
Imagem

Tag

`python:3.9.6-alpine`

Nome da
Imagem

Tag

- Caso não seja especificada, por padrão o Docker assume a tag “latest”
 - Há algumas discussões a respeito dessa tag
- Podem ser definidas várias tags para uma mesma imagem

Comandos para usar containers Docker

docs.docker.com/engine/reference/commandline/container

- `docker container run <imagem>`
 - Cria e executa um novo container da imagem especificada
- `docker container ls`
 - Lista os containers em execução, use -a para listar containers parados também
- `docker container exec`
 - Executa um comando em um container em execução
- `docker container cp`
 - Copia um arquivo entre o container e a máquina hospedeira (host)
- `docker container start / stop / kill / rm`
 - Inicia / pára / "mata" / exclui um container já criado

labs.play-with-docker.com

Prática: Python+Flask

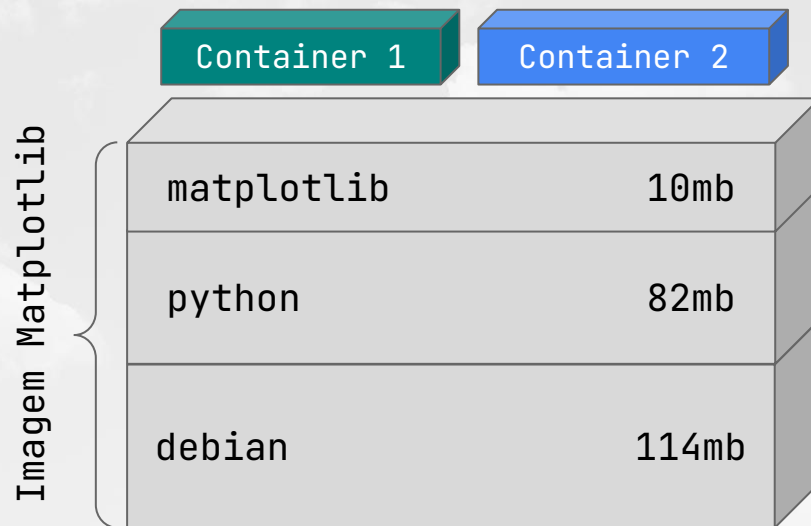
- Vamos usar a imagem do Python do hub.docker.com
 - `python:3.7-alpine` (45mb)
- Vamos usar o Exercício 1 em github.com/larcc-group/escola-inverno-2022-docker



Construindo imagens Docker

Imagem ou Container?

- Uma imagem Docker é construída em camadas
 - que são reaproveitadas
- Um container é apenas uma camada “alterável” sobre as camadas (fixas) da imagem



Definindo uma imagem: Dockerfile

- É um arquivo de texto com comandos que definem o passo-a-passo para construir uma imagem
- docs.docker.com/engine/reference/builder

- Exemplo:

```
FROM python:3.7-alpine
RUN pip install flask
COPY app.py /home
```

Comandos para construir imagens Docker

docs.docker.com/engine/reference/commandline/image

- `docker image ls`
 - Lista as imagens no repositório local
- `docker image build <pasta> -t <tag>`
 - Constrói uma imagem a partir de um Dockerfile
- `docker image rm <imagem>`
 - Remove uma imagem do repositório local
- `docker image pull <imagem>`
 - Baixa uma imagem de um repositório remoto (é implícito no run)
- `docker image push <imagem>`
 - Envia uma imagem construída localmente para um repositório remoto

labs.play-with-docker.com

Prática: Python+Flask

- Vamos usar a imagem do Python do hub.docker.com
 - `python:3.7-alpine` (45mb)
- Vamos usar o Exercício 1 em github.com/larcc-group/escola-inverno-2022-docker



Usando Docker Compose

O que é Docker Compose?

- “Compose é uma ferramenta para definir e rodar aplicações Docker multi-contêineres. [...] com um único comando, você cria e inicia todos os serviços da sua aplicação.”
- O Compose faz a orquestração (execução, conexão, gerenciamento) de múltiplos contêineres que trabalham juntos e se comunicam.

Leia mais: <https://docs.docker.com/compose/>

Definindo a configuração do Compose

- É um arquivo YAML com a configuração dos contêineres
- docs.docker.com/compose/compose-file

- Exemplo:

```
services:
```

```
  web:
```

```
    image: php:8.0-apache
```

```
  banco:
```

```
    image: mysql:8.0
```

Comandos para usar o Docker Compose

docs.docker.com/compose/reference

- **docker compose up**
 - Constrói, (re)cria e inicia os serviços. Use -d para iniciar em segundo plano.
 - Use --build para forçar a compilação da imagem
- **docker compose down**
 - Pára e destrói os recursos (exceto volumes permanentes) criados pelo comando up
- **docker compose stop**
 - Pára os serviços iniciados pelo comando up -d

labs.play-with-docker.com

Prática: usando Docker Compose

Website (Flask) + Banco (Redis)

- Vamos construir uma imagem própria para o site, e usar a imagem do Redis do hub.docker.com
 - `redis:7.0-alpine` (28mb)
- Vamos usar o Exercício 2 em github.com/larcc-group/escola-inverno-2022-docker



Extra: GitHub Packages

- Basicamente um “Docker Hub” integrado ao GitHub
- Na prática permite publicar não só imagens Docker, mas também pacotes NPM, Maven, Nuget, e RubyGems.
- Os pacotes publicados podem ser vinculados ao repositório
- <https://github.com/features/packages>



Recursos extras

- docker-unleashed.readthedocs.io - Curso gratuito em português
- www.docker.com/101-tutorial - Tutorial “oficial” de Docker
- `docker container run -p80:80 docker/getting-started`
- labs.play-with-docker.com - Teste de comandos Docker online
- [en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)) - Página da Wikipedia

Escola de Inverno em Computação

Inscrição gratuita



25 a 29 de julho de 2022

19h às 21h30min | Online



Setrem Faculdade

Obrigado pela atenção!

Prof. Dinei A. Rockenbach



/larcc-group



@larcc_group



/larcc-setrem



@larcc.lab



@larcc.group