VIRT - Treinamento da Plataforma de Virtualizacao Xen/Docker

Agnaldo N. Marinho
Gabriel Silva

http://github.com/agnaldom

August 30, 2017

Docker Containers

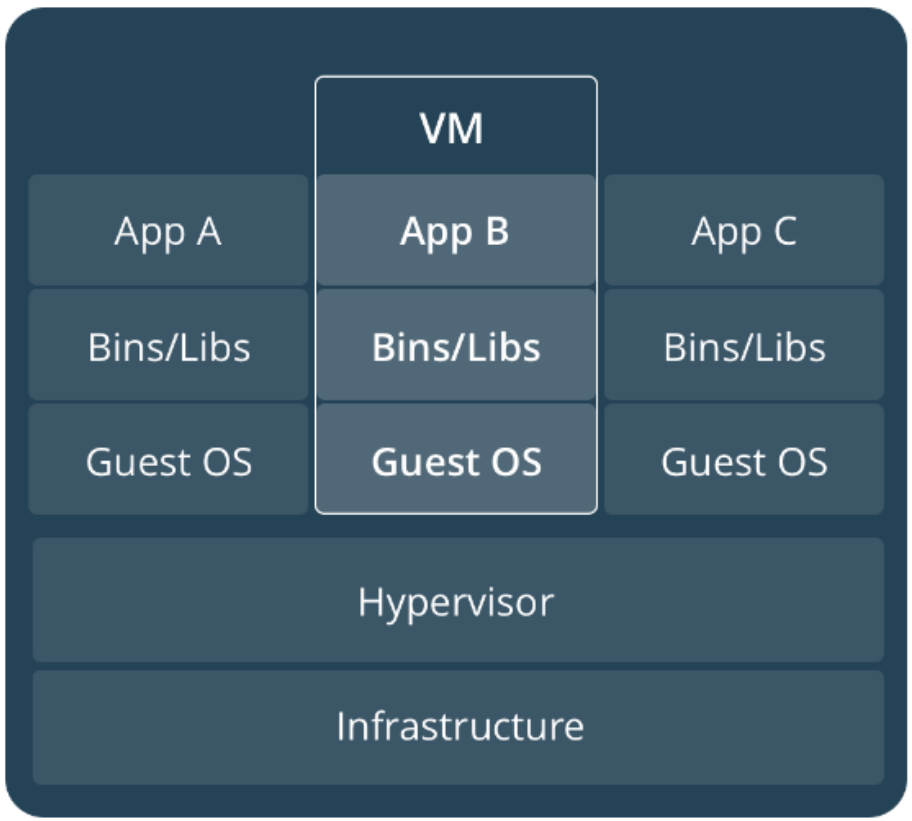
Introdução ao docker

`Docker` é uma plataforma aberta, criada com o objetivo de facilitar o desenvolvimento, a implatação e a execução de aplicações em ambientes isolados.
Usando o Docker, você pode facilmente gerenciar a infraestrutura da aplicação, isso agilizará o processo de criação, manutenção e modificação do seu serviço.

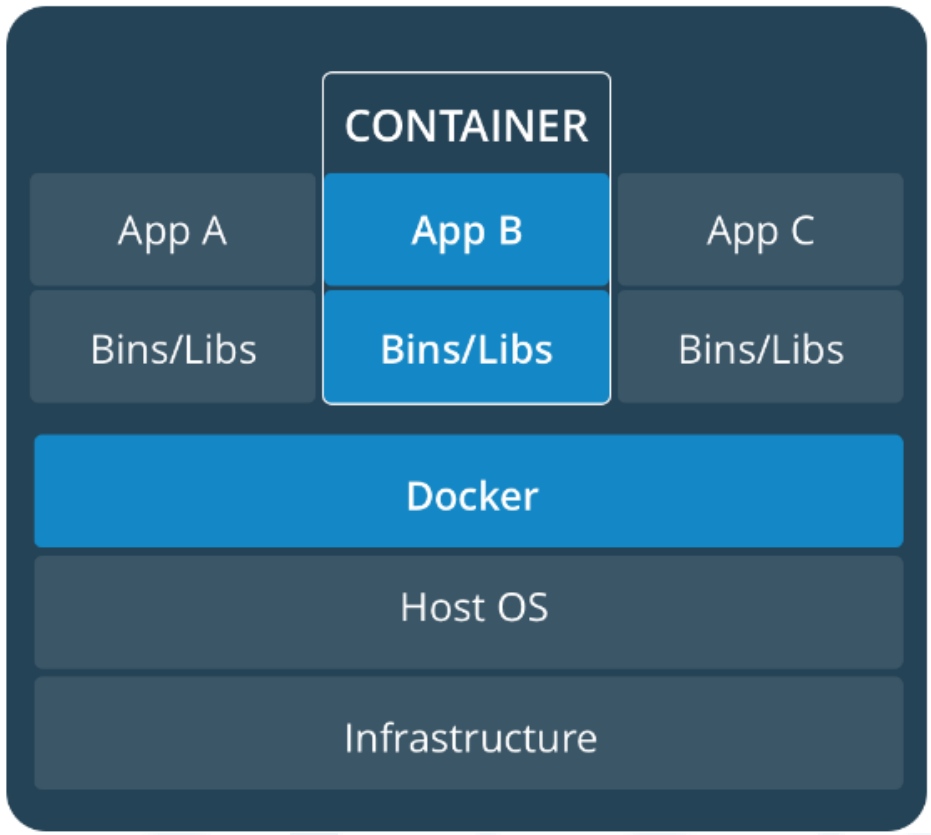
Docker Containers

Container vs Virtual machines

Virtual Machine diagram



Container diagram



Docker Containers

Instalação Docker

- Docker está disponivel em duas edições: Community Edition (CE) e Enterprise Edition (EE).
- Docker (CE) é ideial para desenvolvedores e pequena equipes.
- Docker (EE) é voltado para times de desenvolvimento e TI das empresas.

No nosso caso vamos usar o (CE), e para versão do debian stretch.

Docker Containers

Inciando o seu primeiro container

Depois da instalação, vamos testar a instalação rodando o seguinte:

```
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
03f4658f8b78: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:8be990ef2aeb16dbcb9271ddfe2610fa6658d13f6dfb8bc72074cc
Status: Downloaded newer image for hello-world:latest

Hello from Docker.
This message shows that your installation appears to be working corre
...
```

Docker Containers

Comandos básicos

Para iniciar um container é necessário saber a partir de qual imagem será executado. Para lista as images que o seu Docker Host tem localmente, execute o comando abaixo¿

1. docker image list

Baixando a imagem

1. docker image pull debian

Baixamos uma imgem do debian, caso deseje inspecionar a imagem que acabou de atualizar, bastar usar o comando abaixo:

1. docker image inspect debian

O comando inspect é responsável por informa todos os dados referentes à imagem.

Docker Containers

Criando sua própria imagem no Docker

Uma imagem nada mais é do que um ambiente totalmente encapsulado e pronto para ser replicado onde desejar. Há duas formas de criar images customizadas: com commit e com Dockerfile.

- Criando images com commit

Primeiro criamos um container :

1. docker run -it –name dockercurso-debian debian:latest /bin/bash

Agora que estamos no bash do container, instalamos o nginx:

1. apt-get update
2. apt-get install nginx -y
3. exit

Paramos o container com o comando abaixo:

1. docker container stop dockercurso-debian

Docker Containers

Criando sua própria imagem no Docker

Agora, efetuamos o commit desse container em uma imagem:

1. docker container commit dockercurso-debian dockercurso-nginx

Para visualizar a lista de imagens e encontrar a que acabou de criar, execute novamente o comando abaixo:

1. docker image list

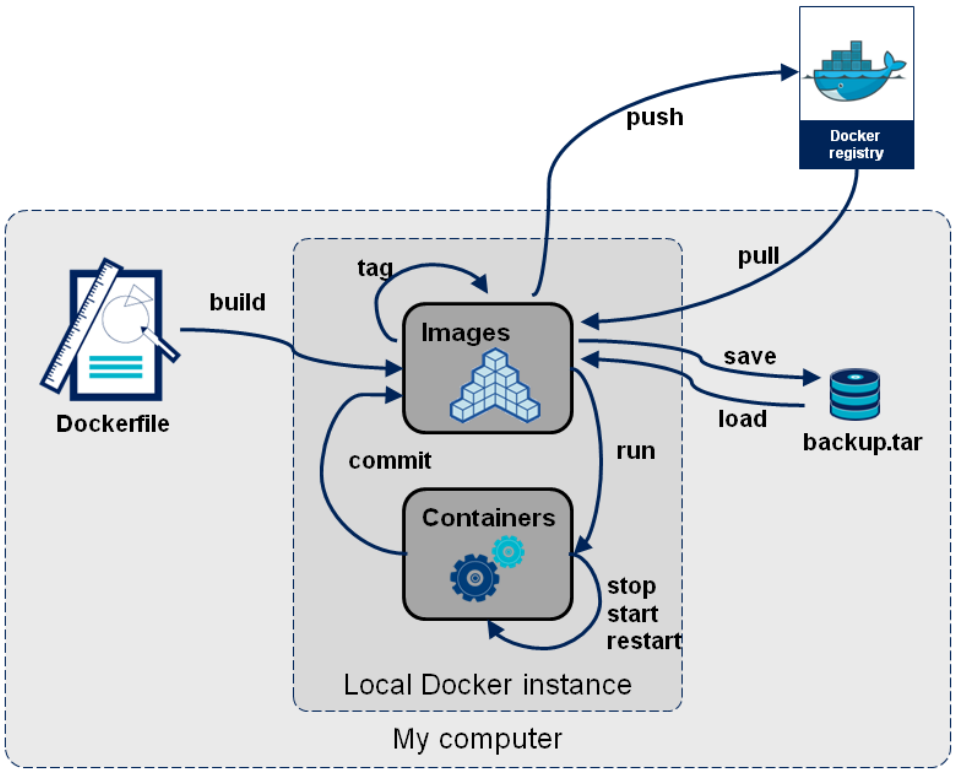
Para testar sua nova image, vamos criar um container a parti dela e verificar se o nginx está instalado:

1. docker run -it –rm dockercurso-nginx dpkg -l nginx

Docker Containers

Criando imagens com Dockerfile

O docker permite que possamos criar images a partir de um arquivo de definição, esse arquivo chama-se Dockerfile. Em resumo, o Dockerfile um arquivo texto com instruç oes, comandos e passos que você executaria manualmente, basicamente o Docker executa uma receita de bolo.



Docker Containers

Criando imagens com Dockerfile

Através do comando docker build, o Docker realizar a execução desses passos e no fina da execução ele encapsula cada layer gerada dentro da imagem.
O Dockerfile deve seguir uma ordem ou formatação correta para que o build seja feito de forma certa.
Exemplo:
RUN apt-get update
onde:
RUN É a instrução;
apt-get update: Argumento que será executado.
Dockerfile Referência Oficial

Docker Containers

Criando imagens com Dockerfile

Docker Containers

Criando imagens com Dockerfile

Docker Containers

Exportação de containers

Imagine que temos um container em execução e queremos exportá-lo para outro host.
Podemos criar uma imagem partindo de um container que está funcionando, e gerar um arquivo .tar usando a opção save. Como mostrar abaixo:

1. docker ps -q
2. b02af9430141
3. docker commit b02af9430141 dockercurso-nginx
4. 9a8c0a1a72d2f9c2815e455a22be91f9cf788f769d2cca5b603bae

De posse da nova imagem que foi gerda.

1. docker save dockercurso-nginx > /tmp/dockercurso-nginx.tar

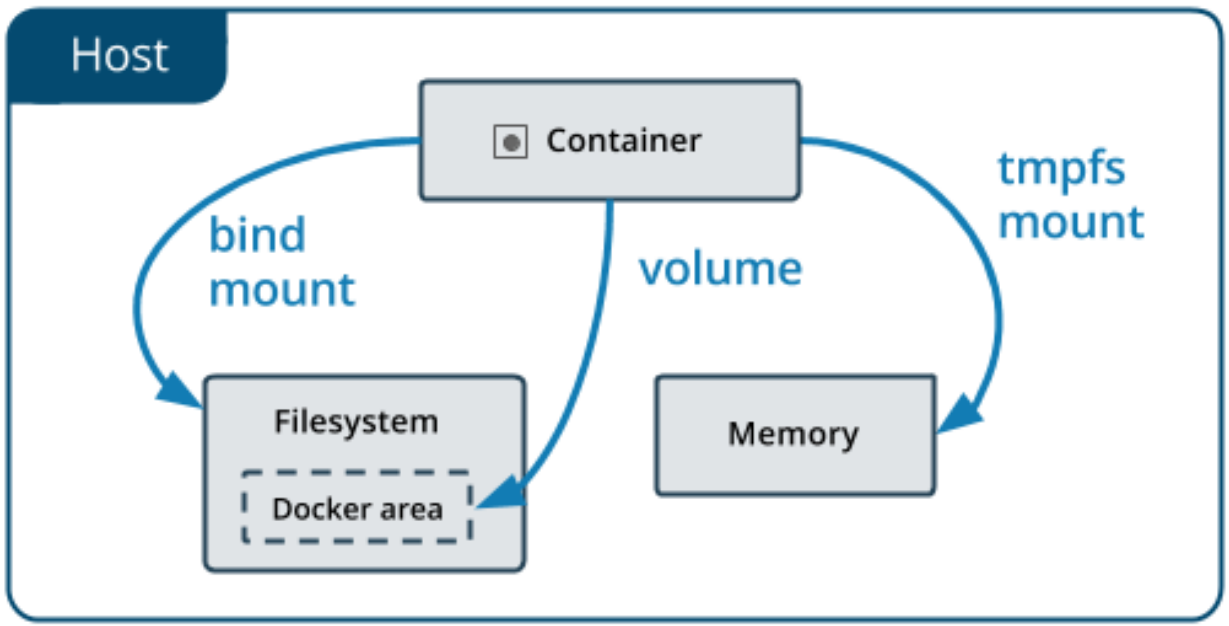
Para fazer a importação desse arquivo, utilizamos a opção load.

1. docker load < /tmp/dockercurso-nginx.tar

Docker Containers

Gerenciando dados em Docker

O Docker oferece três formas diferentes de montar dados em um container para um Docker Host: Volumes, bind mounts, ou tmfs volumes.



Docker Containers

Gerenciando dados em Docker

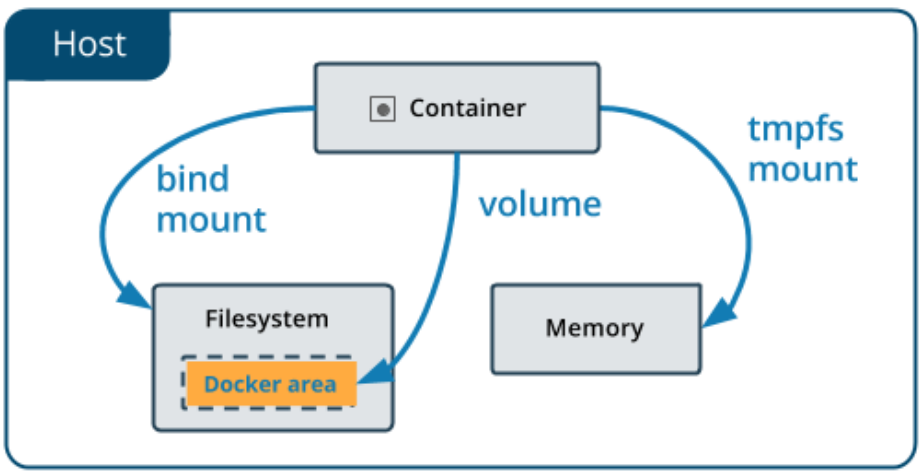
- Volumes: são armazenados em uma parte do sistem de arquivos do host que é gerenciado pelo docker (/var/lib/docker/volumes/ no linux). Os processos não Docker não devem modificar esta parte do sistema de arquivos.
- Bind Mounts: podem ser armazenadas em qualquer lugar no sistema host. Eles podem até ser importantes arquivos ou diretórios do sistema. Os processos não Docker no host ou um contêiner Docker podem modificá-los a qualquer momento.
- tmfs mounts: são armazenados somente na memória do sistema host e nunca são escritas no sistema de arquivos do host.

Docker Containers

Dados em Docker Volumes

Os Volumes têm várias vantagens sobre bind mounts:

- Os volumes são mas fáceis de fazer backup ou migrar do que os bind mounts.
- Você pode gerenciar volumes usando os comando Docker CLI ou a API Docker.
- Os volumes funcionam em recipiente Linux e Windows.
- Os controladores de volume permitem armazenar volumes em hosts remotos ou provedores de nuvem, criptografar o conteúdo do volumes ou adcionar outros funcionanalidades.
- O conteúdo do um novo volume pode ser pré-preenchido por um container.



Docker Containers

Usando Volumes

Originalmente, o flag -v ou –volumes foi usado para contêines independentes e o –mount foi usado para serviços do swarm, a parte da versão 17.06, você também pode usar –mount para contêiner independente.

Criando volume:

1. docker volume create site

Listando Volumes:

```
agnaldoneto@moju ~ % docker volume ls
DRIVER              VOLUME NAME
local               site
```

Inspecionando o volume:

```
agnaldoneto@moju ~ % docker volume inspect site
[
    {
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/site/_data",
        "Name": "site",
        "Options": {},
        "Scope": "local"
    }
]
```

Removendo volume:
docker volume rm site
Documentação Oficial

Docker Containers

Docker Compose v3

Docker Containers

Docker Compose

Usar o Compose é basicamente um processo de três passos:

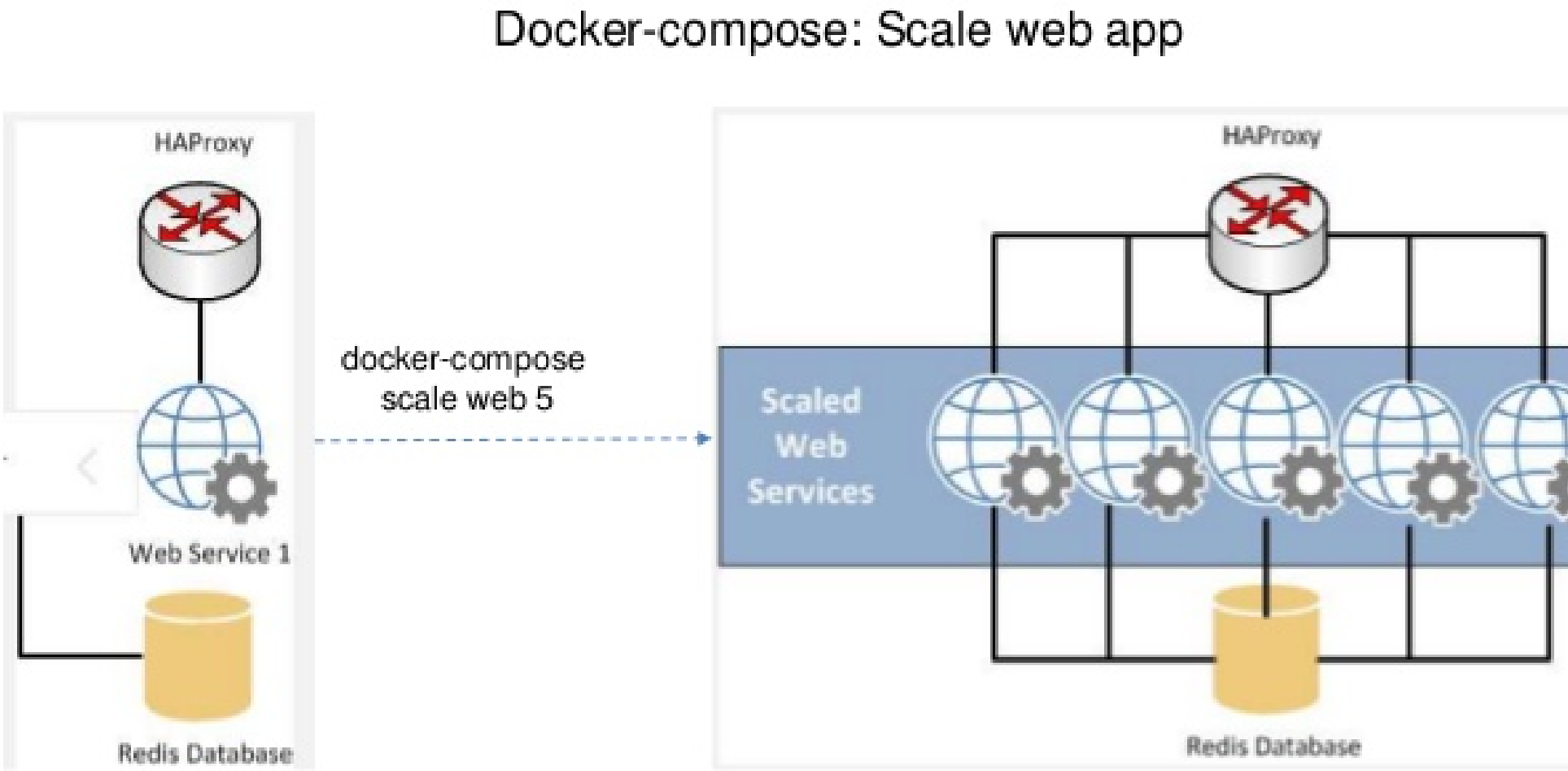
1. Defina o ambiente do seu aplicativo com um Dockerfile para que ele possa ser reproduzido em qualquer lugar.
2. Defina os serviços que compõem seu aplicativo no docker-compose.yml para que eles possam ser executados juntos em ambiente isolado.
3. Por fim, execute docker-compose up e Compose irá iniciar e executa o aplicativo inteiro.

Docker Containers

Docker Compose Joomla

Docker Containers

Escalando Containes, usando Docker-compose



License

Get the source of this theme and the demo presentation from

`http://github.com/famuvie/beamerthemesimple`