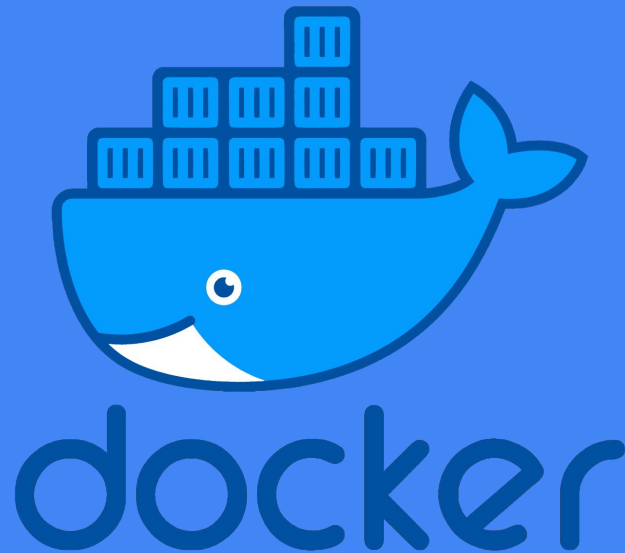


Workshop DOCKER

Conceitos + aplicação prática
utilizando Docker Compose



Thalyson A. Rodrigues
Desenvolvedor Web/Mobile



O que é Docker?

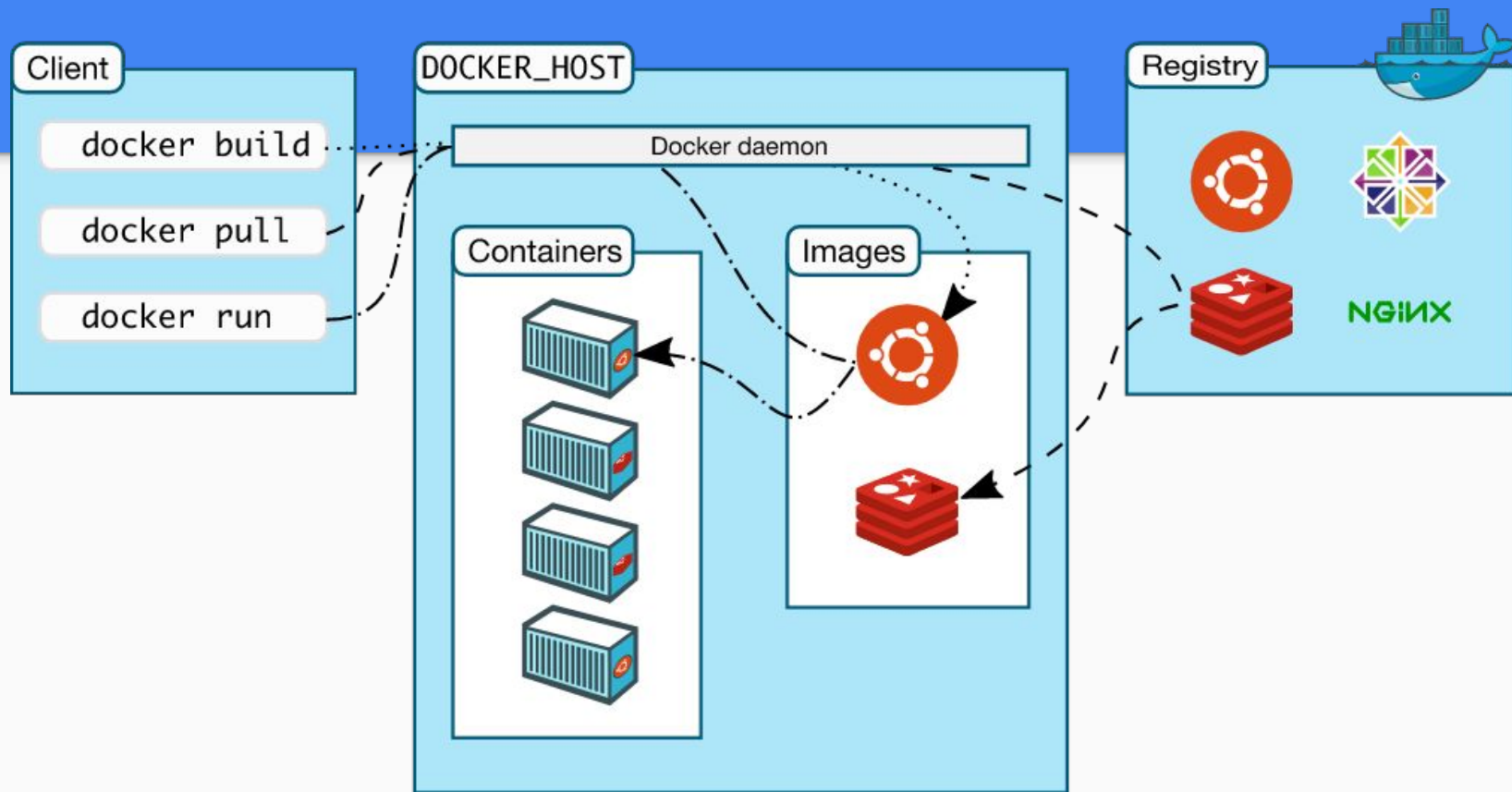
Docker é uma plataforma aberta escrita com a linguagem Go para desenvolvimento, envio e execução de apps. Com Docker você pode gerenciar sua infraestrutura da mesma forma como gerencia seus apps, diminuindo significativamente o atraso de escrever código e implantá-los em produção.

Okay... mas e como funciona?

O Docker oferece a capacidade de empacotar e executar um aplicativo em um ambiente vagamente isolado denominado **contêiner**. O isolamento e a segurança permitem que você execute vários contêineres simultaneamente em um determinado host. Os contêineres são leves e contém tudo o que é necessário para executar o aplicativo, portanto, você não precisa depender do que está instalado atualmente no host. Você pode **compartilhar** contêineres facilmente enquanto trabalha.

O contêiner se torna sua unidade de distribuição e teste.

Arquitetura Docker



Palavras-chave no contexto Docker:

Registry: registro de armazenamento de imagens Docker;

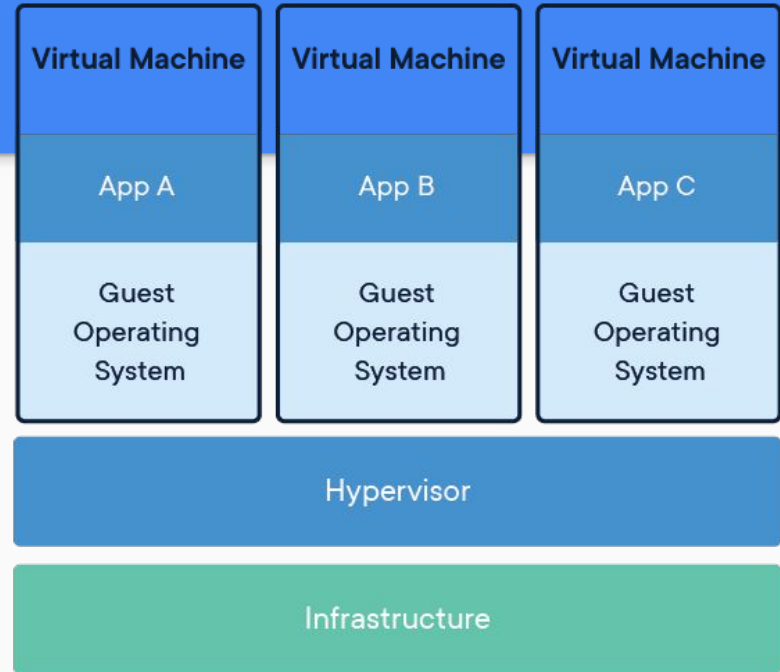
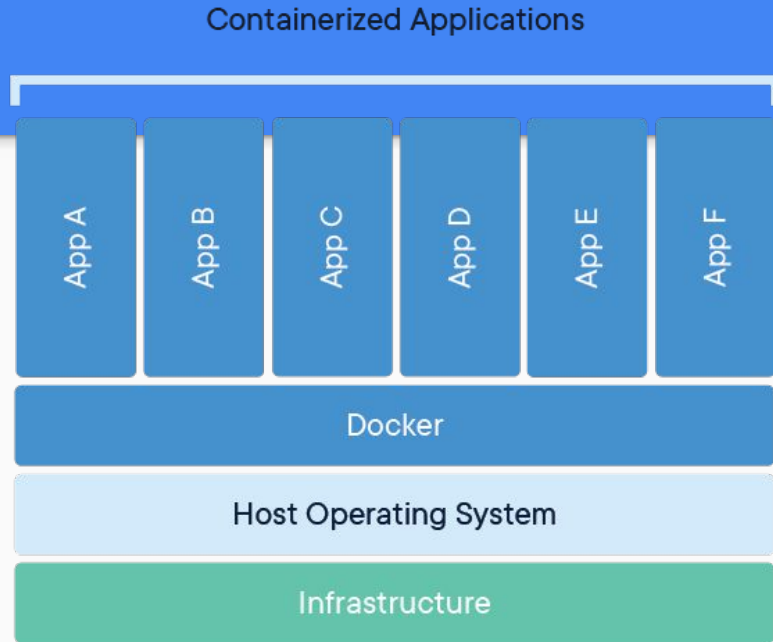
Imagens: é a abstração da definição do nosso container composto por sistema de arquivos, configurações, variáveis de ambiente, tecnologias, etc;

Containers: é a instância em execução de nossa imagem;

Docker CLI: Interface de linha comando Docker utilizada para realizar interações e gerenciar nossos containers e imagens;

Docker Daemon: API que executará nossas solicitações da CLI.

Contêineres vs VM's



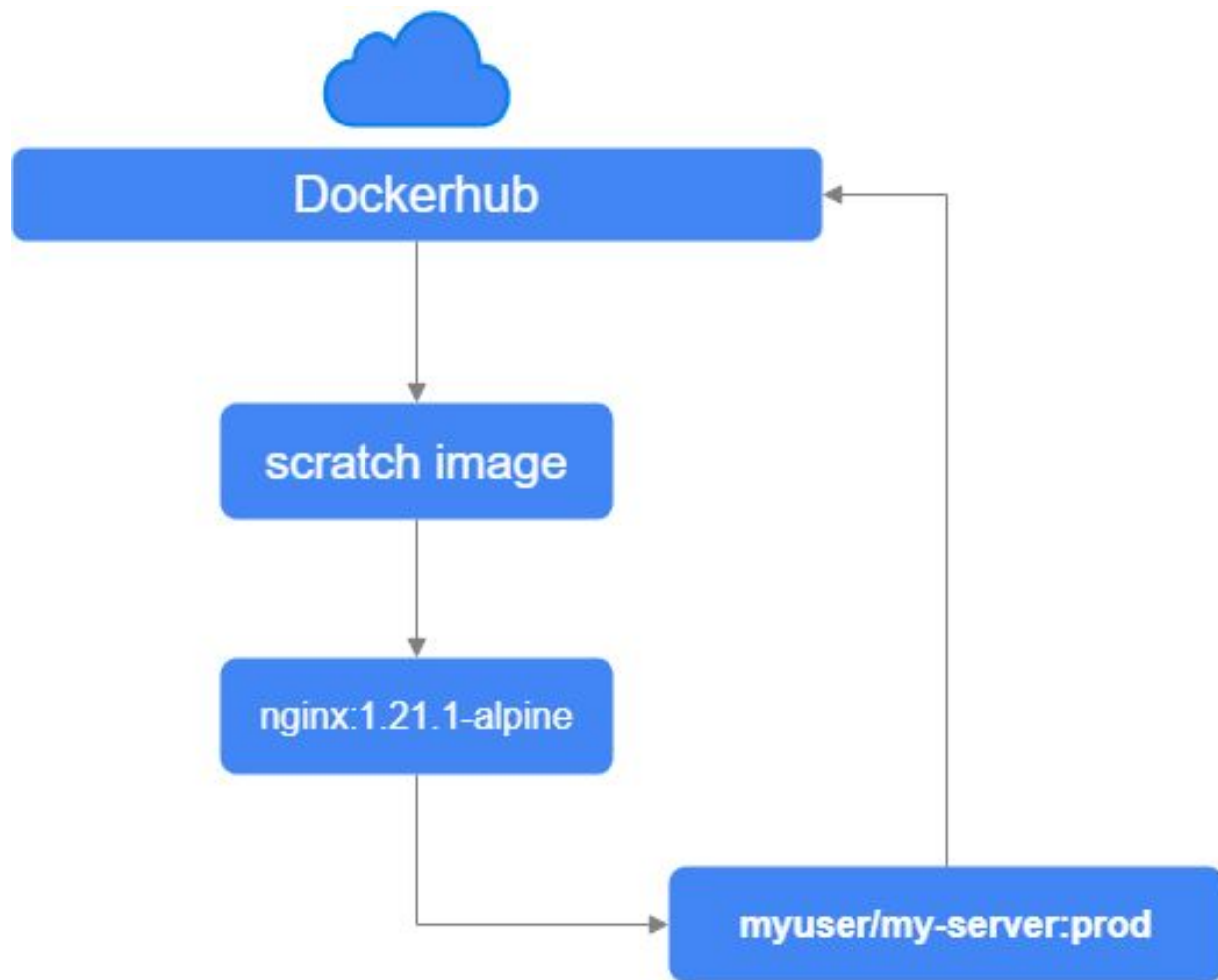
<https://www.redhat.com/pt-br/topics/containers/containers-vs-vms>

<https://www.docker.com/resources/what-container>

E como construir contêineres no Docker?

A partir de imagens já existentes encontradas em:

- **Dockerhub** (Imagens oficiais e da comunidade Docker);
- **GHCR** (Github Container Registry);
- **ECR** (Elastic Container Registry AWS);
- Outros...



As imagens são seguras?

docker scan [args]

O Docker oferece imagens oficiais e imagens construídas pela comunidade. Assim como em todo canto existem pessoas má intencionadas e algumas destas imagens da comunidade podem estar sim infectadas. Uma das ferramentas que o Docker traz consigo é o comando **scan**, o qual permite analisar as vulnerabilidades de uma imagem docker construída.

Limitações da versão Free

docker pull [args]

Ao precisar realizar muitos **pulls** de imagens o Docker poderá limitar a suas requisições (entre 100 a 200).

Limitações de apenas 1 repositório privado por conta.

Também será limitado a quantidade de **scans** para imagens com que deseja analisar as vulnerabilidades.

Outros detalhes podem ser vistos em:

<https://www.docker.com/pricing>

Alguns comandos do Docker CLI:

- **docker ps [args]:** lista os *containers* em execução;
- **docker image ls [args]:** lista imagens baixadas localmente;
- **docker build [args]:** constrói uma imagem a partir de um Dockerfile;
- **docker stats [args]:** mostra as estatísticas de recursos utilizados por *containers*;
- **docker run [args]:** executa um *container* e faz download da imagem caso não exista localmente;
- **docker stop [args]:** interrompe um ou mais *containers* em execução;
- **docker start [args]:** inicia um novo *container*;
- **docker pull [args]:** faz o download de uma imagem docker;
- **docker rm [args]:** remove um ou mais *containers*;
- **docker rmi [args]:** remove uma imagem criada caso não contenha dependências (*containers* criados).

Docker Compose

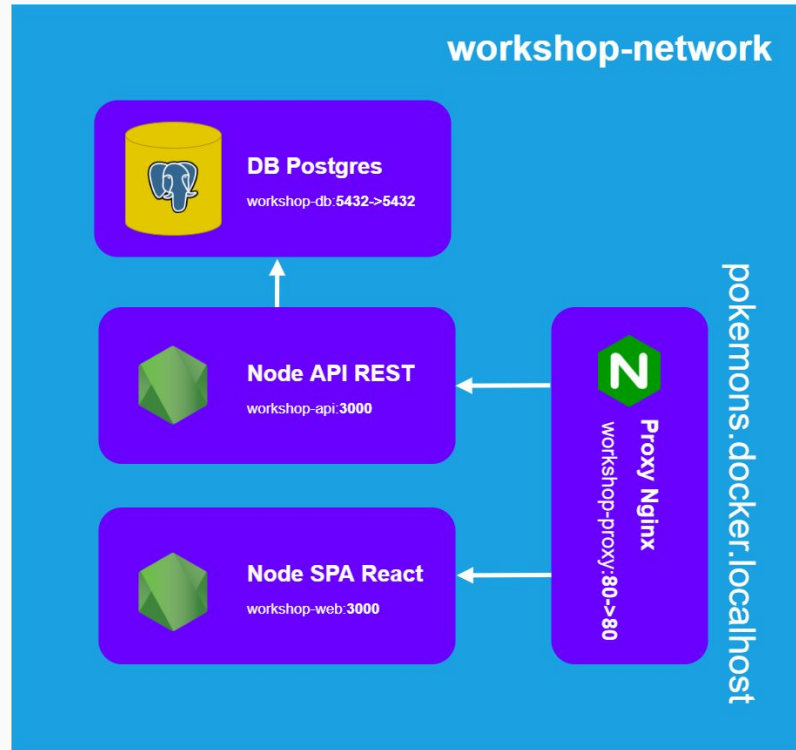
Compose é uma ferramenta para definir e executar aplicativos Docker de vários contêineres. Com o Compose, você usa um arquivo YAML **declarativo** para configurar os serviços do seu aplicativo. Então, com um **único** comando, você cria e inicia todos os serviços de sua configuração.

Mão na massa!

Para pôr em prática os conceitos abordados até o momento vamos criar a infraestrutura no nosso cenário:

*“Preciso criar uma infraestrutura para uma API REST que disponibilize os dados de pokémons localizados em um banco de dados Postgres. Essa API foi construída pela equipe de desenvolvimento com a tecnologia NodeJS, e uma SPA React que consumirá desta API. Precisamos também implementar um servidor Nginx que será o ponto de entrada do nosso domínio **pokemons.docker.localhost** para delegar nossas chamadas para nossas aplicações.”*

Arquitetura do nosso cenário:



\$ **git clone** <https://github.com/grupoatto/workshop-docker.git>

Referências

- <https://docs.docker.com/get-started/overview/>
- <https://docs.docker.com/compose/>
- <https://www.youtube.com/watch?v=yb2udL9GG2U&t=577s>
- <https://www.youtube.com/watch?v=AVNADGzXrrQ>
- <https://www.youtube.com/watch?v=Kz9KYsSDnQE>