LUIZ ZENHA

PÓS TECH

SOFTWARE ARCHITECTURE

DOCKERIZAÇÃO

AULA 02

SUMÁRIO

O QUE VEM POR AÍ?.....3
HANDS ON 4
SAIBA MAIS.....5
O QUE VOCÊ VIU NESTA AULA? 10
REFERÊNCIAS.....11
PALAVRAS-CHAVE.....12

O QUE VEM POR AÍ?

Vamos falar sobre o Registry utilizado pelo Docker, o que é imagem e como utilizá-las para executar nossos containers. Além disso, também veremos os comandos básicos para gerir as imagens em nossos sistemas.

HANDS ON

Nesse Hands On, vamos aprofundar o nosso conhecimento nas imagens do Docker. Para isso, vamos criar um Docker com a imagem do nginx, para criar esse servidor web. Depois, vamos acessar a porta 80 do container para visualizar se ele está retornando corretamente a informação publicada.

SAIBA MAIS

IMAGENS E REGISTRY

Primeiramente, vamos entender o que é a imagem. Quando falamos “imagem”, podemos pensar num arquivo estático que podemos usar para criar um container. Nesse arquivo, vamos colocar as instruções do que deve ser instalado e executado.

Podemos pensar no seguinte exemplo: vamos instalar um container Linux com um servidor Apache Tomcat, que irá realizar a organização do serviço WEB.

Quando pensamos em imagens, devemos analisar sempre em camadas que podem ser reutilizadas quantas vezes quisermos. Então, pensando no exemplo citado, vamos criar a nossa primeira imagem, ou se você preferir, nossa primeira camada. Nela, vamos colocar as configurações do sistema operacional que, no nosso exemplo, será o Linux Ubuntu.

Nessa camada, também colocaremos algumas instruções do que precisa ser configurado no nosso sistema operacional, como um SSH, um FTP, enfim, qualquer serviço que precisemos.

Pronto! Nossa primeira camada está pronta! Podemos reutilizá-la em vários outros containers. Então, sempre que precisamos de um container com esse sistema operacional, com SSH e FTP, podemos usar essa imagem. E, como ela é imutável, sempre temos a certeza de que ela será recriada igualmente.

As imagens são sempre divididas em camadas. A camada base é sempre inalterável, sendo utilizada apenas para leitura. Não conseguimos manipular essa estrutura, pois nela contém o Kernel, a estrutura de inicialização do container, controles de acesso e namespaces para blindar e proteger nosso container.

A segunda camada que podemos trabalhar é referência do sistema operacional que iremos utilizar como por exemplo, o Ubuntu.

Após definir a imagem base do sistema, vamos criar uma nova imagem para hospedar um site, para isso, instalamos o Apache para executar uma aplicação web, e, ao criar essa imagem que irá executar esse serviço do Apache, teremos então uma imagem com 3 camadas (Kernel base, sistema operacional e Apache).

A cada criação de uma imagem com origem em uma imagem anterior, vamos criando novas camadas, sempre lembrando que as alterações que fazemos é necessariamente naquela camada que estamos editando e nunca na imagem de origem, por isso falamos que a imagem é um arquivo estático e imutável.

Agora que temos as imagens, precisamos utilizar um registry, que tem como principal função guardar nossas imagens de forma versionada e organizada, para que tenhamos fácil acesso para download e utilizemos para criar/executar nossos containers.

Podemos ter um registry privado ou público, o Docker Hub é um registry público que também oferece formas de trabalhar com imagens privadas. Por exemplo, no caso de termos uma imagem que executa uma API do nosso sistema, se não estivermos falando de um projeto open source, não convém deixarmos a imagem pública para que qualquer um faça o download e execute o container.

O “catálogo” de imagens do Docker Hub pode ser acessado através do link: <https://hub.docker.com>, nele podemos realizar a busca de diversos tipos de imagens para nosso uso. Se buscarmos por Ubuntu encontraremos os detalhes da imagem que executamos nos passos anteriores.

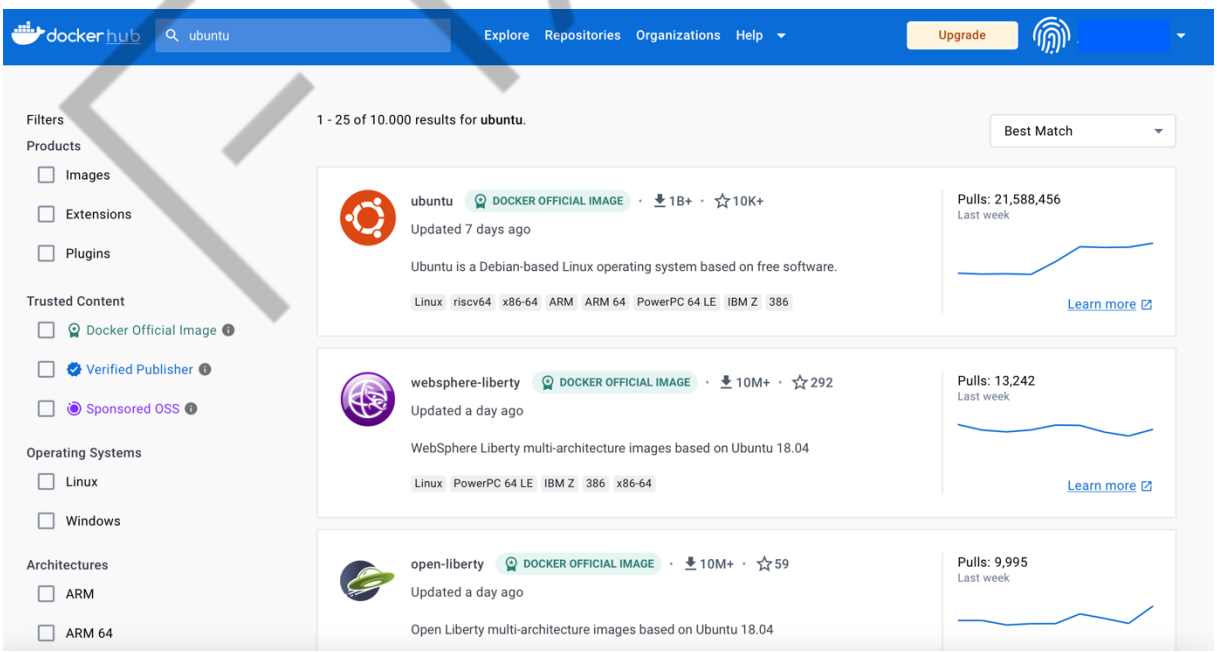

Figura 1 – Catálogo de imagens do Docker Hub
Fonte: Docker Hub (2023)

Para utilizar imagens privadas, é necessário fazer um cadastro e pagar pelo serviço do Docker Hub.

Existem também outros fornecedores que podem prover um container registry privado para sua utilização, como é o caso do Google Cloud, que tem o serviço Container Registry, AWS ECR (Elastic Container Registry) e o Github Packages. Cada um possui uma forma distinta de cobrar por esse serviço, sendo geralmente por armazenamento e tráfego de dados, porém possuem uma quota gratuita de uso.

Normalmente a escolha do Registry privado acaba sendo baseada de acordo com o provedor cloud utilizado, visto que praticamente todos já oferecem esse serviço.

Para ambientes produtivos e principalmente para empresas é importante também que o registry escolhido ofereça um serviço de análise de vulnerabilidades, ou ao menos que sejam compatíveis para se integrar com ferramentas disponíveis no mercado. Isso garantirá uma camada extra de segurança para as aplicações corporativas, evitando que sejam portas de entradas para ataques maliciosos.

GERENCIANDO IMAGENS

Quando iniciamos um container, o primeiro passo do Docker é saber se localmente temos aquela imagem disponível em nossa máquina. Isto para evitar o download desnecessário.

Com o uso de diversas imagens é comum ficarmos com as mais antigas que já não são mais utilizadas. Neste caso, podemos apaga-las (na nossa aula em vídeo demonstramos como realizar o processo). O comando para vermos as imagens locais é ‘docker images’ ou o ‘docker image ls’, ambos irão listar as imagens e trazer informações como repositor, o nome da imagem, tag, image id, data de criação e o espaço ocupado em disco.

Execute o comando a seguir em seu terminal e veja o resultado:

```
$ docker images
```

Comando 2 – Listando imagens locais
Fonte: Elaborado pelo autor (2023)

Para apagar podemos executar o comando a seguir, passando o image id da imagem que desejamos apagar:

```
$ docker image rm IMAGE_ID
```

Comando 3 – Apagar imagem local
Fonte: Elaborado pelo autor (2023)

Se existir um container que esteja utilizando a imagem no momento da exclusão, em execução ou mesmo parado, o Docker não permitirá a exclusão da imagem.

Para isso, ou será necessário parar e remover o container que esteja executando essa imagem, ou podemos também executar o comando passando o parâmetro -f para forçar a remoção:

```
$ docker image rm -f IMAGE_ID
```

Comando 3 – Apagar imagem local
Fonte: Elaborado pelo autor (2023)

Como alternativa para remover imagens temos o comando ‘docker rmi’, que é a forma reduzida do comando ‘docker image rm’.

Quando temos muitas imagens antigas que já não usamos e precisamos liberar armazenamento em disco, pode-se usar o comando ‘docker image prune’. Esse comando irá automaticamente apagar todas as imagens que não estiverem sendo usadas:

```
$ docker image prune
```

Comando 4 – Apagar imagem local
Fonte: Elaborado pelo autor (2023)

Agora que vimos diversas formas de apagar as imagens que não estamos usando, temos um comando muito importante e muito usado no dia-a-dia dos devs que utilizam containers, que é o ‘docker pull’. Ele é usado para baixar as imagens que iremos utilizar em nosso computador deixando-as salvas e, assim, acelerando o processo quando formos utilizá-las

Veja a seguir como executar o comando:

```
$ docker pull image:tag
```

Comando 5 – Baixar imagem do registry
Fonte: Elaborado pelo autor (2023)

O nome da imagem e a tag podem ser encontradas no registry, no nosso caso, usamos o Docker Hub. Um outro comando que é bastante utilizado é o Docker push, que ao contrário do pull, serve para enviarmos uma imagem para o registry para que você utilize posteriormente, seja distribuindo de forma pública ou em um registry privado.

O QUE VOCÊ VIU NESTA AULA?

Nessa aula nos aprofundamos sobre Docker, em especial nas imagens e no Docker hub, que são itens fundamentais para trabalharmos com containers Docker.

Vimos os comandos para gerir as imagens e como encontrar imagens que necessitamos no registry público, o Docker Hub.

O que achou desta aula? Dê as suas opiniões e tire as suas dúvidas na nossa comunidade no Discord! Estaremos lá te esperando, junto com os nossos professores e professoras, Community manager e colegas da turma.

REFERÊNCIAS

ÁLVARES, Bernard. As camadas que compõe uma Docker Image. **Medium**. 23 de março de 2020. Disponível em: https://medium.com/@bernard.luz/as-camadas-que-compõe-uma-docker-image-f77cfa7d04ce. Acesso em: 13 mar. 2023.

ESPINOSA, Allan. Docker High Performance. [s.l]: Packt Publishing, 2019.

Registros, imagens e contêineres do Docker. **Microsoft Learn**. 2023. Disponível em: https://learn.microsoft.com/pt-br/dotnet/architecture/microservices/container-docker-introduction/docker-containers-images-registries. Acesso em: 13 mar 2023.

RIBEIRO, Uirá, Comandos Docker: veja 8 comandos básicos mais utilizado. **Certificação Linux**. 19 de novembro 2021. Disponível em: https://www.certificacaolinux.com.br/comandos-docker/ Acesso em: 13 mar 2023.

ROMERO, Daniel. Containers com Docker: Do desenvolvimento à produção. [s.l.]: Editora Casa do Código, 2015.

PALAVRAS-CHAVE

Palavras-chave: Docker, Container, Docker Hub, Images, Registry.