Contêineres com Docker

Introdução

Cada vez mais as arquiteturas de micro serviços têm sido empregadas por empresas de todo mercado. Este módulo tem por objetivo introduzir a técnica de engenharia de aplicações orientada a contêineres, bem como trazer contato prático através da ferramenta Docker.
Neste módulo iremos abordar as principais motivações para o uso do Docker e de Contêineres, um hands-on com a ferramenta Docker Desktop (uma versão local da plataforma de conteinerização) onde será mostrado como criar aplicações encapsuladas em imagens, execução do contêiner, e principais comandos do Docker.

Objetivos da aula

Os objetivos deste módulo são mostrar como utilizar o Docker, como criar aplicações e imagens em contêineres através comandos, e como fazer uso da ferramenta local para gerir suas aplicações em micro-servidas.

Resumo

De acordo com a definição da própria página do projeto, Docker é "uma plataforma desenhada para ajudar desenvolvedores a construir,

compartilhar e executar aplicações modernas". Esta plataforma vem sendo utilizada cada vez mais pela TI de empresas de todos os tamanhos e de diferentes setores por ser uma ferramenta eficiente e robusta na implementação de micro serviços. Diferentemente da arquitetura monolítica onde todo um sistema é executado em um único servidor, o micro serviço realiza a "quebra lógica" deste grande sistema em diferentes componentes independentes, de modo que fica mais fácil a construção de cada um e a manutenção do todo. Ainda, como cada serviço é executado em um espaço de memória encapsulado e enxuto (contêiner), fica mais fácil gerenciar complexas questões como a escalabilidade e a alta disponibilidade.

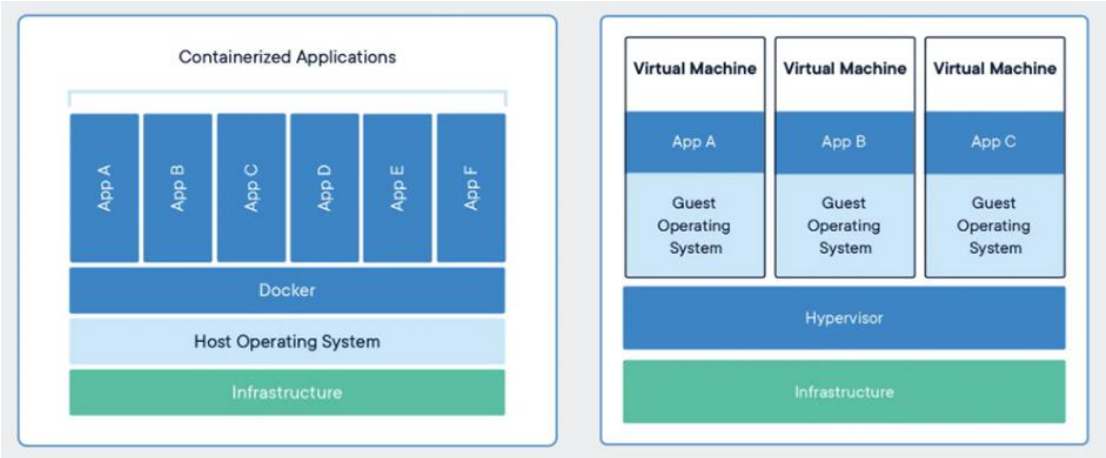

Figura 1: comparação entre a arquitetura com contêineres e a arquitetura com máquinas virtuais.
Fonte: https://www.docker.com/resources/what-container/

Na Figura 1, você pode observar a diferença entre as arquiteturas de contêineres e de máquinas virtuais. Enquanto que nas máquinas virtuais temos um inteiro sistema operacional sendo executado para cada aplicação, na arquitetura de contêineres as mesmas aplicações podem ser executadas fazendo uso apenas dos recursos e bibliotecas de software que necessitam, economizando drasticamente processamento e memória do sistema hospedeiro e da infraestrutura.

Em ambiente corporativo, geralmente o Docker é executado como um "engine" (motor de processos) que gerencia a construção, montagem de imagem, e execução. Tanto a documentação completa quanto o download do projeto podem ser obtidos através do link: https://www.docker.com/. Já em ambiente local e acadêmico, podemos utilizar uma versão Desktop do Docker, disponível para download em https://www.docker.com/products/docker-desktop/. Por fim, temos ainda uma plataforma web chamada Docker Hub (acessível em https://hub.docker.com/) onde podemos criar contas para hospedar imagens Docker construídas e customizadas conforme cada necessidade, que ficam armazenadas na nuvem e podem ser compartilhadas entre toda a comunidade. A Figura 2 nos traz um print do dashboard funcional do Docker Desktop, e a Figura 3 nos mostra a landing page do Docker Hub.

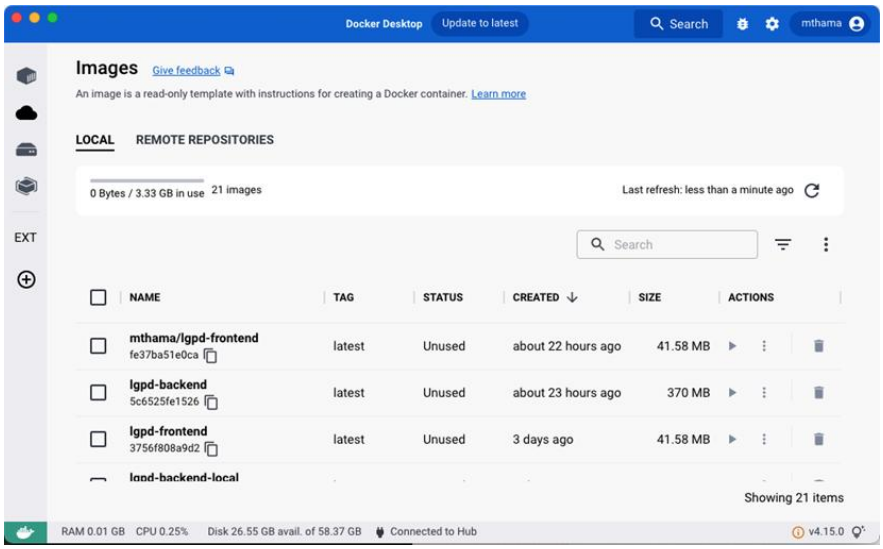

Figura 2: dashboard funcional do Docker Desktop, mostrando as imagens disponíveis localmente.
Fonte: catálogo do professor autor.

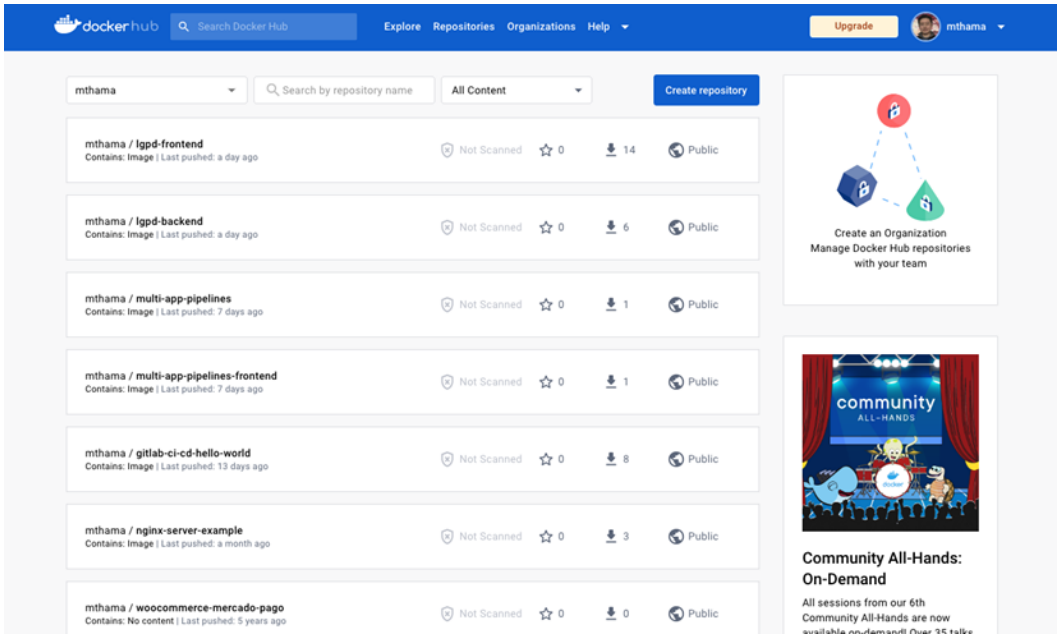

Figura 3: landing page do Docker Hub, mostrando as imagens armazenadas em nuvem. Fonte: catálogo do professor autor.

Seja qual for a versão do Docker utilizada, há sempre o arquivo Dockerfile no qual definimos como deverá ser construída uma imagem. Este arquivo possui uma sintaxe relativamente simples, mas suficiente para definir quais compiladores serão utilizados para construir nosso código, quais componentes (conteúdos html, javascripts, arquivos .jar, etc) deverão ser copiados para a imagem, e quais comandos prévios serão executados quando a imagem for iniciada. Em cada imagem é também definida a plataforma sob a qual a aplicação irá executar. Para construir uma imagem usamos a seguinte linha de comando:

$ docker build <caminho do arquivo Dockerfile>

É possível ainda definirmos o nome da imagem e a versão da mesma, através da opção "*-t*". Em nossa vídeo aula, mostramos a construção, a execução, e o envio de uma imagem para Docker Hub, seguindo os seguintes passos:

1. Com o Docker Desktop aberto, navegue via prompt até a raíz do projeto: https://github.com/FaculdadeDescomplica/Advanced-BackEnd/docker-nginx-hello-world;

2. Executar o comando para criar a imagem chamada "*nginx-server-example*":
   $ docker build . -t nginx-server-example

3. Executar o comando para levantar a execução do contêiner:
   $ docker run -p 8080:80 nginx-server-example

4. Acesse o Hub em https://hub.docker.com/ e faça login em sua conta. Não se esqueça de configurar o Multi-Factor Authentication.

5. Faça o login no Docker Desktop local de sua máquina com suas credenciais:
   $ docker login

6. Liste as imagens presentes localmente, anotando o ID da imagem da sua aplicação recém criada:
   $ docker images

7. Crie uma tag para referenciar a imagem no Docker Hub, informando o ID da imagem:
   $ docker tag <image_id> mthama/nginx-server-example:latest

8. Execute o comando para enviar a imagem para o Docker Hub, informando o ID da imagem:
   $ docker image push mthama/nginx-server-example:latest

Os comandos acima listados criam, executam, e enviam ao Docker Hub uma imagem Docker operando um servidor *nginx* disponível na porta 8080. Você pode conferir a página principal do servidor em http://localhost:8080.

Em outro de nossos exemplos, executamos uma aplicação Angular, no projeto "*gitlab-ci-cd-hello-world*". Diferente do primeiro caso, iniciamos este projeto em localhost com os seguintes comandos:

1. Navegue via prompt até a raíz do projeto "*gitlab-ci-cd-hello-world*".

2. Download e instalação das dependências do Angular:
   $ npm install

3. Execução da aplicação Angular em localhost:
   $ ng serve -o

4. Visitar: http://localhost:4200/

Na sequência podemos seguir com a construção da imagem desta mesma aplicação, para ser executada em contêiner:

1. Construção dos binários e arquivos html:
   $ ng build

2. Criação da imagem através do Dockerfile:
   $ docker build -t ng-docker-k8s:1.0 .

3. Execução da imagem em contêiner:
   $ docker run -it --name hellodescomplica --rm -p 80:80 ng-docker-k8s:1.0

4. Visitar: http://localhost:80/

Por fim, podemos mencionar a versatilidade do Docker que nos permite utilizar a ferramenta *Docker Compose* para executar múltiplas aplicações em forma coesa e coordenada. O *Docker Compose* utiliza um arquivo de definição de stack nomeado, por padrão, de "*docker-compose.yml*" em que declaramos quais imagens serão utilizadas no conjunto de aplicações, quais as portas mapeadas por cada uma destas aplicações, aspectos de rede, além de dependências e interdependências existentes entre as aplicações. O comando para executar uma stack de aplicações se dá da seguinte forma dentro do diretório que contém o *docker-compose.yml*:

$ docker-compose up

Conteúdo bônus

Tópicos avançados

Podemos ir muito mais longe, criando e executando até mesmo uma aplicação Hadoop. Para proceder com este hand-on, utilize o projeto "*docker-hadoop*". Siga os seguintes passos:

1. Com o Docker Desktop aberto, navegue via prompt até a raíz do projeto:
   https://github.com/FaculdadeDescomplica/Advanced-BackEnd/docker-hadoop

2. Faça a construção da imagem do Docker com a instalação do Hadoop:
   $ docker build . -t hadoop:latest

3. Execute a imagem, mapeando as portas lógicas utilizadas pelo Hadoop:
   $ docker run -it --name hadoop --rm -p 50070:50070 -p 8088:8088 hadoop:latest

4. Após o sistema ser levantado, você poderá observar a página principal da infraestrutura em http://localhost:50070/explorer.html#/example e o dashboard do cluster em http://localhost:8088/cluster

5. Para submeter a aplicação que vem nos exemplos do próprio Hadoop basta entrar no contêiner e executar o exemplo:
   $ docker container exec -it hadoop bash
   $ /root/hadoop/bin/hadoop jar /root/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.1.jar pi 1 10

Caro estudante, você consegue acessar os códigos utilizados na disciplina no link a seguir:

https://github.com/FaculdadeDescomplica/Advanced-BackEnd

Referência Bibliográfica

Principais Comandos do Docker. Docker Command Line Reference, 2023. Disponível em: https://docs.docker.com/engine/reference/commandline/docker/. Acesso em: 24/01/2023.

Página do Projeto Docker. Docker.com, 2023. Disponível em: https://www.docker.com/. Acesso em: 24/01/2023.

Referência do Dockerfile. Docker.com, 2023. Disponível em: https://docs.docker.com/engine/reference/builder/. Acesso em: 24/01/2023.

Página do Docker Hub. Hub.docker.com, 2023. Disponível em: https://hub.docker.com/. Acesso em: 24/01/2023.