



INSTITUTO FEDERAL

Ceará

Campus Fortaleza

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
CAMPUS FORTALEZA

AVALIAÇÃO 9 - SISTEMAS OPERACIONAIS DE REDES 2

Alunos: Antonio Gilmar Oliveira de Paula

Emily Ferreira de Lima

Professor: Taveira

TUTORIAL SOBRE OS PRINCIPAIS TÓPICOS DO DOCKER-COMPOSE

Introdução

O docker-compose facilita o gerenciamento de contêineres do docker, basta apenas configurar um arquivo do tipo .yaml, seguindo a estrutura básica da estrutura dos arquivos dessa extensão. O problema do uso do docker convencional é o fato da dificuldade de trabalhar com vários contêineres ao mesmo tempo, precisa criar e configurar um por um, já com o docker-compose você configura quantos contêineres você quiser, através de configurações de um só arquivo, o docker-compose.yaml (é o padrão, mas é possível mudar o nome do arquivo).

COMANDOS	EXPLICAÇÃO
docker-compose up	cria e inicia os contêineres
docker-compose build	realiza apenas a etapa de build das imagens que serão utilizadas
docker-compose logs	visualiza os logs dos contêineres
docker-compose restart	reinicia os contêineres
docker-compose ps	lista os contêineres
docker-compose scale	permite aumentar o número de réplicas de um contêiner
docker-compose start	inicia os contêineres
docker-compose stop	paralisa os contêineres

docker-compose down	paralisa e remove todos os contêineres e seus componentes como rede, imagem e volume
---------------------	--

Estrutura do arquivo

Como já dito anteriormente, o arquivo precisa ter uma configuração inicial para que o docker-compose possa gerenciar os contêineres de uma maneira sólida e descrita em um arquivo.

```
version: "3.7"
services:
  app:
    image: elixir:1.10
    container_name: "app"
    volumes:
      - ./:/var/www/application/
    working_dir: "/var/www/application/"
    expose:
      - "4000"
    tty: true

  nginx:
    image: nginx:1.17
    container_name: "nginx"
    ports:
      - "80:80"

  postgres:
    image: postgres:9.5
    container_name: "postgres"
    volumes:
      - ./postgres:/var/lib/postgres
    environment:
      - POSTGRES_DB=your_db
      - POSTGRES_USER=your_user
      - POSTGRES_PASSWORD=your_secret
    ports:
      - "5432:5432"
```

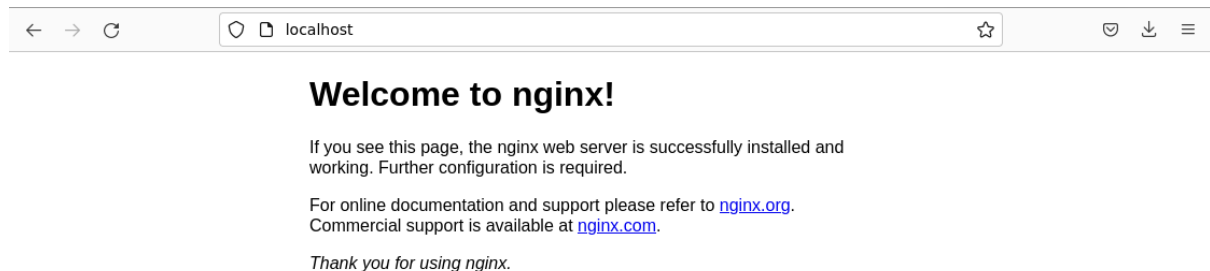
No arquivo acima, vemos um exemplo de uma aplicação Elixir rodando com um servidor HTTP nginx com um banco de dados Postgres. Todos rodando sob um container diferente.

Usando o docker-compose

1 - Iniciando os containers

```
emilly@emilly-virtualbox: ~/Desktop/test$ sudo docker-compose up -d
[sudo] password for emilly:
/snap/docker/2285/lib/python3.6/site-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 3.6 is no longer supported by the Python core team. Therefore, support for it
is deprecated in cryptography and will be removed in a future release.
  from cryptography.hazmat.backends import default_backend
Creating network "test_default" with the default driver
Pulling app (elixir:1.10)...
1.10: Pulling from library/elixir
620af4e91dbf: Pull complete
fae29f309a72: Pull complete
28fca74d99b6: Pull complete
8b5db87f5b42: Pull complete
fa488706e613: Pull complete
b8ebf84fbc7a: Pull complete
bac4ae85016e: Pull complete
b0e7e6b3123b: Pull complete
6a7a567f6522: Pull complete
Digest: sha256:6d8a818e87d1b0ca5b3657ac28d0800d0a44c28d6a4e3c0e0291845ac260cdc0
Status: Downloaded newer image for elixir:1.10
Pulling nginx (nginx:1.17)...
1.17: Pulling from library/nginx
afb6ec6fd1c: Pull complete
b90c53a0b692: Pull complete
11fa52a0f4dc: Pull complete
Digest: sha256:6ffff5753e3b34c36e24e37039ee9eae1fe38a6420d8ae16ef37c92d1eb26699
Status: Downloaded newer image for nginx:1.17
Pulling postgres (postgres:9.5)...
9.5: Pulling from library/postgres
fa1690ae9228: Pull complete
b73f6e87b158: Pull complete
b73a0c44ddb4: Pull complete
b7e5342b01d4: Pull complete
578aad0862c9: Pull complete
a0b157088f7a: Pull complete
e39046f06fc5: Pull complete
ee19407bdc48: Pull complete
e53b7c20aa96: Pull complete
e135edcc0831: Pull complete
fed07b1b1b94: Pull complete
18d9026fcfbd: Pull complete
4d2d5fae97d9: Pull complete
d419466e642d: Pull complete
Digest: sha256:75ebf479151a0fd77bf2fed40ef7ece8d518c23264734c48f2d1de42b4eb40ae
Status: Downloaded newer image for postgres:9.5
Creating nginx ... done
Creating app ... done
Creating postgres ... done
emilly@emilly-virtualbox: ~/Desktop/test$
```

2 - Visualizando o servidor funcionando



3 - “Desligando” os containers

```
emilly@emilly-virtualbox: ~/Desktop/test$ sudo docker-compose down
/snap/docker/2285/lib/python3.6/site-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 3.6 is no longer supported by the Python core team. Therefore, support for it
is deprecated in cryptography and will be removed in a future release.
  from cryptography.hazmat.backends import default_backend
Stopping postgres ... done
Stopping app ... done
Stopping nginx ... done
Removing postgres ... done
Removing app ... done
Removing nginx ... done
Removing network test_default
emilly@emilly-virtualbox: ~/Desktop/test$
```

Conclusão

Com os nossos estudos, chegamos a conclusão de que vale muito a pena estudar e usar o docker-compose pois facilita a utilização profissional e consistente do docker, além da possibilidade de versionar a configurar o docker-compose.yml. Obrigado(a)!