06

Subindo os serviços

Transcrição

*Se você utiliza Linux, provavelmente não conseguirá utilizar o **Docker Compose**, pois o mesmo não é instalado na instalação padrão. Para instalá-lo, primeiramente faça este exercício (https://cursos.alura.com.br/course/docker-novo/task/29559) e prossiga com o treinamento.*

Com o **docker-compose.yml** pronto, podemos subir os serviços, mas antes devemos garantir que temos todas as imagens envolvidas neste arquivo na nossa máquina. Para isso, dentro da pasta do nosso projeto, executamos o seguinte comando:

```
$ cd Desktop/alura-docker-cap06/
~/Desktop/alura-docker-cap06$ sudo docker-compose build
```

Com os serviços criados, podemos subi-los através do comando `docker-compose up` . Esse comando irá seguir o que escrevemos no **docker-compose.yml**, ou seja, cria a rede, o *container* do MongoDB, os três *containers* da aplicação e o *container* do NGINX. Depois, são exibidos alguns *logs*, sendo que cada um dos *containers* fica com uma cor diferente, para podermos distinguir melhor.

Se tudo funcionou, quando acessarmos o NGINX, seremos redirecionados para algum dos *containers* da nossa aplicação. Configuramos a porta **80** para acessar o NGINX, então vamos acessar no navegador a página http://localhost:80/ (http://localhost:80/). Ao acessar a página, vemos a seguinte mensagem no *log*:

```
alura-books-1 | Exibindo a Home!
```

Ou seja, fomos redirecionados para o primeiro *container*. Ao atualizar a página, vemos no *log* que fomos redirecionados para o segundo *container*, ou seja, o *load balancer* está funcionando.

Vamos então alimentar o banco de dados acessando a página http://localhost:80/seed (http://localhost:80/seed), e novamente acessar a página inicial, a http://localhost:80/ (http://localhost:80/), assim veremos os livros sendo exibidos.

Então, com um único comando, levantamos todos os *containers*, montamos o *load balancer*, servimos os arquivos estáticos, criamos o banco e colocamos todos eles na mesma rede, bem mais prático do que estávamos fazendo anteriormente.

Para parar a execução, utilizamos o atalho **CTRL + C**. E não somos obrigados a ficar vendo esses *logs*, podemos utilizar a já conhecida *flag* `-d` :

```
docker-compose up -d
```

E com o comando `docker-compose ps` , podemos ter uma visualização simples dos serviços que estão rodando:

```
          Name                        Command             State             Ports
-------------------------------------------------------------------------------------------
alura-books-1                 npm start                     Up      0.0.0.0:9022->3000/tcp
alura-books-2                 npm start                     Up      0.0.0.0:9021->3000/tcp
```

```
alura-books-3               npm start                    Up      0.0.0.0:9023->3000/tcp
aluradockercap06_mongodb_1  docker-entrypoint.sh mongod  Up      27017/tcp
nginx                       nginx -g daemon off;         Up      443/tcp, 0.0.0.0:88->80/tcp
```

Agora que não estamos mais vendo os *logs*, como paramos os serviços? Para isso, utilizamos o comando `docker-compose down`. Esse comando para os *containers* e os remove.

E não é por que eles são serviços, que eles não tem um *container* por debaixo dos panos, então nós conseguimos interagir com os *containers* utilizando todos os comandos que já vimos no treinamento, por exemplo para testar a comunicação entre eles:

```
docker exec -it alura-books-1 ping alura-books-2
```

Mas também podemos utilizar o nome do **serviço**, não precisamos necessariamente utilizar o nome do *container*:

```
docker exec -it alura-books-1 ping node2
```

Assim conseguimos fazer a comunicação tanto com o nome do *container* quanto com o nome do serviço, pois os dois estão atrelados ao mesmo IP.