



## Entendendo o Docker Compose

### Transcrição

Nesta aula, estudaremos uma tecnologia chamada **Docker Compose**, que nos auxiliará a lidar com múltiplos *containers* simultaneamente.

Na aula anterior, para subir a aplicação **alura-books**, foi necessário subirmos dois *containers*, executando os seguintes comandos:

```
docker run -d --name meu-mongo --network minha-rede  
mongo  
docker run --network minha-rede -d -p 8080:3000  
douglasq/alura-books:cap05
```

[COPIAR CÓDIGO](#)

Isso tudo depois de termos construído pelo menos a imagem **douglasq/alura-books**

### O problema

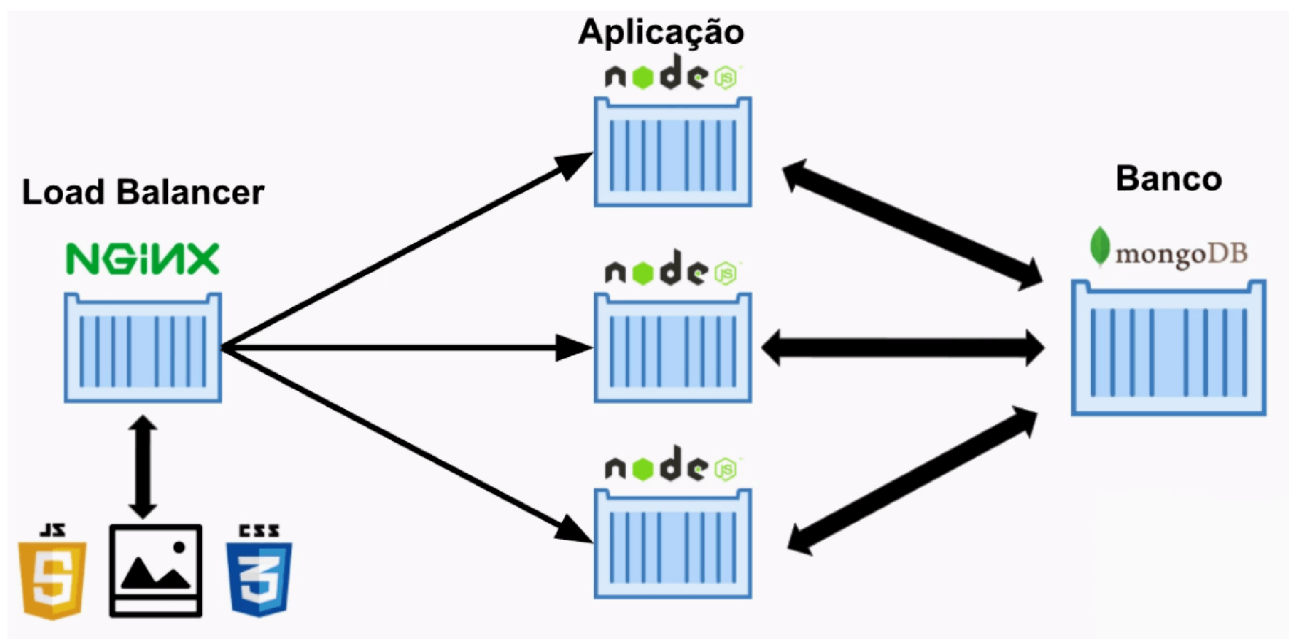
Esses dois comandos criam dois *containers*, mas subindo eles desse jeito manual, é muito comum esquecermos de passar alguma *flag*, ou subir o *container* na ordem errada, sem a devida rede, ou seja, é um trabalho muito manual e facilmente suscetível a erros, isso com somente dois *containers*.

Esse modo de subir os *containers* na mão é bom se quisermos criar um ambiente rapidamente, ou são poucos *containers*, mas quando a aplicação começa a crescer, temos que digitar muitos comandos.

## Funcionamento das aplicações em geral

Na vida real, sabemos que a aplicação é maior que somente dois *containers*, geralmente temos dois, três ou mais *containers* para segurar o tráfego da aplicação, distribuindo a carga. Além disso, temos que colocar todos esses *containers* para se comunicar com o banco de dados em um outro *container*, mas quanto maior a aplicação, devemos ter mais de um *container* para o banco também.

E claro, se temos três aplicações rodando, não podemos ter três endereços diferentes, então nesses casos utilizamos um *Load Balancer* em um outro *container*, para fazer a distribuição de carga quando tivermos muitos acessos. Ele recebe as requisições e distribui para uma das aplicações, e ele também é muito utilizado para servir os arquivos estáticos, como imagens, arquivos CSS e JavaScript. Assim, a nossa aplicação controla somente a lógica, as regras de negócio, com os arquivos estáticos ficando a cargo do *Load Balancer*.



Se formos seguir esse diagrama, teríamos que criar cinco *containers* na mão, e claro, cada *container* com configurações e *flags* diferentes, além de termos que nos preocupar com a ordem em que vamos subi-los.

## Docker Compose

Ao invés de subir todos esses *containers* na mão, o que vamos fazer é utilizar uma tecnologia aliada do Docker, chamada **Docker Compose**, feito para nos auxiliar a orquestrar melhor múltiplos *containers*. Ele funciona seguindo um arquivo de texto **YAML** (extensão **.yaml**), e nele nós descrevemos tudo o que queremos que aconteça para subir a nossa aplicação, todo o nosso processo de *build*, isto é, subir o banco, os *containers* das aplicações, etc.

Assim, não precisamos ficar executando muitos comandos no terminal sem necessidade. E esse será o foco desta aula, montar uma aplicação na estrutura descrita anteriormente na imagem, que é uma situação comum no nosso dia-a-dia.