



Gerando imagem Docker da aplicação

Transcrição

[00:00] Olá, bem-vindos de volta ao curso de Spring Boot na Alura. Agora que já vimos como funciona essa parte de Build, de Deploy da aplicação e como rodar pelo prompt de comandos, chegou a hora de vermos outra abordagem, de fazer Deploy utilizando os contêiners, no caso container Docker, que é um tipo de tecnologia que está bastante popular, sendo usada por muitos times de desenvolvimento e de infraestrutura.

[00:26] Então neste capítulo vamos aprender como eu posso fazer para colocar o Docker na aplicação com Spring Boot, como eu faço para rodar e tudo mais.

[00:35] Só um parênteses importante, o curso é focado no Spring Boot, não é um treinamento de Docker, então nesse curso não vamos abordar como funciona o Docker, como instalar, configurar.

[00:47] Até porque temos curso de Docker na Alura. Por exemplo, tem o curso “Docker: Criando containers sem dor de cabeça”. Então caso você não tenha conhecimentos em Docker é importante que você assista, faça esse outro treinamento, pois vamos focar mais na parte do Spring Boot, e não no Docker em si. Assim não perdemos o foco do treinamento.

[01:08] Então vamos lá. Na última aula vimos como fazer para gerar o Build da aplicação via WAR. Tivemos que fazer algumas alterações no projeto. Então eu vou desfazer essas mudanças e vou trabalhar com o modelo tradicional do Spring Boot, utilizando um JAR.

[01:25] Então vou apagar o método que sobrescrevemos; vou apagar o `extends` ; organizar os `imports`. Vamos deixar a classe `main` do jeito que ela estava antes, do jeito tradicional.

[01:36] No `pom.xml` vou apagar a tag `<packaging>war</packaging>` , porque assim o padrão é JAR. E vou apagar aquela dependência que tínhamos adicionado do Tomcat com o escopo de `provided` .

[01:52] Vou salvar. E agora nosso projeto voltou ao padrão tradicional do Spring Boot, que é ser *buildado* no formato JAR. Feito isso, vamos utilizar o Docker agora. Vamos ver como fazer para colocar o Docker na aplicação, trabalhando com JAR mesmo, no modelo tradicional.

[02:11] Tem várias abordagens para você colocar o Docker numa aplicação, sendo que uma delas, a mais simples de todas, é criar aquele arquivo Docker file e ensinar para o Docker como ele deve criar a imagem da nossa aplicação. Nós vamos usar essa abordagem, que é a mais simples e mais comum.

[02:28] O projeto só começou a dar um erro por conta da dependência que eu removi. Então antes vou clicar com o botão direito, ir na opção “Maven > Update Project...”, dar um “Ok”. E o Maven vai atualizar o projeto, vai recompilar e vai parar com esse erro que ele fica acusando.

[02:48] Às vezes quando você mexe nas dependências e salva ele dá esse problema, pelo menos no caso do Eclipse. E é só você vir nessa opção “Maven > Update Project...”, que ele atualiza o projeto, recompila e para com o erro. Já parou.

[03:05] Então a ideia é que eu preciso criar o Docker file e esse arquivo tem que ser criado no diretório raiz. Então com o botão direito em cima da pasta “forum” vamos escolher a opção “New”, e no Eclipse tem que ser a opção “New > File”. E digitamos embaixo o nome do arquivo: “Dockerfile”, tudo junto e com a letra D em maiúsculo.

[03:24] Não tem extensão esse arquivo, então é só “Dockerfile” mesmo. Clico em “Finish”, ele vai criar um arquivo vazio. E eu já tenho copiado, vou só colar quais são as propriedades, mas eu falo um pouco sobre elas para explicar para vocês. Então vou dar um “Ctrl + V” e está aqui o nosso Docker file.

[03:43] Na primeira linha eu estou dizendo qual é a imagem base que ele vai utilizar. Como é uma aplicação Java vamos usar o `openjdk:8-jdk-alpine`. Ele vai baixar uma imagem, um sistema operacional que já vem com a JDK instalada.

[03:56] O segundo comando é para criar um usuário nesse sistema operacional. Nós estamos criando um usuário com um grupo chamado Spring justamente para rodar a aplicação com o usuário Spring ao invés de rodar a aplicação com o usuário root, que tem uns problemas de segurança quanto a isso.

[04:14] Depois nós falamos para usar o usuário Spring, que foi criado no passo anterior. Crio um parâmetro chamado `JAR_FILE=target/*.jar`. Ele vai pegar o nosso JAR que foi gerado pelo Build do Maven e vai colocar o nome desse arquivo nessa variável chamada `JAR_FILE`.

[04:36] Na sequência ele vai fazer um `COPY`, vai copiar esse arquivo JAR. E tem um asterisco porque eu não sei o nome do arquivo, vai depender de como está configurado no Maven. Mas vamos copiá-lo para o sistema operacional com o nome `app.jar`.

[04:50] E por fim, a última linha é qual é o comando que vai ser executado no nosso contêiner, quando criarmos o contêiner de fato. Ele vai rodar da mesma maneira que rodamos na última aula, via linha de comando. É um `[“java”, “-jar”, “/app.jar”]`.

[05:06] Agora como padronizamos para o arquivo se chamar `app.jar`, estamos garantindo que o nome do arquivo vai ser esse.

[05:12] Então bem simples, esse é o Docker file. Pega a imagem do JDK, cria um usuário e um grupo Spring, usa esse grupo e usuário Spring no momento, cria uma variável `JAR_FILE` com o diretório, com o arquivo JAR que foi gerado pelo Maven, copia ele para o diretório raiz do sistema operacional com o nome `app.jar` e roda o comando `“java”, “-jar”` para rodar.

[05:35] Com isso ensinamos para o Docker como ele cria uma imagem da nossa aplicação. Feito isso, não precisa mexer em mais nada no nosso projeto.

[05:42] Agora já podemos ir no terminal e pedir para o Docker criar uma imagem. Estou no diretório raiz do projeto, no diretório fórum, vou dar um `ls`. E agora podemos pedir para ele criar uma imagem, usando o comando `docker build -t`.

[06:02] Para dar um nome para nossa imagem vou chamar de `alura/forum`. Então `docker build -t alura/forum`. O espaço + ponto final é para pegar o diretório raiz, o diretório atual.

[06:10] Rodando esse comando o Docker vai criar uma imagem da nossa aplicação. Ainda não estamos rodando o container, só criando uma imagem. Ele vai ler o Docker file, vai baixar qual é a imagem raiz, que é a `jdk-alpine`, vai fazer as configurações e montar nossa imagem.

[06:29] E pronto, ele já rodou tudo certo e disse que criou a nossa imagem com esse nome de tag `alura/forum`.

[06:37] Vou dar uma limpada no terminal e vamos rodar o comando `docker image list` para ele listar as imagens. E apareceu a nossa imagem:

`alura/forum`, a tag `latest`, o ID da imagem. E no meu está dizendo que foi criado duas horas atrás porque eu já tinha criado essa imagem antes, só para ele ficar em cache e não demorar muito no vídeo. E ela ficou com 150 MB.

[06:59] Então essa é a nossa imagem `alura/forum`. Então a partir desse momento já temos a imagem já podemos iniciar um container Docker, passar

parâmetros, configurações e colocar para rodar. Só que em vez de rodar via linha de comando, do jeito tradicional, como se fosse simulando um servidor, nós rodamos através de um container Docker.

[07:18] No próximo vídeo veremos como que fazer para rodar, como fazer para passar as variáveis de ambiente, dentre outras coisas.

[07:24] No vídeo de hoje era só para aprender a criar o Docker file e criar a imagem Docker no nosso sistema operacional. Então espero que vocês tenham gostado desse vídeo, vejo vocês no próximo vídeo. Um abraço e até lá.