

Tutorial de instalação do Rails com Docker no Windows

Para construir apps e outras coisas com Ruby on Rails, precisamos configurar alguns softwares e o ambiente de desenvolvimento em seu computador. Este processo de configuração pode se tornar complexo e desanimador, para resolver este problema existem algumas formas mais rápidas que permitem que com poucos passos você tenha um ambiente prontinho para você começar a criar seus projetos.

Já ouviu falar em Docker? Docker é tipo uma máquina virtual dentro da sua máquina real. Dessa forma você pode ter qualquer ambiente de desenvolvimento na sua máquina sem conflitos e horas de instalações.

E como faço essa mágica acontecer? Primeiramente você precisará instalar o software Docker de acordo com seu sistema operacional.

Docker Windows 10 (Pro ou Enterprise):

<https://hub.docker.com/editions/community/docker-ce-desktop-windows>

Docker Toolbox - Windows 10 (Home), 8, ou 7:

<https://docs.docker.com/toolbox/overview/>

Instruções Docker para Windows 10

Abra o prompt de comando do seu sistema operacional e execute este comando para verificar se o Docker foi instalado corretamente:

```
docker --version
```

Estando tudo certo vamos prosseguir com a configuração. O próximo passo é fazer o download da “máquina virtual” para a nossa real.

```
docker pull jacksonpires/ubuntu-rails-ssh
```

O comando acima fará o download da imagem jacksonpires/ubuntu-rails-ssh, criada por um professor de Rails muito conhecido na internet. Esta imagem ficará

disponível em seu computador logo depois que finalizado. Na sequência rode o seguinte comando:

```
docker run -d -P -v $(pwd):/projects --name railscontainer  
jacksonpires/ubuntu-rails-ssh
```

Esse comando inicializará uma “máquina virtual” baseada na imagem que baixamos e também mapeará a pasta na qual você estiver para dentro do container. Sendo “\$(pwd)” o caminho da pasta onde você criará seu projeto.

Exemplo no Windows:

```
docker run -d -P -v C:\projects:/projects --name railscontainer  
jacksonpires/ubuntu-rails-ssh
```

Lembrando que o primeiro caminho é da pasta na sua máquina real e o segundo é o caminho na “máquina virtual”.

Para que possamos acessar esta “máquina virtual” por SSH (porta 22) é necessário verificar qual a porta correspondente na máquina real. Para isso execute o seguinte comando:

```
docker port railscontainer
```

O retorno será algo como este do exemplo:

```
22/tcp -> 0.0.0.0:32785  
3000/tcp -> 0.0.0.0:32784
```

Para acessar a máquina virtual, que é Ubuntu (linux), você abrirá o Git Bash e rodará o seguinte comando:

```
ssh app@localhost -p 32785
```

Caso não tenha Git Bash, poderá utilizar o Putty para fazer a conexão.

Perceba que o número da porta (em negrito) é o mesmo do equivalente a porta 22 na sua máquina (do comando anterior).

Após a execução do comando acima será solicitada uma senha e você deverá digitar “app” que é a senha padrão para esta máquina virtual.

Instruções Docker Toolbox

Abra o Docker Quickstart Terminal e execute este comando para verificar se o Docker foi instalado corretamente:

```
docker --version
```

Estando tudo certo vamos prosseguir com a configuração. O próximo passo é fazer o download da “máquina virtual” para a nossa real. Ainda neste terminal execute o seguinte comando:

```
docker pull jacksonpires/ubuntu-rails-ssh
```

O comando acima fará o download da imagem jacksonpires/ubuntu-rails-ssh, criada por um professor de Rails muito conhecido na internet. Esta imagem ficará disponível em seu computador logo depois que finalizado.

Acesse pelo terminal a pasta em que deseja salvar seus projetos e execute o seguinte comando:

```
docker run -d -P -v $(pwd):/projects --name railscontainer  
jacksonpires/ubuntu-rails-ssh
```

Esse comando inicializará uma “máquina virtual” baseada na imagem que baixamos e também mapeará a pasta na qual você estiver para dentro do container.

Para que possamos acessar esta “máquina virtual” por SSH (porta 22) é necessário verificar qual a porta correspondente na máquina real. Para isso execute o seguinte comando:

```
docker port railscontainer
```

O retorno será algo como este do exemplo:

```
22/tcp -> 0.0.0.0:32785
```

```
3000/tcp -> 0.0.0.0:32784
```

Para acessar a máquina virtual, que é Ubuntu (linux), você executará o seguinte comando:

```
ssh app@192.168.99.100 -p 32785
```

Após a execução do comando acima será solicitada uma senha e você deverá digitar “app” que é a senha padrão para esta máquina virtual.

Após acesso a máquina virtual

Ao entrar no container você poderá verificar as versões do Ruby e do Rails, rodando...

```
ruby -v
```

e

```
rails -v
```

Acesse a pasta “/projects” usando o seguinte comando:

```
cd /projects/
```

Prontinho. Nesse momento você poderá criar uma aplicação Rails para fazer um teste. Rode:

```
rails new railsgirls
```

Após a aplicação ser criada entre na pasta e inicie o servidor Rails.

```
cd railsgirls
```

```
rails s -b 0.0.0.0
```

Agora é só acessar o navegador no endereço localhost, indicando a porta para qual foi mapeada a porta 3000. No nosso caso vai ser <http://localhost:32784> ou <http://192.168.99.100:32784>.

Alguns erros que podem acontecer:

1. Ao acessar o localhost poderá aparecer uma mensagem de erro relacionado com a gem do sqlite3. Para resolver, acesse seu projeto na sua máquina real:

C:\projects\railsgirls\Gemfile

Altere onde estiver 'sqlite3' para 'sqlite3', '~> 1.3.6'

Na sua máquina virtual execute o seguinte comando:

bundle install

rails s -b 0.0.0.0

Acesse novamente no seu navegador <http://localhost:32784>.

2. Ao acessar por ssh a máquina, caso receba a mensagem de “connection refused” execute o comando “docker-machine ip” para pegar o IP da máquina virtual. Troque o localhost pelo IP correspondente.

3. Ao executar o comando “rails new railsgirls” caso mostre a mensagem “permission denied” execute o seguinte comando dentro da pasta *projects* “sudo mkdir railsgirls” e posteriormente “sudo chmod 777 railsgirls”. Depois poderá executar o comando “rails new railsgirls” sem problema pois o diretório já foi criado com permissão de escrita.

Comandos Docker úteis:

A seguir alguns comandos que serão úteis na utilização do docker e sua máquina virtual. Lembrando que todos deverão ser executado no terminal da sua máquina real.

Lista todos os containers em funcionamento

docker container ls

Para um container

docker container stop railscontainer

Lista todos os containers (mesmo os parados)

docker container ls -a

Inicia um container

docker container start railscontainer

#Lista as portas do container

docker port railscontainer

#Mostra o IP da máquina

docker-machine ip

Fonte:

<https://medium.com/@jacksonpires/usando-docker-para-desenvolver-com-ruby-on-rails-3eff32d05006>