

Agora que fizemos o teste local e vimos que tudo está funcionando corretamente, devemos levar nossa aplicação para o ambiente de produção na Amazon através do *Elastic Beanstalk*, para que isso seja possível o primeiro passo é levar a imagem com o perfil de produção para um repositório o qual a Amazon consiga realizar o acesso da imagem, utilizaremos o Docker Hub.

Assim como fizemos no ambiente de desenvolvimento, vamos passar para os containers que serão configurados no servidor da Amazon as informações de conexão ao banco de dados, bem como o acesso ao bucket através das variáveis de ambiente. Uma vez que essas variáveis de ambiente serão do sistema operacional, precisamos mudar os métodos para `System.getenv` e não mais `System.getProperty`. Para isso, devemos alterar as seguintes classes:

JPAConfiguration

Pacote: *br.com.casadocodigo.loja.configuration*

```
@Bean
@Profile("prod")
public DataSource dataSourceProd() {
    DriverManagerDataSource dataSource = new DriverManagerDataSource(
        dataSource.setUsername(System.getenv("USUARIO"));

    dataSource.setPassword(System.getenv("SENHA"));
    dataSource.setUrl(System.getenv("JDBC_CONNECTION_STRING"));
    dataSource.setDriverClassName("com.mysql.jdbc.Driver");
    return dataSource;
}
```

[COPIAR CÓDIGO](#)

AmazonConfiguration

Pacote: *br.com.casadocodigo.loja.configuration*

```
else {
    access_key = System.getenv("ACCESS_KEY");
```

```
secret_key = System.getenv("SECRET_KEY");  
}
```

[COPIAR CÓDIGO](#)

FileSaver

Pacote: *br.com.casadocodigo.loja.infra.prod*

```
@Profile("prod")  
public String writeProd(MultipartFile file) {  
    try {  
        String bucket=System.getenv("BUCKET");
```

[COPIAR CÓDIGO](#)

Feito isso, altere o perfil do projeto para **prod**. Vá até a classe **ServletSpringMVC** presente no pacote *br.com.casadocodigo.loja.configuration*

```
@Override  
public void onStartup(ServletContext servletContext) throws ServletException {  
    super.onStartup(servletContext);  
    servletContext.addListener(RequestContextListener.class);  
    servletContext.setInitParameter("spring.profiles.active", "prod");  
}
```

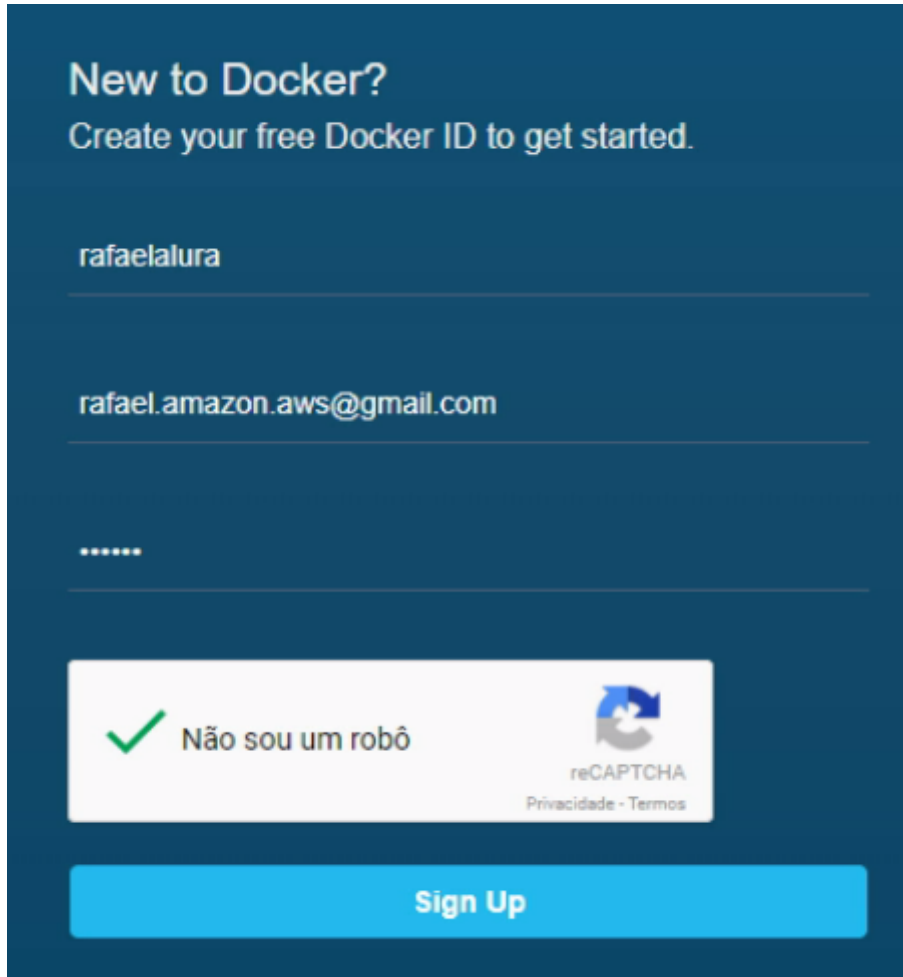
[COPIAR CÓDIGO](#)

Agora que fizemos os devidos ajustes para o ambiente de produção, vamos exportar o projeto para um arquivo **war**. Para isso, clique com o botão direito do mouse no projeto e selecione a opção **Export** e escolha a opção *WAR file*. Escolha a pasta de destino como sendo o **Beanstalk_prod** e salve o arquivo como sendo **ROOT.war**

Uma vez que salvamos o projeto na pasta **Beanstalk_prod**, falta construirmos a imagem customizada com o perfil de produção para enviar ao *Docker Hub*, vamos

então ao [Docker Hub](#) criar um repositório para receber a imagem customizada com a aplicação da Casa do Código.

O primeiro passo no Docker Hub será criar uma conta, caso ainda não tenha feito. Para isso, coloque as informações do nome de identificação, e-mail, senha e clique no botão do Recaptcha, para finalizar clique em **Sign Up**



Ao clicar no botão **Sign Up**, o Docker Hub deverá enviar um e-mail pedindo confirmação, podemos clicar no botão **Confirm Your Email**



Please confirm your email address

You have created a Docker ID with the username: rafaalgcp

Confirm Your Email

(This link will expire in 2 days.)

Clicando nesse link, devemos ser redirecionados para a parte de **Log in** no Docker Hub, coloque seu e-mail e senha e na parte de seu perfil crie um repositório clicando no botão **Create Repository**

Welcome to Docker Hub

Here are a few things to get you started.



Create Repository

Na próxima tela de configuração, dê o nome para o repositório como sendo **casadocodigo** e clique no botão **Create**

rafaelalura

casadocodigo

Short Description (100 Characters)

Full Description

Visibility

public

Create

Agora que já temos o repositório criado, só falta criar a imagem customizada da Casa do Código com o perfil de produção. Vimos que o arquivo Dockerfile que fizemos no ambiente de desenvolvimento, atendeu nossa necessidade, copie esse mesmo arquivo e leve para a pasta **Beanstalk_producao** onde já temos o arquivo **ROOT.war**. Para construir a imagem customizada abra o terminal no Linux ou no Mac ou o Docker Quickstart Terminal no Windows, mude os diretórios até chegar a pasta **Beanstalk_producao** e digite:

```
docker build -t "[DOCKER ID]/casadocodigo:v1-prod" .
```

[COPIAR CÓDIGO](#)

Para que possamos enviar a imagem para o repositório no Docker Hub, devemos primeiramente informar nossas credenciais:

```
docker login
Username: [DOCKER ID]
Password: [MINHA SENHA]
```

[COPIAR CÓDIGO](#)

Feito isso, envie a imagem para o repositório que configuramos no Docker Hub com o comando:

```
docker push [NOME DA IMAGEM]
```

[COPIAR CÓDIGO](#)

Posteriormente, vá até o repositório no Docker Hub. Qual é o resultado? A imagem está no repositório?