

Push da imagem para Docker Hub

Transcrição

O script que o instrutor segue durante a aula é o seguinte:

```
# Instalar o plugin: Parameterized Trigger

# Modificar o Job para startar com 2 parametros:
# Geral:
Este build é parametrizado com 2 parametros de string
Nome: image
Valor padrão: <seu-usuario-no-dockerhub>/django_todolist_image_build

Nome: DOCKER_HOST
Valor padrão: tcp://127.0.0.1:2376

# No build step: Build / Publish Docker Image
# Mudar o nome da imagem para: <seu-usuario-no-dockerhub>/django_todolist_image_build
# Marcar: Push Image e configurar **suas credenciais** no dockerhub

# Mudar no job de teste a imagem para: ${image}
docker run -d -p 82:8000 -v /var/run/mysqld/mysqld.sock:/var/run/mysqld/mysqld.sock -v /var/lib,
```

[00:00] Olá pessoal, tudo bem? Nessa aula a gente vai aprender a trabalhar com parâmetros, como que nós passamos parâmetros de um job, pra outro job. Pra isso nós vamos ter que instalar um novo plugin.

[00:14] Então a gente vem em "Gerenciar o Jenkins", "Gerenciar plugins", "Disponíveis", e nós vamos procurar esse plugin aqui. Esse plugin vai ser o plugin que a gente vai utilizar pra passar parâmetros. Então nós vamos clicar aqui em instalar e assim que a instalação terminar a gente volta.

[00:37] Legal, a instalação acabou. Então vamos voltar lá no nosso job, configurar isso aqui. Então a gente vem em Jenkins, no nosso job, e configurar. Pra esse job, ele em si não precisa de parâmetros pra execução porque ele vai construir a imagem, mas ele vai passar parâmetros pra um próximo de job que é o que vai subir a aplicação em dev. Pra isso eu preciso pré-definir esses parâmetros agora.

[01:06] Como é que eu faço? Nós clicamos aqui, "Este build é parametrizado", e nós vamos criar dois parâmetros pra ele. O primeiro parâmetro vai ser o seguinte: parâmetro string, o nome dele vai ser image, que vai ser a imagem que nós vamos construir. Se nós olharmos aqui embaixo, nós temos o nome de uma imagem na hora que nós construímos o container. Então a gente vai copiar essa imagem aqui e vai passar pra cá.

[01:44] Colamos a imagem aqui e nós temos que fazer uma alteração nesse valor padrão, porque? Agora a gente vai aprender a registrar essa imagem lá no Docker Hub. Pra isso, primeiro vamos configurar o valor aqui do nome da imagem, no nosso caso vai ser aluracursos/ o nome da imagem. Vamos colocar ele aqui em minúsculo.

[02:15] Nós vamos, na próxima aula, aprender a configurar esse repositório. Feito isso, nós vamos adicionar um outro parâmetro que é um parâmetro de string pra que eu passe pro próximo job. Esse parâmetro vai ser DOCKER_HOST, isso é só

para garantir que o próximo job execute no mesmo servidor que o primeiro job executou os builds. Então, pra isso, nós passamos o valor tcp e o IP do servidor. E agora nós clicamos em aplicar.

[02:55] Bom, salvamos agora os parâmetros. Nós temos que fazer duas configurações pra enviar essa imagem pro registro agora. A primeira é, aqui no processo de build da imagem, nós vamos marcar uma opção chamada Push image e ele vai jogar essa imagem pra um registro. Pra qual o registro? Depende das credenciais que a gente utilizar.

[03:19] Então nós vamos vir aqui, vamos clicar aqui em Add dentro de Jenkins. E aqui é muito simples, ele vai pedir um usuário e uma senha, então nós vamos colocar o usuário aluracursos e, nesse nosso caso, a senha é inserida aqui de acordo com a configuração da sua conta lá no Docker Hub. Desse jeito, vamos colocar assim o ID dele como dockerhub.

[03:55] Por padrão, o job vai fazer o push da imagem direto pro Docker Hub. Como que ele sabe que é direto pro Docker Hub? Porque o daemon que tá rodando lá no meu servidor, onde o Docker foi exposto, tá configurado pro registro padrão que é o Docker Hub. Se você na sua empresa configurou o seu docker pra fazer o push pra um outro registro, um registro privado por exemplo, automaticamente ele vai pro registro privado já que eu tô acessando o Cloud do Docker instalado.

[04:24] Então, feito isso, eu vou dar uma description aqui chamada dockerhub, só pra gente colocar um valor e adicionar. E agora, se eu clicar aqui no dropdown, eu tenho o aluracursos pro Docker Hub.

[04:41] Então a última configuração que a gente tem que fazer, depois de colocar as credenciais, é mudar a imagem que vai ser aluracursos/ o nome dessa imagem. Por que a gente faz isso? Porque, basicamente, eu tô indo ao Docker Hub registrar uma imagem, se eu não colocar qual é o usuário e debaixo de qual usuário ele não funciona.

[05:04] Então agora a gente dá um salvar e manda construir com parâmetros. Ele traz os parâmetros default pra gente, que o próximo job vai utilizar, e eu dou um construir. Ele tá agendando a construção do job e, assim que a gente terminar o build, a gente volta.

[05:25] Legal, ele acabou o build da imagem. Vamos dar uma olhadinha lá no Docker Hub pra ver imagem registrada? Então a gente vai em "hub.docker.com/u/aluracursos". Então ele abriu aqui o Docker Hub e eu tenho a imagem que a gente acabou de fazer o push dela.

[05:47] Qual é a vantagem agora? Qualquer um dentro da minha companhia, se o registro for privado, ou nesse caso qualquer pessoa que tenha acesso ao Docker Hub consegue fazer o pull dessa imagem.

[05:58] Legal, agora na próxima aula nós vamos configurar um novo job que vai fazer o run dessa imagem pro desenvolvedor poder acessar a aplicação. A gente se vê lá.