04

Faça o que eu fiz na aula

Bom pessoal, agora vamos configurar o build do nosso projeto, os seja, vamos acabar com aquele papo de "na minha máquina funciona"! Como no passo anterior fizemos um build de uma imagem igual a de produção, vamos configurar tudo o que precisamos para que nosso projeto funcione, como conexão a banco de dados, migrations etc.

Até esse momento, toda vez em em que fazemos um push no git, o gitlab está disparando os passos da CI utilizando uma imagem própria. Até então não temos controle do que está instalado nela, ou se ela tem todos os pacotes que precisamos para realizarmos os testes. Então vamos configurar e disponibilizar para o gitlab nosso próprio executor com tudo o que precisamos. Para isso, precisaremos utilizar o gitlab-runner, então vamos deixar o trabalho de build e teste para esse runner e assim que tiver terminado, ele envia os dados para o gitlab.

Para isso precisaremos subir em nossa máquina um container com o gitlab-runner. Vamos então fazer um pull da imagem do gitlab-runner e em seguida disponibilizar um container dessa imagem.

```
docker pull gitlab/gitlab-runner:latest
```

Com a imagem disponível em nossa máquina vamos subir um container e deixar o serviço de pé

```
docker run -d --name gitlab-runner --restart always -v /Users/Shared/gitlab-runner/config:/etc/gitla
```

Um detalhe nesse comando são com os volumes que estou montando, em máquinas linux utilize o diretório " `/src` " pois o diretório " `Users/Shared` " não existe. Caso você esteja trabalhando em máquinas linux, quando montamos um volume temos que criar no diretório `/src` , pois o diretório `Users/shared` não existe

Se rodarmos em nosso terminal um " `docker ps` ", podemos verificar que agora temos um container gitlab-runner de pé, pronto para uso.

Agora que temos um container gitlab-runner rodando em nossa máquina, vamos configurá-lo e sincronizá-lo com o gitlab. Com isso feito, podemos garantir que o passo que precisarmos dar depois terá como base um ambiente semelhante ao de produção.

Para isso precisamos registrar nosso runner no gitlab. Vamos acessar no gitlab a área responsável pelos "runners" no menu settings-> CI/CD

Clicando no menu Runners, podemos ver que temos runners compartilhados, e runners específicos. Para criarmos nosso próprio runner, vamos precisar copiar uma chave de segurança que está disponível no menu:

Com a chave copiada, vamos agora sincronizar nosso gitlab runner com o gitlab. Para isso, vamos entrar pelo terminal no container do gitlab-runner e vamos registrar o runner no gitlab. Digite então no terminal o comando para entrar no container:

```
docker exec -it gitlab-runner bash
```

Após digitar esse comando, estamos dentro do container gitlab-runner. Agora precisamos registrar um serviço para que os passos do gitlab-ci sejam executados pelo nosso runner. Para isso digite no terminal:

```
gitlab-runner register
```

Ao digitar o comando de register, você precisará informar o endereço do gitlab que está utilizando:

```
https://gitlab.com/
```

Em seguida, precisará digitar a chave de segurança que copiou do gitlab.

Com a chave informada, vamos dar um nome para esse runner: vou chamá-lo de "meu-runner". Em seguida, podemos adicionar tags ao runner que será utilizada para vincular os jobs a esse executor, então vamos chamá-lo de “runner-build ” e dar um enter para próxima etapa. Precisamos agora informar ao registro que tipo de runner queremos registrar. Aqui podemos fazer runners do tipo kubernetes, parallels, shell, ssh, virtualbox, docker, docker-ssh, docker+machine e docker-ssh+machine, mas para nosso cenário vamos registrar um do tipo docker. Para isso, basta digitar `docker` no terminal e em seguida informar qual a imagem que pretendemos utilizar.

Como criamos uma imagem docker igual ao ambiente de produção, no passo de build da imagem vamos utilizá la em nosso runner. Dessa forma garantimos que o executor tenha as mesmas configurações de produção. Para isso informe o nome da imagem que criamos " `jnlucas/minha-imagem:latest` "

Pronto, nosso runner está configurado!