DigitalHouse>

Certified Tech Developer
The Ultimate Degree

Infraestrutura II

Configurando nosso servidor para Deploy - Parte 7

É hora de completar nossa pipeline, finalizando o processo de Deploy.

Objetivo final da prática

- Proteger nossa branch principal: Main
- Atualizar nossa pipeline com o Job de deploy
- Abrir uma MR para main e executando a Pipeline
- Liberar a porta 8080 na nossa EC2
- O resultado

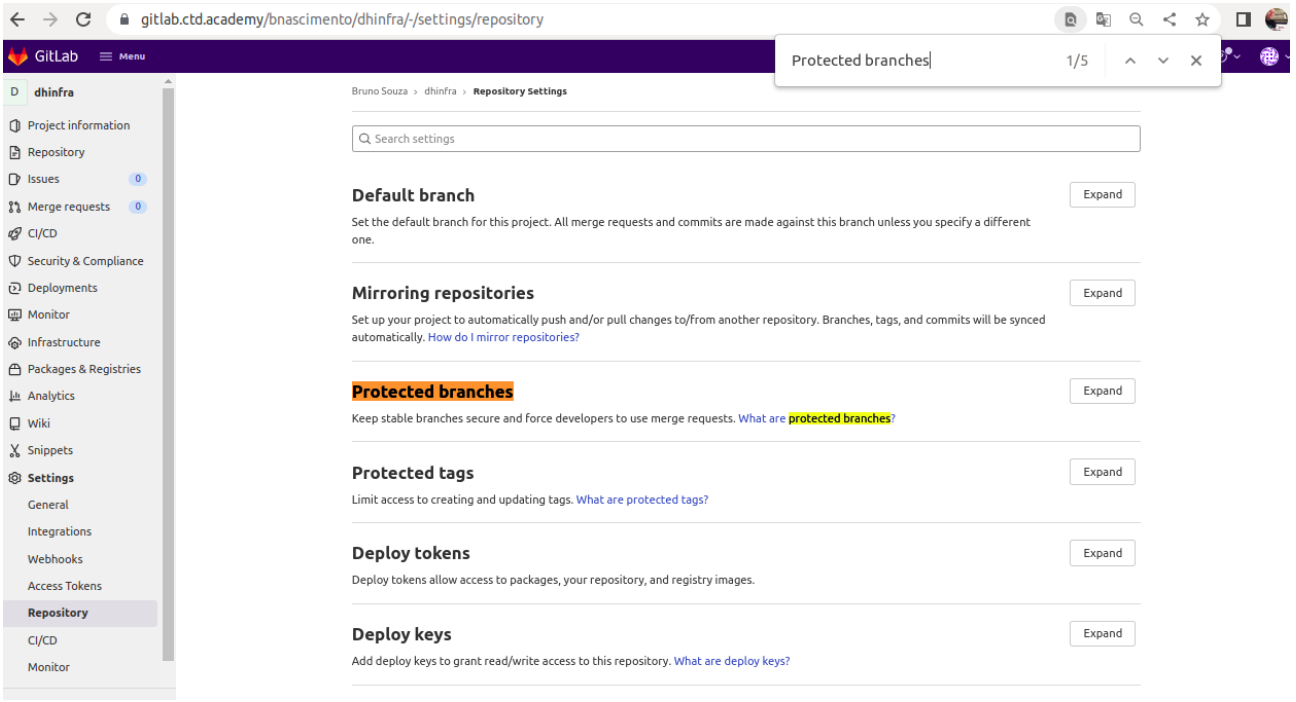
Mãos à obra!

Na aula anterior criamos todo nosso ambiente e configuramos nosso gitlab, agora precisamos modificar nossa pipeline para ter a ação de deploy

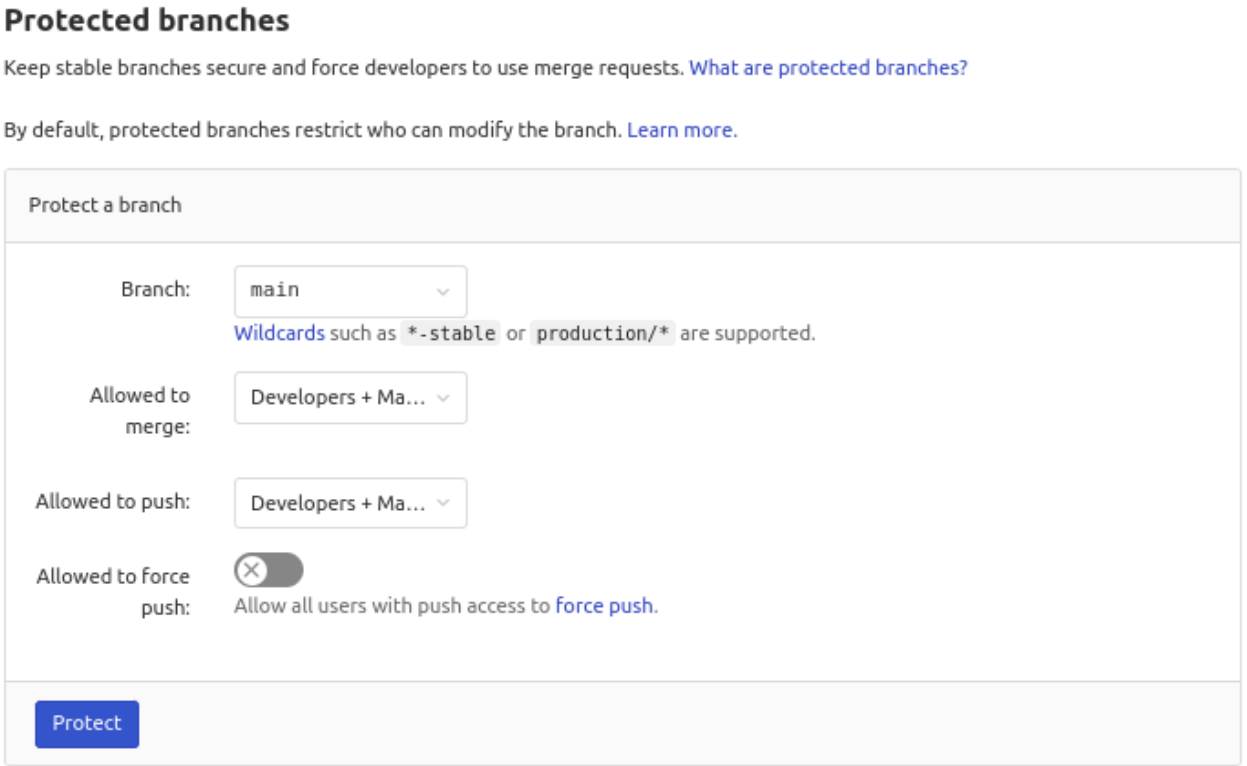
Proteger nossa branch principal: Main

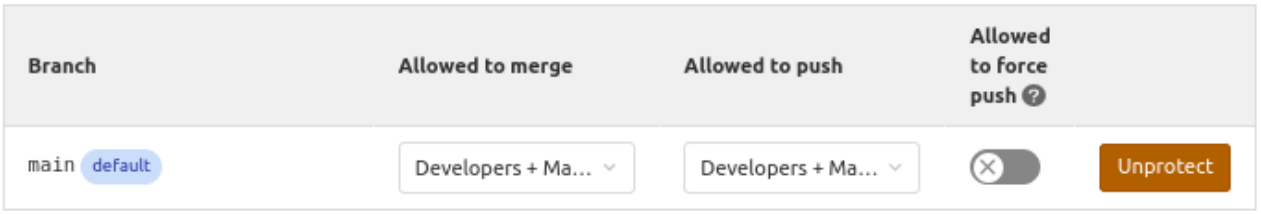
Vamos proteger a main para receber merges somente com Merge Requests para aumentar a curadoria e confiabilidade do nosso código

- Para isso, vamos no nosso repo **dhinfra** > settings > repository e clicar em **“Protected Branches”**



- Em seguida vamos expandir a opção, preencher o form e clicar em **“Protect”**





Atualizar nossa pipeline com o Job de deploy

Agora vamos adicionar o step de deploy no nosso **.gitlab-ci.yml**

Importante: alterem e já comitem para branch main

- Adicionar o stage

```
image: maven:3.6.3

stages:
 - build
 - test
 - package
 - deploy
```

- Configurar o Stage

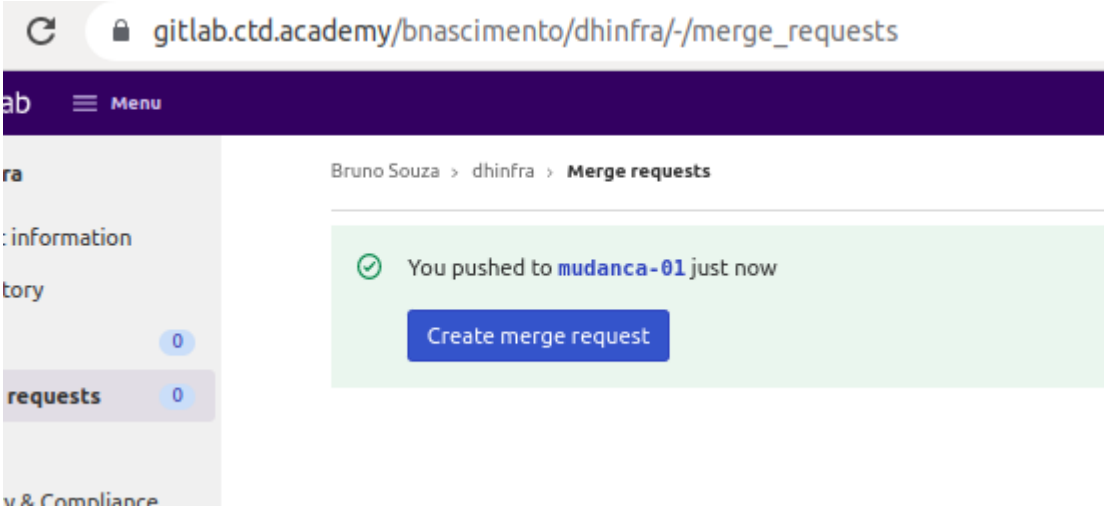
```
deploy_job:
   stage: deploy
   image: alpine:3.11
   before_script:
         - apk update && apk add openssh-client bash # instalando o
client openssh para permitir conexao
       - mkdir -p ~/.ssh
       - eval $(ssh-agent -s) # habilitando o ssh agent
        - echo "$SSH_PRIVATE_KEY" | tr -d '\r' | ssh-add - #cadastrando
nossa key
       - touch ~/.ssh/config
             -  echo  -e  "Host  *\n\tStrictHostKeyChecking  no\n\n"  >
~/.ssh/config #permitindo acesso a qualquer  host sem pré verificação
(yes/no)
   script:
```

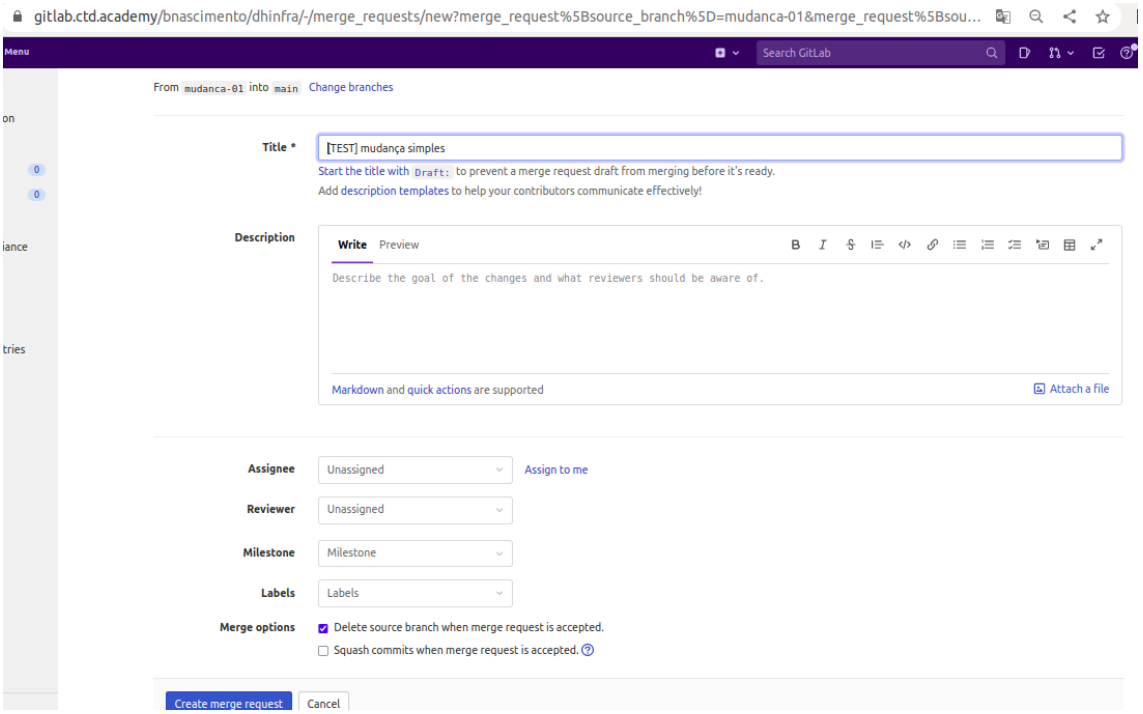
```
      - echo "Deploying started..."
      - echo "[0] Copying new jar to Server...."
                        -    scp    ./target/dhinfra-1.0-SNAPSHOT.jar
ubuntu@$BACKEND_PROD_IP:~/ #copiando nosso .jar para a /home/ubuntu do
servidor
      - echo "[1] Stop old process"
        - ssh ubuntu@$BACKEND_PROD_IP "sudo systemctl stop dhapp" #
pausando o nosso pequeno servidor
      - echo "[1] Start process"
        - ssh ubuntu@$BACKEND_PROD_IP "sudo systemctl start dhapp" #
iniciando o nosso pequeno servidor novamente
      - echo "Finished deploying the app."
  only:
      - main #permite a execução somente na branch master
```

Abrir uma MR para main e executando a Pipeline

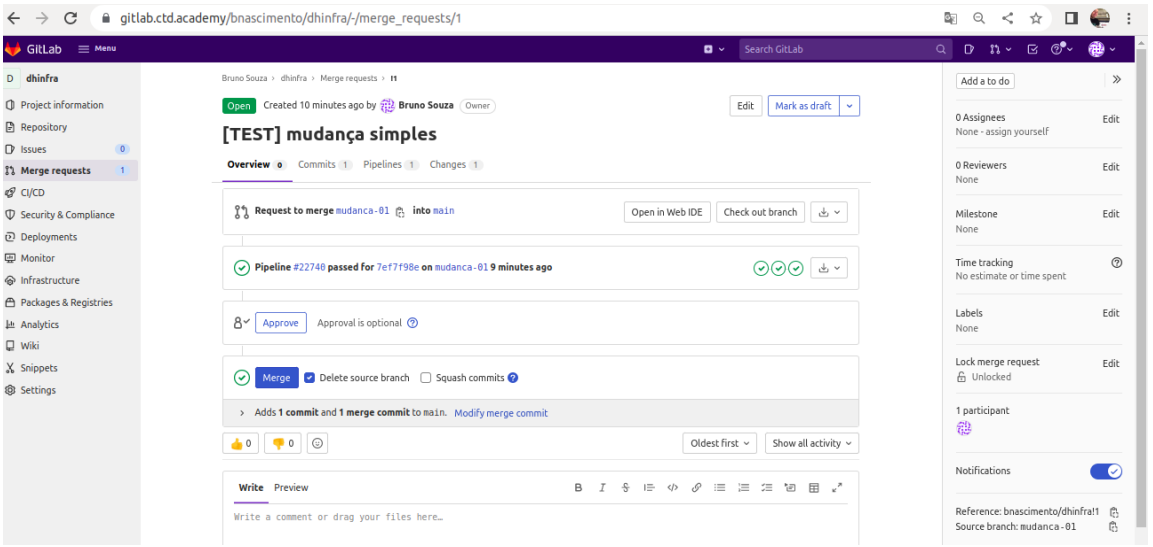
Agora devemos fazer uma leve modificação no nosso código, podendo ser no próprio Readme.md, porém em uma branch diferente

Para isso:

- Crie uma branch chamada de **mudanca-01**
- Faça qualquer alteração simples (pode ser textual no readme, comentários no código, ...)
- Abra um Merge Request da **mudanca-01** para **main**
  - 



- Aprove o seu próprio merge request, clicando em **“merge”**



- Observe a pipeline rodar

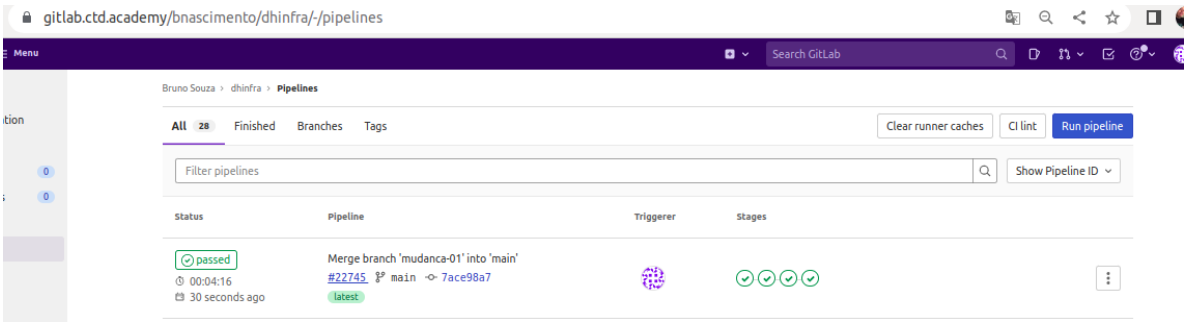
Liberar a porta 8080 na nossa EC2

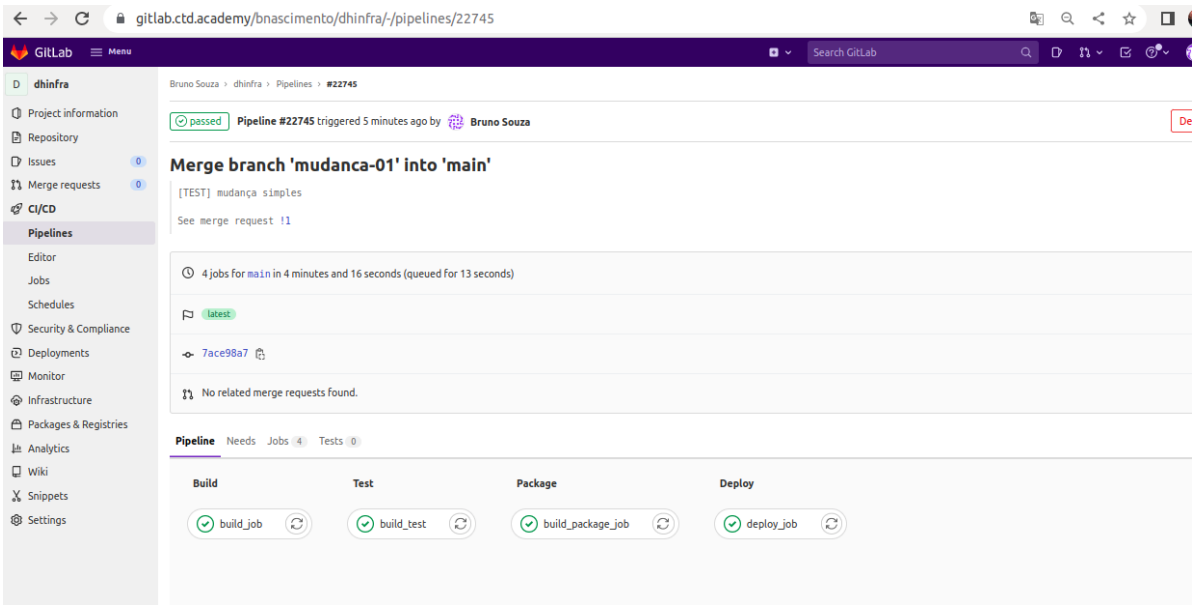
Missão:

Para testar o resultado da nossa aplicação é necessário liberar a porta 8080 no security group associado na noassa EC2. Você pode fazer isso via Terarform (IaC) ou via Console WEb da AWS

O resultado

Se tudo estiver correto, então:

- Sua pipeline tem que estar assim:
  - 

  - 

- Seu servidor, acessado via web

- Seu servidor, via console: