Notícias

Desenvolvimento e Deploy

Plataformas de Desenvolvimento

2020/2021

João Fernandes, nº21250698

João Simões, nº21250567

Índice

1. Introdução

Este trabalho foi realizado no âmbito da unidade curricular Plataformas de Desenvolvimento, do Mestrado em Engenharia Informática do ramo de Engenharia de Software. O trabalho consistiu em desenvolver uma Rest Api que fizesse operações CRUD a uma base de dados, criação de uma interface gráfica para criar a ligação com a Rest Api e por fim fazer o *build* das duas aplicações, enviar as imagens docker criadas para um repositório público e por fim fazer o deploy do sistema completo numa máquina.

2. Arquitetura

O pensamento inicial foi o que podíamos fazer de maneira fácil e rápida para cumprir com os requisitos pedidos para o trabalho. Portanto foi escolhido para implementar a Rest Api o NodeJS, para a interface da aplicação React, para a build dos dois projetos pipelines do Jenkins e para fazer o deploy da aplicação o ansible. A Figura 1, representa a arquitetura do sistema o qual vai ser explicada no próximo parágrafo.

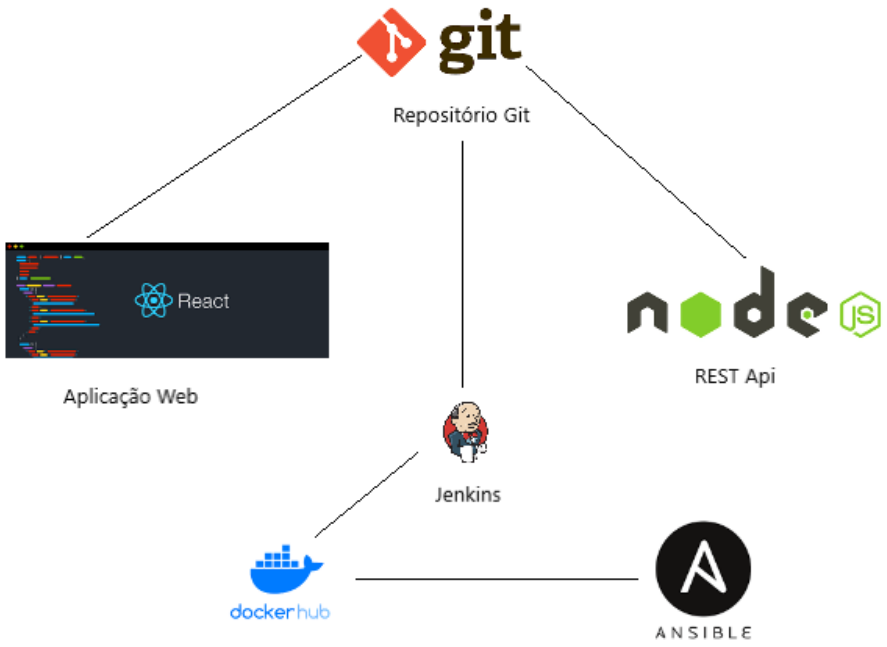


Figura 1 Arquitetura do Sistema

O sistema tem um repositório em que estão as duas aplicações a aplicação web e a Rest Api, de seguida o Jenkins o que faz é ligar-se ao repositório e fazer o build das aplicações e criar as suas respetivas imagens e de seguida fazer o push delas para o repositório criado no docker hub. No fim é utilizado o ansible para fazer o deploy da nossa aplicação no qual faz uso do docker hub, para obter as imagens respetivas.

3. Aplicação

3.1. RestAPI

Foi usado Node.js para desenvolver a Rest Api, pois foi fácil de instalar e interligar com a base de dados. Para dar suporte para a aplicação web e esta consiga fazer as operações CRUD, foi criado então 4 *endpoints*, em que cada um corresponde a uma operação CRUD, acedida através do endpoint geral '***/api/noticias***', no porto 4000. Na X podem ver a lista dos endpoints criados como o tipo de pedido é feito

HTTP	Endpoint	Descrição
GET	/api/noticias	Retorna todas as notícias, armazenadas que possam ser vista.
POST	/api/noticias	Insere uma notícia na base de dados.
PUT	/api/noticias	Altera atributos de uma notícia especifica.
DELETE	/api/noticias	Apaga uma notícia especifica.

3.2. Frontend

Neste tópico será feito a apresentação da aplicação web, em que se poderá ser visto as páginas criadas para o fazer a gestão de notícias e que fazem ligação com a Rest Api. Na Figura 2 pode se ver a página inicial da aplicação em que irá mostrar a listagem de todas as notícias ativas.

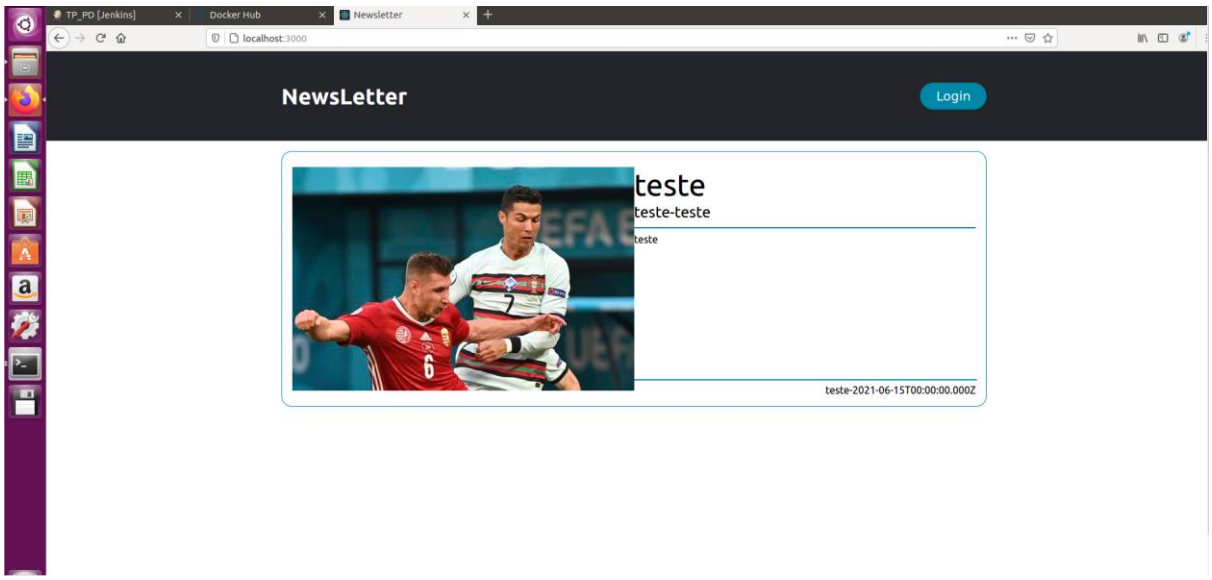


Figura 2 Página Inicial

Na Figura 3 é demonstrado a página de login para aceder à zona de gestão de notícias bastará inserir as credenciais em que o utilizador é admin@news.com e a palavra-passe adminnews.

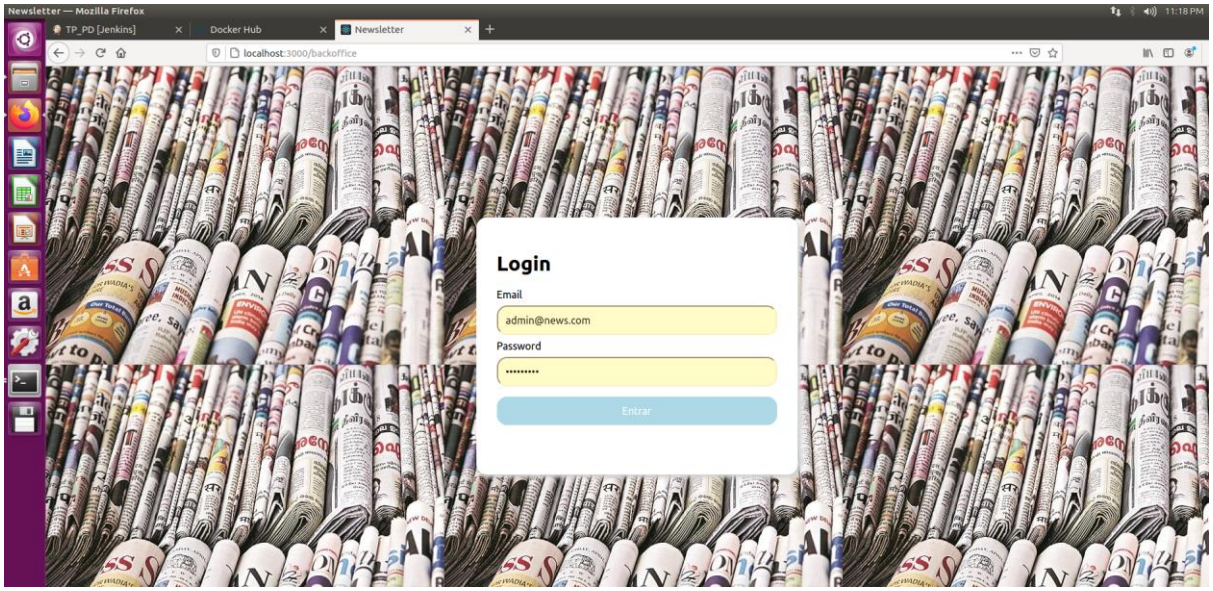


Figura 3 Página Login

Na Figura 4 irá mostrar todas as notícias numa tabela ao clicar na linha irá ser redirecionado para a página de detalhes. Ao clicar no botão de add new irá ser redirecionado para a página da criação de notícias.

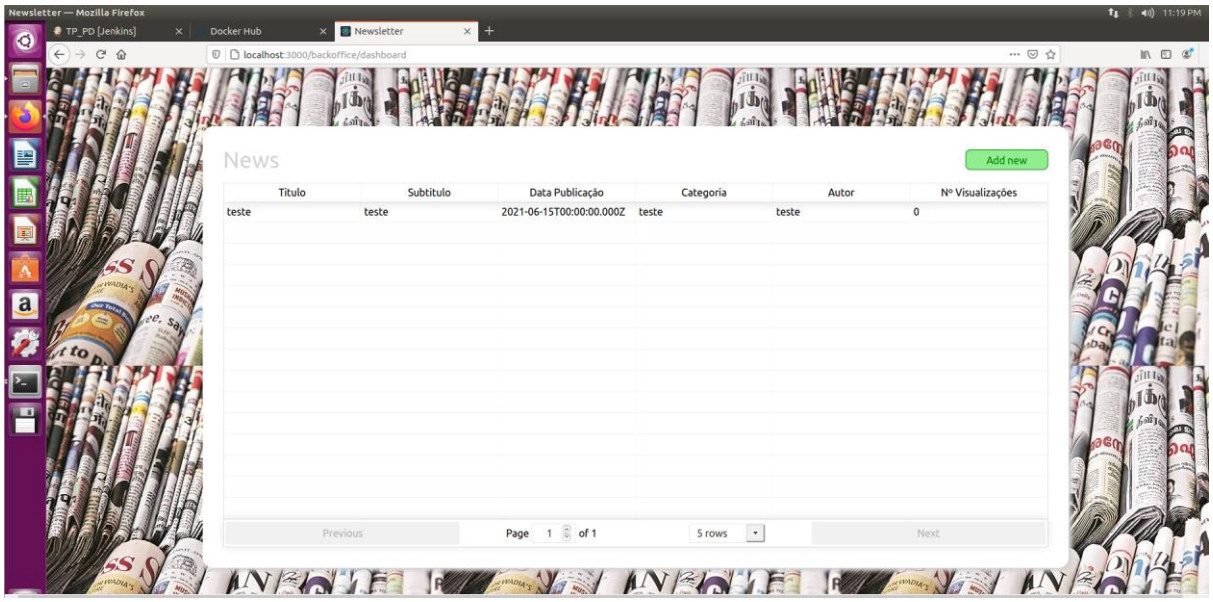


Figura 4 Dashboard

Na Figura 5 será apresentada a notícia escolhida no formulário em que esta pode ser editada ou removida.

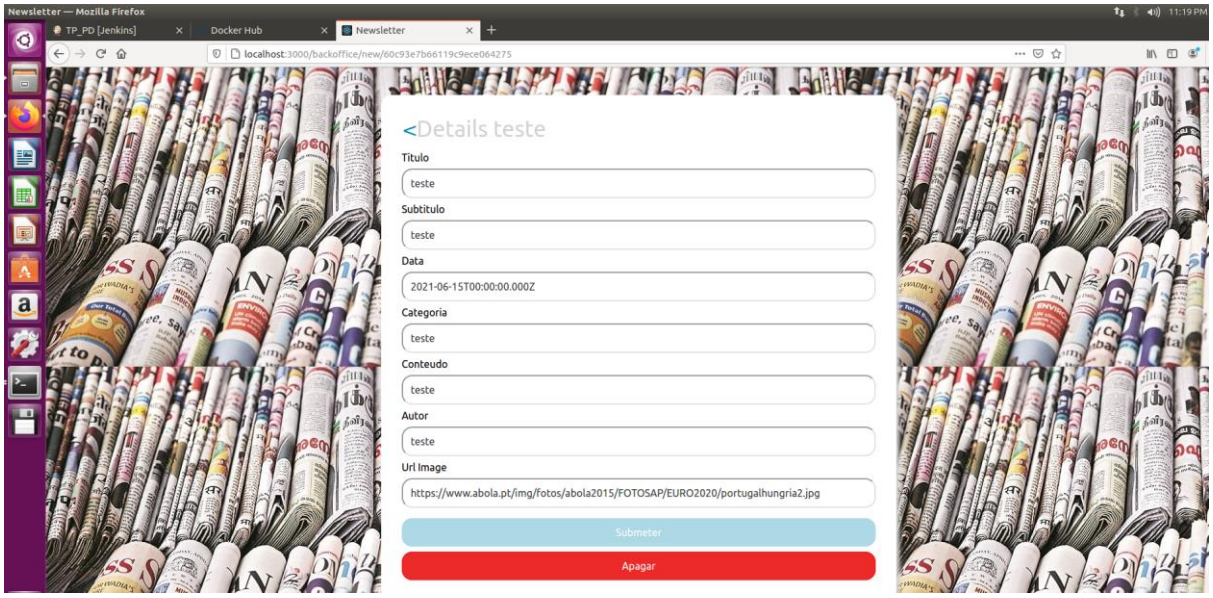


Figura 5 Detalhes de noticia

Na Figura 6, podem verificar o formulário que permite a criação de uma noticia, ao preencher serão redirecionados para a página de de dashboard.

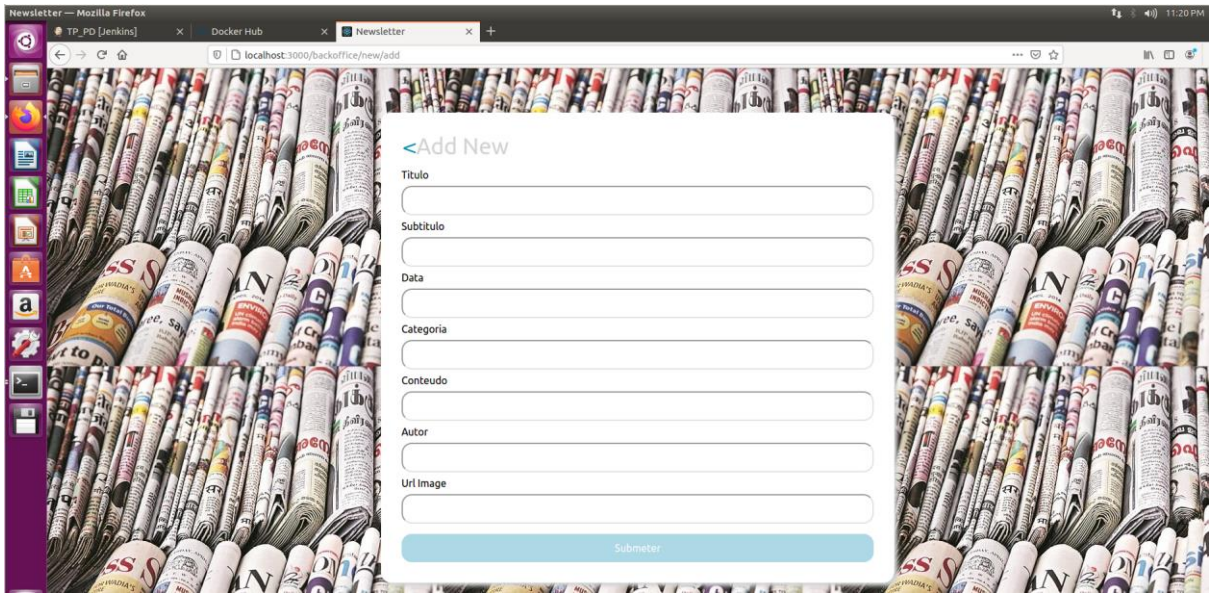
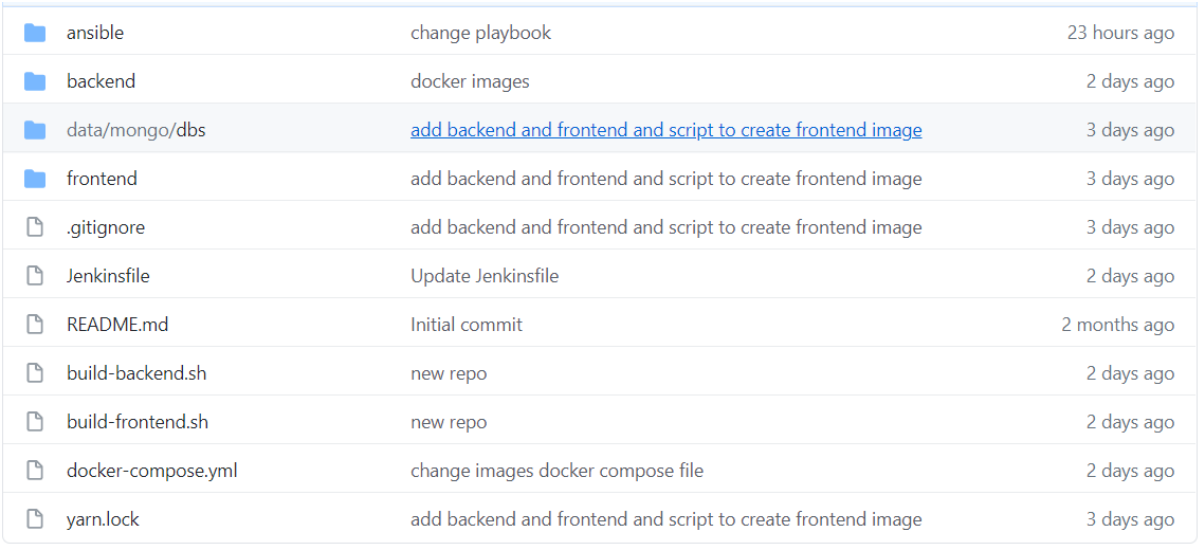


Figura 6 Criar uma noticia

4. Git

Como indicado no capítulo 2, foi utilizado um repositório Git, este com o nome de TP_PD para gerir as nossas aplicações e guarda todos os documentos necessário, como podem ver na seguinte Figura 7.



ansible	change playbook	23 hours ago
backend	docker images	2 days ago
data/mongo/dbs	add backend and frontend and script to create frontend image	3 days ago
frontend	add backend and frontend and script to create frontend image	3 days ago
.gitignore	add backend and frontend and script to create frontend image	3 days ago
Jenkinsfile	Update Jenkinsfile	2 days ago
README.md	Initial commit	2 months ago
build-backend.sh	new repo	2 days ago
build-frontend.sh	new repo	2 days ago
docker-compose.yml	change images docker compose file	2 days ago
yarn.lock	add backend and frontend and script to create frontend image	3 days ago

Figura 7 Repositório Git

Neste parágrafo será explicado o repositório, encontram-se 4 pastas nele, a primeira representa todos os ficheiros que o nosso ansible precisa de usar para fazer o deploy, o segundo contem a nossa aplicação da Rest Api, a terceira é um volume que foi criado para lançarmos a nossa aplicação na nossa máquina de desenvolvimento utilizando o docker-compose como suporte, a última pasta corresponde à aplicação web. A seguir salientar 4 ficheiros que são importantes o primeiro é o 'Jenkinsfile' que é o ficheiro que contêm o script da nossa pipeline, continuando na pipeline é utilizado ainda mais 2 ficheiros o 'build-frontend.sh' e o 'build-backend.sh' que serão executados para criar as imagens de docker para cada uma das aplicações, por ultimo o ficheiro do 'docker-compose.yml', que é onde existe o código para correr todos os containers necessários e colocar o sistema a funcionar, atualmente utilizado para propósitos de desenvolvimento como referido anteriormente.

5. Docker Hub

Para este trabalho foi criado um repositório Docker Hub com o nome de tppd2021, em que contêm todas as imagens necessárias do sistema. Para aceder a este repositório serão aqui colocadas as suas credenciais, que são as seguintes utilizador 'tppd2021´ e palavra-passe 'tppd2021_'. Como podem ver na Figura 8, existem três imagens no repositório a 'tppd2021/docker_jenkins' utilizada para lançar o jenkins ligado já com o docker, a 'tppd2021/web_client' que é a imagem da nossa aplicação e por fim a 'tppd2021/rest_api' que é a imagem da nossa Rest Api.

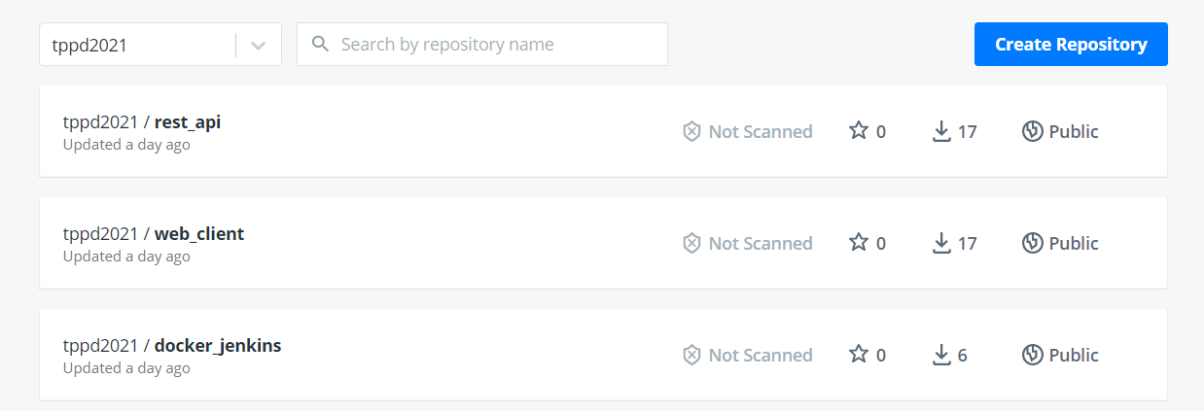


Figura 8 Repositório Docker Hub.

6. Jenkins

Como referido anteriormente em vários capítulos anteriores o Jenkins é utilizado para fazer o build e push das imagens para o repositório do docker hub. Em seguida será explicado os passos que são necessário para instalar o Jenkins na máquina.

6.1. Instalação

Este manual de instruções, vai abordar a instalação do Jenkins e setup da imagem Docker, numa máquina que o sistema operativo seja derivado do Linux.

PASSO 1:

O primeiro passo será instalar o docker na nossa máquina, seguindo os comandos das imagens seguintes.

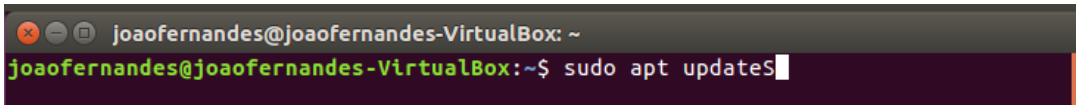


Figura 9 Atualizar a lista de pacotes.

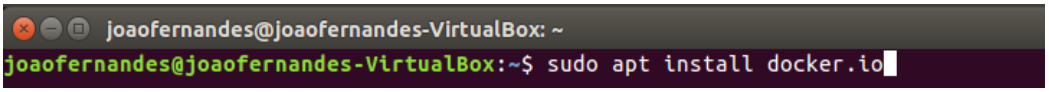


Figura 10 Comando instalar docker

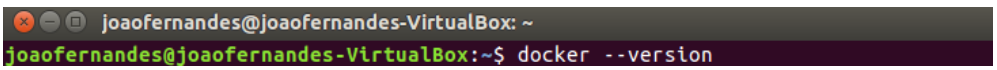


Figura 11 Verificar versão do docker que foi instalado

PASSO 2:

O segundo passo é utilizar a imagem docker que se encontra no repositório do docker hub descrito no capítulo anterior. Para isso será executado um comando que irá correr um comando com essa imagem e fazer a ligação do docker e do jenkins, para que seja possível utilizar o docker na pipeline. O comando a executar é o que se encontra descrito abaixo deste parágrafo.

docker run -p 8080:8080 -p 50000:50000 -v /var/jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock --name jenkins -d tppd2021/docker_jenkins

O Jenkins se tudo correr bem ficará já a correr e à escuta do porto 8080, podendo acedê-lo via localhost. Para verificar que o container docker encontra-se a correr sem erros execute o seguinte comando **docker ps**. Ao aceder ao http://localhost:8080/ irá ser apresentada a página inicial no qual será necessário um token para aceder ao seguinte passo que será a criação de um utilizador e instalação de plugins iniciais. Para ter acesso ao token será seguir o caminho que é indicado na página de duas formas ou a partir do volume criado para o jenkins ou entrando dentro do container do jenkins utilizando o seguinte comando **docker exec -it jenkins bash**.

PASSO 3:

Depois de termos o Jenkins já todo inicializado é necessário instalar um plugin extra que é o NodeJS, para fazê-lo bastará seguir os seguintes passos que se encontram nas imagens a seguir.

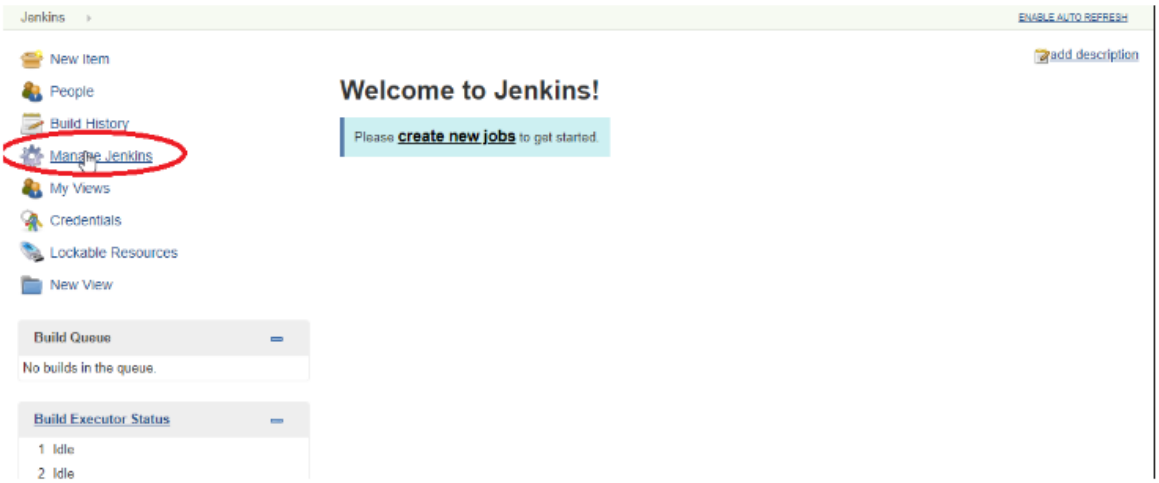


Figura 12 Aceder ao manager do jenkins

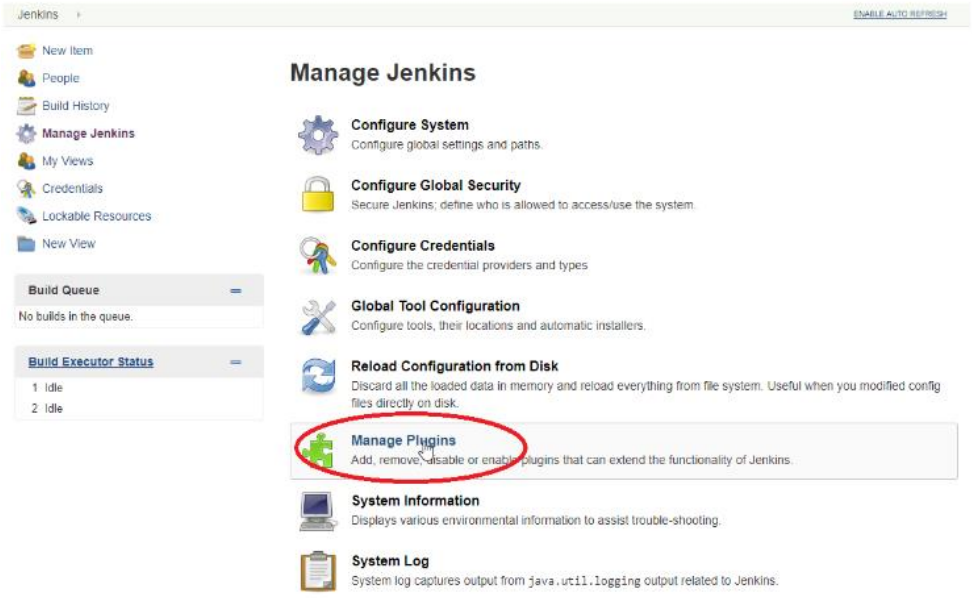


Figura 13 Manager de Plugins

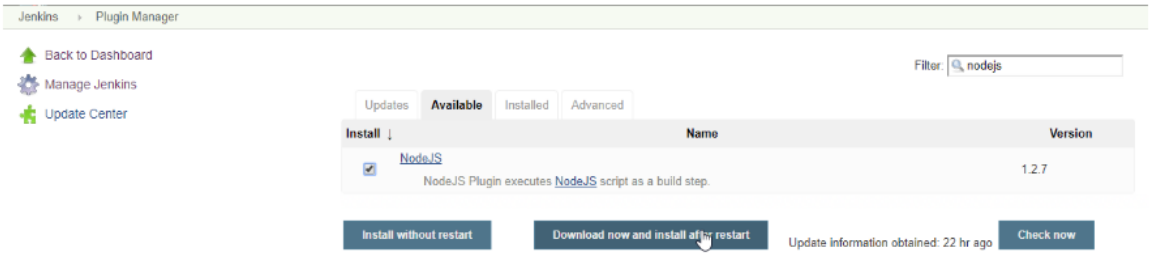


Figura 14 Selecionar o Plugin a tab dos available

PASSO 4:

Após a conclusão do passo anterior, vai ser realizada a criação do pipeline.

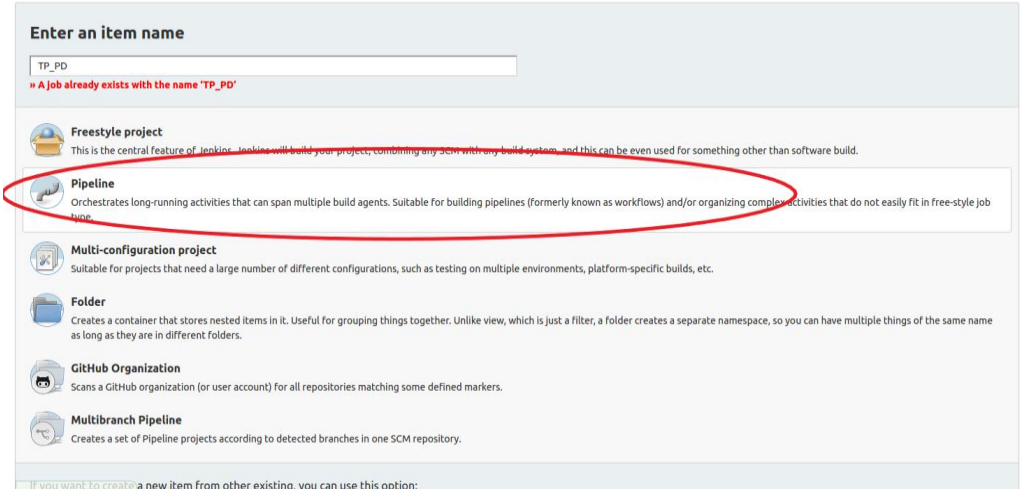


Figura 15 Criação de Pipeline

Deseguida é inserido o repositório URL, https://github.com/joaojsimoes/TP_PD.git,e a branch “main”, na tab Pipeline.

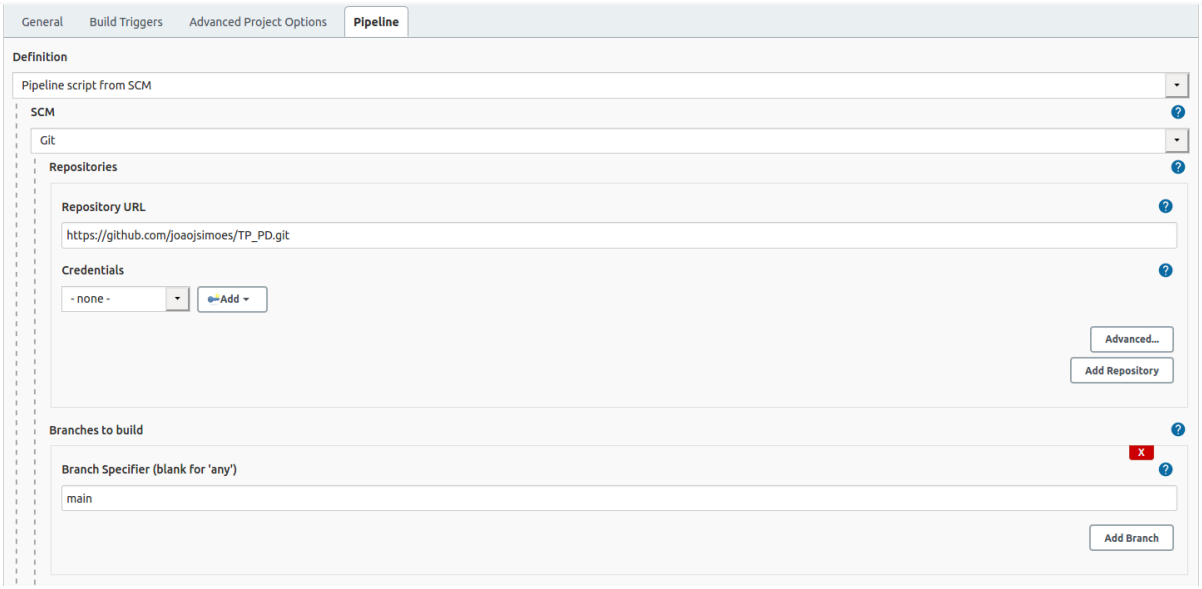


Figura 16 Selecionar o repositório e seleção do ficheiro

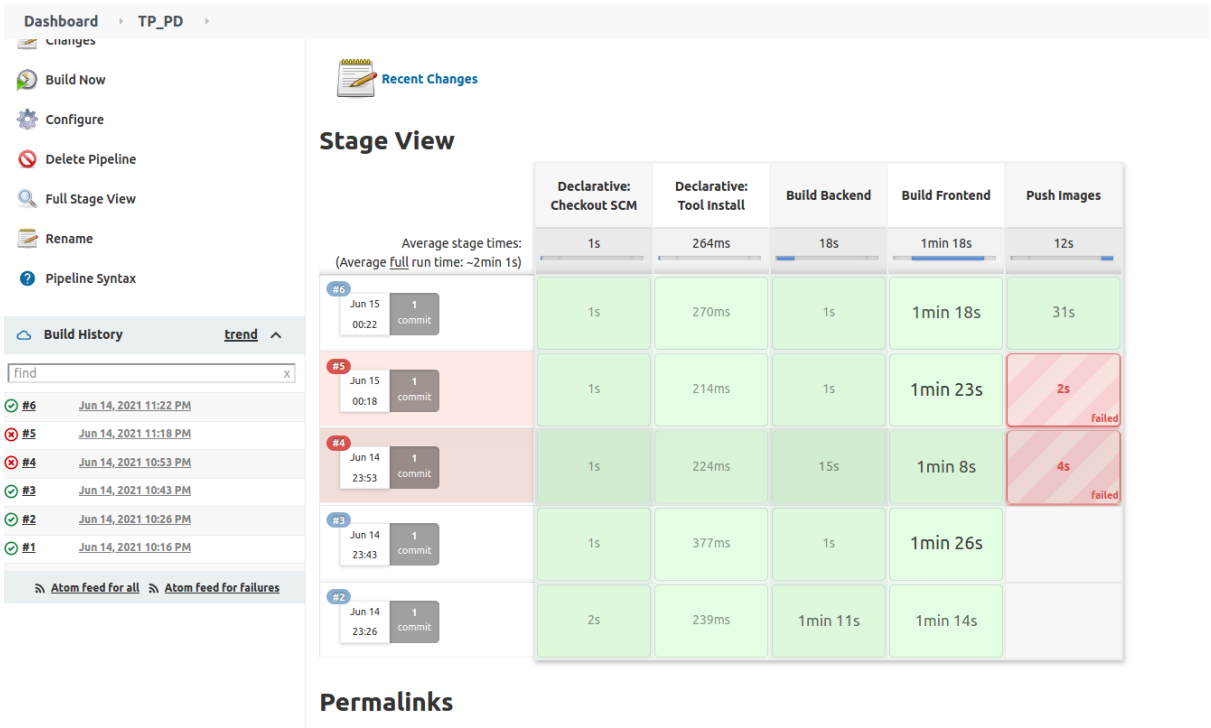


Figura 17 Stage View

Após está fácil e rápida configuração verificamos se o Build é executado com sucesso como se pode ver na Figura 17.

6.2. Pipeline

Nesta parte será descrito o ficheiro de pipeline criado com o nome de Jenkinsfile, como pode ser visto na Figura 18, no inicio foi indicado que será necessário usar o nodejs para conseguir utilizar o npm, de seguida pode foram criados 3 stages em que irão ser para fazer o build da rest api a outra para o build

da aplicação web, o ultimo stage foi criado para que seja possível fazer o push das imagens criadas para o repositório docker hub criado.

```
pipeline {
    agent any
    tools {nodejs "nodejs"}
    stages {
        /*stage('Cloning Git') {
            steps {
                git url:'https://github.com/joaojsimoes/TP_PD',branch: 'main'
            }
        }*/
        stage('Build Backend') {
            steps {
                sh 'sh build-backend.sh'
            }
        }
        stage('Build Frontend') {
            steps {
                sh 'sh build-frontend.sh'
            }
        }
        stage('Push Images') {
            steps {
                sh 'docker login -u "tppd2021" -p "tppd2021_" docker.io'
                sh 'docker push tppd2021/web_client'
                sh 'docker push tppd2021/rest_api'
            }
        }
    }
}
```

Figura 18 Jenkinsfile

7. Deploy Ansible

Neste tópico vai ser referido como foi feito o deploy de toda a aplicação utilizando o ansible, os ficheiros que foram criados e quais os passos que foram tomados.

7.1. Instalação

Para instalar o Ansible basta seguir os seguintes comandos:

$ sudo apt-add-repository ppa:ansible/ansible

$ sudo apt update

$ sudo apt install ansible

O resultado da versão do Ansible deverá ser obtido correndo o seguinte comando:

$ ansible --version

Será também preciso assegurar que o ssh e o daemon sshd estão operacionais:

$ which ssh/usr/bin/ssh

$ which sshd/usr/bin/sshd

Caso não retorne informação poderá ser instalado o ssh da seguinte forma:

$ sudo apt-get install ssh

7.2. Ficheiros de Inventário e Playbook

No ficheiro de inventário, como se pode ver na Figura 19 simplesmente estará indicado o host, o utilizador e a palavra-passe, este ficheiro irá necessitar de ser alterado caso queira correr na própria máquina, pelo menos o user e a palavra-passe.

```
---

all:
  hosts:
    app_machine:
      ansible_ssh_host: "localhost"
      ansible_ssh_user: "joaosimoes"
      ansible_ssh_pass: "guest"
```

Figura 19 Ficheiro inventário

Na Figura 21 e Figura 22, encontra-se o ficheiro de playbook, em que se vai encontrar as tasks que irão correr na máquina passam por instalar coisas necessárias de seguida irá ser feito o pull das imagens necessárias para correr o sistema, sendo elas o mongo, mongo-express, tppd2021/web_client e tppd2021/rest_api, de seguida serão parados todos os containers anteriormente que estavam em execução, depois será removida e criada a network e o volume, por fim serão criados os containers para toda a aplicação. Por fim bastará aceder a dois links o da web cliente que será incluído no http://localhost:3000/ e caso queiram aceder à base de dados e visualizar os dados http://localhost:8081/ .

Para que seja possível correr o playbook bastará correr o seguinte comando **ansible-playbook playbook.yml -K -v -i hosts.yml** em que terá de ser indicado qual o caminho do playbook e o de inventário para onde eles se encontram, ao correr será pedido ao utilizador a palavra-passe dele como se pode ver um exemplo na Figura 20.

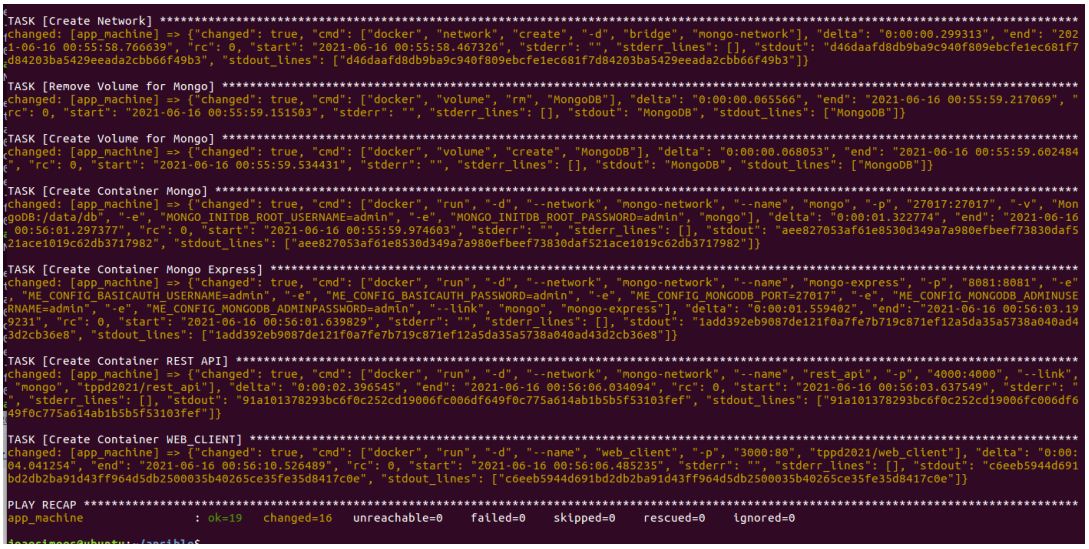


Figura 20 Exemplo de execução do playbook

```yaml
---
- hosts: all
  become: true
  tasks:
    - name: Install aptitude using apt
      apt: name=aptitude state=latest update_cache=yes force_apt_get=yes

    - name: Install required system packages
      apt: name={{ item }} state=latest update_cache=yes
      loop: [ 'apt-transport-https', 'ca-certificates', 'curl', 'software-properties-common']

    - name: Pull Image Mongo
      command: docker pull mongo

    - name: Pull Image Mongo Express
      command: docker pull mongo-express

    - name: Pull Image Web Client
      command: docker pull tppd2021/web_client

    - name: Pull Image Rest API
      command: docker pull tppd2021/rest_api

    - name: Stop and Remove Container Web Client
      command: docker rm -f web_client

    - name: Stop and Remove Container Rest Api
      command: docker rm -f rest_api

    - name: Stop and Remove Container Mongo Express
      command: docker rm -f mongo-express

    - name: Stop and Remove Container Mongo
      command: docker rm -f mongo

    - name: Remove Network
      command: docker network rm mongo-network

    - name: Create Network
      command: docker network create -d bridge mongo-network

    - name: Remove Volume for Mongo
      command: docker volume rm MongoDB

    - name: Create Volume for Mongo
      command: docker volume create MongoDB
```

Figura 21 Ficheiro Playbook Parte 1

```yaml
- name: Create Container Mongo
  command: docker run -d --network mongo-network --name mongo -p 27017:27017 -v MongoDB:/data/db -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=admin mongo

- name: Create Container Mongo Express
  command: "docker run -d --network mongo-network --name mongo-express -p 8081:8081 -e ME_CONFIG_BASICAUTH_USERNAME=admin -e ME_CONFIG_BASICAUTH_PASSWORD=admin -e ME_CONFIG_MO

- name: Create Container REST API
  command: docker run -d --network mongo-network --name rest_api -p 4000:4000 --link mongo tppd2021/rest_api

- name: Create Container WEB_CLIENT
  command: docker run -d --name web_client -p 3000:80 tppd2021/web_client
```

Figura 22 Ficheiro Playbook Parte 2

8. Conclusão

Com a finalização deste trabalho prático, o grupo ganha um maior conhecimento a nível de quando se pretende criar um pipeline para CI/CD. E a configuração das ferramentas envolvidas no processo. A realização deste projeto trouxe algumas dificuldades ao longo do desenvolvimento, pelo desconhecimento e falta de uso de algumas ferramentas usadas. Mas achamos que estes obstáculos encontrados, melhoram o desenvolvimento do trabalho em equipa, como se pode ver no resultado final.