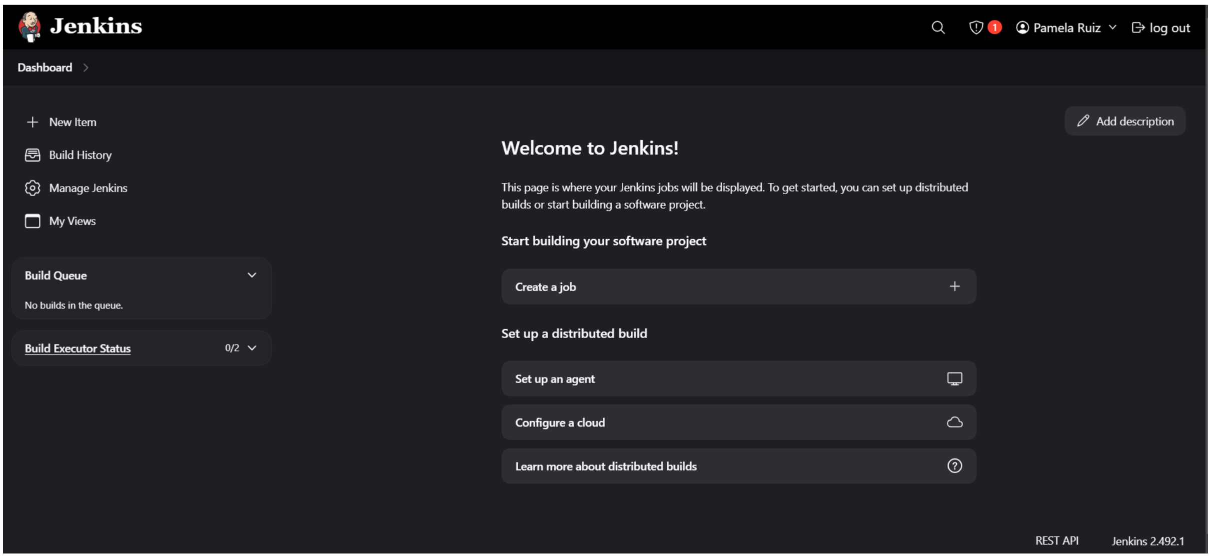
1. I’ve downloaded the installer for windows (.msi)
2. I’ve executed the installer and follow the instructions
3. Jenkins is running in localhost, port 8080



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

Build Queue

No builds in the queue.

4. Once Jenkins is installed, I need to set up my Node.js project. I chose to download the test project from the link provided in the description of the task.
5. The project is saved in the following link https://github.com/PameRuiz/DevOps/tree/main/Jenkins-CI. Jenkinsfile has been added here.

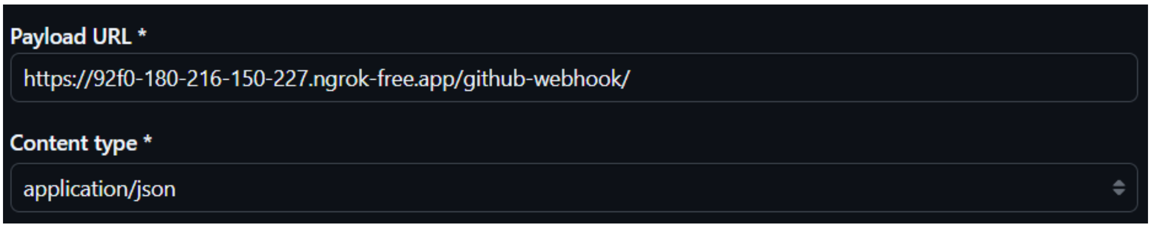
The content of Jenkinsfile is:

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }
        stage('Install') {
            steps {
                 bat 'cd Jenkins-CI & npm install'
            }
        }
        stage('Test') {
            steps {
                bat 'cd Jenkins-CI &  npm test'
            }
        }
    }
}
```

This file is configured to run in any agent available. It is running on my local machine so there is only one agent for the moment.

6. For configuring the webhook I needed to make some changes.
    a. Since I had my Jenkins server running locally. I had to use ngrok for making it accessible from Internet

b. In github I created a new webhook with the following data



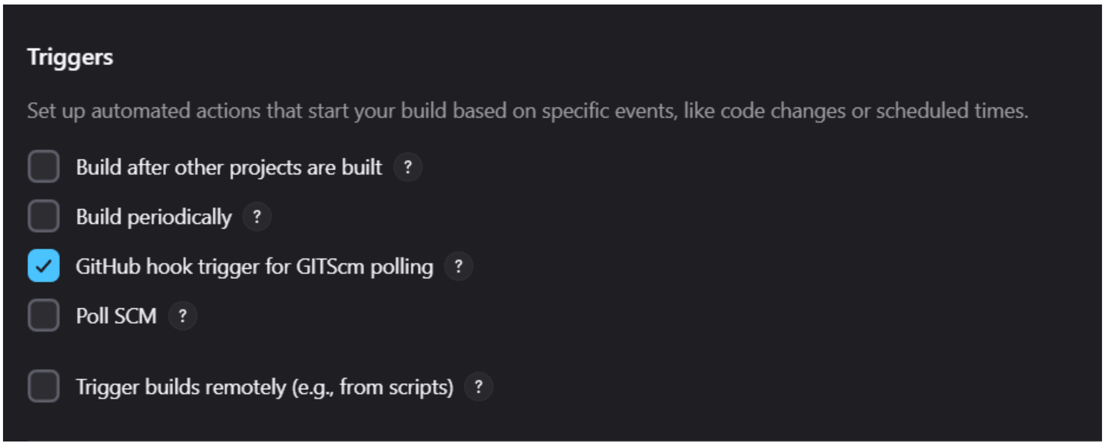
Payload URL *

https://92f0-180-216-150-227.ngrok-free.app/github-webhook/

Content type *

application/json

c. In jenkins I set the configuration in the job for being triggered by changes in github



Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

□ Build after other projects are built ?

□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

□ Trigger builds remotely (e.g., from scripts) ?

d. Once everything was configured I was able to test the trigger by sending changes to the project

7. After configuring Jenkins and the pipeline. I have made a new Jenkinsfile running with docker. The objective was to build a docker image for running my app.
8. For doing this first I prepared a Dockerfile. It was based on an image with alpine and node already installed. I have created my node_modules folder, copied the package.json, run npm install and expose the port where the app will be running

```
FROM node:10-alpine

RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app

WORKDIR /home/node/app

COPY package*.json ./

USER node

RUN npm install

COPY --chown=node:node . .

EXPOSE 3000

CMD [ "npm", "start" ]
```

9. Testing this file locally was possible running the following command **docker build . -t myapp:latest** (this command will be run by Jenkins later)

10. for testing that everything went alright I run docker run -p 3000:3000 myapp with the -d option I am binding the port so I can access from my machine.
11. After building and testing the container the Jenkinsfile will look something like this:

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }
        stage('Install') {
            steps {
                 bat 'cd Jenkins-CI & docker build . -t myapp:latest'
            }
        }
    }
}
```