

DESAFÍO 9

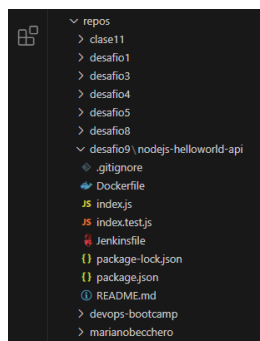
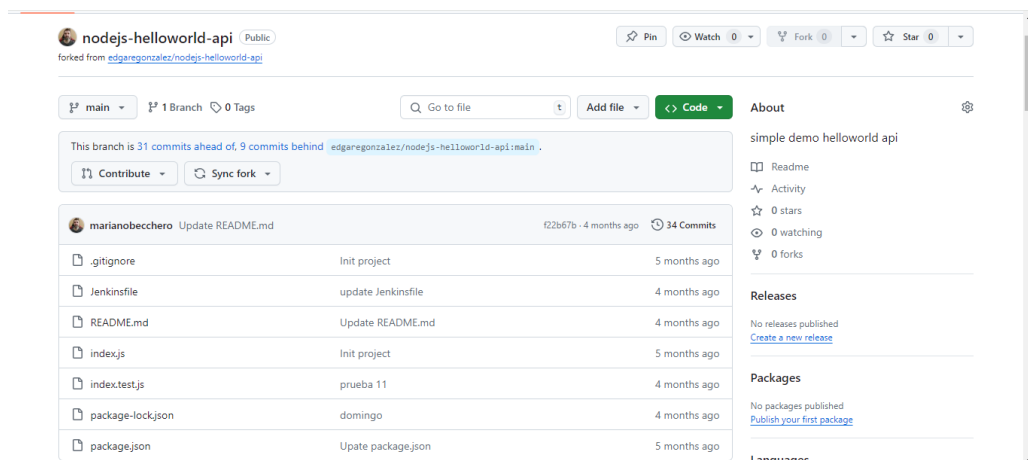
Institución: Educación IT

Alumno: Becchero Mariano

Objetivo: Este desafío tiene como objetivo integrar una práctica que vimos en clase sobre Docker, GitHub Actions y la configuración de la registry de Docker Hub. El objetivo es crear un pipeline completo que realice el build y delivery de nuestra aplicación como una imagen de contenedor y la publique en una registry.

GUÍA DE PASOS

- 1- GitHub: En este paso creé un repositorio a partir de un template de un ejemplo de app de Node.js proporcionado por el profesor. Luego cloné el repositorio a mi computadora personal para realizar los siguientes pasos del desafío.



2- Docker Hub: Creo un repositorio privado en mi cuenta.

Create repository

Namespace: Repository Name:

Short description:

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. [Get more](#)

☐ Public
Appears in Docker Hub search results

☒ Private
Only visible to you

[Cancel](#) [Create](#)

Pushing images

You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to replace `tagname` with your desired image repository tag.

3- Creo un token en Docker Hub

Personal access tokens

You can use a personal access token instead of a password for Docker CLI authentication. Create multiple tokens, control their scope, and delete tokens at any time. [Learn more](#)

[Generate new token](#)

Description	Scope	Status	Source	Created	Last Used
Token Docker Hub	Read, Write, Delete	Active	Manual	Aug 18, 2024 at 12:34:40	Never
Generated through desktop auth...	Read, Write, Delete	Active	Auto-generated	Aug 18, 2024 at 12:30:27	Never

Rows per page: 10 1-2 of 2

4- Añado las credenciales en mi cuenta de GitHub

Environments

☐ Wait timer
Set an amount of time to wait before allowing deployments to proceed.

☒ Enable custom rules with GitHub Apps (Beta)
[Learn about existing apps](#) or [create your own protection rules](#) so you can deploy with confidence.

☒ Allow administrators to bypass configured protection rules
[Save protection rules](#)

Deployment branches and tags
Limit which branches and tags can deploy to this environment based on rules or naming patterns.
No restriction

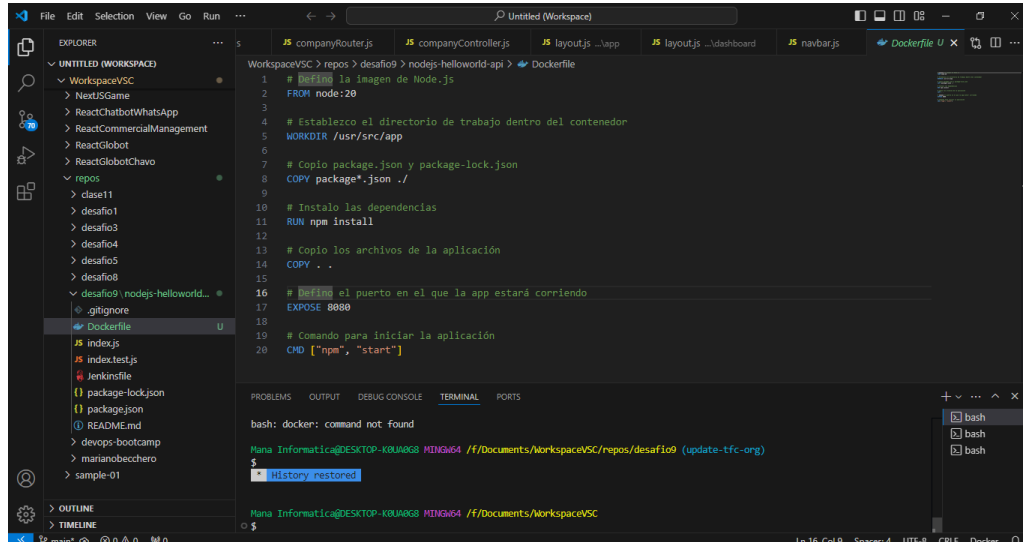
Environment secrets
Secrets are encrypted environment variables. They are accessible only by GitHub Actions in the context of this environment by using the [secret context](#).

[Add environment secret](#)

Name	Last updated
DOCKERHUBTOKEN	now
DOCKERHUBUSERNAME	now

Environment variables

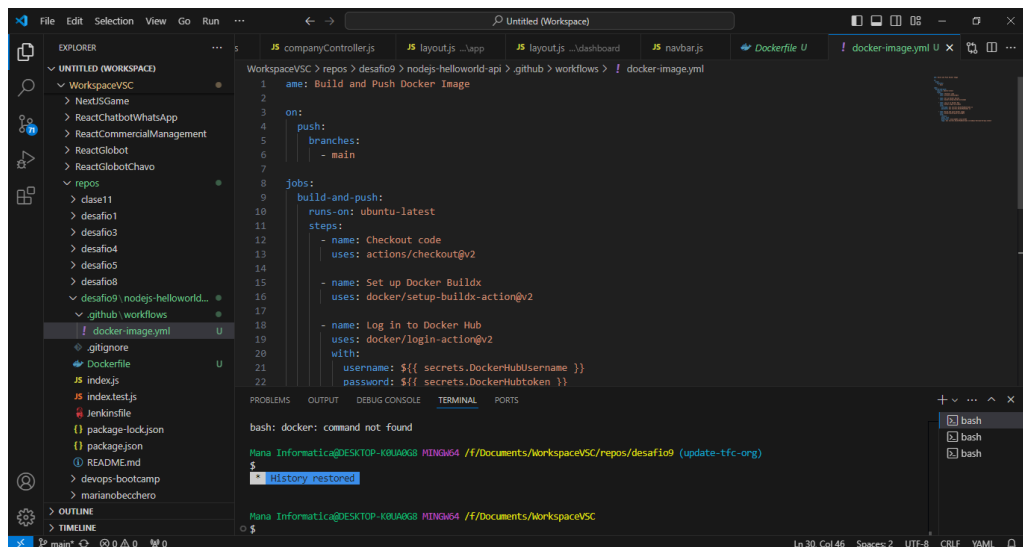
5- Añado los archivos necesarios a mi proyecto



The screenshot shows the VS Code interface with the Explorer on the left showing the project structure. The main editor displays a new file named 'Dockerfile' with the following content:

```
1 # Definir la imagen de Node.js
2 FROM node:20
3
4 # Establezco el directorio de trabajo dentro del contenedor
5 WORKDIR /usr/src/app
6
7 # Copio package.json y package-lock.json
8 COPY package*.json ./
9
10 # Instalo las dependencias
11 RUN npm install
12
13 # Copio los archivos de la aplicación
14 COPY . .
15
16 # Defino el puerto en el que la app estará corriendo
17 EXPOSE 8080
18
19 # Comando para iniciar la aplicación
20 CMD ["npm", "start"]
```

The terminal at the bottom shows the command 'docker: command not found' and the user's shell prompt.

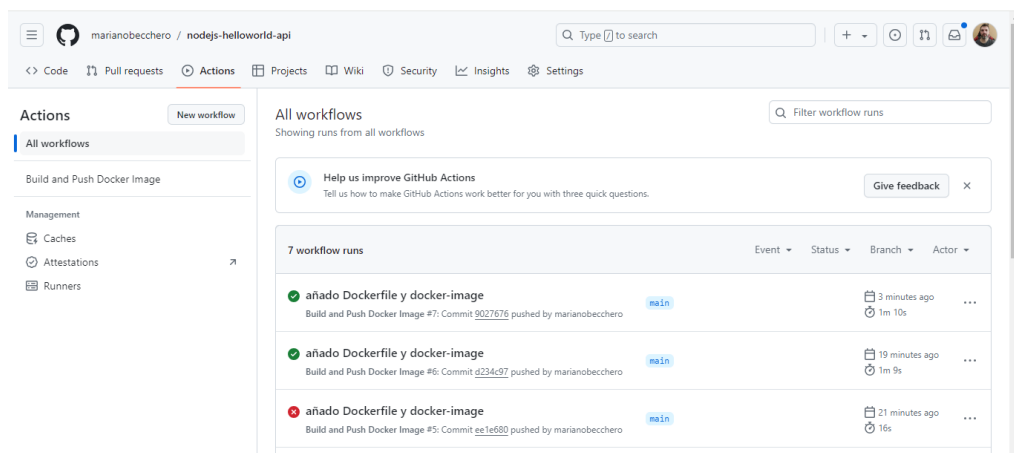


The screenshot shows the VS Code interface with the Explorer on the left showing the project structure. The main editor displays a new file named 'docker-image.yml' with the following content:

```
1 name: Build and Push Docker Image
2
3 on:
4   push:
5     branches:
6       - main
7
8 jobs:
9   build-and-push:
10    runs-on: ubuntu-latest
11    steps:
12      - name: Checkout code
13        uses: actions/checkout@v2
14
15      - name: Set up Docker Buildx
16        uses: docker/setup-buildx-action@v2
17
18      - name: Log in to Docker Hub
19        uses: docker/login-action@v2
20        with:
21          username: ${{ secrets.DockerHubUsername }}
22          password: ${{ secrets.DockerHubToken }}
```

The terminal at the bottom shows the command 'docker: command not found' and the user's shell prompt.

6- Pusheo los cambios y se ejecuta el action



dockerhub

Explore

Repositories

Organizations

Search Docker Hub

ctrl+k

M

marianobecchero / Repositories / desafio9 / General

Using 1 of 1 private repositories.

General

Tags

Builds

Collaborators

Webhooks

Settings

marianobecchero/desafio9

Updated 2 minutes ago

Desafio 9

This repository does not have a category

Docker commands

To push a new tag to this repository:

docker push marianobecchero/desafio9:tagname

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	—	3 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds >](#)