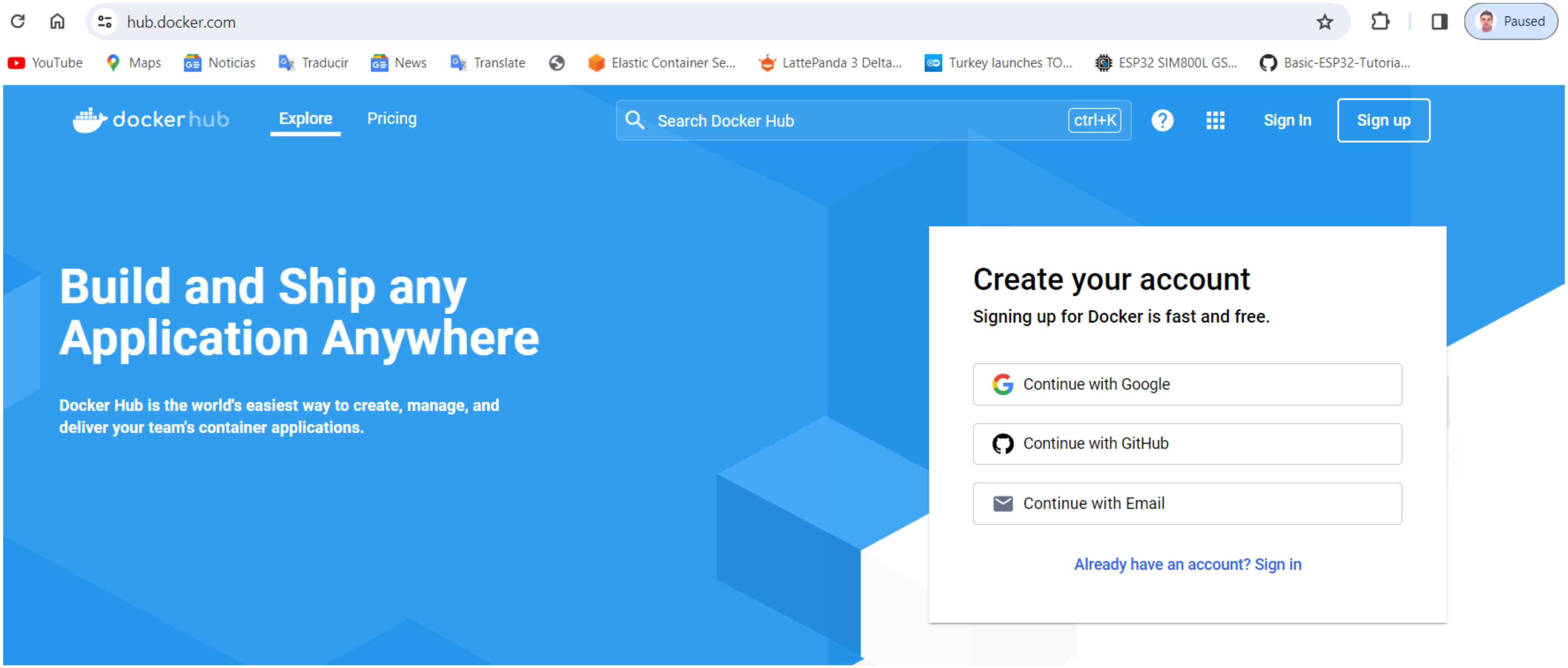
GitHub Actions: how to create .NET 8 Web API Docker image and Upload it to Docker Hub

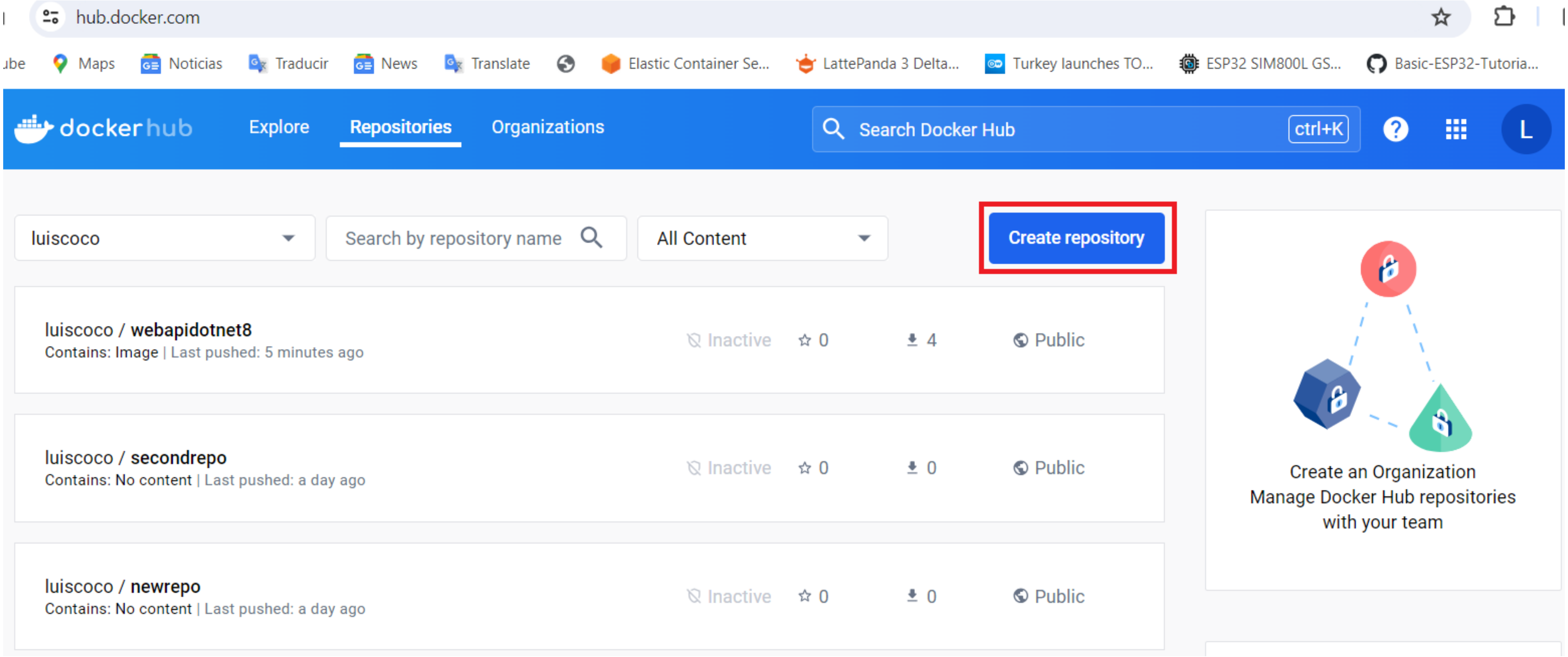
1. Create a new Docker Hub repo for storing your .NET Web API Docker image

Create a new account in Docker Hub or if you already have an account Sign in

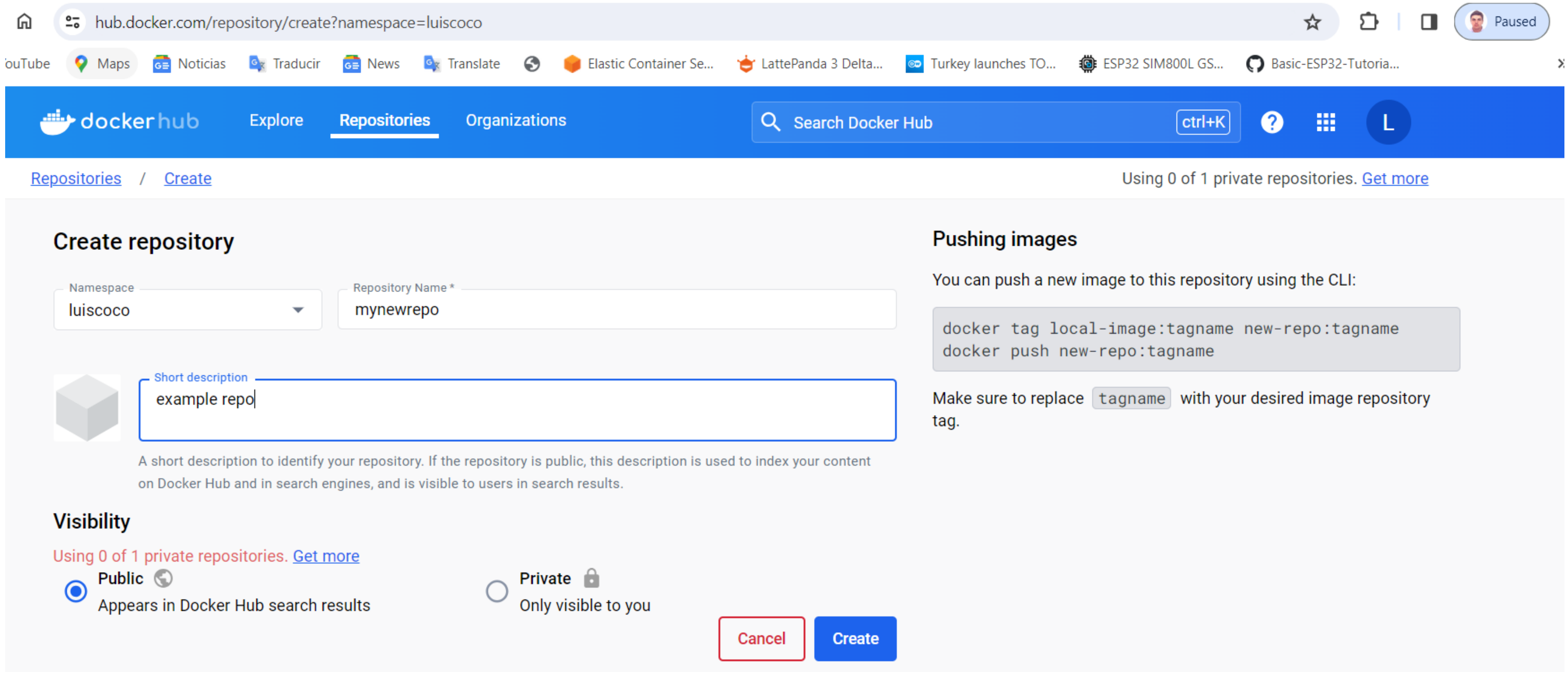
https://hub.docker.com/



Create a new repository in Docker Hub for storing your Docker image



luiscoco / **webapidotnet8** Contains: Image | Last pushed: 5 minutes ago	Inactive	☆ 0	4	Public
luiscoco / **secondrepo** Contains: No content | Last pushed: a day ago	Inactive	☆ 0	0	Public
luiscoco / **newrepo** Contains: No content | Last pushed: a day ago	Inactive	☆ 0	0	Public



Repositories / Create

Using 0 of 1 private repositories. Get more

Create repository

Namespace: luiscoco

Repository Name *: mynewrepo

Short description: example repo

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

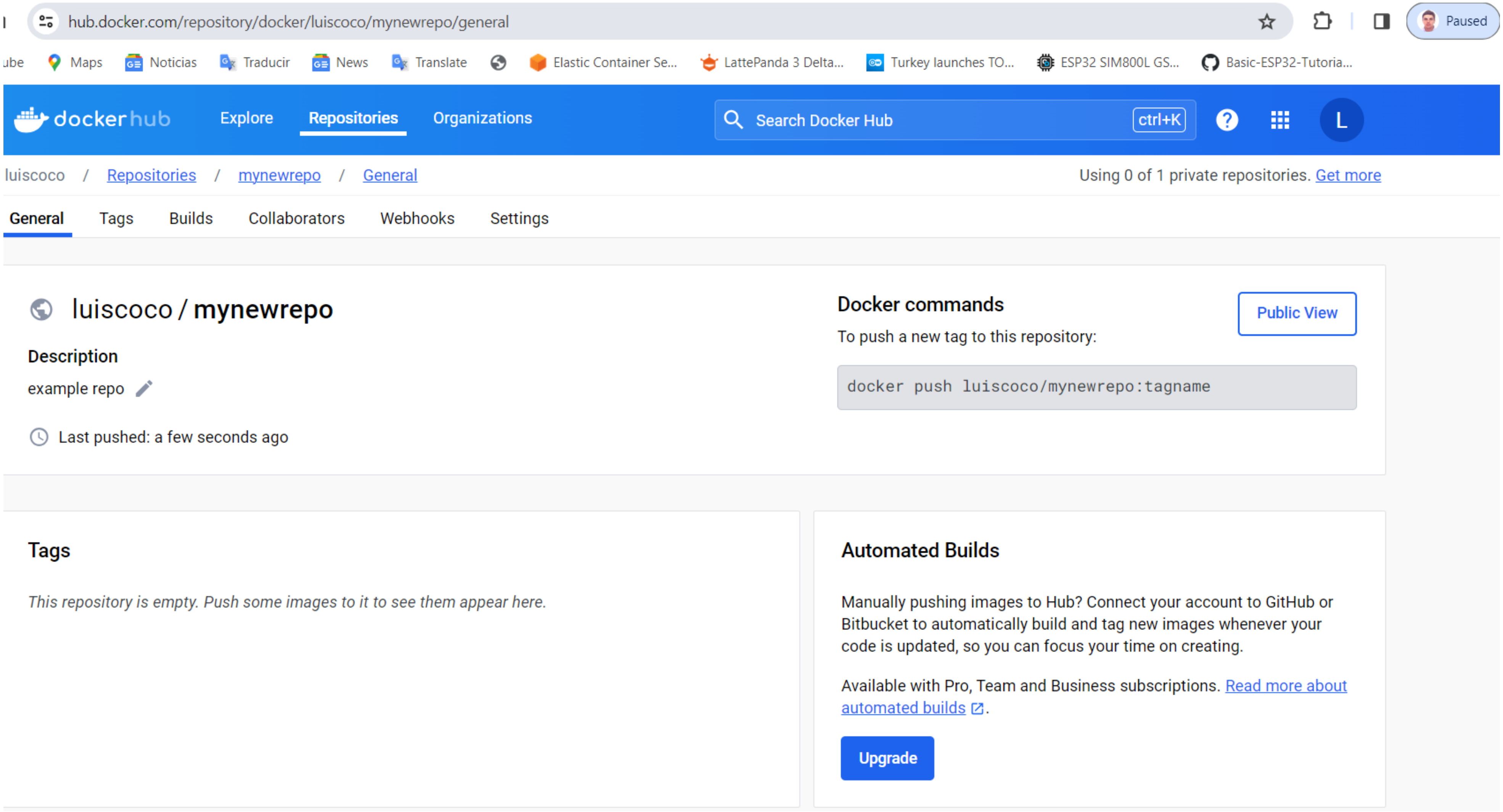
Cancel Create

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.



luiscoco / mynewrepo

Description

example repo

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository:

```
docker push luiscoco/mynewrepo:tagname
```

Tags

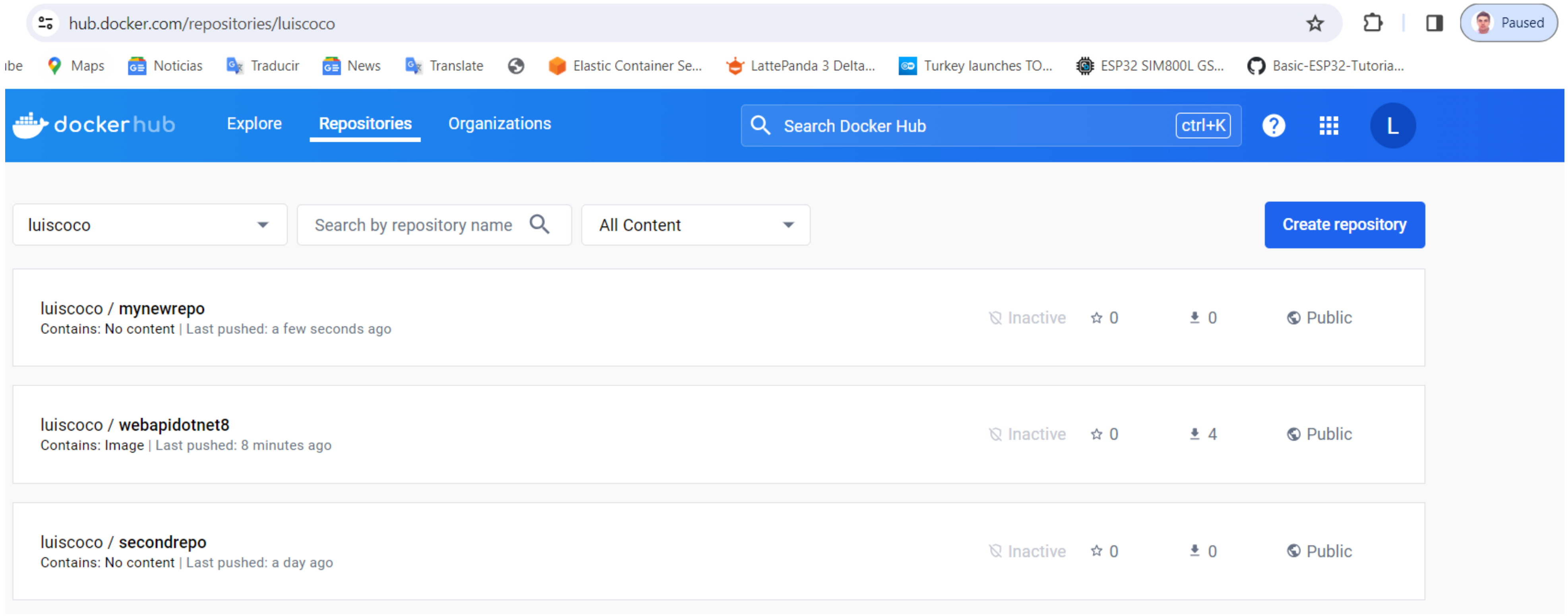
This repository is empty. Push some images to it to see them appear here.

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.

See your new Docker Hub repo in the list



luiscoco / **mynewrepo**
Contains: No content | Last pushed: a few seconds ago
Inactive ☆ 0 ⬇ 0 Public

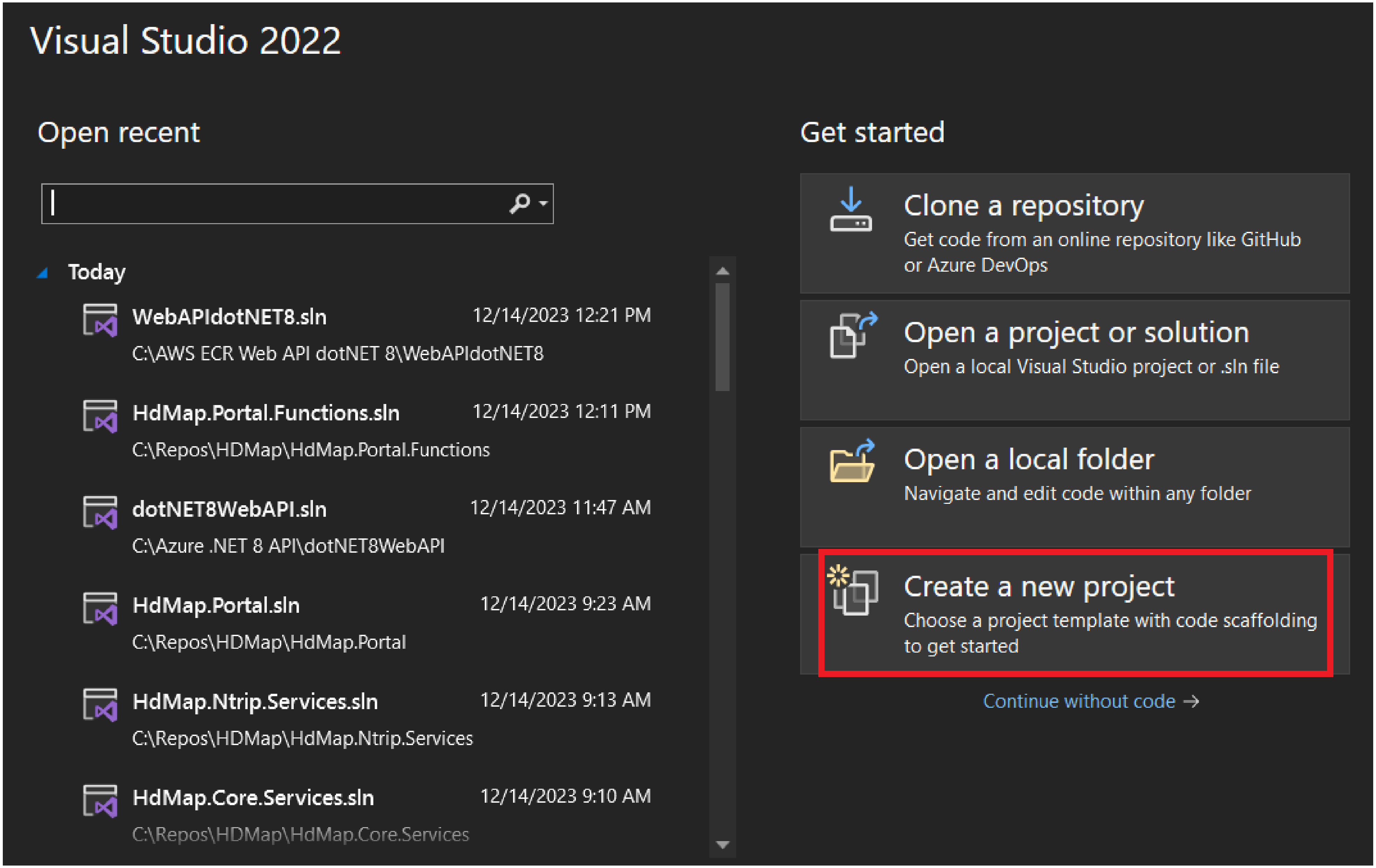
luiscoco / **webapidotnet8**
Contains: Image | Last pushed: 8 minutes ago
Inactive ☆ 0 ⬇ 4 Public

luiscoco / **secondrepo**
Contains: No content | Last pushed: a day ago
Inactive ☆ 0 ⬇ 0 Public

2. Create a new .NET 8 Web API with Visual Studio 2022 Community Edition

We run Visual Studio 2022

We press the option "Create a new project"



Visual Studio 2022

Open recent

Today

WebAPIdotNET8.sln C:\AWS ECR Web API dotNET 8\WebAPIdotNET8	12/14/2023 12:21 PM
HdMap.Portal.Functions.sln C:\Repos\HDMap\HdMap.Portal.Functions	12/14/2023 12:11 PM
dotNET8WebAPI.sln C:\Azure .NET 8 API\dotNET8WebAPI	12/14/2023 11:47 AM
HdMap.Portal.sln C:\Repos\HDMap\HdMap.Portal	12/14/2023 9:23 AM
HdMap.Ntrip.Services.sln C:\Repos\HDMap\HdMap.Ntrip.Services	12/14/2023 9:13 AM
HdMap.Core.Services.sln C:\Repos\HDMap\HdMap.Core.Services	12/14/2023 9:10 AM

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

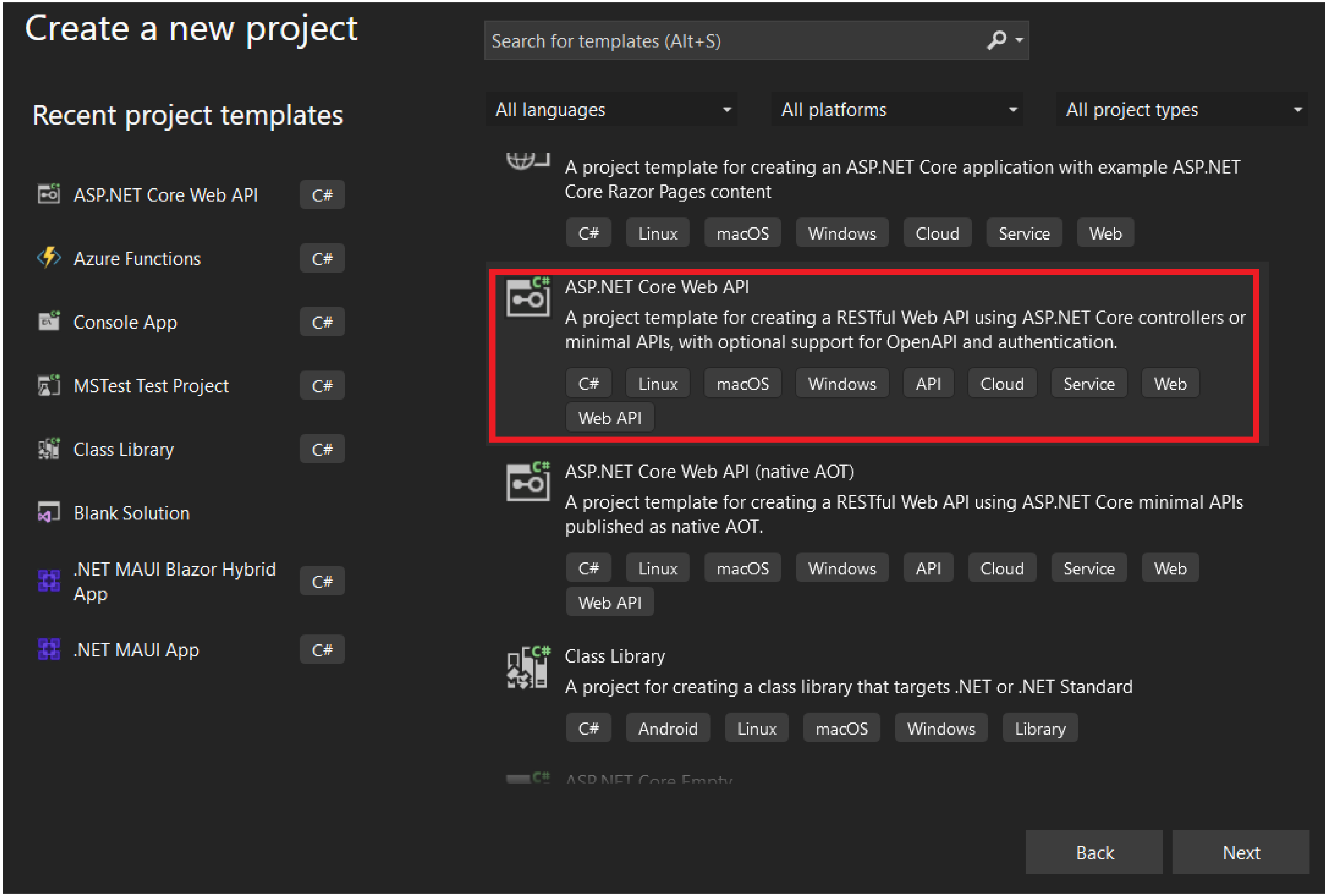
Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

We selec the Web API project template



Create a new project

Search for templates (Alt+S)

All languages | All platforms | All project types

Recent project templates

- ASP.NET Core Web API C#
- Azure Functions C#
- Console App C#
- MSTest Test Project C#
- Class Library C#
- Blank Solution
- .NET MAUI Blazor Hybrid App C#
- .NET MAUI App C#

A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content

C# Linux macOS Windows Cloud Service Web

ASP.NET Core Web API

A project template for creating a RESTful Web API using ASP.NET Core controllers or minimal APIs, with optional support for OpenAPI and authentication.

C# Linux macOS Windows API Cloud Service Web Web API

ASP.NET Core Web API (native AOT)

A project template for creating a RESTful Web API using ASP.NET Core minimal APIs published as native AOT.

C# Linux macOS Windows API Cloud Service Web Web API

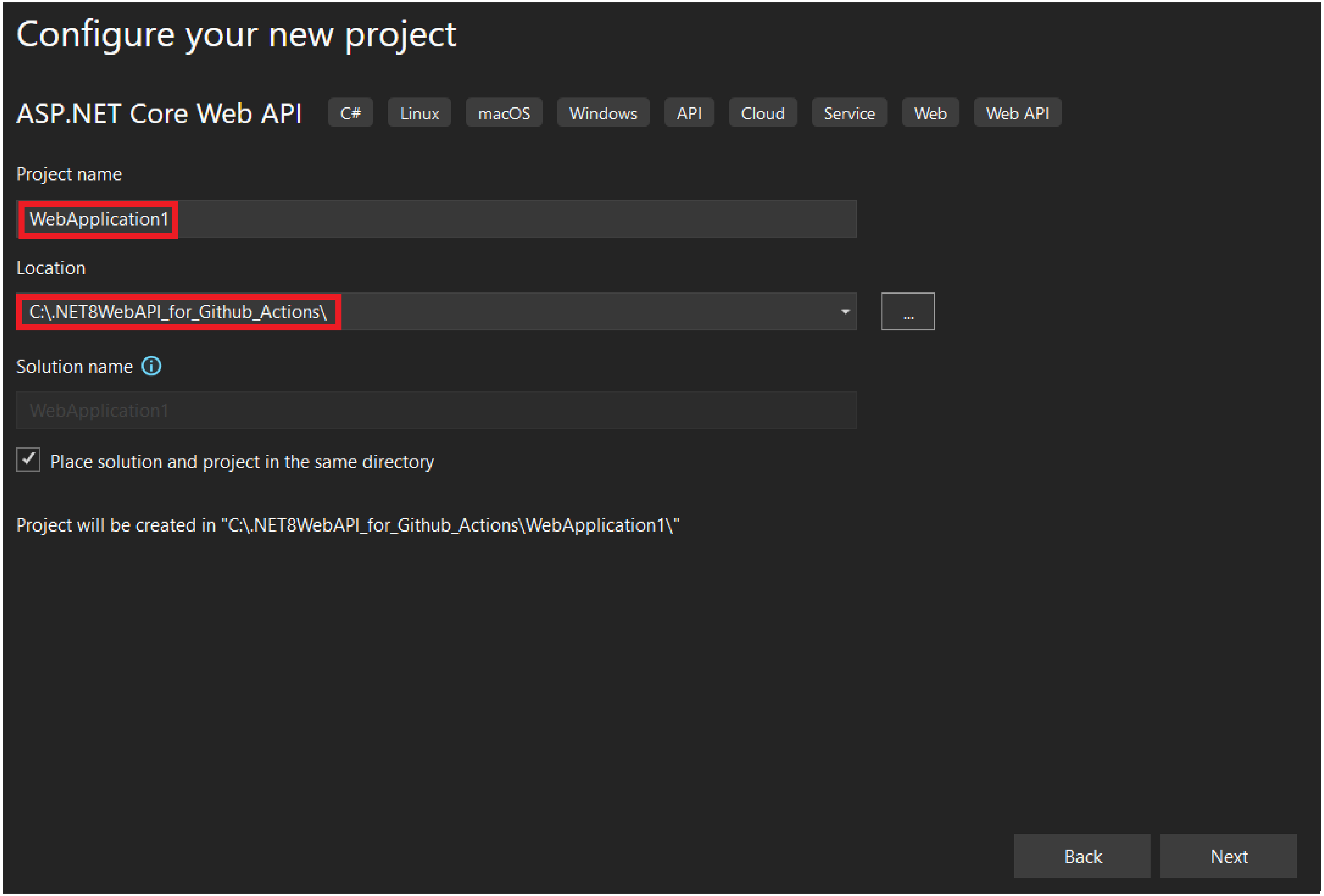
Class Library

A project for creating a class library that targets .NET or .NET Standard

C# Android Linux macOS Windows Library

ASP.NET Core Empty

We set the solution name and location



Configure your new project

ASP.NET Core Web API

Project name

WebApplication1

Location

C:\.NET8WebAPI_for_Github_Actions\

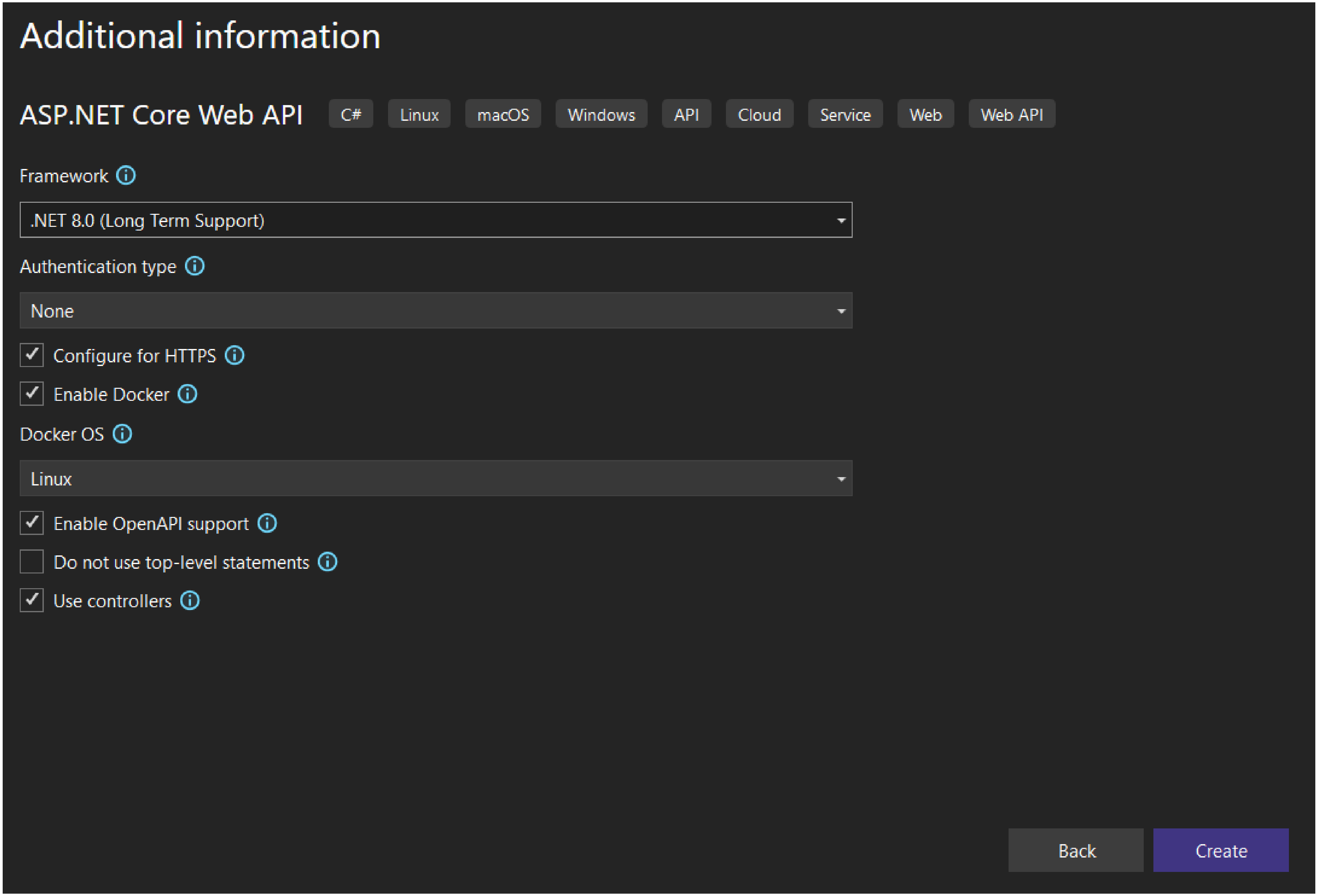
Solution name

WebApplication1

☑ Place solution and project in the same directory

Project will be created in "C:\.NET8WebAPI_for_Github_Actions\WebApplication1\"

We select the new solution main features: .NET 8 framework, **Docker enable (for creating automatically the Dockerfile)**, etc



Additional information

ASP.NET Core Web API

Framework

.NET 8.0 (Long Term Support)

Authentication type

None

☑ Configure for HTTPS

☑ Enable Docker

Docker OS

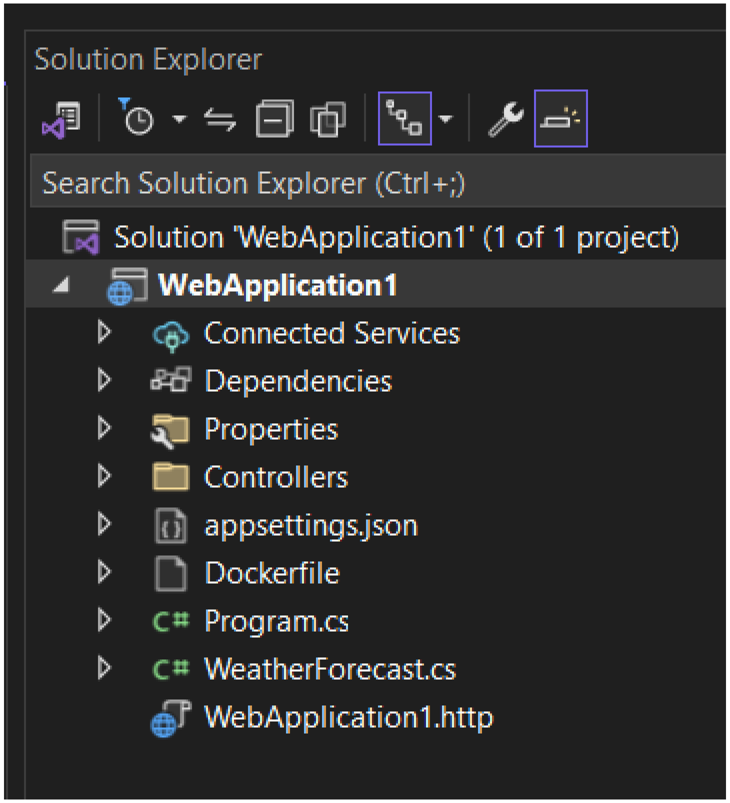
Linux

☑ Enable OpenAPI support

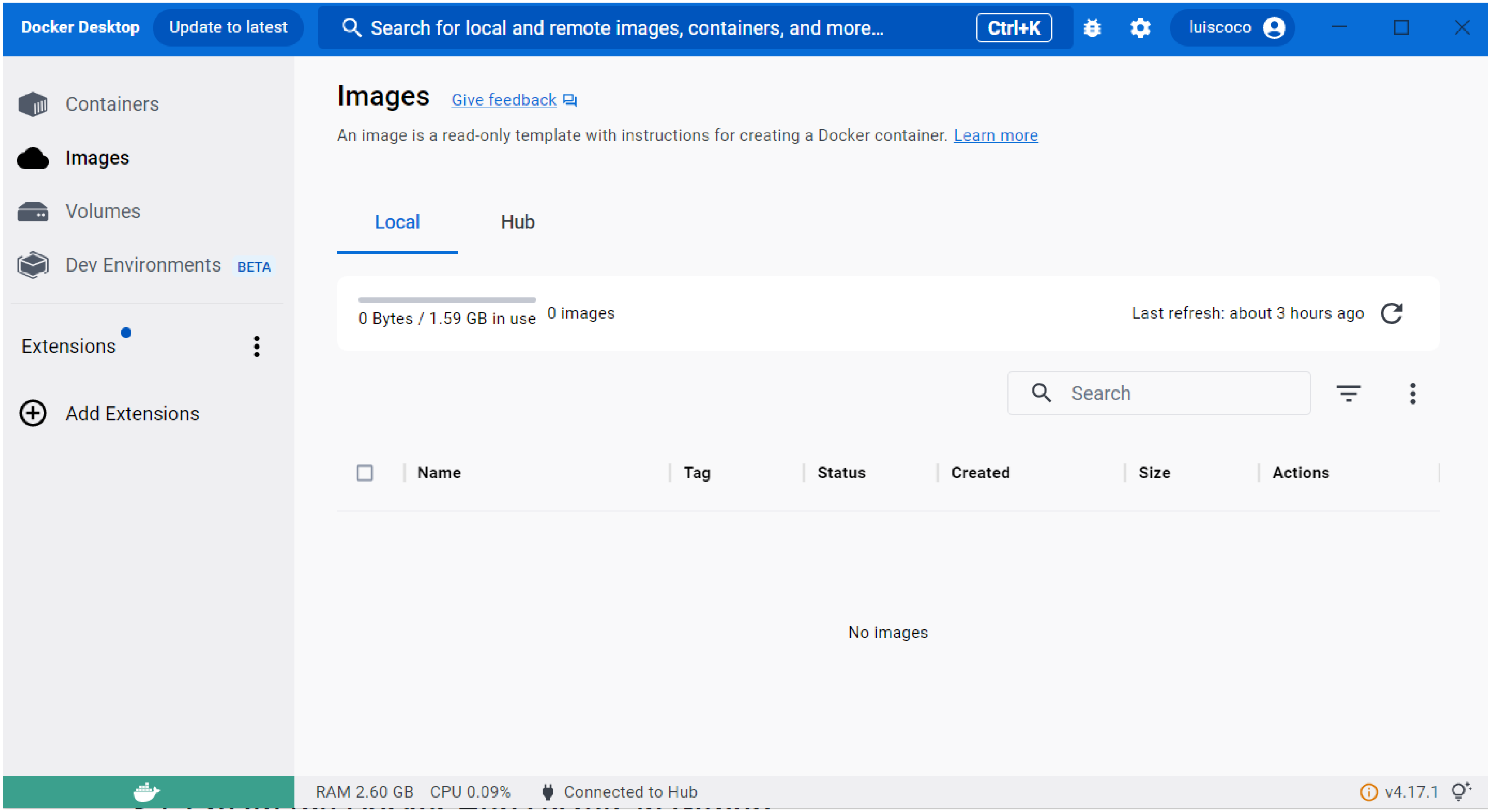
☐ Do not use top-level statements

☑ Use controllers

This is the Solution folders structure:



Now if We didn't start yet the Docker Desktop a warning message will appear requesting us to run it.



An image is a read-only template with instructions for creating a Docker container. Learn more

0 Bytes / 1.59 GB in use 0 images

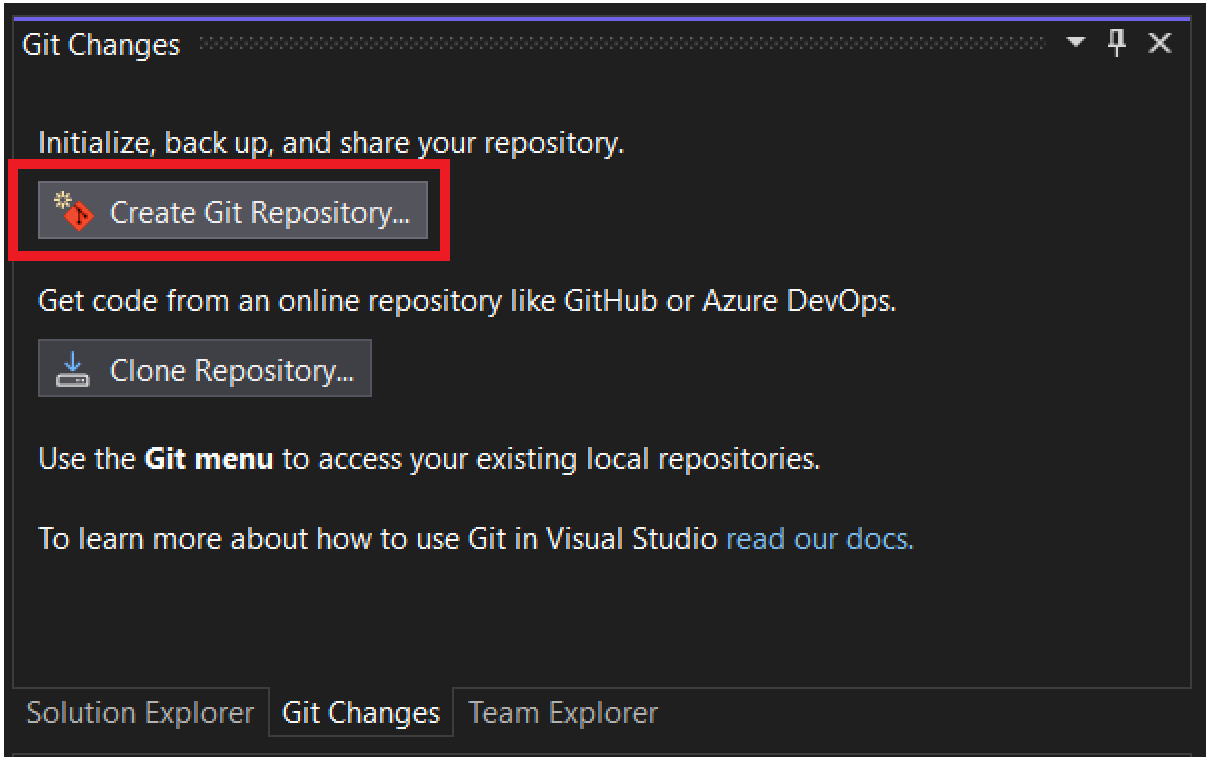
Last refresh: about 3 hours ago

No images

RAM 2.60 GB CPU 0.09% Connected to Hub v4.17.1

3. Create a Github repository in Visual Studio 2022 Community Edition

We select the "Git Changes" tab and press the "Create Git Repository..." button



Git Changes

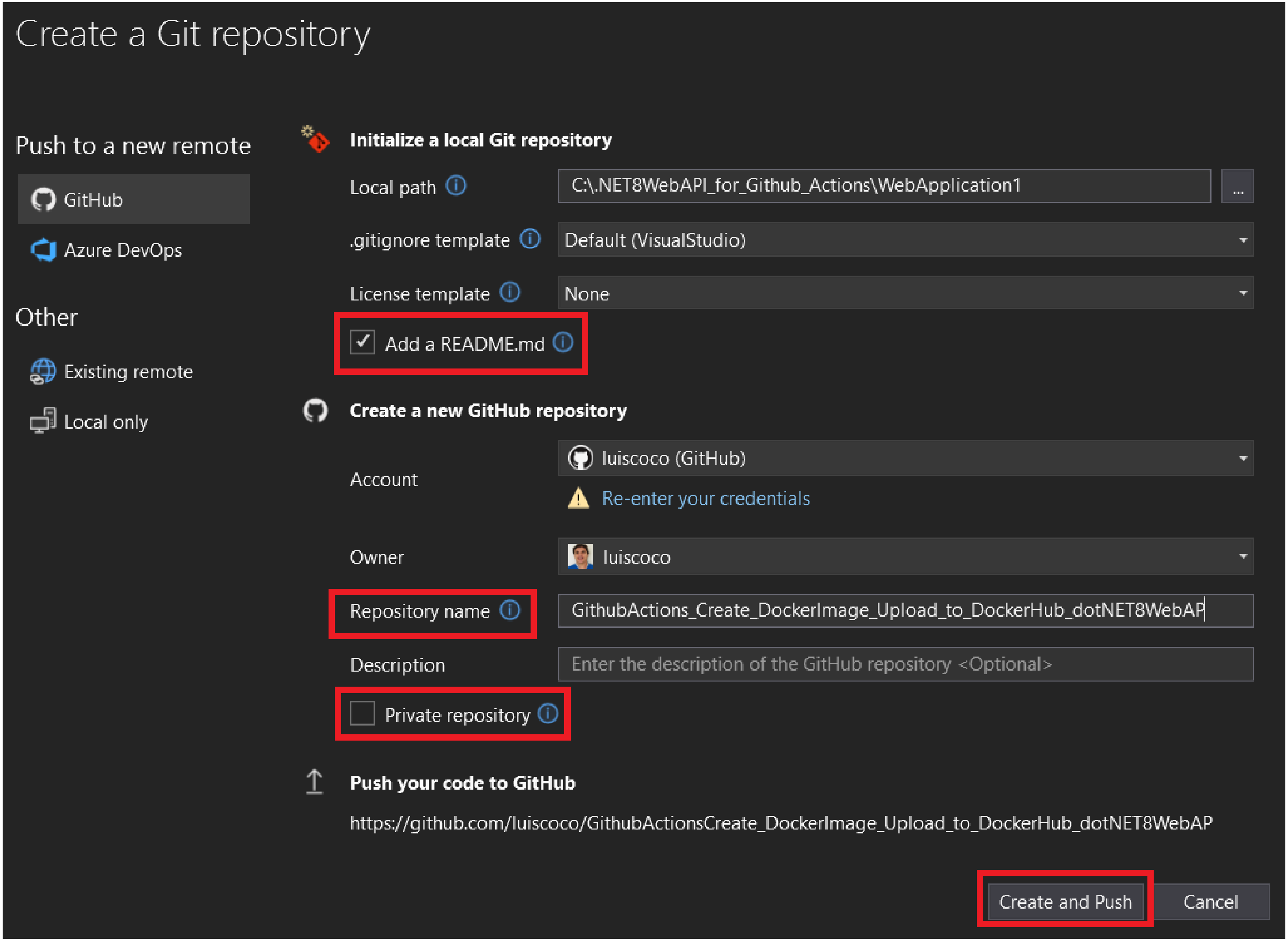
Initialize, back up, and share your repository.

Get code from an online repository like GitHub or Azure DevOps.

Use the **Git menu** to access your existing local repositories.

To learn more about how to use Git in Visual Studio read our docs.

We input the repository name



Local path: C:\.NET8WebAPI_for_Github_Actions\WebApplication1

.gitignore template: Default (VisualStudio)

License template: None

☑ Add a README.md

Account: luiscoco (GitHub)

Owner: luiscoco

Repository name: GithubActions_Create_DockerImage_Upload_to_DockerHub_dotNET8WebAP

Description: Enter the description of the GitHub repository <Optional>

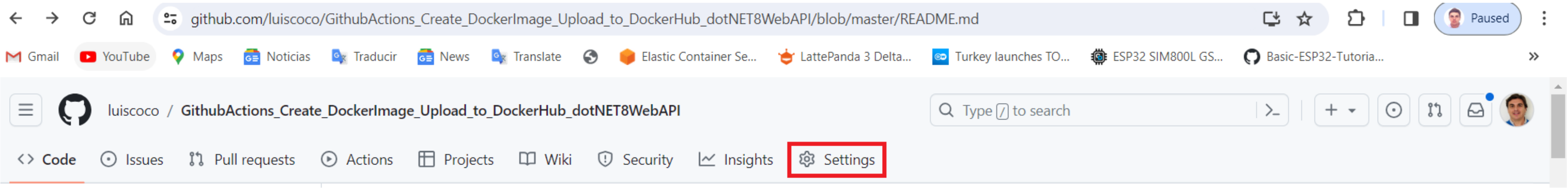
☐ Private repository

https://github.com/luiscoco/GithubActionsCreate_DockerImage_Upload_to_DockerHub_dotNET8WebAP

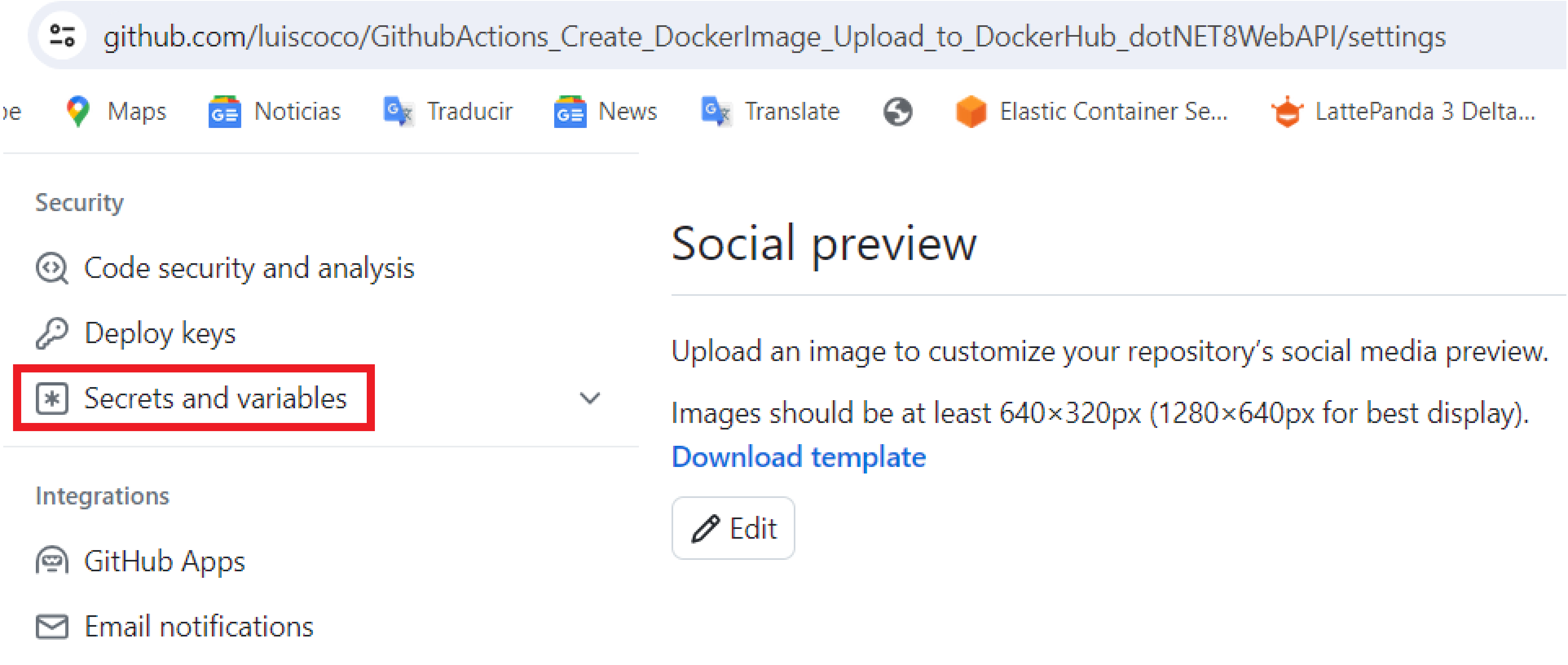
4. Create the Github Action Workflow for creating the Docker image and Upload it to Docker Hub

4.1. Create the Docker Hub secrets in Github

We navigate to the "**settings**" option inside our Github repository

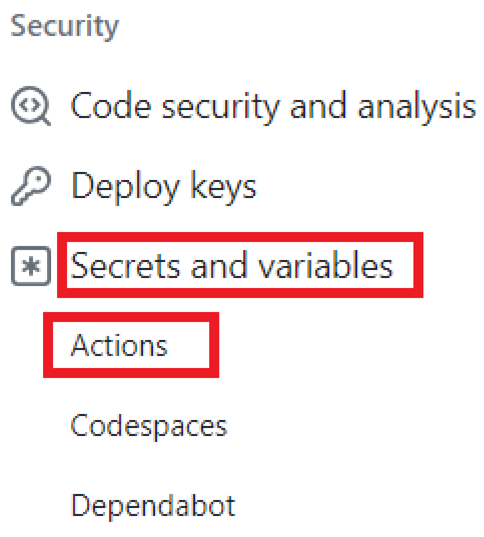


We now select the "**Secretas and variables**" menu option

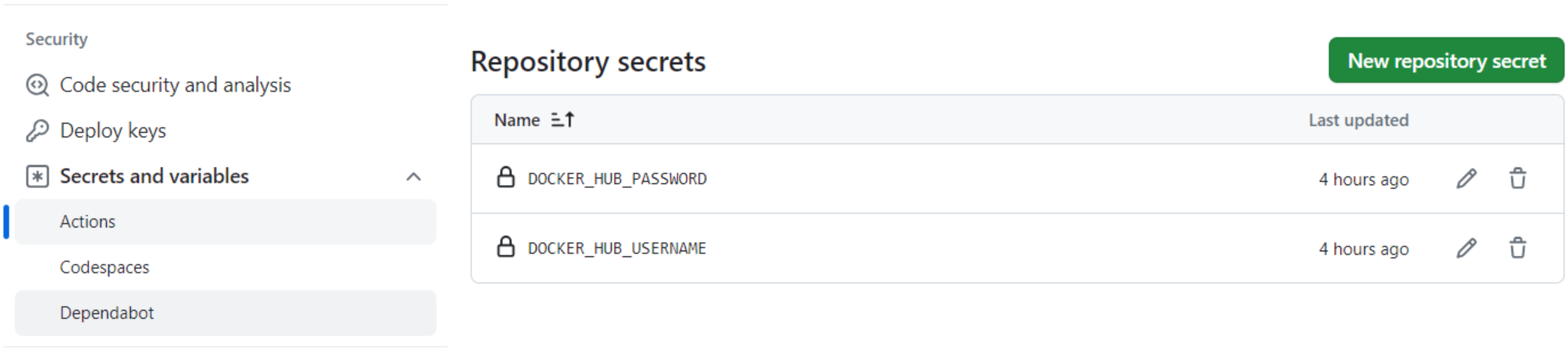


github.com/luiscoco/GithubActions_Create_DockerImage_Upload_to_DockerHub_dotNET8WebAPI/settings
Maps
Noticias
Traducir
News
Translate
Elastic Container Se...
LattePanda 3 Delta...
Security
Code security and analysis
Deploy keys
Secrets and variables
Integrations
GitHub Apps
Email notifications
Social preview
Upload an image to customize your repository's social media preview.
Images should be at least 640×320px (1280×640px for best display).
Download template
Edit

Select the "**Actions**" option for creating a new repository secret



Create two secrets, one of them for storing the Docker Hub user name, and the other one for storing the Docker Hub password.

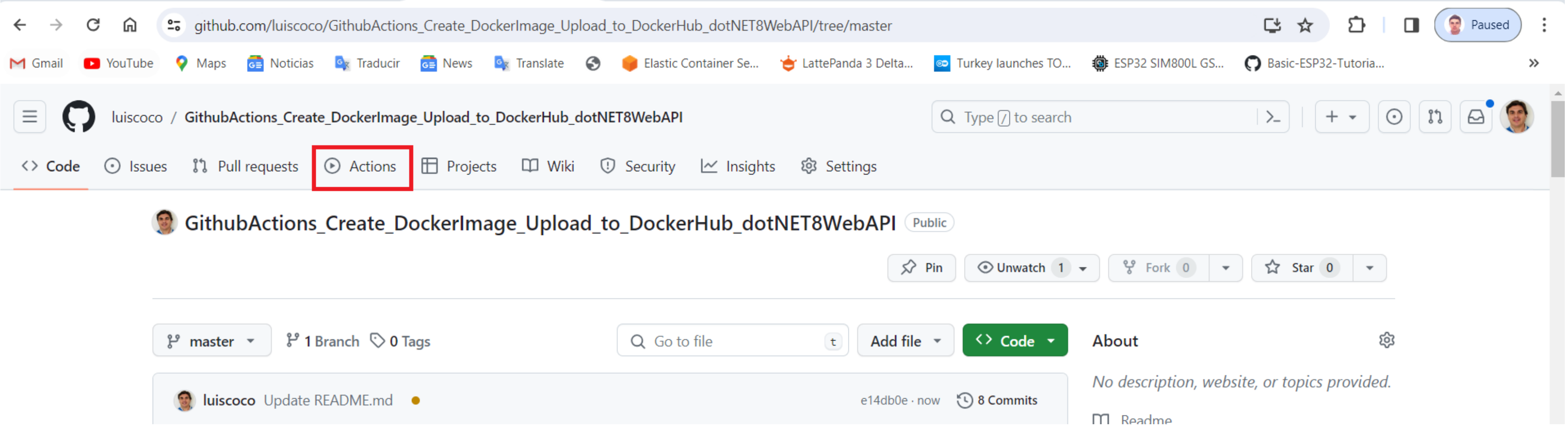


Repository secrets

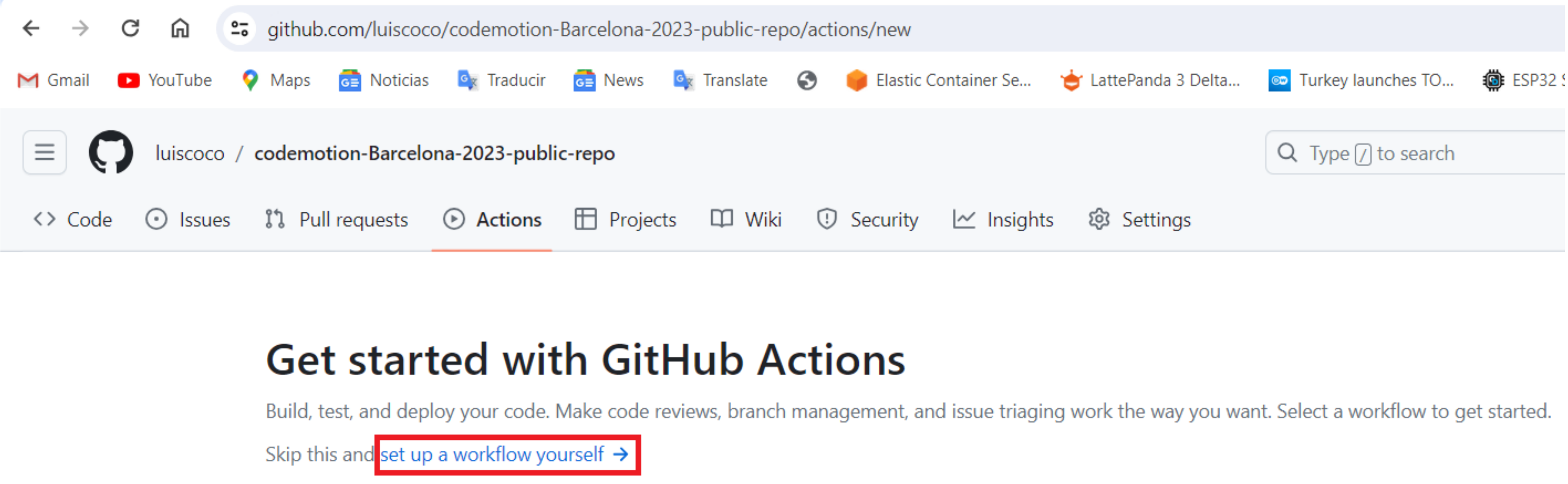
Name	Last updated
DOCKER_HUB_PASSWORD	4 hours ago
DOCKER_HUB_USERNAME	4 hours ago

4.2. Create the Github Action Workflow

In our new Github repo we press in the "Actions" button for creating a new Github Action Workflow



The we selec the option "set up a workflow yourself"



Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and set up a workflow yourself →

We input the workflow yml file source code

```yaml
name: Docker Image CI

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3

    - name: Log in to Docker Hub
      uses: docker/login-action@v1
      with:
        registry: docker.io
        username: ${{ secrets.DOCKER_HUB_USERNAME }}
        password: ${{ secrets.DOCKER_HUB_PASSWORD }}

    - name: Build and push Docker image
      run: |
        IMAGE_ID=docker.io/luiscoco/webapidotnet8
        # Build the Docker image
        docker build . --file Dockerfile --tag $IMAGE_ID:latest
        # Push the image to Docker Hub
        docker push $IMAGE_ID:latest
```

This code represents a **GitHub Actions workflow** defined in a **YAML** format.

It is used to automate the build and deployment of a **Docker image** to **Docker Hub** when changes are pushed to the "**master**" branch or when a pull request is opened against the "master" branch in a GitHub repository.

Workflow Name and Trigger Events:

The **workflow** is named "**Docker Image CI**"

It is **triggered** by two **GitHub events**:

push event on the "**master**" branch: This means the workflow will run whenever changes are pushed to the "master" branch.

pull_request event on the "**master**" branch: This means the workflow will also run when a pull request is opened against the "master" branch.

```
name: Docker Image CI
on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]
```

Jobs:

The workflow defines a single job named "build" that will run on an "**ubuntu-latest**" runner (a **virtual machine** provided by GitHub Actions).

```
jobs:
  build:
    runs-on: ubuntu-latest
```

Job Steps:

The job consists of several steps that will be **executed sequentially**.

The first step is to checkout the repository's source code using the **actions/checkout@v3** action.

```
  steps:
  - uses: actions/checkout@v3
```

Logging in to Docker Hub:

The second step is named "Log in to Docker Hub," and it uses the docker/login-action@v1 action.

It logs in to Docker Hub using the provided Docker Hub username and password stored in **GitHub secrets**: DOCKER_HUB_USERNAME and DOCKER_HUB_PASSWORD

Secrets are a way to securely store sensitive information.

```
  - name: Log in to Docker Hub
    uses: docker/login-action@v1
    with:
      registry: docker.io
      username: ${{ secrets.DOCKER_HUB_USERNAME }}
      password: ${{ secrets.DOCKER_HUB_PASSWORD }}
```

Building and Pushing Docker Image:

The third step is named "Build and push Docker image."

It runs a series of Docker commands in a shell script:

VERY IMPORTANT! It sets the IMAGE_ID variable with the **Docker Hub repository name (docker.io/luiscoco/webapidotnet8)**.

It builds a Docker image using a Dockerfile from the repository, tags it as "latest," and assigns it the image ID.

It then pushes the Docker image to Docker Hub using the same image ID.

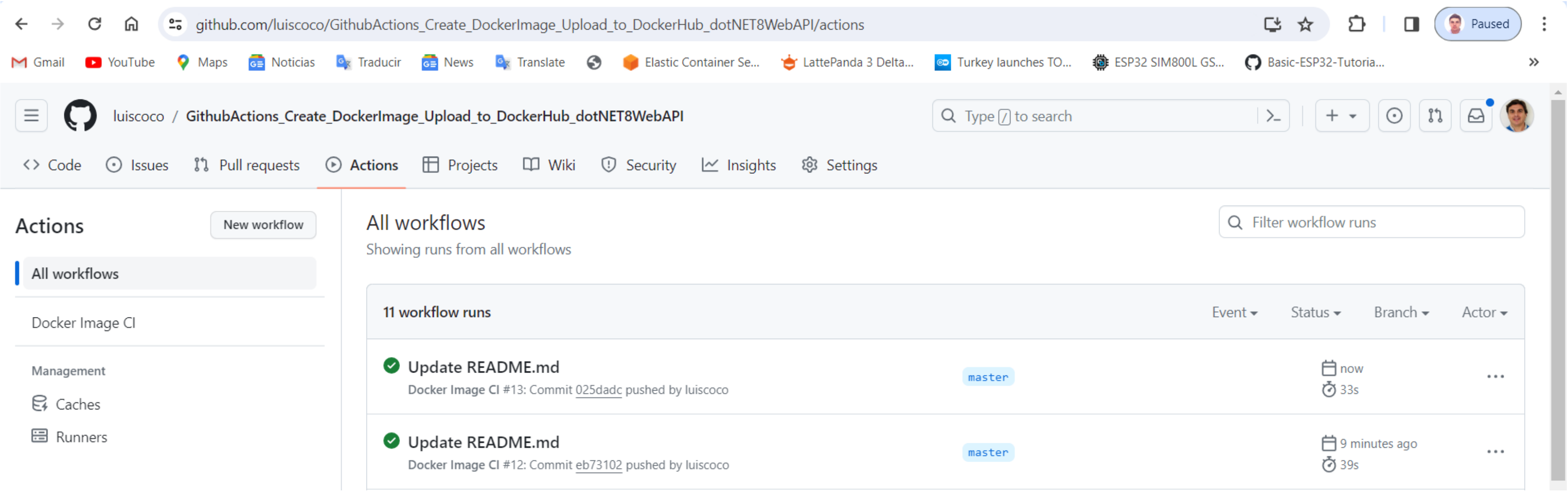
```
  - name: Build and push Docker image
    run: |
      IMAGE_ID=docker.io/luiscoco/webapidotnet8
      # Build the Docker image
      docker build . --file Dockerfile --tag $IMAGE_ID:latest
      # Push the image to Docker Hub
      docker push $IMAGE_ID:latest
```

In summary, this GitHub Actions workflow is designed to automate the building and pushing of a Docker image to Docker Hub when changes are made to the "master" branch of the associated GitHub repository or when pull requests are opened against the "master" branch.

It ensures that the Docker image is kept up to date and is available for deployment.

4.3. Verify the docker image was uploaded to Docker Hub

If we press on the Actions button we can verify the workflow builds were successfully

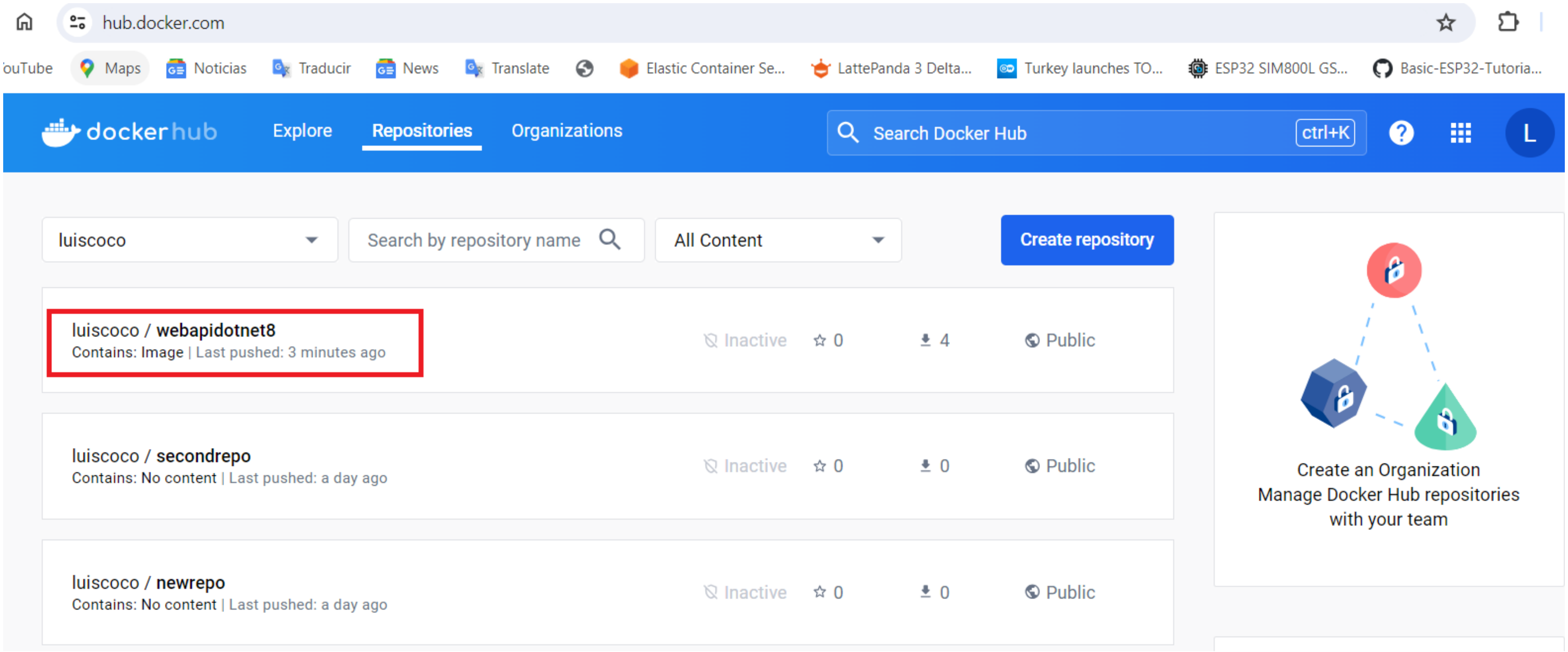


11 workflow runs

✓ Update README.md
Docker Image CI #13: Commit 025dadc pushed by luiscoco
master
now
33s

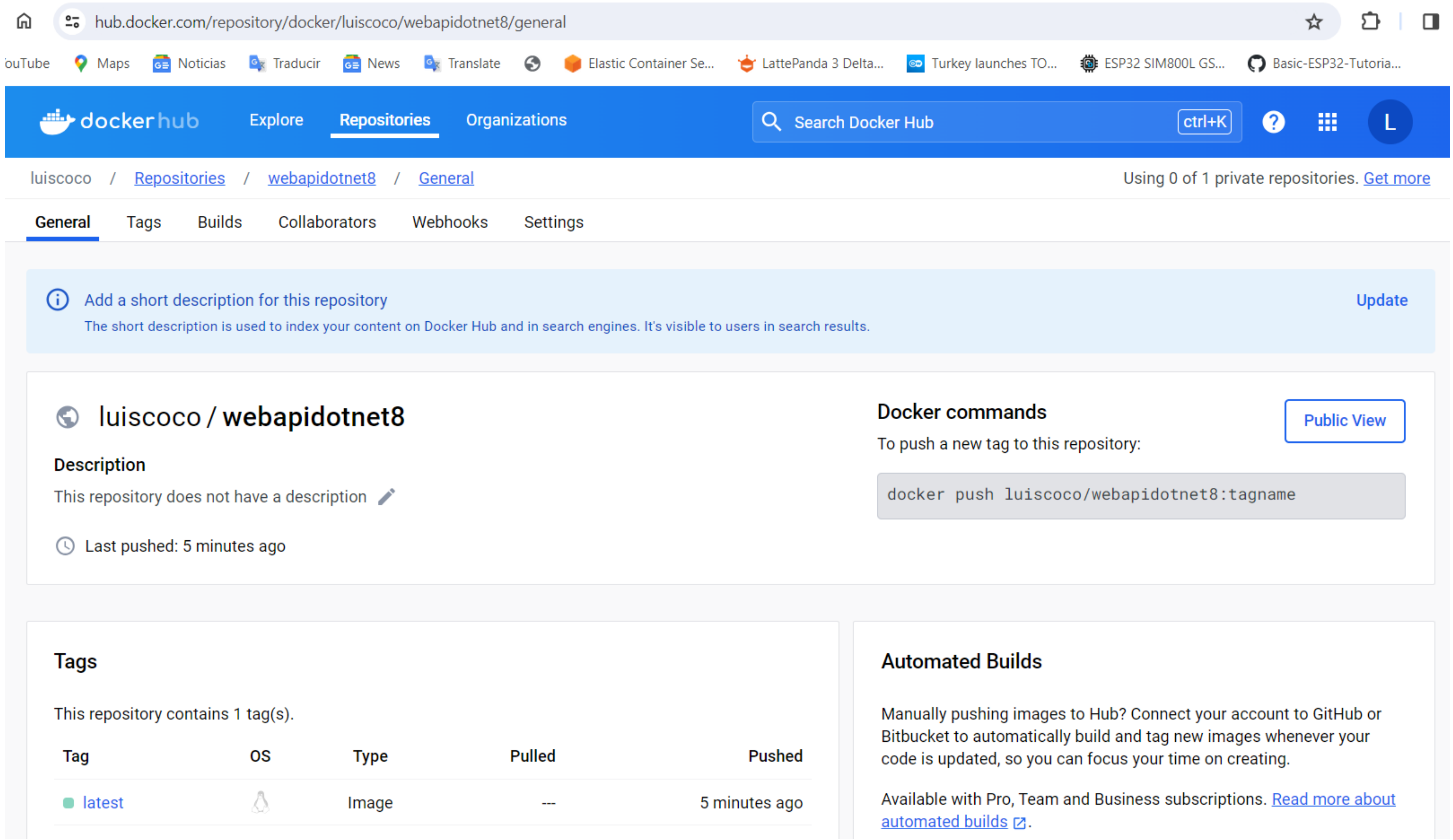
✓ Update README.md
Docker Image CI #12: Commit eb73102 pushed by luiscoco
master
9 minutes ago
39s

Finally, we can log in to Docker Hub and navigate to our repository and check the image was uploaded a few minutes ago



Repository	Status	Stars	Downloads	Visibility
luiscoco / webapidotnet8 Contains: Image | Last pushed: 3 minutes ago	Inactive	0	4	Public
luiscoco / secondrepo Contains: No content | Last pushed: a day ago	Inactive	0	0	Public
luiscoco / newrepo Contains: No content | Last pushed: a day ago	Inactive	0	0	Public

We also can enter in the repo to see the docke image details



Using 0 of 1 private repositories. Get more

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

luiscoco / webapidotnet8

Description

This repository does not have a description

Last pushed: 5 minutes ago

Docker commands

To push a new tag to this repository:

```
docker push luiscoco/webapidotnet8:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	5 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.