



---

# [FUNDAMENTALS] CLOUD & DEVOPS ASSESSMENT MEXICO, ARGENTINA

---

Azure Practical Task



OCTOBER 1, 2024  
[LEARN.EPAM.COM](https://learn.epam.com)

## Contents

1. Create Azure Web App with Azure App Services. ....	1
2. Create GitHub repo with simple personal website.....	2
Create GitHub repo .....	2
Commit your first change .....	2
Next steps .....	4
3. Deploy an application with GitHub Actions and Azure Static Web Apps. ....	5
Important .....	10
References .....	11

# Azure Practical Task

## 1. Create Azure Web App with Azure App Services.

Go to [Azure portal](https://portal.azure.com). Azure Static Web Apps is very simple and easy to work with. There are 3 ways to create the Azure Static Web Apps, i.e., VS Code, Azure portal, and Azure CLI.

Select an option from the 3 given, in this example was selected **Access student benefits**.

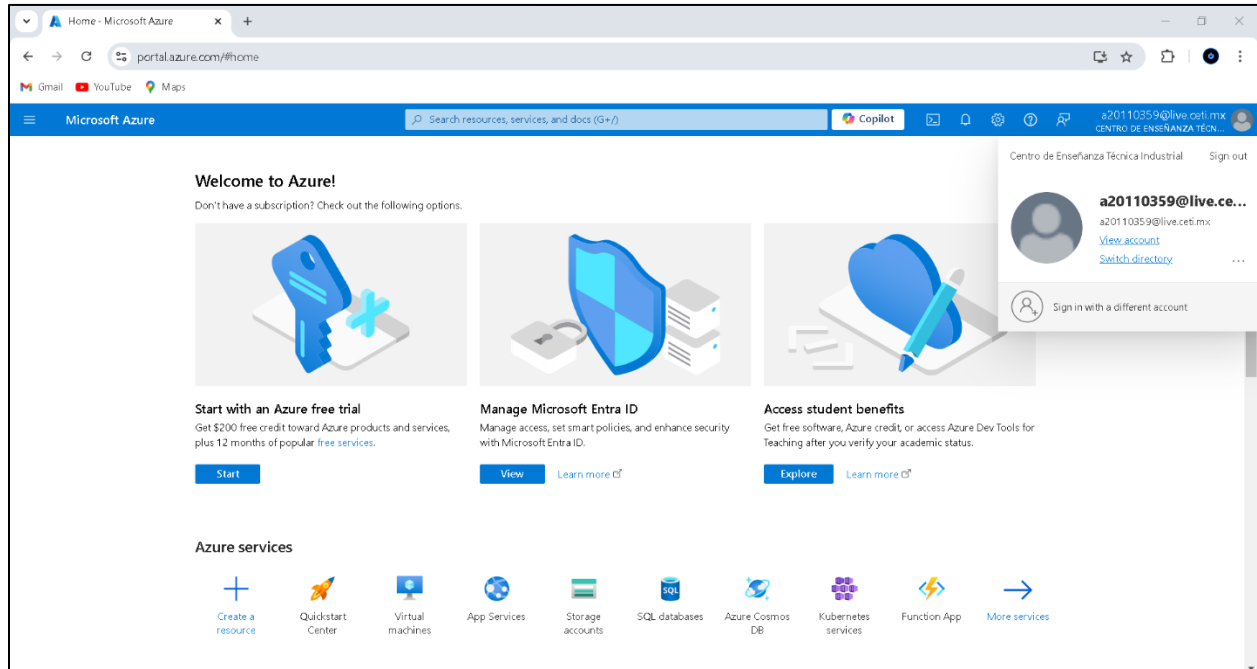


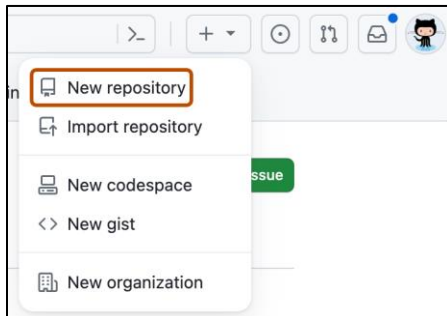
Figure 1 Create Azure Portal Account

## 2. Create [GitHub](#) repo with simple personal website.

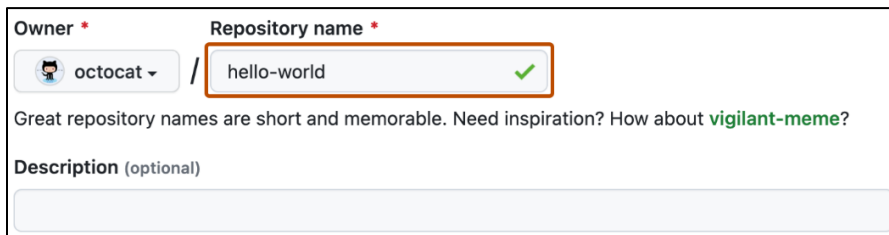
### Create [GitHub](#) repo

GitHub repositories store a variety of projects. In this guide, you'll create a repository and commit your first change.

1. In the upper-right corner of any page, select, then click **New repository**.



2. Type a short, memorable name for your repository. For example, "hello-world".

A screenshot of the 'Create new repository' form on GitHub. The 'Owner' field shows 'octocat' with a dropdown arrow. The 'Repository name' field contains 'hello-world' and is highlighted with a red box, with a green checkmark to its right. Below the name field, there is a suggestion: 'Great repository names are short and memorable. Need inspiration? How about vigilant-meme?'. The 'Description (optional)' field is empty and has a light blue border.

3. Optionally, add a description of your repository. For example, "My first repository on GitHub."
4. Choose a repository visibility.
5. Select **Initialize this repository with a README**.
6. Click **Create repository**.

Congratulations! You've successfully created your first repository and initialized it with a *README* file.

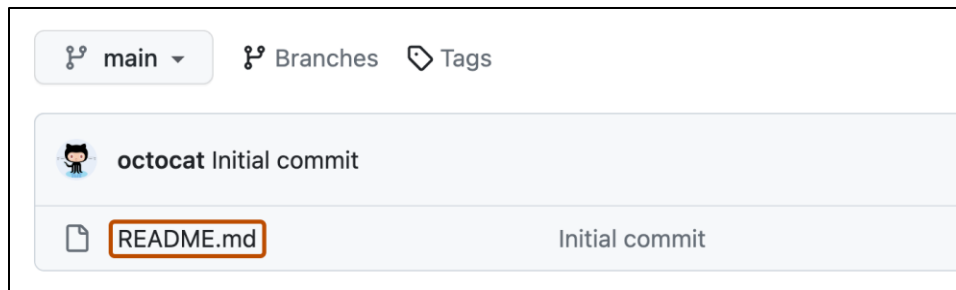
### Commit your first change

A [commit](#) is like a snapshot of all the files in your project at a particular point in time.

When you created your new repository, you initialized it with a *README* file. *README* files are a great place to describe your project in more detail or add some documentation such as how to install or use your project. The contents of your *README* file are automatically shown on the front page of your repository.

Let's commit a change to the *README* file.

1. In your repository's list of files, select **README.md**.

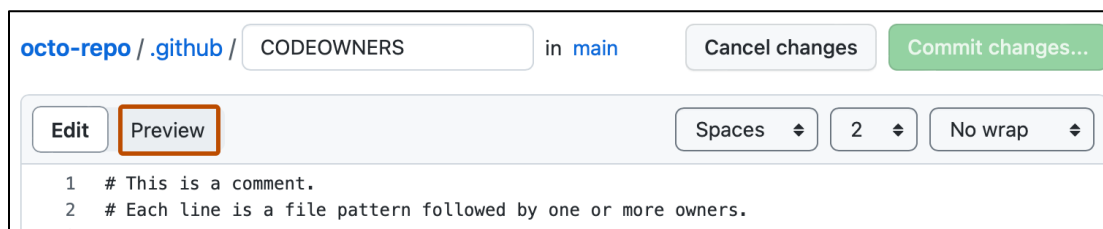


2. In the upper right corner of the file view, click to open the file editor.

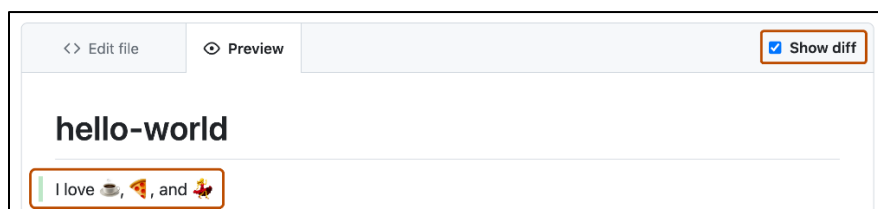


3. In the text box, type some information about yourself.

4. Above the new content, click **Preview**.



5. Review the changes you made to the file. If you select **Show diff**, you will see the new content in green.



6. Click **Commit changes...**

7. In the "Commit message" field, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message.

8. Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is the default branch, you should choose to create a new branch for your commit and then create a pull request.

☐ Commit directly to the `main` branch.

☒ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Propose changesCancel

9. Click **Commit changes** or **Propose changes**.

## Next steps

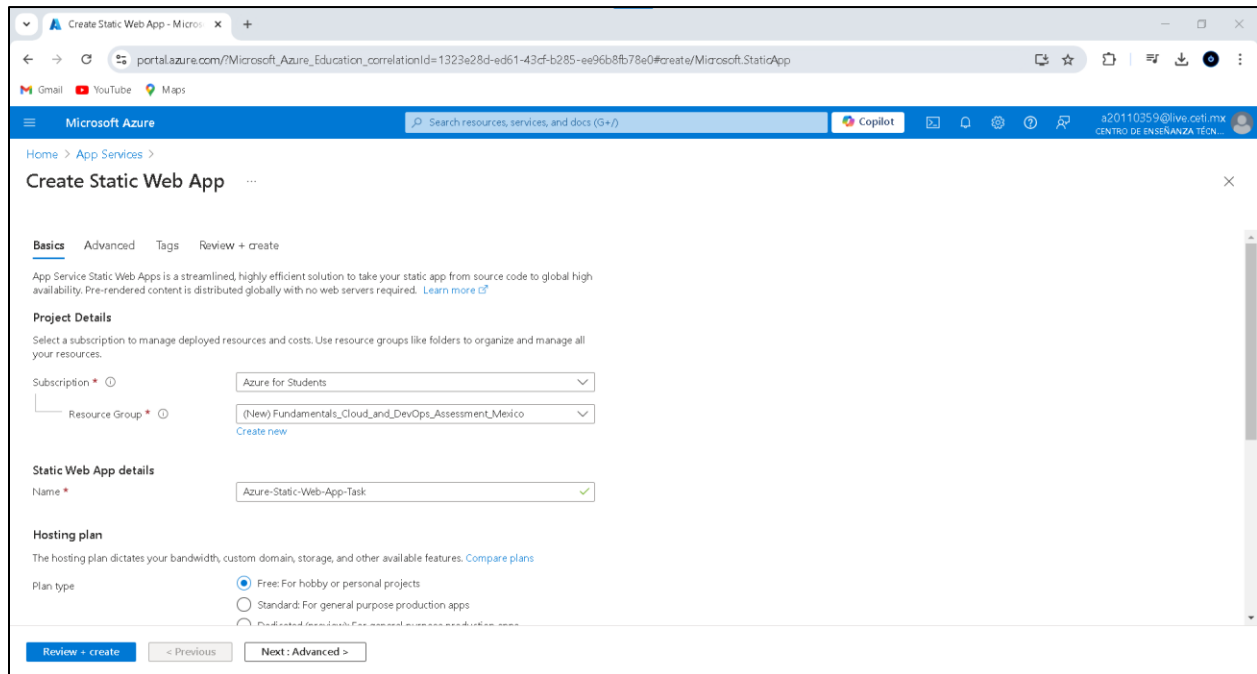
You have now created a repository, including a *README* file, and created your first commit on GitHub.

- You can now clone a GitHub repository to create a local copy on your computer. From your local repository you can commit and create a pull request to update the changes in the upstream repository.
- You can find interesting projects and repositories on GitHub and make changes to them by creating a fork of the repository. Forking a repository will allow you to make changes to another repository without affecting the original.
- Each repository on GitHub is owned by a person or an organization. You can interact with the people, repositories, and organizations by connecting and following them on GitHub.
- GitHub has a great support community where you can ask for help and talk to people from around the world. Join the conversation on [GitHub Community](#).

**Create the files to generate the simple web site inside your cloned GitHub repository.**

### 3. Deploy an application with GitHub Actions and Azure Static Web Apps.

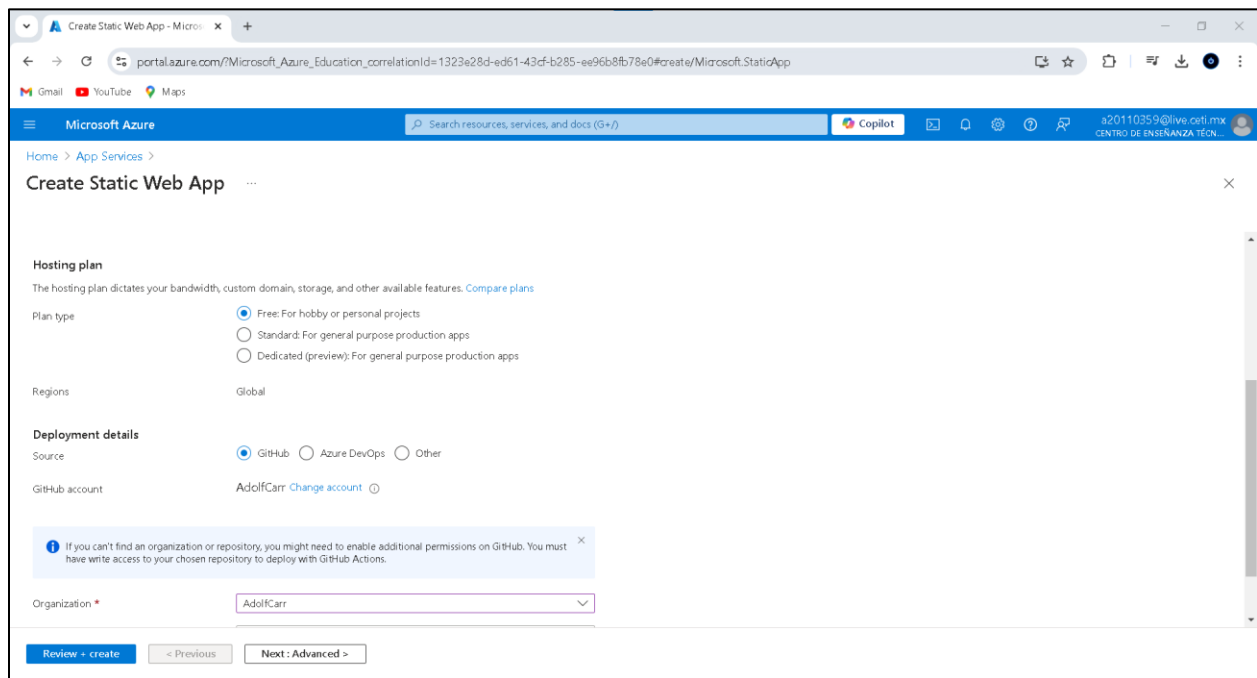
Give a resource Group name at the **Static Web App details**.



The screenshot shows the 'Create Static Web App' wizard in the Azure Portal, specifically the 'Basics' tab. The 'Project Details' section is active, showing the 'Subscription' as 'Azure for Students' and the 'Resource Group' as '(New) Fundamentals\_Cloud\_and\_DevOps\_Assessment\_Mexico'. The 'Static Web App details' section shows the 'Name' as 'Azure-Static-Web-App-Task'. The 'Hosting plan' section shows the 'Plan type' as 'Free: For hobby or personal projects'. The 'Review + create' button is visible at the bottom.

Figure 2 Deploy of the Static Web App on the Azure Portal

Select the **Plan type**, **Deployment details** and then select the **Organization**.



The screenshot shows the 'Create Static Web App' wizard in the Azure Portal, specifically the 'Advanced' tab. The 'Hosting plan' section shows the 'Plan type' as 'Free: For hobby or personal projects'. The 'Deployment details' section shows the 'Source' as 'GitHub' and the 'GitHub account' as 'AdolfCarr'. The 'Organization' field is set to 'AdolfCarr'. A message box indicates that if the user can't find an organization or repository, they might need to enable additional permissions on GitHub. The 'Review + create' button is visible at the bottom.

Figure 3 Deploy of the Static Web App on the Azure Portal

Select the desired **Repository** to deploy on Azure Cloud services and choose the appropriate branch from which you will deploy the project. Additionally, select the type of **Build Presets** for your project. Complete the remaining required information, and then click the **Next: Advanced** button.

The screenshot shows the 'Create Static Web App' wizard in the Azure Portal. The 'Basics' tab is active, displaying the following configuration:

- Organization:** AdolCarr
- Repository:** Simple-personal-website
- Branch:** WIP
- Build Presets:** HTML
- App location:** /
- Api location:** e.g. "api", "functions", etc...
- Output location:** /

At the bottom, the 'Next: Advanced' button is highlighted, indicating the next step in the deployment process.

Figure 4 Deploy of the Static Web App on the Azure Portal

Select **Azure Functions and staging details** API and continue.

The screenshot shows the 'Advanced' tab of the 'Create Static Web App' wizard. The configuration includes:

- Region for Azure Functions API and staging environments:** Central US
- Distributed Functions (preview):** A checkbox that is currently unchecked.
- Note:** Distributed functions requires a standard hosting plan. [Change hosting plan](#)

The 'Next: Tags' button is highlighted at the bottom, indicating the next step in the deployment process.

Figure 5 Deploy of the Static Web App on the Azure Portal



Set **Tags** if needed, read the hint for more information about it, then continue.

The screenshot shows the 'Create Static Web App' wizard in the Azure Portal, specifically the 'Tags' tab. The page has a blue header with the Microsoft Azure logo and a search bar. Below the header, there's a navigation bar with 'Home' and 'Create Static Web App'. The main content area is titled 'Create Static Web App' and has tabs for 'Basics', 'Advanced', 'Tags', and 'Review + create'. The 'Tags' tab is active, showing a description of tags and a form to add them. The form has three columns: 'Name', 'Value', and 'Resource'. There are two empty input fields for 'Name' and 'Value', and a dropdown menu for 'Resource' showing '2 selected'. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Review + create >'.

Figure 6 Deploy of the Static Web App on the Azure Portal

Check the details of the instance to be generated and then click on the button **Create**.

The screenshot shows the 'Create Static Web App' wizard in the Azure Portal, specifically the 'Review + create' tab. The page has a blue header with the Microsoft Azure logo and a search bar. Below the header, there's a navigation bar with 'Home' and 'Create Static Web App'. The main content area is titled 'Create Static Web App' and has tabs for 'Basics', 'Advanced', 'Tags', and 'Review + create'. The 'Review + create' tab is active, showing a 'Summary' section with the 'Static Web App' icon and a 'Details' section with a table of configuration details. At the bottom, there are buttons for 'Create', '< Previous', 'Next >', and a link to 'Download a template for automation'.

Details	
Subscription	dabdfabf-5d78-4199-913b-32d035882649
Resource Group	Azure-Practical-Task-Static-Web-App_group
Name	Azure-Practical-Task-Static-Web-App
Region	centralus
SKU	Free
Repository	https://github.com/AdolfCarr/Simple-personal-website
Branch	WIP
App location	/
API location	/
Output location	/

Figure 7 Deploy of the Static Web App on the Azure Portal

Now your site web is completely deployed on the Cloud Azure. Click on **Go to resource**.

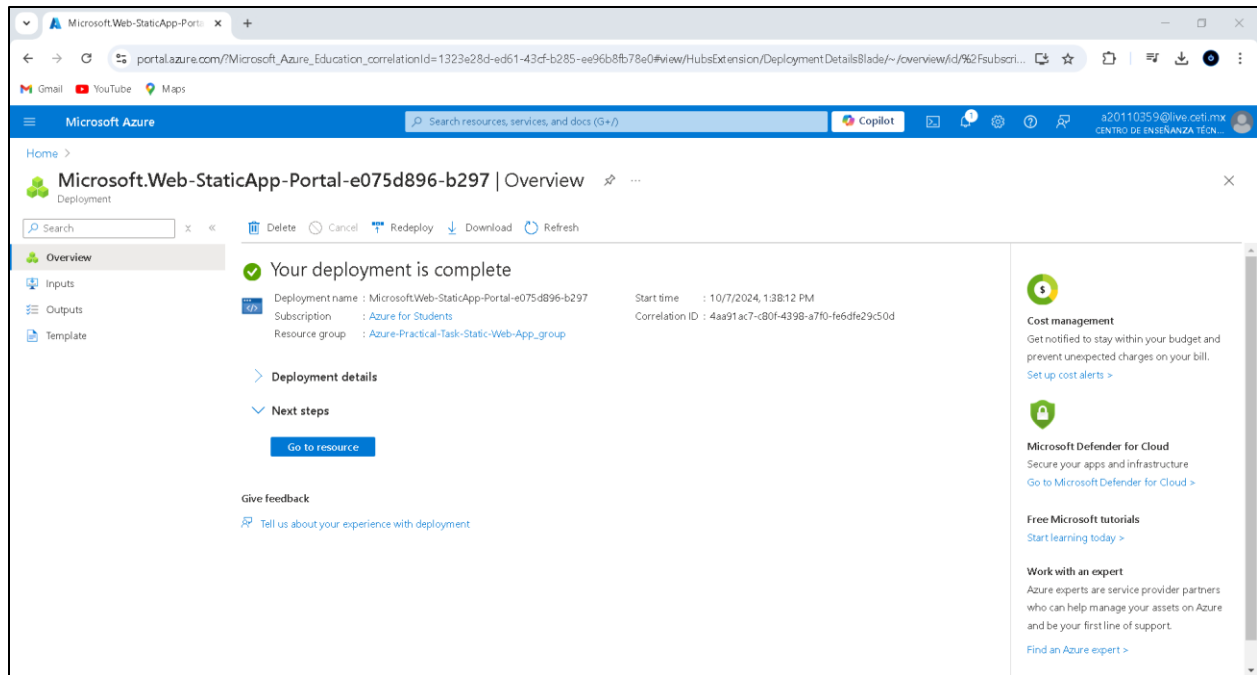


Figure 8 Deploy of the Static Web App on the Azure Portal

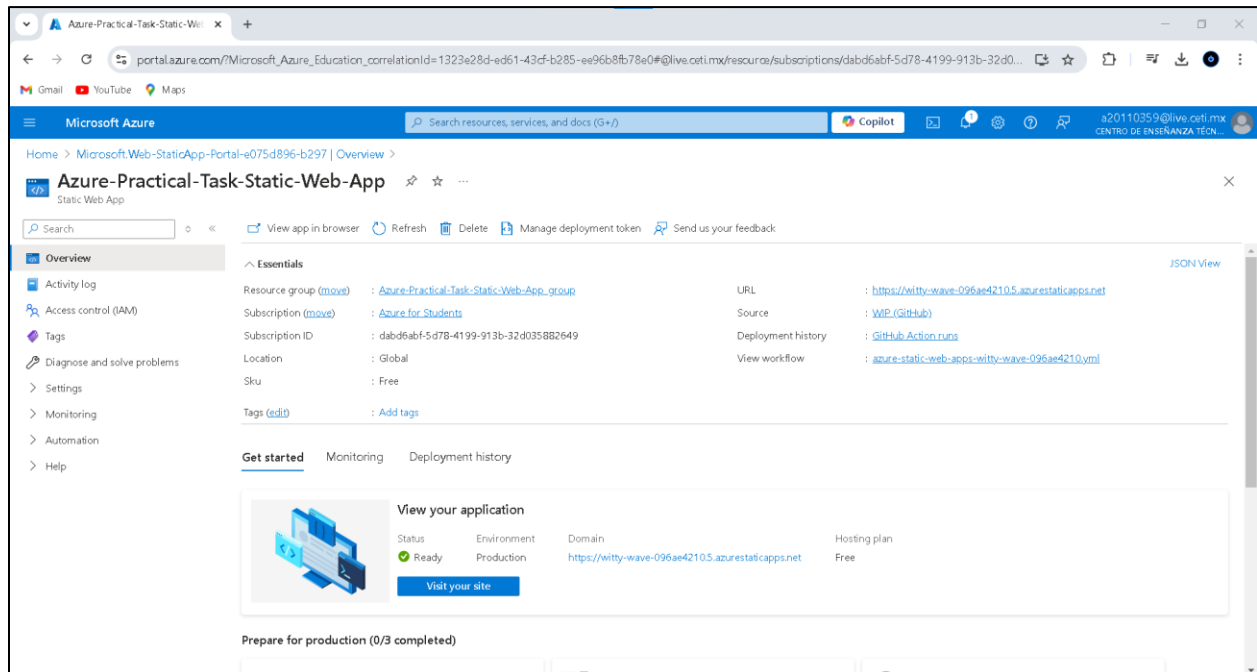
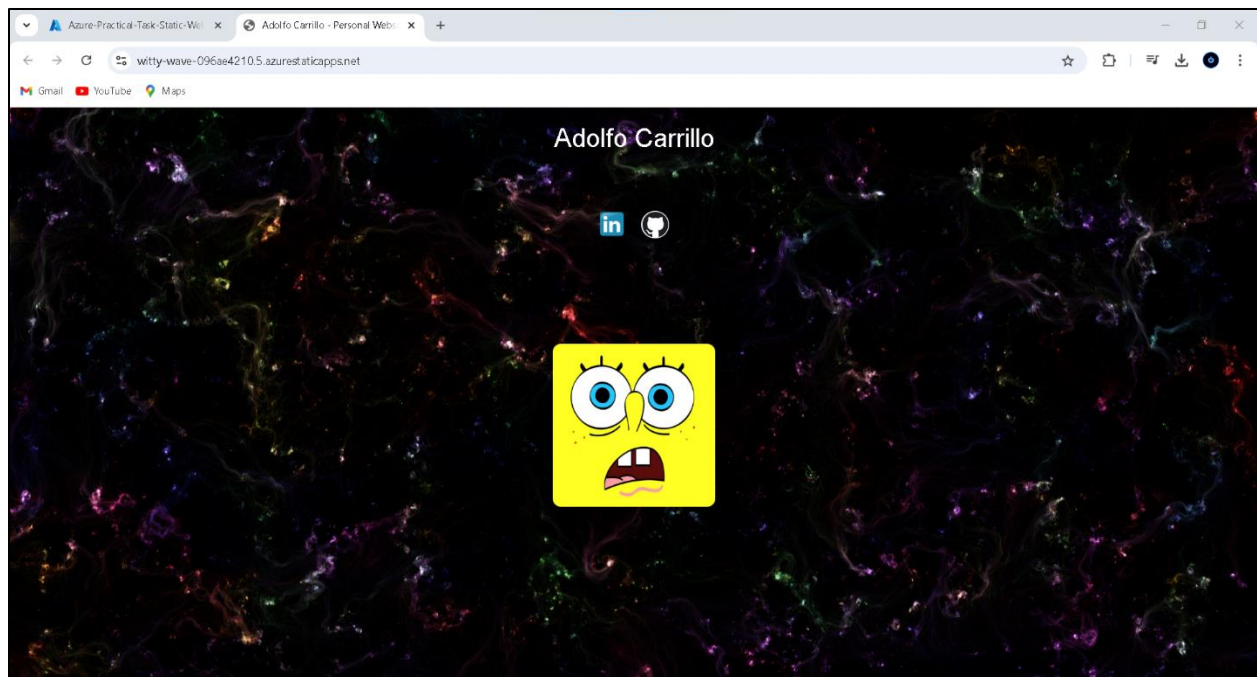


Figure 9 Deploy of the Static Web App on the Azure Portal

Here pressing on the **Visit your site** button, you can see the static app deployed on Cloud Azure under the DNS provided by Microsoft.

Here is the Azure Static Web App developed for this task.



*Figure 10 Deployed of the Static Web App on the Azure Portal*

This is the web site address provided by Microsoft with the DNS azurestaticapps.net:

<https://witty-wave-096ae4210.5.azurestaticapps.net/>

Direct link to the deployed web site: [Adolfo Carrillo](#)

The repository was set with an access modifier of type 'Public' because a non-individual reviewer was assigned. Here next is the address to the repository used on this deployment.

GitHub repository used on this deployment: [Simple-personal-website](#)

## Important

Please note that the CI/CD jobs and actions are configured and managed by Azure Cloud. This configuration can be found in the root directory of the project's GitHub repository, specifically in the `.github/workflows` folder. Inside this folder, you will find the `.yml` file that automates the entire deployment workflow on Azure Cloud.

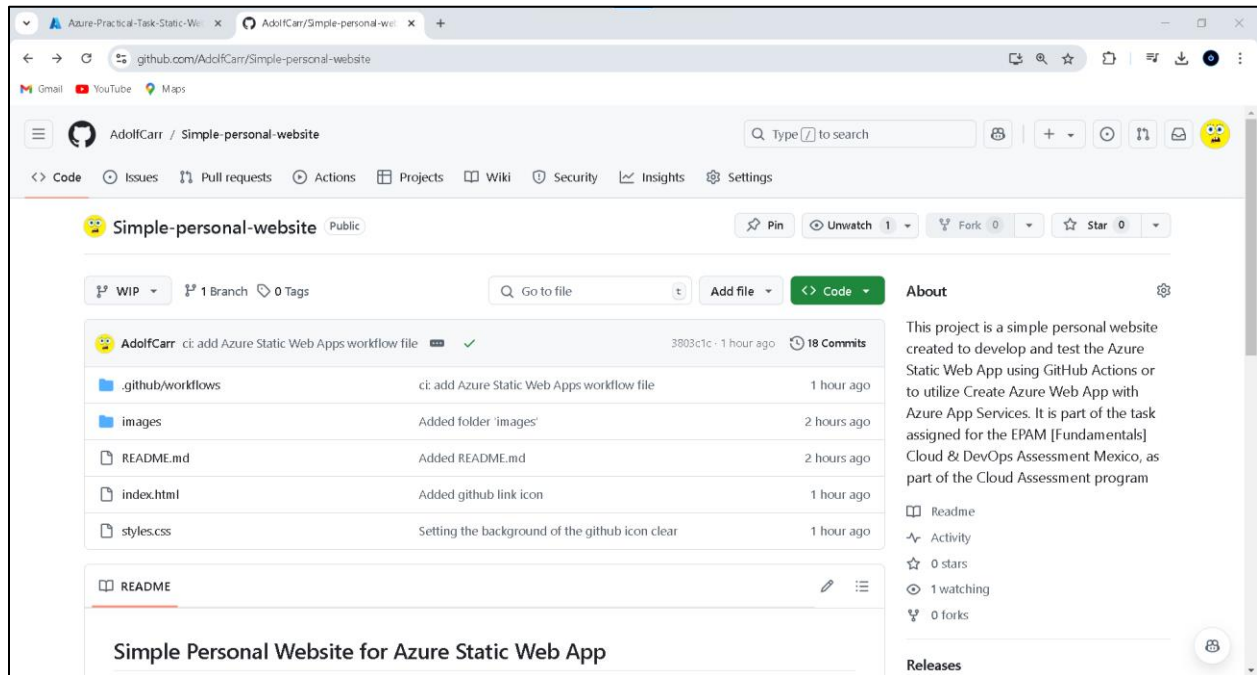


Figure 11 GitHub repository of the Static Web App

## References

- GitHub, Inc. (2024). *creating-and-managing-repositories/quickstart-for-repositories* . Retrieved from docs.github.com: <https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories?tool=webui>
- Microsoft. (2024). *Azure Static Web Apps* . Retrieved from learn.microsoft.com: <https://learn.microsoft.com/en-us/training/paths/azure-static-web-apps/>
- Planet, D. (2023, October 03). *Create Azure Static Web App service with GitHub | Azure Static web app*. Retrieved from www.youtube.com: <https://www.youtube.com/watch?v=XM6h2l3ln0E>