SUMMARY, ARCHITECTURE AND CONFIGURATIONS

Static Website Deployment Using Azure Blob Storage and GitHub Actions

A DevOps-Focused Capstone Implementation

By:

Chibueze Benjamin Chikelu

TechCrush Capstone Project – Group 3

SUMMARY

This capstone project demonstrates how I deployed a static website to Azure Blob Storage and automated the deployment using GitHub Actions. As a beginner DevOps engineer, my goal was to understand and implement a real-world CI/CD workflow while learning how Azure cloud resources are provisioned and authenticated using the Azure CLI.

I started by selecting a static website template from GitHub, modifying it to reflect my identity and skills, and preparing it for cloud deployment. The deployment was initially performed manually using Azure CLI and a Bash script. After confirming that the site was live, I implemented automation through GitHub Actions, which now redeploys the site automatically whenever I push updates to the **main** branch.

During the project, I encountered several real-world issues such as Azure login MFA errors, missing RBAC permissions, subscription access challenges, and service principal authentication difficulties. These challenges helped me deepen my understanding of Azure identity, role-based access control (RBAC), and secure deployments using secrets in CI/CD pipelines.

This documentation explains the architecture, tools, deployment steps, error troubleshooting, and automation pipeline used to complete the project.

PROJECT ARCHITECTURE

The architecture of this project is a **three-layered DevOps workflow** designed to deploy and manage a static website using **Azure Cloud Services** and **GitHub Actions** for automation.

ARCHITECTURE COMPONENTS

Local Development Environment (VS Code + Git Bash)

The project started in a local environment where the static website template (HTML, CSS, Images) was customized.

The folder structure was organized as follows:

Chibueze_capstone/

└── ChibuezeStaticSite/

    ├── index.html

    ├── style.css

    ├── img/

    ├── chibuezeDeploy.sh

    └── .github/workflows/deploy.yml

All local changes were version-controlled using **Git** and pushed to **GitHub**.

Azure Infrastructure
The cloud infrastructure consisted of:

- **Resource Group: ChibuezeCapstone** (container for all cloud resources).
- **Storage Account: chibuezeblob** created using the **az storage account create** command.
- **Static Website Hosting:** Enabled via the Azure CLI
- The website files were uploaded to the **$web** container, which automatically serves them through the public endpoint

Automation Layer (CI/CD using GitHub Actions)

- A **workflow file (deploy.yml)** inside .github/workflows automated the deployment process whenever changes were pushed to the main branch.
- The workflow used the following key steps:

1. **Checkout Repository**
   Use the actions module to pull the latest version of your code.

2. **Azure Login**
   Authenticates to Azure using the credentials stored securely in **GitHub Secrets (AZURE_CREDENTIALS)**.

3. **Upload Website Files**
   Executes:

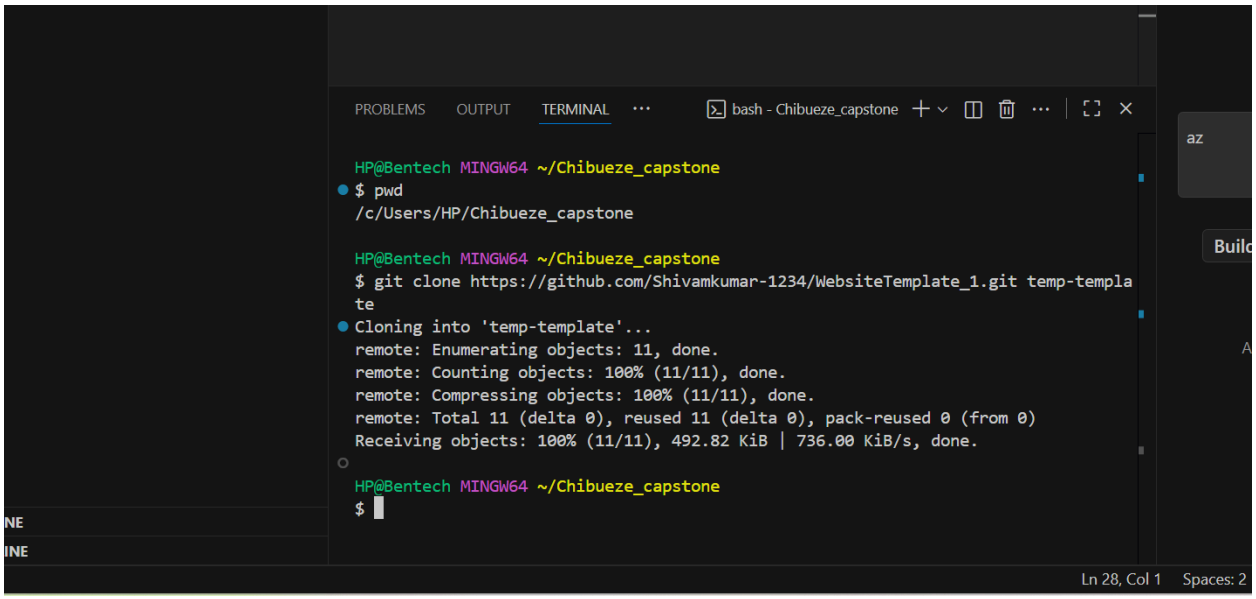
```
az storage blob upload-batch \
--account-name chibuezeblob \
--destination '$web' \
--source . \
--overwrite
```

MAIN PROJECT

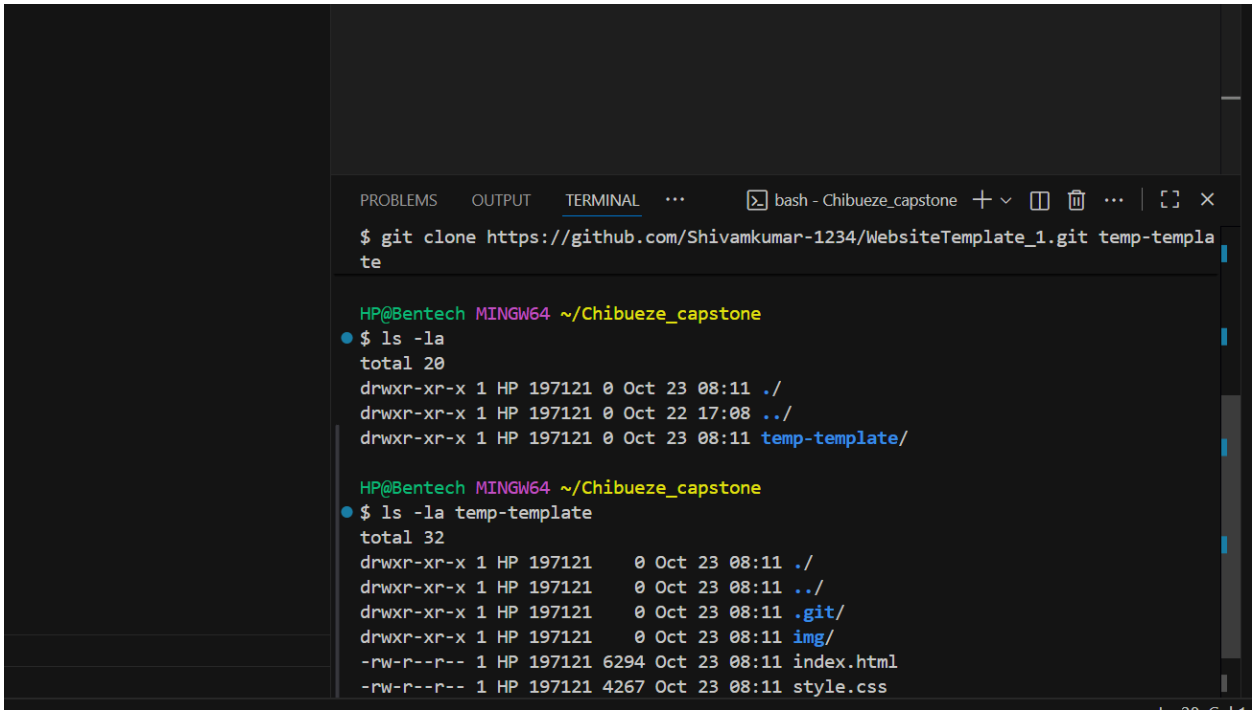
Firstly, I created a parent directory called Chibueze_capstone which will house all the directories and files needed to complete this project

STEP 1 – Cloning the Static Website Template from GitHub

I downloaded (cloned) a static website template containing index.html, style.css and images from GitHub into my local machine. This served as the foundation of my portfolio site before customizing and deploying it.

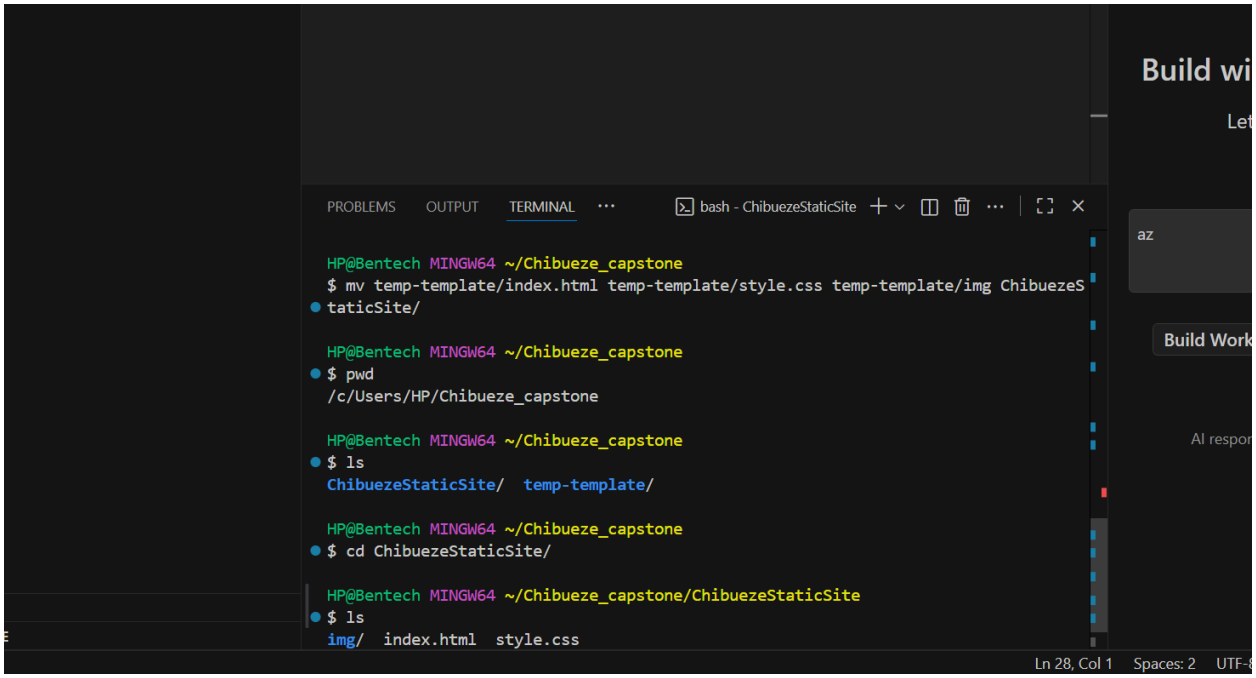


After cloning the template, I performed the list command to make sure the template folder is now in my local environment;



Indeed, the static site template was there in my local environment named **temp-template** and when I listed the contents of the folder, it included the **index.html** file, **style.css**, **img** as well the .git directory showing its from github.

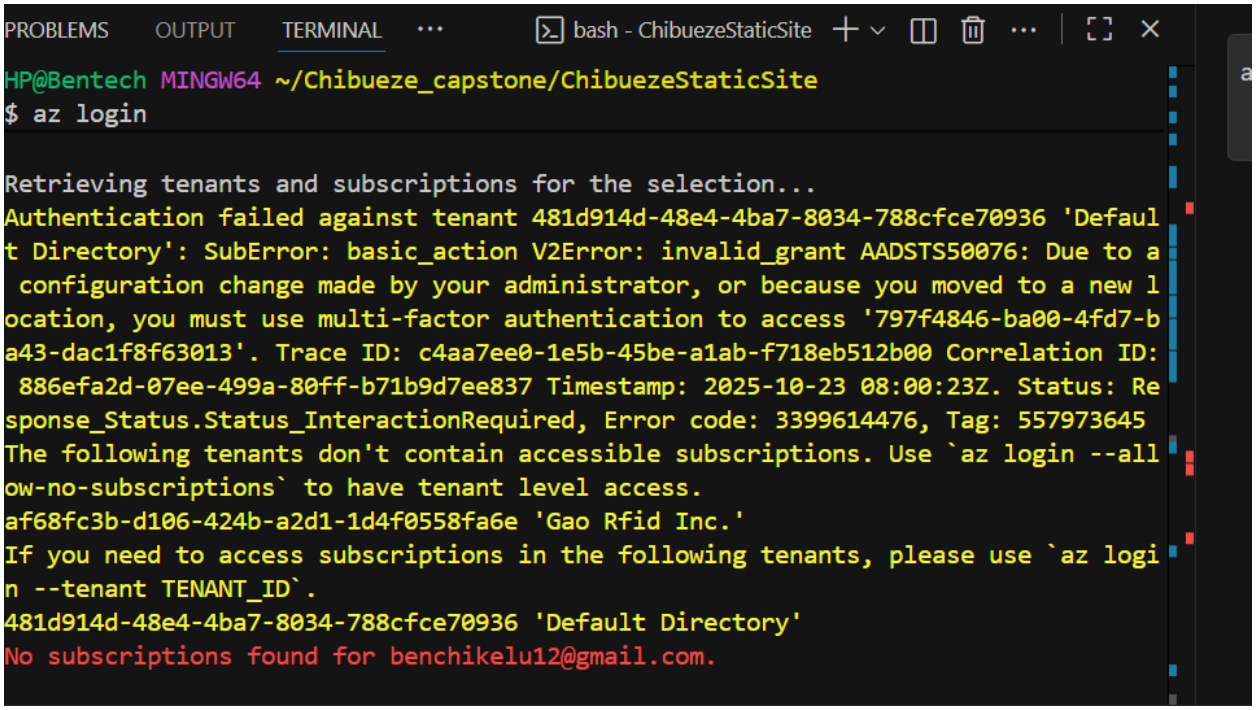
After this, I ran a command that not only created a new directory called **ChibuezeStaticSite** but also moved index.html, style.css and img into the directory as I looked to customize my project. The screenshot attached below shows this process:



I then deleted the temp-template folder and changed directory into ChibuezeStaticSite where I moved the files and then touched a new file called **chibuezedeploy.sh** which will contain my script for this project. I proceeded to customize both the index.html and style.css files to make it more fitting for my personal portfolio-like static site.

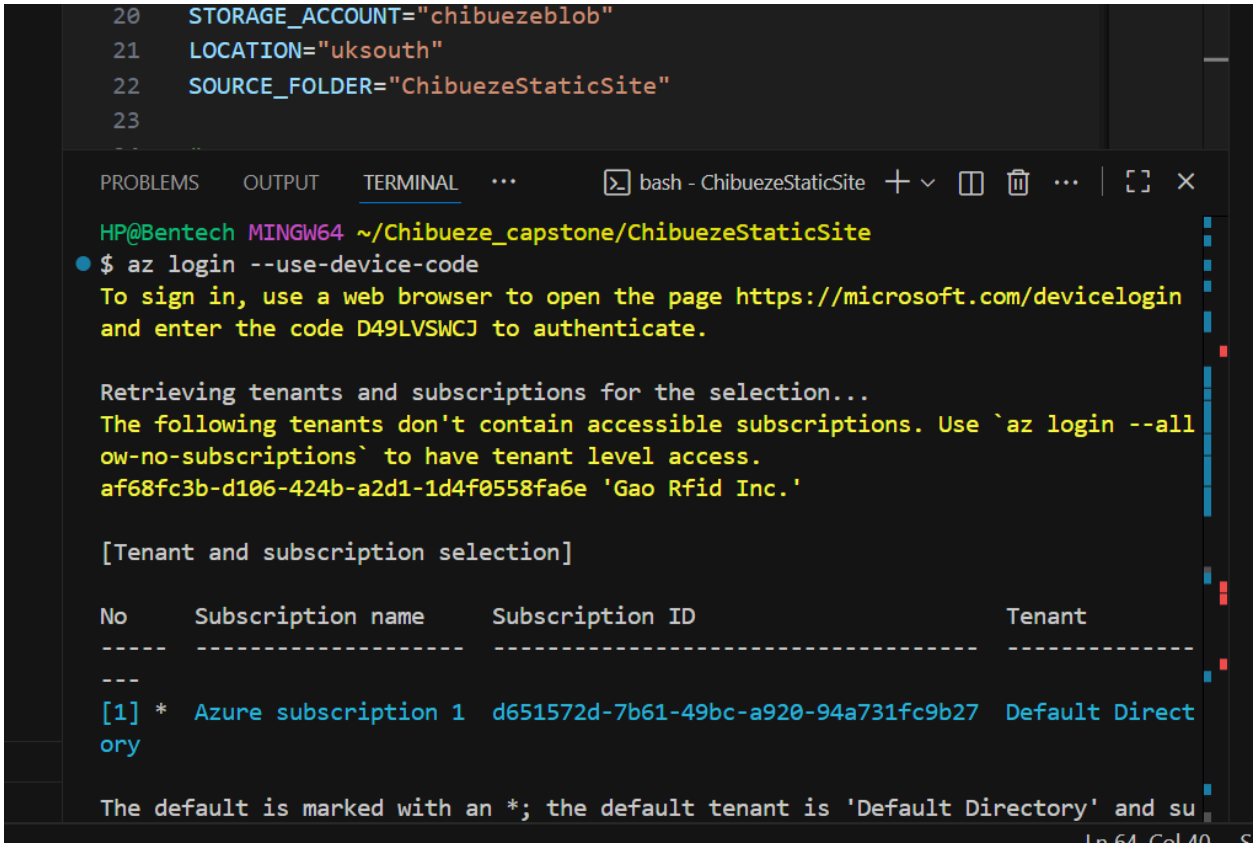
I then proceeded to code Chibuezedeploy.sh where I wrote the bash script to deploy a static website to azure blob storage.

After writing my script, I logged in into my azure account by running the az login command, chmod +x my script and ran it using ./Chibuezedeply.sh , below is the screenshots of what happened after;



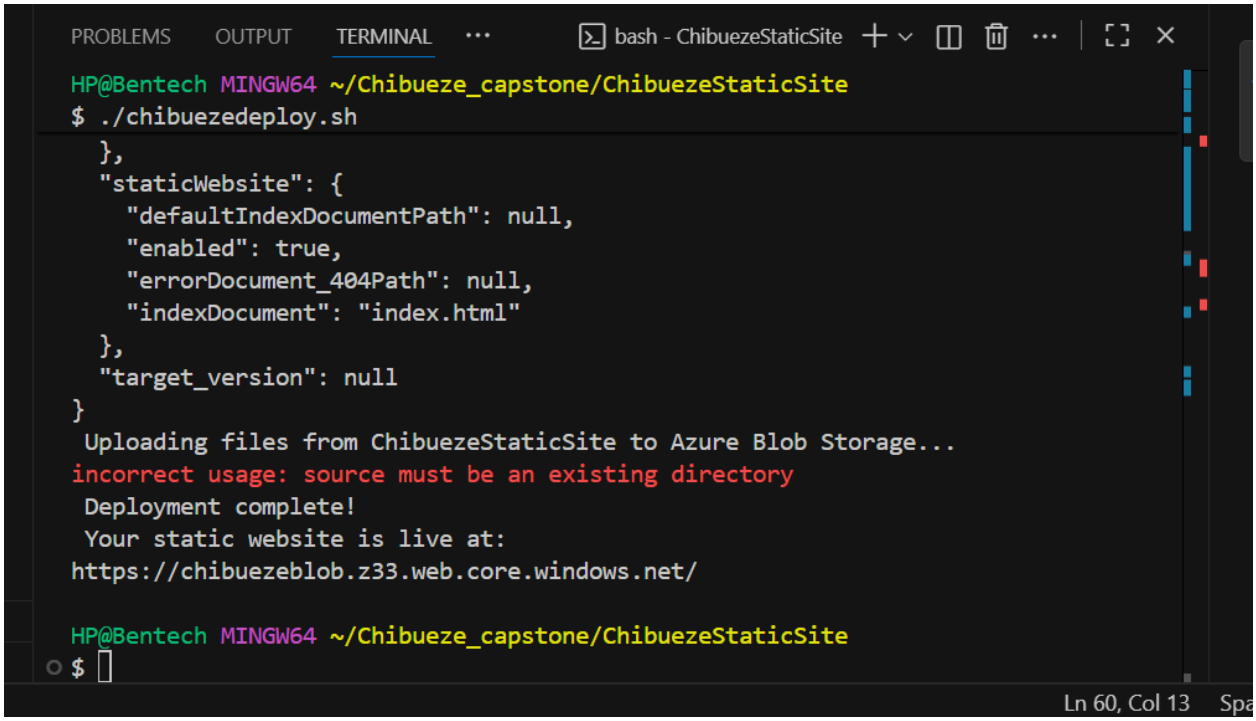
Error showing that I was not on the correct Azure subscription, I had multiple tenants & the CLI couldn't detect which subscription to use. As a result, commands such as az group create and az storage account create threw a MissingSubscription error, which indicated my CLI session was not associated with my chosen Azure subscription.

To resolve this I had to run **az login --use-device-code**. After selecting the correct subscription during login, Azure CLI successfully authenticated me under the right tenant and attached the active subscription. Below is the screenshot;



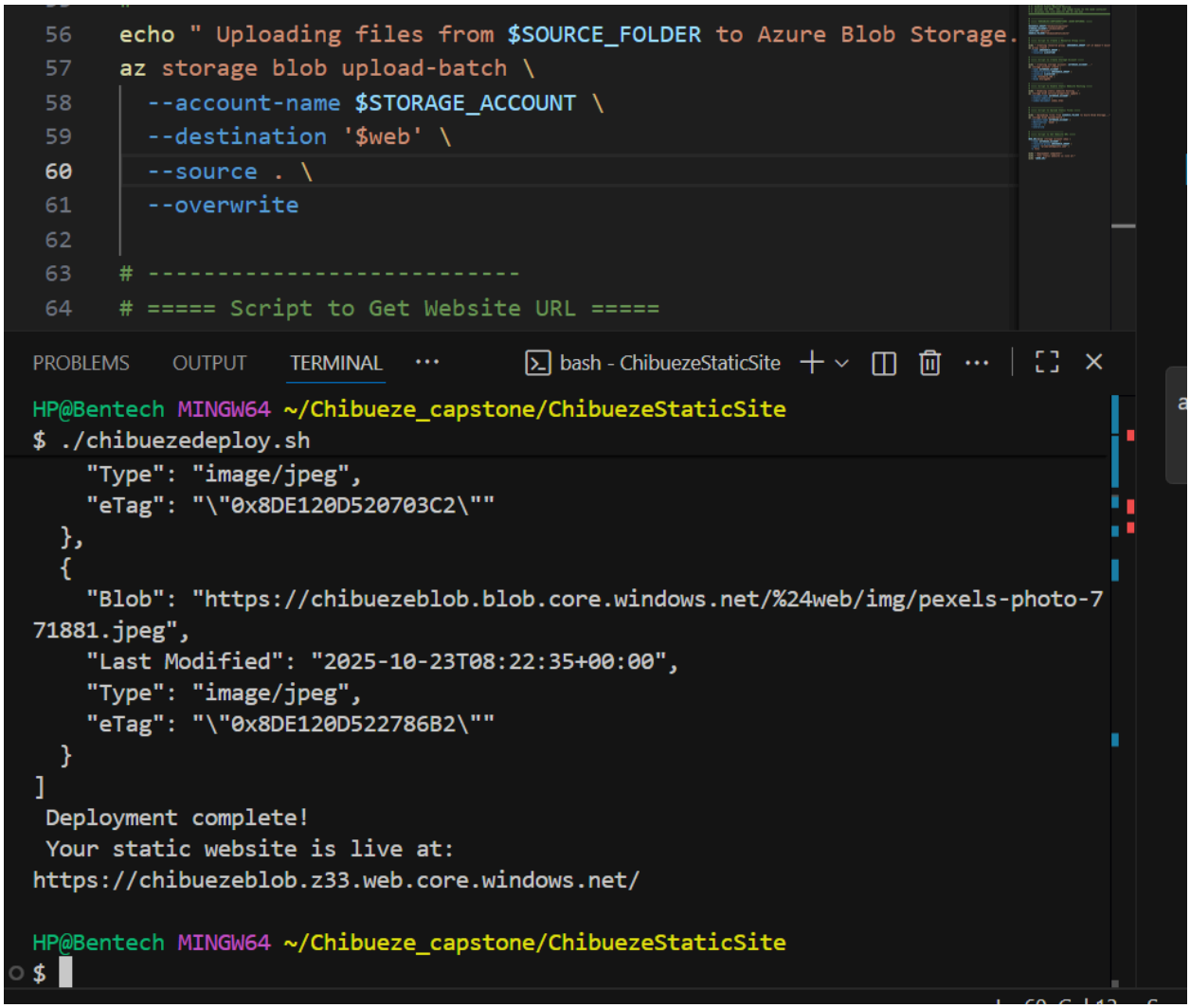
Further more, I used this command **az account set --subscription <Your-Subscription-ID>** to set my particular subscription Id and then used this command **az role assignment create --assignee <Object-ID> --role "Contributor" --scope /subscriptions/<Subscription-ID>** to assign a contributor role required to interact with storage resources using Azure AD authentication. I did this using the powershell because bash sometimes fails to properly interpret assigning roles.

While uploading the source_folder(ChibuezeStaticSite), an error occurred because the deployment script was executed from a directory where the source folder path was incorrect ;

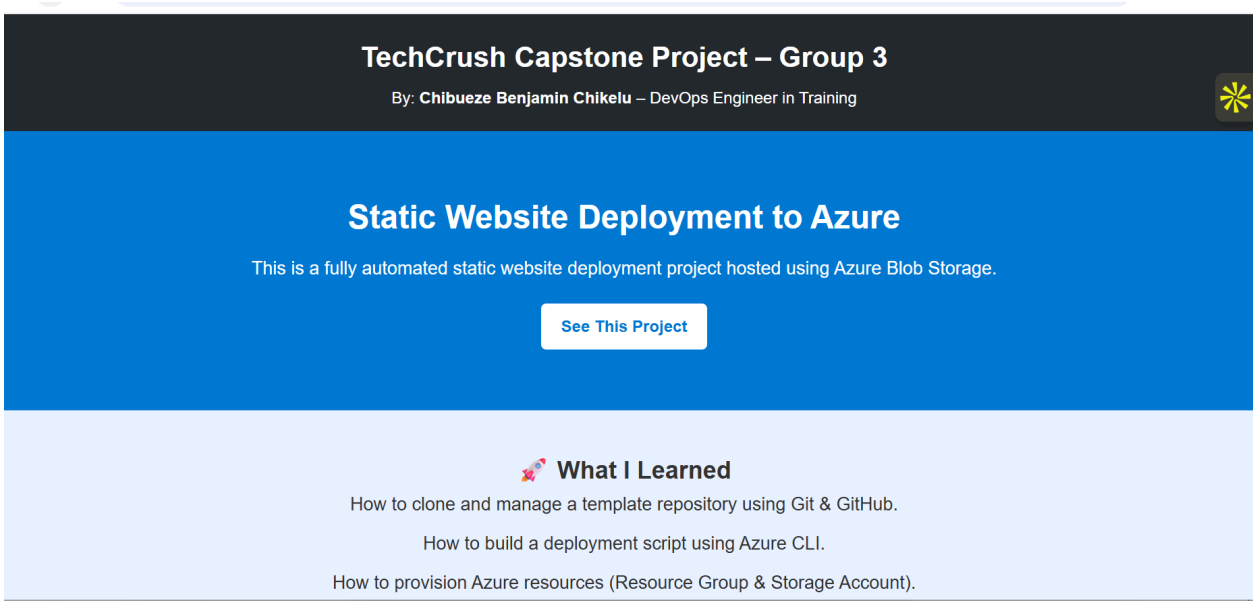


This error occurred because the script attempted to upload files from a source directory (**ChibuezeStaticSite**) based on the assumption that the script was being executed from its parent directory. However, I was already *inside* the **ChibuezeStaticSite** folder when I ran the script, so Azure CLI could not find the folder path as specified.

To fix this, I quickly went to my Chibuezedeploy.sh script and then Instead of changing directories, I modified the script to use "." (dot) as the source location and ran the script again, the screenshot below shows the result:



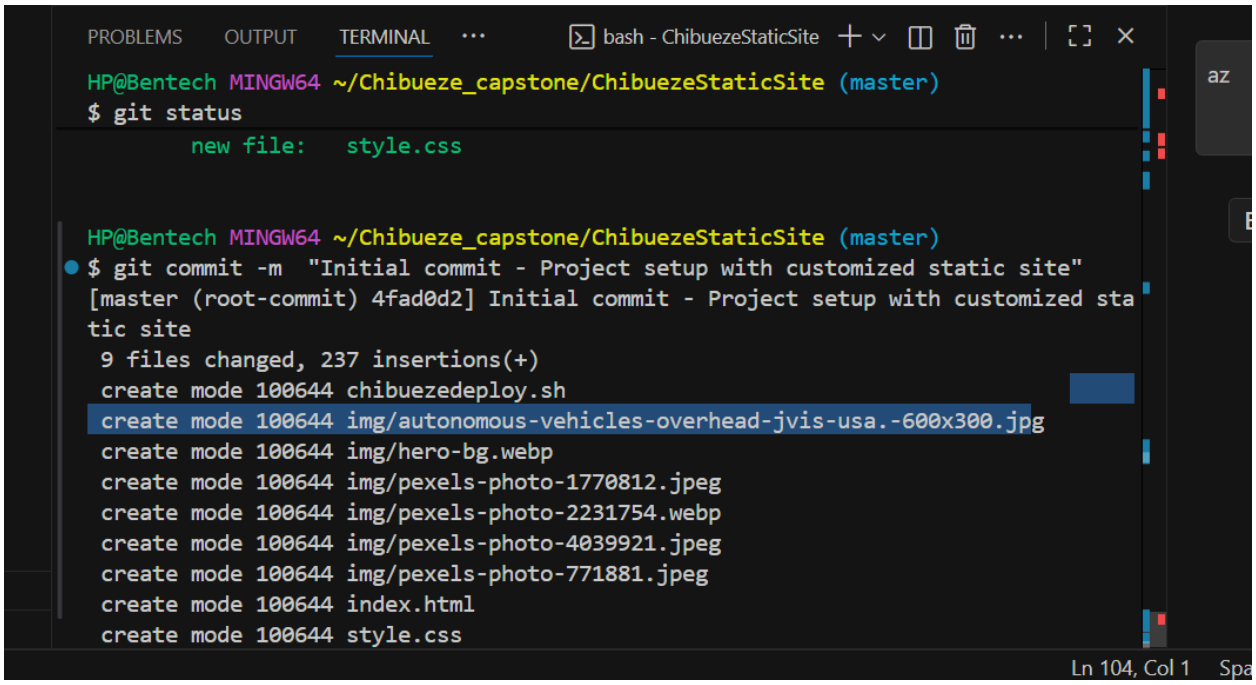
The adding of . in the source made sure the script would run from the current working directory, and yes the script was successfully ran this and my static website URL displayed, my first job was done and I became very happy, below is the screenshot of customized portfolio-like static website after successful deployment;



Well, I had a huge of relief after viewing my site which I customized to be like a portfolio that I would never forget in my life “my first project as a cloud engineer in training”, I was proud of the journey so far but I also had the bonus question in my mind so I set out to equally get that one done.

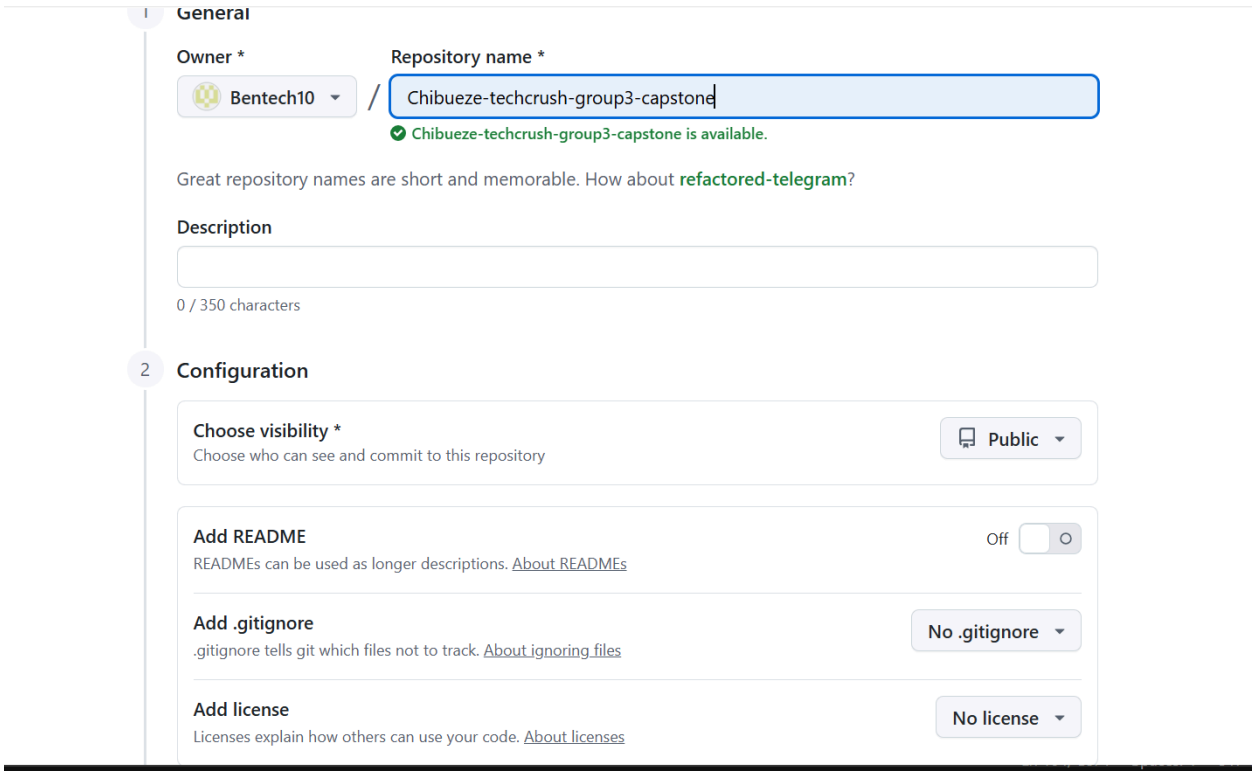
BONUS: Automate deployment with GitHub Actions and ensure uploads happen with new site files on push.

After finishing my deployment, I immediately initialized Git using the git init command and further more added all files for git to see using git add . and the committed the files to git as the screenshot below shows;



The files were successfully commited, and next, I had to create a repository in the github environment.

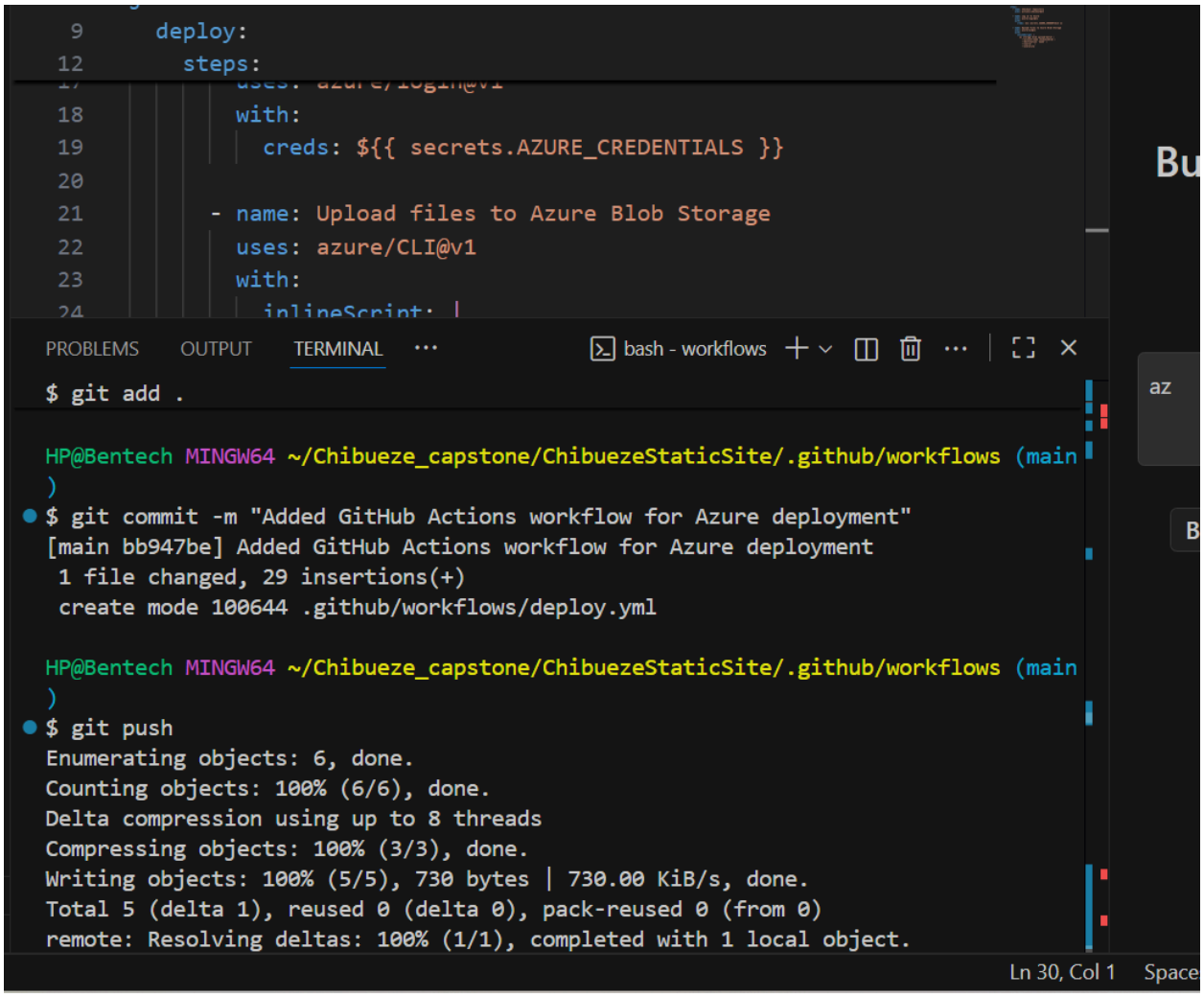
The screenshot below shows the creation of the repository that I'm gonna push all my files to in order to automate the deployment.



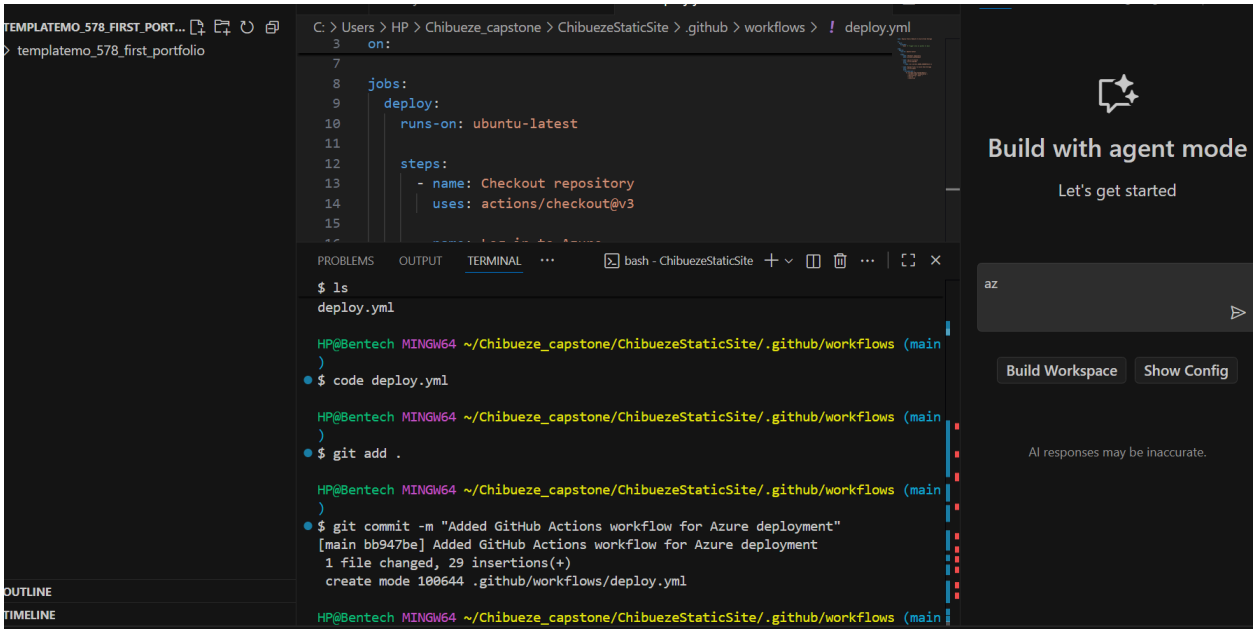
I created a repo with the name **Chibueze-techcrush-group 3 capstone**, after which I pushed my files into the repository by running the commands below one after another;

git remote add origin
https://github.com/Bentech10/Chibueze-techcrush-group3-capstone.git
git branch -M main
git push -u origin main

This allowed me to push the files in the Cli to the github repository where they are now properly tracked.



After pushing the files to github, **mkdir .github/workflows** directories, cd into workflows/ where I then touched deploy.yml, and then went on to add the code.
Below is the screenshot showing the process:



After this, I added, committed and pushed the files once again to github.

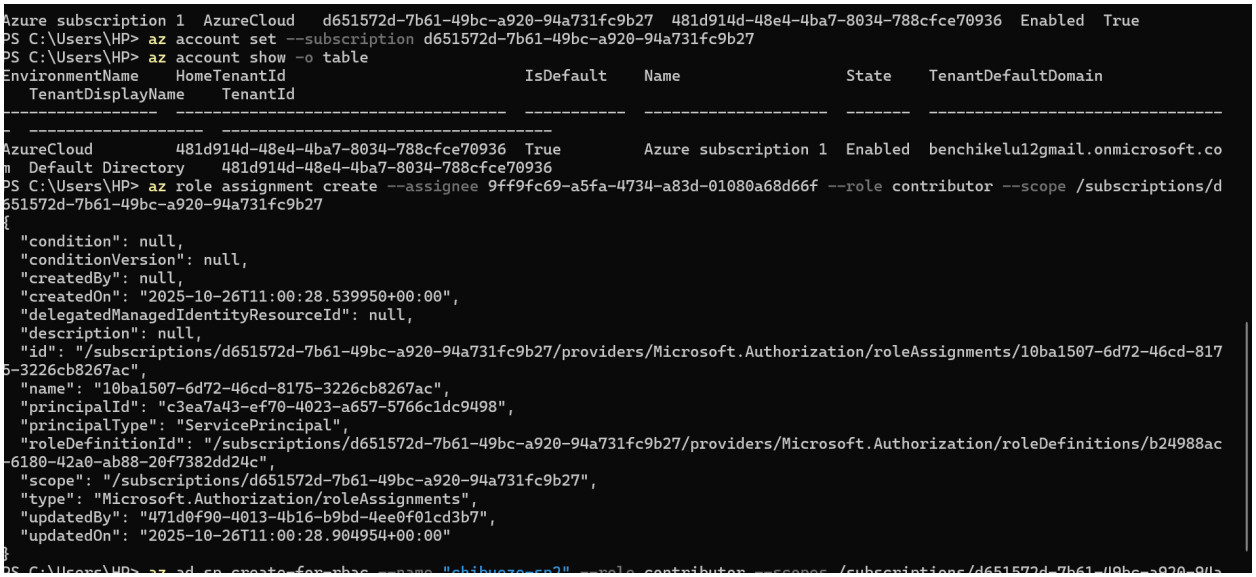
I then set out to generate a Json output which I will add to my github secrets in order to automate deployment by running this command;
az ad sp create-for-rbac --name "chibueze-sp" --role contributor --scopes /subscriptions/d651572d-7b61-49bc-a920-94a731fc9b27 --sdk-auth

This returned an error command that left me sweating for some time. After all back and forth looking for solutions to clear the alone and move ahead, I was able to resolve it
Below is the screenshot of the error ;

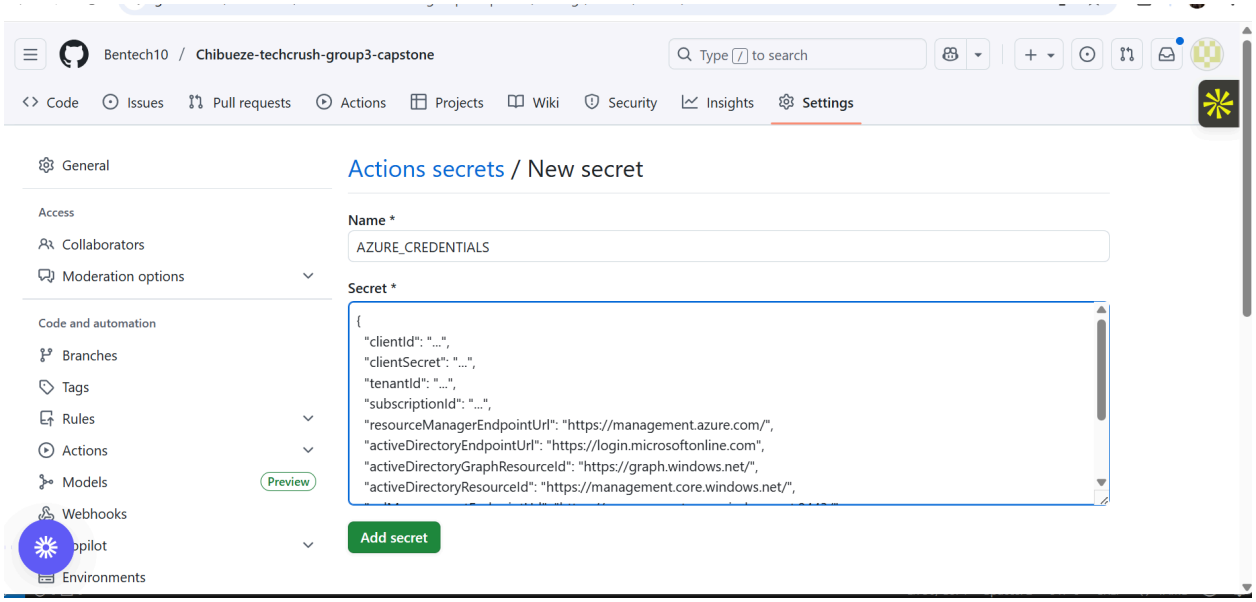
This error usually means: **Azure doesn't know *which subscription* you're trying to work with**, or the request was made at a level that requires a tenant-level provider instead of a subscription-level one.

In a bid to get this sorted, I used the **az account list -o table** to get my subscription ID and further set it as the subscription to use for my deployment by doing **az account set --subscription d651572d-7b61-49bc-a920-94a731fc9b27**.

It still returned same error and after going back and forth for a long time and making so many researches on how to resolve the error, I finally found a breakthrough when it was suggested I run the command using powershell rather than gitbash as gitbash might be misinterpreting the command. I immediately logged to my powershell using **az login --use-devide-code.** I ran the **az ad sp create-for-rbac --name "chibueze-sp" --role contributor --scopes /subscriptions/d651572d-7b61-49bc-a920-94a731fc9b27 --sdk-auth** command once again and it went through, below is the screenshot;

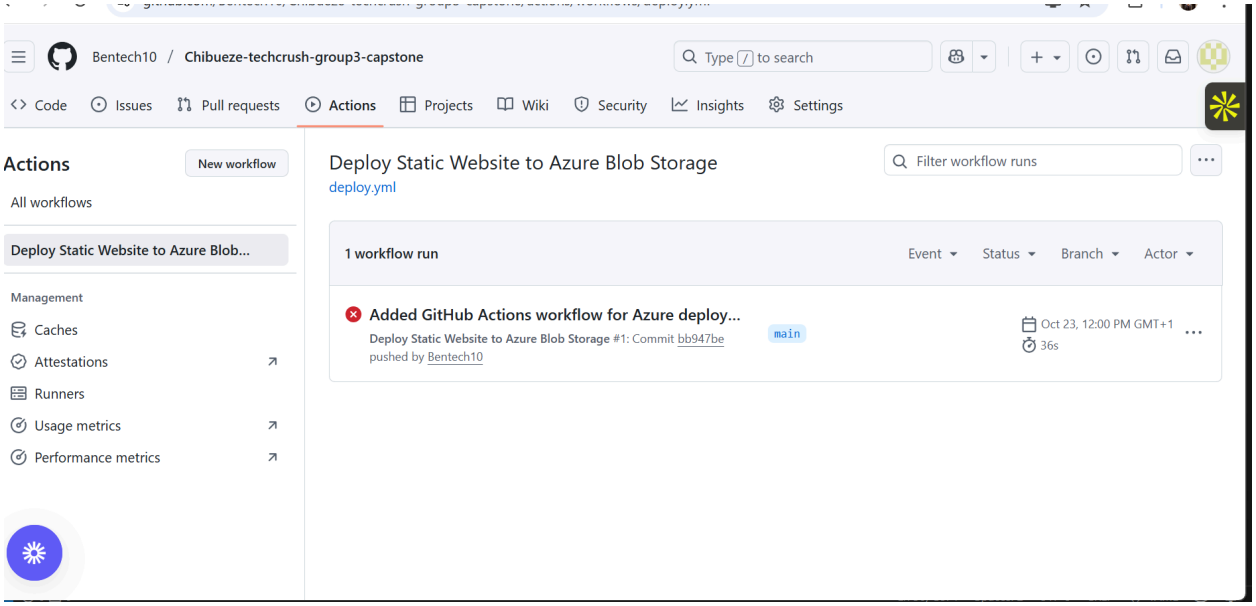


The Json output was finally out after all and I was overjoyed, I then went to github, clicked on my repo, clicked settings and navigated to github secrets where I pasted the json output and created the **azure_credential secret.**



A screenshot showing the json secret output pasted on my github environment.

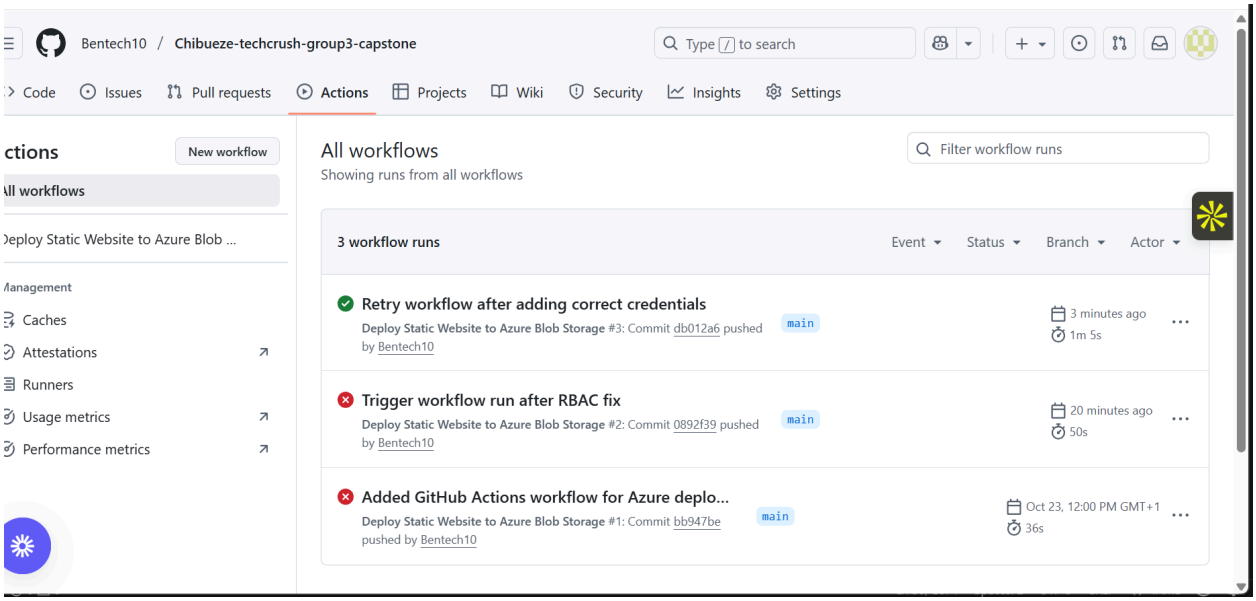
But then when I went to my github environment, clicked on my repo, navigated to actions and met an error message, the screenshot is below;



I tried again and got another failure, so I researched to know why, I finally tried out this command in the screenshot below;

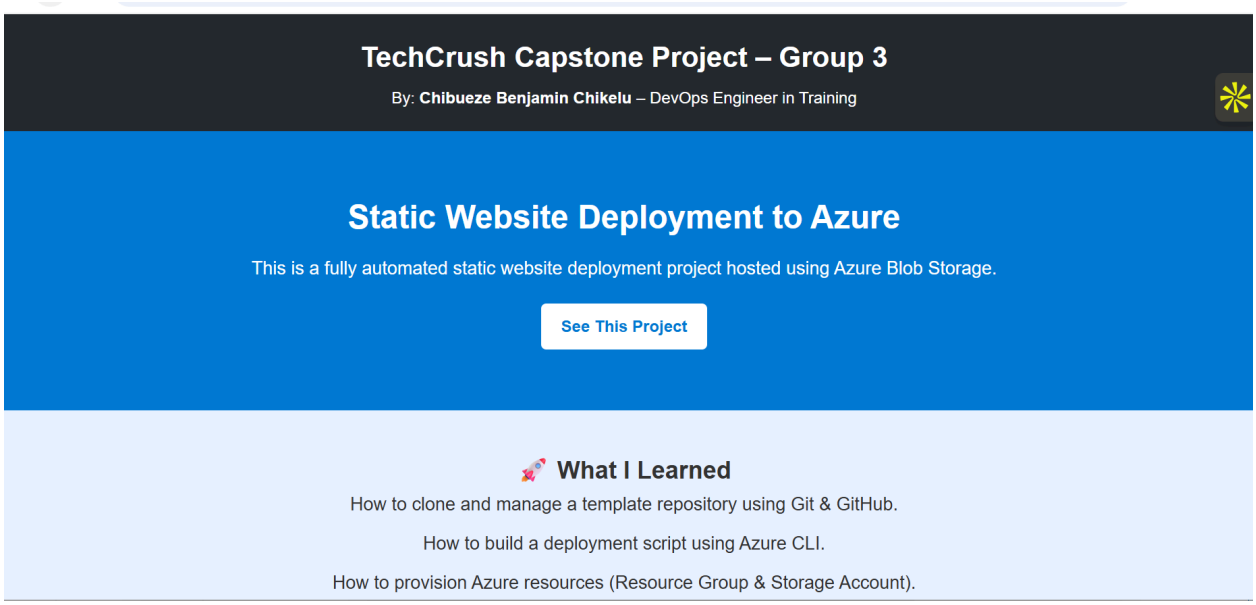
```
PROBLEMS  OUTPUT  TERMINAL  ···        bash - ChibuezeStaticSite

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite (main)
$ echo "# workflow retry" >> retry2.txt
git add .
git commit -m "Retry workflow after adding correct credentials"
git push
warning: in the working copy of 'retry2.txt', LF will be replaced by CRLF the nex
t time Git touches it
[main db012a6] Retry workflow after adding correct credentials
 1 file changed, 1 insertion(+)
 create mode 100644 retry2.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes | 314.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Bentech10/Chibueze-techcrush-group3-capstone.git
   0892f39..db012a6  main -> main

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite (main)
$
```

And this time it went through!, the screenshot below shows everything;

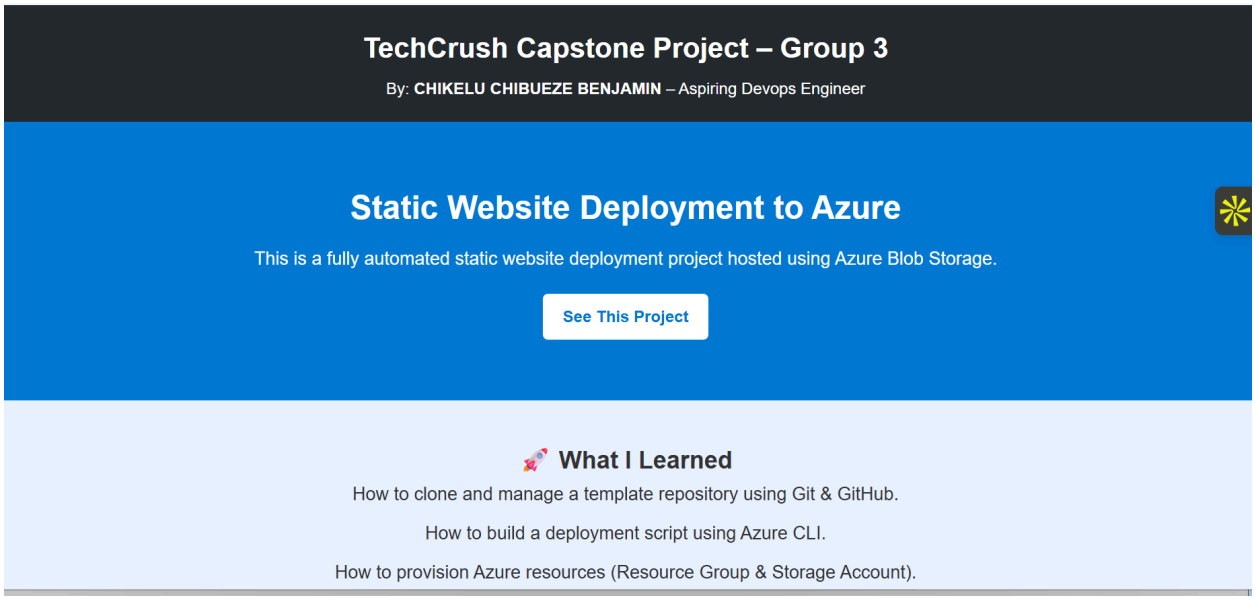


And yes, the deployment worked and my site is still running as well.

To test this I just tweaked a little thing on my index.html file and it worked, I changed my name from Chibueze Benjamin Chikelu that was in the file previously to CHIKELU CHIBUEZE BENJAMIN, I also changed from Devops Engineer in Training to Aspiring Devops Engineer, below is the sites screenshots before and after making changes in the html file that was automatically deployed via github;

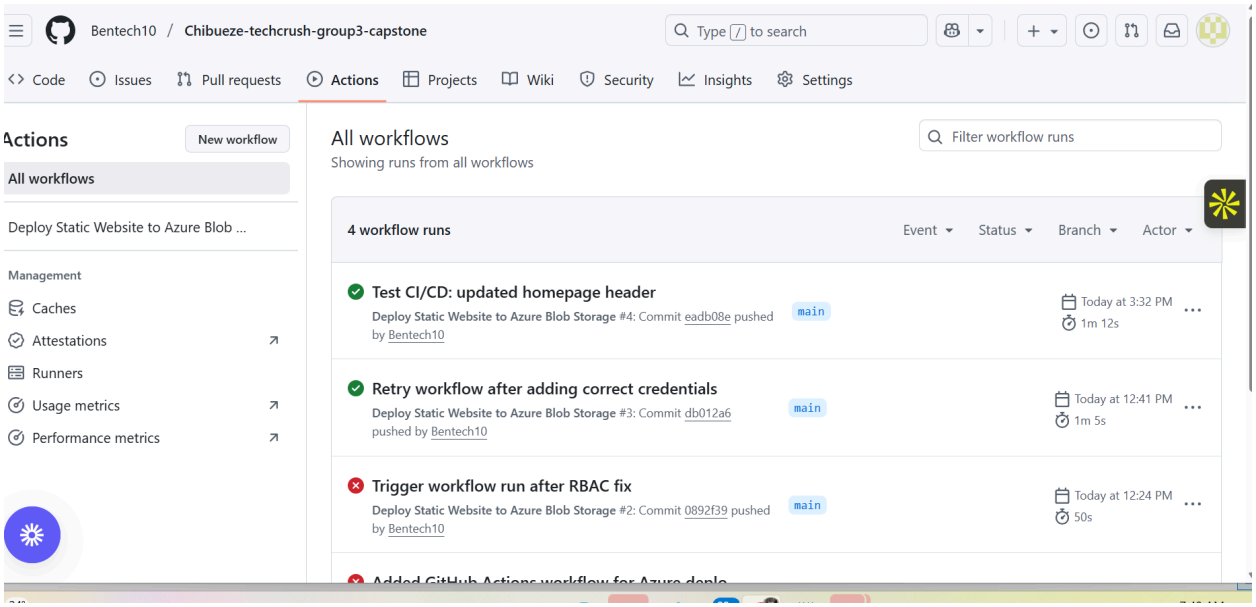


Original static website View!



Website view after a few tweaks that automatically got deployed to github.

To end this amazing experience, below is the screenshot of my github showing that the changes I made got automatically deployed to github; showing that the github workflow is working;



A screenshot showing that the change I made in html file was automatically deployed to github as well!

SUMMARY OF CHALLENGES FACED AND THEIR RESOLUTIONS

Challenge	Description	Resolution
MissingSubscription Error	Occurred during RBAC and service principal creation when the subscription ID was not properly set.	Fixed by setting the active subscription manually using: az account set --subscription <SUBSCRIPTION_ID>

Source Directory Not Found	When the upload command didn’t detect the correct path to website files.	Fixed by changing the source path to the current directory using --source ..
Role Assignment Failure	The SP didn’t have permission to the resource group.	Resolved by switching to PowerShell and re-running the command successfully with full permissions.