

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:

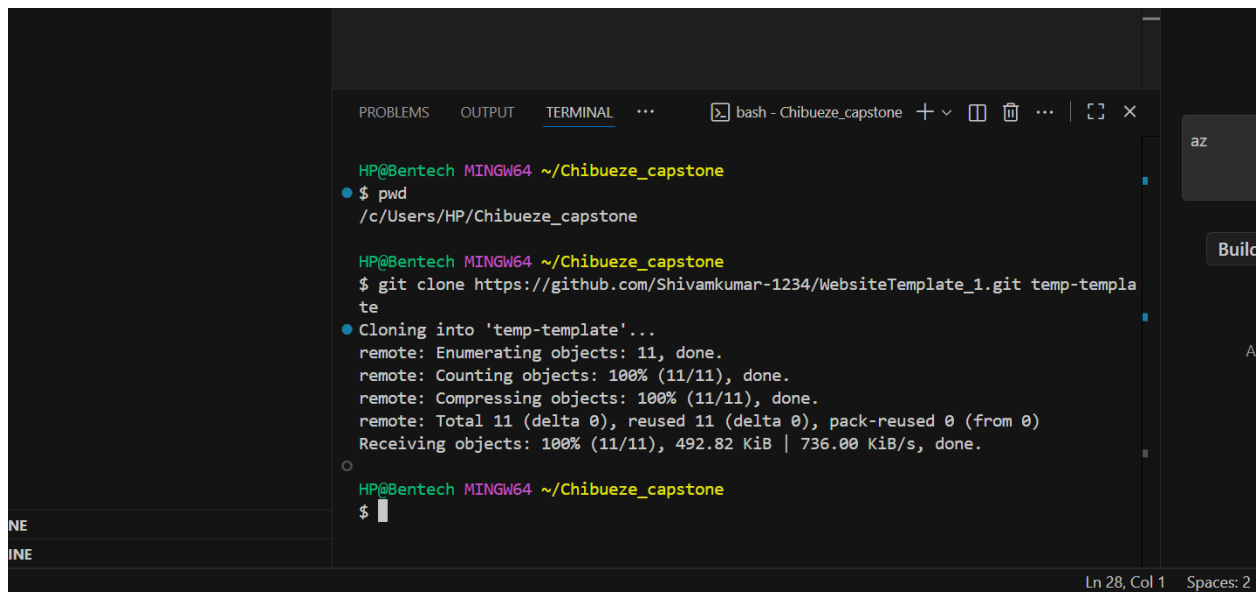
```
az storage blob upload-batch \  
  
--account-name chibuezeblob \  
  
--destination '$web' \  
  
--source . \  
  
--overwrite
```

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.

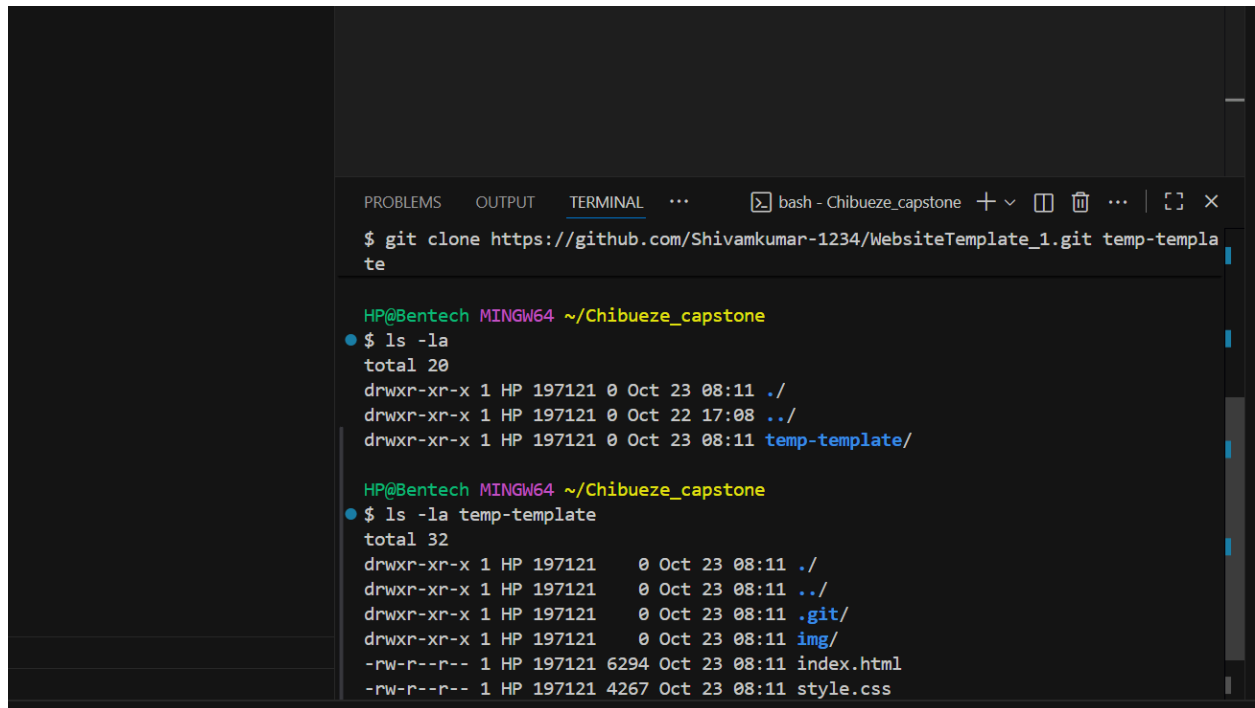
A screenshot of a terminal window with a dark background. The terminal shows the user's current directory as ~/Chibueze_capstone. They run the 'pwd' command, which returns /c/Users/HP/Chibueze_capstone. Then, they run 'git clone https://github.com/Shivamkumar-1234/WebsiteTemplate_1.git temp-template'. The terminal displays the progress of cloning, including enumerating, counting, and compressing objects, and finally receiving the objects. The status bar at the bottom right indicates 'Ln 28, Col 1 Spaces: 2'.

```
HP@Bentech MINGW64 ~/Chibueze_capstone
$ pwd
/c/Users/HP/Chibueze_capstone

HP@Bentech MINGW64 ~/Chibueze_capstone
$ git clone https://github.com/Shivamkumar-1234/WebsiteTemplate_1.git temp-template
Cloning into 'temp-template'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (11/11), 492.82 KiB | 736.00 KiB/s, done.

HP@Bentech MINGW64 ~/Chibueze_capstone
$
```

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



```
PROBLEMS OUTPUT TERMINAL ... bash - Chibueze_capstone + v [ ] [ ] ... | [ ] [ ] x

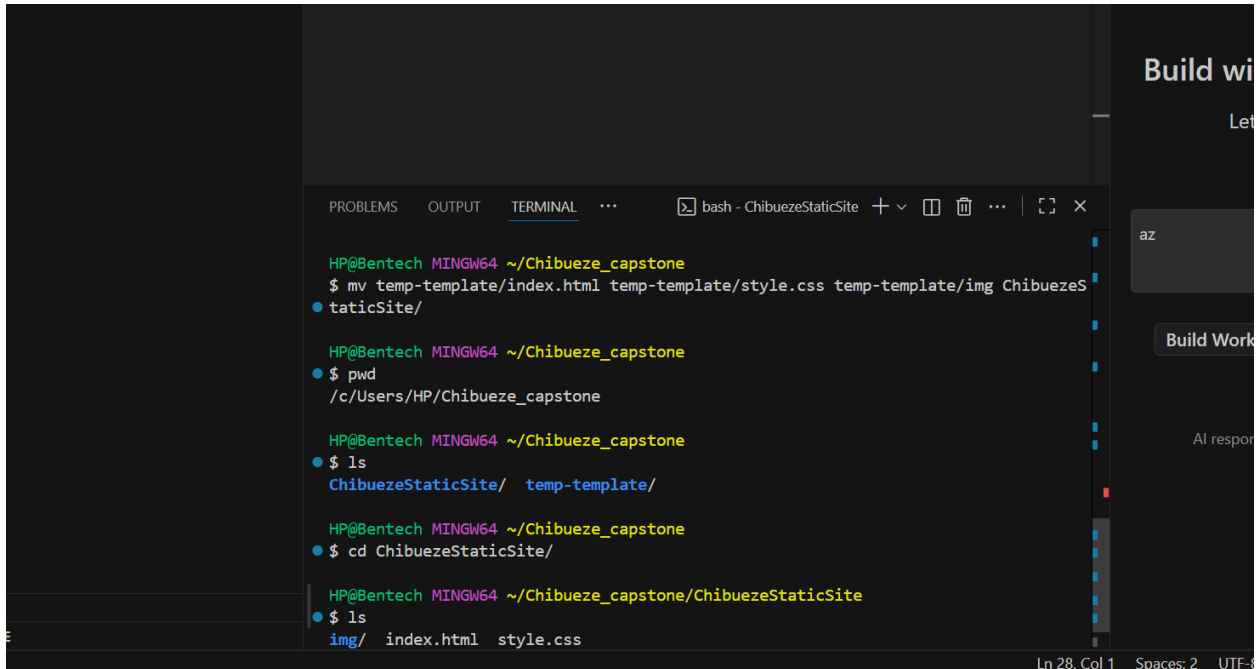
$ git clone https://github.com/Shivamkumar-1234/WebsiteTemplate_1.git temp-templa
te

HP@Bentech MINGW64 ~/Chibueze_capstone
● $ ls -la
total 20
drwxr-xr-x 1 HP 197121 0 Oct 23 08:11 ./
drwxr-xr-x 1 HP 197121 0 Oct 22 17:08 ../
drwxr-xr-x 1 HP 197121 0 Oct 23 08:11 temp-template/

HP@Bentech MINGW64 ~/Chibueze_capstone
● $ ls -la temp-template
total 32
drwxr-xr-x 1 HP 197121 0 Oct 23 08:11 ./
drwxr-xr-x 1 HP 197121 0 Oct 23 08:11 ../
drwxr-xr-x 1 HP 197121 0 Oct 23 08:11 .git/
drwxr-xr-x 1 HP 197121 0 Oct 23 08:11 img/
-rw-r--r-- 1 HP 197121 6294 Oct 23 08:11 index.html
-rw-r--r-- 1 HP 197121 4267 Oct 23 08:11 style.css
```

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:

A screenshot of a Visual Studio Code terminal window. The terminal title bar reads 'bash - ChibuezeStaticSite'. The prompt is 'HP@Bentech MINGW64 ~/Chibueze_capstone'. The user enters a series of commands: first, 'mv temp-template/index.html temp-template/style.css temp-template/img ChibuezeStaticSite/' which moves files from a 'temp-template' directory to 'ChibuezeStaticSite/'; then 'pwd' showing the current directory as '/c/Users/HP/Chibueze_capstone'; then 'ls' showing the contents of the current directory as 'ChibuezeStaticSite/' and 'temp-template/'; then 'cd ChibuezeStaticSite/' to change into that directory; and finally 'ls' again, which shows 'img/' and 'index.html style.css'. The terminal interface includes tabs for 'PROBLEMS', 'OUTPUT', and 'TERMINAL'. On the right side of the editor, there are panels for 'Build with', 'az', 'Build Work', and 'AI respon'. The status bar at the bottom indicates 'Ln 28, Col 1', 'Spaces: 2', and 'UTF-8'.

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;


```
PROBLEMS OUTPUT TERMINAL ... bash - ChibuezeStaticSite + v [] [] ... | [] x
HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite
$ az login

Retrieving tenants and subscriptions for the selection...
Authentication failed against tenant 481d914d-48e4-4ba7-8034-788cfce70936 'Default Directory': SubError: basic_action V2Error: invalid_grant AADSTS50076: Due to a configuration change made by your administrator, or because you moved to a new location, you must use multi-factor authentication to access '797f4846-ba00-4fd7-ba43-dac1f8f63013'. Trace ID: c4aa7ee0-1e5b-45be-a1ab-f718eb512b00 Correlation ID: 886efa2d-07ee-499a-80ff-b71b9d7ee837 Timestamp: 2025-10-23 08:00:23Z. Status: Response_Status.Status_InteractionRequired, Error code: 3399614476, Tag: 557973645
The following tenants don't contain accessible subscriptions. Use `az login --allow-no-subscriptions` to have tenant level access.
af68fc3b-d106-424b-a2d1-1d4f0558fa6e 'Gao Rfid Inc.'
If you need to access subscriptions in the following tenants, please use `az login --tenant TENANT_ID`.
481d914d-48e4-4ba7-8034-788cfce70936 'Default Directory'
No subscriptions found for benchikelu12@gmail.com.
```

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;

```
20 STORAGE_ACCOUNT="chibuezeblob"
21 LOCATION="uksouth"
22 SOURCE_FOLDER="ChibuezeStaticSite"
23

PROBLEMS OUTPUT TERMINAL ... bash - ChibuezeStaticSite + v [ ] [ ] ... [ ] x

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite
$ az login --use-device-code
To sign in, use a web browser to open the page https://microsoft.com/devicelogin
and enter the code D49LVSWCJ to authenticate.

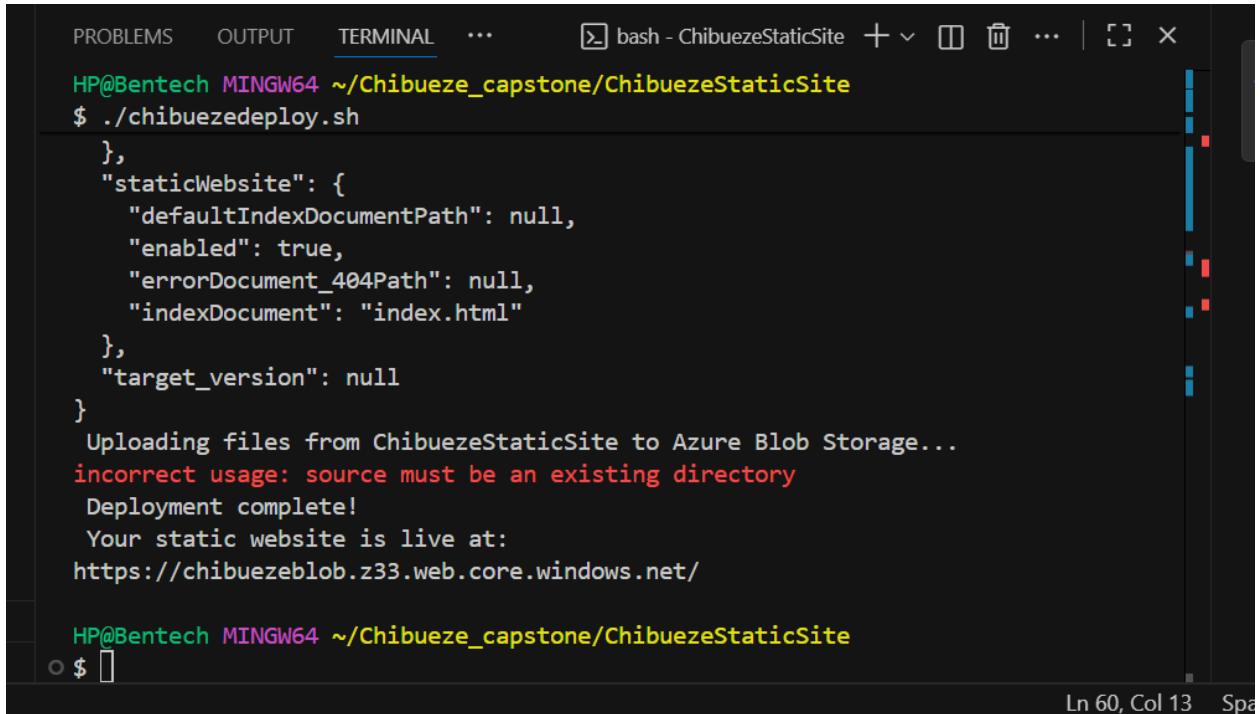
Retrieving tenants and subscriptions for the selection...
The following tenants don't contain accessible subscriptions. Use `az login --all
ow-no-subscriptions` to have tenant level access.
af68fc3b-d106-424b-a2d1-1d4f0558fa6e 'Gao Rfid Inc.'
```

No	Subscription name	Subscription ID	Tenant
[1] *	Azure subscription 1	d651572d-7b61-49bc-a920-94a731fc9b27	Default Directory

The default is marked with an *; the default tenant is 'Default Directory' and su

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 ;

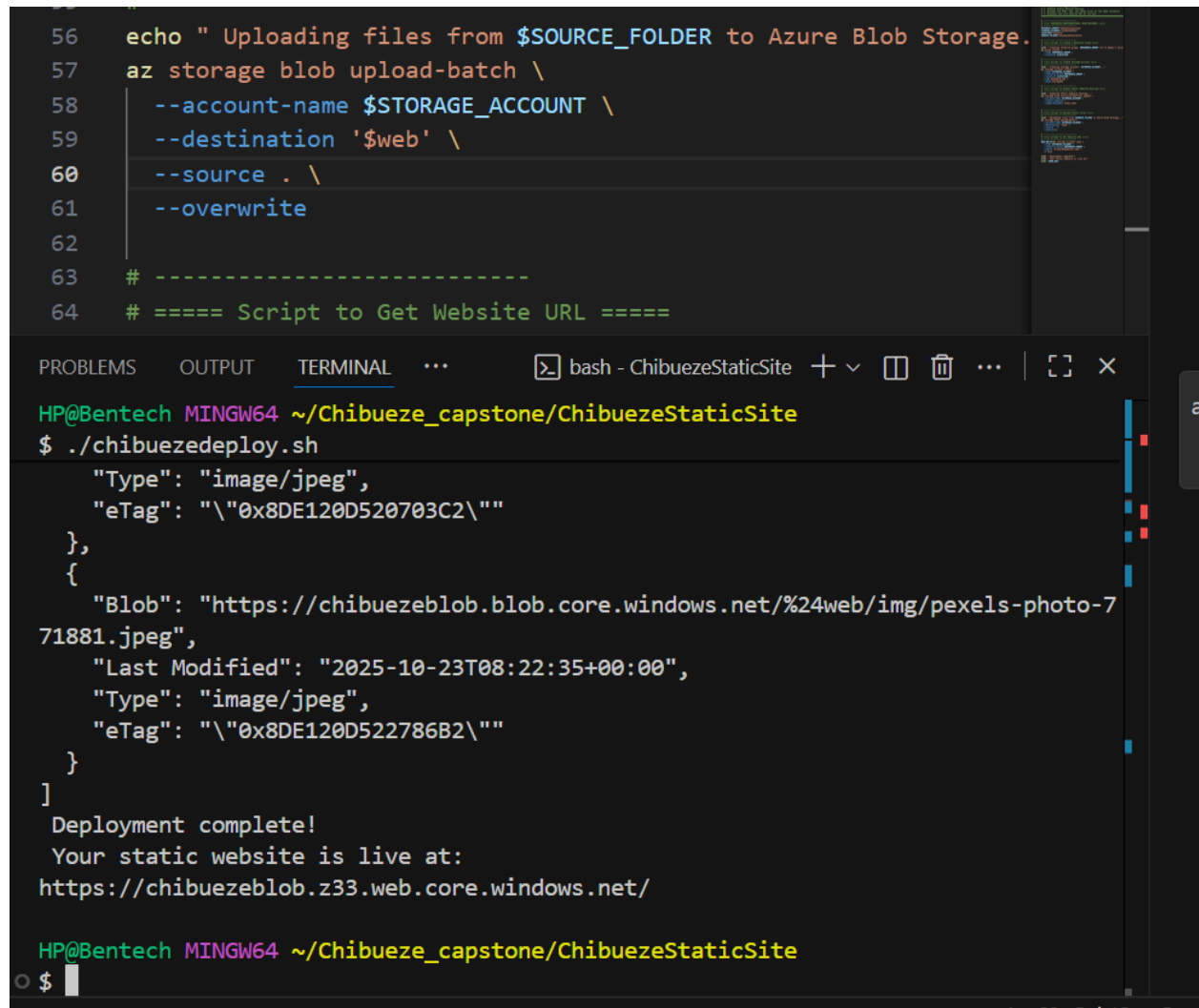


```
HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite
$ ./chibuezedeploy.sh
},
  "staticWebsite": {
    "defaultIndexDocumentPath": null,
    "enabled": true,
    "errorDocument_404Path": null,
    "indexDocument": "index.html"
  },
  "target_version": null
}
Uploading files from ChibuezeStaticSite to Azure Blob Storage...
incorrect usage: source must be an existing directory
Deployment complete!
Your static website is live at:
https://chibuezeblob.z33.web.core.windows.net/

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite
$
```

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:



```
56 echo " Uploading files from $SOURCE_FOLDER to Azure Blob Storage."
57 az storage blob upload-batch \
58   --account-name $STORAGE_ACCOUNT \
59   --destination '$web' \
60   --source . \
61   --overwrite
62
63 # -----
64 # ===== Script to Get Website URL =====
```

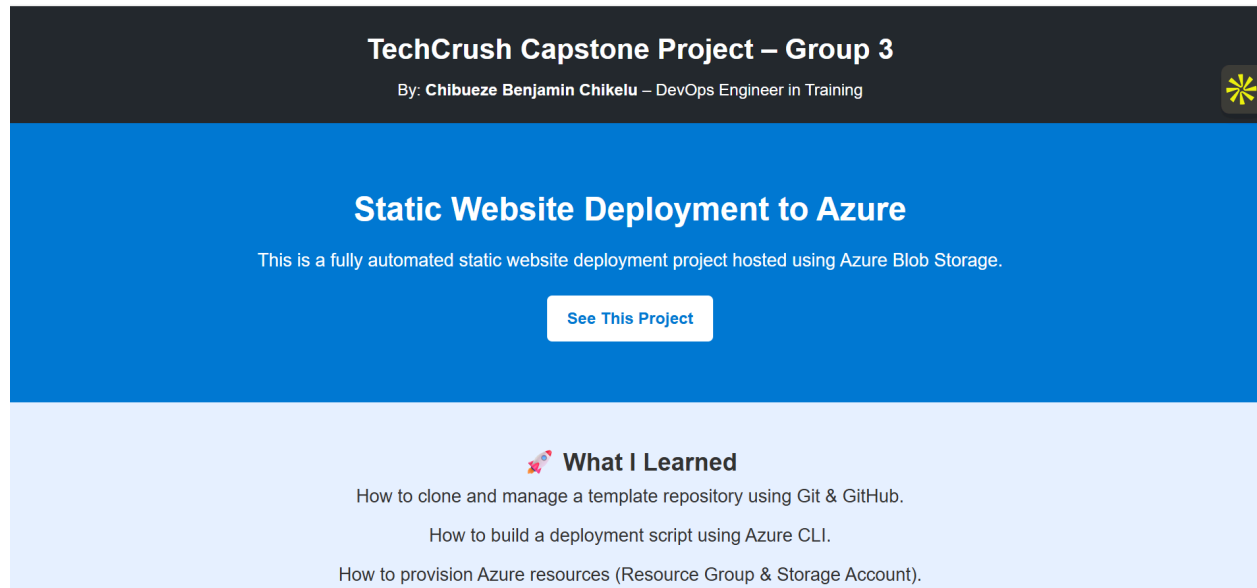
HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite
\$./chibuezedeploy.sh

```
{
  "Type": "image/jpeg",
  "eTag": "\"0x8DE120D520703C2\""
},
{
  "Blob": "https://chibuezeblob.blob.core.windows.net/%24web/img/pexels-photo-771881.jpeg",
  "Last Modified": "2025-10-23T08:22:35+00:00",
  "Type": "image/jpeg",
  "eTag": "\"0x8DE120D522786B2\""
}
]
```

Deployment complete!
Your static website is live at:
<https://chibuezeblob.z33.web.core.windows.net/>

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite
\$

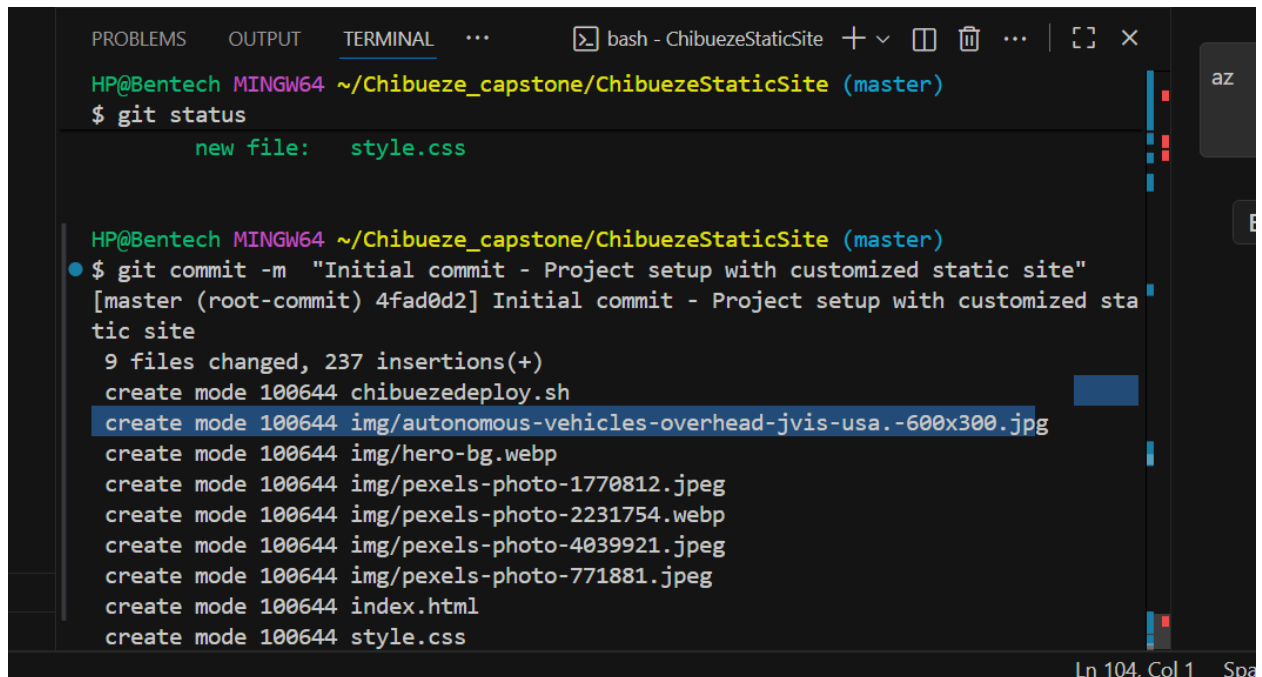
The adding of . in the source made sure the script would run from the current working directory, and yes the script was successfully ran this and my static website URL displayed, my first job was done and I became very happy, below is the screenshot of customized portfolio-like static website after successful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

A screenshot of a Windows terminal window with a dark background. The title bar shows 'bash - ChibuezeStaticSite'. The terminal text shows the user 'HP@Bentech' in a 'MINGW64' shell at the directory '~/Chibueze_capstone/ChibuezeStaticSite' on the 'master' branch. The first command is '\$ git status', which shows 'new file: style.css'. The second command is '\$ git commit -m "Initial commit - Project setup with customized static site"', which shows the commit hash '[master (root-commit) 4fad0d2]' and a list of 9 files created, including 'chibuezedeploy.sh', several image files in the 'img/' directory, 'index.html', and 'style.css'. The status bar at the bottom right indicates 'Ln 104, Col 1' and 'Spa'.

```
PROBLEMS  OUTPUT  TERMINAL  ...  bash - ChibuezeStaticSite  + v  [ ]  [ ]  ...  [ ]  x

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite (master)
$ git status

        new file:   style.css

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite (master)
● $ git commit -m "Initial commit - Project setup with customized static site"
[master (root-commit) 4fad0d2] Initial commit - Project setup with customized static site
 9 files changed, 237 insertions(+)
 create mode 100644 chibuezedeploy.sh
 create mode 100644 img/autonomous-vehicles-overhead-jvis-usa.-600x300.jpg
 create mode 100644 img/hero-bg.webp
 create mode 100644 img/pexels-photo-1770812.jpeg
 create mode 100644 img/pexels-photo-2231754.webp
 create mode 100644 img/pexels-photo-4039921.jpeg
 create mode 100644 img/pexels-photo-771881.jpeg
 create mode 100644 index.html
 create mode 100644 style.css

Ln 104, Col 1  Spa
```

The files were successfully committed, 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.

The screenshot displays the GitHub repository creation interface, divided into two main sections: General and Configuration.

General Section:

- Owner *:** Bentech10 (selected from a dropdown menu).
- Repository name *:** Chibueze-techcrush-group3-capstone (entered in a text box). A green checkmark below the text box indicates "Chibueze-techcrush-group3-capstone is available."
- Description:** A text area with a placeholder "Great repository names are short and memorable. How about **refactored-telegram**?" and a character count "0 / 350 characters".

Configuration Section:

- Choose visibility *:** Public (selected from a dropdown menu). A note below says "Choose who can see and commit to this repository".
- Add README:** A toggle switch is set to "Off". A note below says "READMEs can be used as longer descriptions. [About READMEs](#)".
- Add .gitignore:** A dropdown menu is set to "No .gitignore". A note below says ".gitignore tells git which files not to track. [About ignoring files](#)".
- Add license:** A dropdown menu is set to "No license". A note below says "Licenses explain how others can use your code. [About licenses](#)".

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.

```
9      deploy:
10        steps:
11          - uses: azure/login@v1
12            with:
13              creds: ${ secrets.AZURE_CREDENTIALS }
14
15          - name: Upload files to Azure Blob Storage
16            uses: azure/CLI@v1
17            with:
18              inlineScript: |
19                az storage blob upload-batch --account-name $AZURE_STORAGE_ACCOUNT --account-key $AZURE_STORAGE_KEY --source . --destination $AZURE_STORAGE_CONTAINER
```

PROBLEMS OUTPUT TERMINAL ...

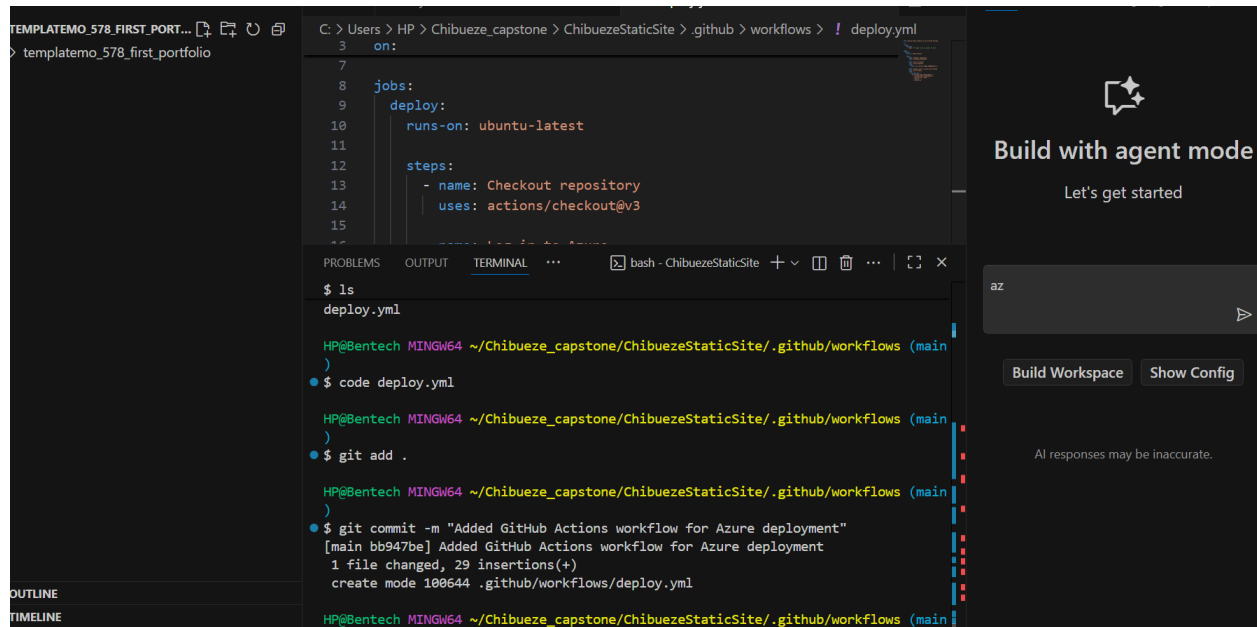
bash - workflows + - [] [] ... [] []

```
$ git add .

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
$ git commit -m "Added GitHub Actions workflow for Azure deployment"
[main bb947be] Added GitHub Actions workflow for Azure deployment
1 file changed, 29 insertions(+)
create mode 100644 .github/workflows/deploy.yml

HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 730 bytes | 730.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
```


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:



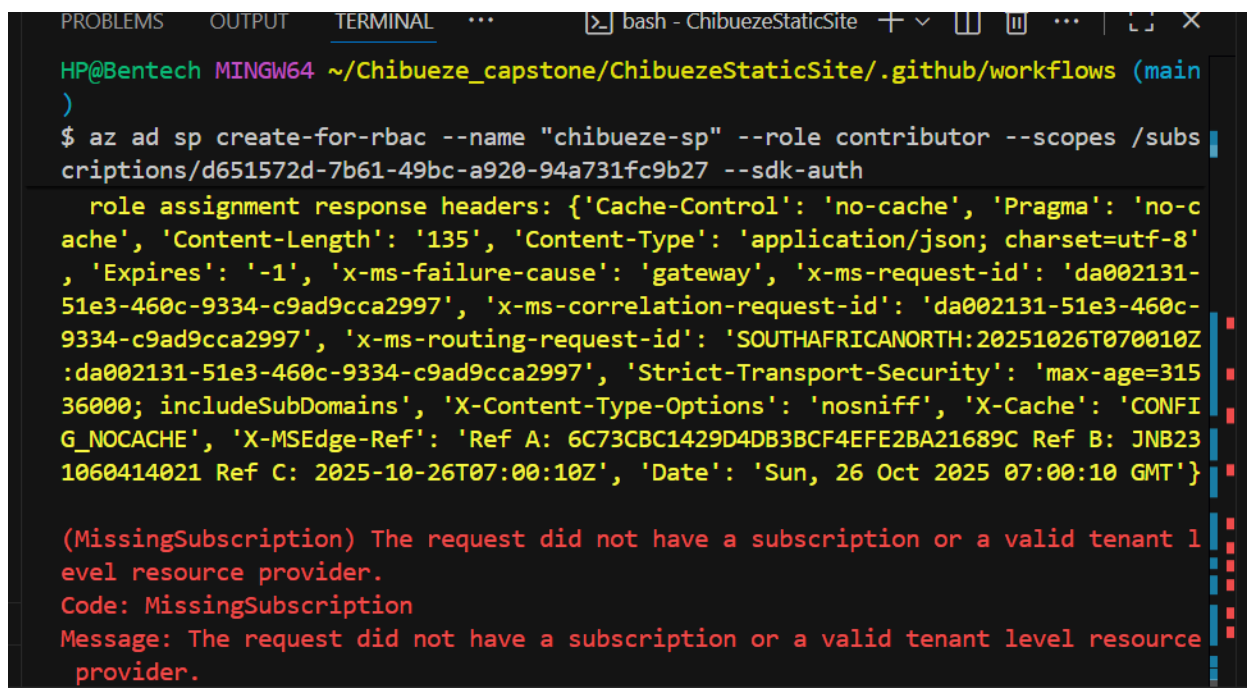
```
3 on:
7
8 jobs:
9   deploy:
10    runs-on: ubuntu-latest
11
12   steps:
13     - name: Checkout repository
14       uses: actions/checkout@v3
15
16 $ ls
17 deploy.yml
18
19 HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
20 • $ code deploy.yml
21
22 HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
23 • $ git add .
24
25 HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
26 • $ git commit -m "Added GitHub Actions workflow for Azure deployment"
27 [main bb947be] Added GitHub Actions workflow for Azure deployment
28 1 file changed, 29 insertions(+)
29 create mode 100644 .github/workflows/deploy.yml
30
31 HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
```

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 ;



```
HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite/.github/workflows (main)
$ az ad sp create-for-rbac --name "chibueze-sp" --role contributor --scopes /subscriptions/d651572d-7b61-49bc-a920-94a731fc9b27 --sdk-auth
role assignment response headers: {'Cache-Control': 'no-cache', 'Pragma': 'no-cache', 'Content-Length': '135', 'Content-Type': 'application/json; charset=utf-8', 'Expires': '-1', 'x-ms-failure-cause': 'gateway', 'x-ms-request-id': 'da002131-51e3-460c-9334-c9ad9cca2997', 'x-ms-correlation-request-id': 'da002131-51e3-460c-9334-c9ad9cca2997', 'x-ms-routing-request-id': 'SOUTHAFRICANORTH:20251026T070010Z:da002131-51e3-460c-9334-c9ad9cca2997', 'Strict-Transport-Security': 'max-age=31536000; includeSubDomains', 'X-Content-Type-Options': 'nosniff', 'X-Cache': 'CONFIG_NOCACHE', 'X-MSEdge-Ref': 'Ref A: 6C73CBC1429D4DB3BCF4EFE2BA21689C Ref B: JNB231060414021 Ref C: 2025-10-26T07:00:10Z', 'Date': 'Sun, 26 Oct 2025 07:00:10 GMT'}

(MissingSubscription) The request did not have a subscription or a valid tenant level resource provider.
Code: MissingSubscription
Message: The request did not have a subscription or a valid tenant level resource provider.
```

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**.

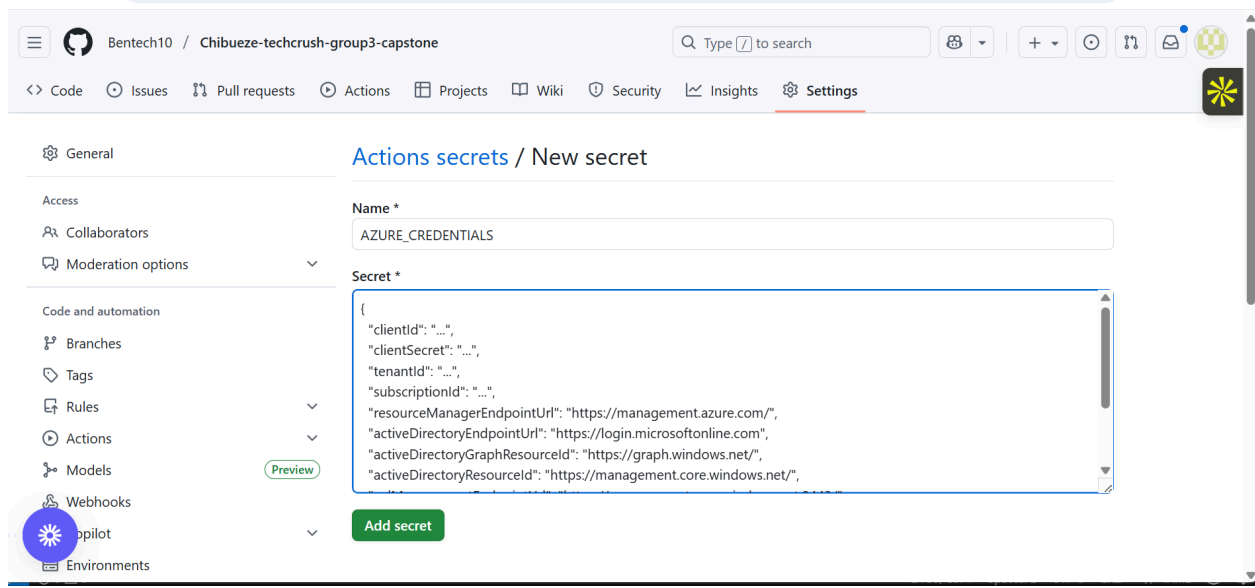
It still returned same error and after going back and forth for a long time and making so many researches on how to resolve the error, I finally found a breakthrough when it was suggested I run the command using powershell rather than gitbash as gitbash might be misinterpreting the command. I immediately logged to my powershell using **az login --use-device-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;

```

Azure subscription 1 AzureCloud d651572d-7b61-49bc-a920-94a731fc9b27 481d914d-48e4-4ba7-8034-788cfce70936 Enabled True
PS C:\Users\HP> az account set --subscription d651572d-7b61-49bc-a920-94a731fc9b27
PS C:\Users\HP> az account show -o table
EnvironmentName HomeTenantId IsDefault Name State TenantDefaultDomain
-----
AzureCloud 481d914d-48e4-4ba7-8034-788cfce70936 True Azure subscription 1 Enabled benchikelu12gmail.onmicrosoft.co
Default Directory 481d914d-48e4-4ba7-8034-788cfce70936
PS C:\Users\HP> az role assignment create --assignee 9ff9fc69-a5fa-4734-a83d-01080a68d66f --role contributor --scope /subscriptions/d
651572d-7b61-49bc-a920-94a731fc9b27
{
  "condition": null,
  "conditionVersion": null,
  "createdBy": null,
  "createdOn": "2025-10-26T11:00:28.539950+00:00",
  "delegatedManagedIdentityResourceId": null,
  "description": null,
  "id": "/subscriptions/d651572d-7b61-49bc-a920-94a731fc9b27/providers/Microsoft.Authorization/roleAssignments/10ba1507-6d72-46cd-817
5-3226cb8267ac",
  "name": "10ba1507-6d72-46cd-8175-3226cb8267ac",
  "principalId": "c3ea7a43-ef70-4023-a657-5766c1dc9498",
  "principalType": "ServicePrincipal",
  "roleDefinitionId": "/subscriptions/d651572d-7b61-49bc-a920-94a731fc9b27/providers/Microsoft.Authorization/roleDefinitions/b24988ac
6180-42a0-ab88-20f7382dd24c",
  "scope": "/subscriptions/d651572d-7b61-49bc-a920-94a731fc9b27",
  "type": "Microsoft.Authorization/roleAssignments",
  "updatedBy": "471d0f90-4013-4b16-b9bd-4ee0f01cd3b7",
  "updatedOn": "2025-10-26T11:00:28.904954+00:00"
}
PS C:\Users\HP> az ad sp create-for-rbac --name "chibuzo-so2" --role contributor --scopes /subscriptions/d651572d-7b61-49bc-a920-94a

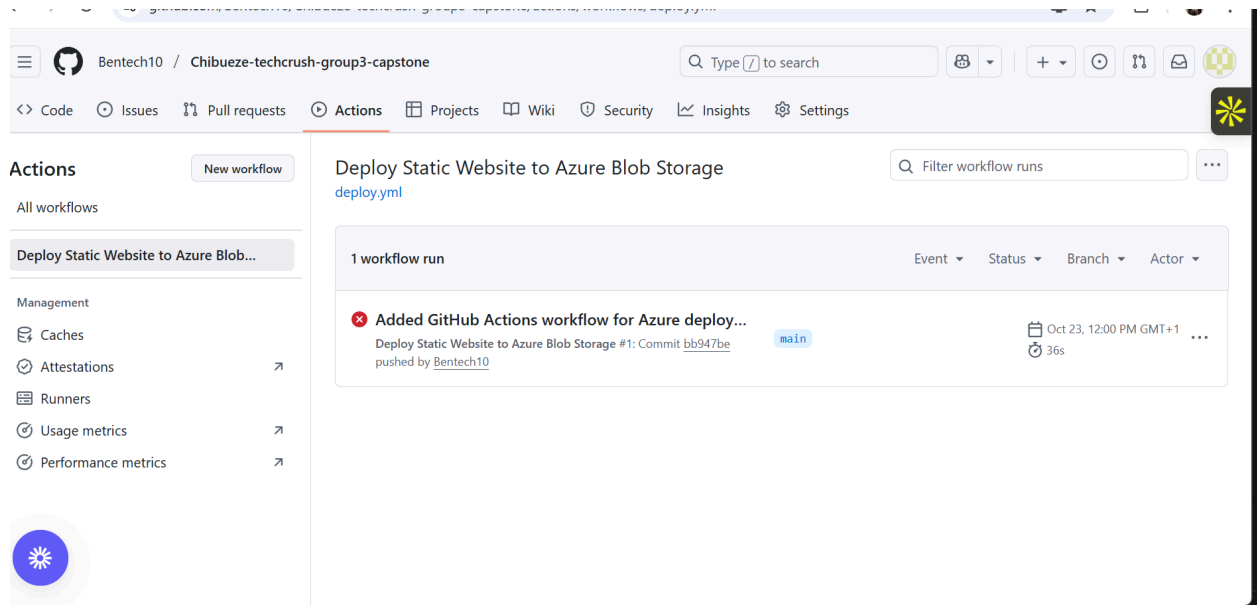
```

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;



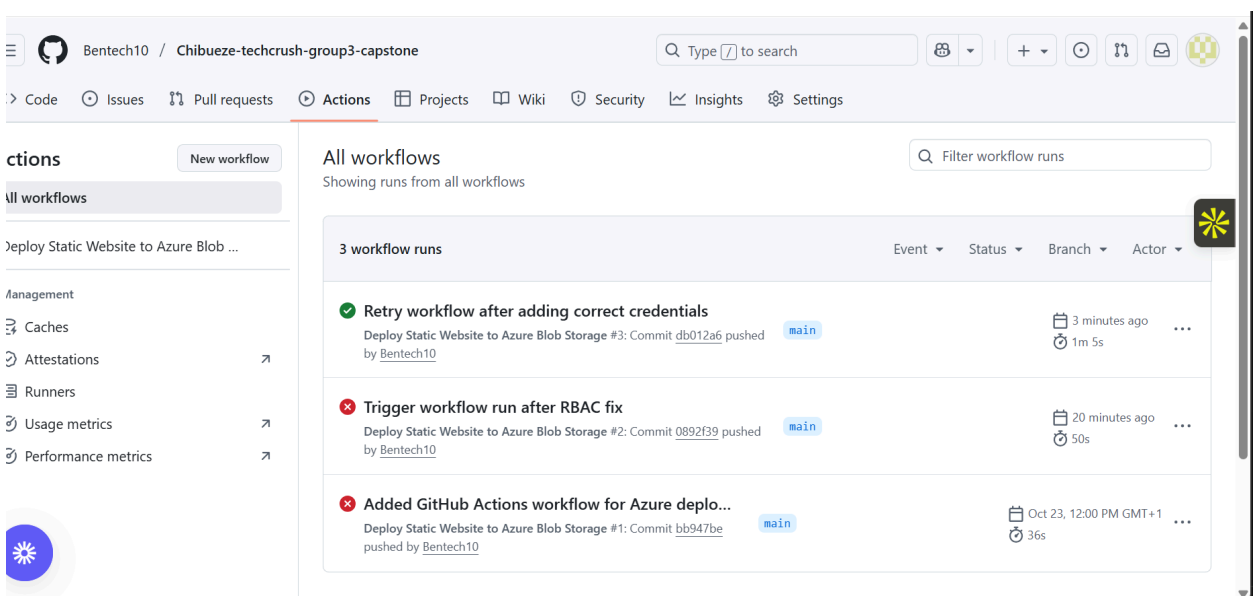
I tried again and got another failure, so I researched to know why, I finally tried out this command in the screenshot below;

```
PROBLEMS OUTPUT TERMINAL ... bash - ChibuezeStaticSite + v [ ] [ ] ... [ ] x

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 next 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

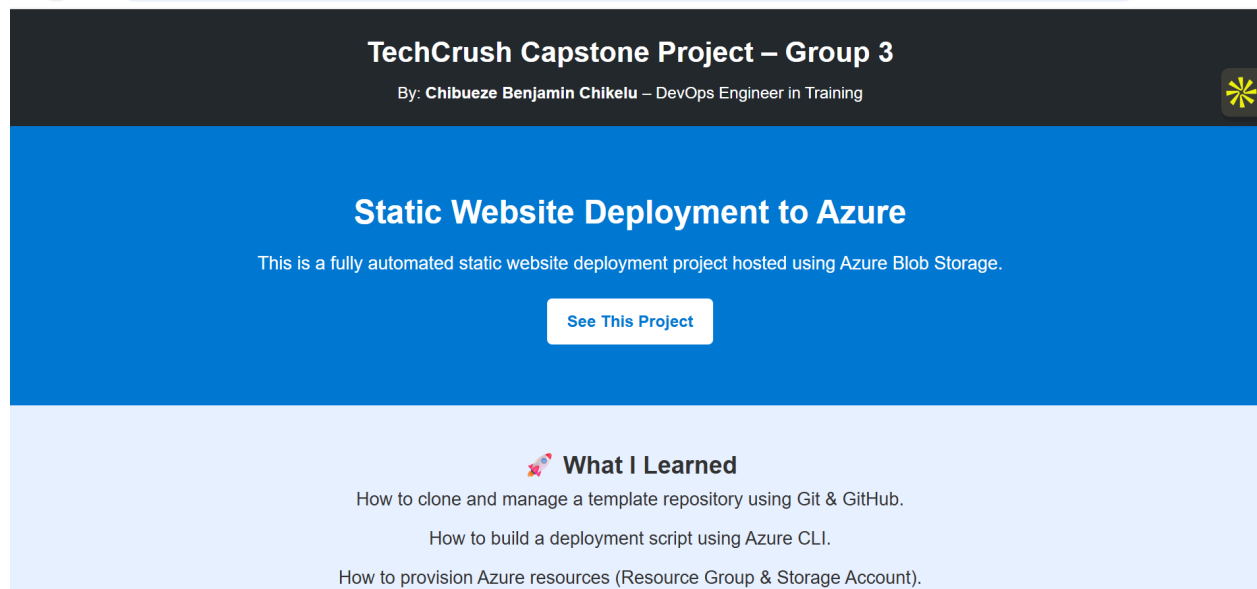
HP@Bentech MINGW64 ~/Chibueze_capstone/ChibuezeStaticSite (main)
$
```

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

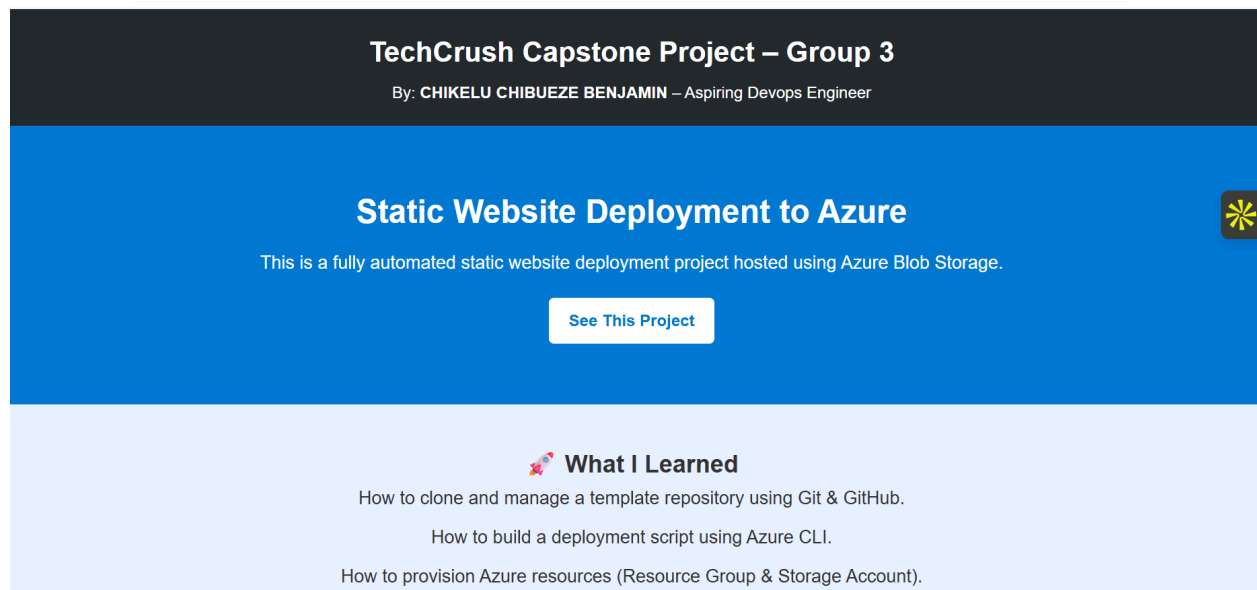


And yes, the deployment worked and my site is still running as well.

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;

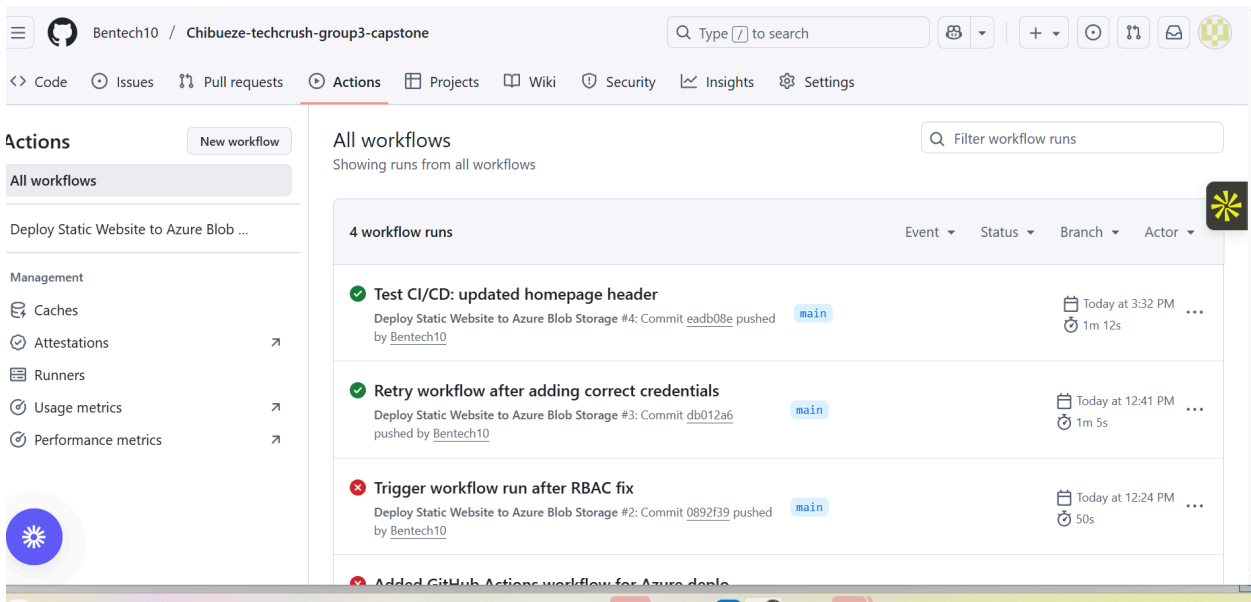


Original static website View!



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 <code>--source ..</code>
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.