Group 3 Assignment

Static Website Deployment to Azure via CLI + GitHub Action

Objective:

Deploy a static website to Azure Blob Storage using a bash Script.

Tasks:

1. using a static website template from anywhere

2. Use Azure CLI to provision a storage account and enable static website hosting

3. Uploads files with CLI.

Bonus: Automate deployment with GitHub Actions and ensure uploads happens with new site files on push.

Deliverables: Static website code, CLI provisioning script, GitHub Actions workflow (if it applies) screenshots of live site.

Note: Ensure that static website used is unique.

Solution

Section A

Static website provisioning on Azure Storage

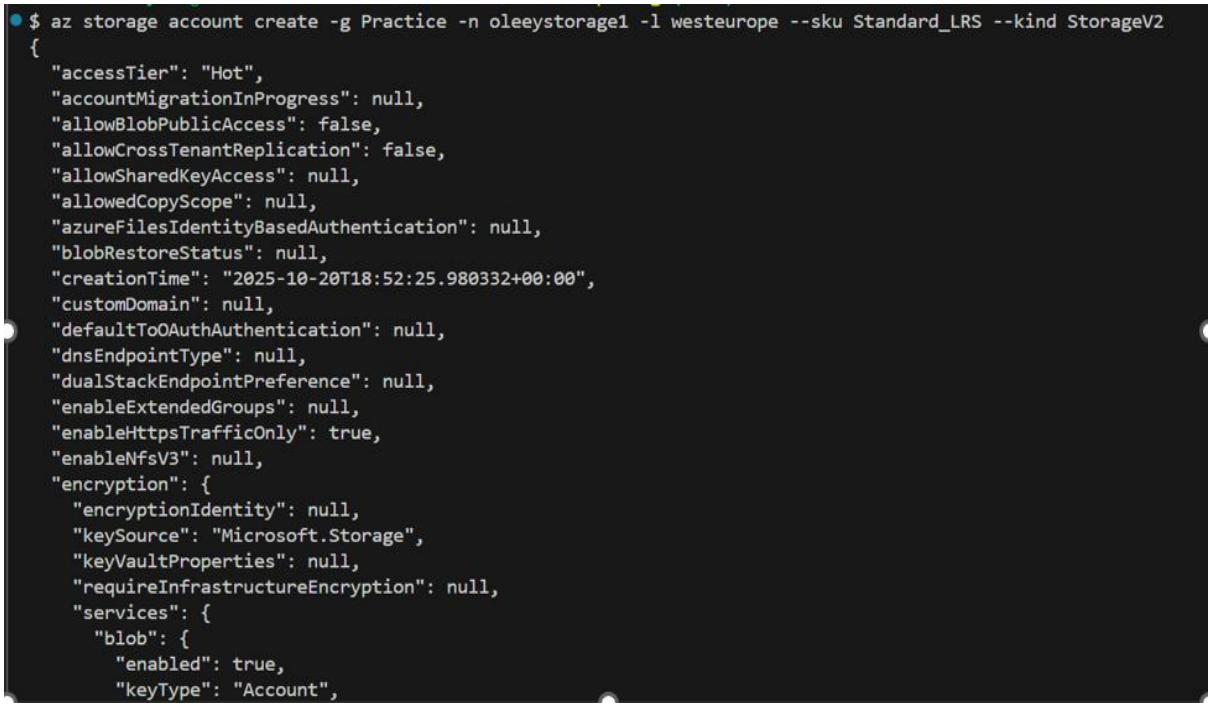
1. Login to Azure using the CLI

```
$ az login
Select the account you want to log in with.
```

2. Note the azure account has existing resource groups.

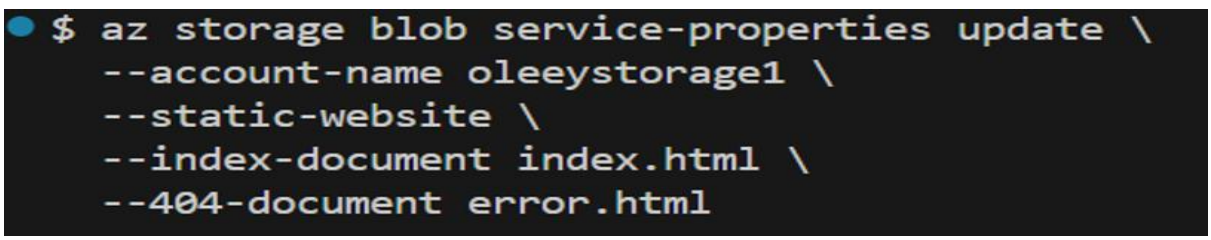
```
$ az group list -o table
Name              Location    Status
----------------  ----------  ---------
Practice          westeurope  Succeeded
NetworkWatcherRG  westeurope  Succeeded
```

3. Provisioning of a storage account called oleeystorage1.



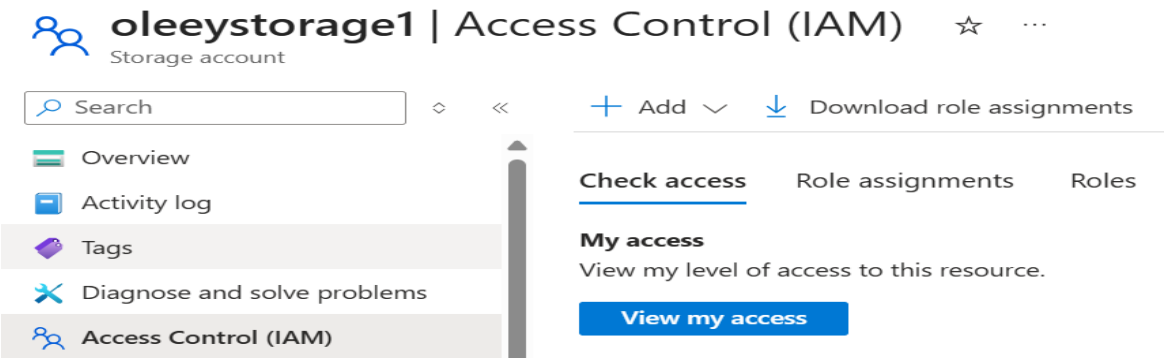
```
$ az storage account create -g Practice -n oleeystorage1 -l westeurope --sku Standard_LRS --kind StorageV2
{
  "accessTier": "Hot",
  "accountMigrationInProgress": null,
  "allowBlobPublicAccess": false,
  "allowCrossTenantReplication": false,
  "allowSharedKeyAccess": null,
  "allowedCopyScope": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2025-10-20T18:52:25.980332+00:00",
  "customDomain": null,
  "defaultToOAuthAuthentication": null,
  "dnsEndpointType": null,
  "dualStackEndpointPreference": null,
  "enableExtendedGroups": null,
  "enableHttpsTrafficOnly": true,
  "enableNfsV3": null,
  "encryption": {
    "encryptionIdentity": null,
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
        "enabled": true,
        "keyType": "Account",
    "file": "https://oleeystorage1.file.core.windows.net/",
    "internetEndpoints": null,
    "ipv6Endpoints": null,
    "microsoftEndpoints": null,
    "queue": "https://oleeystorage1.queue.core.windows.net/",
    "table": "https://oleeystorage1.table.core.windows.net/",
    "web": "https://oleeystorage1.z6.web.core.windows.net/"
  },
  "primaryLocation": "westeurope",
  "privateEndpointConnections": [],
  "provisioningState": "Succeeded",
  "publicNetworkAccess": null,
  "resourceGroup": "Practice",
  "routingPreference": null,
  "sasPolicy": null,
  "secondaryEndpoints": null,
  "secondaryLocation": null,
  "sku": {
    "name": "Standard_LRS",
    "tier": "Standard"
  },
  "statusOfPrimary": "available",
  "statusOfSecondary": null,
  "storageAccountSkuConversionStatus": null,
  "tags": {},
  "type": "Microsoft.Storage/storageAccounts",
  "zones": null
}
```

4. Enabling a static website



```
$ az storage blob service-properties update \
    --account-name oleeystorage1 \
    --static-website \
    --index-document index.html \
    --404-document error.html
```

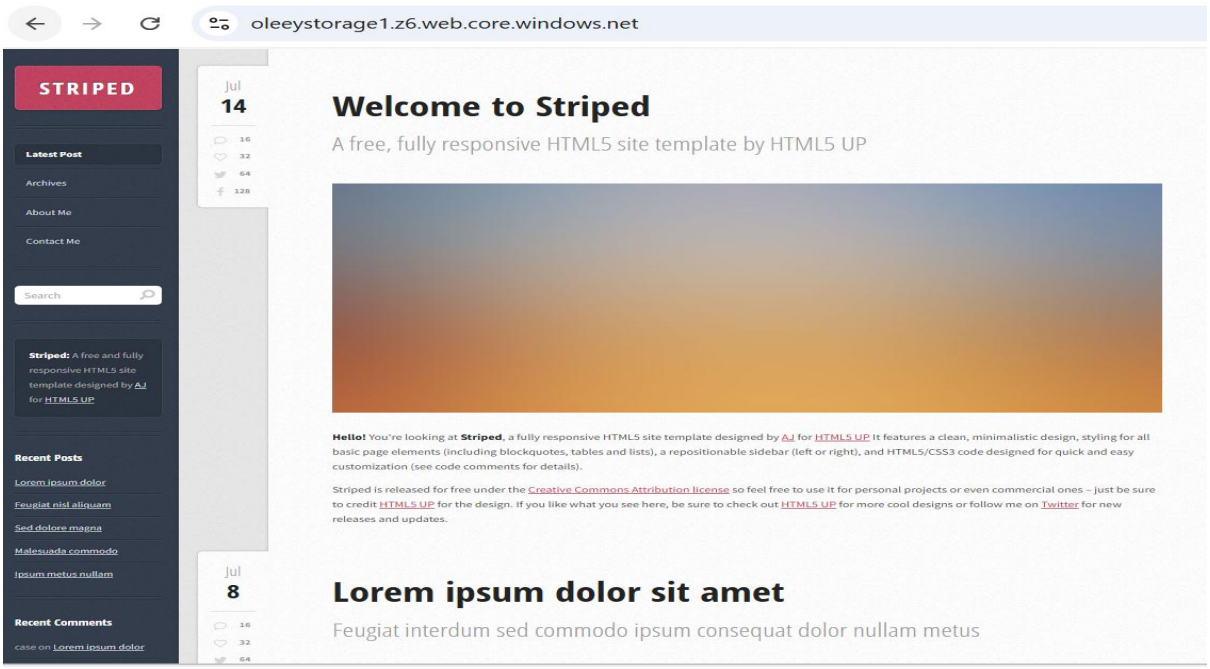
5. Upload of a static html website folder to azure storage account: A storage blob data contributor role was given to the azure account to enable upload. This was done using the IAM in the Azure GUI.




```
$ az storage blob upload-batch \
  --account-name oleeystorage1 \
  --source ./WebsiteTemp \
  --destination '$web' \
  --auth-mode login
Finished[#############################################################]  100.0000%
[
  {
    "Blob": "https://oleeystorage1.blob.core.windows.net/%24web/index.html",
    "Last Modified": "2025-10-25T22:31:11+00:00",
    "Type": "text/html",
    "eTag": "\"0x8DE1416331ABC48\""
  },
  {
    "Blob": "https://oleeystorage1.blob.core.windows.net/%24web/LICENSE.txt",
    "Last Modified": "2025-10-25T22:31:11+00:00",
    "Type": "text/plain",
    "eTag": "\"0x8DE1416333A74A2\""
  },
```

The website url link: Click Here

```
$ az storage account show \
  --name oleeystorage1 \
  --query "primaryEndpoints.web" \
  -o tsv
https://oleeystorage1.z6.web.core.windows.net/
```



Section B

Github Actions Automation

1. Creation of the “workflows” folder in the “.github” folder

```
$ mkdir -p .github/workflows
```

2. Creation of the yml file

```
$ touch frontend_site.yml
```

The yml file code:

```
name: Deploy_Static_Website

on:
  push:
    branches: [ main ]
    paths:
      - 'WebsiteTemp/**'
  workflow_dispatch:

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
```

```
    uses: actions/checkout@v4

  - name: Azure Login
    uses: azure/login@v2
    with:
      creds: ${{ secrets.AZURE_CREDENTIAL }}

  - name: Upload to Azure Storage
    uses: azure/CLI@v2
    with:
      inlineScript: |
        az storage blob upload-batch \
          --account-name oleeystorage1 \
          --account-key ${{ secrets.AZURE_STORAGE_KEY }} \
          --destination '$web' \
          --source ./WebsiteTemp \
          --overwrite true

  - name: Logout
    run: az logout
    if: always()
```

3. The folders and file were committed and push to github using the code below:
   i. git add .
   ii. git commit -m “committing all changes”
   iii. git push origin main
4. The workflow run, “Deploy_Static_Website” created in the yml file is showing under “Actions” in github after pushing to the github repository.

Section C

1. Creation of Azure Service Principal. This is to enable the github actions login to azure.

```
az ad sp create-for-rbac \
  --name "github-actions-deploy" \
  --role contributor \
  --scopes /subscriptions/$(az account show --query id -o
tsv)/resourceGroups/Practice \
  --sdk-auth
```

```
$ az ad sp create-for-rbac \
  --name "github-actions-deploy" \
  --role contributor \
  --scopes /subscriptions/bac445ec-3a7a-41f6-a705-3f2fff3e0e12/reso
  --sdk-auth
Option '--sdk-auth' has been deprecated and will be removed in a fu
Found an existing application instance: (id) c390176b-ef0a-472c-a5f
Creating 'contributor' role assignment under scope 'C:/Users/adedam
5ec-3a7a-41f6-a705-3f2fff3e0e12/resourceGroups/Practice'
  Role assignment creation failed.
```

The role assignment failed because the application instance already exists. A "storage blob data contributor" role was assigned to the application using the "IAM" in the GUI. A new client secret key was also generated using the GUI. The information was put in a json format as seen below.

{
  "clientId": "c571bbfa-xxxx",
  "clientSecret": "Dbxxxxx",
  "subscriptionId": "bac445ec-3a7a-41f6-a705-3f2fff3e0e12",
  "tenantId": "a2253388-ec26-40fe-8727-270b60c01cbc"
}

The information provided above was stored in the github secrets and variables as "AZURE_CREDENTIAL" while the storage account key was saved as "AZURE_STORAGE_KEY". Both secrets were referenced as variables in the yml file above. Below is the code to get the storage account key:

az storage account keys list -g Practice -n oleeystorage1.

The workflow run was tested manually after setting up the "AZURE_CREDENTIAL" and "AZURE_STORAGE_KEY". Also, the html file was modified, and the change was pushed to the github repository which triggers the workflow run as seen below.