Dr. N.G.P INSTITUTE OF TECHNOLOGY, COIMBATORE - 641048

AN AUTONOMOUS INSTITUTION

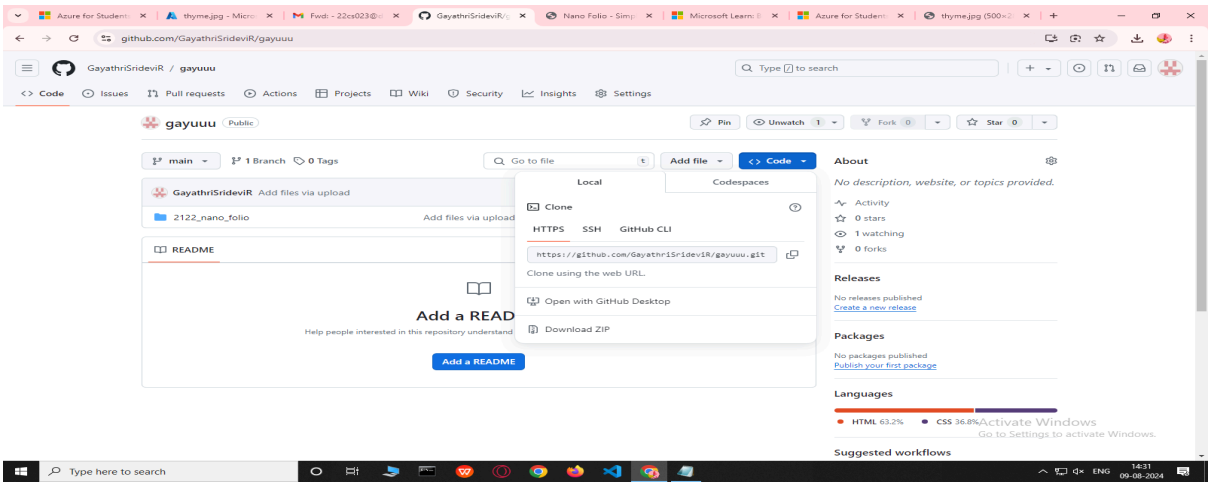
Reg. No.	**:**	710722104023
Name	**:**	GAYATHRI SRIDEVI R
Class	**:**	II Year CSE A
Course Name	**:**	Microsoft azure Fundamentals
Company	**:**	Pinesphere Solution, Coimbatore.
Start Date	**:**	06.08.2024
End Date	**:**	10.08.2024

CREATING A VIRTUAL MACHINE (VM) IN MICROSOFT AZURE:

Creating A Virtual Machine (Vm) In Microsoft Azure Involves The Following Steps:

1. Sign in to the Azure portal.
2. Navigate to "Create a resource" and select "Virtual Machine."
3. Choose a subscription, resource group, and region.
4. Configure VM settings, including size, OS, and storage.
5. Set up networking, security, and management options.
6. Review and create the VM, then monitor its deployment.

The VM will be ready to use after deployment.



HOST A WEBSITE FROM GITHUB ON A VIRTUAL MACHINE (VM) IN MICROSOFT AZURE

1. Set Up the VM: Ensure your Azure VM is running and accessible via SSH or RDP. Install a web server like Apache or Nginx on the VM.

2. Clone the GitHub Repository: SSH into the VM and clone your website's repository from GitHub using git clone <repository-url>.

3. Deploy the Website: Move the cloned repository to the web server's root directory, typically /var/www/html for Apache or the appropriate directory for Nginx.

4. Configure the Web Server: Update the web server configuration files to serve your website. Restart the server to apply changes.

5. Open Ports: Ensure that the necessary ports (e.g., port 80 for HTTP) are open in the Azure network security group settings to allow web traffic.

6. Access the Website: Access your website by entering the VM's public IP address or domain name in a web browser.

COMMANDS:

Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
gayathri [~]$ ssh gayathri@4.186.16.42
gayathri@4.186.16.42's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1010-azure x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Fri Aug 9 06:13:31 UTC 2024

System load: 0.05 Processes: 140
Usage of /: 5.8% of 28.02GB Users logged in: 1
Memory usage: 4% IPv4 address for eth0: 10.0.0.5
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

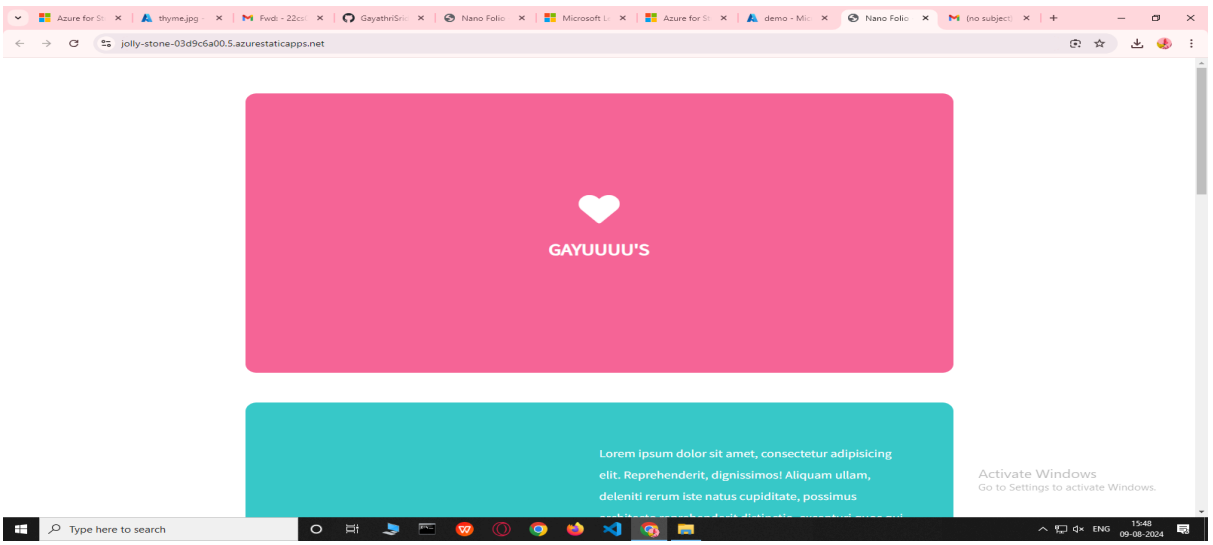
13 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug 9 05:57:52 2024 from 4.186.11.194
gayathri@demo:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
gayathri@demo:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
gayathri@demo:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7).
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
gayathri@demo:~$ sudo systemctl start nginx
gayathri@demo:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with
/usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
gayathri@demo:~$ cd /var/www/html

```
gayathri@demo:/var/www/html$ sudo rm -rf *
gayathri@demo:/var/www/html$ ls
gayathri@demo:/var/www/html$  sudo git clone ^C
fatal: repository '^C' does not exist
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset:
enabled)
     Active: active (running) since Fri 2024-08-09 05:01:12 UTC; 1h 25min ago
       Docs: man:nginx(8)
   Main PID: 2383 (nginx)
      Tasks: 3 (limit: 9459)
     Memory: 2.5M (peak: 3.0M)
        CPU: 23ms
     CGroup: /system.slice/nginx.service
             ├─2383 "nginx: master process /usr/sbin/nginx -g daemon on;
master_process on;"
             ├─2384 "nginx: worker process"
             └─2385 "nginx: worker process"

Aug 09 05:01:12 demo systemd[1]: Starting nginx.service - A high performance
web server and a reverse proxy server.Aug 09 05:01:12 demo systemd[1]:
Started nginx.service - A high performance web server and a reverse proxy
server.
```



CREATION OF STORAGE ACCOUNT IN MICROSOFT:

To Create A Storage Account In Microsoft Azure, Follow These Steps:

1. Sign in to Azure Portal: Log in to the Azure portal at https://portal.azure.com.

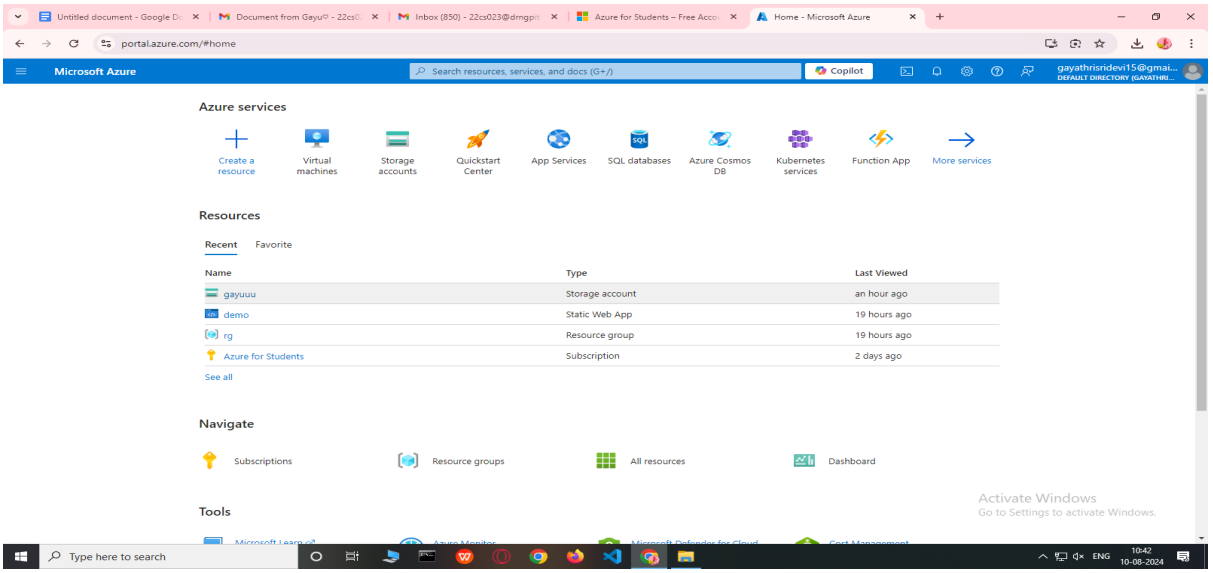
2. Create a Resource: Click on "Create a resource" and select "Storage account" under the "Storage" category.

3. Configure the Basics: Choose a subscription, resource group, and storage account name. Select the region, performance tier (Standard or Premium), and replication option (e.g., LRS, GRS).

4. Set Advanced Options: Configure additional settings like access tier (Hot or Cool), security options, and networking.

5. Review and Create: Review the configuration and click "Create" to deploy the storage account.

6. Access the Storage Account: After deployment, access the storage account to manage containers, blobs, files, tables, or queues.



MANAGING OF STORAGE ACCOUNT

To Upload An Image Into A Container In An Azure Storage Account, Follow These Steps:

a. Access the Storage Account: Sign in to the Azure portal and navigate to your Storage Account.

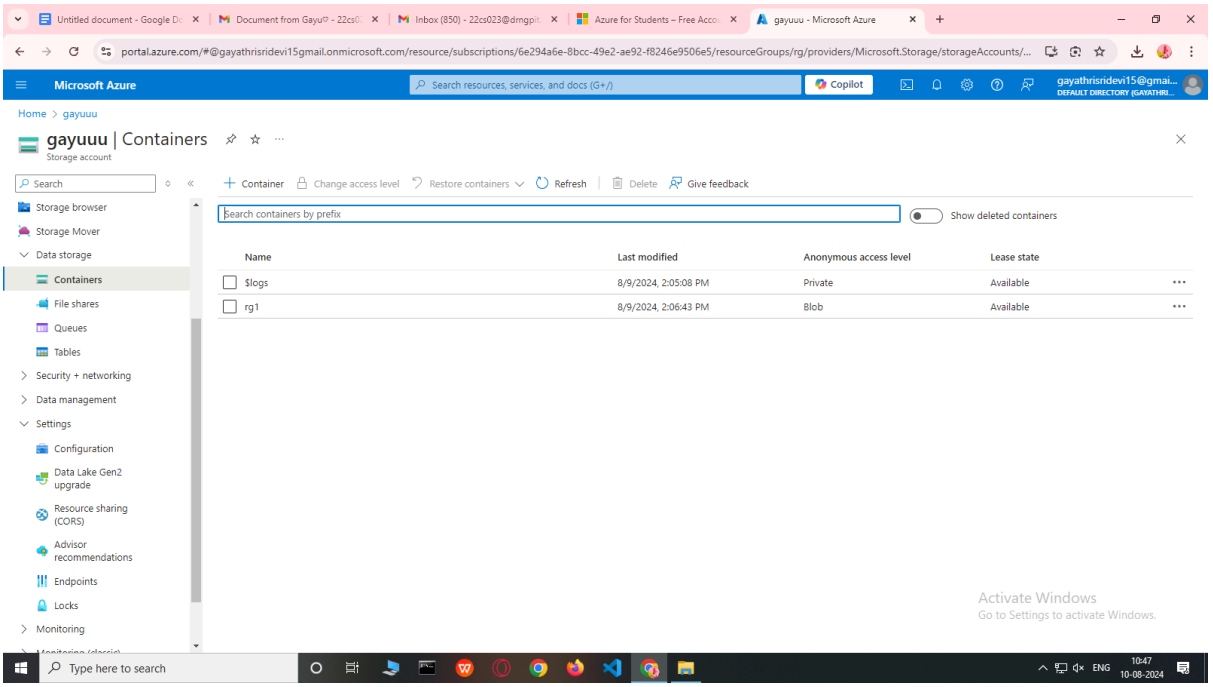
b. Create a Container: In the Storage Account, select "Containers" and click "Add Container." Name the container and set the access level (private, blob, or container).

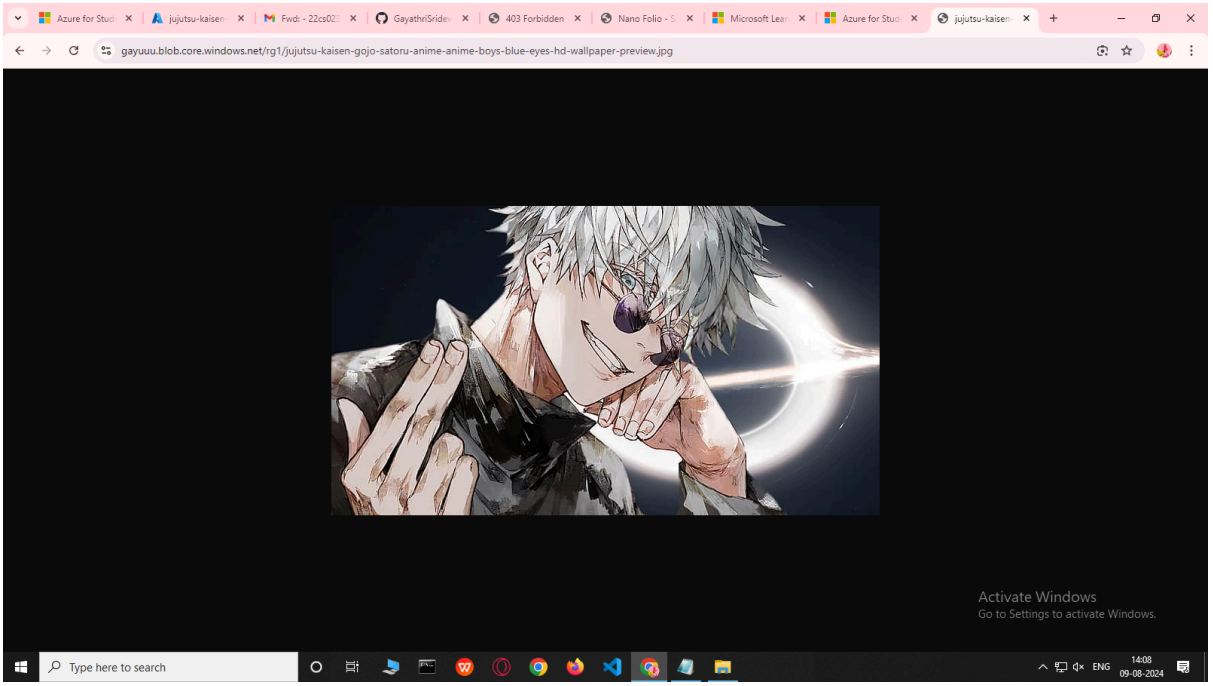
c. Open the Container: Once created, click on the container to open it.

d. Upload the Image: Click the "Upload" button within the container. In the upload window, browse your local machine to select the image file.

e. Configure Upload Settings: Optional - You can set advanced upload options like overwriting existing files, setting metadata, or assigning blob tier.

f. Start the Upload: Click "Upload" to start the process. Once the upload is complete, your image will be stored in the container and accessible based on the access level you set.





URL : https://gayuuu.blob.core.windows.net/rg1/thyme.jpg

STATIC WEB PAGE :

Deploying a Static Web Page on Azure

Using Azure Static Web App:

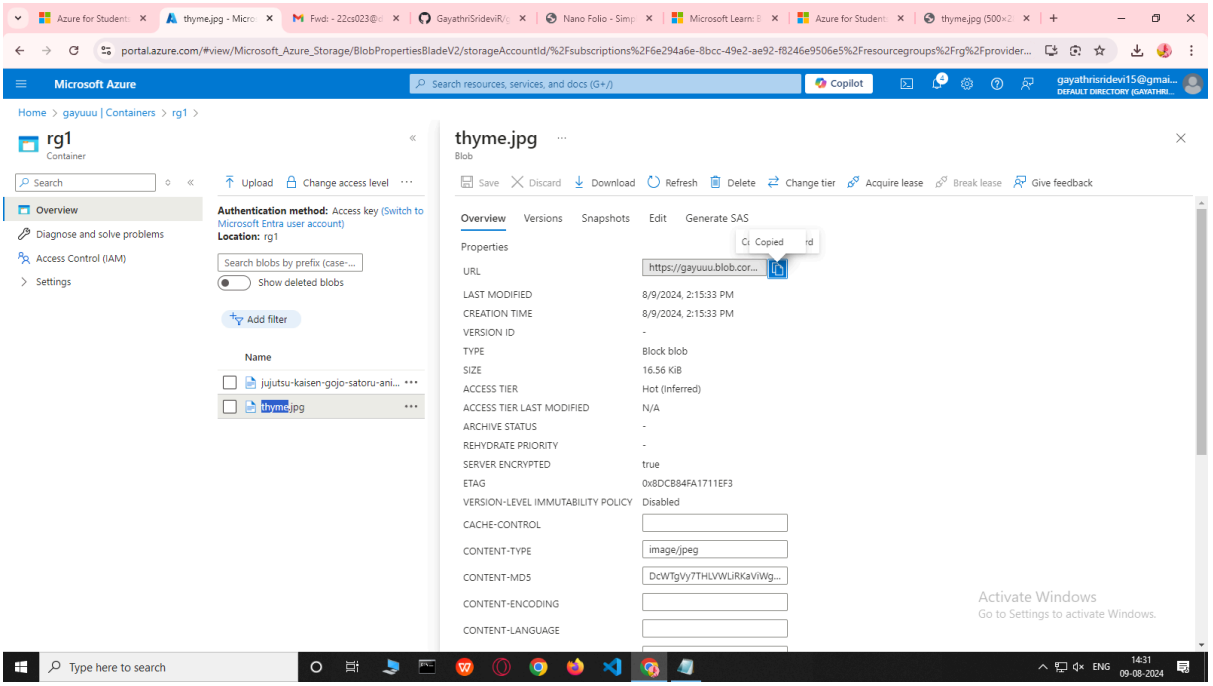
Prepare Your Site: Develop your static site and push it to a GitHub repository.

Set Up Azure Static Web Apps:

1. Sign in to Azure Portal.
2. Click **Create a resource** > **Static Web Apps**.
3. Connect to your GitHub repo and branch.

Deploy and Access:

1. Azure deploys your site automatically.
2. Access it via the provided URL.



URL:https://jolly-stone-03d9c6a00.5.azurestaticapps.net/

OUTPUT: