Azure capstone Project - 1

Capstone Project Submission Report

Project Title: Implementing an Azure Architecture for a Company's Website

Objective:

The objective of this project is to implement an architecture for the company's website using Azure services. The architecture includes deploying web pages, configuring Application Gateways, Traffic Manager, VNets, and Storage Accounts, ensuring optimal traffic distribution between Central US and West US regions.

Architecture Overview:

- **Azure Traffic Manager**
- **Regions:**
  - Central US
  - West US
- **Application Gateways:**
  - Application Gateway 1 (Central US)
  - Application Gateway 2 (West US)
- **Virtual Networks:**
  - VNet 1 (Central US)
  - VNet 2 (West US)
- **Virtual Machines:**
  - VM1-CentralUs - 64.236.113.85 (Upload Page)
  - VM2-CentralUs - 128.203.154.154 (Home Page)
  - VM1-WestUs - 13.91.51.169 (Upload Page)
  - VM2-WestUs - 40.86.182.121 (Home Page)
- **Storage Account:**
  - Static website hosting error.html
  - Container named "upload" for file uploads

Steps to Implement the Architecture:

Step 1: Create Virtual Networks and Subnets

1. **Navigate to Azure Portal**:
   - Go to the Azure portal.
   - Create two virtual networks (VNet1 in Central US and VNet2 in West US).
2. **Configure Address Spaces**:
   - Ensure non-overlapping address spaces.
   - Example:
     - VNet1: 10.0.0.0/16
     - VNet2: 10.1.0.0/16

3. **Create Subnets**:
    - Create subnets in each virtual network.

Step 2: Create Virtual Machines

1. **Create VM1 and VM2 in Each Region**:
    - Deploy two virtual machines in each region (Central US and West US).
    - Assign each VM to the respective VNet and subnet.

Step 3: Set Up Storage Account

1. **Create a Storage Account**:
    - Create a storage account in the desired region.
    - Configure it for static website hosting.
2. **Upload error.html**:
    - Upload the error.html file to the storage account for 403 and 502 errors.
3. **Create a Container**:
    - Create a container named "upload" for file uploads.

Step 4: Clone GitHub Repository and Run Scripts

1. **Clone the Repository**:
    - SSH into each VM and clone the GitHub repository:
    - bash
    - git clone https://github.com/azcloudberg/azproject
2. **Run vm1.sh on VM1**:
    - Navigate to the GitHub directory and run the script:
    - bash
    - . /vm1.sh
3. **Run vm2.sh on VM2**:
    - Navigate to the GitHub directory and run the script:
    - bash
    - . /vm2.sh
4. **Configure Storage in config.py**:
    - Edit the config.py file on VM1 with storage account details.
5. **Start the Application**:
    - Run the application on VM1:
    - bash
    - sudo python3 app.py

Step 5: Configure Application Gateways

1. **Set Up Application Gateways**:
    - Create Application Gateway 1 in Central US and Application Gateway 2 in West US.
    - Configure routing rules:
        - http://azure1044.trafficmanager.net -> Home Page (VM2)
        - http://azure1044.trafficmanager.net/upload -> Upload Page (VM1)
2. **Configure Error Pages**:

- Point Application Gateway's 403 and 502 errors to error.html hosted in the storage account.

Step 6: Configure Traffic Manager

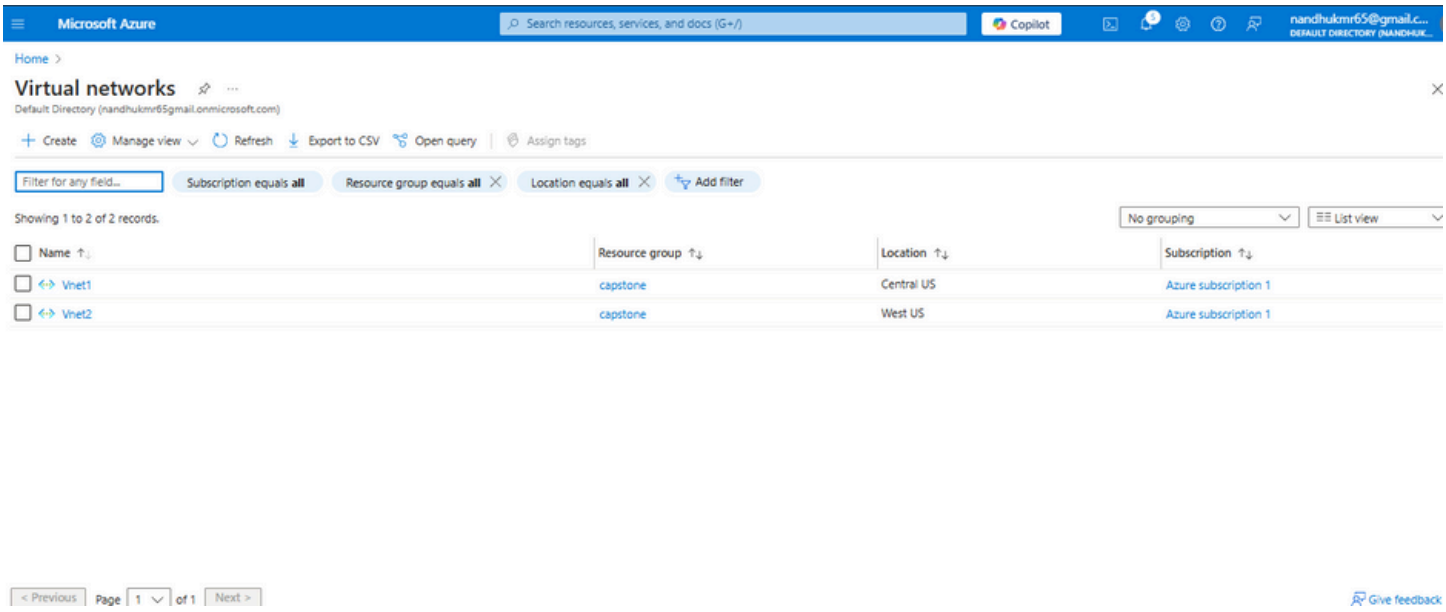
1. **Set Up Traffic Manager**:
   - Create an Azure Traffic Manager profile.
   - Add endpoints for Application Gateway 1 and Application Gateway 2 to ensure traffic distribution.

Step 7: Final Verification

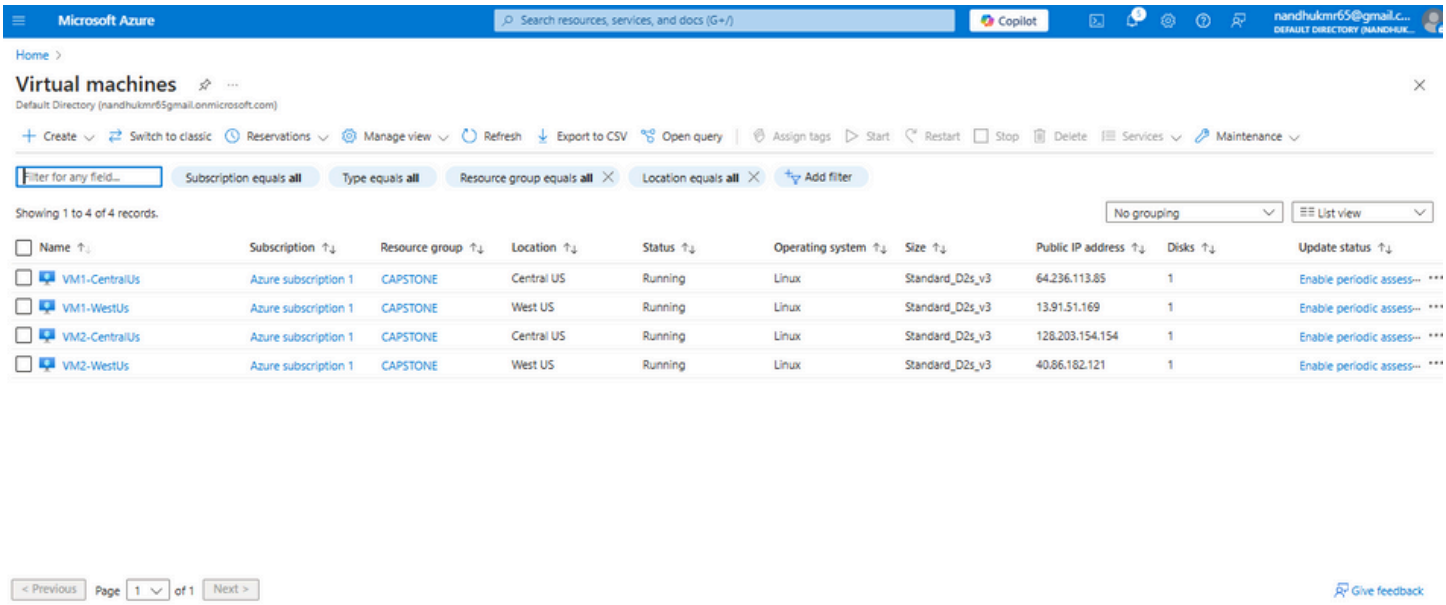
1. **Upload Image to Upload Page**:
   - Access the upload page via http://azure1044.trafficmanager.net/upload.
   - Upload an image to the upload page.
2. **Verify Image in Storage Account Container**:
   - Navigate to the Azure portal.
   - Go to the storage account and access the "upload" container.
   - Verify that the uploaded image is present in the "upload" container.

Screenshots:

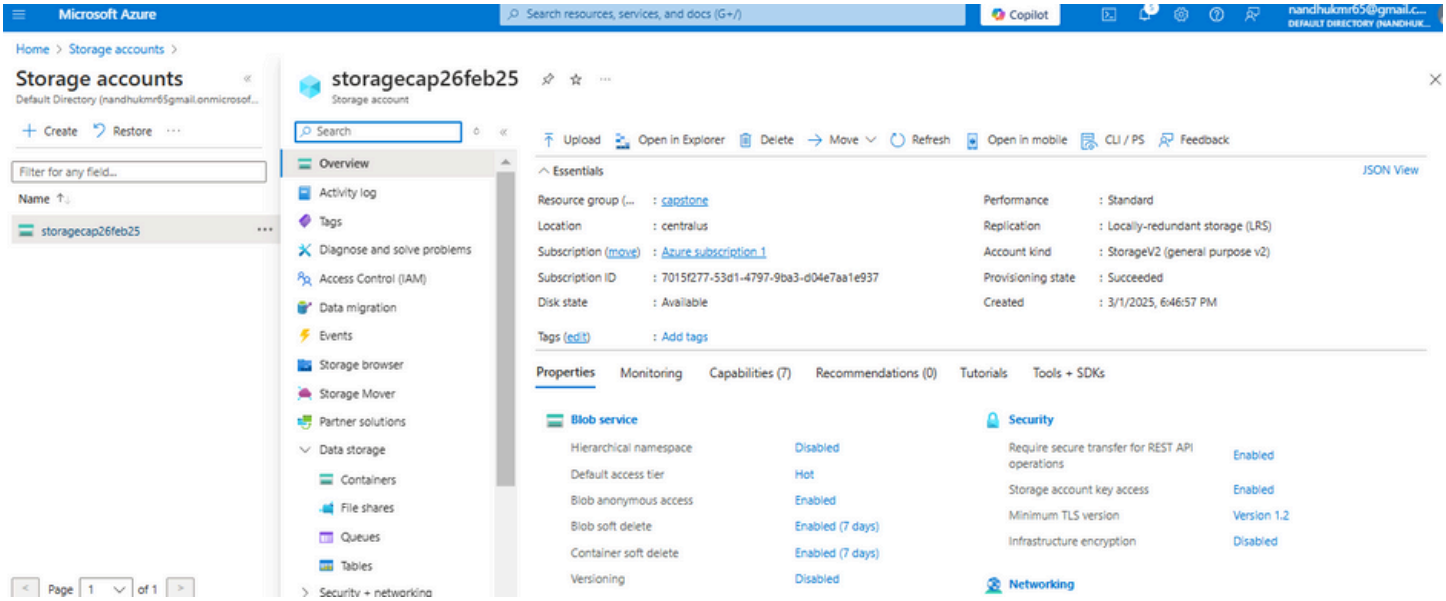
Screenshot 1: Azure Portal - Virtual Network Creation



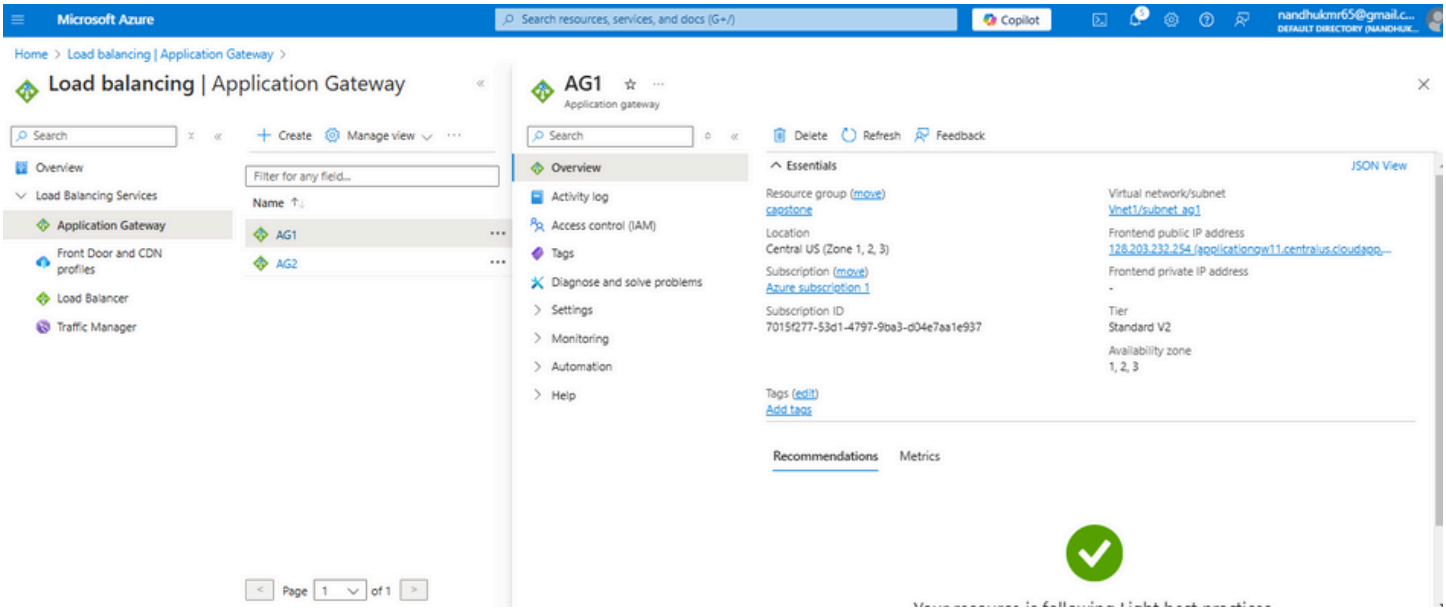
Screenshot 2: Azure Portal - Virtual Machine Deployment



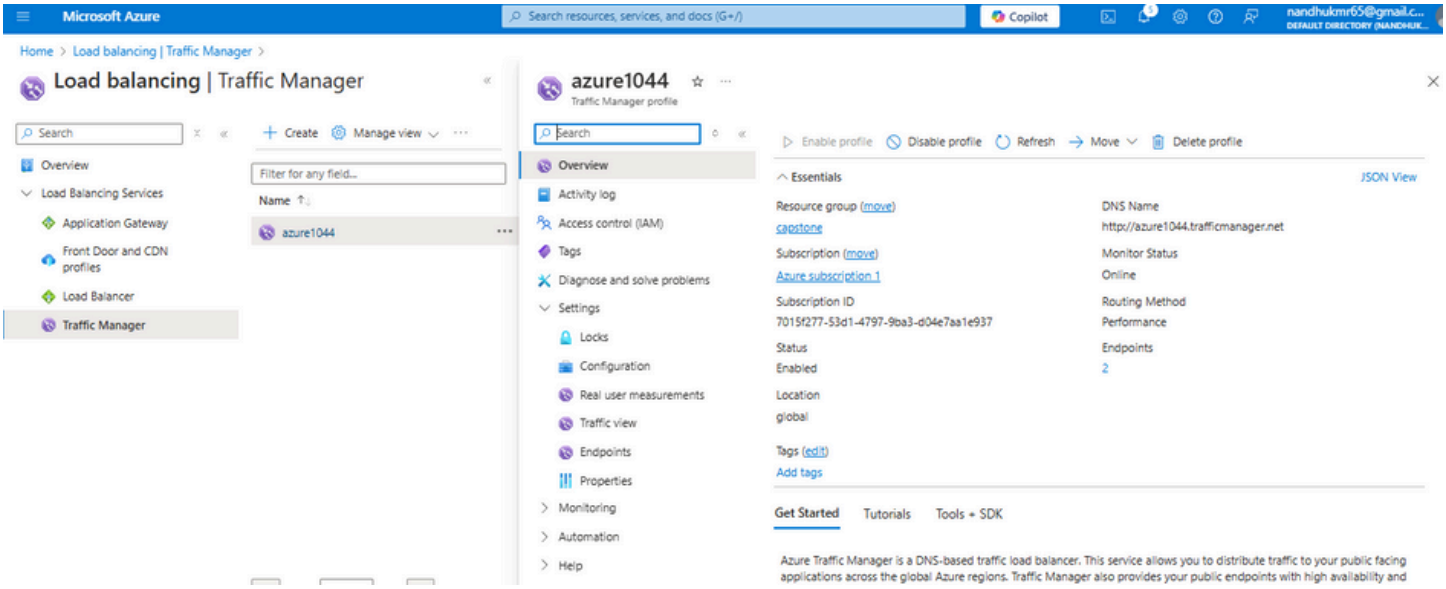
Screenshot 3: Azure Portal - Storage Account Configuration



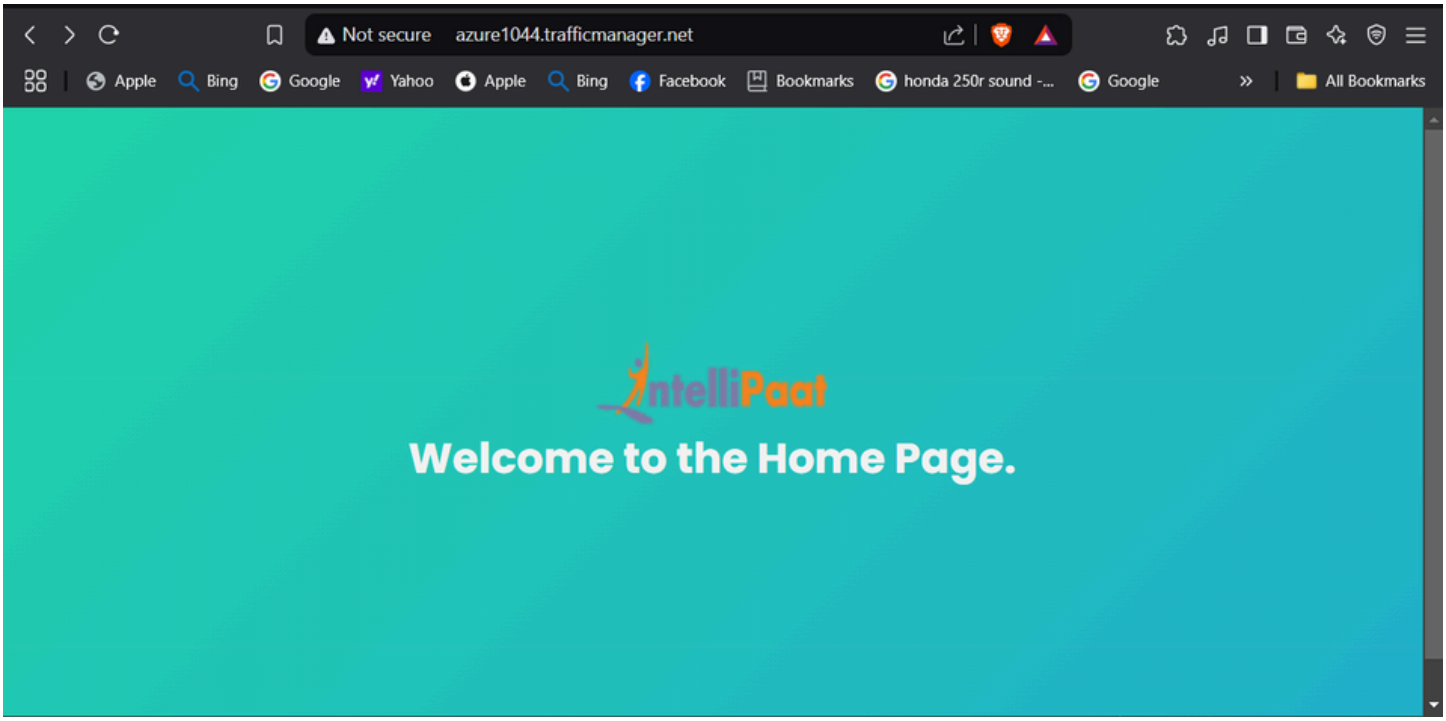
Screenshot 4: Azure Portal - Application Gateway Configuration

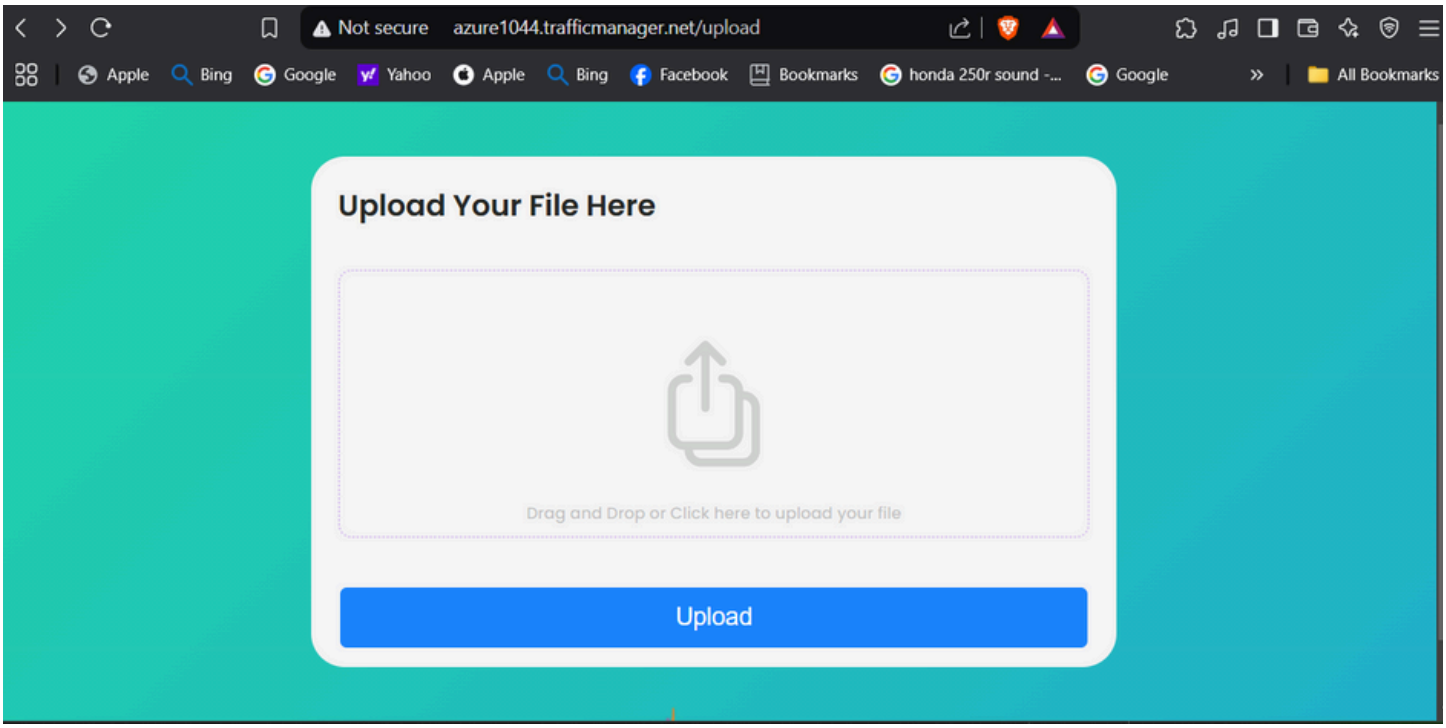


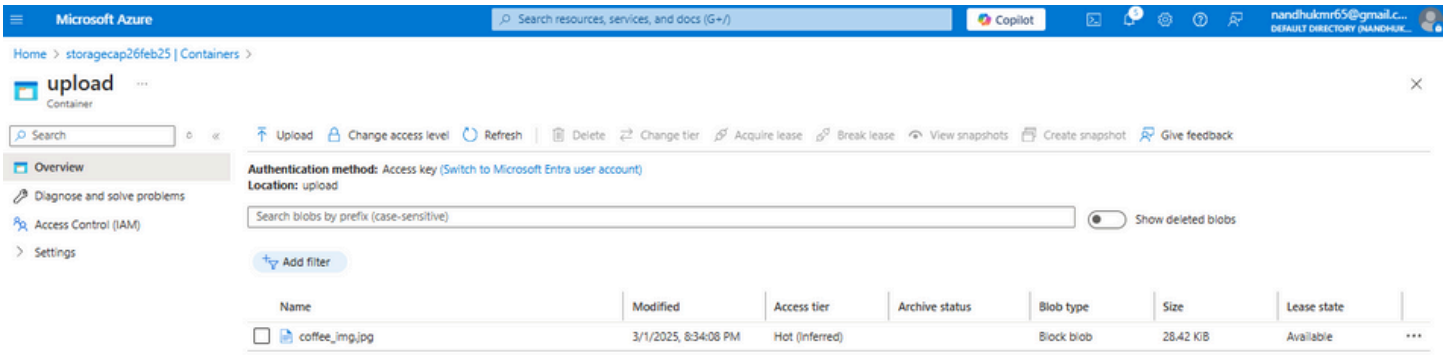
Screenshot 5: Azure Portal - Traffic Manager Configuration



Screenshot 6: Final Verification - Image in Storage Account Container







Conclusion:

This project successfully implements an Azure architecture to deploy web pages, configure Application Gateways, Traffic Manager, VNets, and Storage Accounts. The traffic is distributed optimally between Central US and West US regions, ensuring high availability and performance for the company's website. The final verification step confirmed that the upload page functionality works as expected by uploading an image and verifying its presence in the storage account container.

Thank you for reviewing this submission. Please let me know if there are any further requirements or if I can assist with anything else.