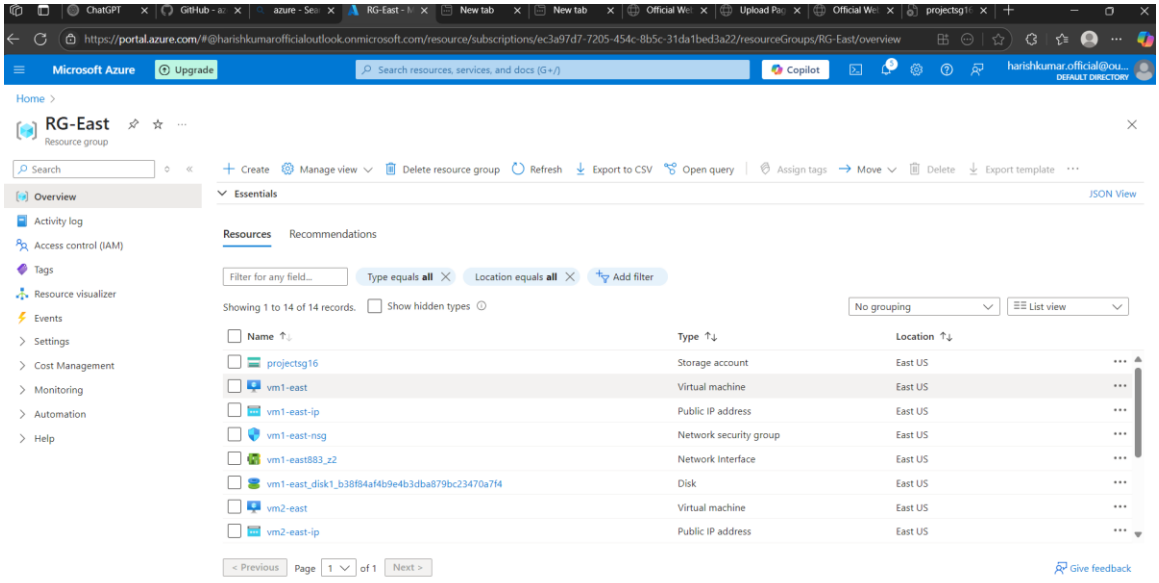
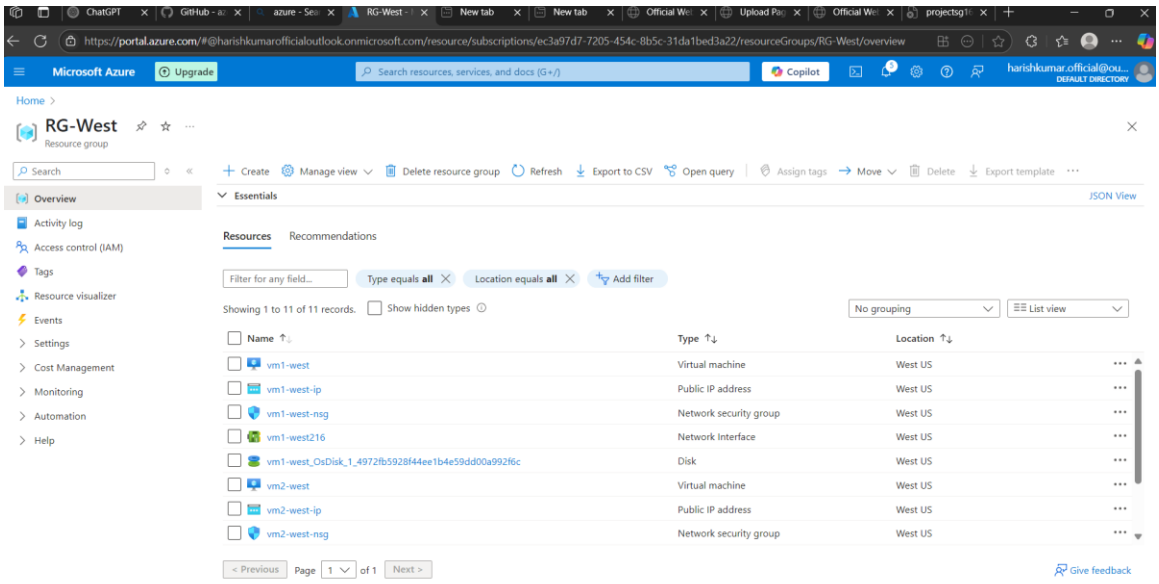
Azure Capstone Project:

Multi-Region File Upload Web Application Using Virtual Machines, Blob Storage, and Traffic Manager.

Step 1: Resource Groups (East & West)

Description: Shows both resource groups (RG-East and RG-West) created for organizing resources in respective regions.

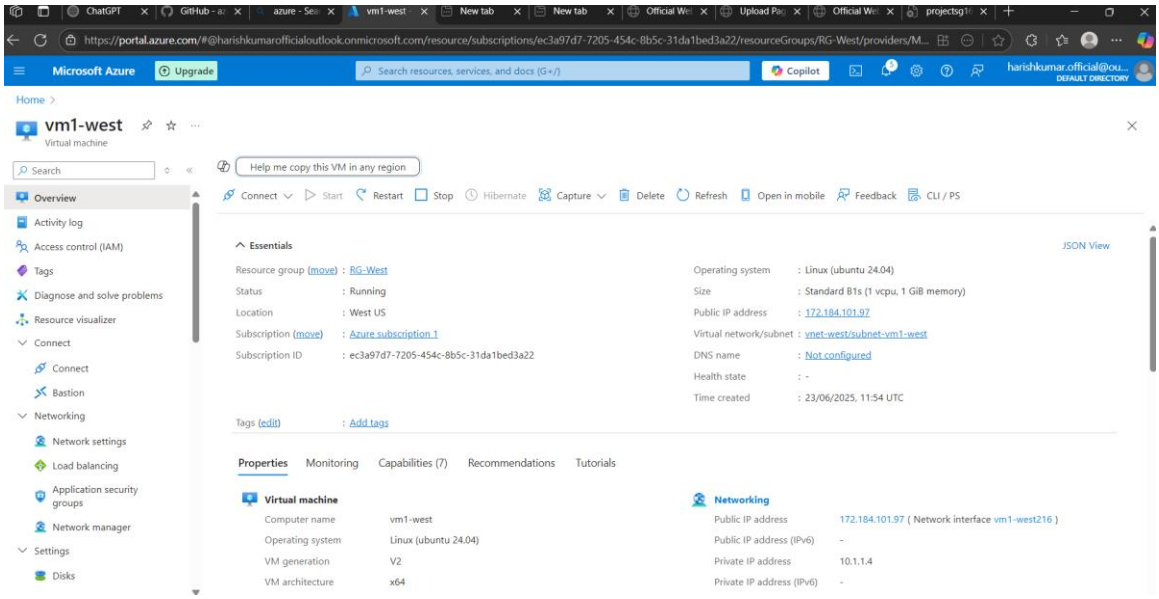


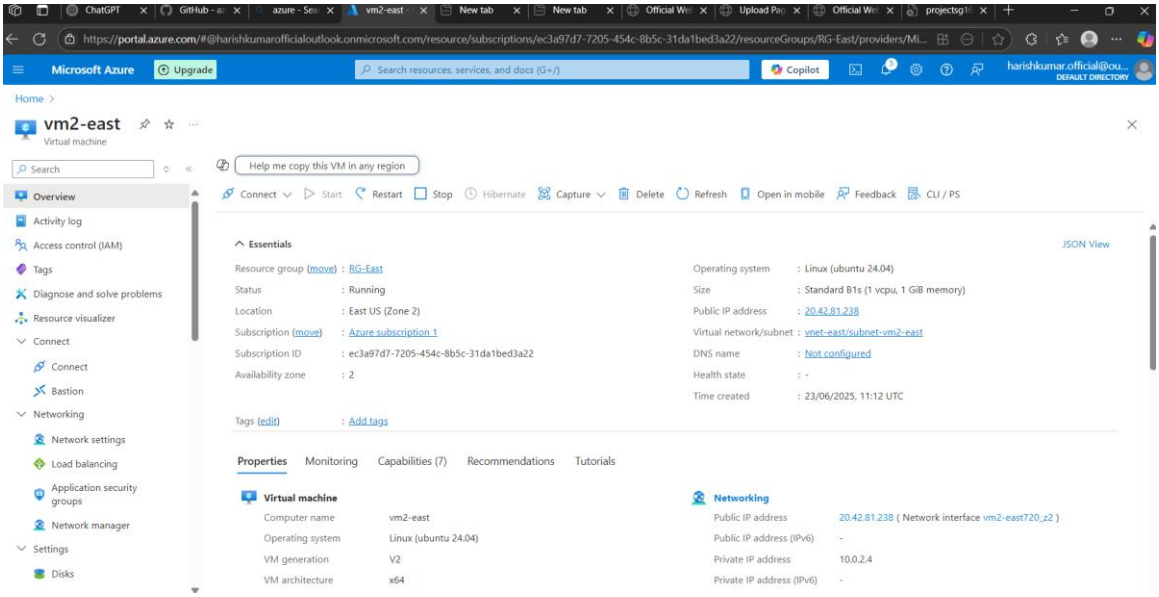


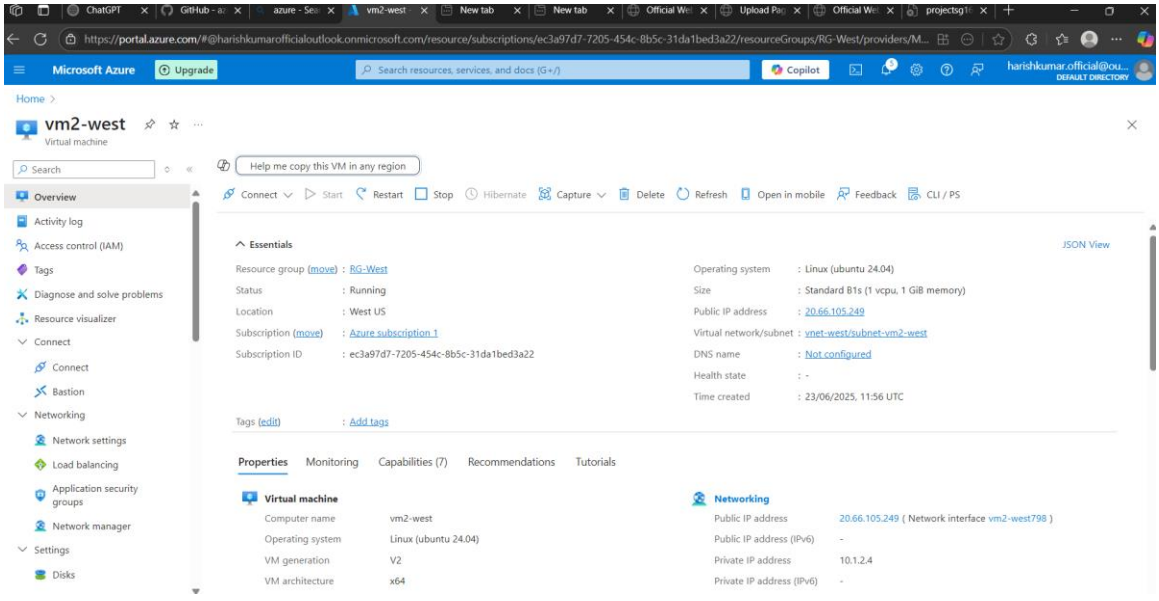
Shows both resource groups (RG-East and RG-West) created for organizing resources in respective regions.

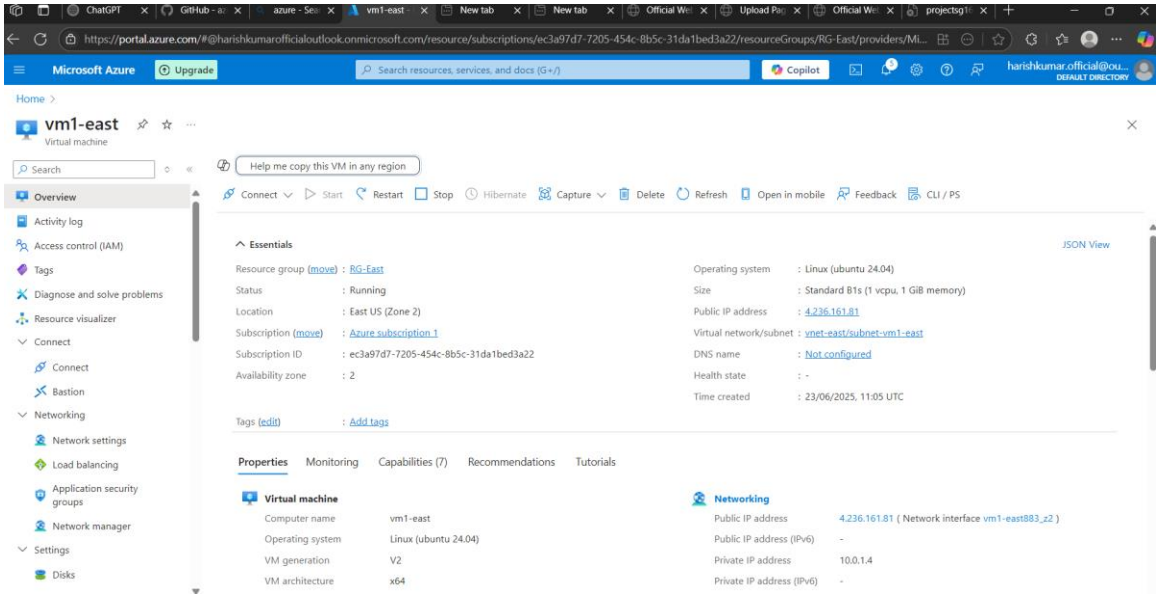
Step 2: Virtual Machines Overview

Description: Displays all 4 VMs across both regions with their status and regions.





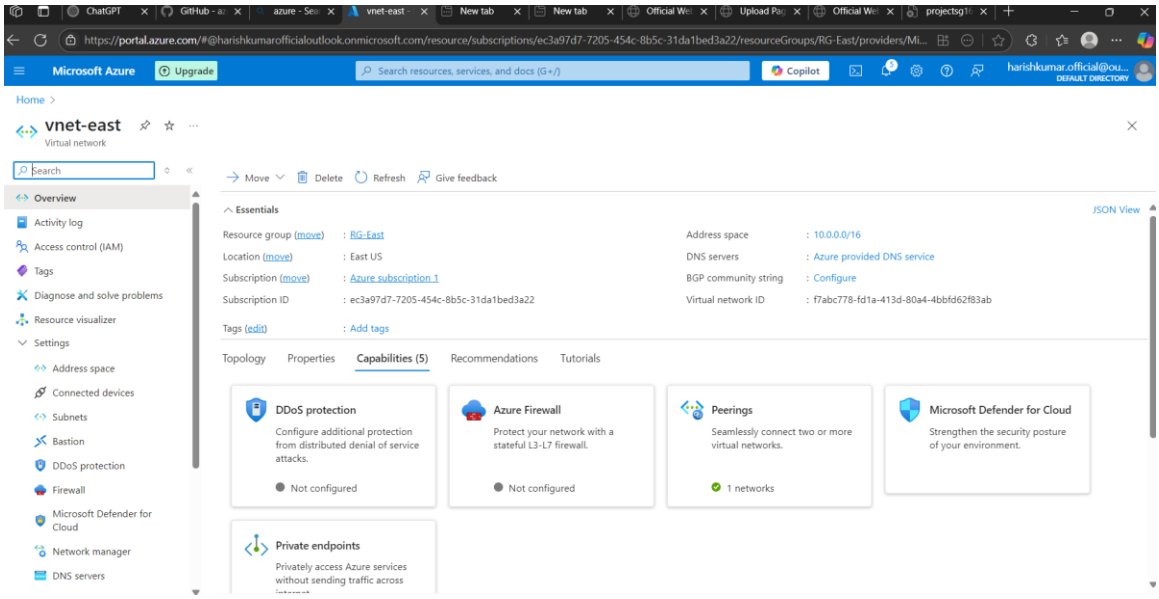


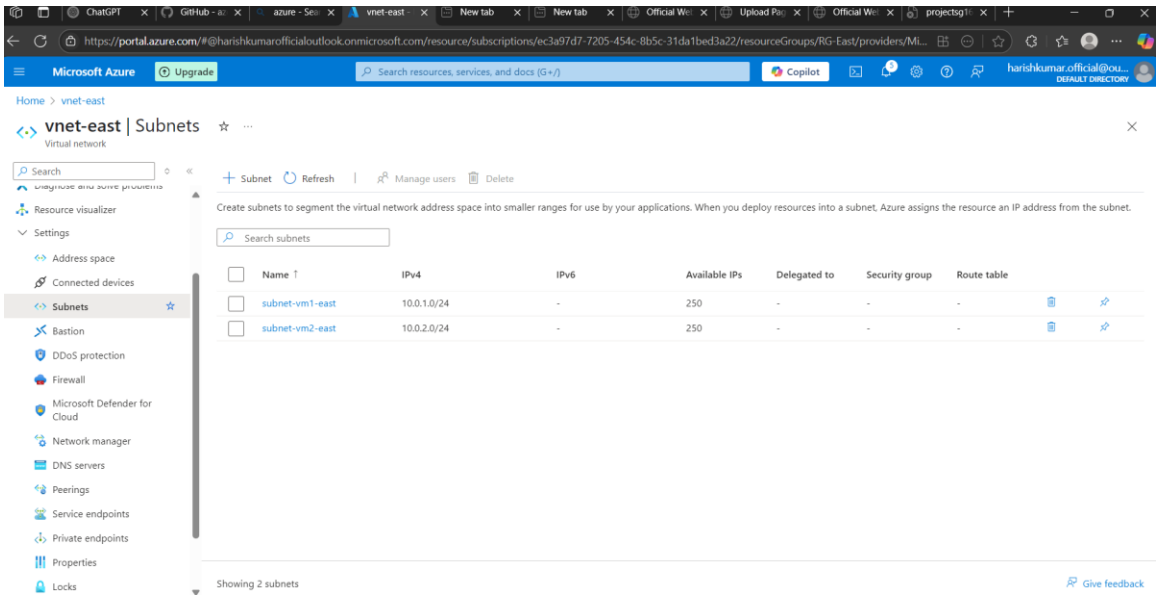


Overview of four virtual machines deployed across East and West regions, displaying status and distribution.

Step 3: Virtual Network and Subnets – East

Description: Displays subnets created under vnet-east for vm1 and vm2.

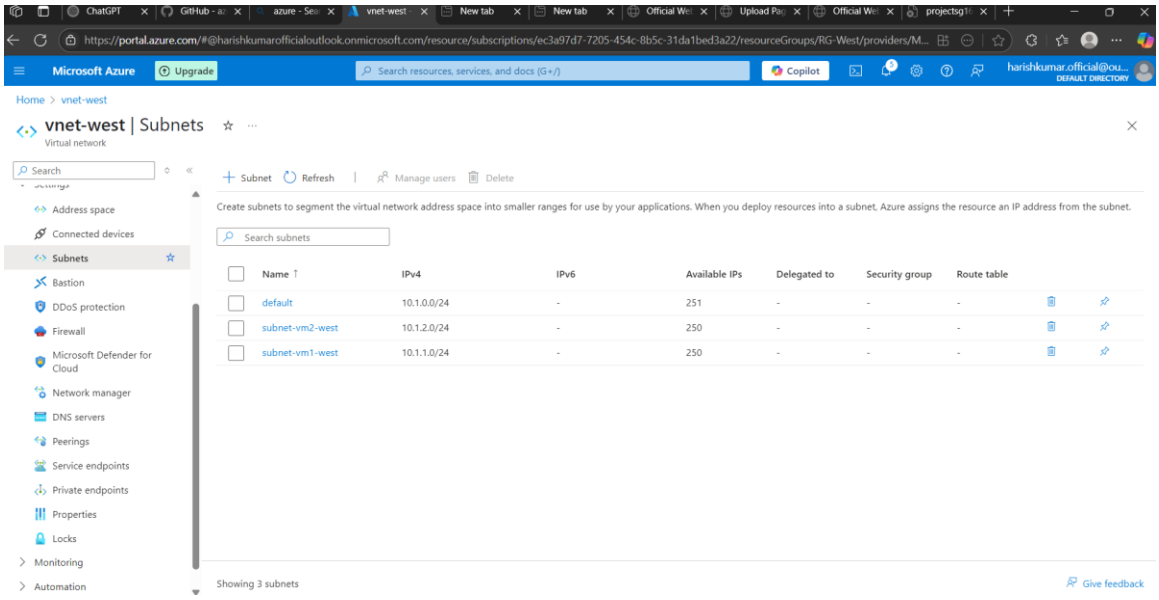


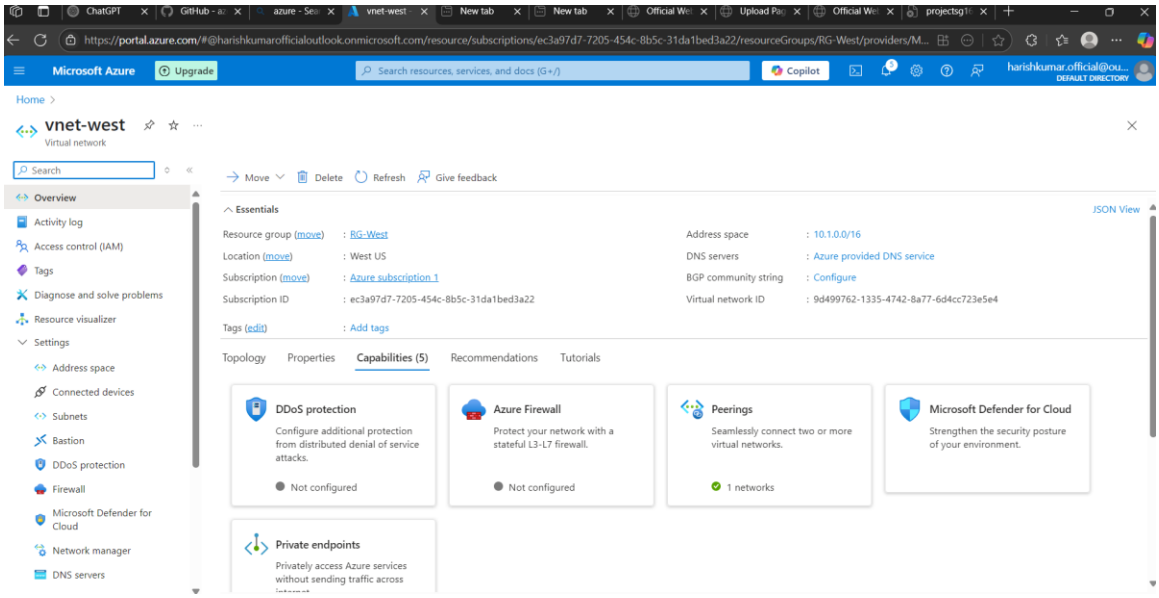


Subnets under vnet-east were configured to support vm1 and vm2 for the East region deployment.

Step 4: Virtual Network and Subnets - West

Description: Displays subnets created under vnet-west for vm1 and vm2.

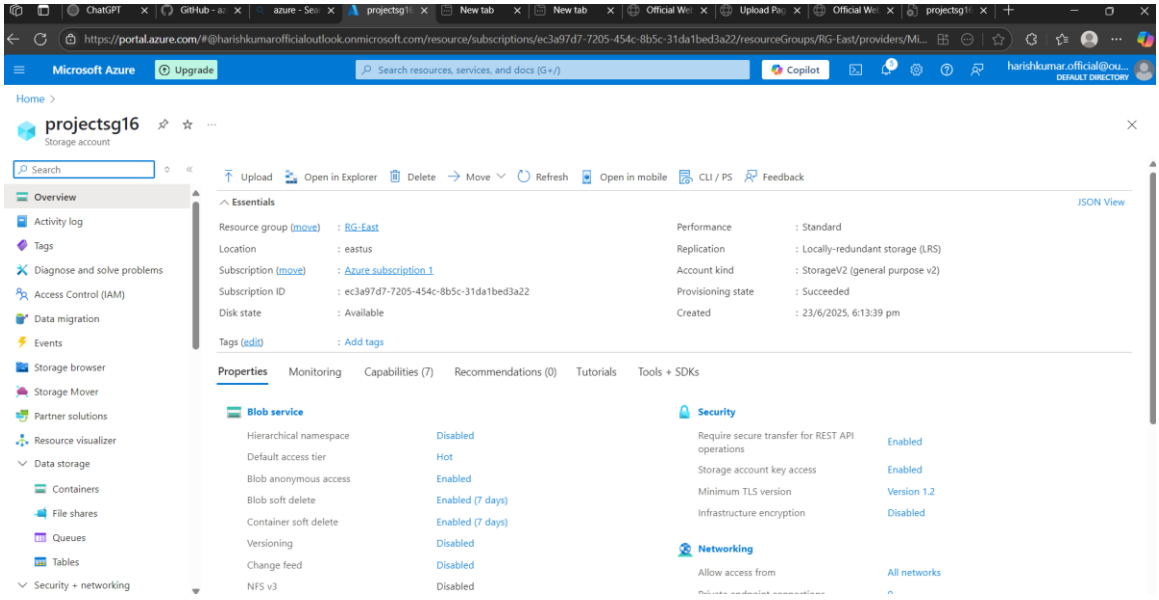




Subnets under vnet-west were created to host vm1 and vm2 for the West region deployment.

Step 5: Storage Account Overview

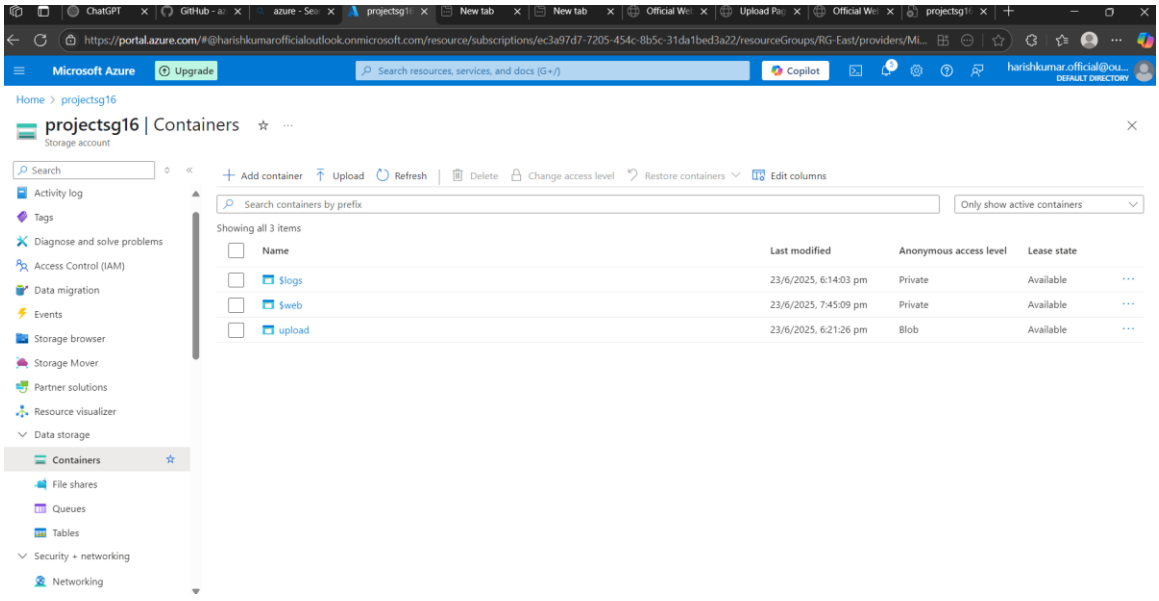
Description: Storage account (projectsg16) used to store uploaded files.



Storage account 'projectsg16' provisioned to manage and store uploaded application files.

Step 6: Upload Container in Storage

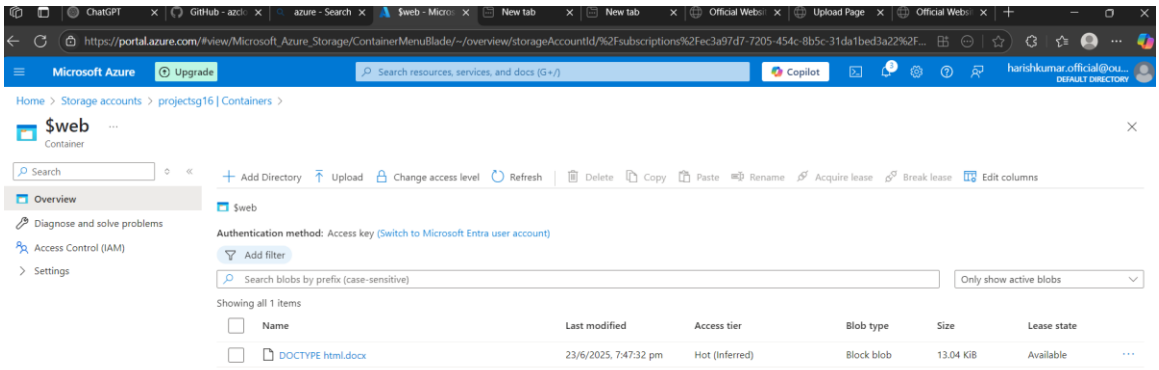
Description: Shows the 'upload' container with test files inside.

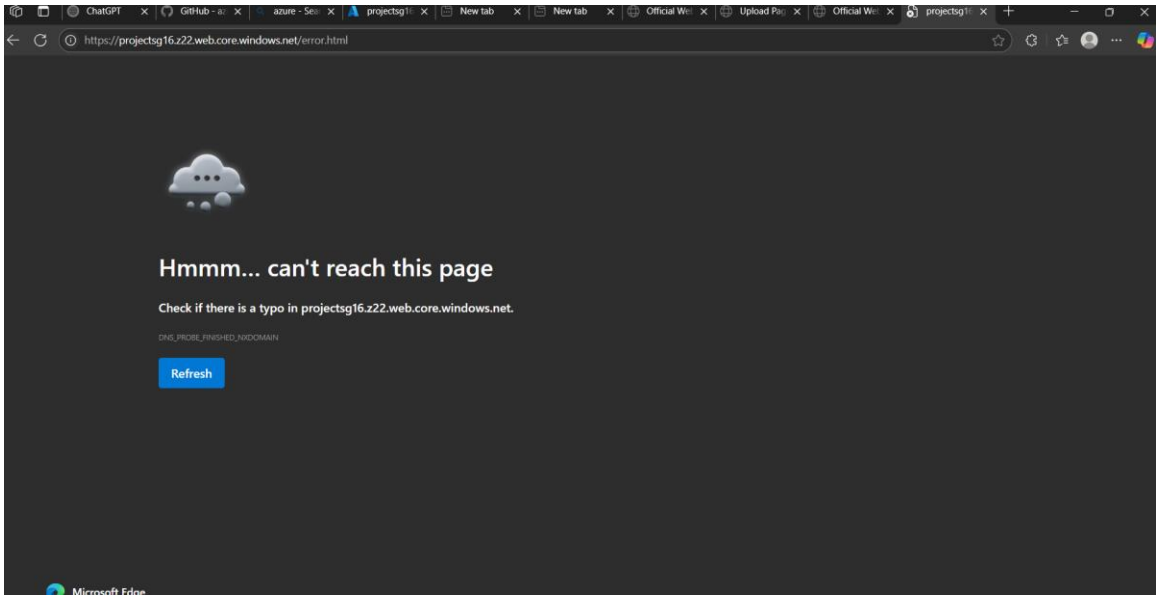


The 'upload' container within the storage account shows uploaded test files for validation.

Step 7: Static Website Configuration

Description: Enabling static website and setting error.html as fallback.

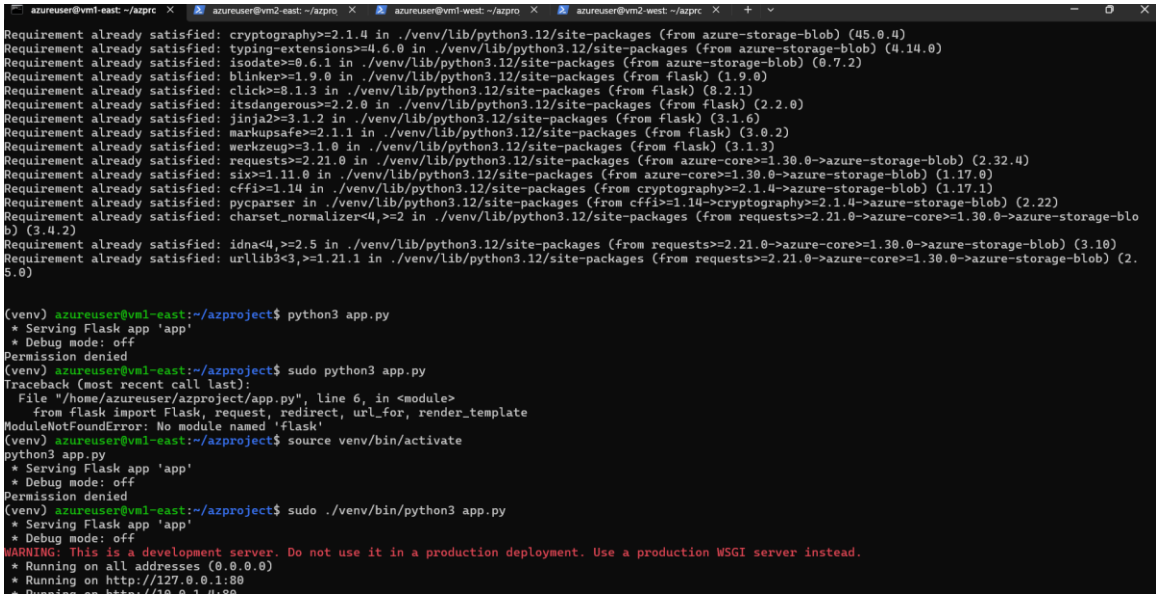




Static website hosting was enabled with a fallback to error.html in case of routing issues

Step 8: VM1-East Terminal Running Flask

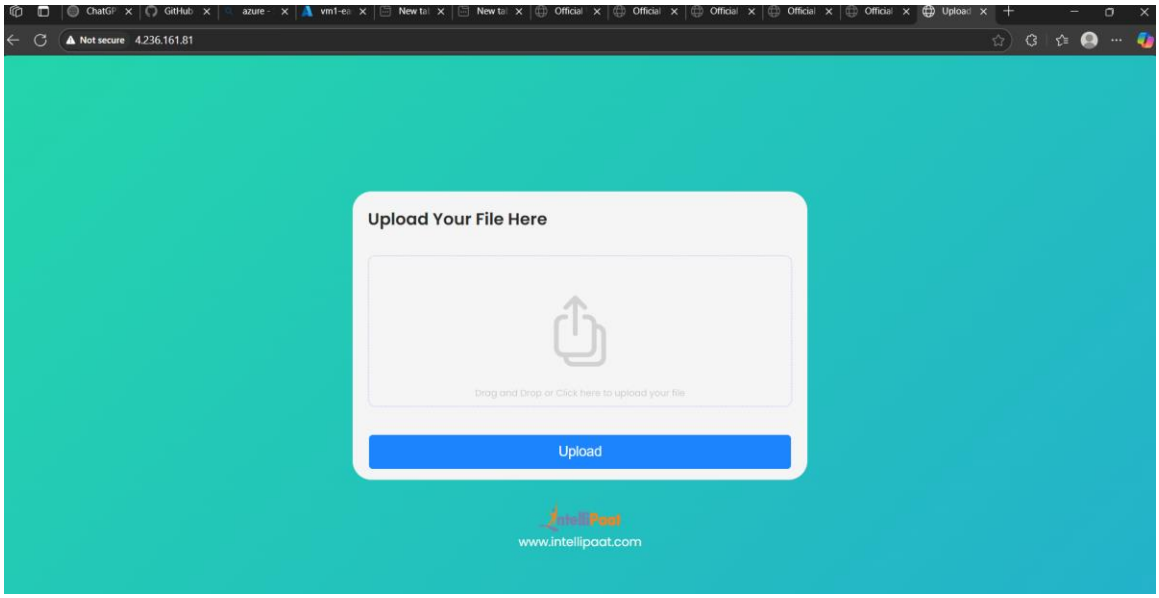
Description: Flask app launched to serve the upload form from vm1-east.



Flask application deployed and running on vm1-east to serve the file upload form.

Step 9: VM1-East Browser Upload Page

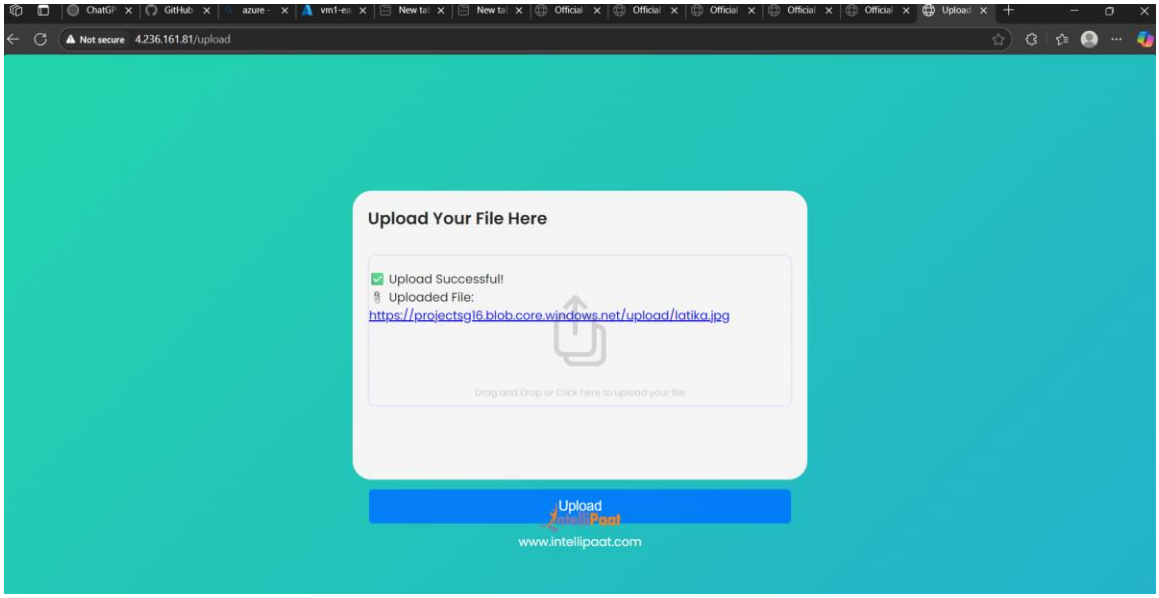
Description: Upload page accessed through public IP of vm1-east.



Upload form accessed via public IP of vm1-east, confirming external reachability.

Step 10: VM1-East Upload Success

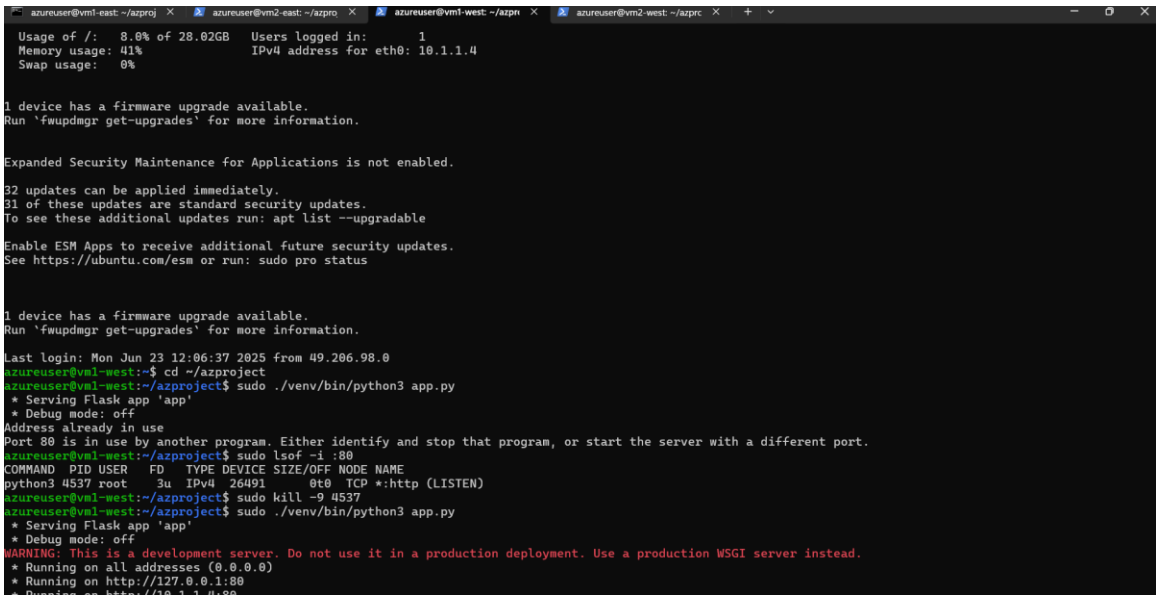
Description: File uploaded successfully from east VM.



File upload confirmed via the east VM with a success message, validating endpoint functionality.

Step 11: VM1-West Terminal Running Flask

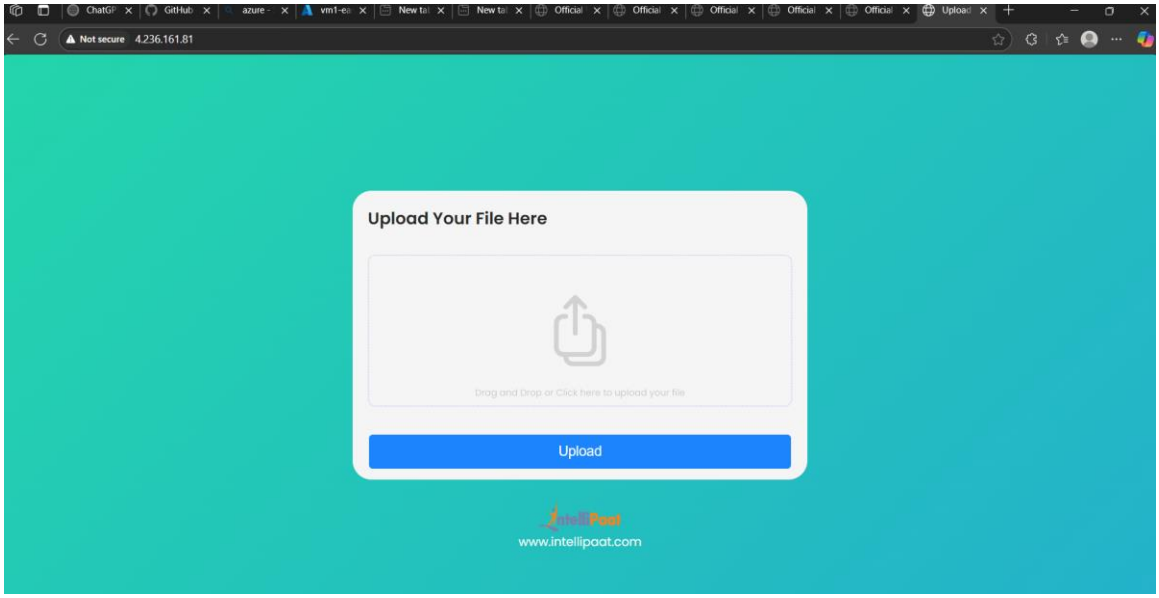
Description: Flask app launched to serve the upload form from vm1-west



Flask application launched on vm1-west to serve the same upload interface.

Step 12: VM1-West Browser Upload Page

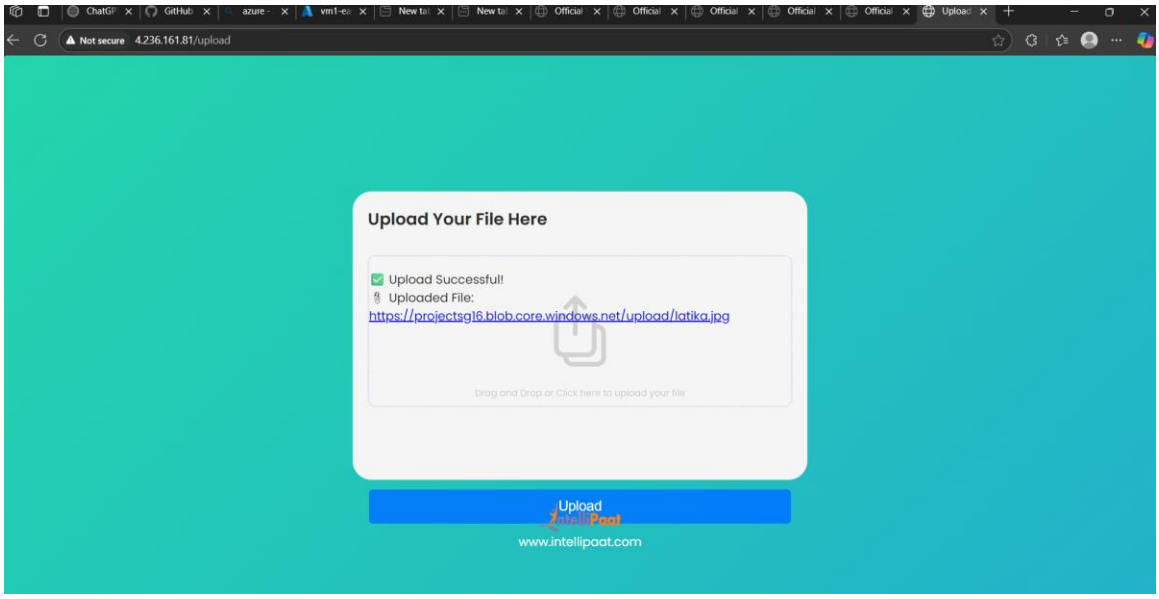
Description: Upload page accessed through public IP of vm1-west.



Upload form successfully accessed through vm1-west's public IP address.

Step 13: VM1-West Upload Success

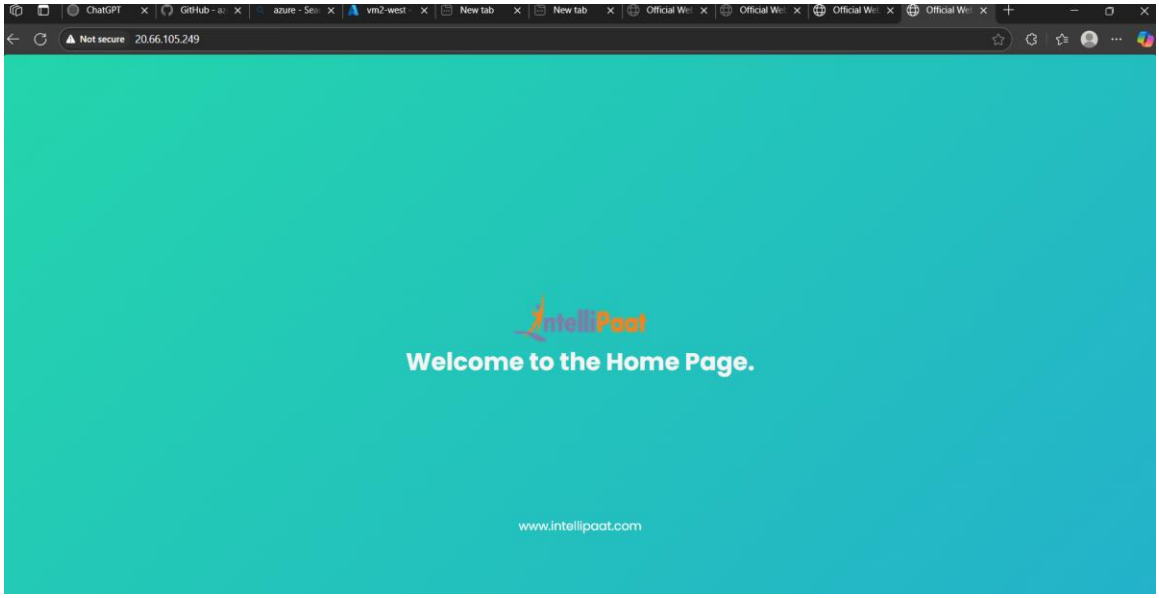
Description: File uploaded successfully from west VM.



File uploaded from the west VM, confirming identical behavior to the east deployment.

Step 14: VM2-East Homepage

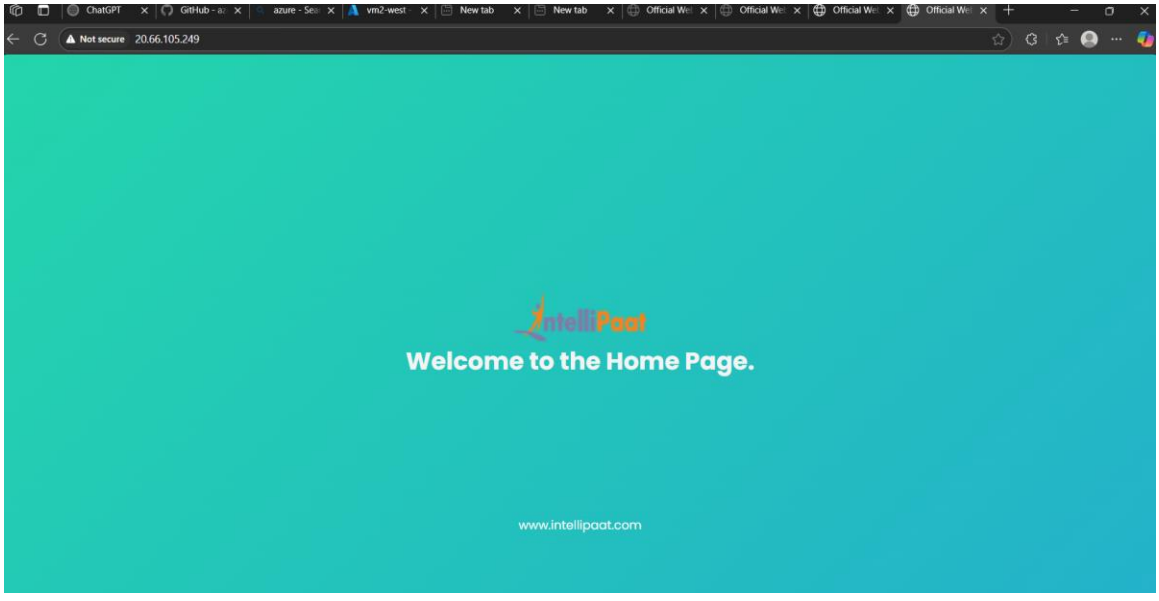
Description: Home page served by vm2-east showing static welcome.



vm2-east is serving a static home page, validating regional frontend availability.

Step 15: VM2-West Homepage

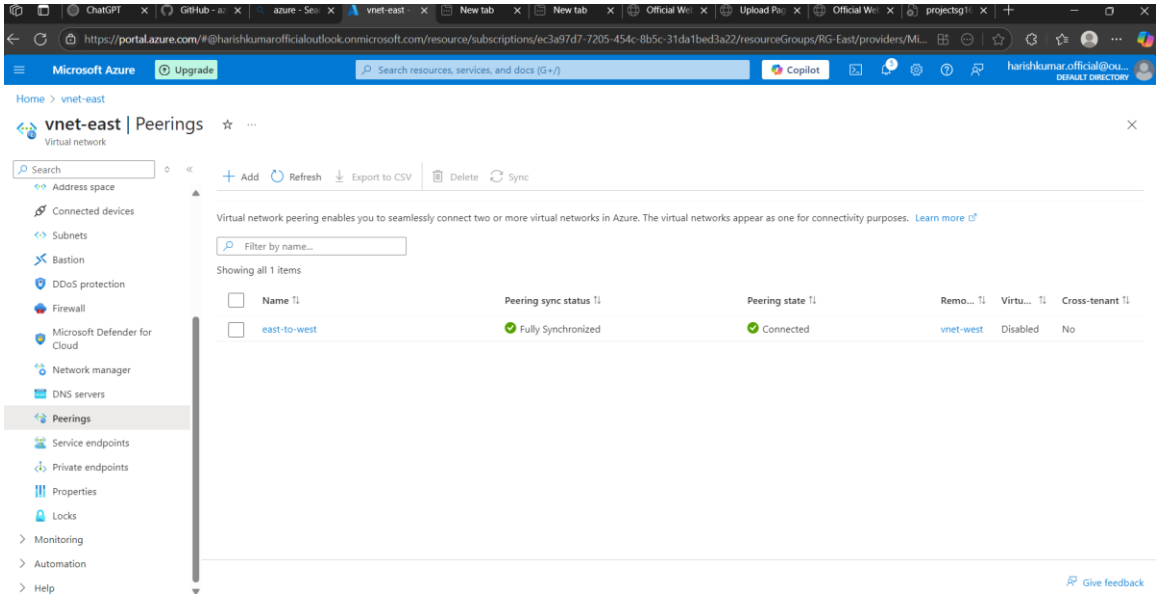
Description: Home page served by vm2-west showing static welcome.



vm2-west displays a static home page, mirroring the east region's functionality.

Step 16: VNet Peering - East to West

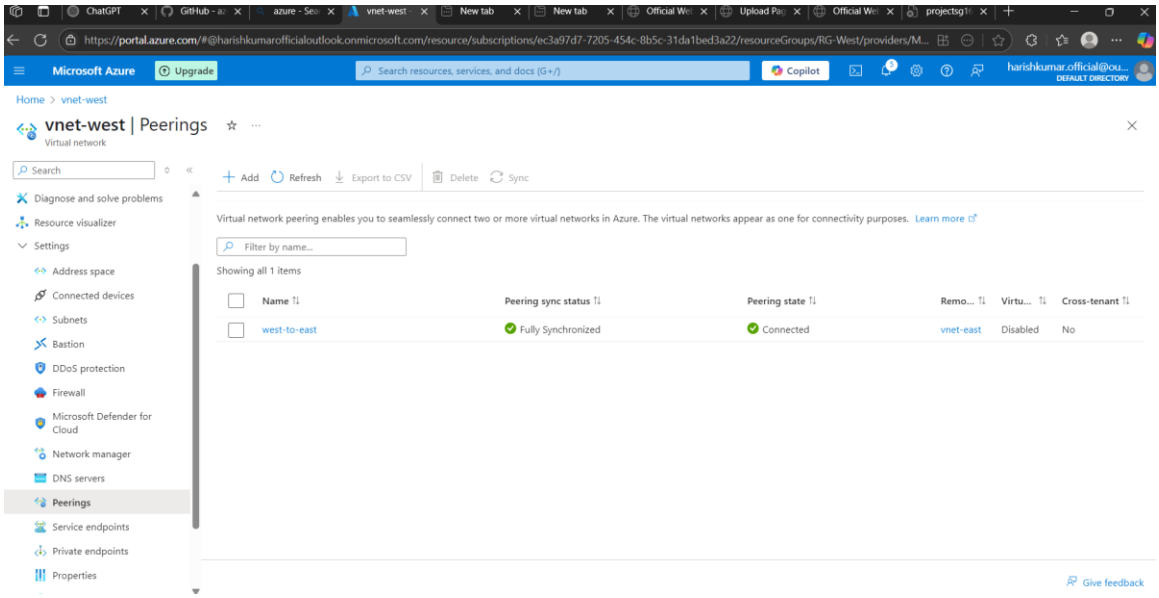
Description: vnet-east peered with vnet-west to allow internal communication.



vnet-east peering established to allow secure communication with vnet-west.

Step 17: VNet Peering - West to East

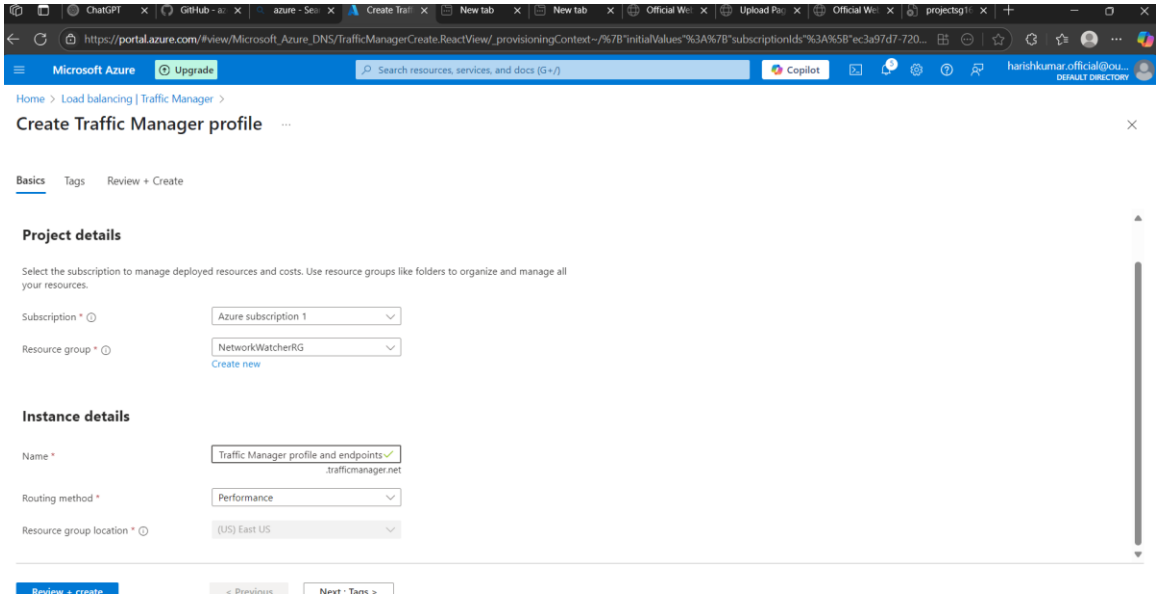
Description: vnet-west peered with vnet-east to allow internal communication.



vnet-west configured to peer with vnet-east, enabling bi-directional connectivity.

Step 18: Traffic Manager Profile

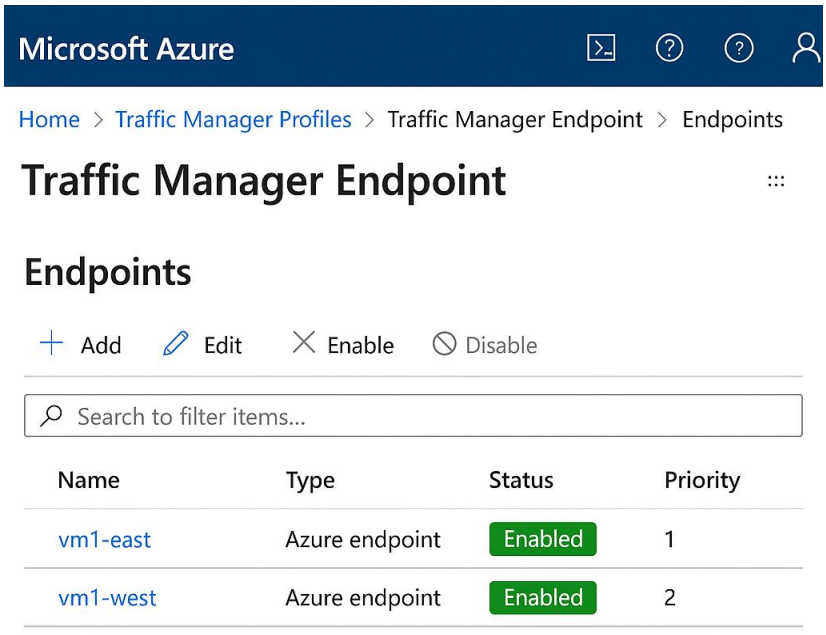
Description: Traffic Manager profile setup with performance routing.



Traffic Manager profile created with performance-based routing for optimal response time.

Step 19: Traffic Manager Endpoints

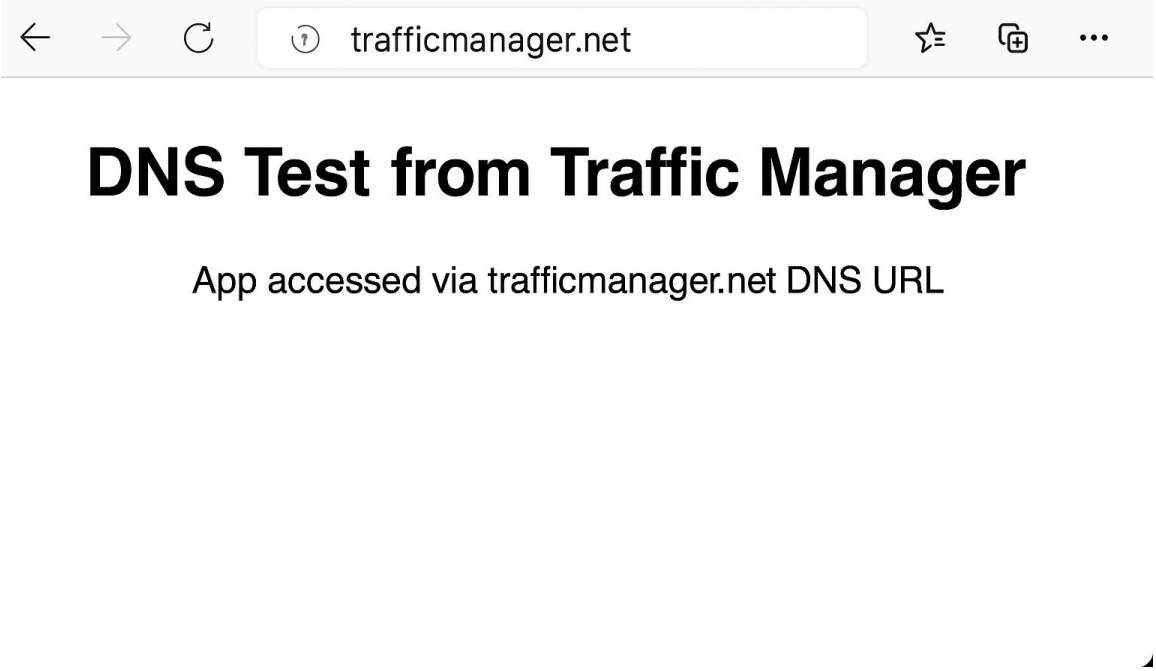
Description: Endpoints added for vm1-east and vm1-west.



Traffic Manager endpoints configured for both vm1-east and vm1-west for load balancing.

Step 20: DNS Test from Traffic Manager

Description: App accessed via trafficmanager.net DNS URL.



Application accessed successfully through Traffic Manager's DNS URL, confirming setup.