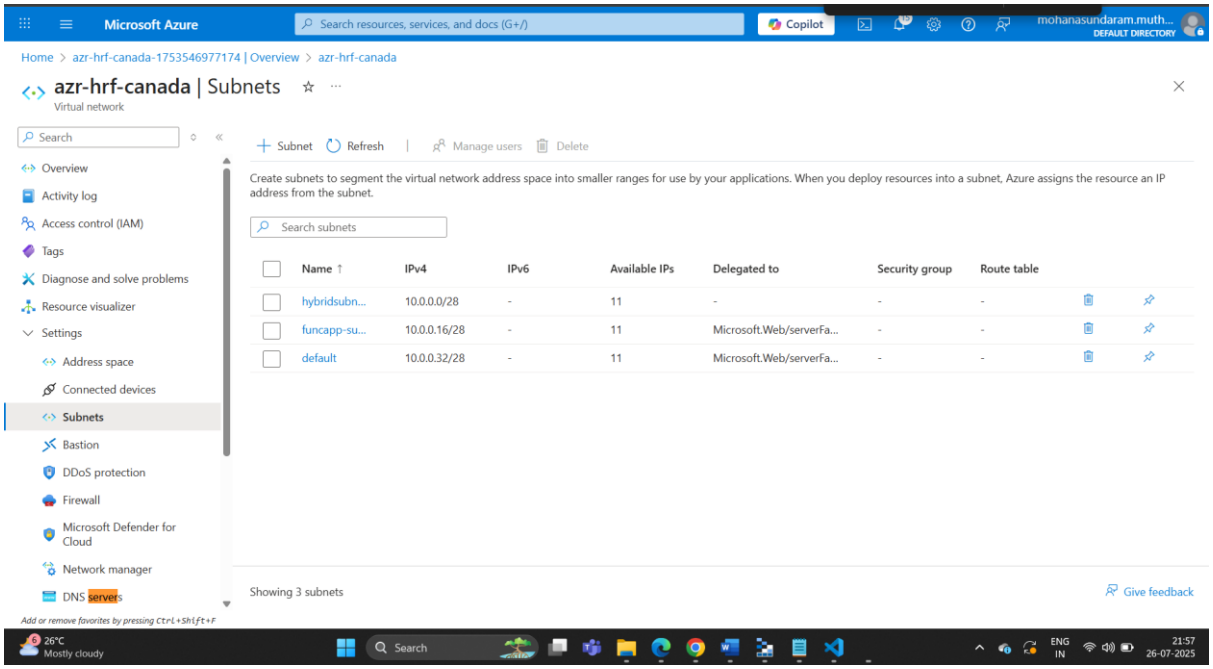
This step is continuation of Phase 1, in Phase 1 we have done everything as public we are doing everything as private:

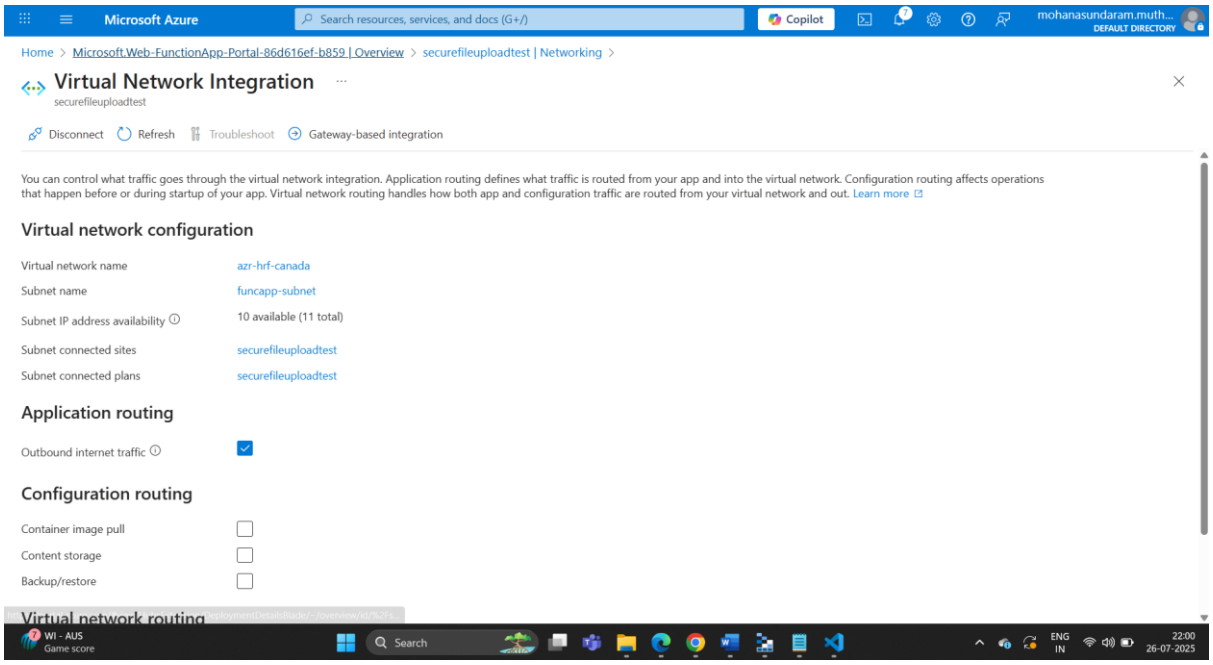
1.Created Vnet and Three subnets

Hybridsubnet-Private endpoints
Function-appsubnet- delegated to Function app(Vnet integration)
Default-delegated to web app(Vnet integration)

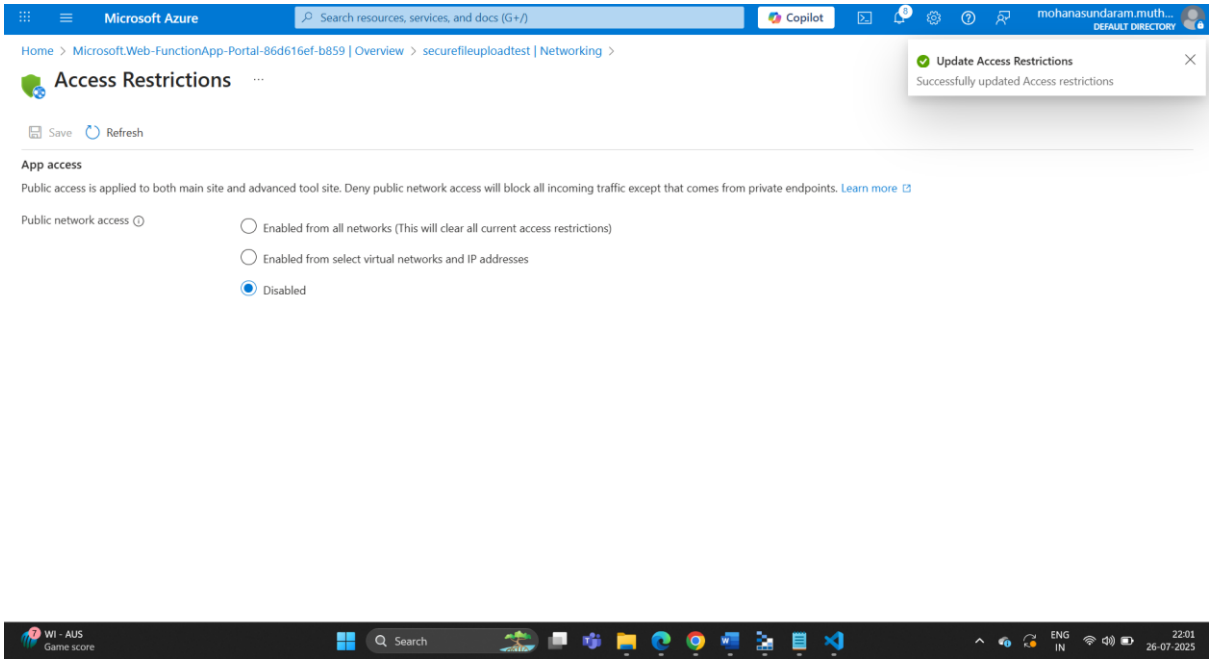
Note: Delegated subnets cannot be used by other resources,
If it delegated for function app it cannot be used by web app also



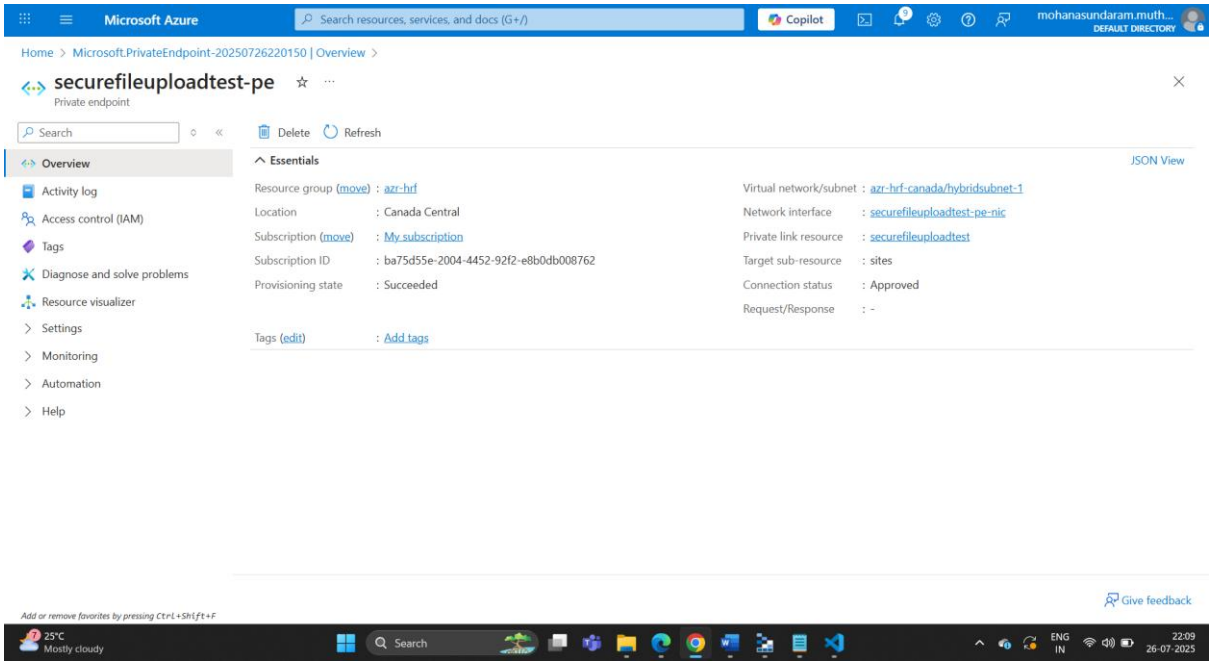
Enabled Vnet integration for my web app



To enable private access, disabling public access in function app by below image

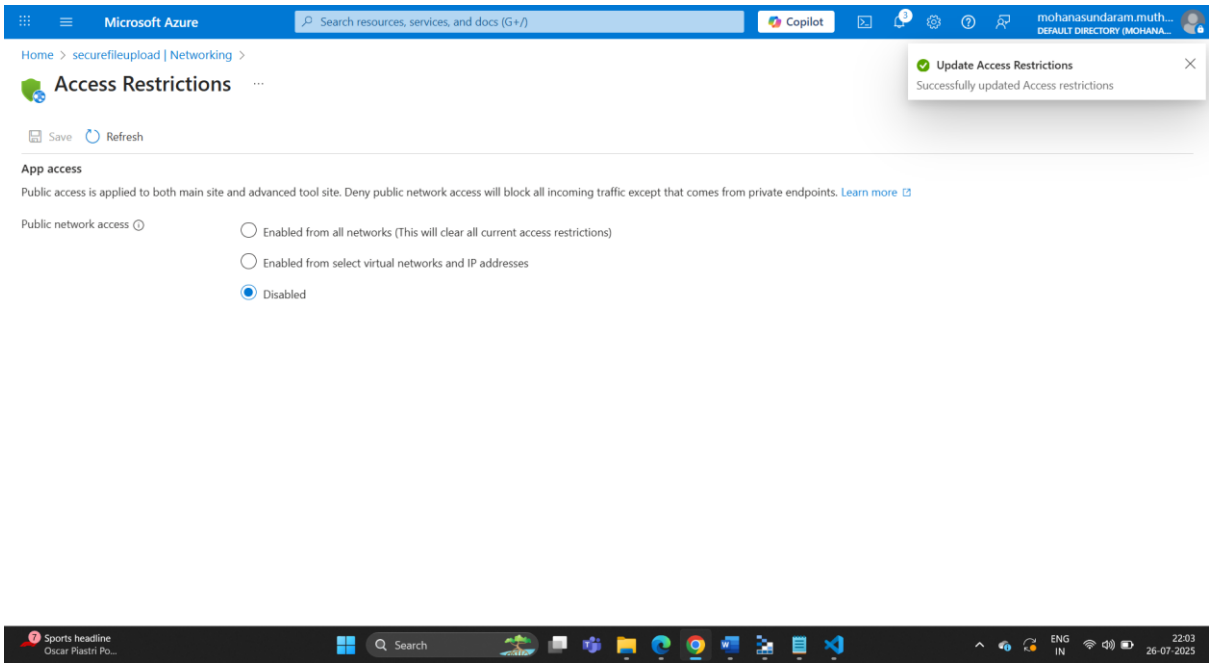


Craeted PE for functionapp by hybridsubnet

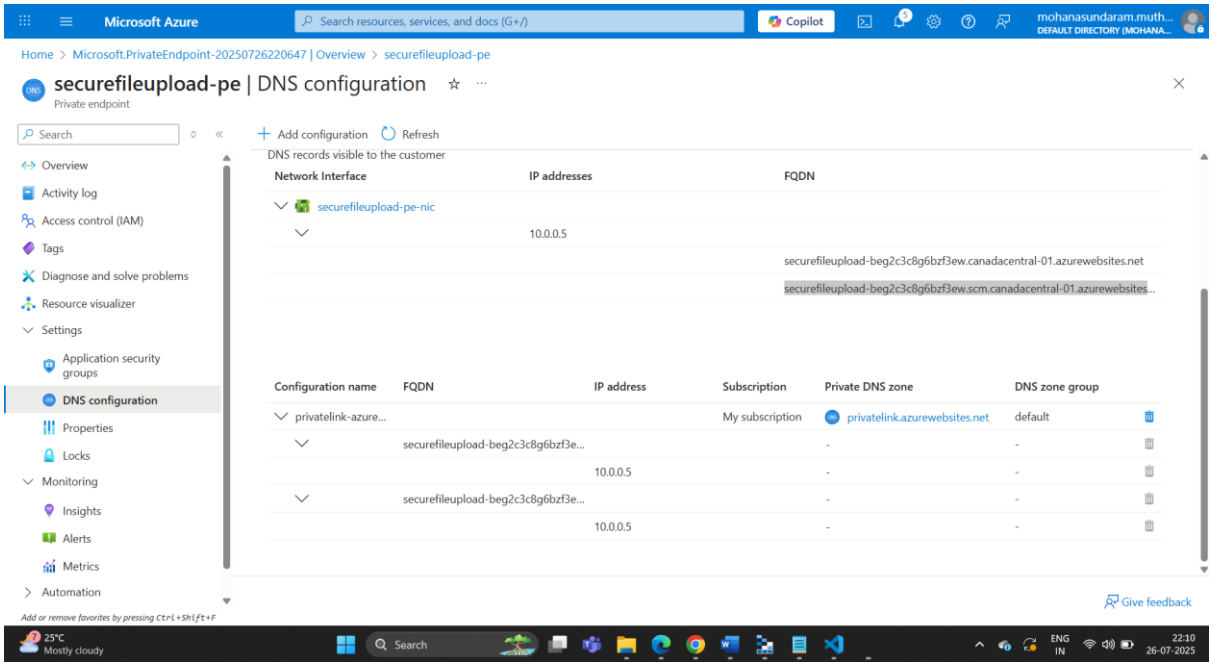


Web app:

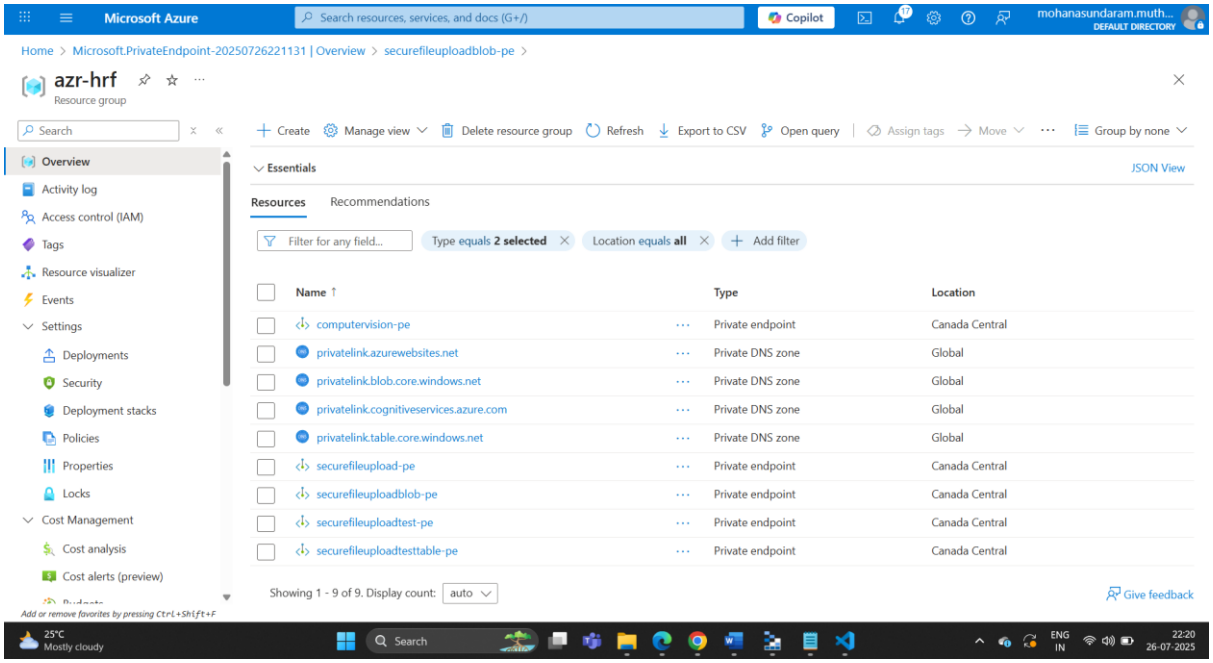
Public access disabled for web app and created Vnet Integration:

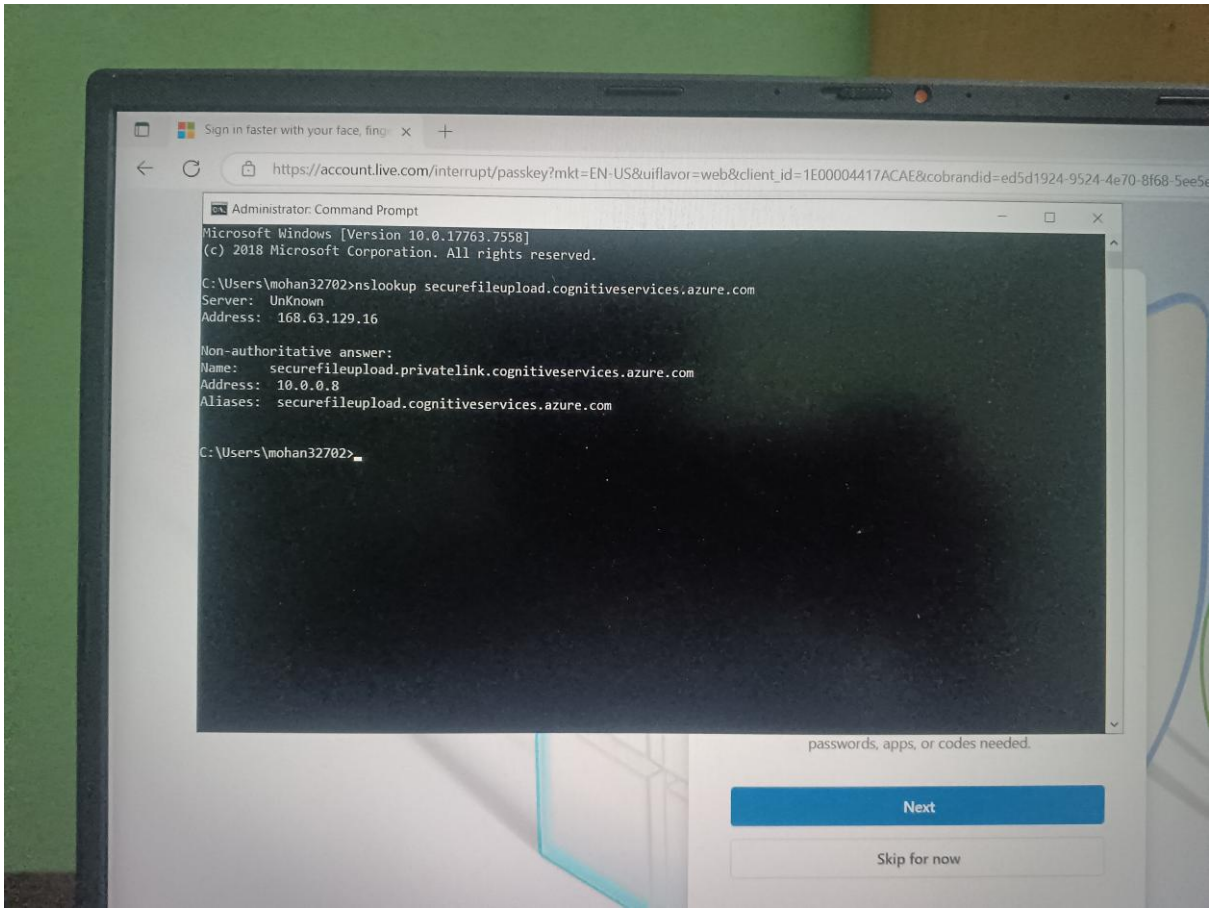


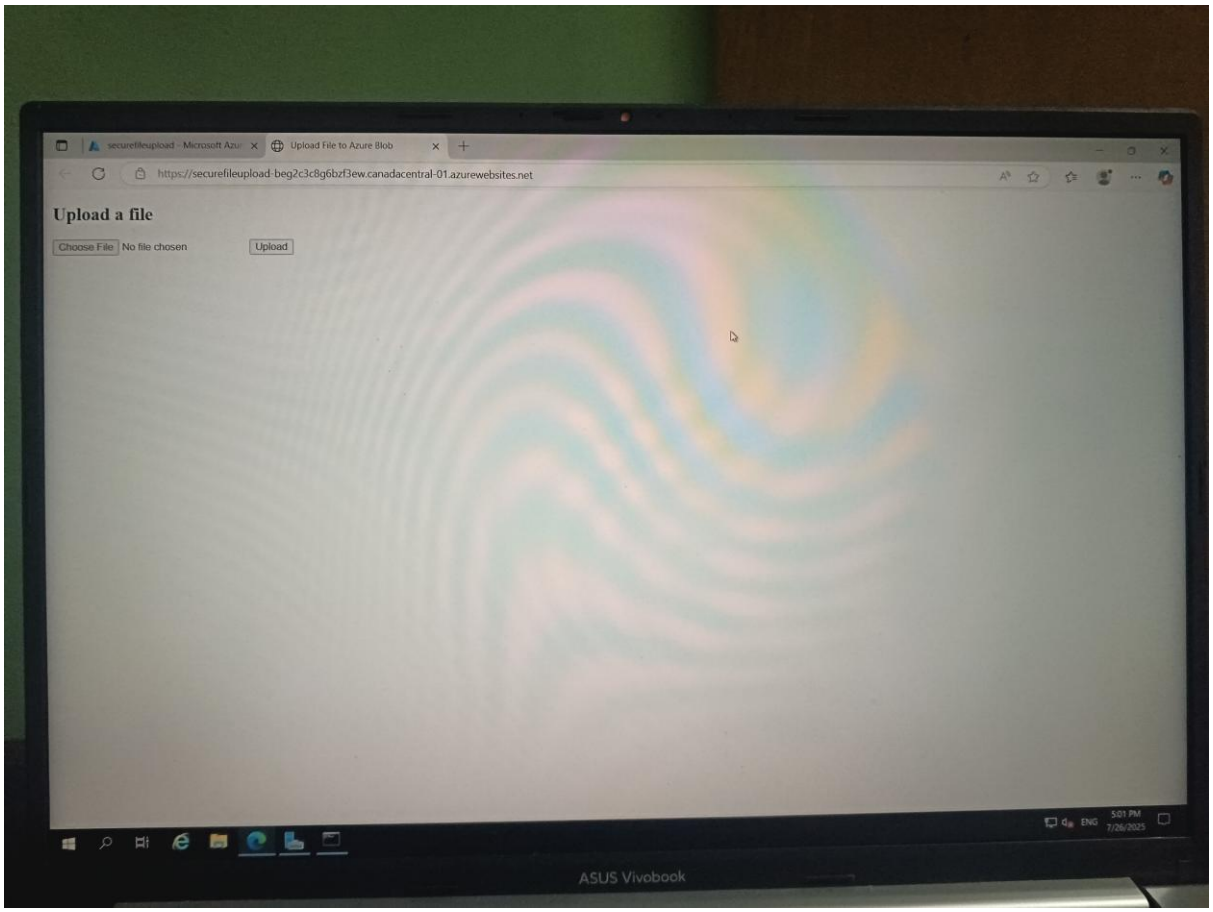
Created PE for web app



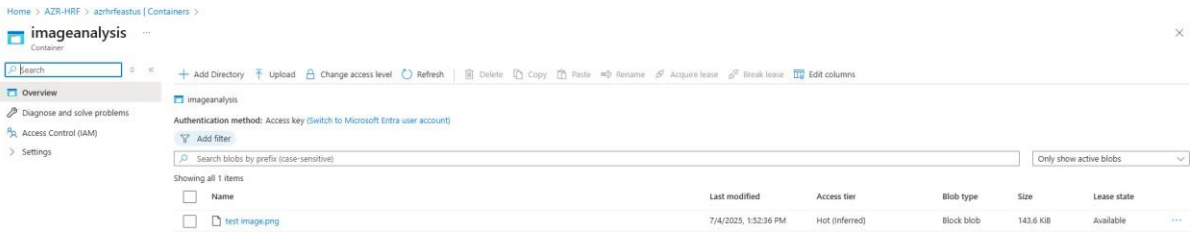
Created PE for storage Table,Blob, function app,web app, and computer vision service and Private DNS and Vnet is linked automatically during pe creation itself


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
mohanasundaram.muth...
DEFAULT DIRECTORY
Home > Microsoft.PrivateEndpoint-20250726221131 | Overview > securefileuploadblob-pe >
azr-hrf
Resource group
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Events
Settings
Deployments
Security
Deployment stacks
Policies
Properties
Locks
Cost Management
Cost analysis
Cost alerts (preview)
Add or remove favorites by pressing Ctrl+Shift+F
Create
Manage view
Delete resource group
Refresh
Export to CSV
Open query
Assign tags
Move
Group by none
Essentials
JSON View
Resources
Recommendations
Filter for any field...
Type equals 2 selected
Location equals all
Add filter
Name
Type
Location
computervision-pe
Private endpoint
Canada Central
privatelink.azurewebsites.net
Private DNS zone
Global
privatelink.blob.core.windows.net
Private DNS zone
Global
privatelink.cognitiveservices.azure.com
Private DNS zone
Global
privatelink.table.core.windows.net
Private DNS zone
Global
securefileupload-pe
Private endpoint
Canada Central
securefileuploadblob-pe
Private endpoint
Canada Central
securefileuploadtest-pe
Private endpoint
Canada Central
securefileuploadtesttable-pe
Private endpoint
Canada Central
Showing 1 - 9 of 9. Display count: auto
Give feedback
25°C
Mostly cloudy
Search
ENG
IN
22:20
26-07-2025

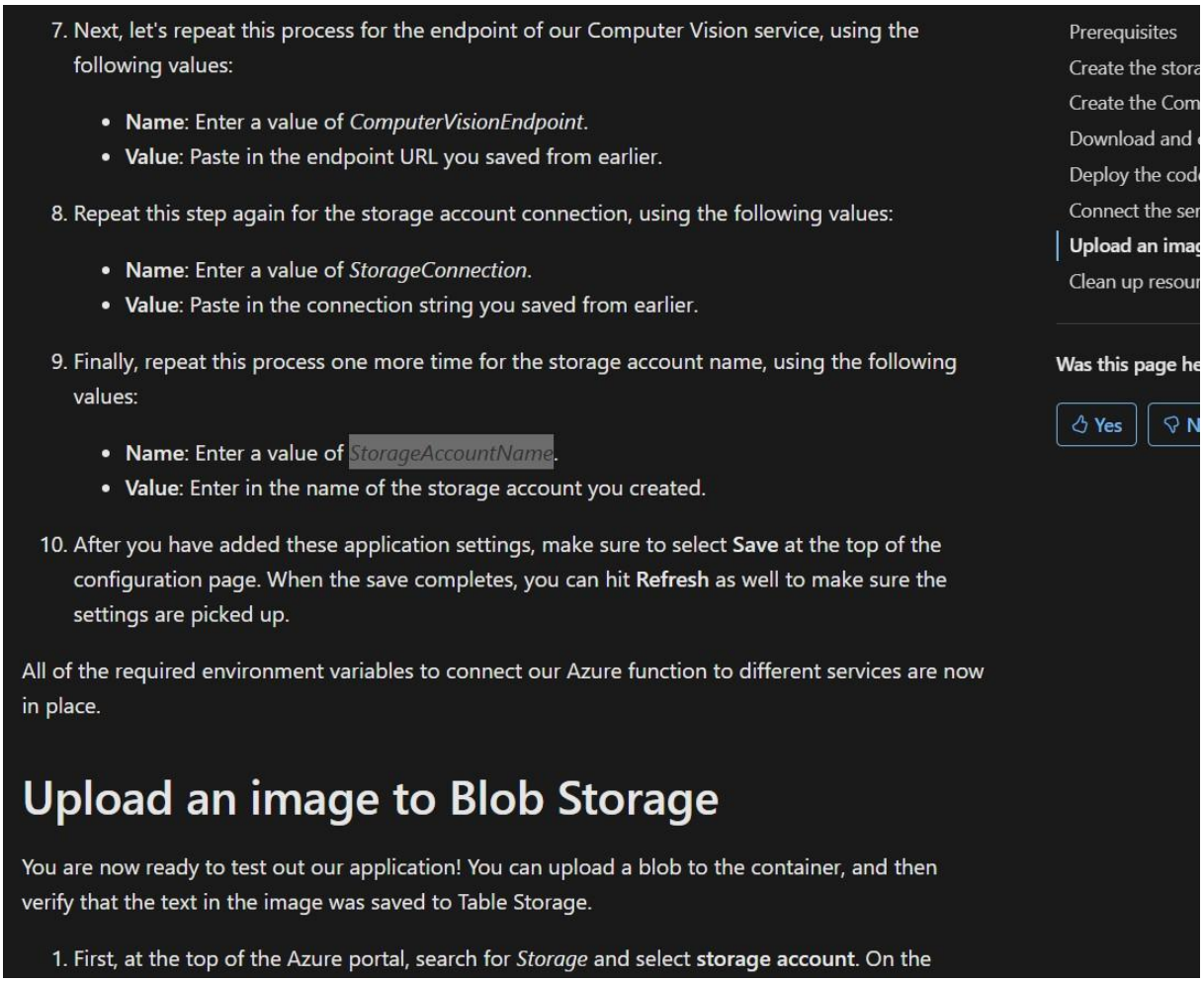

Sign in faster with your face, fing
https://account.live.com/interrupt/passkey?mkt=EN-US&uiflavor=web&client_id=1E00004417ACAE&cobrandid=ed5d1924-9524-4e70-8f68-5ee5
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.7558]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\mohan32702>nslookup securefileupload.cognitiveservices.azure.com
Server: UnKnown
Address: 168.63.129.16
Non-authoritative answer:
Name: securefileupload.privatelink.cognitiveservices.azure.com
Address: 10.0.0.8
Aliases: securefileupload.cognitiveservices.azure.com
C:\Users\mohan32702>
passwords, apps, or codes needed.
Next
Skip for now



Once I have upload the file in web app ui, then the image is stored in Blob container



I have uploaded the below image as input



7. Next, let's repeat this process for the endpoint of our Computer Vision service, using the following values:
   - **Name**: Enter a value of *ComputerVisionEndpoint*.
   - **Value**: Paste in the endpoint URL you saved from earlier.
8. Repeat this step again for the storage account connection, using the following values:
   - **Name**: Enter a value of *StorageConnection*.
   - **Value**: Paste in the connection string you saved from earlier.
9. Finally, repeat this process one more time for the storage account name, using the following values:
   - **Name**: Enter a value of *StorageAccountName*.
   - **Value**: Enter in the name of the storage account you created.
10. After you have added these application settings, make sure to select **Save** at the top of the configuration page. When the save completes, you can hit **Refresh** as well to make sure the settings are picked up.

All of the required environment variables to connect our Azure function to different services are now in place.

Upload an image to Blob Storage

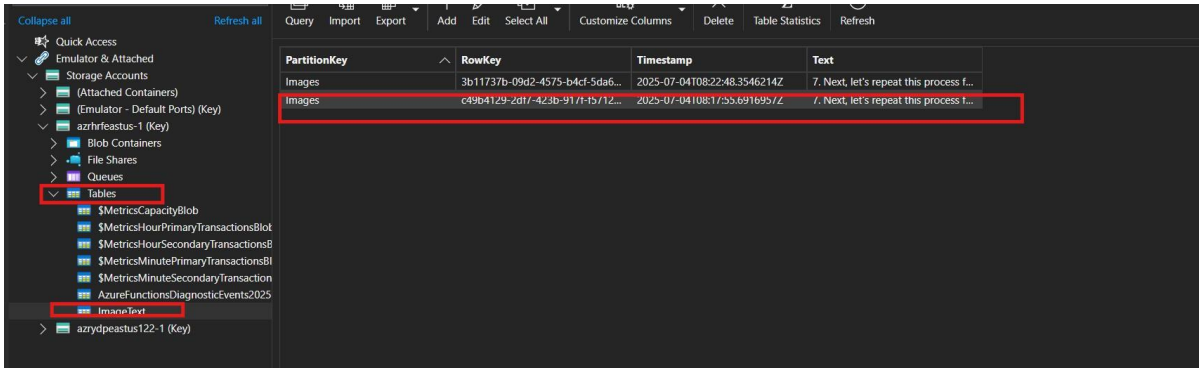
You are now ready to test out our application! You can upload a blob to the container, and then verify that the text in the image was saved to Table Storage.

1. First, at the top of the Azure portal, search for *Storage* and select **storage account**. On the

Prerequisites
Create the stora
Create the Com
Download and
Deploy the cod
Connect the ser
Upload an imag
Clean up resour
Was this page he
Yes

- The Function is triggered automatically via a **Blob Trigger** when an image is uploaded to the container imageanalysis.
- Inside the Function (ProcessImageUpload.cs):
  - Reads environment variables:
    - StorageConnection
    - StorageAccountName
    - ComputerVisionKey
    - ComputerVisionEndpoint
  - Constructs the full image URL
  - Sends it to the Computer Vision API
  - Receives extracted text response
  - Saves it to Azure Table Storage (ImageText)

Get the SAS Token of storage and open MS Storage explorer and table storage you can able to see image is converted to text and stored.



Output text stored in table:
7. Next, let's repeat this process for the endpoint of our Computer Vision service, using thePrerequisitesfollowing values:Create the storName: Enter a value of ComputerVisionEndpoint.Create the ComValue: Paste in the endpoint URL you saved from earlier.Download andDeploy the cod8. Repeat this step again for the storage account connection, using the following values:Connect the se| Upload an ima· Name: Enter a value of
StorageConnection.. Value: Paste in the connection string you saved from earlier.Clean up resou9.
Finally, repeat this process one more time for the storage account name, using the followingWas this page hevalues:Yes· Name: Enter a value of StorageAccountName.Value: Enter in the name of the storage account you created.10. After you have added these application settings, make sure to select Save at the top of theconfiguration page. When the save completes, you can hit Refresh as well to make sure thesettings are picked up.All of the required environment variables to connect our Azure function to different services are nowin place.Upload an image to Blob StorageYou are now ready to test out our application! You can upload a blob to the container, and thenverify that the text in the
image was saved to Table Storage.1. First, at the top of the Azure portal, search for Storage and select storage account. On the