



CLOUD COMPUTING - MICROSOFT AZURE ZEN CLASS - MAIN PROJECT - 1

Name – Abhiram S

Reg. Mail in GUVI – ssabhiram411@gmail.com

Batch - CC2WE-E

Project Scope:

Create a demo using any of the cognitive services and showcase the use of that service. This is similar to what we did in class with compute vision.

Architecture Overview:

1. Creation of Virtual Machine:

- I started by creating a virtual machine (VM) based on a Linux operating system. This VM provided a flexible and isolated environment for your development work.

2. Setting Up Azure AI Translator:

- Within the VM, I aimed to set up a translation service utilizing Azure AI Cognitive Services. The Translator is a component of these services, designed for AI-based translation tasks.

3. Opening the Virtual Machine:

- After the VM was created, I opened it to begin configuring the necessary software and dependencies.

4. Installing Required Packages:

- I executed a series of commands to install essential packages. These included:
 - `sudo apt update`: This command updated the package lists for upgrades and new package installations.
 - `sudo apt install python3-pip`: You installed pip, the Python package manager, to facilitate the installation of additional Python libraries.
 - `sudo apt install python3-requests`: This library was installed to make HTTP requests easier in your Python code.

- 🔧 `sudo apt install python3-pillow`: This library enables image processing capabilities within your Python applications.
- 🔧 `sudo apt install python3-flask`: Flask was installed to help create web applications and APIs.

5. Adding the Azure Translator Endpoint:

- 🔧 In my Python code, I added the endpoint URL for the Azure Translator service. This endpoint is crucial for sending requests to the translation API.

6. Inserting the API Key:

- 🔧 I included the API key in my code, which is necessary for authenticating requests to the Azure Translator service. This key ensures that only authorized applications can access the translation features.

7. Uploading Data to Azure VM:

- 🔧 After setting up the code, I uploaded any necessary data files or configurations to the Azure VM to support the translation function.

8. Executing the Translation Script:

- 🔧 With everything in place, I executed your Python script within the VM. This script called the Azure Translator API, sending a word in English for translation.

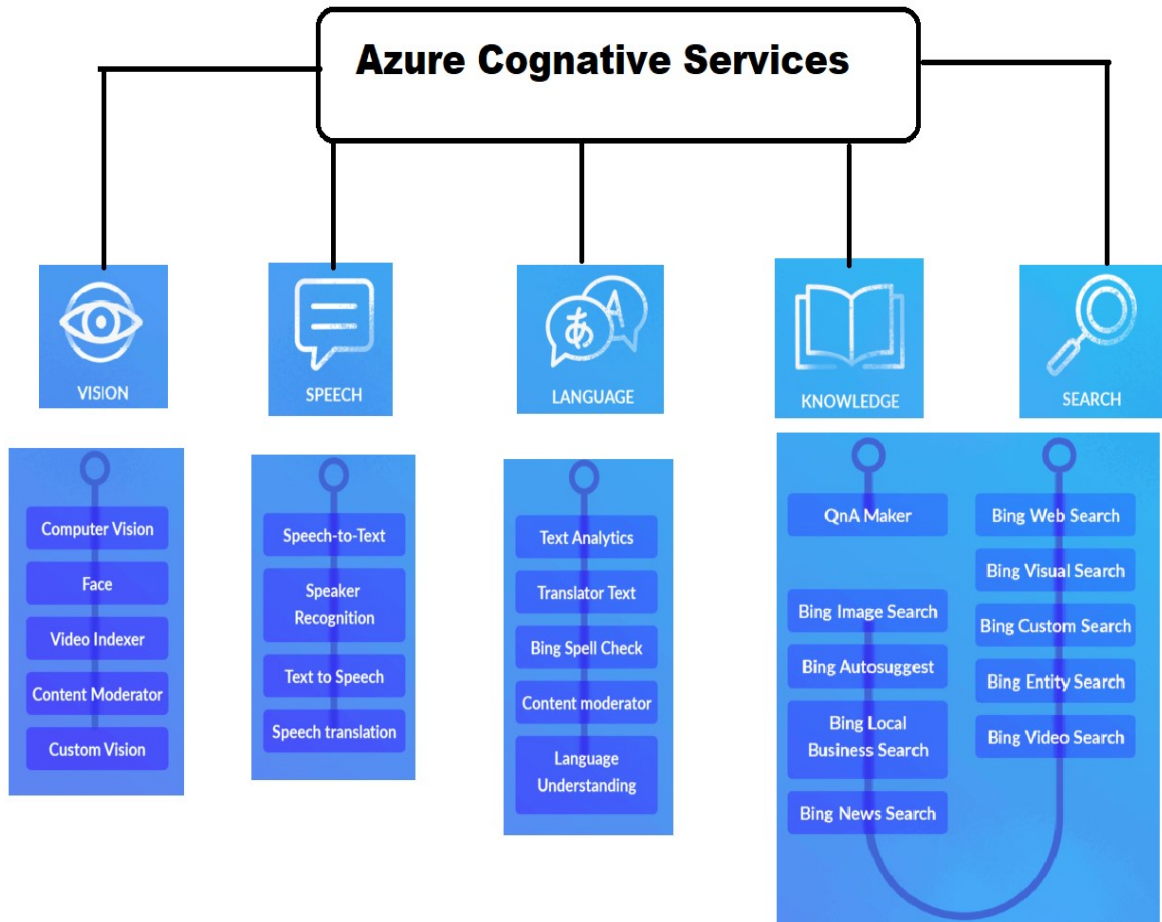
9. Receiving Translation Results:

- 🔧 Upon execution, I successfully received the translated result in Hindi, confirming that the integration between my Python code and Azure Translator was functioning correctly.

10. Conclusion:

- 🔧 This step-by-step process showcases how I effectively utilized a Linux VM and Azure AI Cognitive Services to create a working translation tool that translates English words into Hindi.

Project Architecture:



1.Translator:

The image displays two screenshots from the Microsoft Azure portal. The top screenshot shows the 'Overview' page for a virtual machine named 'myvm1'. The VM is running on Linux (ubuntu 24.04) with a Standard B2s size. It is located in the East US (Zone 1) region. The bottom screenshot shows the 'Overview' page for a Microsoft Cognitive Services Text Translator resource named 'translator411'. The translator is active and located in the East US region. It has a TextTranslation API kind and a Free pricing tier. Below the overview, there is a 'Try it' section with a text input field and a 'Translate' button.

Virtual Machine Details (myvm1):

- Resource group: Abhiram
- Status: Running
- Location: East US (Zone 1)
- Subscription: Azure subscription 1
- Subscription ID: c4498e5c-6513-4872-b5f1-0db5f82228cf
- Availability zone: 1
- Operating system: Linux (ubuntu 24.04)
- Size: Standard B2s (2 vcpus, 4 GiB memory)
- Public IP address: 48.217.81.75
- Virtual network/subnet: myvm1-vnet/default
- DNS name: Not configured
- Health state: -
- Time created: 10/10/2024, 7:44 AM UTC

Text Translator Details (translator411):

- Resource group: Abhiram
- Status: Active
- Location: East US
- Subscription: Azure subscription 1
- Subscription ID: c4498e5c-6513-4872-b5f1-0db5f82228cf
- API Kind: TextTranslation
- Pricing tier: Free
- Endpoints: [Click here to view endpoints](#)
- Manage keys: [Click here to manage keys](#)

Try it Section:

Translate text to another language [Learn more about Translator](#)

This demo runs against your resource and will incur usage against your account.

From: Auto detect To: Spanish

View request View response

- ✚ Create a Virtual Machine with Ubuntu software
- ✚ Then created the Microsoft Cognitive Service Text Translator
- ✚ The created a code to access the translator in the virtual machine
- ✚ In that code I had added the key value and the endpoint
- ✚ After adding I had ssh to the Virtual Machine
- ✚ And then installed the software like `sudo apt update`
- ✚ `sudo apt install python3-pip`, `sudo apt install python3-requests`, `sudo apt install python3-pillow`, `sudo apt install python3-flask`

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

Home > Microsoft.CognitiveServicesTextTranslation-20241010131624 | Overview > translator411

translator411 | Keys and Endpoint

Translator

Search

Regenerate Key1 Regenerate Key2

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource Management

Keys and Endpoint

Encryption

Pricing tier

Networking

Identity

Cost analysis

Properties

Locks

Security

Monitoring

Automation

Help

These keys are used to access your Azure AI services API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1

KEY 2

Location/Region

eastus

Web API Containers

Use the below endpoints while using the Web API. To force the request to be handled by a specific geography, see [here](#).

Text Translation

Document Translation

https://api.cognitive.microsofttranslator.com/

https://translator411.cognitiveservices.azure.com/

```

azureuser@myvm1: $ nano translator-app.py
azureuser@myvm1: $ cat translator-app.py
import requests, uuid, json

# Add your key and endpoint
key = "3889d925a1a4bdfaz2eb4dd628f4f31"
endpoint = "https://api.cognitive.microsofttranslator.com/"

# location, also known as region.
# required if you're using a multi-service or regional (not global) resource. It can be found in the Azure portal on the Keys and Endpoint page.
location = "eastus"

path = '/translate'
constructed_url = endpoint + path

params = {
    'api-version': '3.0',
    'from': 'en',
    'to': ['fr', 'zu']
}

headers = {
    'Ocp-Apim-Subscription-Key': key,
    # location required if you're using a multi-service or regional (not global) resource.
    'Ocp-Apim-Subscription-Region': location,
    'Content-type': 'application/json',
    'X-ClientTraceId': str(uuid.uuid4())
}

# You can pass more than one object in body.
body = [
    {
        'text': 'Full fathom five thy father lies, of his bones are coral made. Those are pearls that were his eyes. Nothing of him that doth fade, but doth suffer a sea-change into something rich and strange'
    }
]

request = requests.post(constructed_url, params=params, headers=headers, json=body)
response = request.json()

print(json.dumps(response, sort_keys=True, ensure_ascii=False, indent=4, separators=(',', ' ')))

```

```

azureuser@myvm1: $ python3 translator-app.py
[
  {
    "translations": [
      {
        "text": "Sur cinq brasses de ton père ment, ses os sont faits de corail. Ce sont des perles qui étaient ses yeux. Rien de celui qui ne se fane, mais subit un changement radical en quelque chose de riche et d'étrange",
        "to": "fr"
      },
      {
        "text": "I-fathom ephelele ezinhlanu uyihlo amanga, amathambo akhe enziwe amakhorali. Lawo ngamaparele ayengamehlo akhe. Akukho lutho kuye oluphelayo, kodwa oluhlupheka ngokushintsha kolwandle lube yinto ecebile futhi engavamile",
        "to": "zu"
      }
    ]
  }
]

```

- After installing I had created a nono translator-app.py folder
- Then added the code into the folder also the text is included in it
- Then viewed by using cat translator-app.py
- To run that I used the code pythin3 translator-app.py

2. Image to text

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

Home > Azure AI services | Computer vision >

Create Computer Vision

Project Details

Subscription * ⓘ Azure subscription 1

Resource group * ⓘ (New) AbhiAI
[Create new](#)

Instance Details

Region ⓘ Central India

Name * ⓘ MyAIcomputer

Pricing tier * ⓘ Free F0 (20 Calls per minute, 5K Calls per month)

[View full pricing details](#)

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Azure AI service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it. This Azure AI services is intended to process Customer Data that includes Biometric Data (as may be further described in product documentation) that Customer may incorporate into its own systems used for personal identification or other purposes. Customer acknowledges and agrees that it is responsible for complying with the Biometric Data obligations contained in the Online Services DPA.

[Online Services DPA](#)

[Responsible Use of AI documentation for Spatial Analysis](#)

By checking this box I certify that I have ☒

Previous

Next

Review + create

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

Home > Microsoft.CognitiveServicesComputerVision-20241007215154 | Overview >

MyAIcomputer

Computer vision

Search

Delete

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource Management

Security

Monitoring

Automation

Help

Essentials

Resource group (move) : AbhiAI

Status : Active

Location : Central India

Subscription (move) : Azure subscription 1

Subscription ID : c4498e5c-6513-4872-b5f1-0db5f82228cf

Tags (edit) : Add tags

API Kind : ComputerVision

Pricing tier : Free

Endpoint : <https://myaicomputer.cognitiveservices.azure.com/>

Manage keys : [Click here to manage keys](#)

Get Started

Get started with your resource in Vision Studio

Try out all Computer Vision features and build your own custom models

Go to Vision Studio

Keys and endpoint

These keys are used to access your Azure AI services API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

Microsoft Azure

Search resources, services, and docs (G+/I) Copilot

Home > CreateVm-canonical.ubuntu-24_04-lts-server-20241007220157 | Overview >

vm1 Virtual machine

Search Connect Start Restart Stop Hibernate Capture Delete Refresh Open in mobile Feedback CLI / PS

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Connect
- Networking
- Settings
- Availability + scale
- Security
- Backup + disaster recovery
- Operations
- Monitoring
- Automation
- Help

Essentials

Resource group (move) : [AbhiAI](#)

Status : Running

Location : Central India (Zone 1)

Subscription (move) : [Azure subscription 1](#)

Subscription ID : c4d98e5c-6513-4872-b5f1-0db5f82228cf

Availability zone : 1

Tags (edit) : [Add tags](#)

Operating system : Linux (ubuntu 24.04)

Size : Standard B2s (2 vcpus, 4 GiB memory)

Public IP address : [98.70.73.52](#)

Virtual network/subnet : [vm1-vnet/default](#)

DNS name : [Not configured](#)

Health state : -

Time created : 10/7/2024, 4:34 PM UTC

Properties Monitoring Capabilities (7) Recommendations Tutorials

Virtual machine

Computer name : vm1

Operating system : Linux (ubuntu 24.04)

VM generation : V2

VM architecture : x64

Agent status : Ready

Agent version : 2.11.1.12

Hibernation : Disabled

Host group : -

Host : -

Networking

Public IP address : [98.70.73.52](#) (Network interface vm1116_21)

Public IP address (IPv6) : -

Private IP address : 10.0.0.4

Private IP address (IPv6) : -

Virtual network/subnet : [vm1-vnet/default](#)

DNS name : [Configure](#)

Size

Size : Standard B2s

Created

```

$ ssh azureuser@vm1.
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\sabhi> ssh azureuser@98.70.73.52
The authenticity of host '98.70.73.52 (98.70.73.52)' can't be established.
ECDSA key fingerprint is SHA256:006YXsQp1E/RIoHV8nMI3XUM00gxvqWx2IZyH4ILJ8s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '98.70.73.52' (ECDSA) to the list of known hosts.
azureuser@98.70.73.52's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1015-azure x86_64)

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

System information as of Mon Oct  7 16:44:51 UTC 2024

System load:  0.0      Processes:    117
Usage of /:   5.0% of 28.02GB Users logged in: 0
Memory usage: 7%      IPv4 address for eth0: 10.0.0.4
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1015-azure x86_64)

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

```

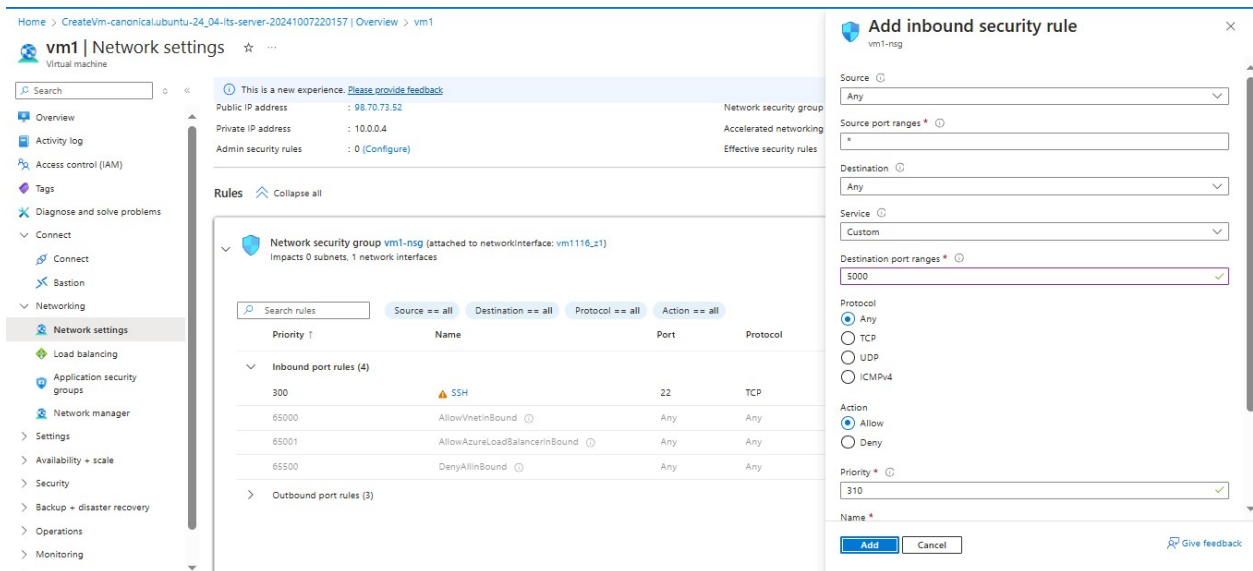
```

sudo apt update
sudo apt install python3-pip
sudo apt install python3-requests
sudo apt install python3-pillow
sudo apt install python3-flask

sudo iptables -A INPUT -p tcp --dport 5000 -j ACCEPT

```

Code to install



The screenshot shows the Azure portal interface for a virtual machine named 'vm1'. The 'Network settings' section is active, displaying the 'Network security group vm1-nsg'. A dialog box titled 'Add inbound security rule' is open, showing the configuration for a new rule. The rule is named 'vm1-nsg', has a source of 'Any', a destination of 'Any', and a service of 'Custom'. The destination port ranges are set to '5000'. The protocol is set to 'Any', and the action is 'Allow'. The priority is set to '310'. The 'Add' button is visible at the bottom of the dialog.

Add port rule in vm port range as 5000

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
azureuser@vm1: $ sudo iptables -A INPUT -p tcp --dport 5000 -j ACCEPT
azureuser@vm1: $ nano app.py
azureuser@vm1: $ ls
app.py
azureuser@vm1: $ mkdir templates
azureuser@vm1: $ ls
app.py  templates
azureuser@vm1: $ cd templates/
azureuser@vm1: ~/templates $

```



```

azureuser@myvm1:~$ ls
app.py
azureuser@myvm1:~$ mkdir templates
azureuser@myvm1:~$ ls
app.py  templates
azureuser@myvm1:~$ cd templates/
azureuser@myvm1:~/templates$ nano upload.html
azureuser@myvm1:~/templates$ nano result.html
azureuser@myvm1:~/templates$ nano error.html
azureuser@myvm1:~/templates$ ls
error.html  result.html  upload.html
azureuser@myvm1:~/templates$ cat upload.html
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Upload Image</title>
</head>
<body>
  <h1>Upload an Image</h1>
  <form method="post" enctype="multipart/form-data">
    <input type="file" name="file">
    <input type="submit" value="Upload">
  </form>
</body>
</html>
azureuser@myvm1:~/templates$ cat result.html
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image Analysis Result</title>
</head>
<body>
  <h1>Image Analysis Result</h1>

  <h2>Analysis:</h2>
  <pre>{{ analysis }}</pre>

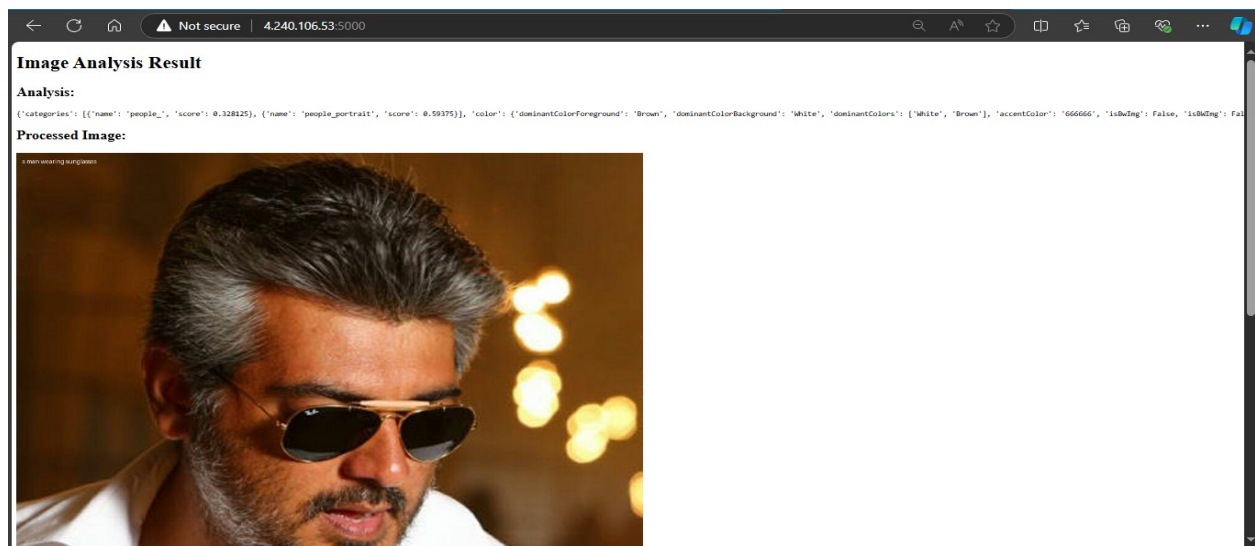
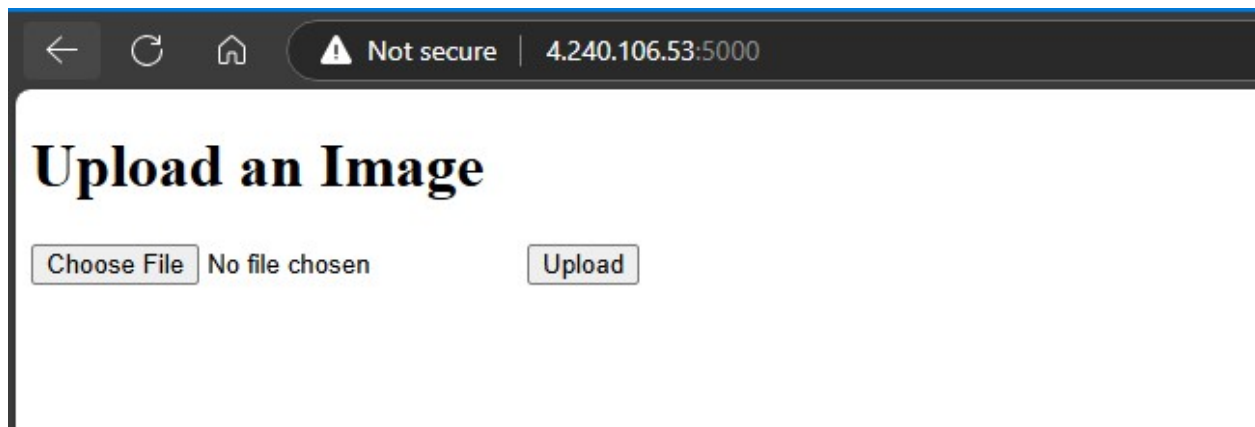
  <h2>Processed Image:</h2>
  
</body>
</html>
azureuser@myvm1:~/templates$

```

```

azureuser@myvm1:~/templates$ cd ..
azureuser@myvm1:~$ ls
app.py  templates
azureuser@myvm1:~$ python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.0.5:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 393-599-125

```



- 📁 Created the Computer Vision account in portal.azure
- 📁 Then created linux disktop and ssh in to it
- 📁 Installed some codes like `sudo apt install python3-pip`, `sudo apt install python3-requests`, `sudo apt install python3-pillow`, `sudo apt install python3-flask`
- 📁 After install added the nano app.py and added the code into it
- 📁 The entered some codes like `mkdir template` , `cd templates/` , `la`
- 📁 Then added the nano value like `update.html`, `result.html`, `error.html`
- 📁 After viwing that by cat option added the codes like `cd ..`, `python3 app.py` to view the details
- 📁 Add port rule in vm port range as 5000
- 📁 Searched that by using http port and ip address and the port number (eg: `http://4.240.106.53:5000`)
- 📁 Then choose the image and upload it the result will bee executed