**Name – Abhiram S**

**Reg. Mail in GUVI –** [**ssabhiram411@gmail.com**](mailto: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.

1. **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.

1. **Opening the Virtual Machine:**

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

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

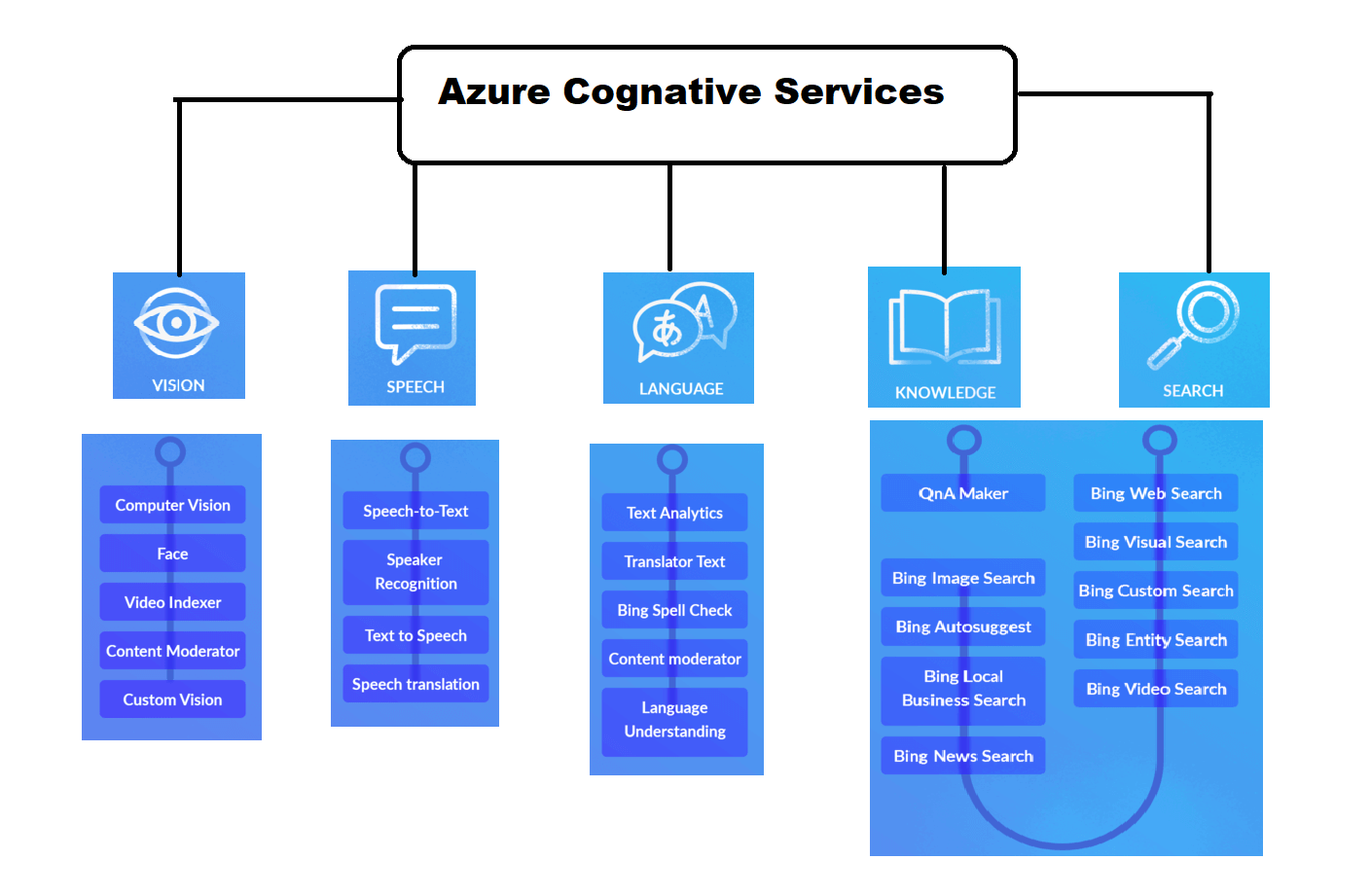
1. **Receiving Translation Results:**

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

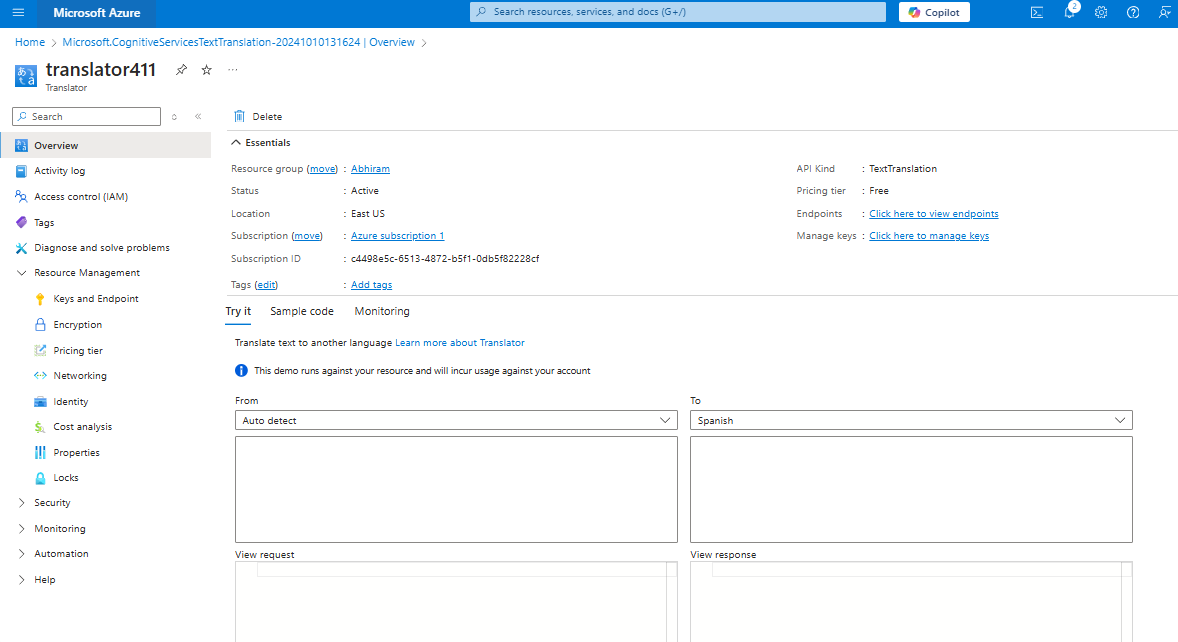
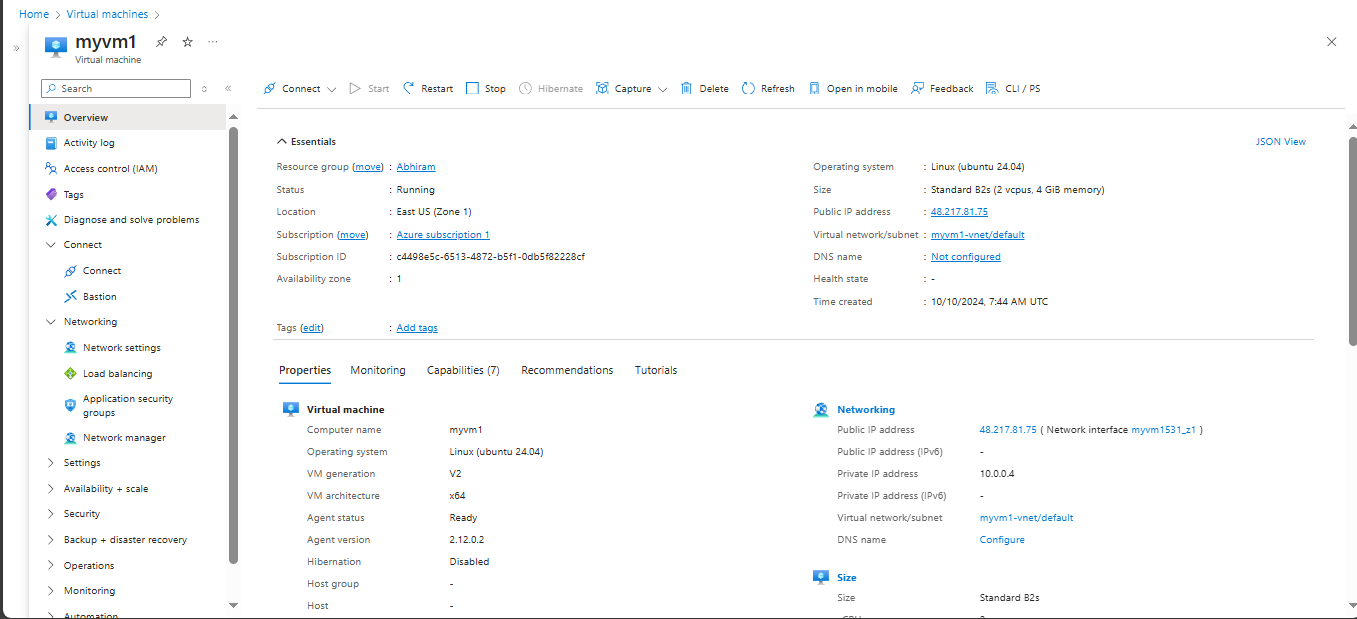
1. **Conclusion:**

* 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.

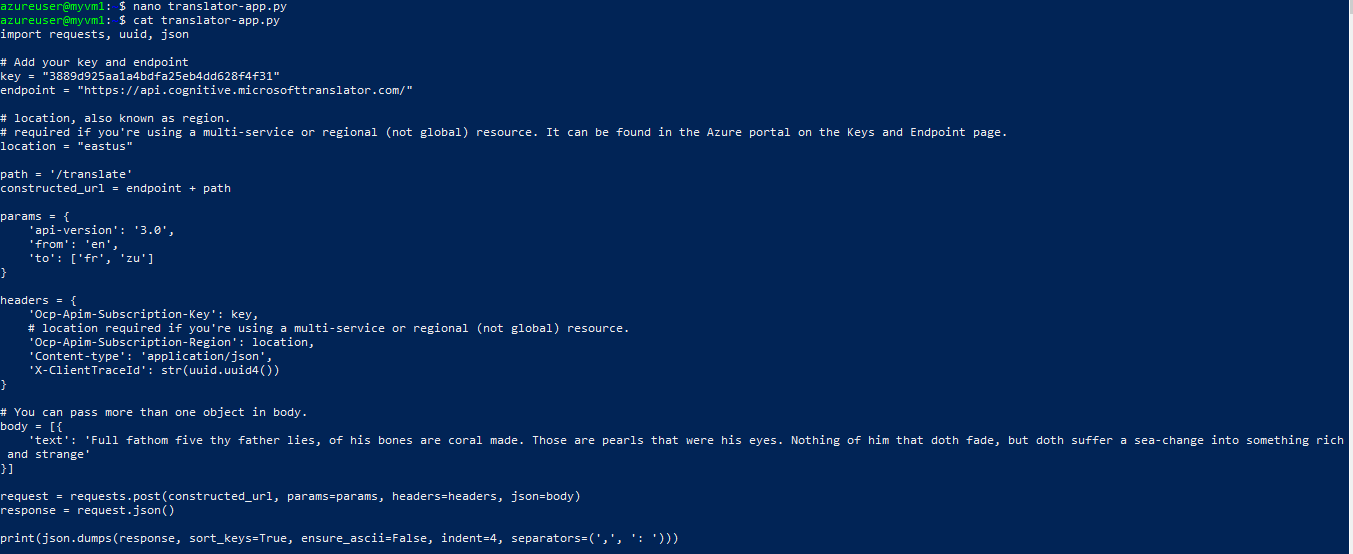
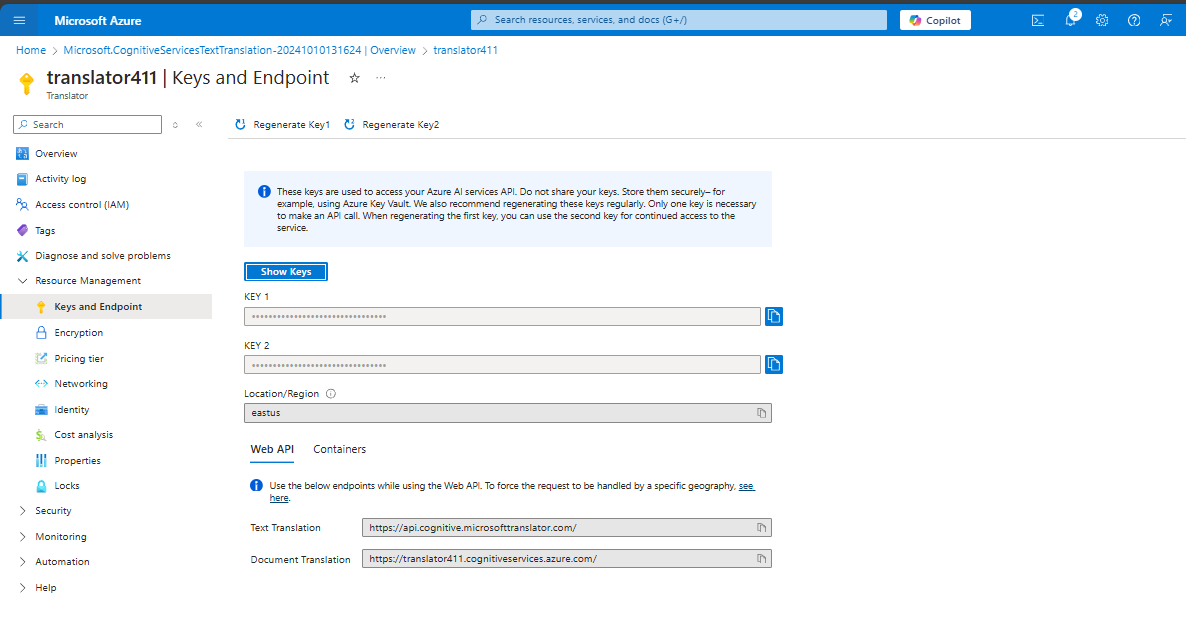
**Project Architecture:**

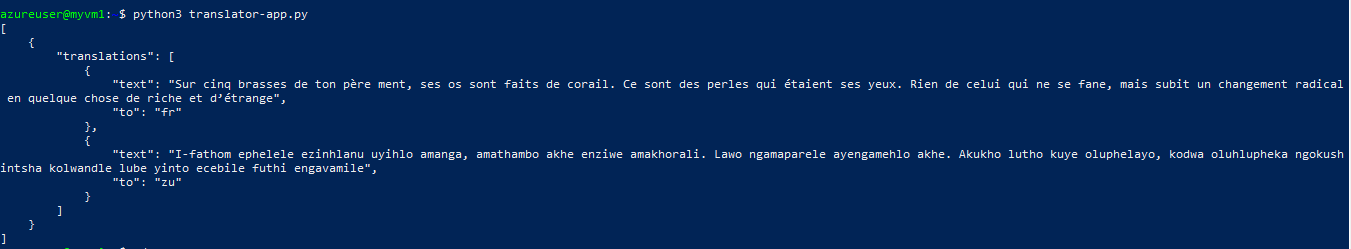
****

**1.Translator:**

****

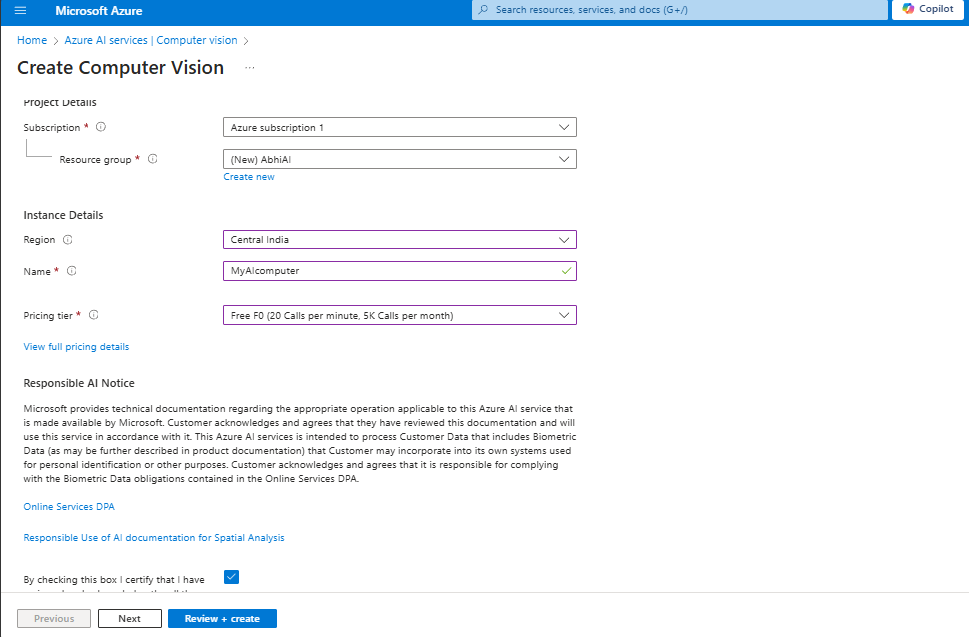
* Create a Vritual Machine with Ubentu software
* Then created the Microsoft Cognative Service Text Translator
* The created a code to acces the translator in the vritual machine
* In that code I had added the key value and the endpoint
* After adding I had ssh to the Vritual Machine
* And then installed 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

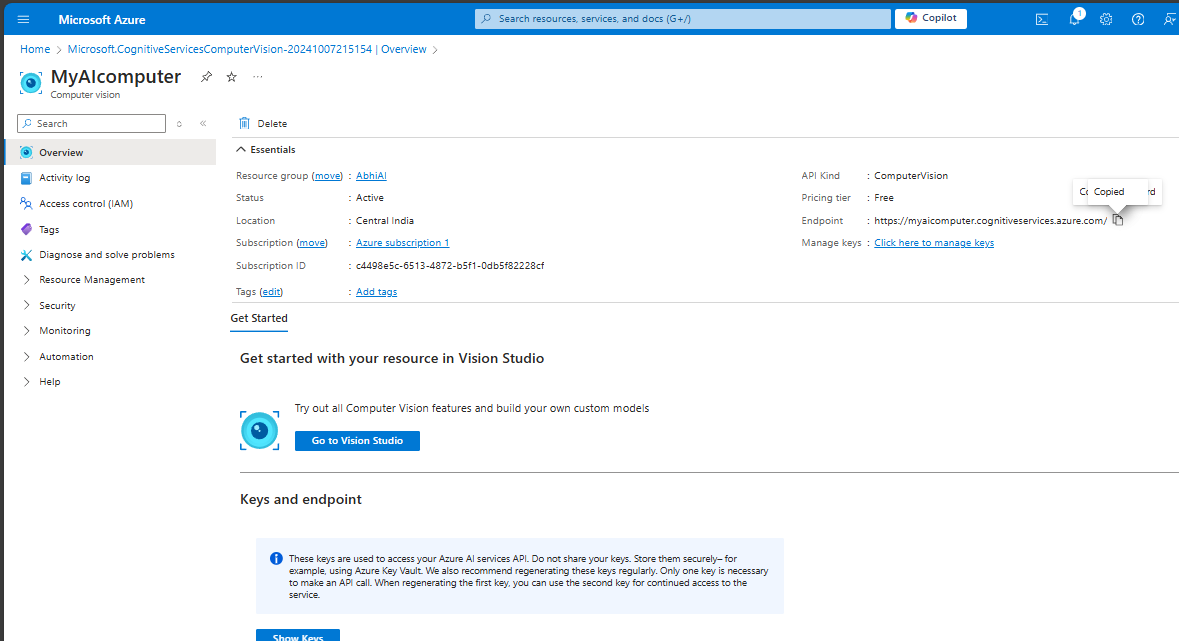
****

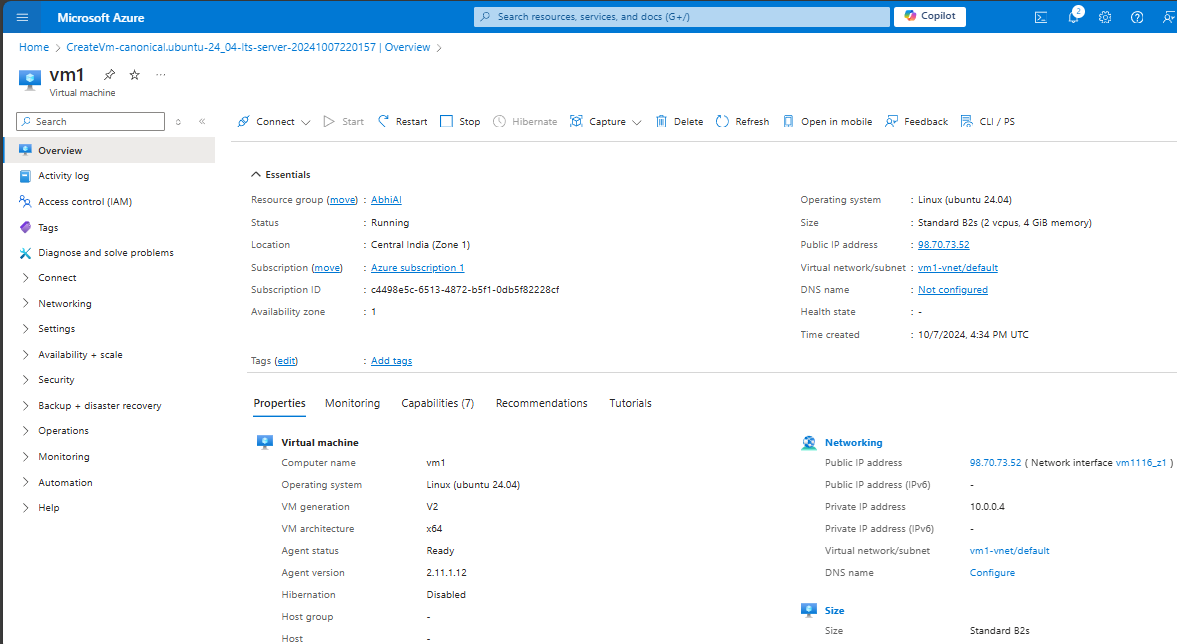
****

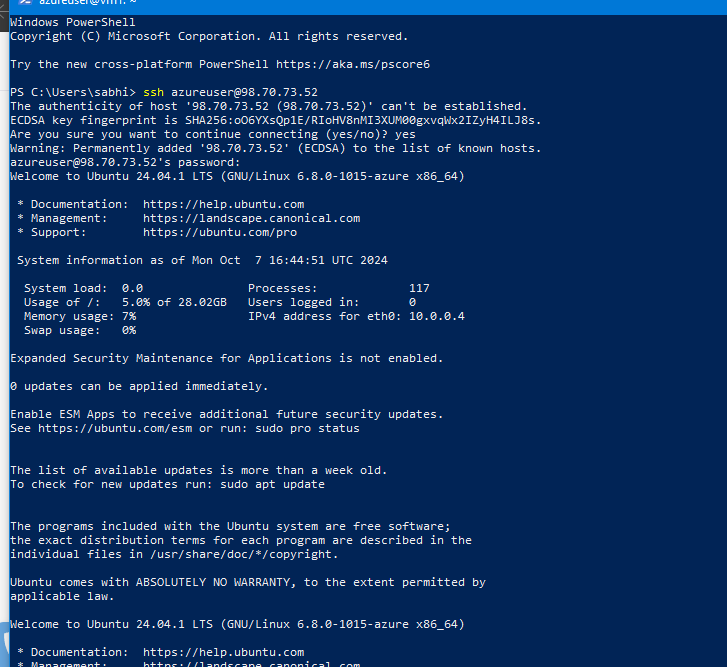
* After insalling I had created a nono translator-app.py folder
* Then added the code into the folder also the text is included in it
* Then viewed by using cat translator-app.py
* To run that I used the code pythin3 translator-app.py

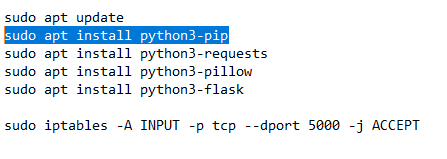
**2.Image to text**

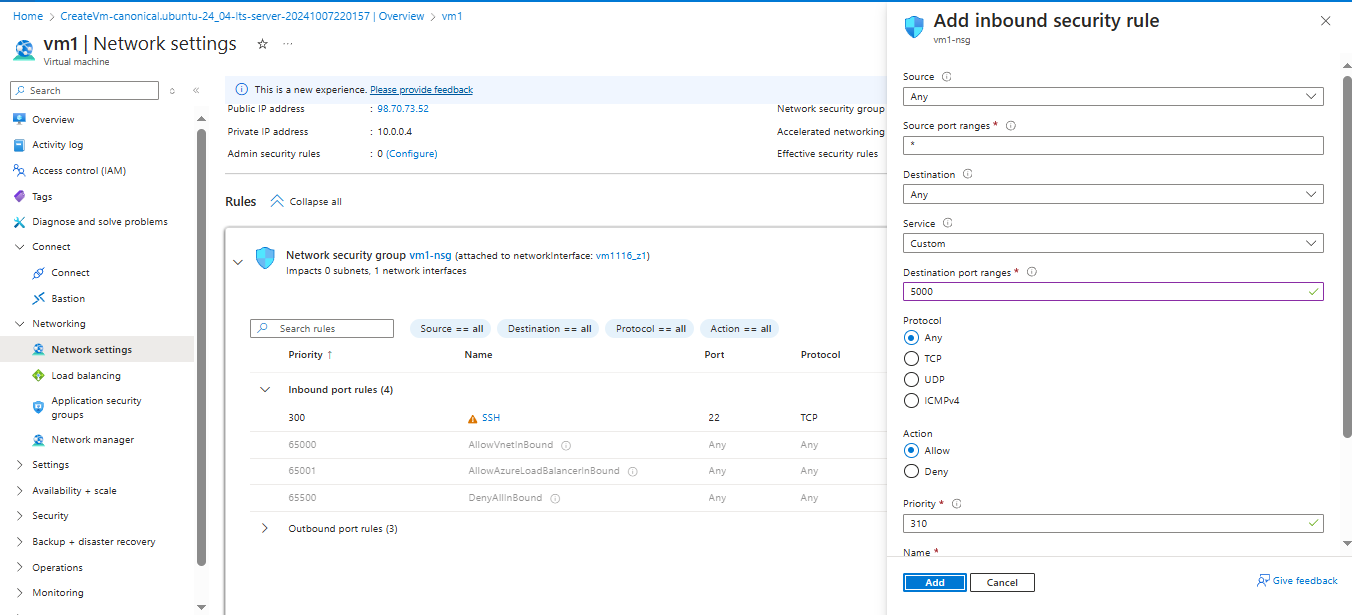
****

****

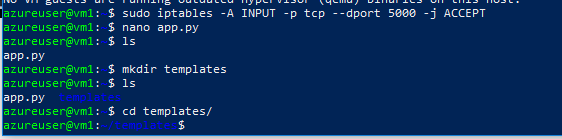
****

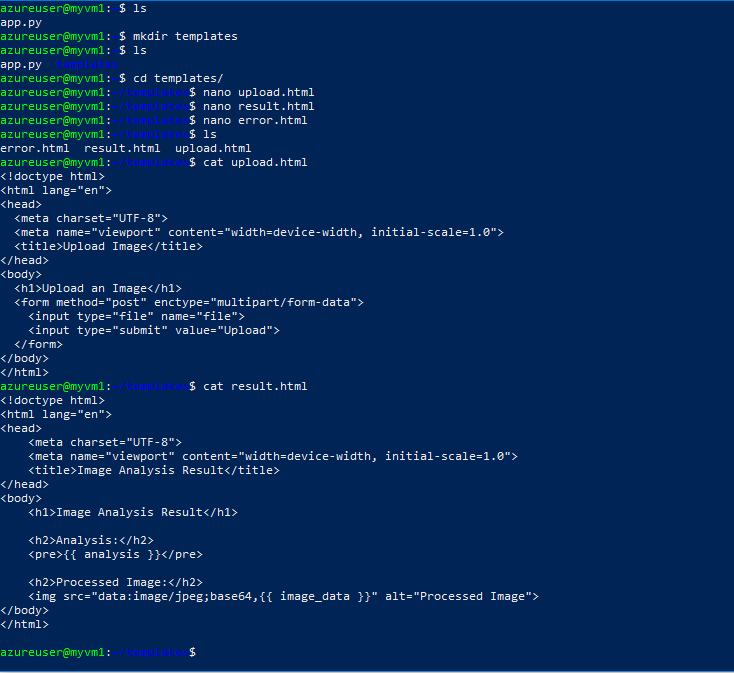
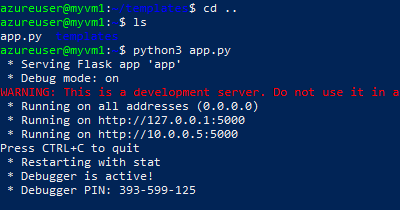
**Created **

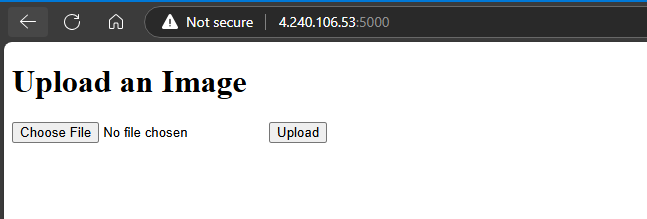
**Code to install**

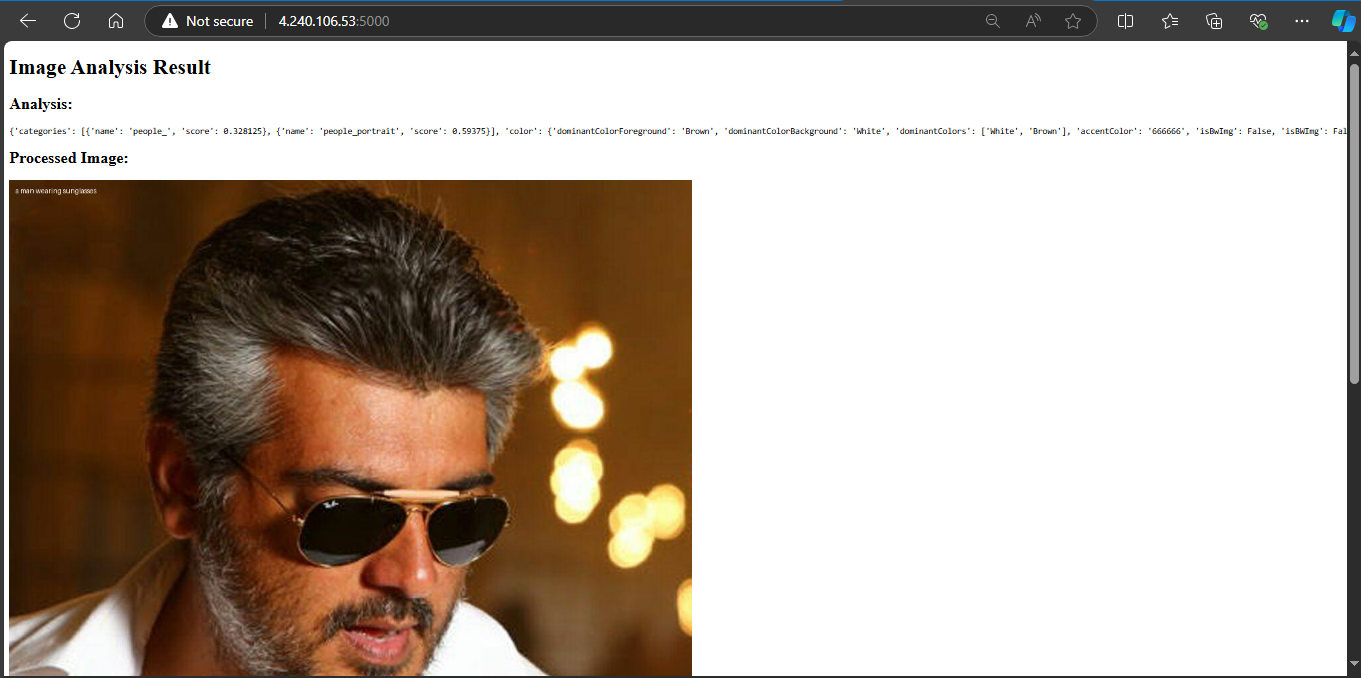
****

**Add port rule in vm port range as 5000**

****

** **

****

****

* 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