

CLOUD COMPUTING - MICROSOFT AZURE

ZEN CLASS - MAIN PROJECT - 1

Name - Punitha Ramesh Keswani
Reg. Mail in GUVI - Punitha.tech@gmail.com
Batch - CC2WE-E

Project Scope

Create a demo using any of the cognitive services and showcase the use of that service:

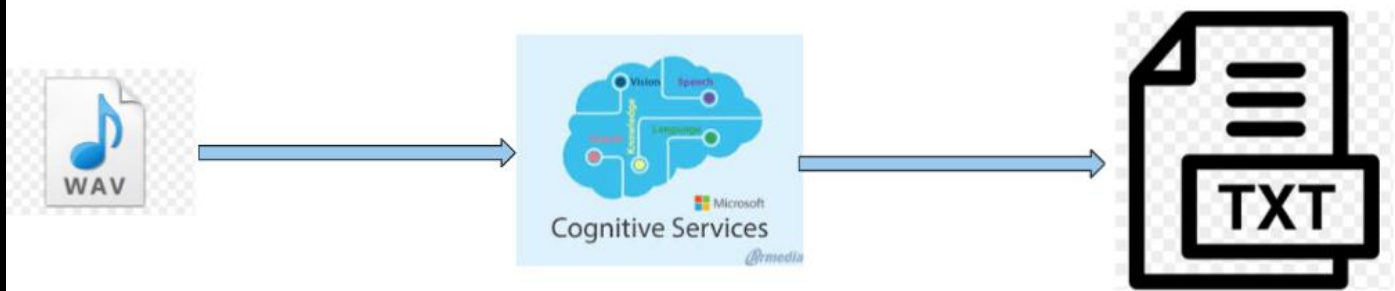
Created Speech service in Azure Portal

Implementation Steps:

1. Create Azure Account: Sign up for an Azure account if you do not have one.
2. Deploy Linux VM: In the Azure portal, create a new Linux virtual machine (e.g., Ubuntu)
3. SSH into VM using putty
4. Install Dependencies:
 - Update the package list: `sudo apt update && sudo apt upgrade`.
 - Install Python and pip: `sudo apt install python3-pip`.
 - Install Azure Speech SDK: Use pip to install the SDK: `pip3 install azure-cognitive-services-speech`.
5. Create Speech Resource: In Azure, create a Speech resource and copy the API key and endpoint.
6. Write Python Script: Create a new file (`speech_to_text.py`) and write code to interact with the Speech service.
7. Upload Audio File: Use `scp` to transfer audio files from your local machine to the VM.
8. Run the Script: Execute the Python script: `python3 speech_to_text.py`.

This streamlined process will help you set up and use Azure Speech-to-Text on a Linux VM efficiently.

Project Architecture



- Login to Azure Portal
- **Create Speech service** in Central US
- Note your Endpoint and Key

Microsoft Azure | Upgrade | Search resources, services, and docs (G+)

Home > Microsoft.CognitiveServicesSpeechServices-20241009160853 | Overview

Project1speech

Speech service

Search | Delete

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
 - Alerts
 - Metrics
 - Diagnostic settings
 - Logs
- Automation
 - Tasks (preview)
 - Export template

Essentials

Resource group (move): [Project1speech](#)

Status: Active

Location: Central US

Subscription (move): [Free Trial](#)

Subscription ID: bd7c576e-80d1-497a-adb5-be2427362fbc

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

API Kind: SpeechServices

Pricing tier: Free

Endpoint: <https://centralus.api.cognitive.microsoft.com/>

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

Get Started

Get started with your resource in Speech Studio

Try out all use cases and see other custom tools for building Speech AI models

[Go to Speech 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](#)

KEY 1

KEY 2

Notifications

More events in the activity log → | Dismiss all

Deployment succeeded

Deployment 'Microsoft.CognitiveServicesSpeechServices-20241009160853' to resource group 'Project1speech' was successful.

[Pin to dashboard](#) | [Go to resource group](#)

a few seconds ago

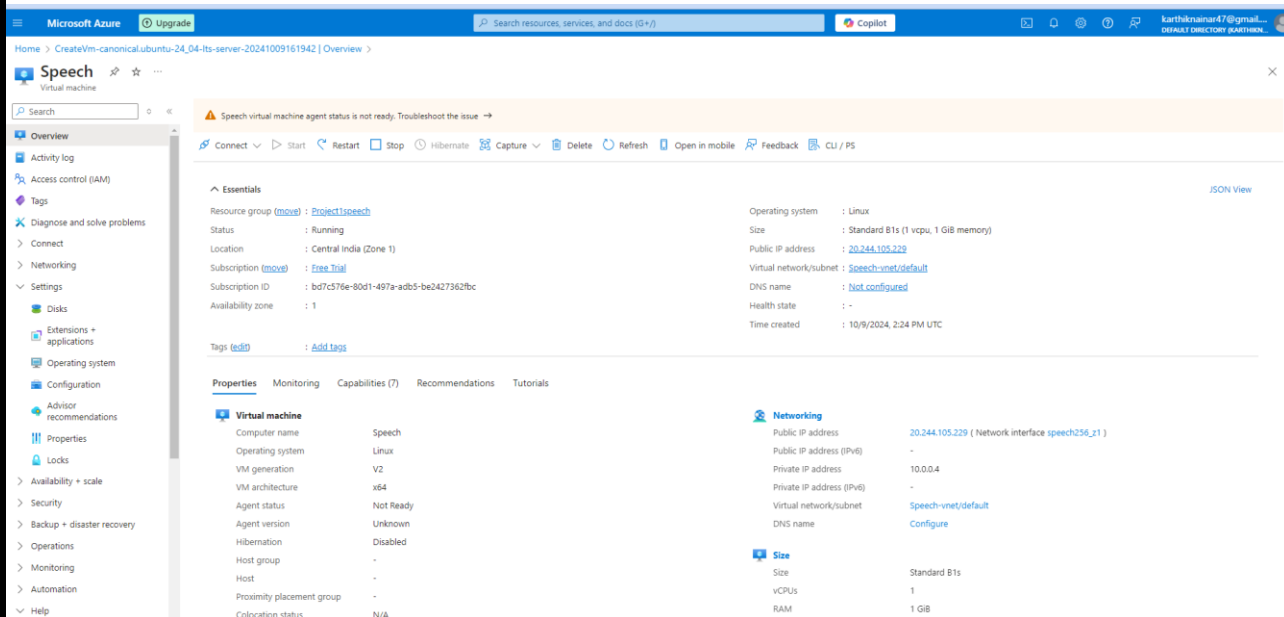
₹16,294.32 credit remaining

Subscription 'Free Trial' has a remaining credit of ₹16,294.32.

Upgrade to a Pay-As-You-Go subscription.

7 minutes ago

■ Create Linux VM



- Ssh linux box through putty
- Create a directory to save your file : Mkdir Punitha
- Cd Punitha
- Copy pcm format wav file from local to remote server
scp "C:\Users\kan\OneDrive - Aalborg Universitet\Personal\Punitha\Guvi\Portfolio Project\AI Service\Titanic.wav" azureuser@20.244.105.229:~/Punitha/
- Optional we can use python script to change .mp3 to .wav by using moviepy library

- .wav file loaded to remote server

```
Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kan>scp "C:\Users\kan\OneDrive - Aalborg Universitet\Personal\Punitha\Guvi\Portfolio Project\AI Service\ ezyZip.pcm" azureuser@20.244.105.229:~/Punitha/
The authenticity of host '20.244.105.229 (20.244.105.229)' can't be established.
ECDSA key fingerprint is SHA256:igLLYmvqiUzYvm0xoqk+ZxVV+EjRZ3oMGe3Bh9e4kTo.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '20.244.105.229' (ECDSA) to the list of known hosts.
azureuser@20.244.105.229's password:
C:/Users/kan/OneDrive - Aalborg Universitet/Personal/Punitha/Guvi/Portfolio Project/AI Service/ ezyZip.pcm: No such file or directory

C:\Users\kan>scp "C:\Users\kan\OneDrive - Aalborg Universitet\Personal\Punitha\Guvi\Portfolio Project\AI Service\ ezyZip.pcm" azureuser@20.244.105.229:~/Punitha/
azureuser@20.244.105.229's password:
ezyZip.pcm
100% 12MB 3.4MB/s 00:03

C:\Users\kan>
```

- Use below commands to check audio is readable or not(Just in Case)
- Validate audio file has data or not
 - sudo apt install sox
 - sox Titanic.wav -n stat

- **Create environment in your remote server to run python file by following below steps**
 - Sudo apt update
 - sudo apt install python3-pip
 - sudo apt-get install python3-venv
 - python3 -m venv venv
 - source venv/bin/activate
 - pip install azure-cognitiveservices-speech
- **Create python code which transfers your .wav to text file**
 - Create file transcribe.py by using vi transcribe.py
 - It includes below code

```
import azure.cognitiveservices.speech as speechsdk
import time
import os

# Set up the Azure Speech configuration
speech_key = "your key"
service_region = "centralus"
speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)

# Set the audio file path
audio_file = "Titanic.wav"

# Check if the audio file exists
if not os.path.isfile(audio_file):
    print(f"Error: The audio file '{audio_file}' does not exist.")
    exit(1)

# Set up the audio configuration
audio_config = speechsdk.audio.AudioConfig(filename=audio_file)

# Create a speech recognizer object
speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_config)

# Create an empty list to store the transcription results
transcriptions = []

# Define an event handler for continuous recognition
def continuous_recognition_handler(evt):
    if evt.result.reason == speechsdk.ResultReason.RecognizedSpeech:
        transcriptions.append(evt.result.text)
    elif evt.result.reason == speechsdk.ResultReason.NoMatch:
        print("No speech could be recognized.")
    elif evt.result.reason == speechsdk.ResultReason.Canceled:
        print(f"Recognition canceled: {evt.result.cancellation_details.reason}")
```

```
# Connect the event handlers
speech_recognizer.recognized.connect(continuous_recognition_handler)

# Start continuous recognition
speech_recognizer.start_continuous_recognition()

# Wait for recognition to complete
done = False # Declare 'done' as a global variable

def stop_recognition(evt):
    global done # Use 'global' to modify the global variable
    print("Recognition session stopped.")
    done = True

speech_recognizer.session_stopped.connect(stop_recognition)

# Start the recognition and wait until session stops
while not done:
    time.sleep(0.5) # This keeps the program alive until the recognition session completes

# Stop continuous recognition
speech_recognizer.stop_continuous_recognition()

# Combine transcriptions into a single string
transcription = ' '.join(transcriptions)

# Write the transcription to a file
output_file = "transcription.txt"
with open(output_file, "w") as file:
    file.write(transcription)

print("Transcription saved to:", output_file)
```

- Execute
-
- `transcribe.py`

```
SyntaxError: no binding for nonlocal 'done' found
(venv) azureuser@Speech:~/Punitha$ ^C
(venv) azureuser@Speech:~/Punitha$ vi file.py
(venv) azureuser@Speech:~/Punitha$ python3 file.py
Recognition session stopped.
Transcription saved to: transcription.txt
(venv) azureuser@Speech:~/Punitha$ cat transcription.txt
Every night in my dreams, I see you. I feel you. That is how I know you. Go. Far across the distance and spaces between. Us you have come to show you. Go on. Wherever you are. I believe that's the heart of. God. One small. You open the
door. Until here in my heart, and my heart will go on. Just one time. And last. For a long time. And never let God see you will come. Love was when I loved you, the one true time I'd hold you in my. Life will always. You are. I believe t
hat's the purpose. Go on. And you'll hear it my heart. And my heart will go well. In the world. There's nothing happy and I know that my heart wins. Gold. We'll stand. For all about this way. You are saving my content. My heart will go
on man. (venv) azureuser@Speech:~/Punitha$
```

Here we can see the txt file and its content of our .wav file.

