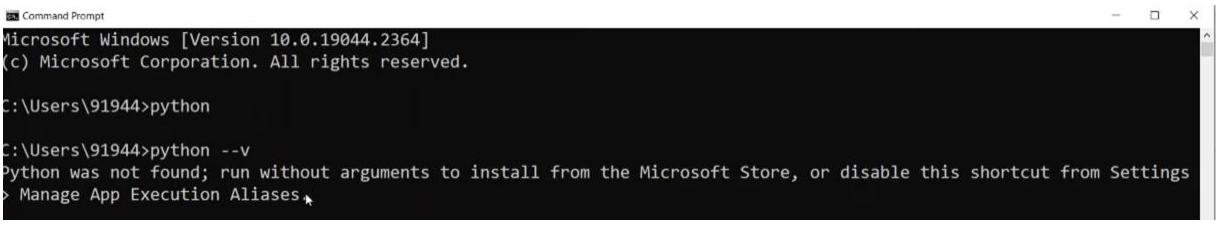
Develop a speech-to-text app using Python & Azure OpenAI

Prerequisites

1. Python

https://www.python.org/downloads/

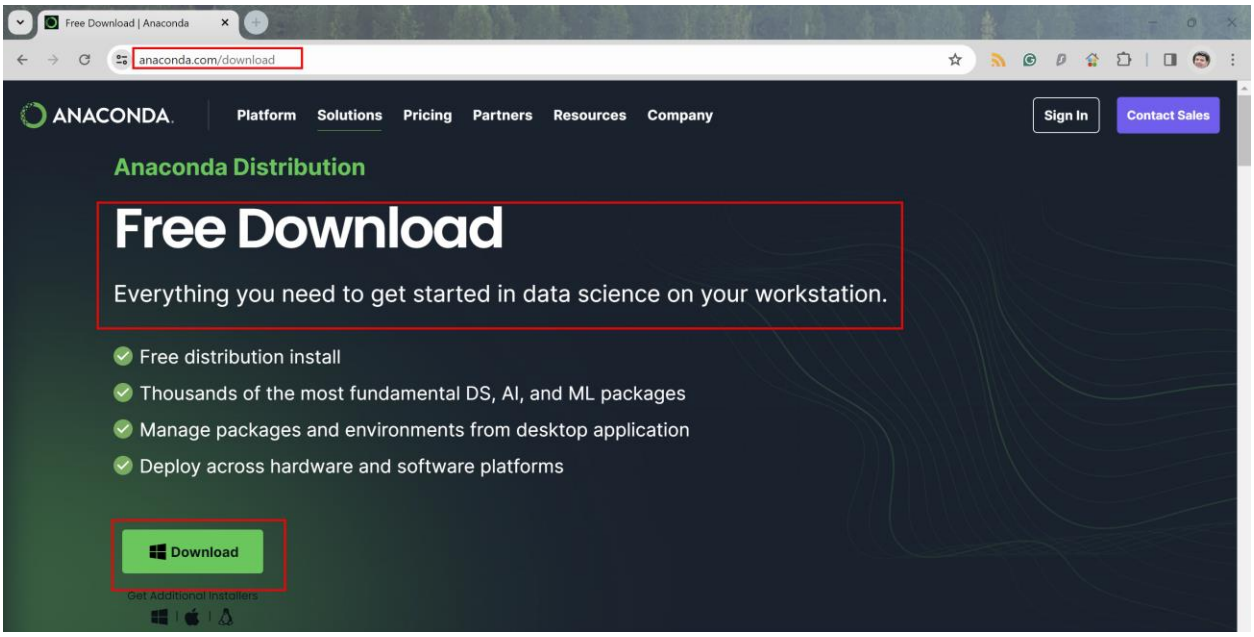
Issue

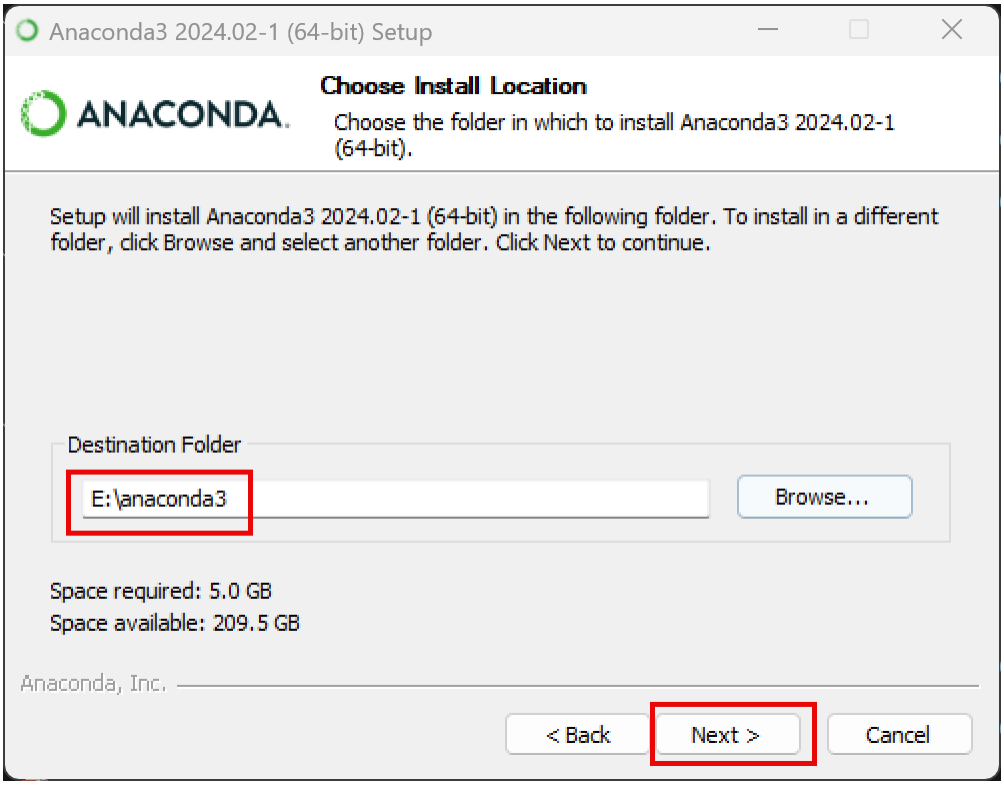


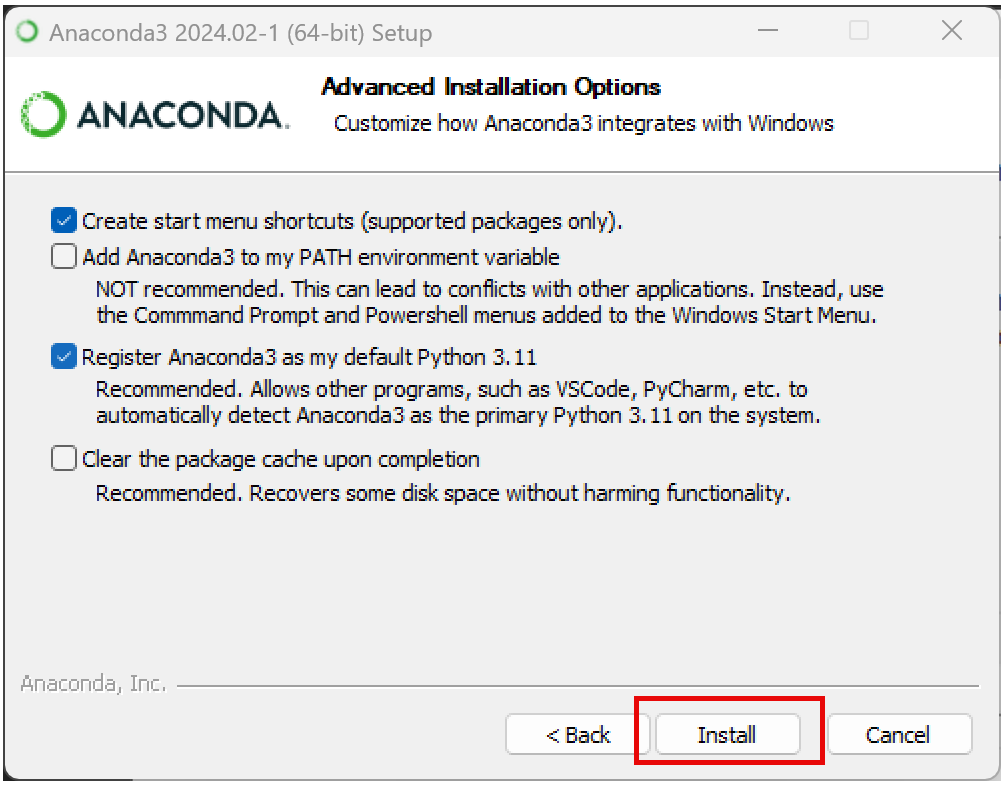
https://www.youtube.com/watch?v=9QrDn_hRSGs

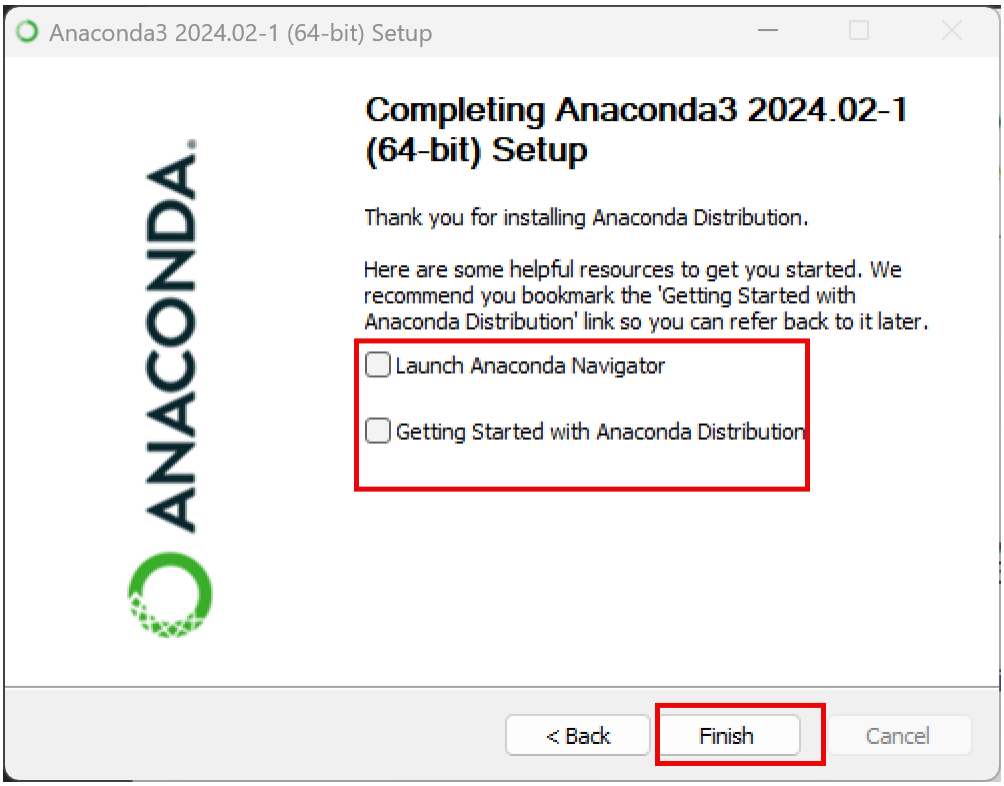
2. Anaconda

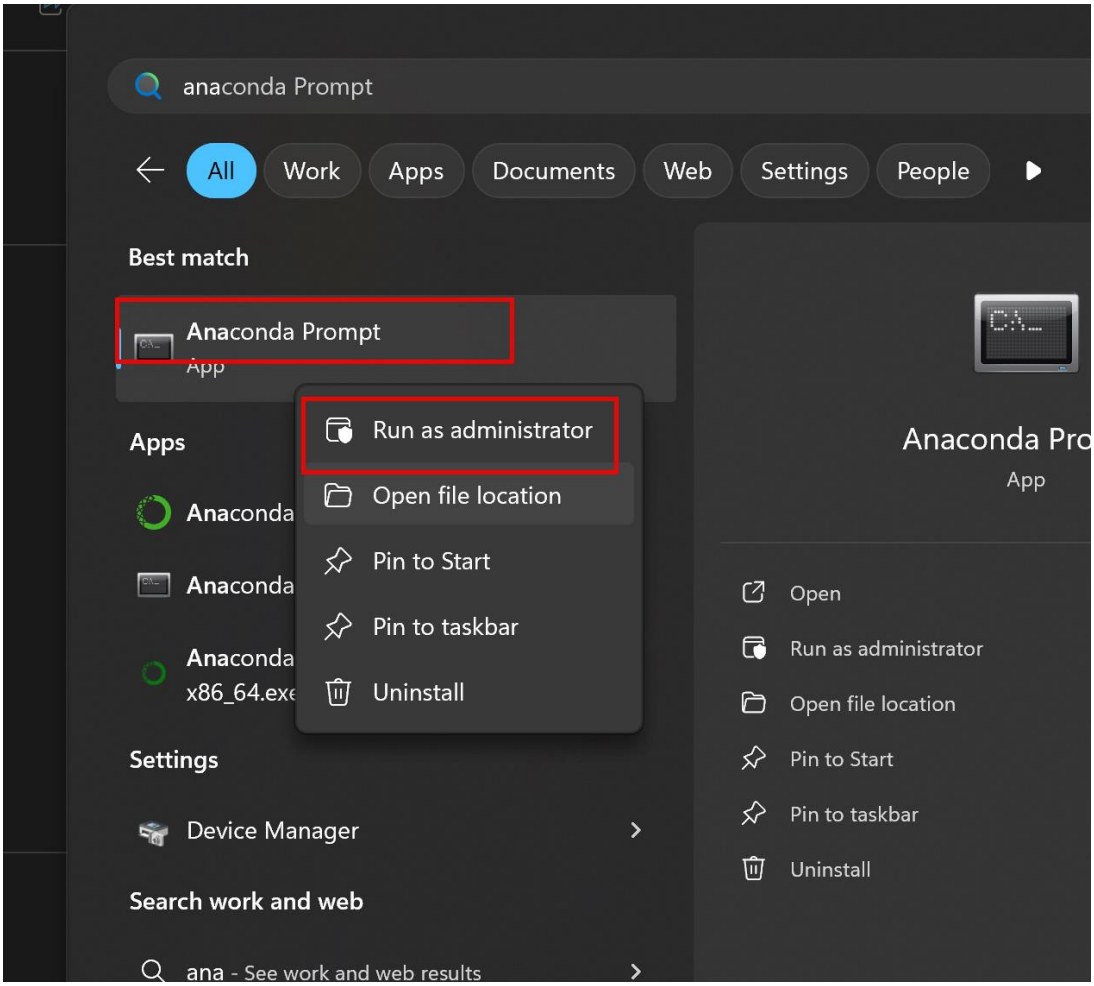
Anaconda simplifies the **process of setting up a Python environment for data science and scientific computing** by providing a pre-configured distribution with commonly used packages and tools.

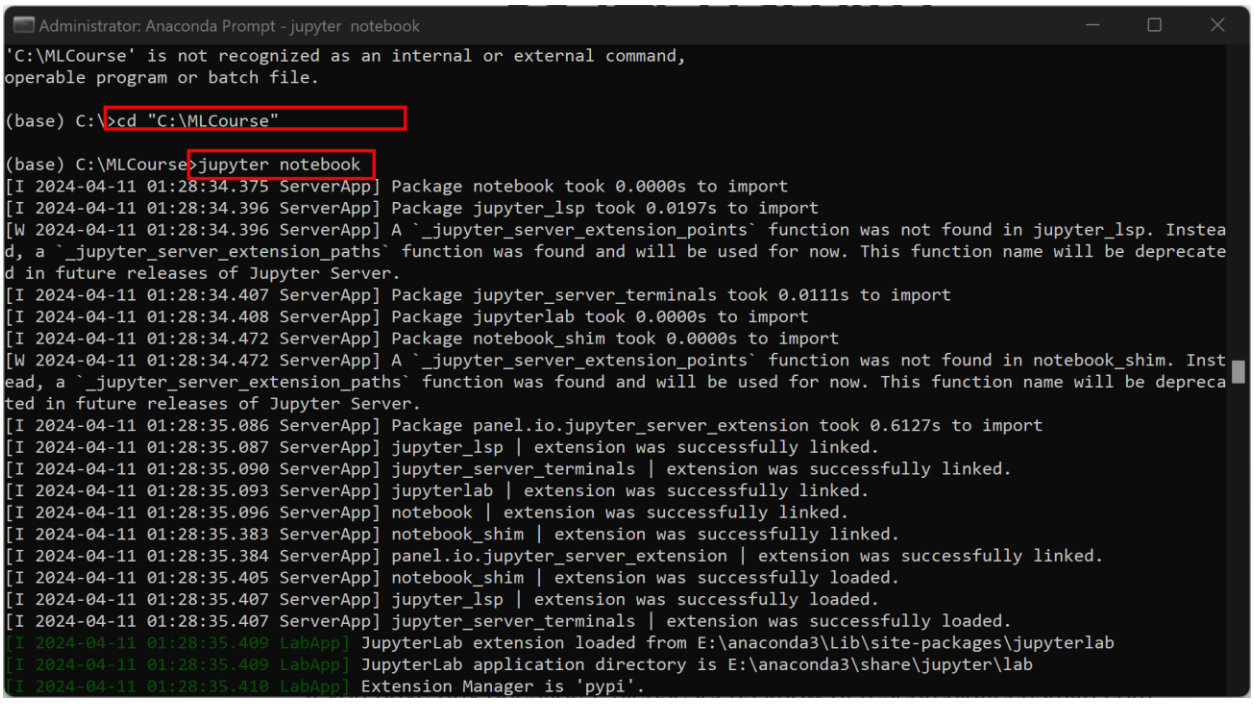

Free Download | Anaconda
anaconda.com/download
ANACONDA.
Platform
Solutions
Pricing
Partners
Resources
Company
Sign In
Contact Sales
Anaconda Distribution
Free Download
Everything you need to get started in data science on your workstation.
Free distribution install
Thousands of the most fundamental DS, AI, and ML packages
Manage packages and environments from desktop application
Deploy across hardware and software platforms
Download
Get Additional Installers


Anaconda3 2024.02-1 (64-bit) Setup
ANACONDA.
Choose Install Location
Choose the folder in which to install Anaconda3 2024.02-1 (64-bit).
Setup will install Anaconda3 2024.02-1 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
E:\anaconda3
Browse...
Space required: 5.0 GB
Space available: 209.5 GB
Anaconda, Inc.
< Back
Next >
Cancel

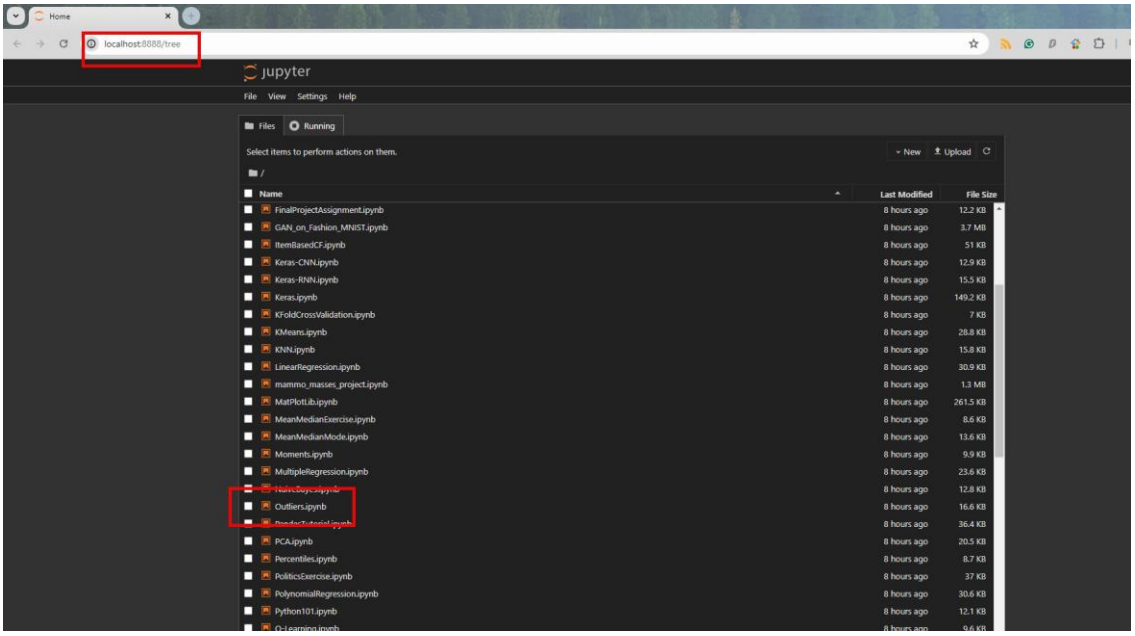

Anaconda3 2024.02-1 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda3 integrates with Windows
Create start menu shortcuts (supported packages only).
Add Anaconda3 to my PATH environment variable
NOT recommended. This can lead to conflicts with other applications. Instead, use the Commmand Prompt and Powershell menus added to the Windows Start Menu.
Register Anaconda3 as my default Python 3.11
Recommended. Allows other programs, such as VSCode, PyCharm, etc. to automatically detect Anaconda3 as the primary Python 3.11 on the system.
Clear the package cache upon completion
Recommended. Recovers some disk space without harming functionality.
Anaconda, Inc.
< Back
Install
Cancel

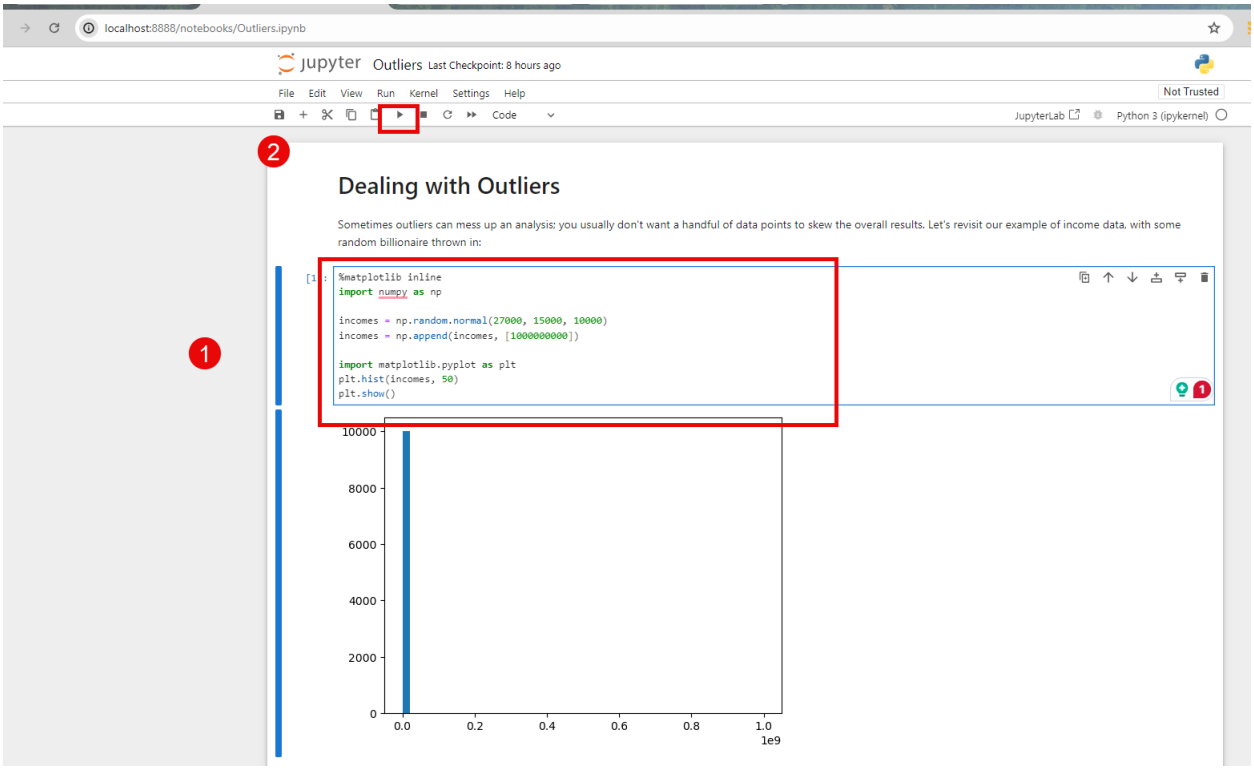

Anaconda3 2024.02-1 (64-bit) Setup
ANACONDA.
Completing Anaconda3 2024.02-1 (64-bit) Setup
Thank you for installing Anaconda Distribution.
Here are some helpful resources to get you started. We recommend you bookmark the 'Getting Started with Anaconda Distribution' link so you can refer back to it later.
Launch Anaconda Navigator
Getting Started with Anaconda Distribution
< Back
Finish
Cancel


anaconda Prompt
All
Work
Apps
Documents
Web
Settings
People
Best match
Anaconda Prompt
App
Apps
Anaconda
Anaconda
Anaconda
x86_64.exe
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall
Settings
Device Manager
Search work and web
ana - See work and web results
Anaconda Pro
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall


Administrator: Anaconda Prompt - jupyter notebook
'C:\MLCourse' is not recognized as an internal or external command,
operable program or batch file.
(base) C:\>cd "C:\MLCourse"
(base) C:\MLCourse>jupyter notebook
[I 2024-04-11 01:28:34.375 ServerApp] Package notebook took 0.0000s to import
[I 2024-04-11 01:28:34.396 ServerApp] Package jupyter_lsp took 0.0197s to import
[W 2024-04-11 01:28:34.396 ServerApp] A `_jupyter_server_extension_points` function was not found in jupyter_lsp. Instead, a `_jupyter_server_extension_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-04-11 01:28:34.407 ServerApp] Package jupyter_server_terminals took 0.0111s to import
[I 2024-04-11 01:28:34.408 ServerApp] Package jupyterlab took 0.0000s to import
[I 2024-04-11 01:28:34.472 ServerApp] Package notebook_shim took 0.0000s to import
[W 2024-04-11 01:28:34.472 ServerApp] A `_jupyter_server_extension_points` function was not found in notebook_shim. Instead, a `_jupyter_server_extension_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-04-11 01:28:35.086 ServerApp] Package panel.io.jupyter_server_extension took 0.6127s to import
[I 2024-04-11 01:28:35.087 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-04-11 01:28:35.090 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-04-11 01:28:35.093 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-04-11 01:28:35.096 ServerApp] notebook | extension was successfully linked.
[I 2024-04-11 01:28:35.383 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-04-11 01:28:35.384 ServerApp] panel.io.jupyter_server_extension | extension was successfully linked.
[I 2024-04-11 01:28:35.405 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-04-11 01:28:35.407 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-04-11 01:28:35.407 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-04-11 01:28:35.409 LabApp] JupyterLab extension loaded from E:\anaconda3\Lib\site-packages\jupyterlab
[I 2024-04-11 01:28:35.409 LabApp] JupyterLab application directory is E:\anaconda3\share\jupyter\lab
[I 2024-04-11 01:28:35.410 LabApp] Extension Manager is 'pypi'.

Jupyter notebook

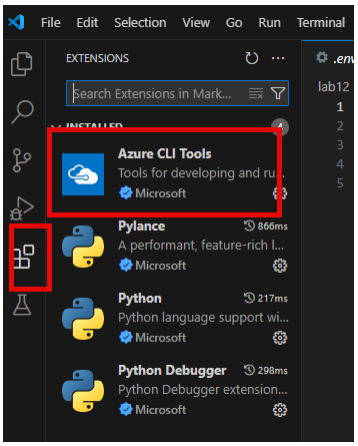




3. VS Code

https://code.visualstudio.com/

Azure CLI tools



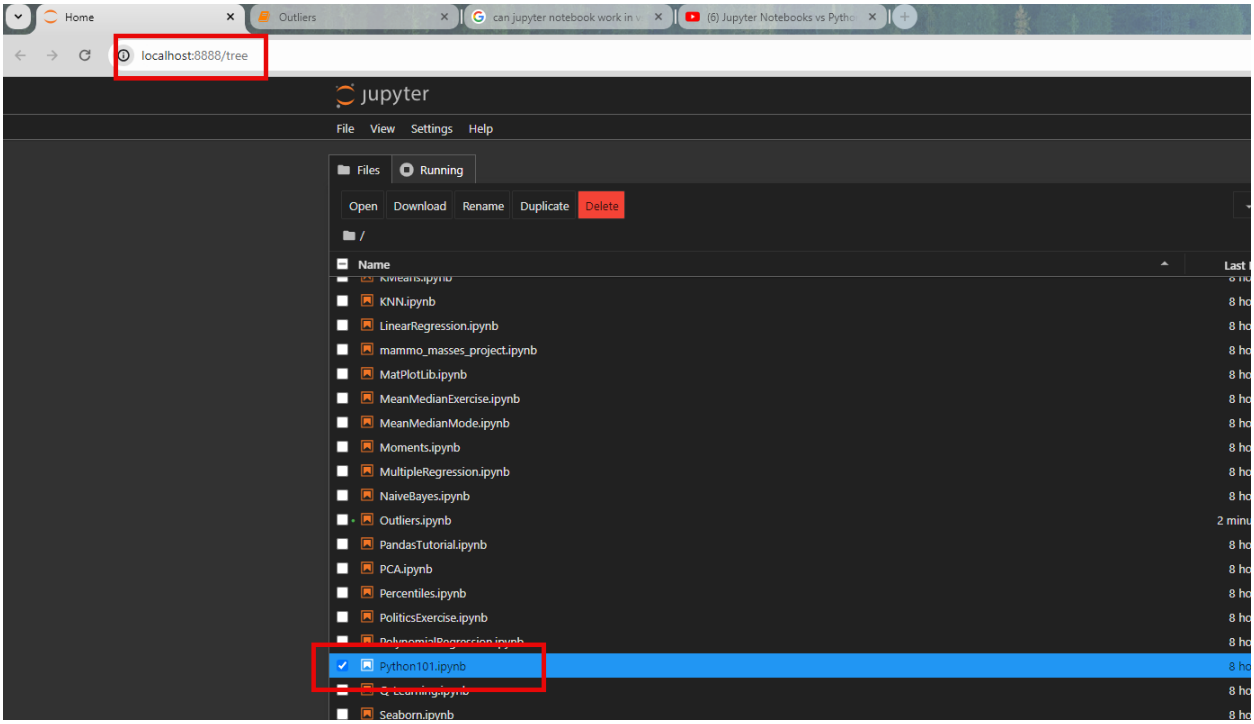
4. Azure OpenAI form

https://customervoice.microsoft.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR7en2Ais5pxKtso_Pz4b1_xUNTZBNzRKNlVQSFhZMU9aV09EVzYxWFdORCQlQCN0PWcu

5. JSON basics

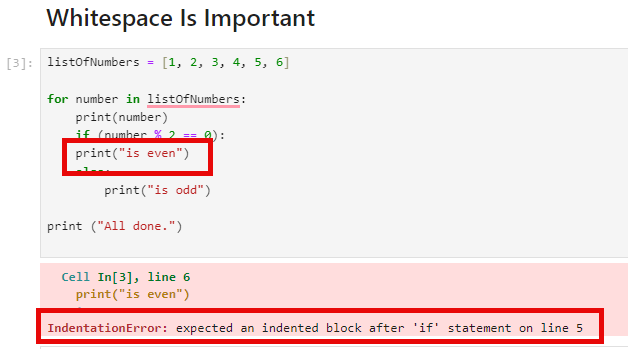
A lot of content on YouTube and other sites

Python 101



Whitespace is important

1. IndentationError
   tabs, spaces to group together

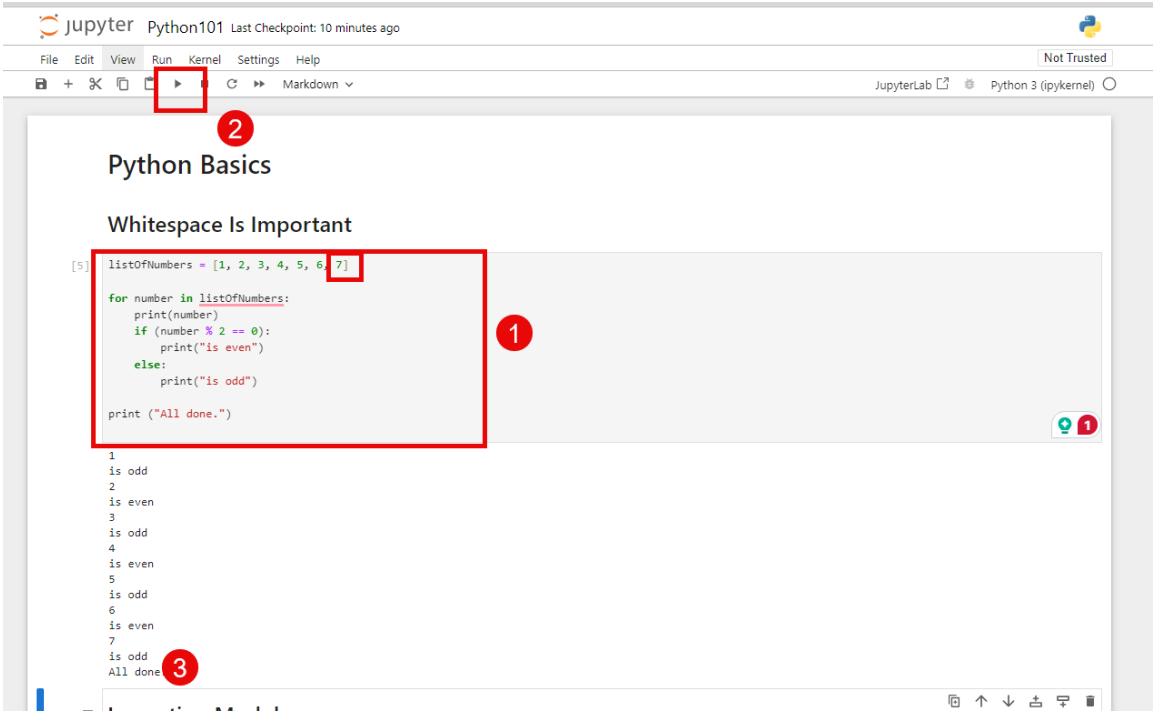


2. **No curly Braces** ({}) like C# and PowerApps

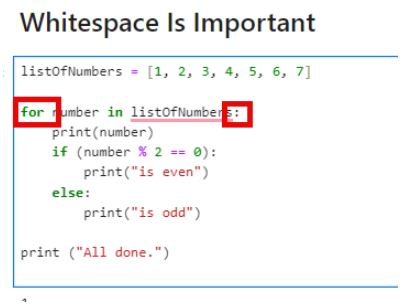
3. listOfNumbers = [1, 2, 3, 4, 5, 6]
   Lists (Just like Array)

4. **No character to terminate** line like C# (;)

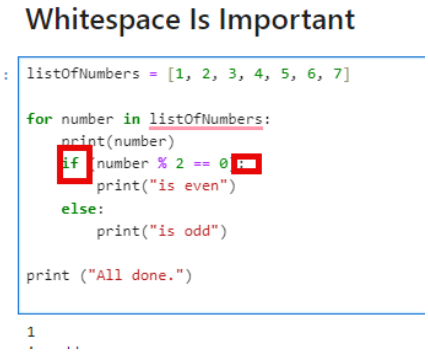
5. Add one more number in list and Execute



6. For statement is similar to other languages but needs : as the end



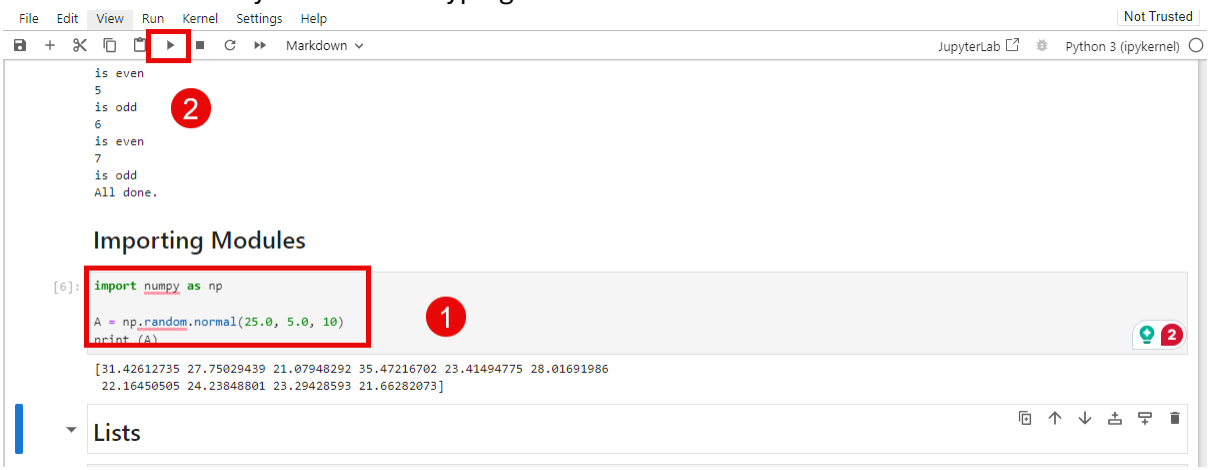
7. Same thing applies to if statement, it needs :



8. We **don't need to define variables ahead of** time nor the **data types**.

9. Python is what's called typed language.
10. But you can cast one variable to another. For e.g. string to integer.

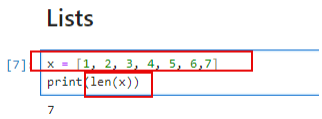
Import module

- Use **import** command (similar of **using** in C#)
- Define alias to save yourself some typing too.
- 

Lists

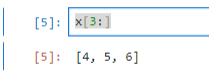
a list is a versatile and **mutable data structure** that can contain a **collection of items**, such as integers, strings, or even other lists

- **len()** (returns the number of elements in that list)



- **slice** part of list by using specific items
  for e.g. x[:3] will extract 1st three item

- x[3:] will last 3 items from the list



- x[-2:] negative from the list.

```
[11]: x[-2:]

[11]: [6, 7]
```

- **extend()**: **Appends the elements** of an iterable (such as another list) to the **end of the list.**

```
[12]: x.extend([7,8])
      x

[12]: [1, 2, 3, 4, 5, 6, 7, 7, 8]
```

- **append()**: Adds an element to the **end of the list.**

```
[8]: x.append(9)
     x

[8]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- List can contain just about any type you want. You can have a list of list.

```
[14]: y = [10, 11, 12]
      listOfLists = [x, y]
      listOfLists

[14]: [[1, 2, 3, 4, 5, 6, 7, 7, 8], [10, 11, 12]]
```

- To retrieve an element of a list, use [] bracket operator

```
[14]: y = [10, 11, 12]
      listOfLists = [x, y]
      listOfLists

[14]: [[1, 2, 3, 4, 5, 6, 7, 7, 8], [10, 11, 12]]

[15]: y[0]

[15]: 10
```

- Sort() - Sorts the elements of the list in **ascending order (0 – 10s)** (by default).

```
16]: z = [3, 2, 1, 0]
     z.sort()
     z

16]: [0, 1, 2, 3]
```

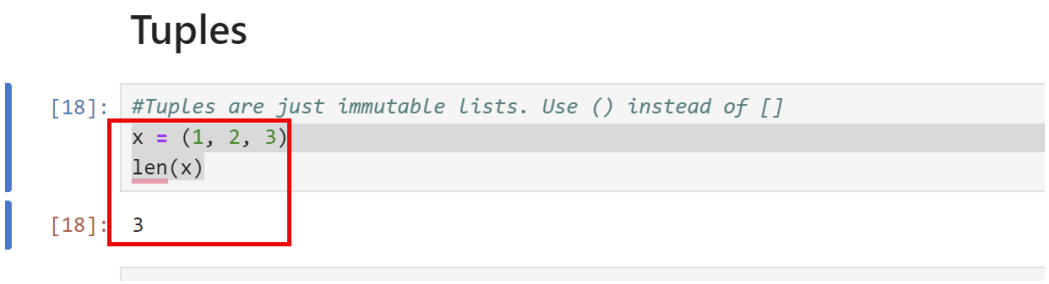
- Reverse() - reverses the order of the elements in the list.

```
[16]: z = [3, 2, 1, 0]
      z.sort()
      z

[16]: [0, 1, 2, 3]

[17]: z.sort(reverse=True)
      z

[17]: [3, 2, 1, 0]
```

Tuples

- tuples are just **immutable lists**. Use () instead of []
  Similar as C# Tuple.

- For e.g.
  x = (1, 2, 3)
  len(x)



- You can reference elements in a tuple in the same way that you would in a list as well.

```
[14]: y = (4, 5, 6)
      y[2]

[14]: 6
```

- You can make a **list of tuples** if you desire.

```
[14]: y = (4, 5, 6)
      y[2]

[14]: 6
```

- Another common use of tuples is in passing around **group of variables that you want to keep together.**

```
[16]: (age, income) = "32,120000".split(',')
      print(age)
      print(income)

      32
      120000
```

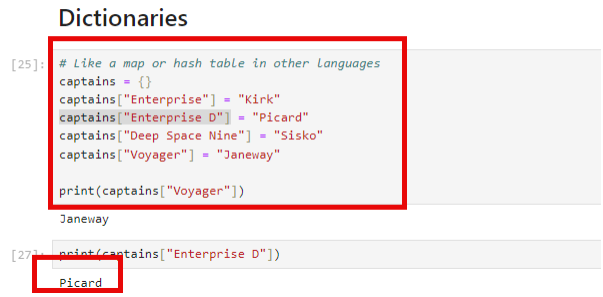
Split is same as C# and PowerApps

Dictionaries

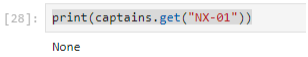
- Useful data structure in Python is the dictionary.
- Another language use has map or hash table. C# the same name Dictionary
- Defined as curly braces {}
-

```
[17]: # Like a map or hash table in other languages
      captains = {}
      captains["Enterprise"] = "Kirk"
      captains["Enterprise D"] = "Picard"
      captains["Deep Space Nine"] = "Sisko"
      captains["Voyager"] = "Janeway"

      print(captains["Voyager"])

      Janeway
```

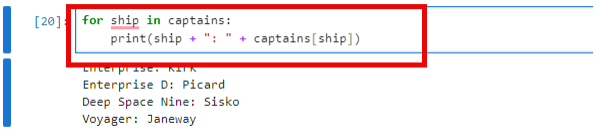
- You can get the value from dictionary as follows:



- If no value from dictionary, then it will be returned as none



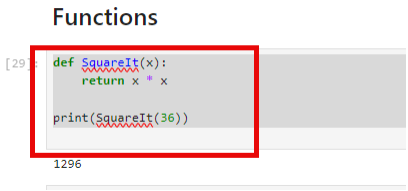
- You Iterate through all dictionary items by using for looping



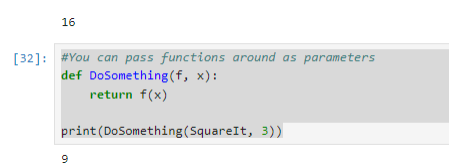
Functions

- To define a function in Python, you use the **def** keyword followed by the *function name* and parentheses containing **any parameters** the function takes

- For e.g.



- Some cool things with Function. You can pass another function to a function as parameter



- Lambda :

- In simpler terms, a lambda function in Python is a small**, anonymous function** that can have any number of parameters but only one expression.

- For e.g.

```
[33]: #Lambda functions let you inline simple functions
      print(DoSomething(lambda x: x * x * x, 3))

      27
```

Boolean

- In Python, the boolean data type represents a binary value, which can be either **True** or **False**.

- Booleans are used to represent the truth values of logical expressions.

Boolean Expressions

```
[34]: print(1 == 3)

      False

[37]: print(True or False)

      True

[40]: print(1 == 3)

      False

[42]: if 1 == 3:
          print("How did that happen?")
      elif 1 > 3:
          print("Yikes")
      else:
          print("All is well with the world")

      All is well with the world
```

Looping

- The **for loop** is used to iterate **over a sequence** (such as a list, tuple, string, or range) or any iterable object.

Looping

```
[28]: for i in range(10):
          print(i)

      0
      1
      2
      3
      4
      5
      6
      7
      8
```

- **Break :** terminates the loop prematurely when a certain condition is met.

```
9

[44]: for x in range(10):
          if (x == 1):
              continue
          if (x > 5):
              break
          print(x)

      0
      2
      3
      4
      5
```

- while loop - The while loop repeatedly executes a block of code as long as a specified condition is True.

```
      5

[30]: x = 0
      while (x < 10):
          print(x)
          x += 1

      0
      1
```

Create and Deploy Azure OpenAI Resource

Caution

- Cost
- Regional availability

Creating new Azure Open AI Instances

- Fine tune on only specific regions
- Keys and Endpoint

Deployment

Azure OpenAI Playground

LAB 1 Speech Service

Whisper – speech to text model

- Inputs Any language.
- Output is English
- 2 ways

- Azure OpenAI
- Azure Speech SDK

User Case

- Transcribe audio files
- Translate audio from other languages to English
- Provide prompt to the model to guide desired output
- Supported files formats such as
  - .mp3
  - .mp4
  - Mpweg
  - Etc

Azure OpenAI vs Azure Speech SDK

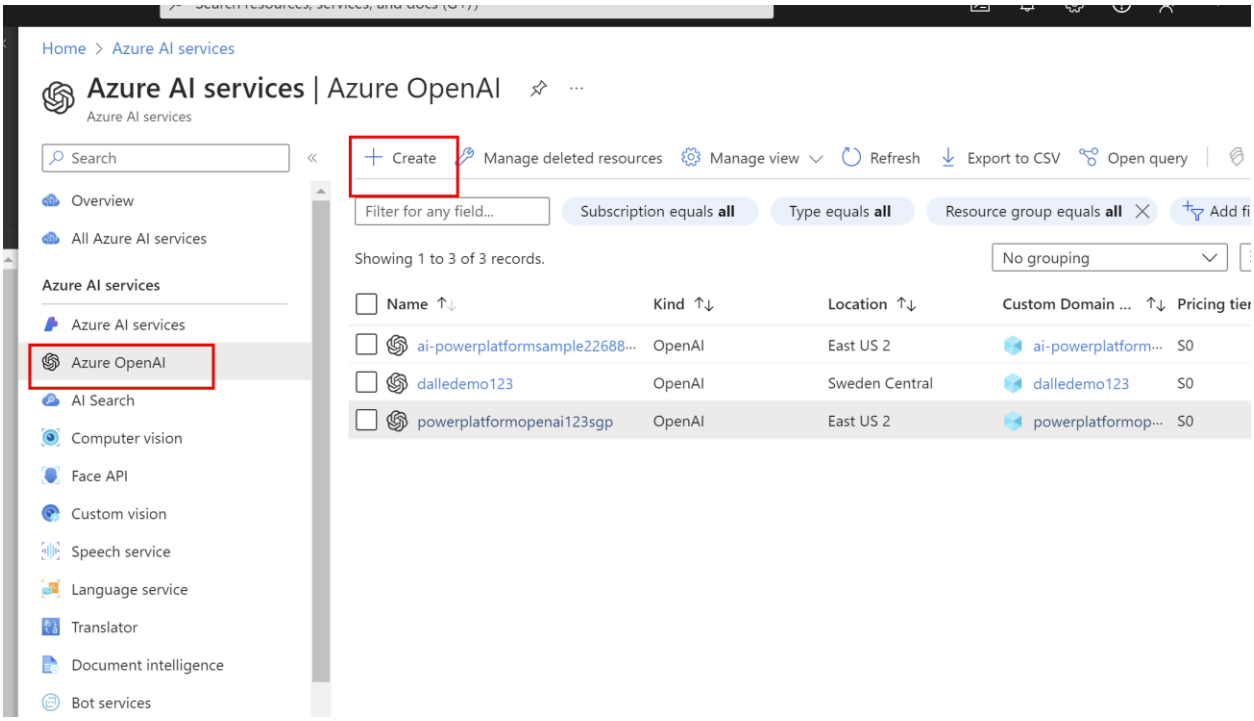
Azure OpenAI	Azure Speech SDK
Transfer files less than or equal to 25 MB	Transfer files greater than 25 MB (up to 1GB)
Does not support diarization (the process of **partitioning an audio stream containing** human speech into homogeneous segments **according to the** identity of each speaker) and no words timestamps supported	The Speech service provides information

Regional Support

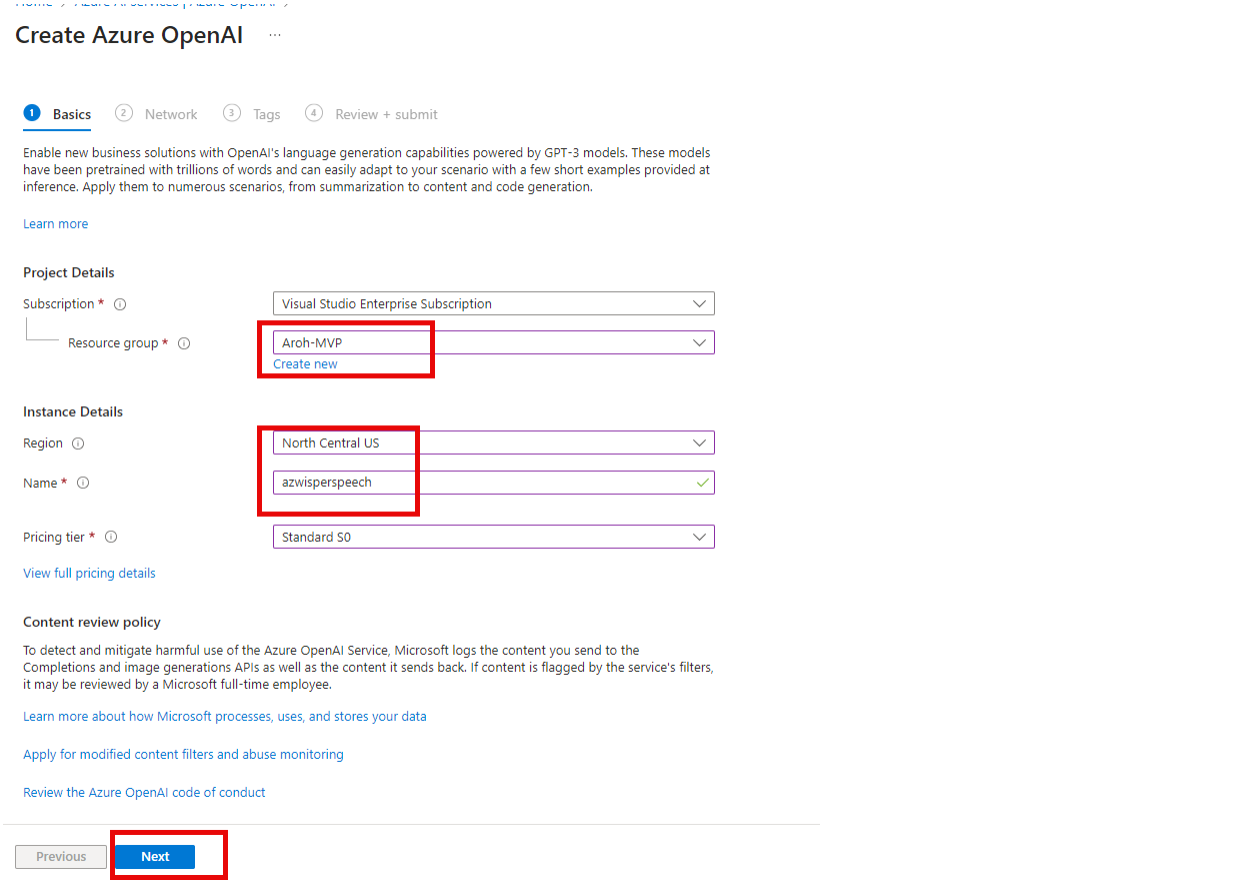
Azure OpenAI	North Central US, West Europe
Azure Speech SDK	East US, Southeast Asia and West Europe

Lab code with Python and VS Code with Azure OpenAI

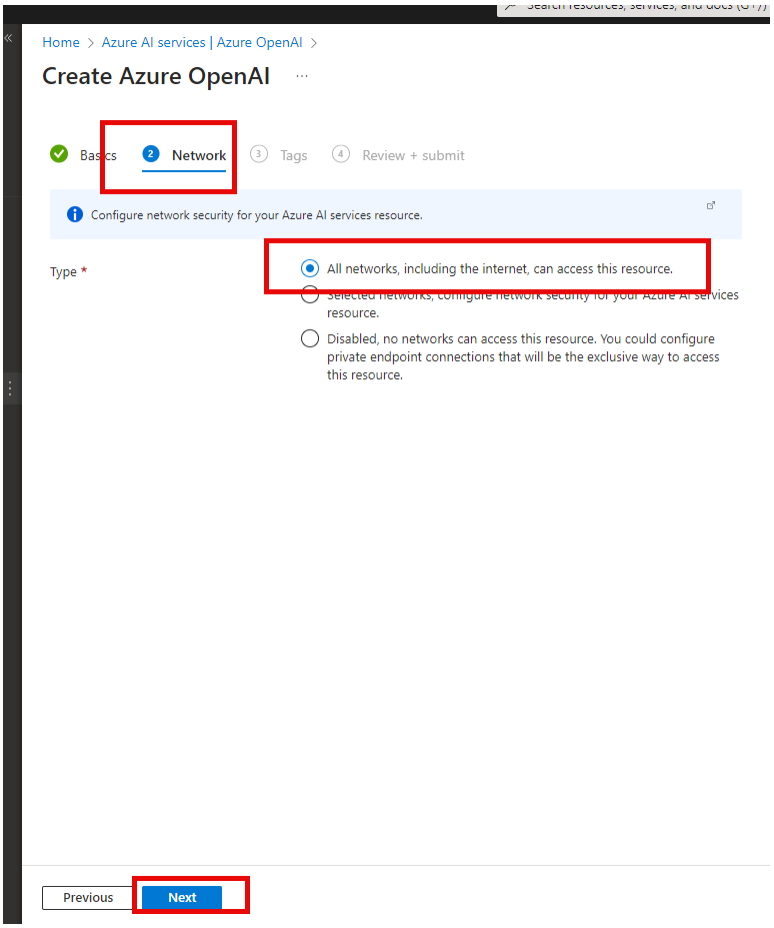
Azure OpenAI – Speech Service


Home > Azure AI services
Azure AI services | Azure OpenAI
Azure AI services
Search
Overview
All Azure AI services
Azure AI services
Azure AI services
Azure OpenAI
AI Search
Computer vision
Face API
Custom vision
Speech service
Language service
Translator
Document intelligence
Bot services
Create
Manage deleted resources
Manage view
Refresh
Export to CSV
Open query
Filter for any field...
Subscription equals all
Type equals all
Resource group equals all
Showing 1 to 3 of 3 records.
No grouping
Name
Kind
Location
Custom Domain ...
Pricing tier
ai-powerplatformsample22688...
OpenAI
East US 2
ai-powerplatform...
S0
dalledemo123
OpenAI
Sweden Central
dalledemo123
S0
powerplatformopenai123sgp
OpenAI
East US 2
powerplatformop...
S0

Basics

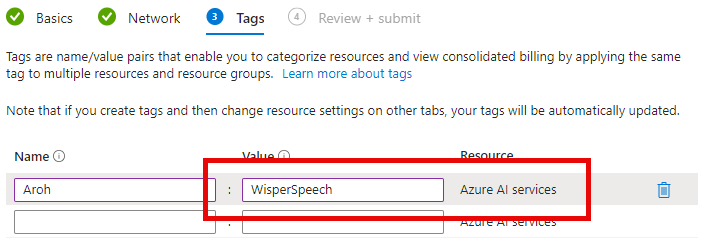

Create Azure OpenAI
1 Basics
2 Network
3 Tags
4 Review + submit
Enable new business solutions with OpenAI's language generation capabilities powered by GPT-3 models. These models have been pretrained with trillions of words and can easily adapt to your scenario with a few short examples provided at inference. Apply them to numerous scenarios, from summarization to content and code generation.
Learn more
Project Details
Subscription *
Visual Studio Enterprise Subscription
Resource group *
Aroh-MVP
Create new
Instance Details
Region
North Central US
Name *
azwisperspeech
Pricing tier *
Standard S0
View full pricing details
Content review policy
To detect and mitigate harmful use of the Azure OpenAI Service, Microsoft logs the content you send to the Completions and image generations APIs as well as the content it sends back. If content is flagged by the service's filters, it may be reviewed by a Microsoft full-time employee.
Learn more about how Microsoft processes, uses, and stores your data
Apply for modified content filters and abuse monitoring
Review the Azure OpenAI code of conduct
Previous
Next

Networking

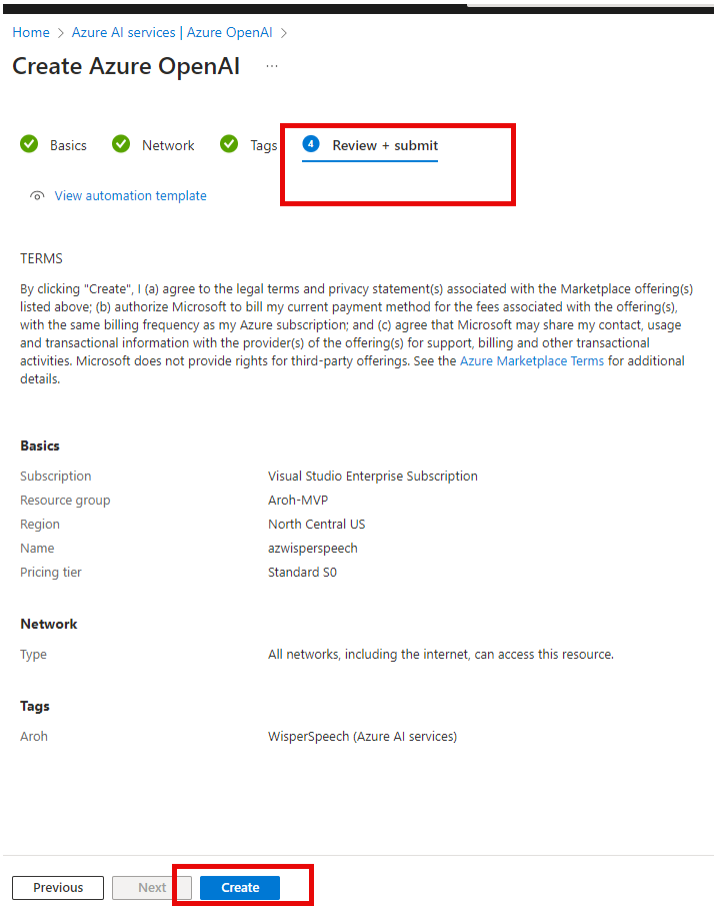


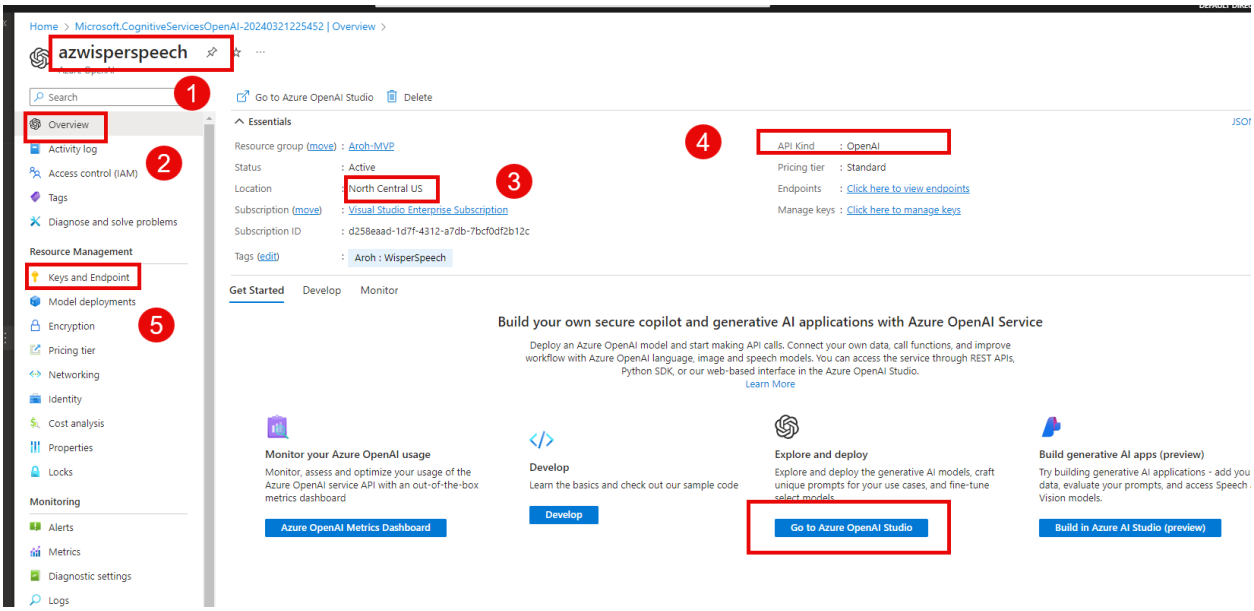
NOTE: All network is never be used in production environment,

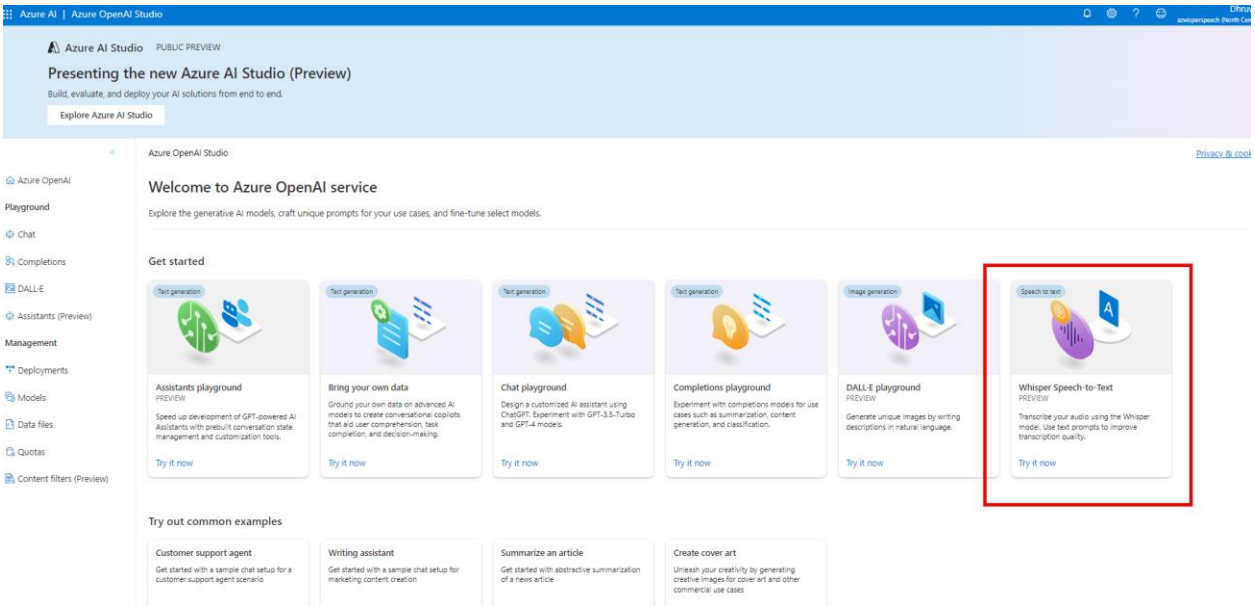
Tags

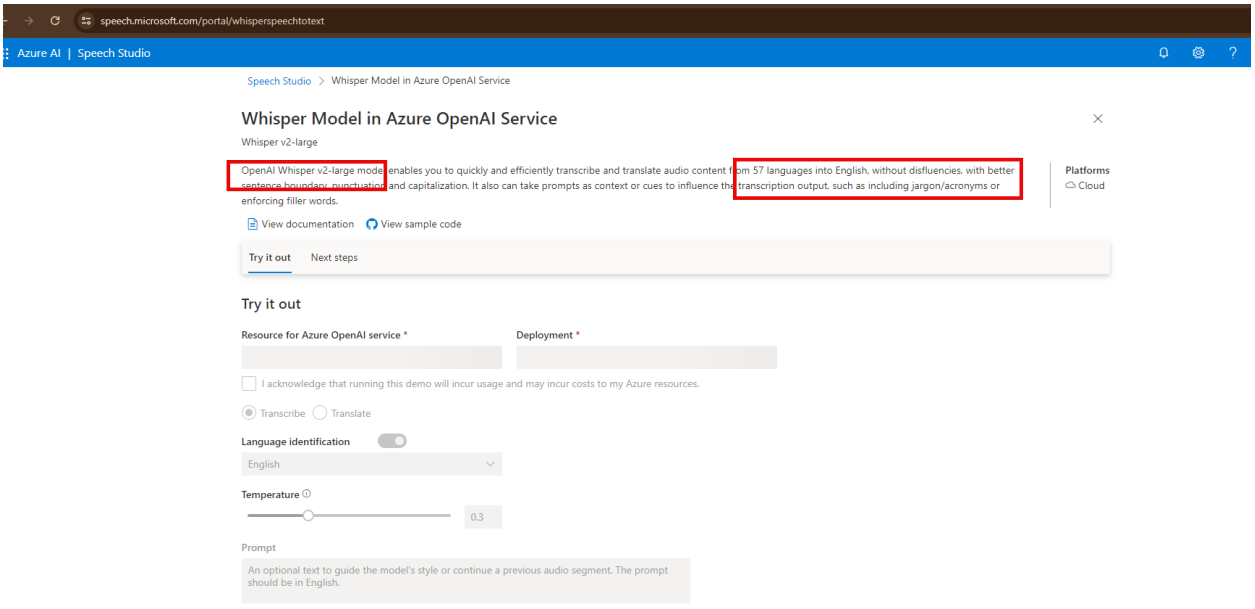


Review and Create

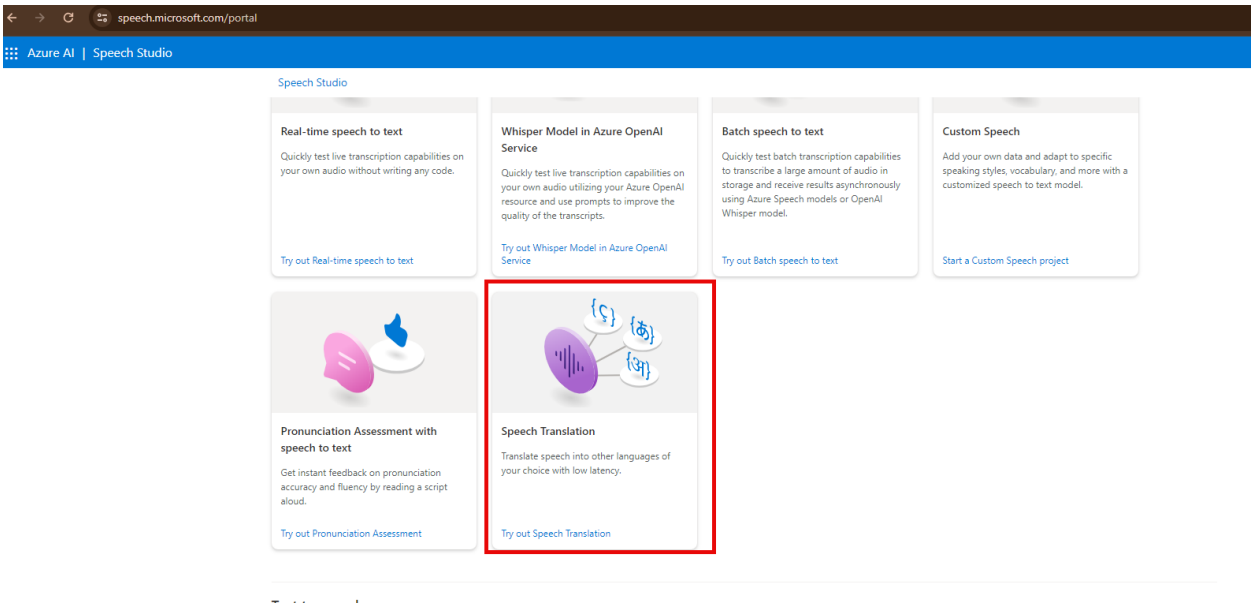

Home > Azure AI services | Azure OpenAI >
Create Azure OpenAI
Basics
Network
Tags
4 Review + submit
View automation template
TERMS
By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the Azure Marketplace Terms for additional details.
Basics
Subscription Visual Studio Enterprise Subscription
Resource group Aroh-MVP
Region North Central US
Name azwisperspeech
Pricing tier Standard S0
Network
Type All networks, including the internet, can access this resource.
Tags
Aroh WisperSpeech (Azure AI services)
Previous
Next
Create


Home > Microsoft.CognitiveServicesOpenAI-20240321225452 | Overview >
azwisperspeech
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource Management
Keys and Endpoint
Model deployments
Encryption
Pricing tier
Networking
Identity
Cost analysis
Properties
Locks
Monitoring
Alerts
Metrics
Diagnostic settings
Logs
Go to Azure OpenAI Studio
Delete
Essentials
Resource group (move) : Aroh-MVP
Status : Active
Location : North Central US
Subscription (move) : Visual Studio Enterprise Subscription
Subscription ID : d258eaad-1d7f-4312-a7db-7bcf0df2b12c
Tags (edit) : Aroh : WisperSpeech
API Kind : OpenAI
Pricing tier : Standard
Endpoints : Click here to view endpoints
Manage keys : Click here to manage keys
Get Started
Develop
Monitor
Build your own secure copilot and generative AI applications with Azure OpenAI Service
Deploy an Azure OpenAI model and start making API calls. Connect your own data, call functions, and improve workflow with Azure OpenAI language, image and speech models. You can access the service through REST APIs, Python SDK, or our web-based interface in the Azure OpenAI Studio.
Learn More
Monitor your Azure OpenAI usage
Monitor, assess and optimize your usage of the Azure OpenAI service API with an out-of-the-box metrics dashboard
Azure OpenAI Metrics Dashboard
Develop
Learn the basics and check out our sample code
Develop
Explore and deploy
Explore and deploy the generative AI models, craft unique prompts for your use cases, and fine-tune
Go to Azure OpenAI Studio
Build generative AI apps (preview)
Try building generative AI applications - add you data, evaluate your prompts, and access Speech Vision models.
Build in Azure AI Studio (preview)
1
2
3
4
5





https://speech.microsoft.com/portal



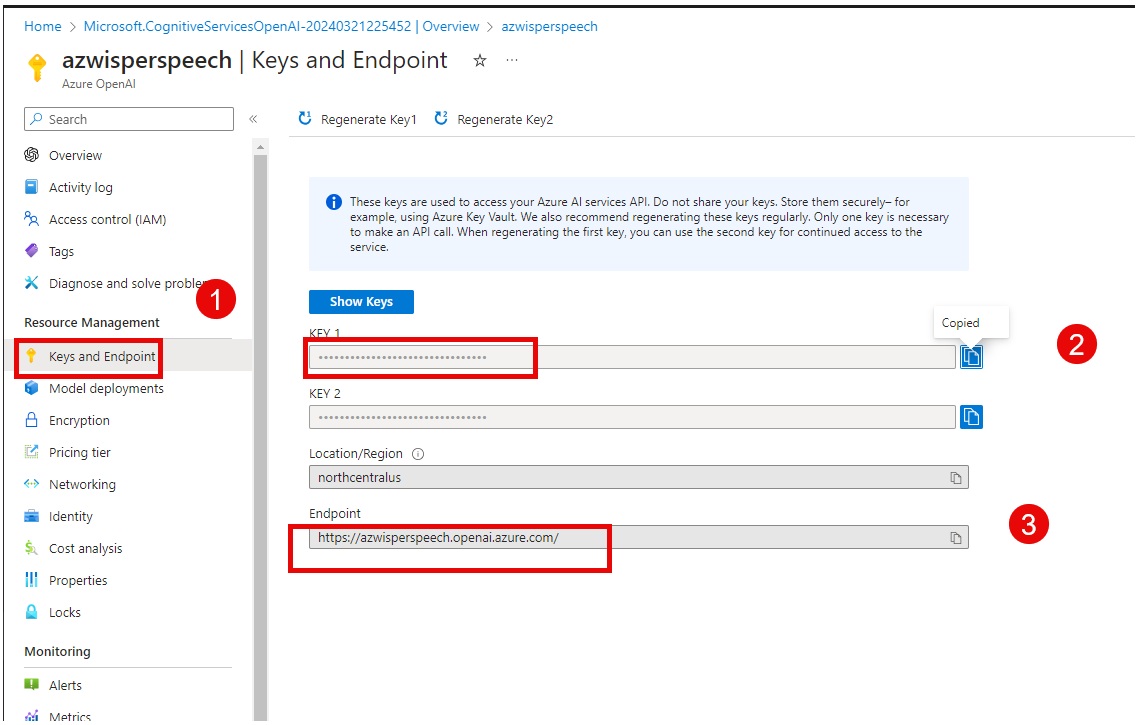
Key and endpoints

Keys

5d0489e3cff0**********

End Point

https://azwisperspeech.openai.azure.com/

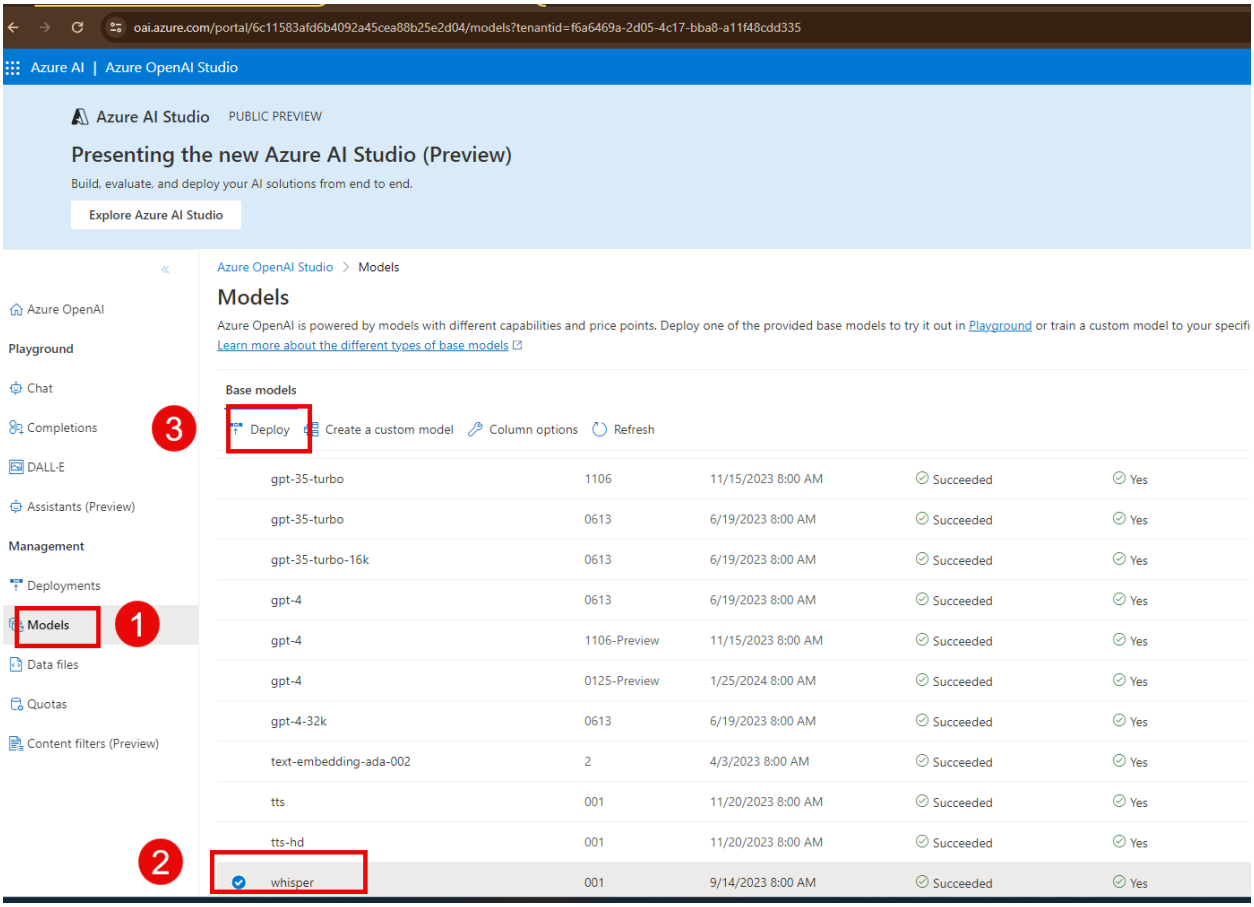


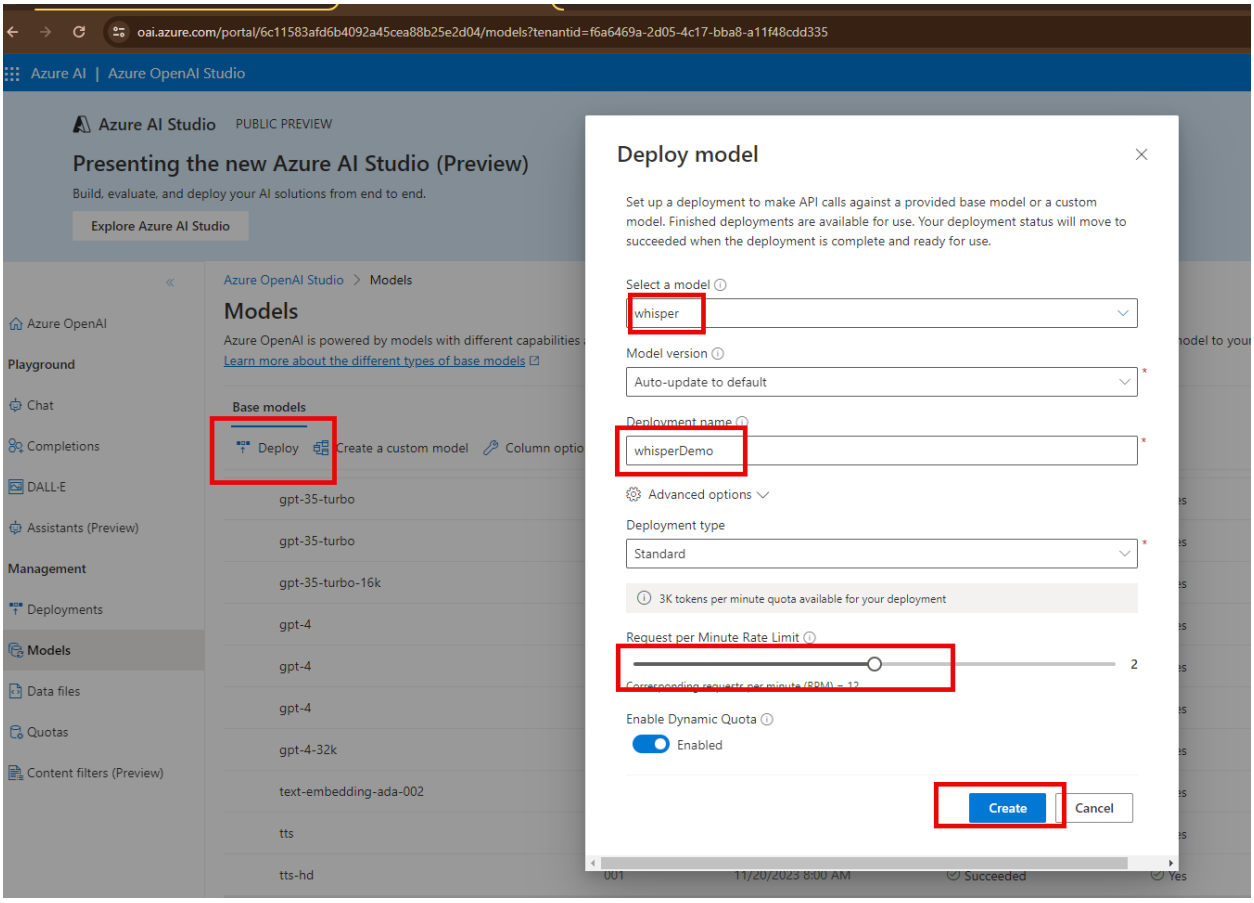
Deployment

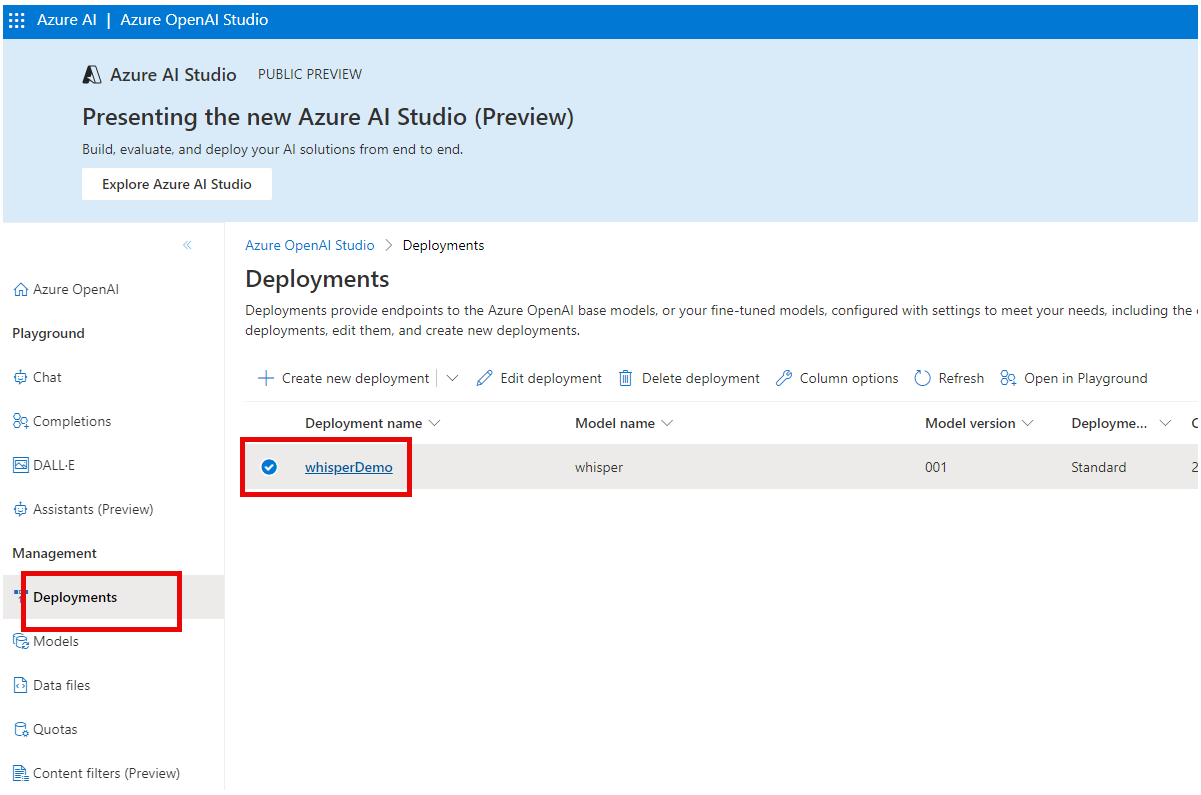
NOTE: You may receive this message
"Authorized failed: the user does not have permissions for this operation. Check the user roles to add the required permissions"

You need to assign proper permissions on Cognitive Services OpenAI Contributor on IAM blade.
Reference: https://learn.microsoft.com/en-us/answers/questions/1329334/making-custom-model-on-azure-openai-studio-is-show

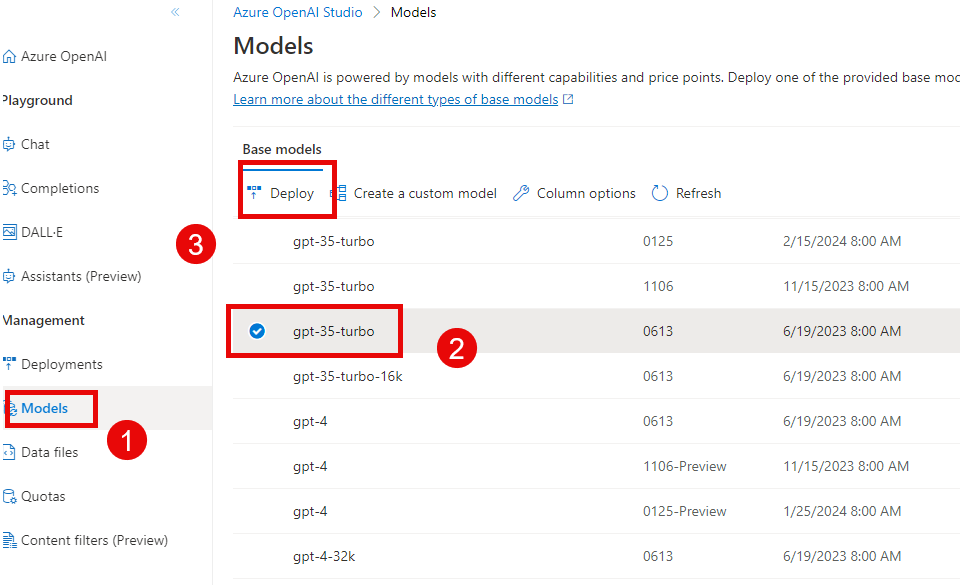
Whisper Model

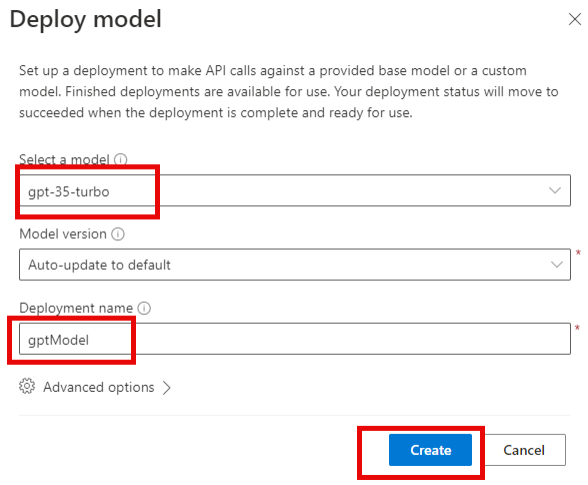


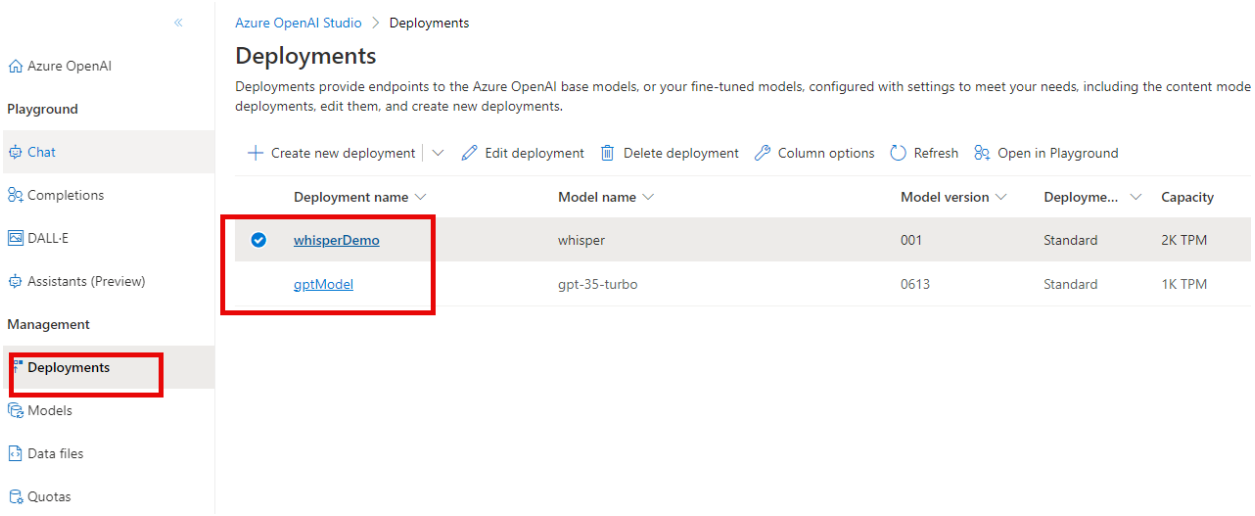

oai.azure.com/portal/6c11583afd6b4092a45cea88b25e2d04/models?tenantid=f6a6469a-2d05-4c17-bba8-a11f48cdd335
Azure AI | Azure OpenAI Studio
Azure AI Studio PUBLIC PREVIEW
Presenting the new Azure AI Studio (Preview)
Build, evaluate, and deploy your AI solutions from end to end.
Explore Azure AI Studio
Azure OpenAI Studio > Models
Models
Azure OpenAI is powered by models with different capabilities
Learn more about the different types of base models
Base models
Deploy
Create a custom model
Column optio
Azure OpenAI
Playground
Chat
Completions
DALL·E
Assistants (Preview)
Management
Deployments
Models
Data files
Quotas
Content filters (Preview)
gpt-35-turbo
gpt-35-turbo
gpt-35-turbo-16k
gpt-4
gpt-4
gpt-4
gpt-4-32k
text-embedding-ada-002
tts
tts-hd
Deploy model
Set up a deployment to make API calls against a provided base model or a custom model. Finished deployments are available for use. Your deployment status will move to succeeded when the deployment is complete and ready for use.
Select a model
whisper
Model version
Auto-update to default
Deployment name
whisperDemo
Advanced options
Deployment type
Standard
3K tokens per minute quota available for your deployment
Request per Minute Rate Limit
2
Enable Dynamic Quota
Enabled
Create
Cancel
11/20/2023 8:00 AM
Succeeded


Azure AI | Azure OpenAI Studio
Azure AI Studio
PUBLIC PREVIEW
Presenting the new Azure AI Studio (Preview)
Build, evaluate, and deploy your AI solutions from end to end.
Explore Azure AI Studio
Azure OpenAI Studio > Deployments
Deployments
Deployments provide endpoints to the Azure OpenAI base models, or your fine-tuned models, configured with settings to meet your needs, including the
deployments, edit them, and create new deployments.
Create new deployment
Edit deployment
Delete deployment
Column options
Refresh
Open in Playground
Deployment name
Model name
Model version
Deployme...
whisperDemo
whisper
001
Standard
Azure OpenAI
Playground
Chat
Completions
DALL·E
Assistants (Preview)
Management
Deployments
Models
Data files
Quotas
Content filters (Preview)

GPT-Turbo


Azure OpenAI Studio > Models
Models
Azure OpenAI is powered by models with different capabilities and price points. Deploy one of the provided base moc
Learn more about the different types of base models
Base models
Deploy
Create a custom model
Column options
Refresh
gpt-35-turbo 0125 2/15/2024 8:00 AM
gpt-35-turbo 1106 11/15/2023 8:00 AM
gpt-35-turbo 0613 6/19/2023 8:00 AM
gpt-35-turbo-16k 0613 6/19/2023 8:00 AM
gpt-4 0613 6/19/2023 8:00 AM
gpt-4 1106-Preview 11/15/2023 8:00 AM
gpt-4 0125-Preview 1/25/2024 8:00 AM
gpt-4-32k 0613 6/19/2023 8:00 AM
Azure OpenAI
Playground
Chat
Completions
DALL·E
Assistants (Preview)
Management
Deployments
Models
Data files
Quotas
Content filters (Preview)
1
2
3



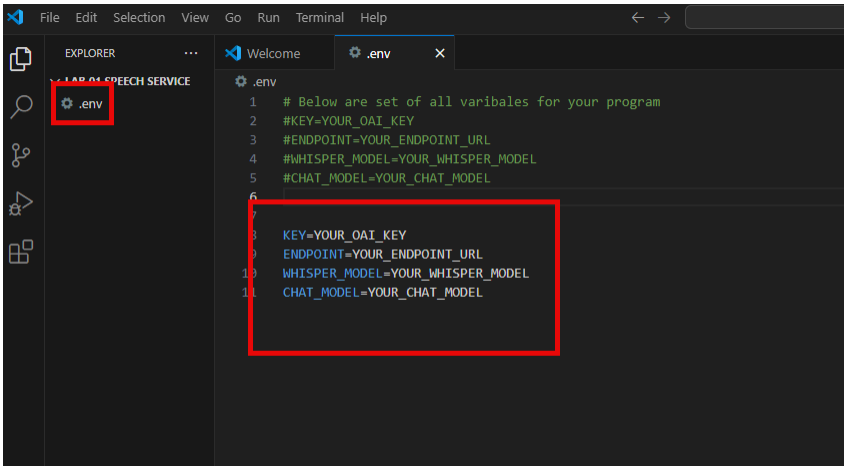


Speech Service – Azure AI

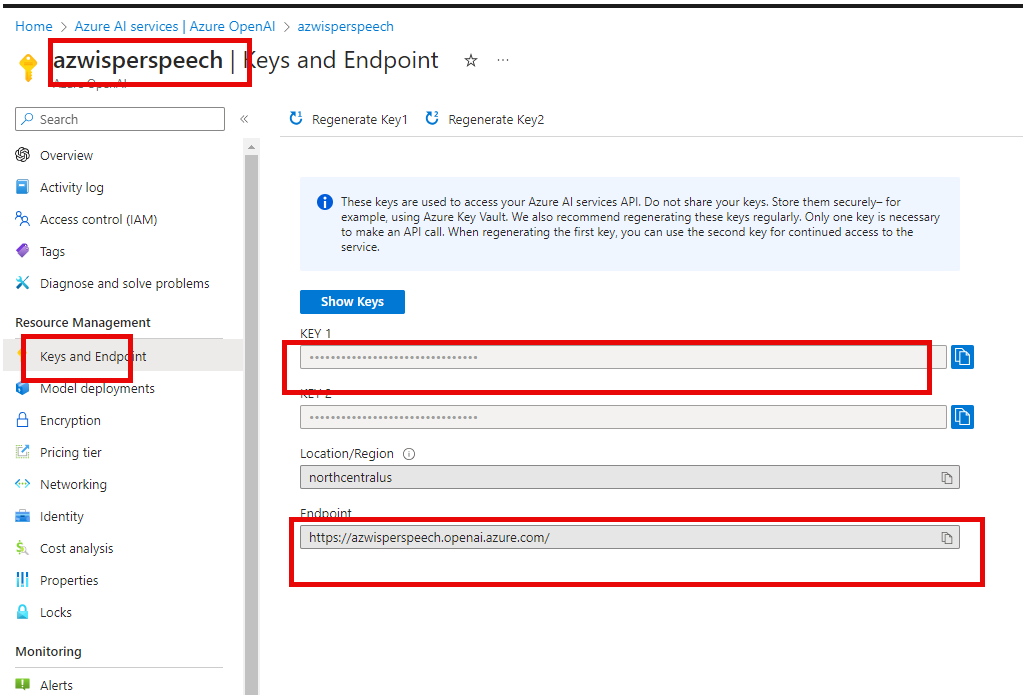
VS Code, Speech Service and Test

Step 1: Setup Environment Variables

Create a new .env file to set all variables used by your program in VS Code

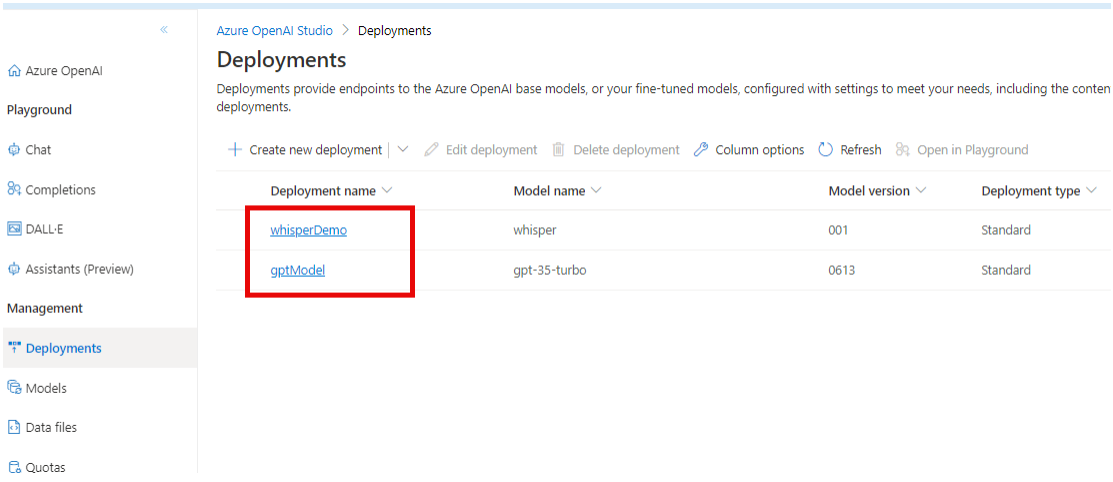


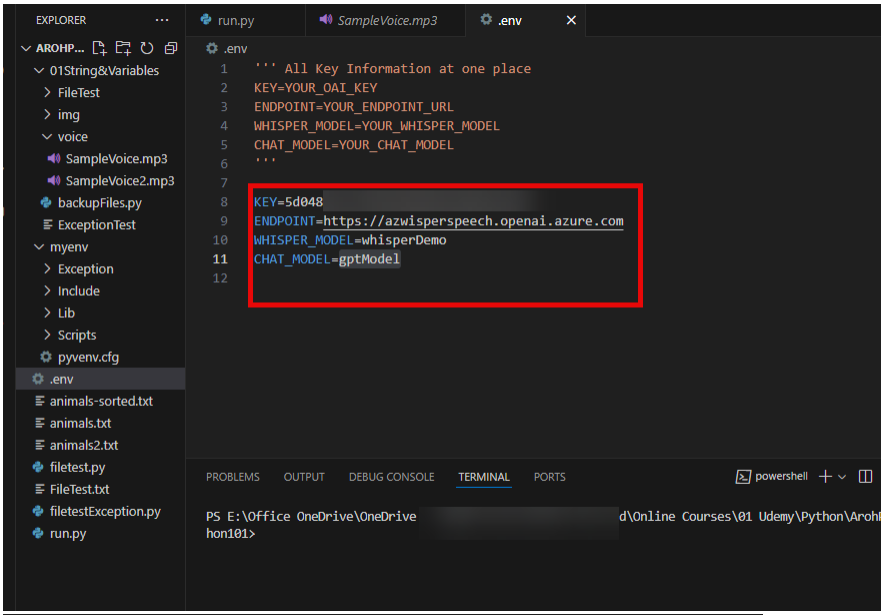
Copy key and endpoint from Azure OpenAI speech service instance

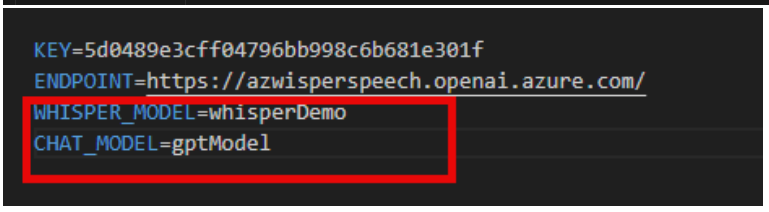


Paste key and endpoint to VS Code .env file

Do the same for whisper model and chat model.
Note: It should be deployment name and NOT the model name.

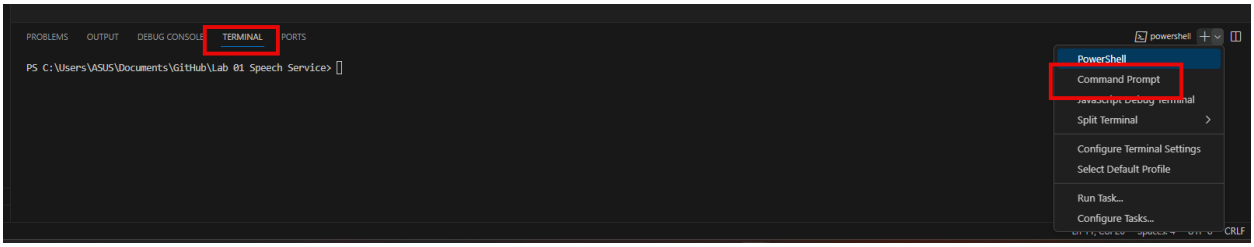




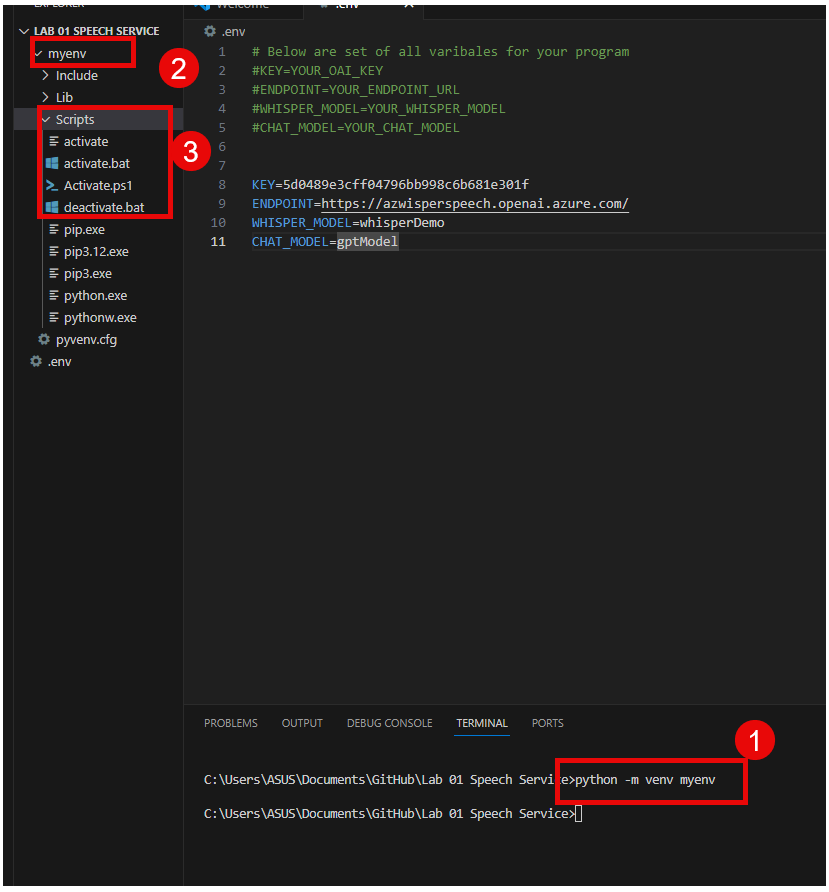


Step 2: Install necessary Python packages

Under terminal change to command prompt

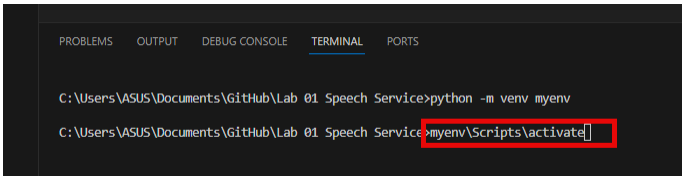


Create a new environment



python -m venv myenv

Activate scripts



myenv\Scripts\activate (backslash)

Install requests modules

Requests module

Pip install **requests**

allows you to send HTTP requests using Python.

```
(myenv) C:\Users\ASUS\Documents\GitHub\Lab 01 Speech Service>pip install requests
Collecting requests
```

os module – NOT Required

os is already installed on Windows

https://stackoverflow.com/questions/48010748/how-to-install-the-os-module

load_dot module

```
(myenv) C:\Users\ASUS\Documents\GitHub\Lab 01 Speech Service>pip install load_dotenv
Collecting load_dotenv
  Downloading load_dotenv-0.1.0-py3-none-any.whl.metadata (1.9 kB)
```

Json – Not Required

AzureOpenAI module

```
(myenv) C:\Users\ASUS\Documents\GitHub\Lab 01 Speech Service>pip install AzureOpenAI
Collecting AzureOpenAI
  Downloading azureopenai-0.0.1-py2.py3-none-any.whl.metadata (128 bytes)
```

OpenAI module

```
(myenv) C:\Users\ASUS\Documents\GitHub\Lab 01 Speech Service>pip install openai
Collecting openai
```

Pip install openai

Create a new file called run.py

- Import key libraries

```
import requests
import os
from dotenv import load_dotenv
import json
from openai import AzureOpenAI
```

- Load environment variables

```
load_dotenv()

'loading environment  variables'
key = os.getenv("KEY")
endpoint_url = os.getenv("ENDPOINT")
whisper_model=os.getenv("WHISPER_MODEL")
chat_model=os.getenv("CHAT_MODEL")
```

- concatenate above environment variables into final URL as POST request

```
'concatenate  above environment variables into final URL as POST request'
final_url = f"{endpoint_url}/openai/deployments/{whisper_model}/audio/transcriptions?api-version=2023-09-01-preview"
```

- Set header part as of POST request (Same as you do Power Automate or Azure Web Apps)

```
'header part of POST request'
headers = {
    "api-key": key
}
```

- load a sample voice sample from the project

```
'load a sample voice sample from the project'
file_path = r"voice/SampleVoice.mp4"
```

- Use Pythons library to open the sample voice in binary mode and later close the filestream after reading

```
'Use Pythons library  to open the sample voice in binary mode and later close the filestream after reading'
with open(file_path, "rb") as file:
    files = {
        "file": (os.path.basename(file_path),file, "application/octet-stream")
    }
```

- use requests library to call HTTP to POST

```
'use requests library to call HTTP to POST'
final_response = requests.post(final_url, headers=headers, files=files).json()
```

- Create a user prompt variable

```
'Create a user prompt variable '
user_prompt = final_response['text']
```

- Create Azure OpenAI library as a client

```
'Create Azure OpenAI library as a client'
client=AzureOpenAI(
    client = endpoint_url,
    api_key=key,
    api_version="2023-05-15"
)
```

- Create a response object for Azure OpenAI chat completions API

```
'Create a response  object for Azure OpenAI chat completions API'
response = client.chat.completions.create(
    model=chat_model,
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": user_prompt}
    ]
)
```

- Print the final response.

```
'print the final reponse'
print(response.choices[0].message.content)
```

References

1. How to install Python Packages on VS Code
   https://www.youtube.com/watch?v=eWk497uCgf0

2. Speech to text Azure AI Service
   https://learn.microsoft.com/en-us/azure/ai-services/openai/reference#chat-completions