Azure + Open AI

What is Azure OpenAI

Azure OpenAI refers to the collaboration between Microsoft Azure (a cloud computing platform), and OpenAI (an artificial intelligence research organization).

Integration of OpenAI Models: Provides access to OpenAI's AI models like GPT, Codex, and DALLE on Microsoft Azure's cloud platform.

Enterprise Applications: Designed for business use, helping companies integrate AI into various functions such as customer support and content generation.

Scalable Infrastructure: Utilizes Azure's cloud infrastructure for scalable AI model deployment and management.

Security and Compliance: Focuses on high security and adherence to regulatory standards to protect user data.

DeveloperFriendly: Offers tools and APIs for developers to easily implement and customize AI solutions.

Customization Options: Allows users to tailor AI models to better fit their specific requirements.

Global Reach: Available worldwide through Azure's extensive global infrastructure.

Collaboration and Innovation: Supports collaborative AI development and innovation within the Azure ecosystem.

Imp. ChatGpT & OpenAI subscriptions are different

Azure + Open AI

- collaboration b/w Azure & Open AI
- Secure, scalable
- It is enterprise version of Open AI

Imp. Not every model is available in every region because of load balancing.

Limit Type	Typical Numbers / Metrics	Notes
Request Rate / Throttling	**20–60 requests per minute** per deployment (depends on SKU)	Exceeding → 429 errors
Concurrent Requests	**5–10 concurrent requests** per deployment by default	Exceeding → throttling; can request increase
Token Limits	**GPT-4:** 8k or 32k tokens, **GPT-3.5:** 4k tokens	Input + output combined
Deployment Limits	Default **~10 deployments** per subscription	Can request more if needed

💡 **Interview tip:** Mention **"Azure OpenAI has request rate limits (20–60/min), concurrent request limits (5–10), and token limits (4k–32k depending on model)"**—this shows practical understanding without memorizing every SKU.

Azure Open AI Pricing:

GPT-4.1 series

GPT-4.1 series is a highly advanced general-purpose model with extensive world knowledge and an enhanced ability to understand user intent, making it particularly adept at creative tasks and agentic planning. The series features a 1 million token context window and has a knowledge cutoff of June 2024.

4

Model	Pricing (1M Tokens)	Pricing with Batch API (1M Tokens)
GPT-4.1-2025-04-14 Global	Input: $2 Cached Input: $0.50 Output: $8	Input: $1 Output: $4

o3

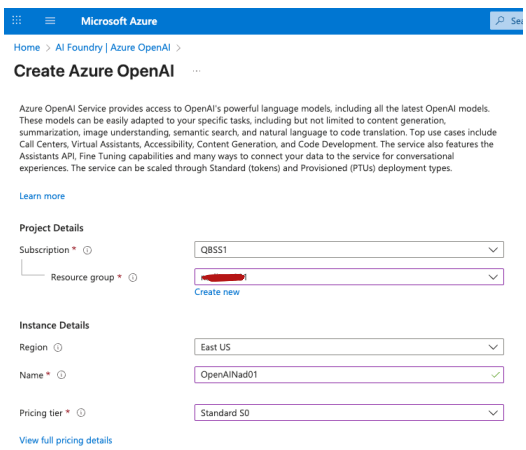
Model	Pricing
o3-deep research Global	Input: $10 Cached Input: $2.50 Output: $40

GPT-5 series

5

Model	Pricing (1M Tokens)
GPT-5 2025-08-07 Global	Input: $1.25 Cached Input: $0.13 Output: $10

Sign up to Azure Open AI



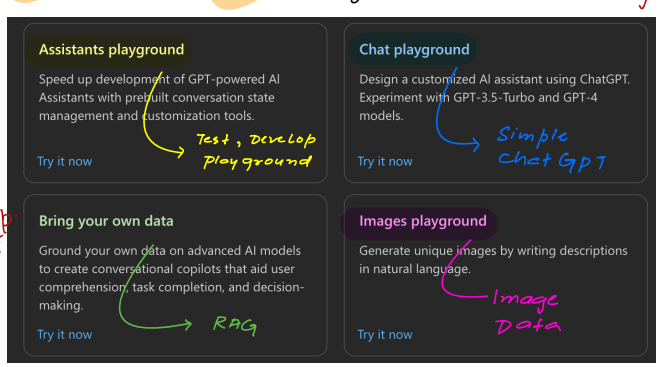
why Open AI + Azure

- Security
- Scalability
- Infrastructure

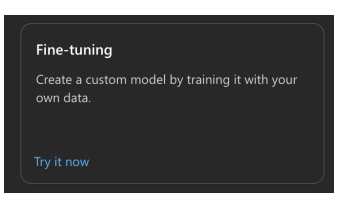
Imp. Old = Azure AI Studio
New = Azure AI Foundry
} Azure AI Studio is now AI Foundry

Links
Azure portal = portal.azure
Azure AI = ai.azure

Azure AI Foundry → previously AI Studio.



Imp.



↳ Base model
↓
Then Finetune

AZURE AI MODELS

Chat	→	GPT
Code	→	Codex
Images	→	Dall E
Video	→	SORA
Embedding	→	Text-Embedding

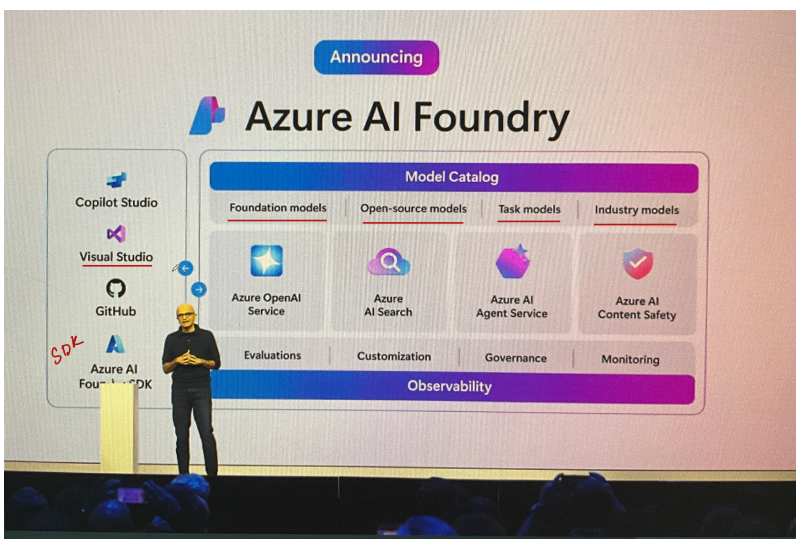
Foundry: Factory where metal is melted and Reshaped

Q/ what is Azure AI Foundry ?

Azure AI Foundry is Microsoft's all-in-one platform for building, testing, and deploying AI apps and agents. It combines:

- A **catalog of AI models** (from Microsoft, OpenAI, Meta, etc.) Imp. not only meta.
- Tools for **fine-tuning, RAG, and agent creation**
- Built-in **governance, security, and monitoring**
- Support for **enterprise use** (projects, access control, compliance)

It's designed to help companies go from prototyping to production faster with a unified, scalable AI development environment.



Q/ Hubs vs projects ?

- **Hub**: A **central workspace** for managing **multiple projects**. It's where organizational settings, resources, billing, and policies are set.
- **Project**: An **isolated workspace** within a hub for building and managing a specific AI solution (e.g., a chatbot, RAG app). Each project has its own data, models, configs, etc.

👉 Think of a **Hub** as the organization or team-level container, and **Projects** as individual AI use cases inside it.

Hub = workspace
project = individual AI solution

Imp. = Azure AI Foundry ; 1. All models
2. Azure open AI (only openAI)

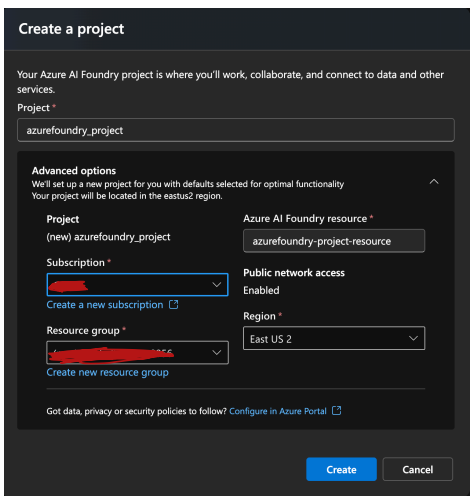
OPTIONS

1. Azure open AI with project - only open AI model
2. Azure open AI without project - other models, SDK

Imp. = If you are solely interested in Azure open AI then there is no need to creck a project.

Creating Project

Go to Azure AI Services



Q. How to compare models?

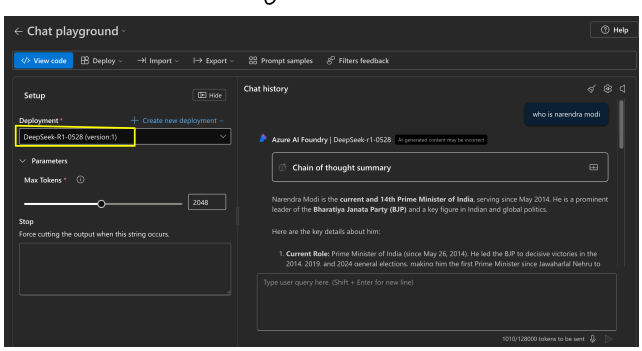
Model catalog

↓

Compare models

Deploying Models

Model catalog ⟶ Choose model → Open in playground



← Deploying models

Imp. O1 are Research Models = more complex / costly

How to Use

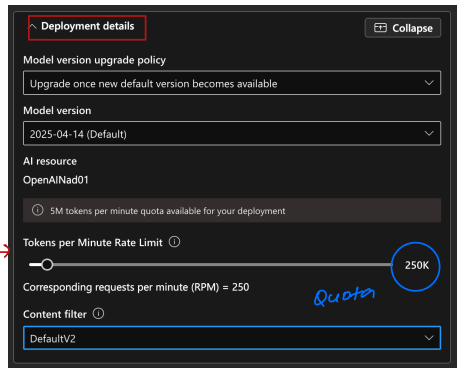
In order to use any of these we need to create Deployments

AZURE AI SERVICES

Service	One-liner
Chat	No-code playground to prototype prompts and get responses via Chat Completions before moving to code. learn.microsoft
Assistants	Build configurable copilots that use models plus tools, threads, and files to run tasks. learn.microsoft
Video (Preview)	Playground/APIs to work with multimodal models that understand or generate video content. learn.microsoft
Audio (Preview)	Use speech models (e.g., Whisper family) for transcription, translation, or audio tasks. learn.microsoft
Images	Try image generation models and export code for image workflows. microsoft
Fine-tuning	Train custom versions of base models on domain data and deploy them securely. learn.microsoft
Azure OpenAI Evaluation (Preview)	Evaluate prompts and systems with built-in metrics and datasets to compare quality and safety. learn.microsoft
Stored completions (Preview)	Persist chat/completion histories as datasets to reuse for evals or fine-tuning. learn.microsoft
Batch jobs	Submit large, asynchronous request batches for cost-efficient high-volume processing. learn.microsoft
Monitoring	Track requests, tokens, PTU usage, and fine-tune metrics with Azure Monitor dashboards. learn.microsoft
Deployments	Manage model deployments tied to an endpoint for use in apps and playgrounds. learn.microsoft
Quota	View and manage capacity limits, rate limits, and PTU allocations for the resource. learn.microsoft
Guardrails + Controls	Configure safety filters, content policies, and governance settings for responsible use. learn.microsoft
Risks + alerts (Preview)	Get proactive risk insights and alerts across usage, safety, and operations. learn.microsoft
Data files	Upload and manage files for assistants, fine-tuning, evals, and tool use. learn.microsoft
Assistant	Workspace to configure an individual Assistant's instructions, tools, and files. learn.microsoft
Vector stores (Preview)	Create and attach retrieval indexes for RAG so assistants can ground answers on documents. learn.microsoft

→ Chat Playground

Deploy ⟶ Choose model ⟶



Imp. If Token limit = 250K

e.g., 2 Deployments = 125K/each

Imp. we dont need to call API here, no code is required

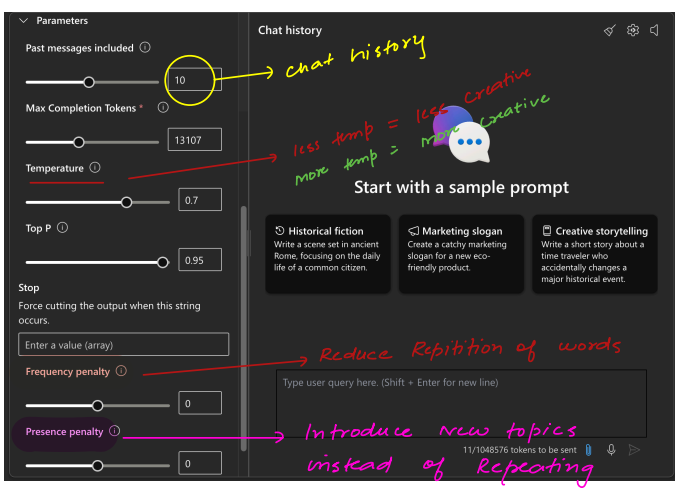
PARAMETERS

Parameter	Meaning (One-liner)
Past Messages	Number of previous conversation turns the model considers for context.
Max Token Completion	Maximum length of the model's response (in tokens).
Temperature	Controls randomness; higher = more creative, lower = more focused.
Top P	Nucleus sampling; limits output to top tokens whose probabilities sum to *P*.
Top K	Limits output choices to the *K* most likely tokens at each step.
Presence Penalty	Encourages the model to introduce new topics instead of repeating.
Frequency Penalty	Reduces repetition of words/tokens already used.

90% probability

K=2

Parameter	Explanation	Example
Top-P (Nucleus Sampling)	The model considers the smallest set of tokens whose **cumulative probability ≥ P** (e.g., P=0.9 → choose from tokens covering 90% of the probability mass).	If possible next words have probs {"cat" 0.5, "dog" 0.3, "fish" 0.1, "car" 0.1}, with P=0.9, it will only consider {cat, dog, fish}.
Top-K	The model only considers the **K most likely tokens** and ignores the rest, no matter their probability mass.	With K=2, only {"cat", "dog"} are considered even if "fish" had 0.1 probability.



Q// what is use of Azure + Open AI?

- Enhanced Integration & Enterprise level security.

OpenAI API CallsVs Azure OpenAI

All models are based on R.L & Human Feedback

OpenAI

Python

```python
import os
from openai import OpenAI

client = OpenAI(
    api_key=os.getenv("OPENAI_API_KEY")
)
```

Azure OpenAI

Python

```python
import os
from openai import AzureOpenAI

client = AzureOpenAI(
    api_key=os.getenv("AZURE_OPENAI_API_KEY"),
    api_version="2023-12-01-preview",
    azure_endpoint=os.getenv("AZURE_OPENAI_ENDPOINT")
)
```

Store API key in .env file so it is not accessible

So 2 additions are;
api_version
azure Endpoint

OpenAI

Python

```python
completion = client.completions.create(
    model="gpt-3.5-turbo-instruct",
    prompt="<prompt>"
)

chat_completion = client.chat.completions.create(
    model="gpt-4",
    messages="<messages>"
)

embedding = client.embeddings.create(
    model="text-embedding-ada-002",
    input="<input>"
)
```

Azure OpenAI

Python

```python
completion = client.completions.create(
    model="gpt-35-turbo-instruct", # This must match the custom deployment name you chose for your model.
    prompt="<prompt>"
)

chat_completion = client.chat.completions.create(
    model="gpt-35-turbo", # model = "deployment_name".
    messages="<messages>"
)

embedding = client.embeddings.create(
    model="text-embedding-ada-002", # model = "deployment_name".
    input="<input>"
)
```

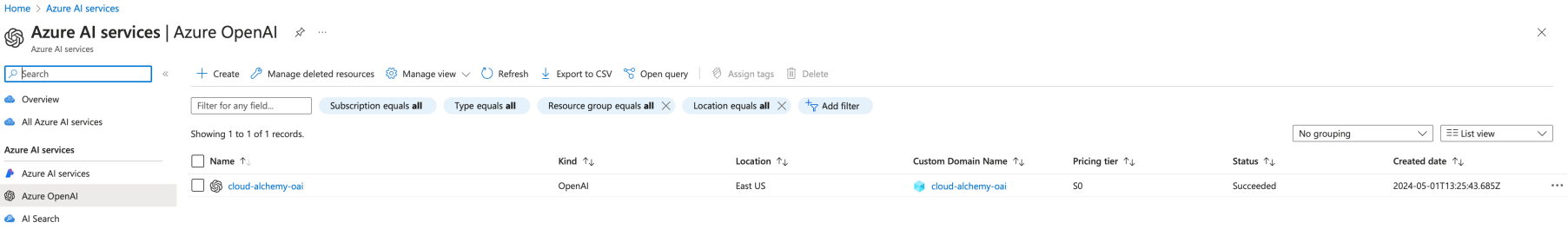
↳ Deployment / Not model

Understanding Azure OpenAI API Calls

azure_endpoint = "https://cloudalchemyoai.openai.azure.com/",
api_key=os.environ.get("OPENAI_API_KEY"),
api_version="20240215preview”

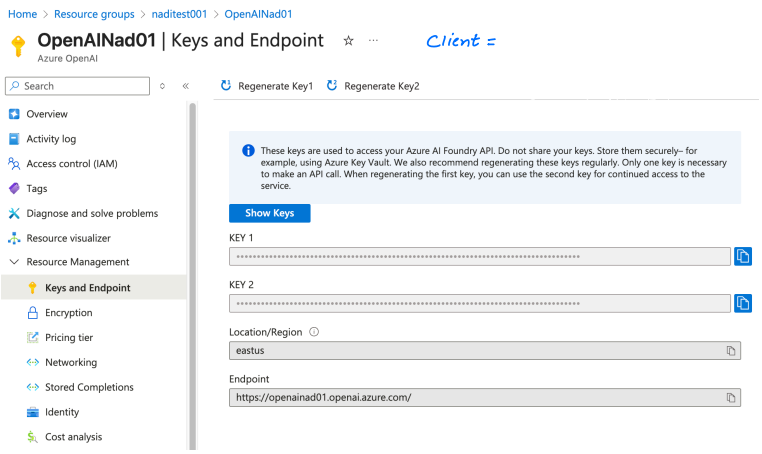
```
completion = client.chat.completions.create(
model="air_text_lab", # model = "deployment_name"
messages = message_text,
temperature=0.7,
max_tokens=800,
top_p=0.95,
frequency_penalty=0,
presence_penalty=0,
stop=None
)
```

```python
#Note: The openai-python library support for Azure OpenAI is in preview.
#Note: This code sample requires OpenAI Python library version 1.0.0 or higher.
import os
from openai import AzureOpenAI

client = AzureOpenAI(
  azure_endpoint = "https://cloud-alchemy-oai.openai.azure.com/",
  api_key=os.getenv("AZURE_OPENAI_KEY"),
  api_version="2024-02-15-preview"
)

message_text = [{"role":"system","content":"You are an AI assistant that helps people find information."},{"role":"user","content":"Who is prime minister of India"},
{"role":"assistant","content":"As of my last update, the Prime Minister of India is Narendra Modi. He has been in office since May 2014. Please note that political positions can change, so it's always a good idea to verify with the latest sources."}]

completion = client.chat.completions.create(
  model="air_text_lab", # model = "deployment_name"
  messages = message_text,
  temperature=0.7,
  max_tokens=800,
  top_p=0.95,
  frequency_penalty=0,
  presence_penalty=0,
  stop=None
)
```



Azure Endpoint URL & API Keys?

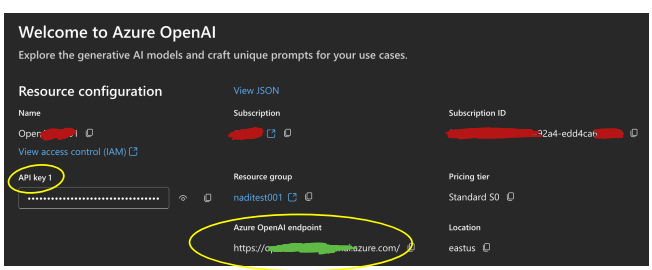
These are very important for making API calls



Step I; Azure Open AI API Calling

Unlike OpenAI call, we need

- Azure AI Api key
- Api version
- Client ID



Q// How to get code;

1. Setup Key, endpoint, in .env file

→ Deploy the chat model (use it in client)

Imp Code = Chat play Ground → View Code

```python
import os
from dotenv import load_dotenv
load_dotenv()
from openai import AzureOpenAI

AZURE_API_CLIENT = os.getenv("AZURE_API_CLIENT")
AZURE_API_KEY = os.getenv("AZURE_API_KEY")

client= AzureOpenAI(
    api_key=AZURE_API_KEY,
    api_version= "2025-01-01-preview",
    azure_endpoint=AZURE_API_CLIENT)
```

```python
chat_prompt = [
    {"role": "user",
     "content": "Who has won the most FIFA World Cups in soccer?"}
]
```

0.0s

```python
# Generate the response

response = client.chat.completions.create(
    model="my-first-gpt",
    messages=chat_prompt
)

print(response.choices[0].message.content)
```

History behind Azure OpenAI

- **2019 Partnership**: Microsoft and OpenAI formed a partnership, emphasizing the integration of OpenAI's technologies with Microsoft Azure.
- **$1 Billion Investment**: Microsoft announced an investment of $1 billion into OpenAI as part of their partnership agreement.
- **2021 Azure OpenAI Service Launch**: Launched to enterprise customers, allowing them access to OpenAI's AI models like GPT3, through Microsoft's Azure platform.
- **Expanding Capabilities**: The service has since expanded to include more sophisticated AI models, such as Codex and DALLE, enhancing its offerings for complex enterprise applications.
- **Enterprise Focus**: Designed specifically for enterprise use, Azure OpenAI aims to support businesses in incorporating AI into their operations securely and at scale.