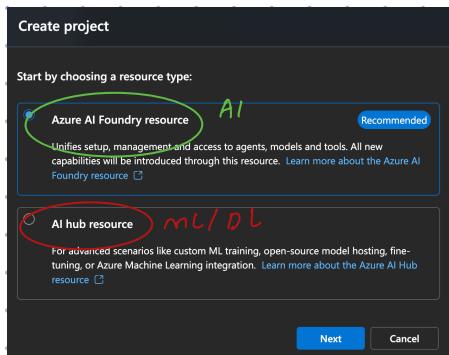


AZURE AGENTIC AI

Q, Difference b/w AI hub & Foundry?

AI hub - organisation workspace for many projects

AI Project - Single Report



In Summary:		
Feature	Azure AI Foundry Resource	AI Hub Resource
Focus	Agentic AI, copilots, generative workflows	Traditional ML & fine-tuning
Best For	AI apps, chatbots, automation agents	ML engineers, data scientists
Integration	OpenAI models, prompt flow, orchestration	Azure Machine Learning
Ease of Use	High – plug-and-play setup	Advanced – manual configuration
Recommendation	<input checked="" type="checkbox"/> Default choice for modern AI projects	Use for deep ML or custom model work

Not needed
Assistant API is the fundamental building block for Agentic AI.

► What it is:
A guided conversational AI designed to assist humans interactively.
It remembers context, follows instructions, and can use tools or data sources — but always in response to user guidance.

Key traits:

- Conversational, context-aware
- Can call APIs or tools (with permission)
- User remains "in control" of the flow
- Often implemented as Azure OpenAI Assistants, copilots, or GPTs

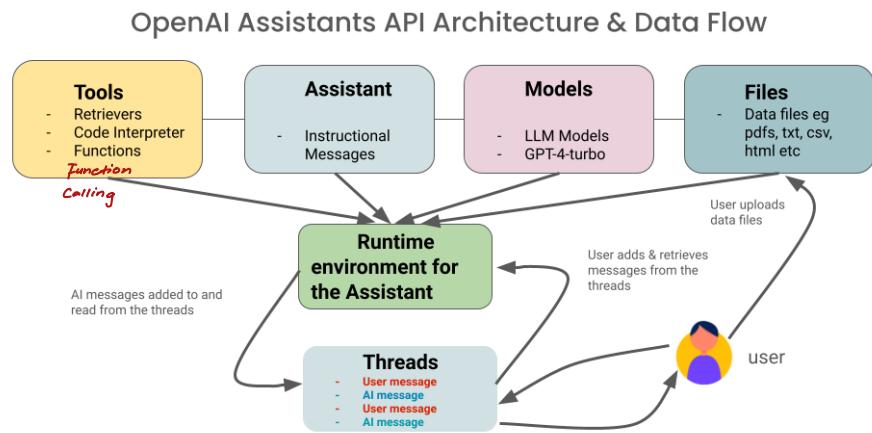
Example:
A Power BI Data Assistant that:

- Answers, "Show me top 5 customers by revenue."
- Then runs a Power BI query and returns a chart.

 It helps, but doesn't act on its own.

Use for: Copilots, support bots, analytics helpers.

Assistants API – Components / Terms		
Component	Description	Example
Assistant	A custom AI that uses Azure OpenAI tools to help users.	An AI customer support agent that helps users fix their devices using Azure OpenAI models.
Thread	A chat session between a user and the Assistant. It saves messages and automatically trims older ones to fit.	A user starts a chat to ask about their order status. The whole conversation is stored in the thread.
Message	A text, image, or file sent by the user or the Assistant.	User: "What's the status of my order #12345?" Assistant: "Your order #12345 has been shipped and will arrive by Friday."
Execution	When the Assistant starts working based on the chat contents. It uses tools and models to complete tasks and adds new messages to the chat.	A user asks the Assistant to summarize a document. The Assistant starts a run to read and summarize the document, then sends the summary.
Run	The detailed actions the Assistant takes during a Run. It shows the tools or messages used to complete tasks.	Step 1: The Assistant reads the user's question. Step 2: The Assistant uses a summarization tool. Step 3: The Assistant creates a summary and sends it to the user.
Run Step		



FUNCTION CALLING customized actions.

allows the assistant to execute external functions or APIs

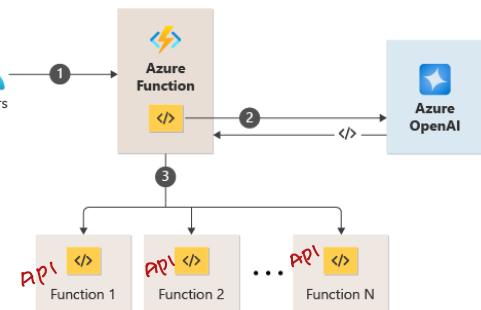
- Function Calling allows the assistant to execute external functions or APIs as part of a conversation.
- Instead of just responding with text, the assistant can:
 - Call a custom function you define (e.g., get weather, fetch customer info).
 - Return structured outputs to the user.
 - Chain multiple steps autonomously.

3. How It Works in Assistants

- You register functions with the assistant (name, inputs, outputs).
- The assistant decides when to call a function based on user input.
- Calls the function and integrates the response into the conversation.
- Can use multiple function calls in a single workflow, making it "action-capable" rather than just "chat-capable."

4. Example

- User: "Book me a flight to Delhi tomorrow morning."
- Assistant identifies it needs flight booking.
 - Calls your registered function `book_flight(destination, date, time)`.
 - Returns the result: "Flight booked! Your PNR is XYZ123."



weather calling Example

- Get weather API
- Def Get weather function
- Run function calling
 - function
 - Tool list (api may use)
- Create Assistant and thread.

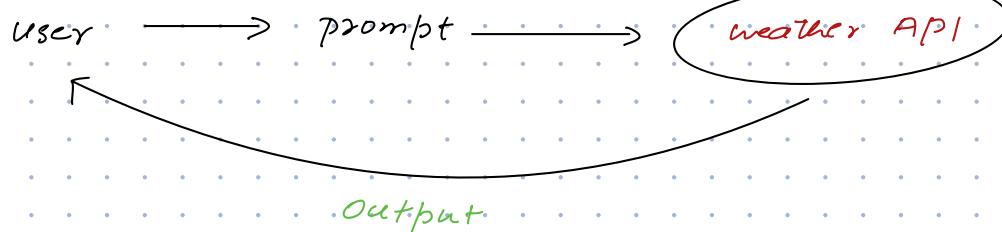
I/P

Latitude
Longitude

Check Code Later

In simple words;

Function call



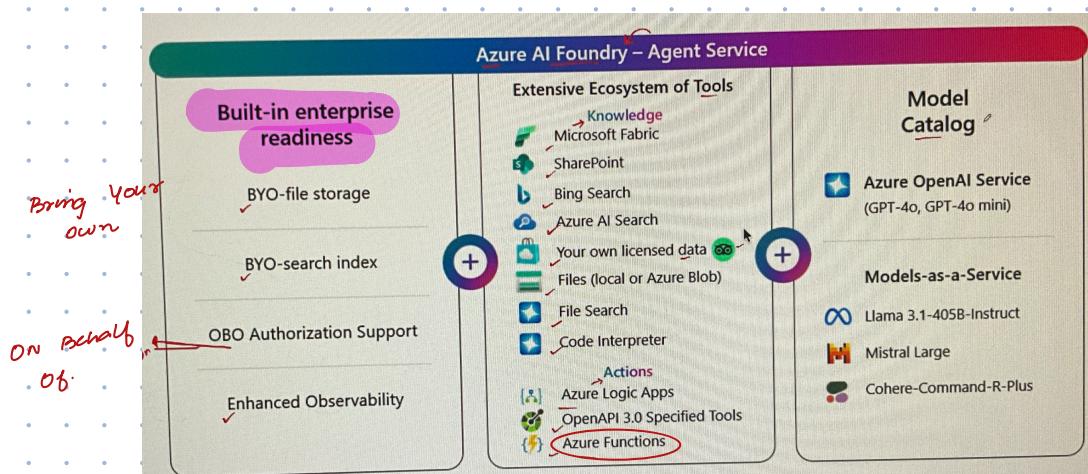
Imp:

AZURE AI AGENT SERVICE

fully managed service by Azure, making agent creation very easy

- Multi Modal
- Scalable and Secure

AZURE AGENT SERVICE ARCHITECTURE



AZURE AGENTS

fully autonomous agents

are pretty similar to Assistant APIs

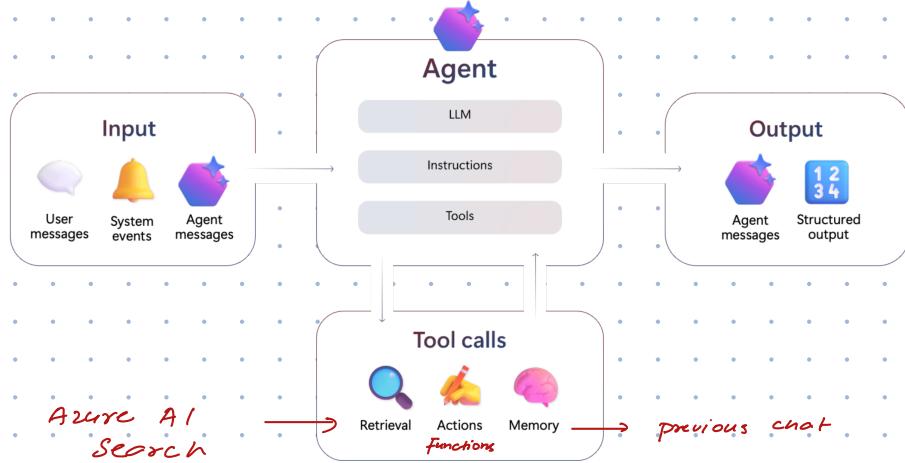
Old

Assistants

New

Agents

Components of AGENTIC AI

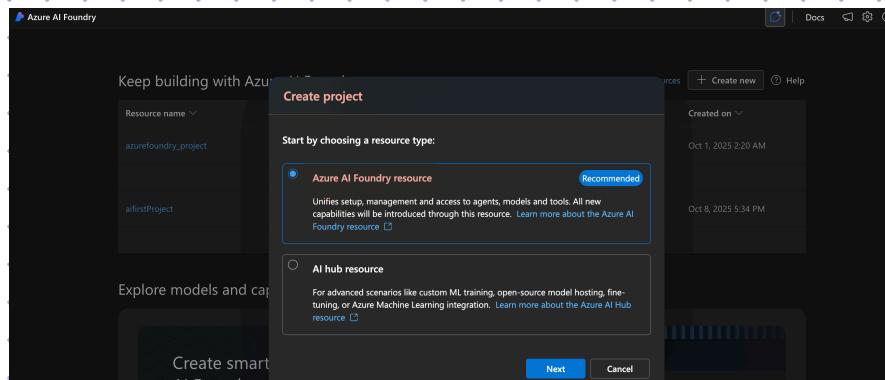


Quotas & Limits

2m Token limit

Limit Name	Limit Value
Maximum number of files per agent/thread (RAG)	10,000
Maximum file size for agents & fine-tuning	512 MB
Maximum size for all uploaded files for agents	300 GB
Maximum file size in tokens for attaching to a vector store	2,000,000 tokens
Maximum number of messages per thread	100,000
Maximum size of text content per message	1,500,000 characters
Maximum number of tools registered per agent	128

GETTING STARTED;



CREATE FIRST AGENT

The screenshot shows the 'Create and debug your agents' interface in the Azure AI Foundry. The left sidebar has 'Agents' selected. The main area shows a table of agents with one entry: **Agent425**. A green circle highlights the **+ New agent** button. The right panel shows the **Agent Description** section, which includes **Knowledge** (0) and **Actions** (0). Annotations in green highlight these sections with arrows pointing to the text: 'For adding Files' and 'Functions'.

Components

KNOWLEDGE

Add knowledge

Expand the knowledge an agent has access to by connecting data sources it can use for grounding responses. Currently only a single instance per each type of data source is supported. [Learn more](#)

Add a data source

Files Upload local files	Azure AI Search Search and indexing	Microsoft Fabric PREVIEW Accelerate data analysis with AI capabilities
SharePoint PREVIEW Securely integrate and manage internal data	Grounding with Bing Search Enhance model output with web data	Grounding with Bing Custom... PREVIEW Enhance model output with data from selected web domains
Tripadvisor Get travel data, guidance, and reviews	Morningstar Access up-to-date analyst research, expert commentary, and essential Morningstar data	

Add action

Give your agent the ability to perform tasks by giving it access to these actions. All actions are available to add through the API or SDK; once added to your code, you'll see them in the playground. [Learn more about actions](#)

Code interpreter Read and interpret information from datasets, generate code, and create graphs and charts.	OpenAPI 3.0 specified tool Trigger APIs with inputs and outputs schema defined using OpenAPI 3.0 spec.	Azure Logic Apps Automate workflows and connects apps, data, and services using Azure Portal's low-code Logic App editor
Custom function Configure a hook for your agent to call your custom function that can get real-time data, perform calculations, and call APIs	Azure Functions Run scalable, event-driven code that hooks into an Azure backend	Documentation samples
		Documentation samples

Agent Playground

Agents playground

+ New agent </> View code 🗑 Delete Create trigger PREVIEW

New thread Thread logs 6t → Tokens thread_mYETo7zYTkpWkr8qO6Erjc

who is tajamul khan

I'm sorry, but I couldn't find any specific information about a person named Tajamul Khan. Can you provide more context or details so that I can assist you better?

1s 6t AI quality 2 View Run Info

Type user query here. (Shift + Enter for new line)

Messages in the Agents playground are visible to anyone with access to this resource and using the API.

Setup

Agent ID asst.XXXMqMUaMd1zKODJrlH3Xhiv

Agent name Agent425

Deployment + Create new deployment gpt-35-turbo (version:0125)

Instructions You are a helpful agent

prompt

> Agent Description

Knowledge (0) Add

AZURE SDK - Python

projects client library

The Azure AI Foundry Projects client library for Python is a unified library that enables you to use multiple client libraries together by connecting to a single project endpoint.

- Install the project client library

```
Bash
pip install azure-ai-projects azure-identity
```

- Create a project client in code. Copy the Azure AI Foundry project endpoint from the Overview page of the project and update the connections string value.

```
Python
from azure.identity import DefaultAzureCredential
from azure.ai.projects import AIProjectClient

project = AIProjectClient(
    endpoint="your_project_endpoint", # Replace with your endpoint
    credential=DefaultAzureCredential())
# The AIProjectClient lets you access models, data, and services in your proj
```

After you create a client, you can use it to access models, run evaluations, and connect to other AI services. The next section lists the available client libraries and shows how to use them for specific Azure AI services.

- Using the project endpoint, you can:
 - Use Foundry Model, including Azure OpenAI
 - Use Foundry Agent Service
 - Run evaluations in the cloud
 - Enable tracing for your app
 - Retrieve endpoints and keys for external resource connections

Must
- Endpoint (project)

Imp. Azure CLI = used for dev cases

INSTALLATION;

Step I ; Create an Environment

Conda create -n myenv

Step II ; Install Homebrew (package manager)

Step III ; Install Azure CLI (Dev)

brew install azure-cli

Step IV ; Login to Azure

AZ login

Step V ; Create Agentic AI

1. Agent
2. Thread
3. Text (message)

Code

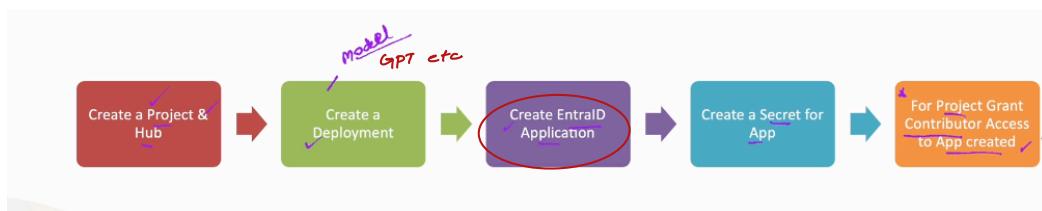
```
# Create and run agent
with project_client:
    agent = project_client.agents.create_agent(
        model="gpt-4o-mini",
        name="my-agent",
        instructions="You are helpful agent",
    )
    print(f"Created agent, agent ID: {agent.id}")

# Create a thread
thread = project_client.agents.create_thread()
print(f"Created thread, thread ID: {thread.id}")

# Create a message
message = project_client.agents.create_message(
    thread_id=thread.id,
    role="user",
    content="Who is PM of India?",
)
print(f"Created message, message ID: {message.id}")

# Run the agent
run = project_client.agents.create_and_process_run([
    thread_id=thread.id,
    agent_id=agent.id]
)
print(f"Run finished with status: {run.status}")
```

Workflow for Agent Creation



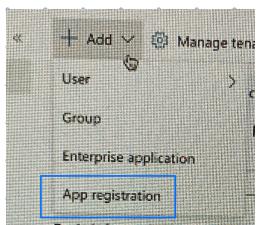
Imp: App needs contributor level access to project

Step I : Create a Hub

Step II : Create project inside hub

Step III : Create app for agentic AI

↳ Microsoft Entra ID



Name = application for agents

Set up Secret for app.

Step IV : Give Contributor Access of project to this APP

IAM

Add Role

Contributor

Application

AZURE AI FUNCTION TOOLS

Function Call ; Get - weather

```
api_settings.env — Edited
1. CLIENT_ID = "72256cc3-e46f-48ee-8ff1-f5738ce8da78"
2. CLIENT_SECRET = "P6Z80~GA3_~yzD6UhM4B~dcgUxBvtjLvi6Stdb2"
3. TENANT_ID = "95ea2a70-d5fe-4239-8648-6357a1f4a759"
4. PROJECT_CONNECTION_STRING = "eastus.api.azureml.ms;95717284-5de1-45e3-a461-5fe0db848584;aoairg;azure-ai-project-for-
agents"
5. OPENWEATHER_MAP_API_KEY = "1dd9e8f6c2c4aa1e9f2bd669da94b02a"
```

Steps = Client - deployed model
Tool - function call
Thread
Message

Code Interpreter Lets agent run and execute the code in the sandbox env.

- Understands code
- Fixes code
- Generates code.
- can be used for DA & visualize

Example 1 — Data Analysis

You ask an AI agent:

"Calculate the average sales per region from this CSV file."

The code interpreter:

- Writes a Python script using pandas
- Runs it inside a sandbox
- Returns the actual average numbers or a chart

Example 1 — Data Analysis

You ask an AI agent:

"Calculate the average sales per region from this CSV file."

The code interpreter:

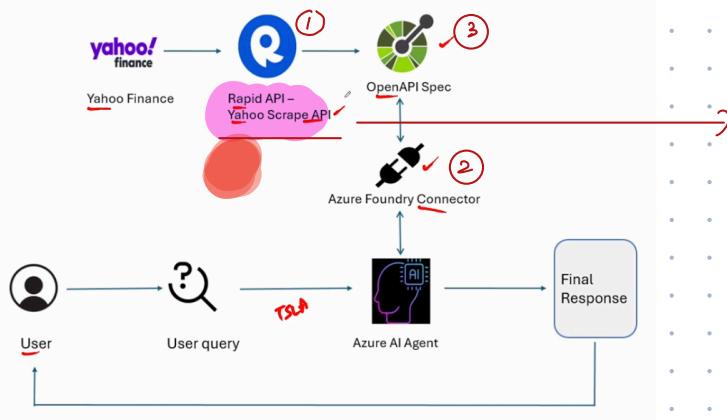
- Writes a Python script using pandas
- Runs it inside a sandbox
- Returns the actual average numbers or a chart

Open API 3.0

Standard for describing RESTful API's

	OpenAPI 3.0	FastAPI
What it is	A standard/specification for describing REST APIs	A Python web framework to build REST APIs
Purpose	Defines how an API should look (structure, endpoints, schemas)	Lets you build and run APIs quickly
Output	A JSON/YAML document (like openapi.json)	A running API server (with built-in OpenAPI docs)
Relation	FastAPI uses OpenAPI 3.0 internally	FastAPI automatically generates OpenAPI 3.0 documentation for your API
So:	<ul style="list-style-type: none"> OpenAPI 3.0 = the blueprint (standard). FastAPI = the tool that builds APIs following that blueprint. 	
They work together, but they are not the same.		

WORKFLOW OF YAHOO FINANCE



Rapid API is an API market place that helps

- find, connect
- manage API's all in one place.

Q: How to set up Rapid API?

Go to profile → console

1. new app;

The screenshot shows the 'Personal Account' section of the Rapid API console. A new application named 'Finance-APP' has been created and is highlighted with a red box. The 'Authorization Keys' table shows an entry for 'Authorization 1' with the value 'rapidapi.com'. A red arrow points from the 'Create App' button to the newly created application.

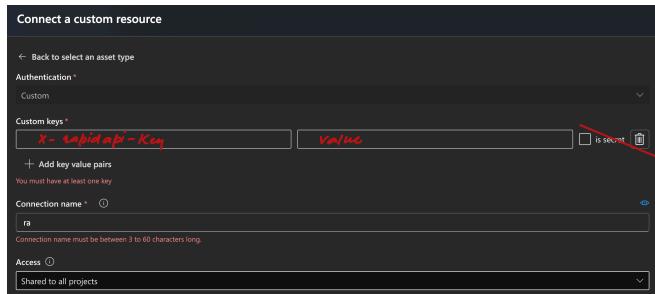
2. Add API in Rapid API

The screenshot shows the 'Discovery' section of the Rapid API console. A search bar at the top contains the text 'Search API's (Google Finance)'. Below the search bar, a list of results is shown, with the first result being 'Google Finance' (highlighted with a blue circle). The 'Code Snippets' and 'Example Responses' sections are visible on the right side of the interface.

3. Let's connect Rapid API and Azure AI;

Go to management center → New Connection

Custom Keys



Code Snippets Example Responses

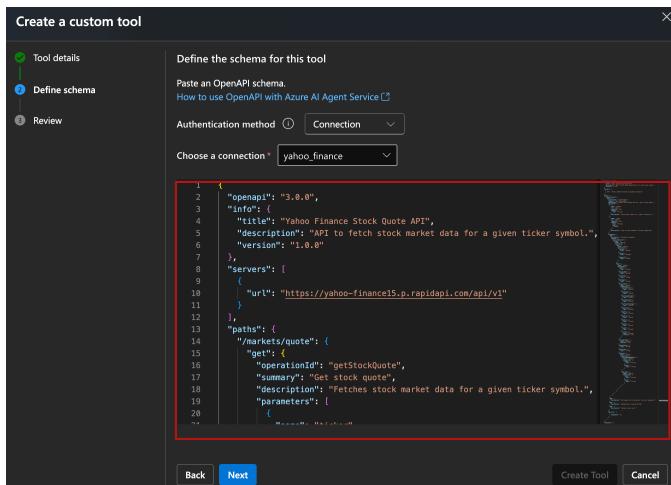
Target: Python Client: http.client

```
import http.client  
  
conn = http.client.HTTPSConnection("yahoo-finance15.p.rapidapi.com")  
  
headers = {  
    'x-rapidapi-key': "711b1b7b9ash19354df688967d3p1eb422jsn53af81a9a681",  
    'x-rapidapi-host': "yahoo-finance15.p.rapidapi.com"  
}
```

4. Let's use Open API 3.0

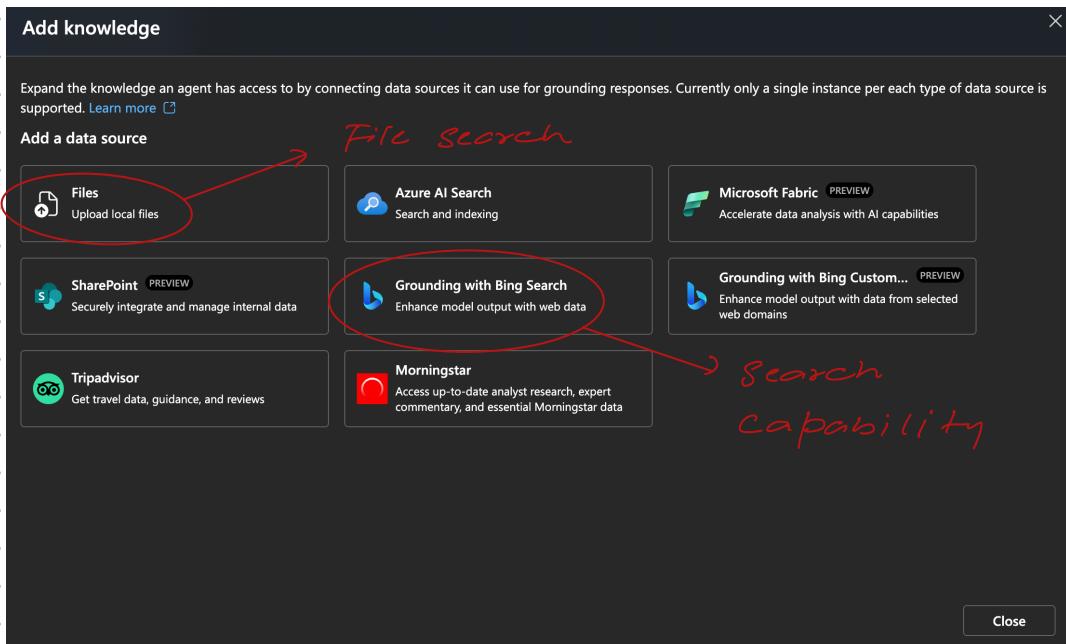
Go to Agent - Open API 3.0

Imp. we need json format



→ This comes from
custom json format

AZURE AI KNOWLEDGE TOOLS



Bing Search adds web search capacity
- Additional cost

Imp. There are added costs

File Search efficient search of docs / content

RAG = Doc + LLM

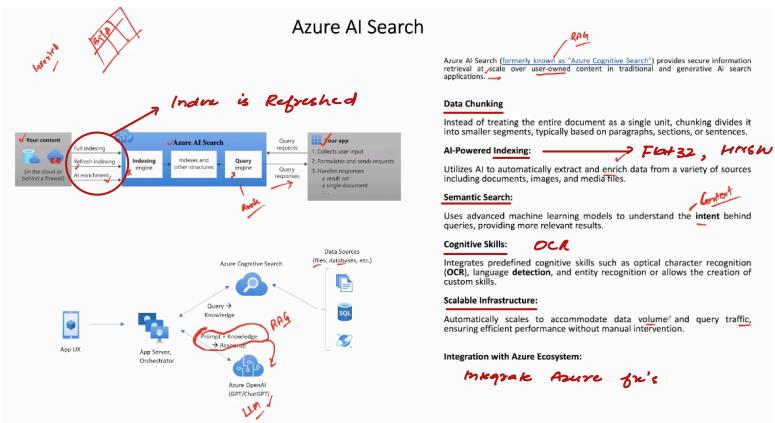
File Search = Search File Content

Feature	Azure RAG	Azure File Search
Output	AI-generated answer / summary	List of matching files or snippets
Data Handling	Can reason over multiple docs	Searches file contents only
Intelligence	Uses LLMs (GPT, OpenAI)	Index & keyword search only
Best for	Conversational AI, answering complex queries	Simple document lookup
Example	"What were the key risks in Q2 report?"	"Show all files with word 'risk' in them"

Imp. File search has added costs

0.10 \$ / GB / hour

AI SEARCH previously klas cognitive search takes care of (tokens, vectors, index, query)



Retriever
(Aug)

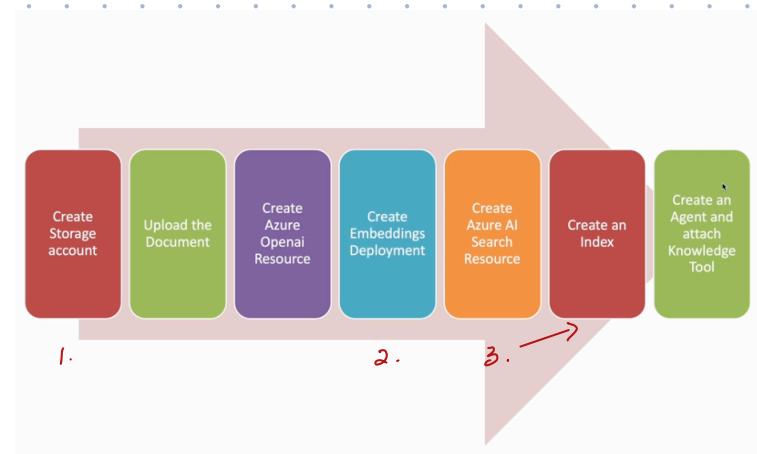
AI SEARCH VS. **FILE SEARCH**

Feature	AI Search <i>Enterprise level</i>	File Search <i>Individual level</i>
Definition	AI Search connects your agent to enterprise data indexed in Azure AI Search (RAG-based system). It enables intelligent retrieval from large document collections.	File Search lets your agent search within specific uploaded files (like PDFs, Word docs, CSVs) stored in a vector database .
Data Source	Enterprise-level document index (e.g., SharePoint, Azure Blob, Cosmos DB).	Individual uploaded documents (stored in Azure or local).
Storage	Uses Azure AI Search index + embeddings for semantic retrieval.	Uses vector embeddings created for each uploaded file.
Scope	Searches across multiple datasets or sources.	Limited to specific files linked to the agent.
Use Case	Ideal for company-wide knowledge bases, FAQs, or enterprise document search.	Ideal for question-answering over a single or few documents.
Performance	Pre-indexed → faster queries and scalable .	Parsed on upload → slower for large data, less scalable .
Integration	Integrated with RAG (Retrieval-Augmented Generation).	Simplified retrieval using file embeddings.

Key Interview Points

1. **AI Search** = "Enterprise-scale search using Azure Cognitive Search with RAG."
2. **File Search** = "Document-level search based on uploaded file embeddings."
3. **AI Search** is better for scalability and multi-source knowledge.
4. File Search is quick to set up — best for demos, small datasets, or user-specific data.
5. In real projects, **AI Search** replaces File Search once the data grows beyond individual files.

WORKFLOW OF AI SEARCH AGENT



1. Storage Account → Azure platform
2. Embedding Deployment → Azure AI
3. AI Search - Index → Azure platform

CREATING FRESH DESK APP

💡 What is Streamlit?

Streamlit is an open-source Python framework that allows you to quickly build interactive web apps for data science and machine learning — without needing any web development skills.

Fresh Desk is a platform used for customer support tasks

💡 What Does a Freshdesk Agent Do?

A Freshdesk Agent is a support representative who uses the Freshdesk platform to handle customer queries, issues, and requests.

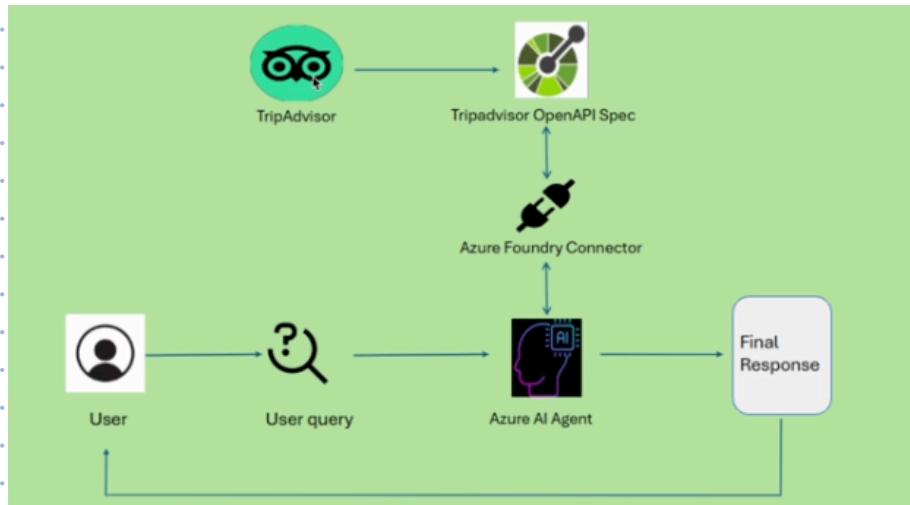
⚙️ Main Responsibilities

Function	Description
🎫 Ticket Management	Respond to and resolve customer tickets from email, chat, or social media
💬 Customer Communication	Interact directly with customers to solve issues
🧠 Knowledge Base Updates	Create and update help articles or FAQs
🤝 Collaboration	Work with other agents or departments to resolve complex cases
📊 Performance Tracking	Use dashboards to monitor response times, satisfaction scores, and workloads

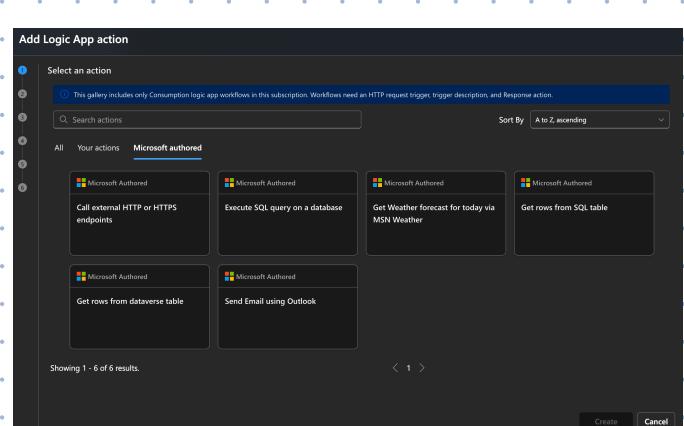
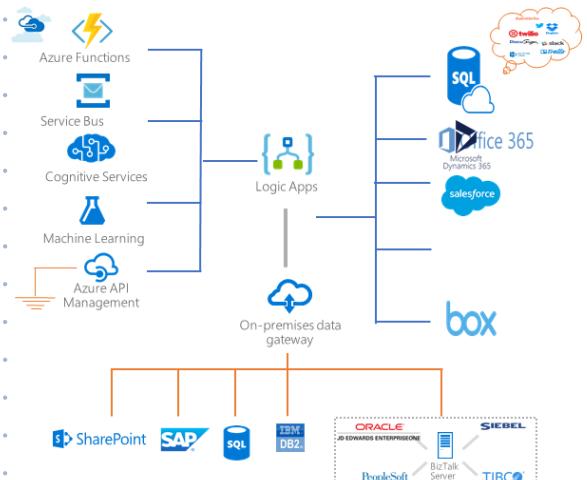
TRIP ADVISOR ; Open Source platform where users share their feedback etc.

Tripadvisor in Azure Context

- Data Source:**
 - Tripadvisor provides data via APIs (for reviews, ratings, hotel info, etc.).
- Use in Azure:**
 - You can connect Tripadvisor API to Azure AI or Azure Data Services.
 - For example:
 - Azure AI Agents can fetch hotel reviews or ratings for users.
 - Azure Data Factory / Synapse can ingest Tripadvisor data for analytics.
- Example Flow:**
 - Authenticate with Tripadvisor API → Call endpoints for hotels/restaurants → Store data in **Azure SQL / Cosmos DB / Blob** → Use **Power BI / Azure AI** for analysis or building a chatbot.
- Purpose:**
 - Build travel recommendation systems, dashboards, or AI agents that provide reviews and insights from Tripadvisor data.



AZURE LOGIC APPS is a low code / no code platform for automating tasks by integrating and connecting services and apps.



Semantic KERNEL

Microsoft's own Langchain

Yes ✓ — Semantic Kernel is Microsoft's version of LangChain.

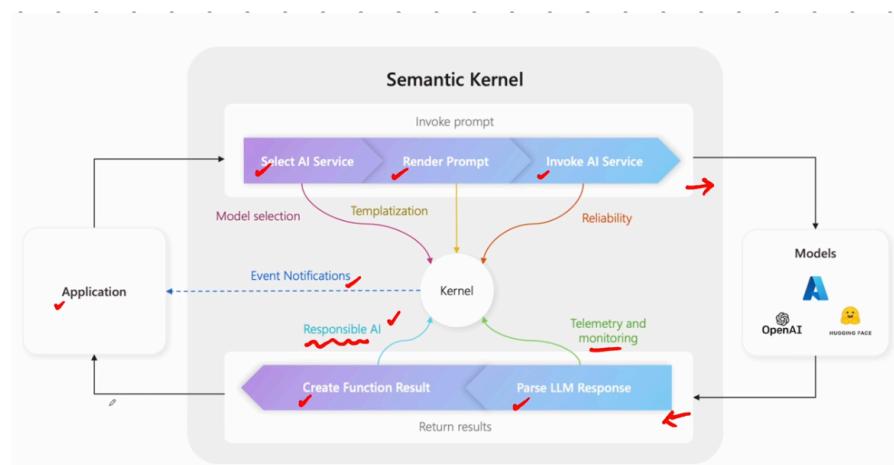
Both are frameworks to build AI agents or apps that connect:

- LLMs (like GPT models)
- Your data
- External tools or APIs

Difference in short:

Feature	Semantic Kernel	LangChain
Creator	Microsoft	LangChain AI
Language Support	C#, Python, Java	Python, JS
Integration	Deeply integrated with Azure & Microsoft tools	Works with many open-source LLMs & APIs
Focus	Enterprise + Azure ecosystem	Broad open-source community

👉 In simple words: *Semantic Kernel = LangChain for Azure/Microsoft ecosystem*.



- Plugins are collections of functions (native code, OpenAPI, or workflows)
- They must include **semantic metadata** so the AI can understand and use them effectively.
- **Three plugin types** are supported: Native (C# methods), OpenAPI-based (e.g., Swagger-defined APIs), and Logic Apps (Azure workflows).
- Plugins enable the AI to perform **retrieval tasks** (e.g., querying data) and **task automation** (e.g., sending emails or triggering actions).
- **Importing plugins** is easy: use built-in methods like ImportPluginFromType (for native) or ImportPluginFromOpenApiAsync (for OpenAPI).

