Copilot Studio and Azure AI Workshop

Lab 6: AI-Powered Contract Search & Insights using Azure Search.

Hands-on Lab Step-by-Step Guide

April 2025

# Lab Overview and Pre-requisites

Learning Objectives

In this lab, participants will learn how to store and index customer contracts in Azure Blob Storage and make them searchable using Azure AI Search. They will configure the search index, apply AI-driven semantic search, and retrieve key contract details such as payment terms, SLAs, and renewal dates.

Pre-requisites

- You will need credentials to a demo tenant that has Copilot Studio and AI Builder trial enabled.
- You will need access to Azure portal with an Azure subscription
- You should have completed Lab 5 (As some of the Azure resources are created in Lab 5)
- Access to Lab 5 Assets (If all 4 contract documents are not already uploaded to Blob in Lab 5)

Lab 6: Enhance the agent to use Azure AI Search to index a large set of customer contract documents and enable semantic search for insights.

1. Sign into **Azure Portal** (https://portal.azure.com) and open the **storage account created in Lab 5** and click on **Containers** under **Data storage** and select your container where all contracts are stored.
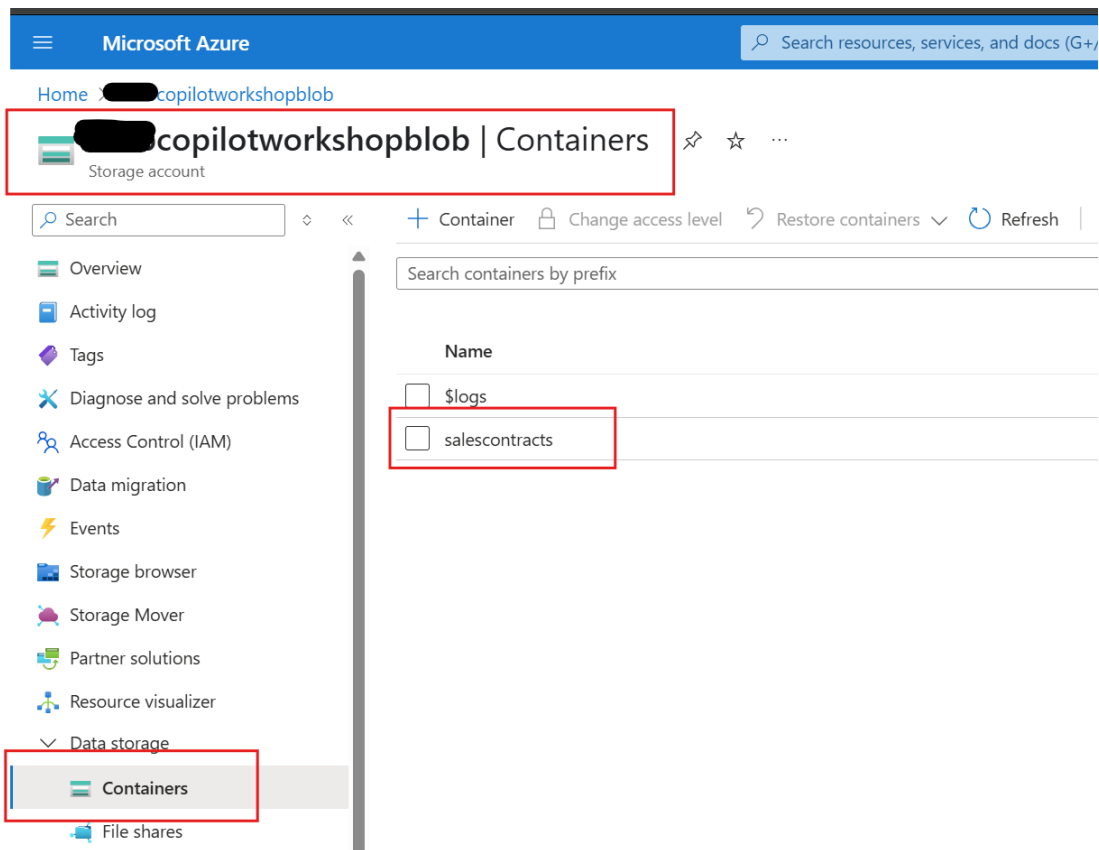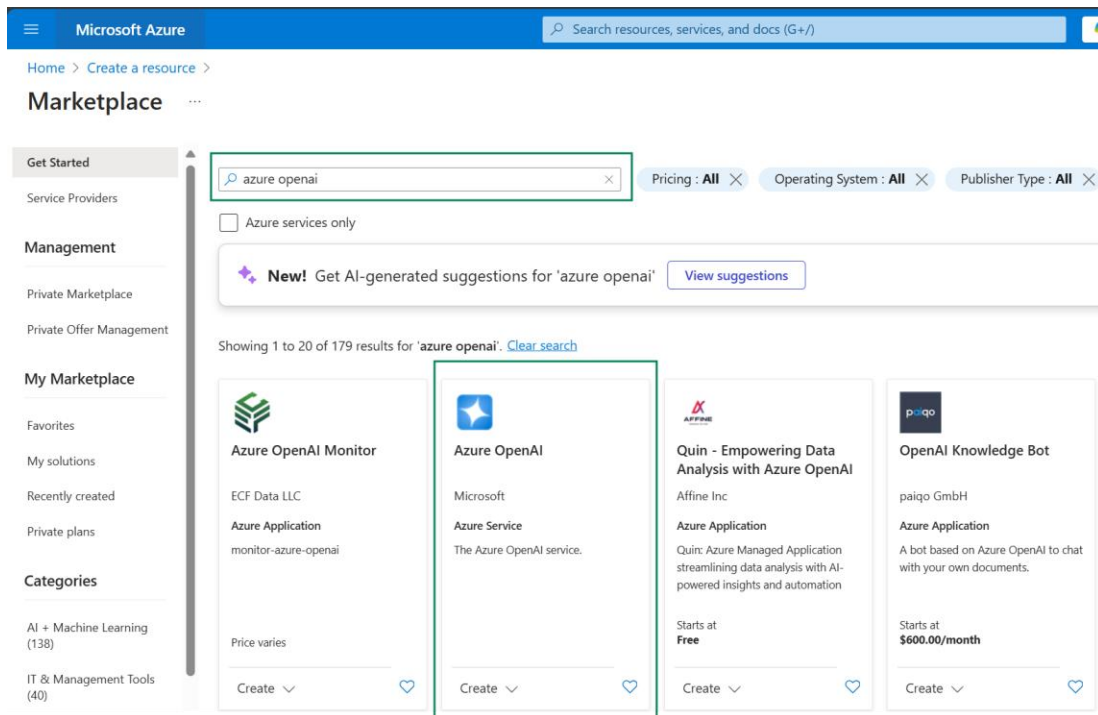

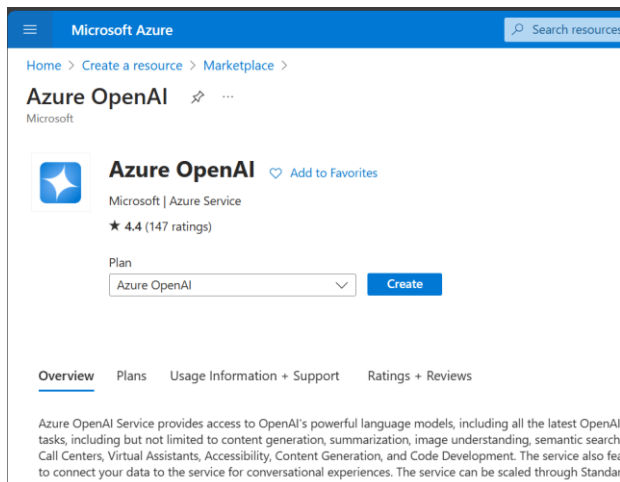2. Ensure your container has all 4 sample contracts available. If not, upload the contracts from **Lab 5 assets** folder
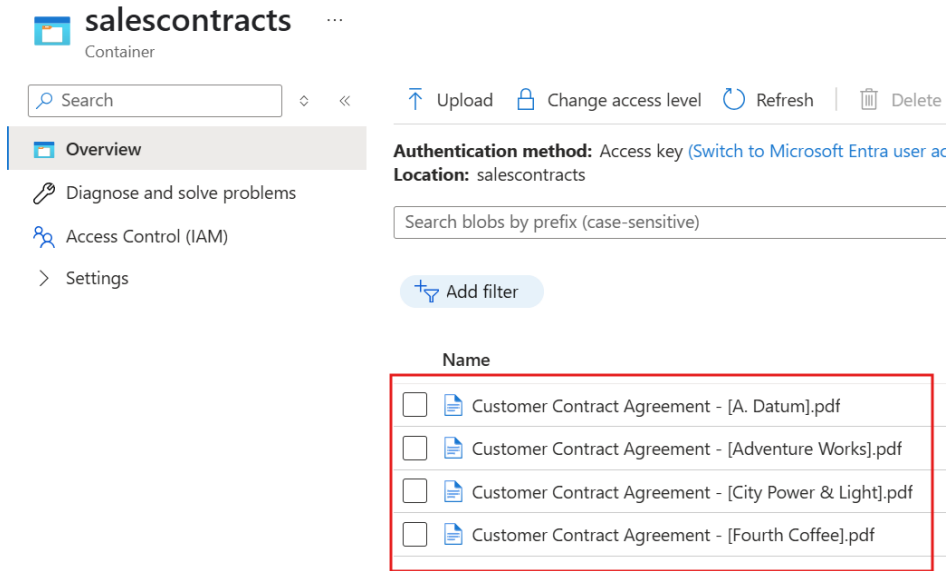
3. Go back to the **Azure Portal home screen** and click on **Create a resource**
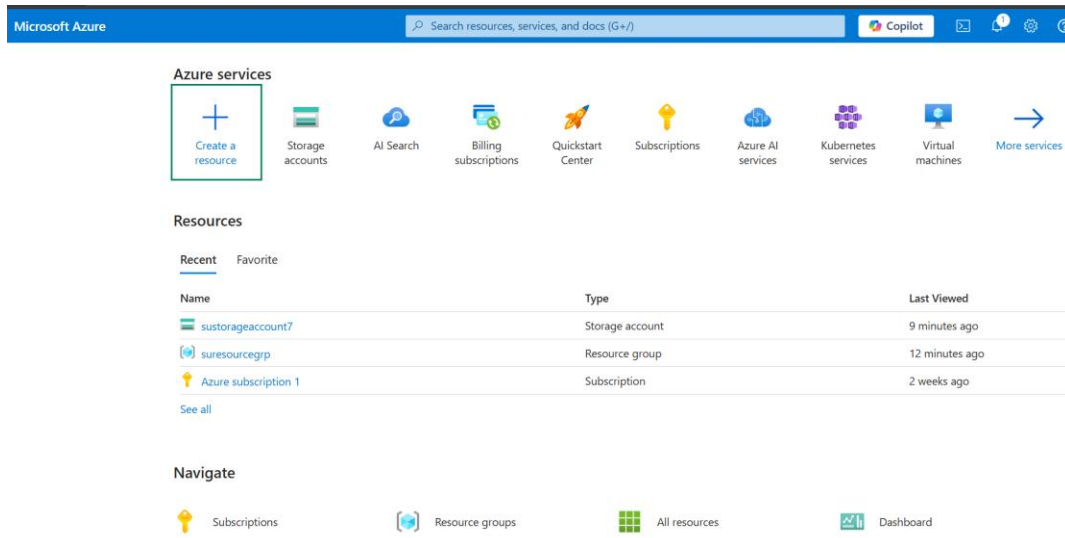
4. Search for **Azure OpenAI** and select it to create a new **Azure OpenAI resource**

5. Select **Create**

6. Select the available **resource group**, **region**, provide a **resource name**, select **pricing tier** as S0 and click on **Next**

Open

7. Click **Next** > **Next** and **Create.** The **Azure OpenAI** resource should get deployed in few minutes.

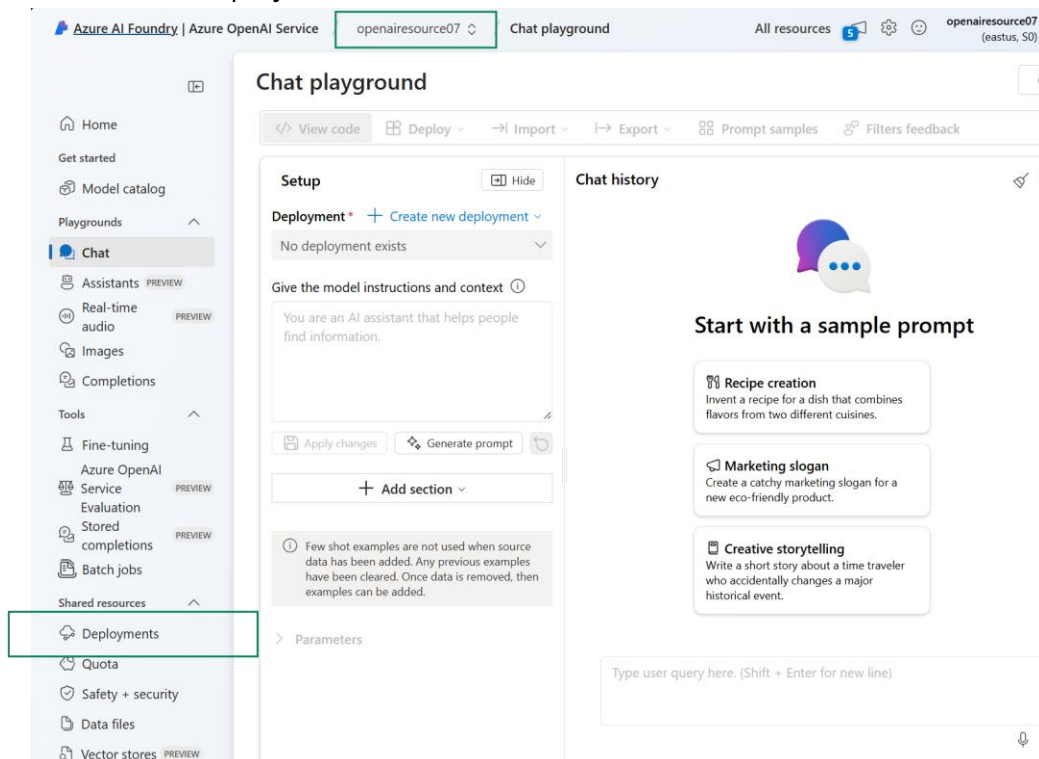8. Go back to the Azure portal home screen and select the newly created Azure OpenAI resource

9. Inside the Azure OpenAI resource, select **Explore Azure AI Foundry portal** and this should open the **Azure AI Foundry** portal in new tab.
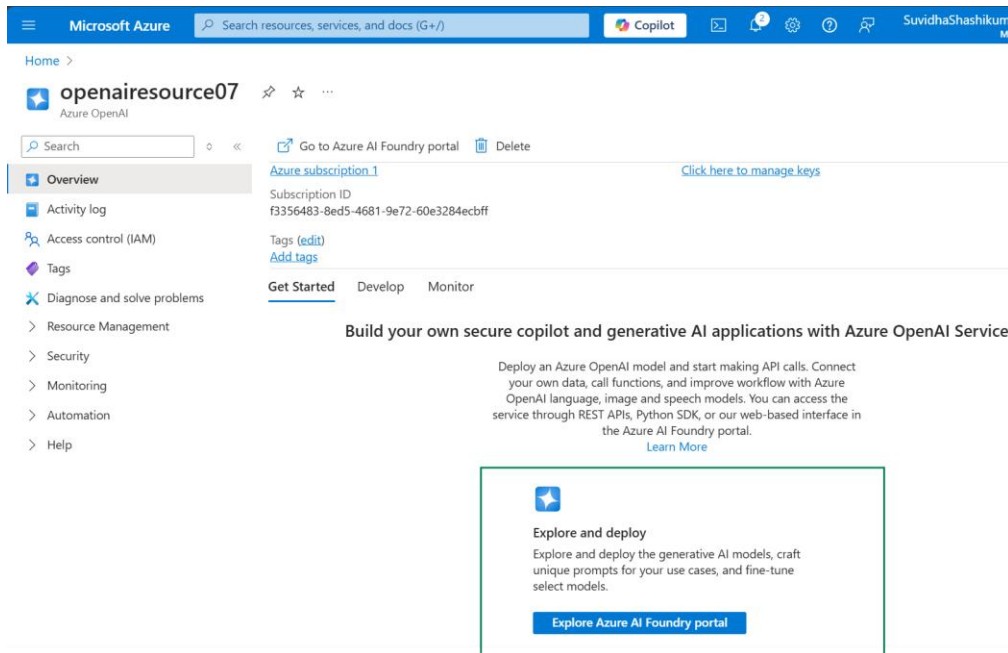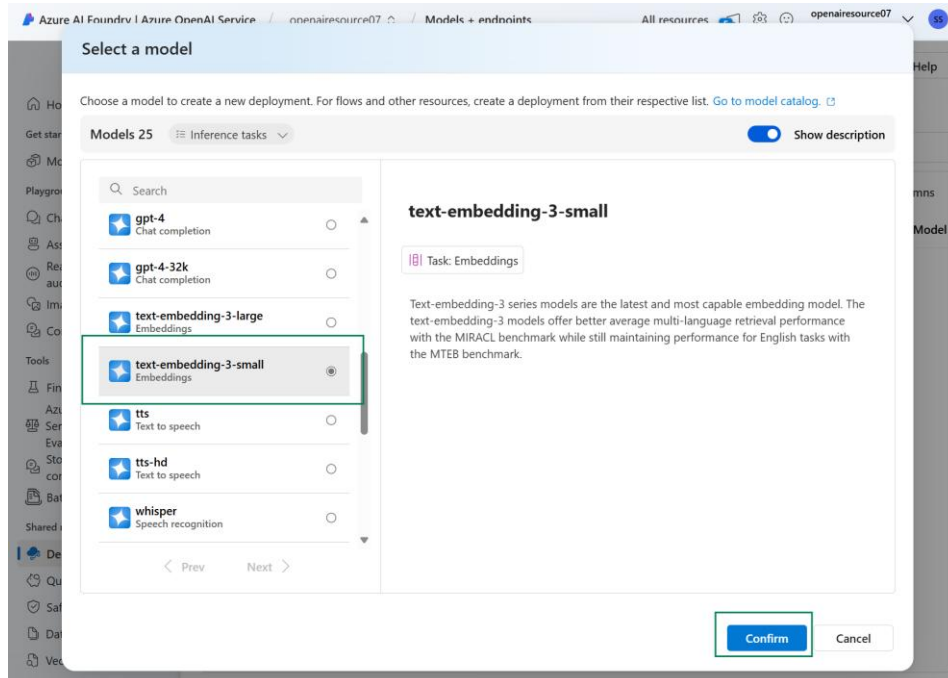
10. In the Azure AI Foundry, make sure your Azure OpenAI service is selected at the top. Next, click on **Deployments**
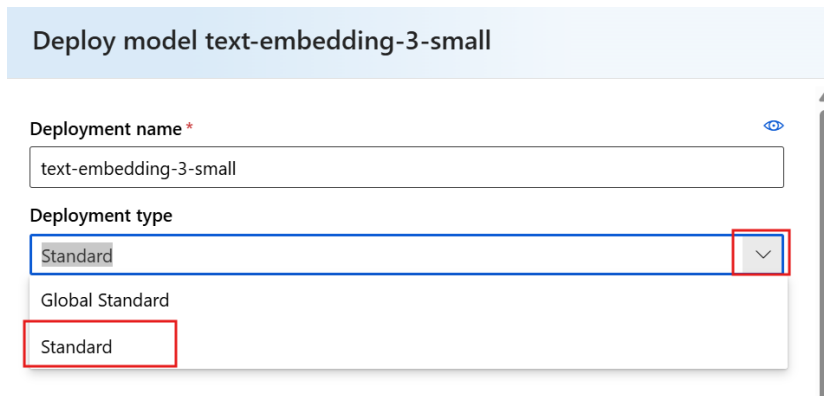

11. In **Model deployments** screen, Select **Deploy Model** > **Deploy base model**.
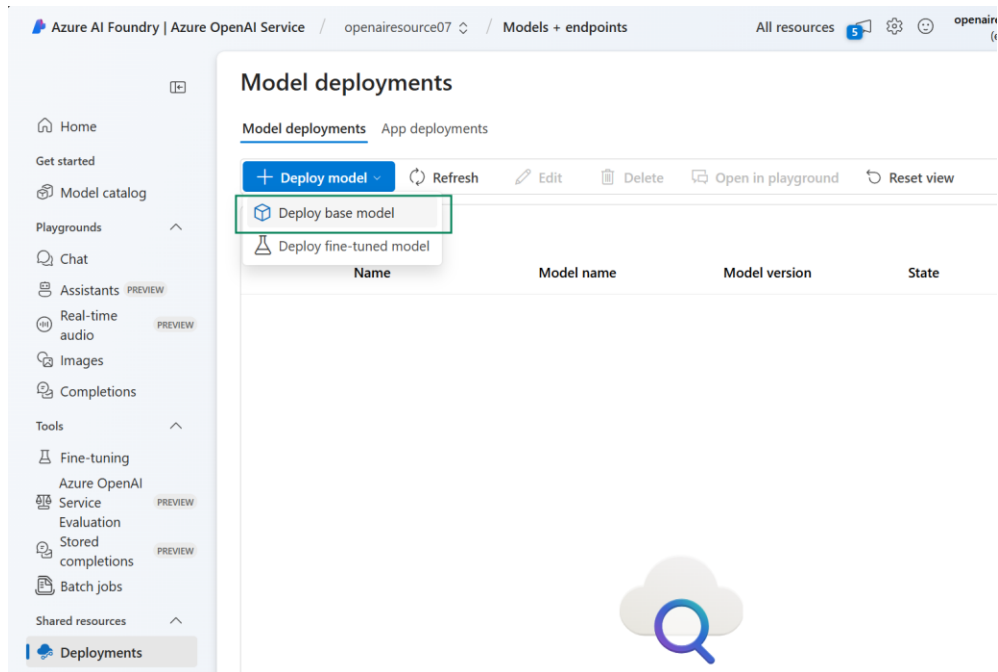
12. In the pop-up window, scroll down to select **text-embedding-3-small** and click on **Confirm.**

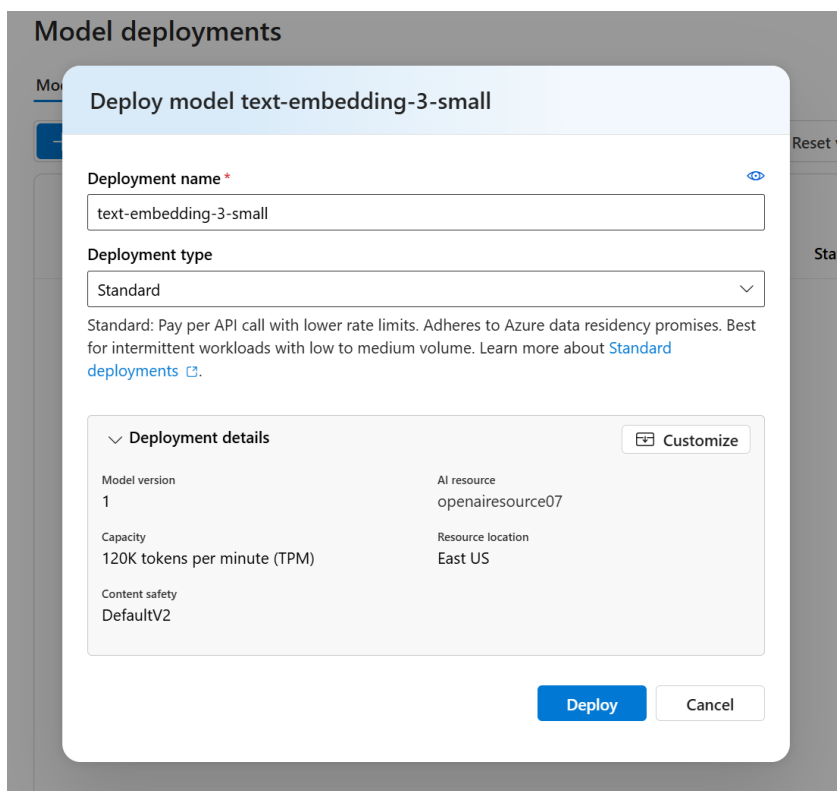13. Change the **Deployment type** to **Standard.**

Deploy model text-embedding-3-small

Deployment name *

text-embedding-3-small

Deployment type

Standard

Global Standard

Standard

14. Select **Deploy** to deploy the text embedding model to Azure OpenAI service.

Model deployments

Deploy model text-embedding-3-small

Deployment name *

text-embedding-3-small

Deployment type

Standard

Standard: Pay per API call with lower rate limits. Adheres to Azure data residency promises. Best for intermittent workloads with low to medium volume. Learn more about Standard deployments ⤴.

∨ **Deployment details**   ⊞ Customize
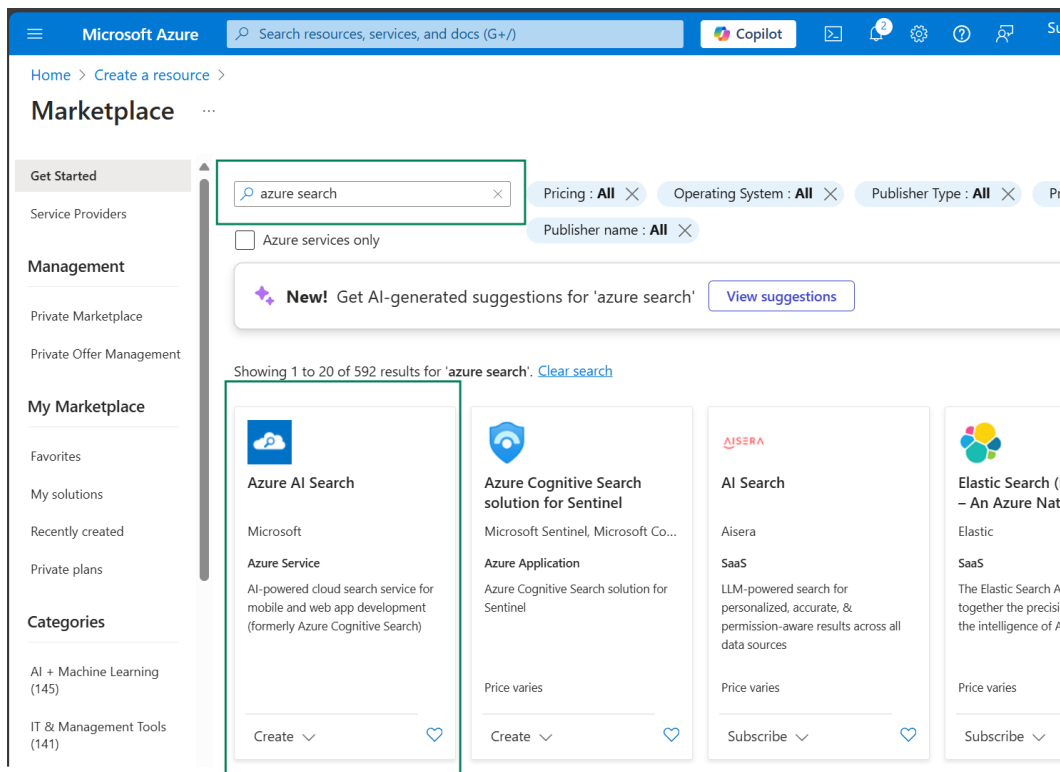
Model version   AI resource
1   openairesource07

Capacity   Resource location
120K tokens per minute (TPM)   East US
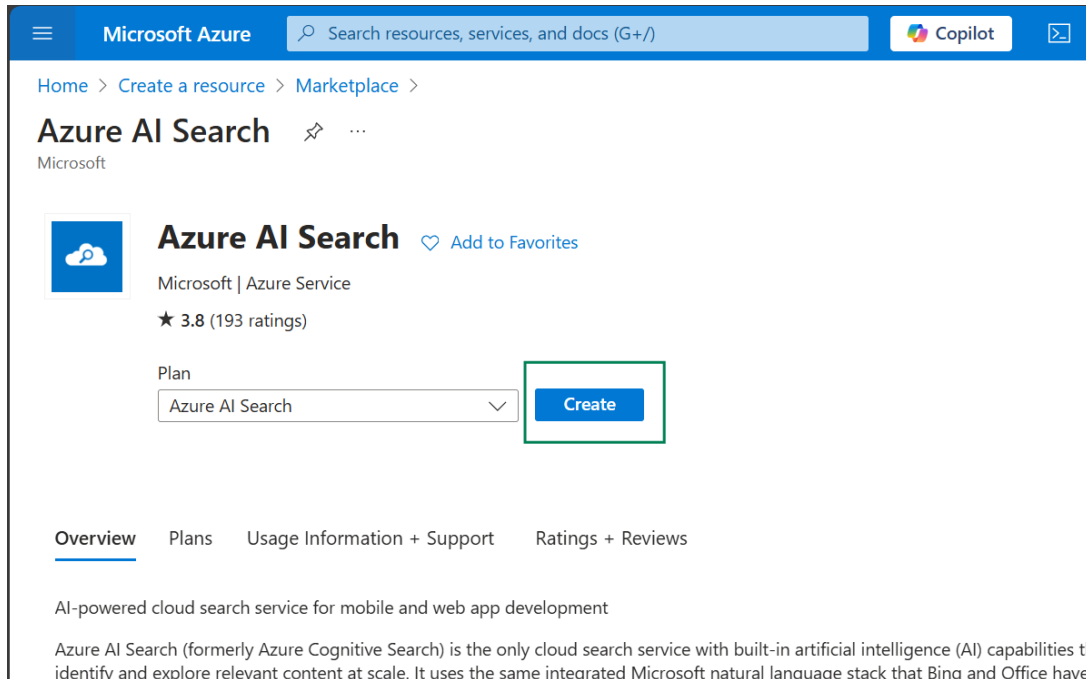
Content safety
DefaultV2

Deploy   Cancel

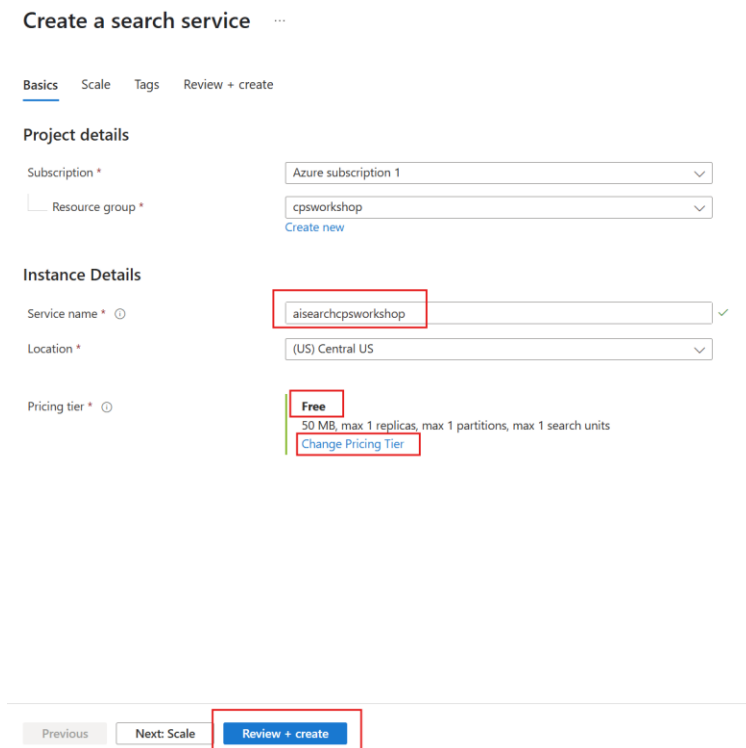15. Go to Azure portal and select **Create a resource**
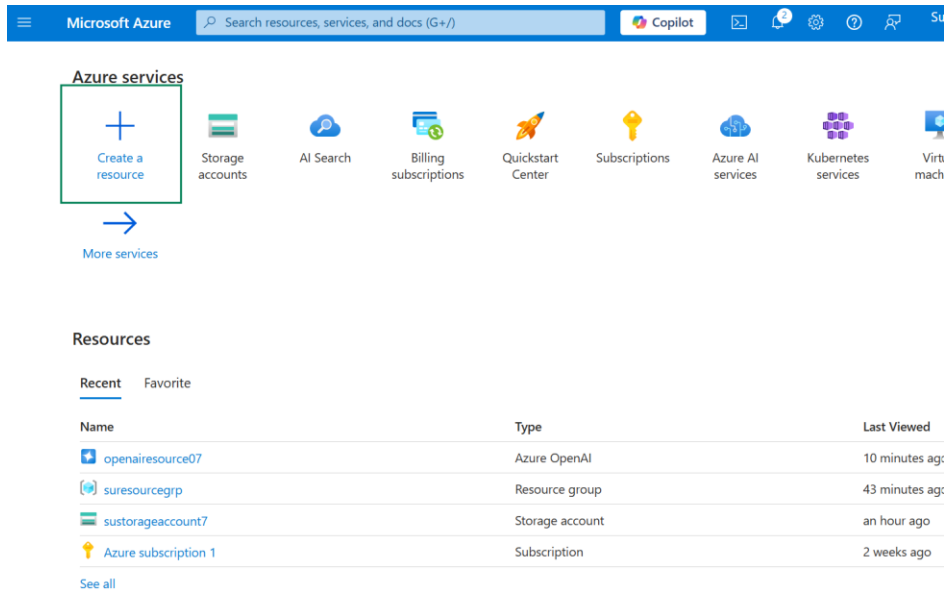
16. Search for *Azure Search* and select Azure AI Search
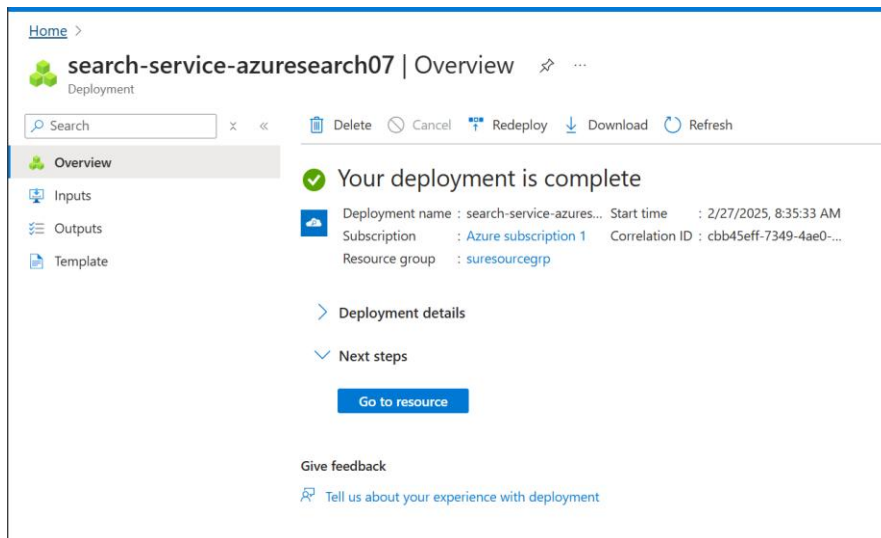

17. Select **Create**

18. In the Create Search Service window, select your **resource group**, provide a **name** for search service, select **region** and click **Change Pricing Tier** to <u>select **Free pricing tier**</u>. Next, select **Review + Create.**
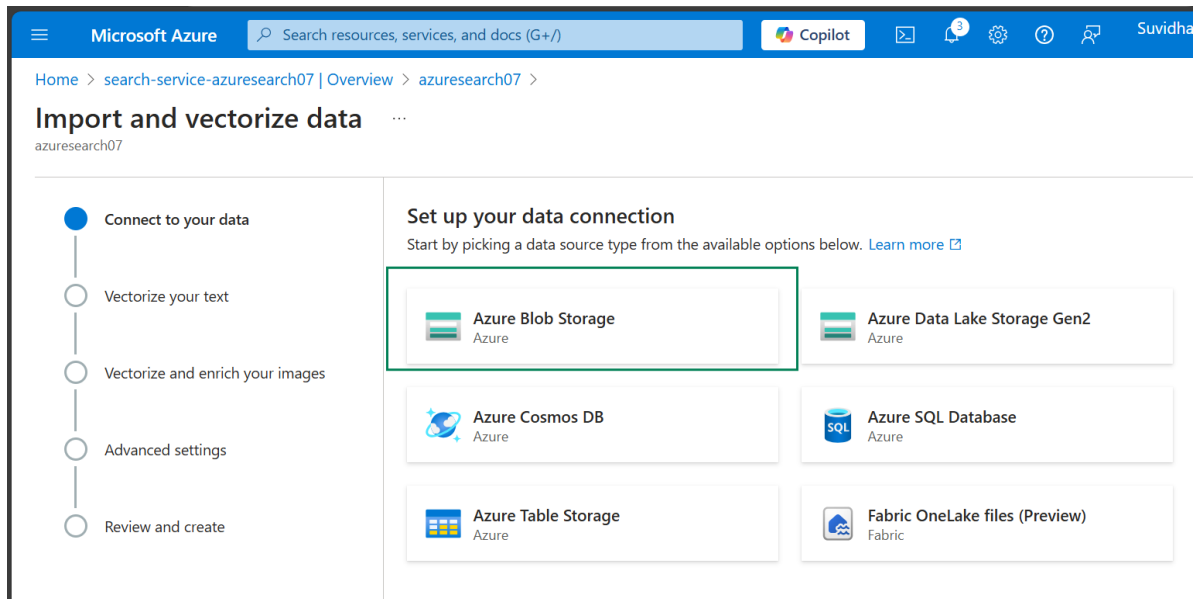
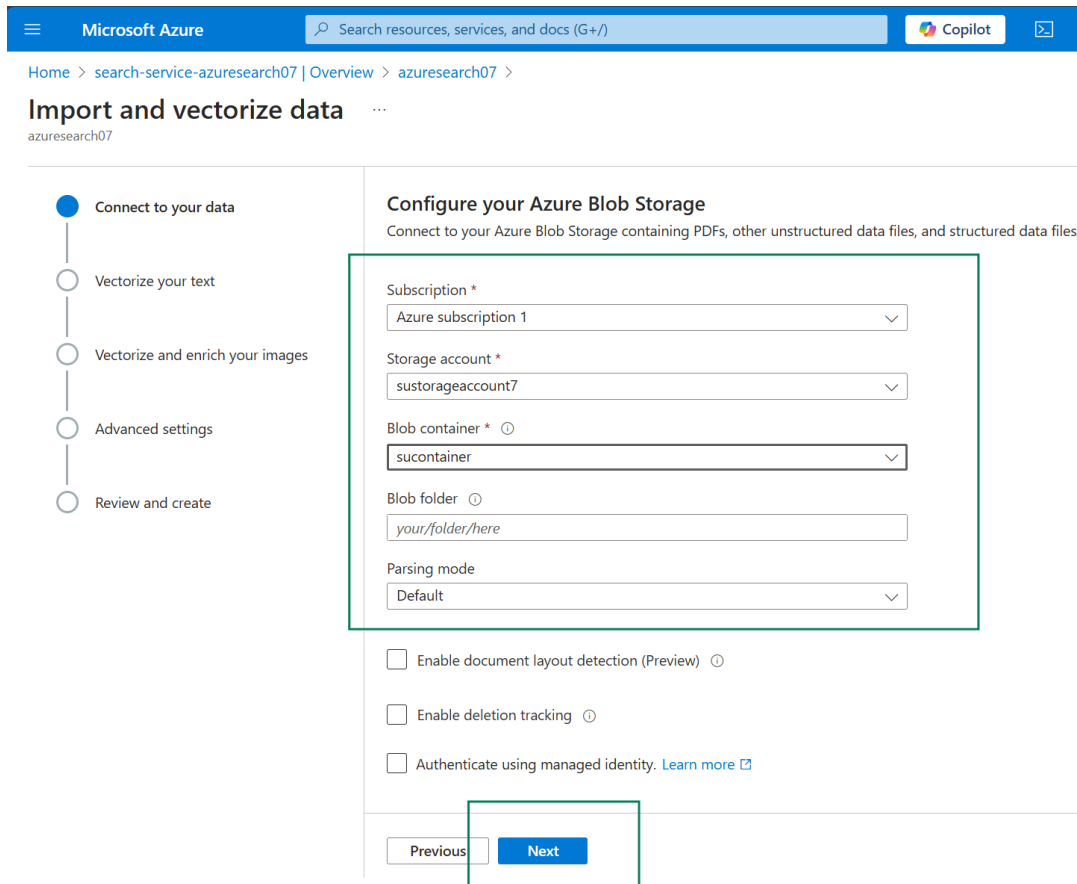19. Select **Create** again to deploy Azure search service. Once deployed, select **Go to resource.**
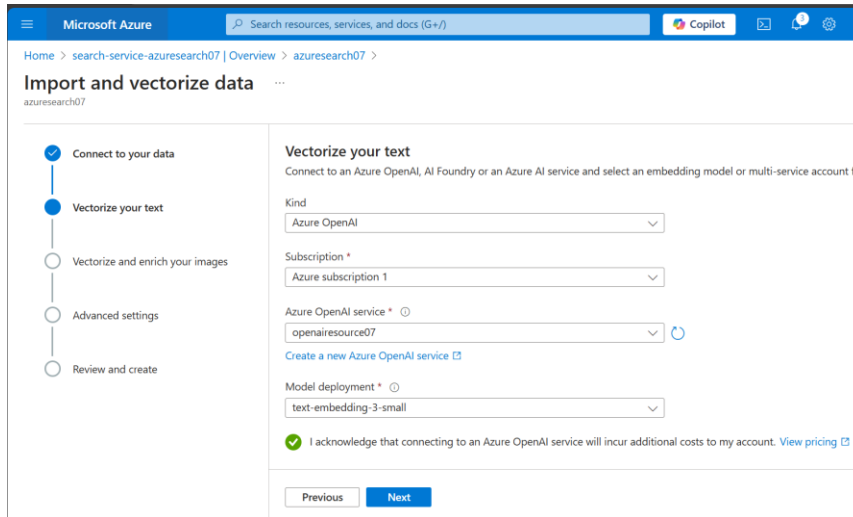

20. In the Azure search service, select **Import and vectorize data**
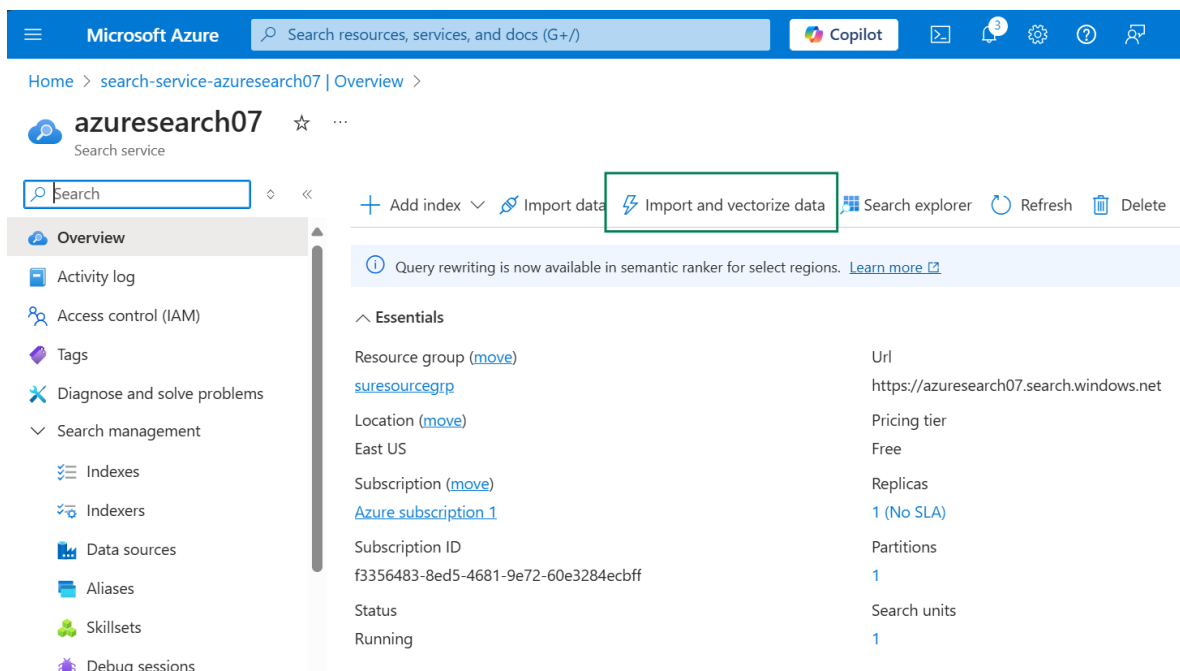
21. Select **Azure Blob Storage**

22. Select the subscription, storage account and container where all contract documents are stored, and click on **Next**
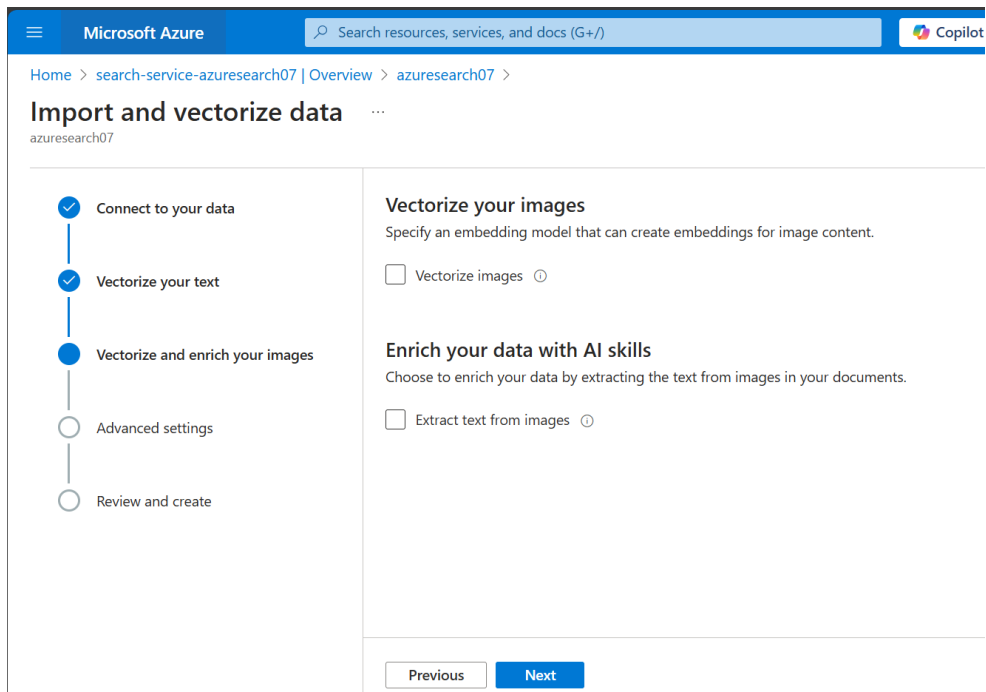
23. In the next screen, choose Kind as Azure OpenAI, your Azure subscription, newly created Azure OpenAI service, the text embedding model values. Click on the box to acknowledge that connecting to an Azure OpenAI service will incur costs and select **Next**

24. Select **Next**

25. Select **Next**

26. Update the **Objects name prefix** to **indexcustomersalescontract** and click **Create** to create a vector index.

27. You should see a success message –
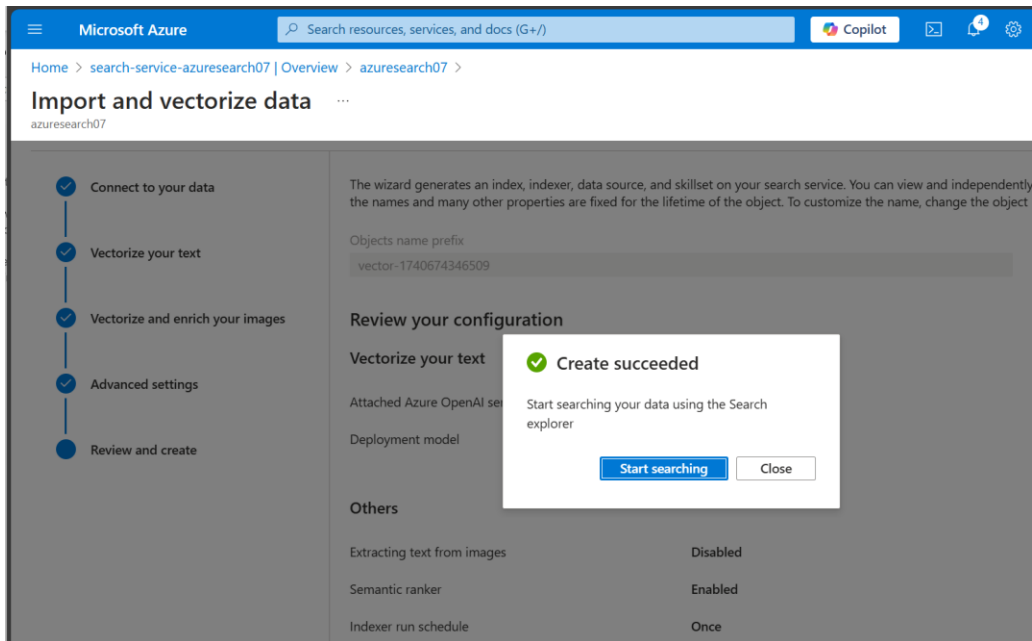
28. Go to **Home** page and open the **Azure search service** again. Copy the endpoint URL into a notepad.
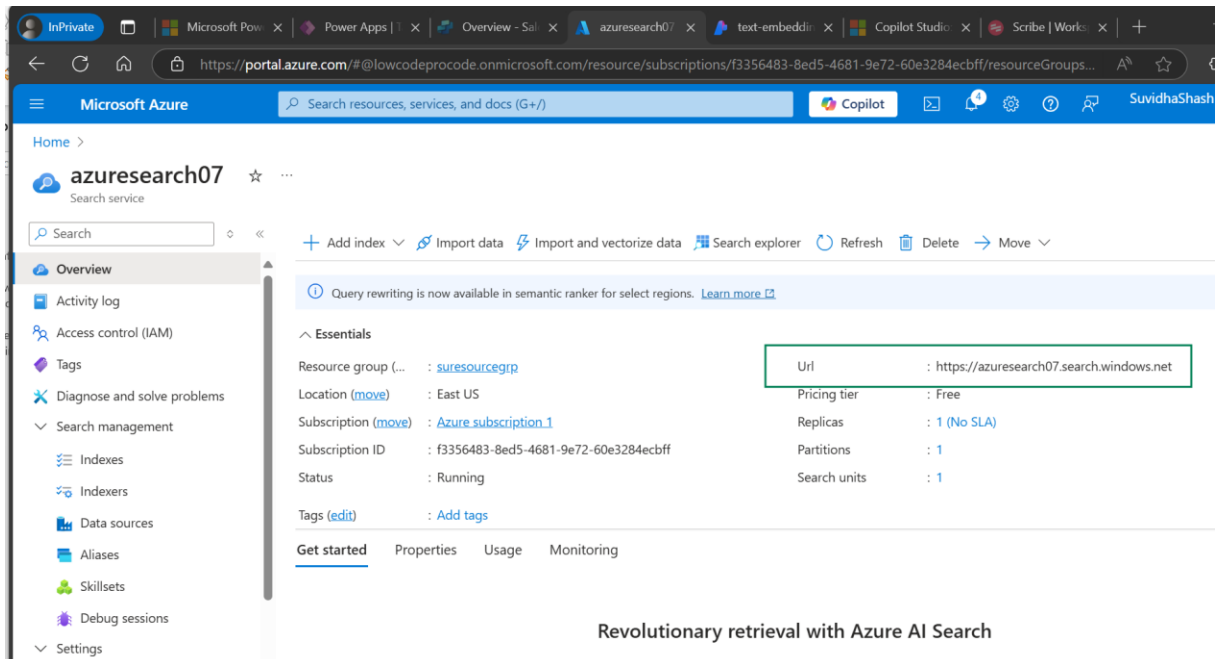
29. In the same window, under **Settings** select **Keys** and copy one of the 2 keys into a Notepad.
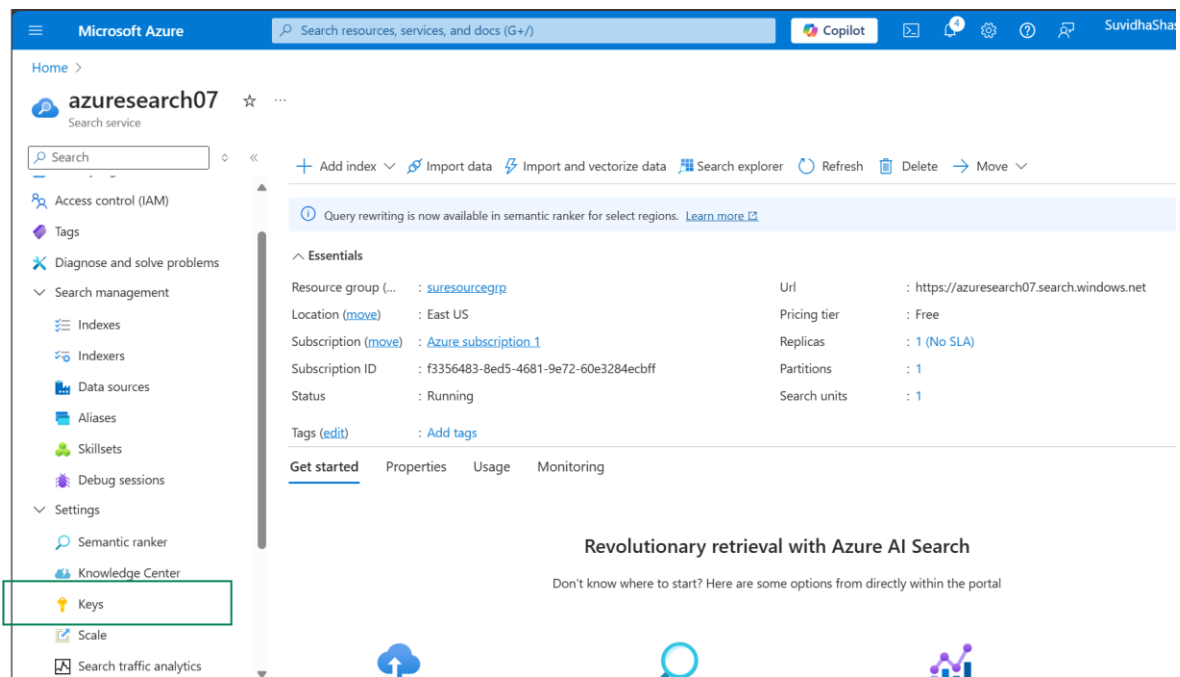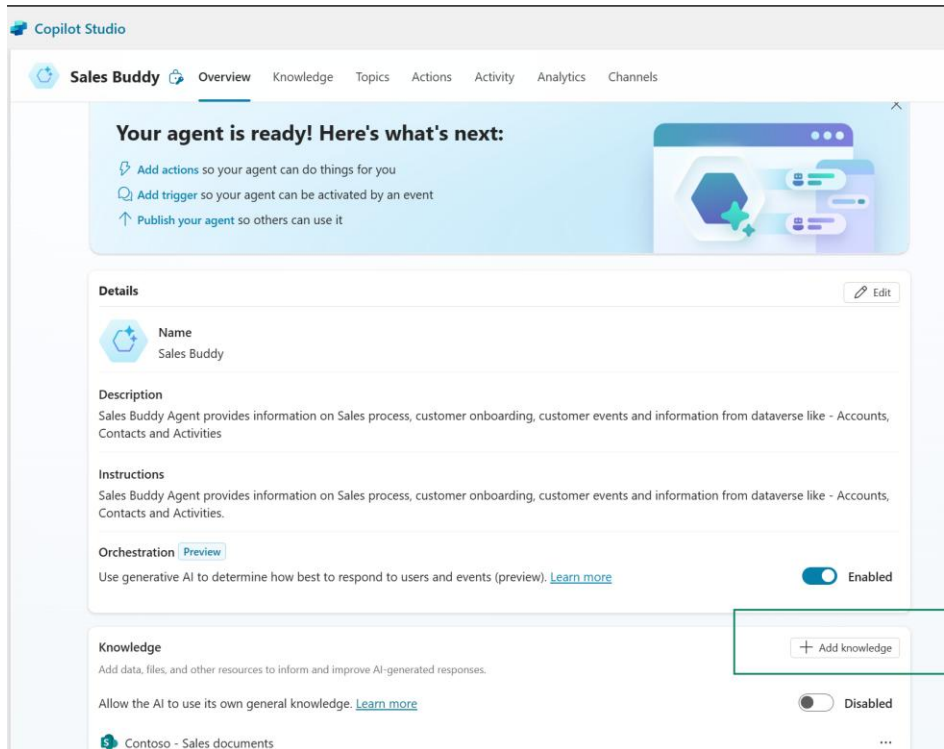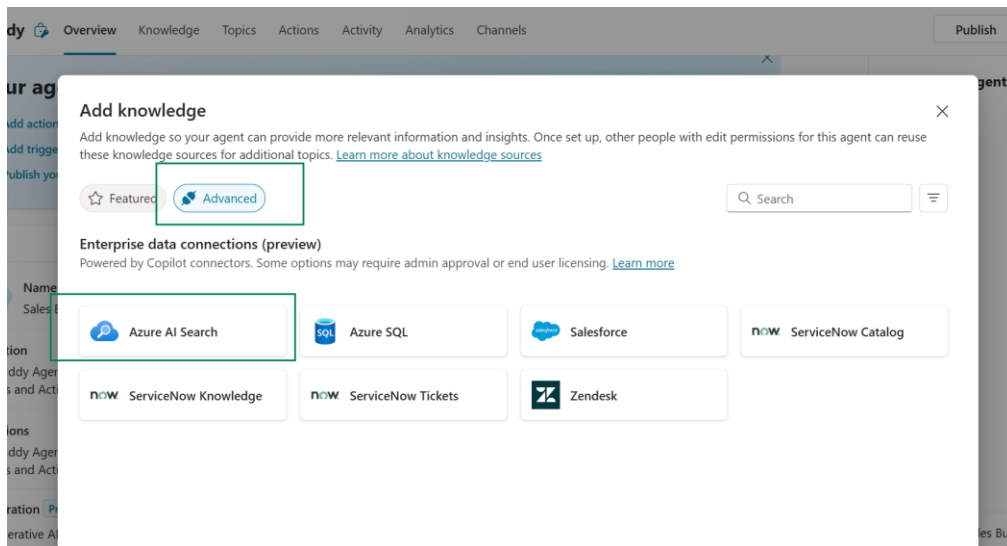

Now, your search service is ready to be used.

30. Open **Copilot Studio** and select the **Sales Buddy** agent. Under **Knowledge**, click on **Add Knowledge.**

31. In the Knowledge window, select **Advanced** tab and select **Azure AI Search**

32. Update **Knowledge name** = *Sales Buddy - Azure AI Search Contracts;*
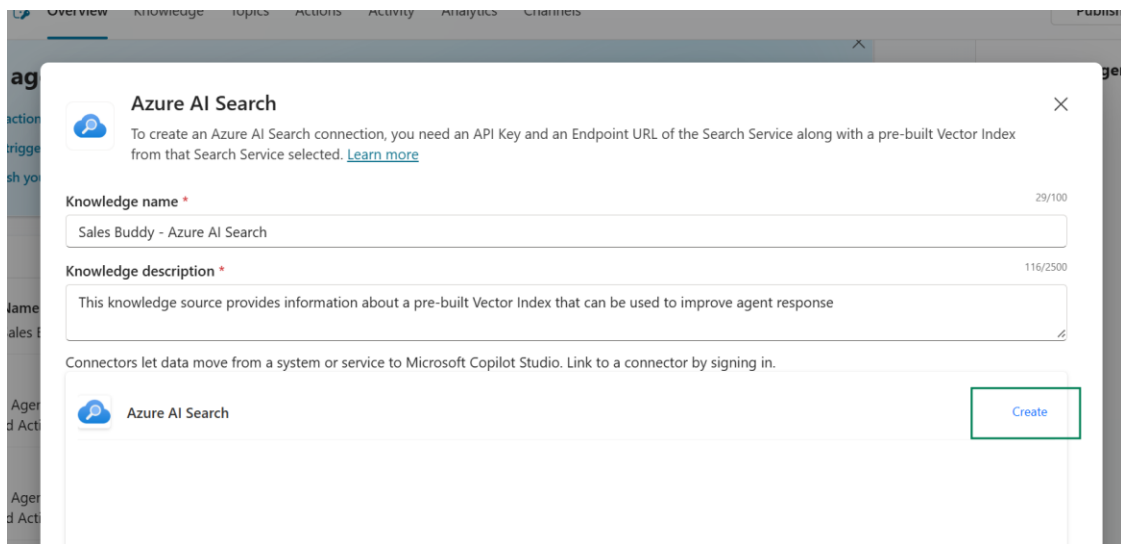**Knowledge Description =**

"*This knowledge source provides access to customer contract agreements. Each contract file follows the format:*

Customer Contract Agreement - [Customer Name].docx

Each contract contains essential details, including:

- *Scope of Services: Defines the services provided*
- *Payment Terms: Specifies the total contract value, milestone-based payments, and due dates.*
- *Service Level Agreement (SLA): Outlines uptime guarantees, response times, and escalation procedures.*
- *Data Privacy & Compliance: Ensures compliance with regulations like GDPR, CCPA, and defines data ownership.*
- *Contract Term & Renewal: States the contract duration, renewal policies, and termination conditions.*
- *Governing Law: Specifies applicable legal jurisdiction "*

Next, click on **Create** to create a connection to **Azure search service**

33. Paste the endpoint URL and Key from Notepad and select **Create.** Once the connection is successful, select **Next.**

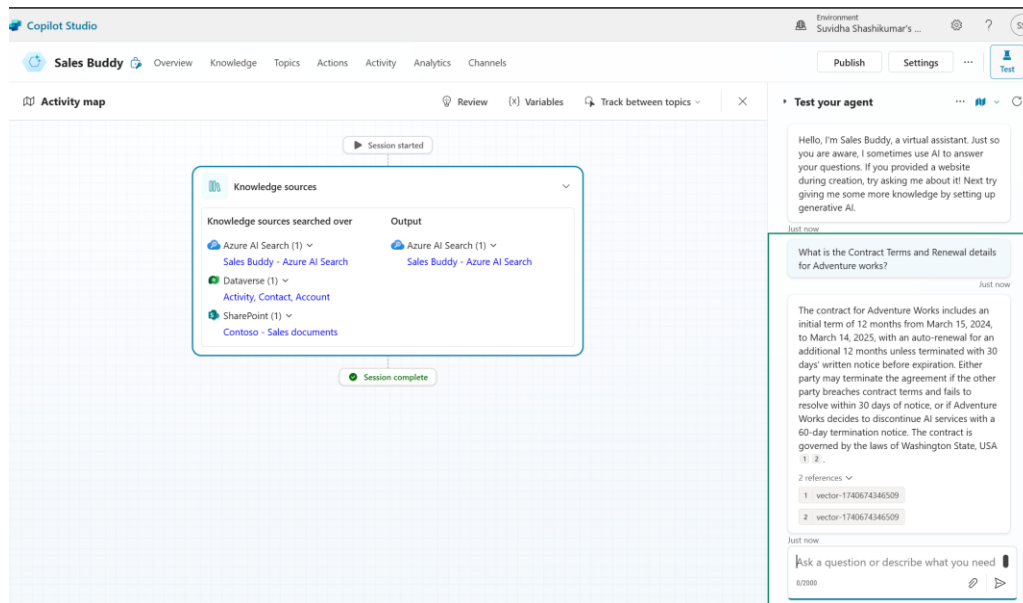34. Select the available vector index and select **Add**

35. You should see a new Azure AI Search knowledge in your Agent.
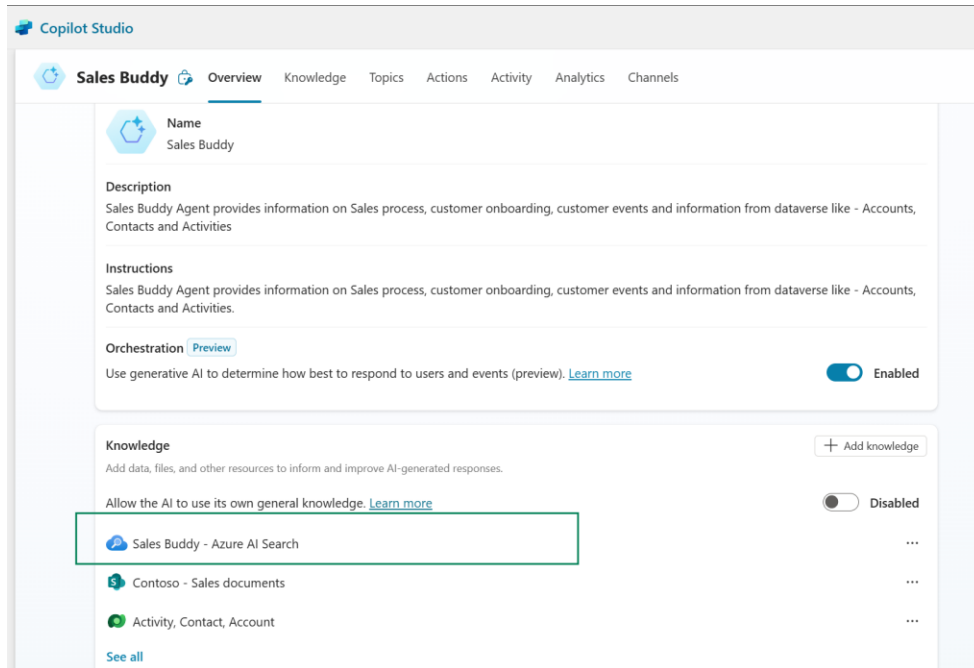
36. Now, we're ready for the test. Open **Test window** and ask question – "*What are the Contract Terms and Renewal details for Adventure works?*"

Congratulations. You have completed Lab 6.