

RAG Innovator Lab: User's Guide

Powerful AI Apps

on Open and Lake-centric Data Foundation

Contents

Overview	3
Agenda	3
Pre-configured User Settings	3
Prerequisites: Prepare AI Foundry Access and Setup.....	4
Provision AI Project.....	4
Role Assignment	6
Get Started with Fabric.....	12
Import Fabric Notebook.....	12
Create Lakehouse.....	14
Add Lakehouse to Notebook.....	17
Prepare OneLake Data for Connection	21
Explore Data.....	21
Prepare Search	22
Check AI Search Resource	25
Create AI Applications on OneLake Data	27
Get Started with PromptFlow.....	27
Create RAG Flow	31
Call Deployed Endpoint	42

Overview

Agenda

No	Task	Duration	Description
1	Microsoft Fabric Overview	60 minutes	
2	Microsoft Fabric Hands-On Lab	60 minutes	
3	Azure AI Foundry Overview	60 minutes	
4	AI Application Development Hands-On Lab	30 minutes	(Optional) Evaluation, +30 minutes
5	Wrap-up and Q&A	30 minutes	

Pre-configured User Settings

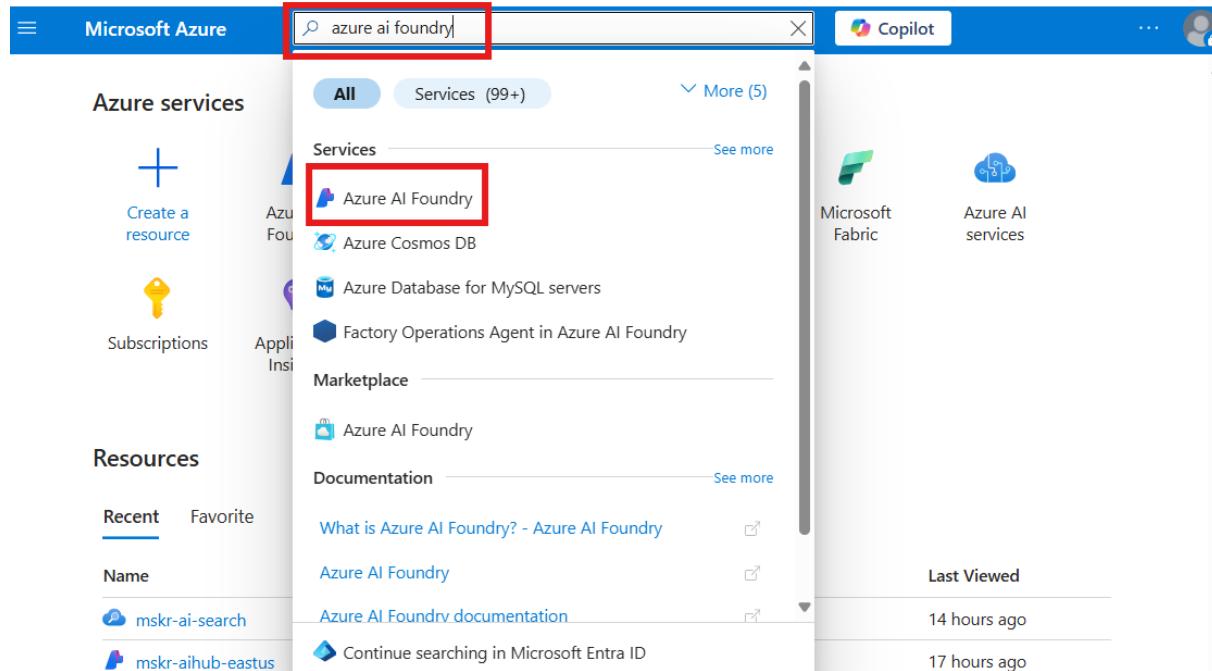
Assignee	Fabric	Azure OpenAI	AI Search	AI Foundry	AI Foundry	Custom Suffix
	Workspace			AI Hub	Compute	
참석자1	_WS_Lab_1	lab-aoai-eus-1	lab-search-1	lab-hub-1	lab-compute-1	user1
참석자2	_WS_Lab_1	lab-aoai-eus-1	lab-search-1	lab-hub-1	lab-compute-1	user2
참석자3	_WS_Lab_2	lab-aoai-eus-2	lab-search-2	lab-hub-2	lab-compute-2	user3
참석자4	...					

* Fabric, Azure 리소스는 참석자 간 공유 가능하지만, Custom Suffix는 참석자 간 unique해야 합니다.

Prerequisites: Prepare AI Foundry Access and Setup

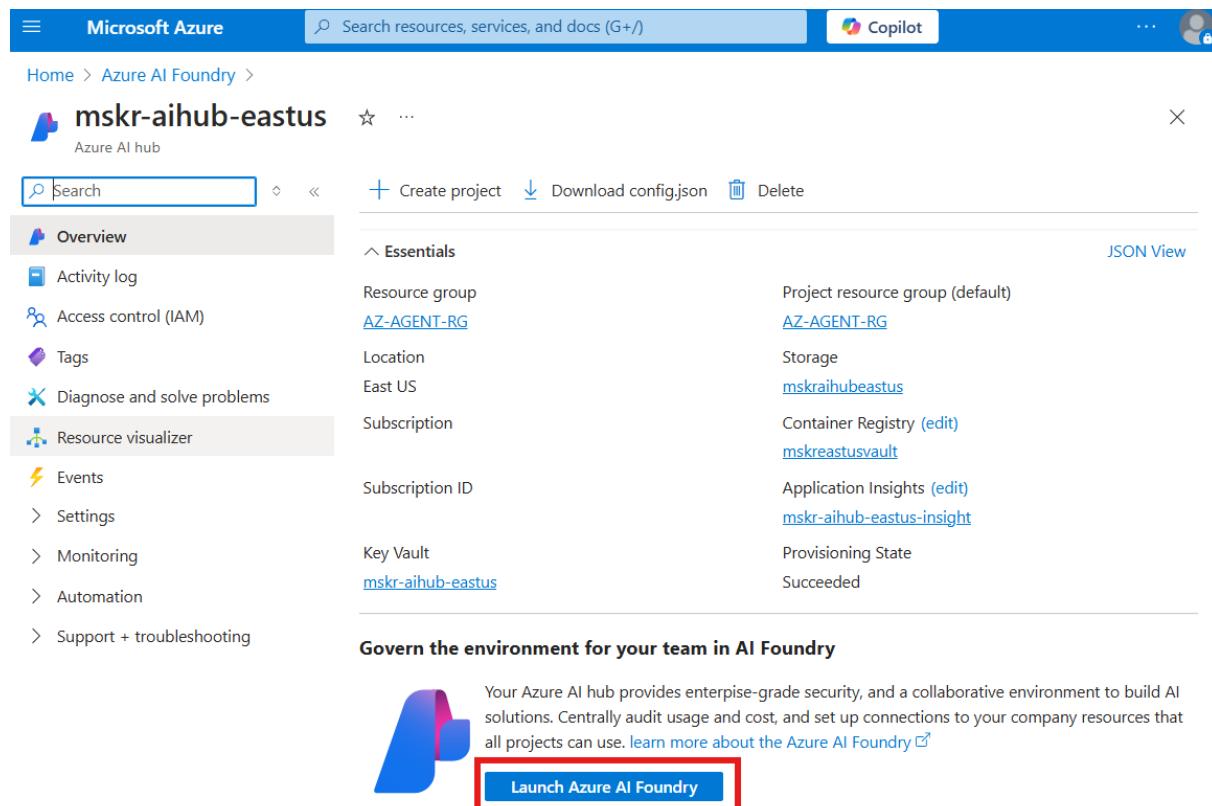
Provision AI Project

1. Azure Portal에 접속해 배정받은 AI Hub 리소스를 선택합니다.



Microsoft Azure search results for 'azure ai foundry':

- Services (99+):
 - Azure AI Foundry (highlighted)
 - Azure Cosmos DB
 - Azure Database for MySQL servers
 - Factory Operations Agent in Azure AI Foundry
- Marketplace:
 - Azure AI Foundry
- Documentation:
 - What is Azure AI Foundry? - Azure AI Foundry
 - Azure AI Foundry
 - Azure AI Foundry documentation
 - Continue searching in Microsoft Entra ID
- Last Viewed:
 - 14 hours ago
 - 17 hours ago



mskr-aihub-eastus - Azure AI hub

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Events

Settings

Monitoring

Automation

Support + troubleshooting

Search

Create project

Download config.json

Delete

Essentials

Resource group	Project resource group (default)
AZ-AGENT-RG	AZ-AGENT-RG

Location	Storage
East US	mskraihubeastus

Subscription	Container Registry (edit)
	mskreatusvault

Subscription ID	Application Insights (edit)
	mskr-aihub-eastus-insight

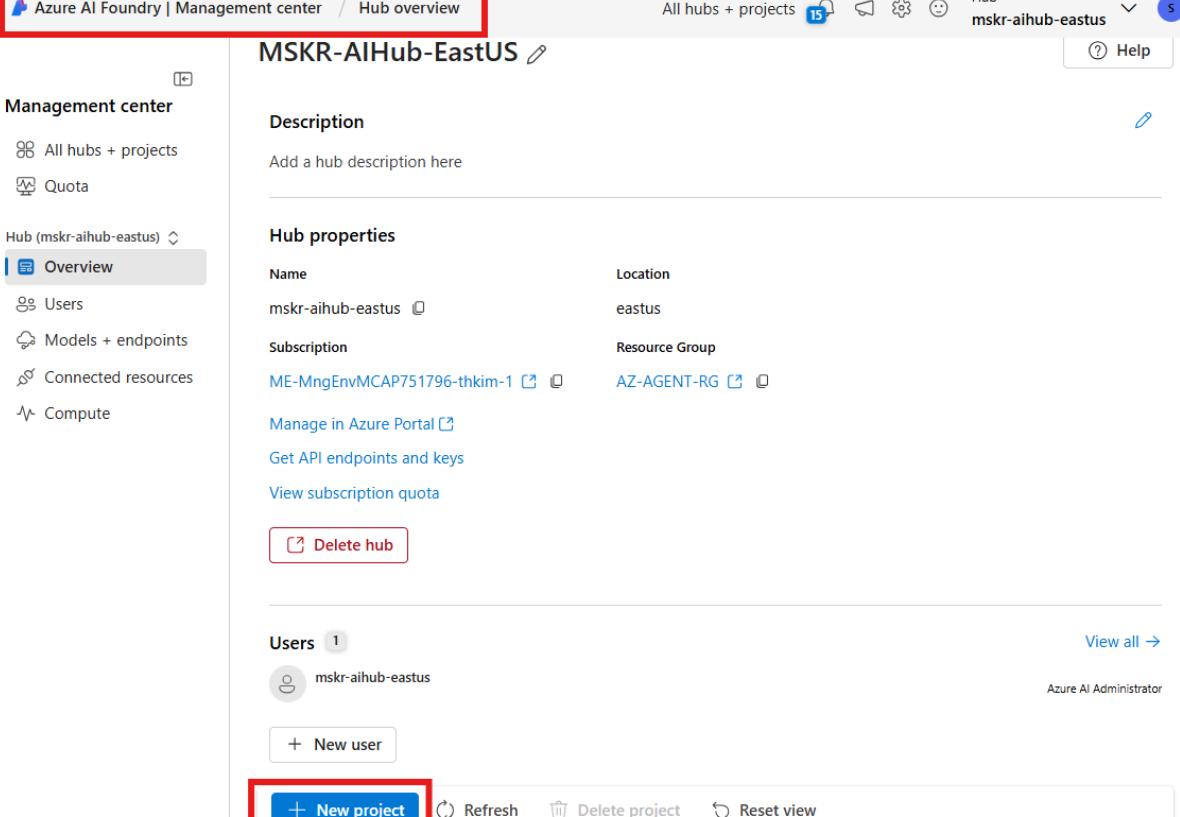
Key Vault	Provisioning State
mskr-aihub-eastus	Succeeded

Govern the environment for your team in AI Foundry

Your Azure AI hub provides enterprise-grade security, and a collaborative environment to build AI solutions. Centrally audit usage and cost, and set up connections to your company resources that all projects can use. [learn more about the Azure AI Foundry](#)

Launch Azure AI Foundry

2. Custom Suffix의 이름으로 새로운 Project를 만듭니다.



Azure AI Foundry | Management center / Hub overview

All hubs + projects 15 Hub mskr-aihub-eastus Help

MSKR-AIHub-EastUS

Description

Add a hub description here

Hub properties

Name	Location
mskr-aihub-eastus	eastus
Subscription	Resource Group
ME-MngEnvMCAP751796-thkim-1	AZ-AGENT-RG

Manage in Azure Portal

Get API endpoints and keys

View subscription quota

Delete hub

Users 1

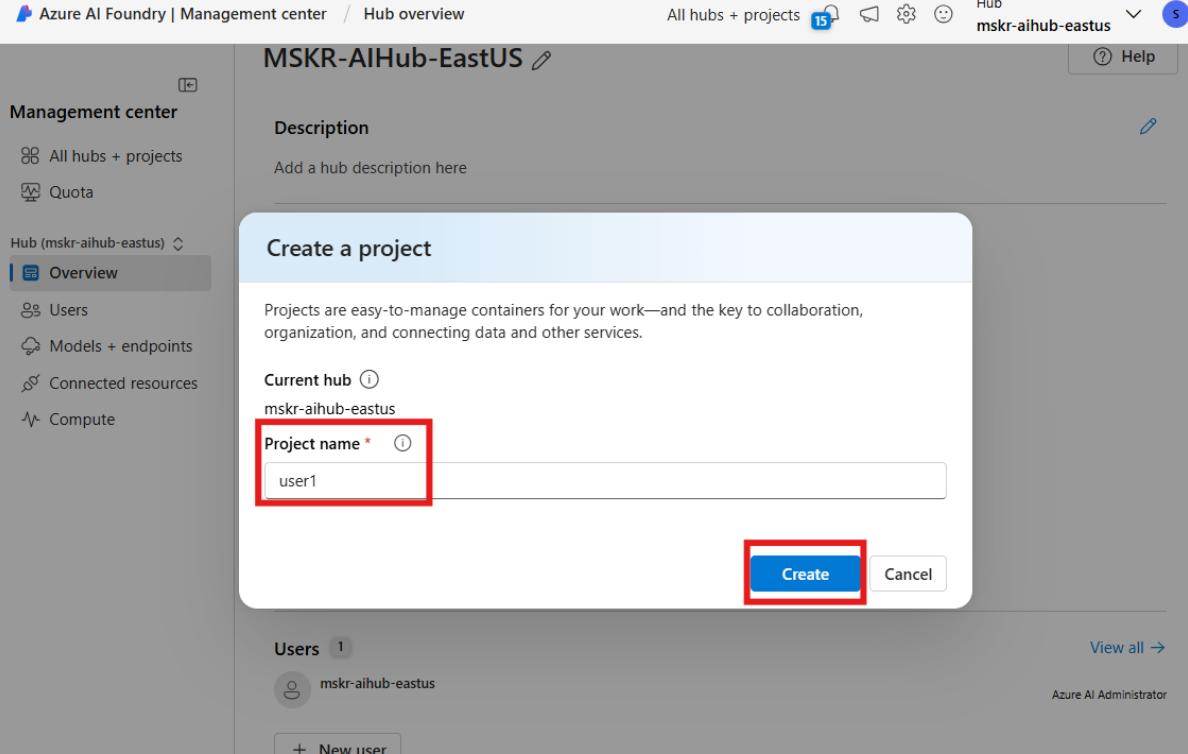
mskr-aihub-eastus

Azure AI Administrator

+ New user

+ New project

Refresh Delete project Reset view



Azure AI Foundry | Management center / Hub overview

All hubs + projects 15 Hub mskr-aihub-eastus Help

MSKR-AIHub-EastUS

Description

Add a hub description here

Create a project

Projects are easy-to-manage containers for your work—and the key to collaboration, organization, and connecting data and other services.

Current hub mskr-aihub-eastus

Project name *

user1

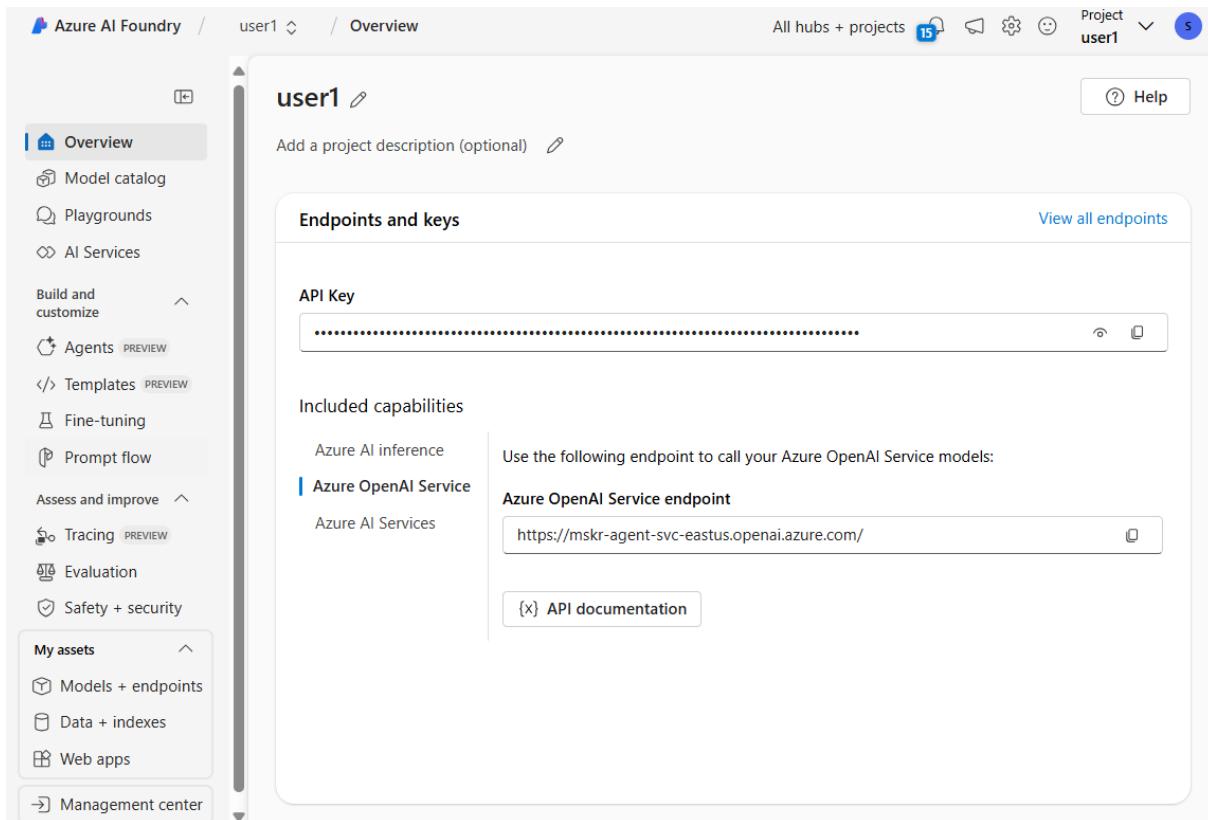
Create Cancel

Users 1

mskr-aihub-eastus

Azure AI Administrator

+ New user



user1

Add a project description (optional)

Endpoints and keys

View all endpoints

API Key

.....

Included capabilities

- Azure AI inference
- Azure OpenAI Service**
- Azure AI Services

Use the following endpoint to call your Azure OpenAI Service models:

Azure OpenAI Service endpoint

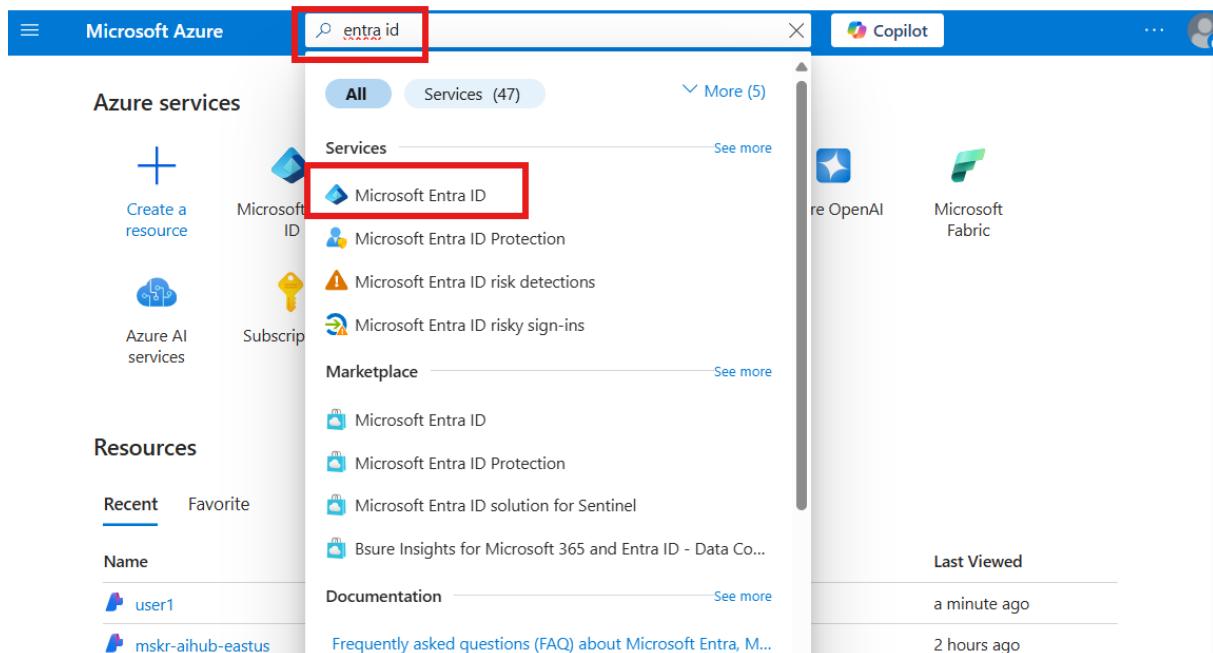
https://mskr-agent-svc-eastus.openai.azure.com/

[x] API documentation

Assign Roles

1. 생성한 AI Project의 Application ID를 확인하고 CLI를 업데이트합니다.

- A. Azure Portal에서 Entra ID 서비스로 이동합니다.



Microsoft Azure

entra id

Azure services

All Services (47)

Services

Microsoft Entra ID

Microsoft Entra ID Protection

Microsoft Entra ID risk detections

Microsoft Entra ID risky sign-ins

Marketplace

Microsoft Entra ID

Microsoft Entra ID Protection

Microsoft Entra ID solution for Sentinel

Microsoft Entra Insights for Microsoft 365 and Entra ID - Data Co...

Documentation

Frequently asked questions (FAQ) about Microsoft Entra, M...

Resources

Recent

Favorite

Name

user1

mskr-aihub-eastus

Microsoft Entra ID

Microsoft Entra ID Protection

Microsoft Entra ID solution for Sentinel

Microsoft Entra Insights for Microsoft 365 and Entra ID - Data Co...

Documentation

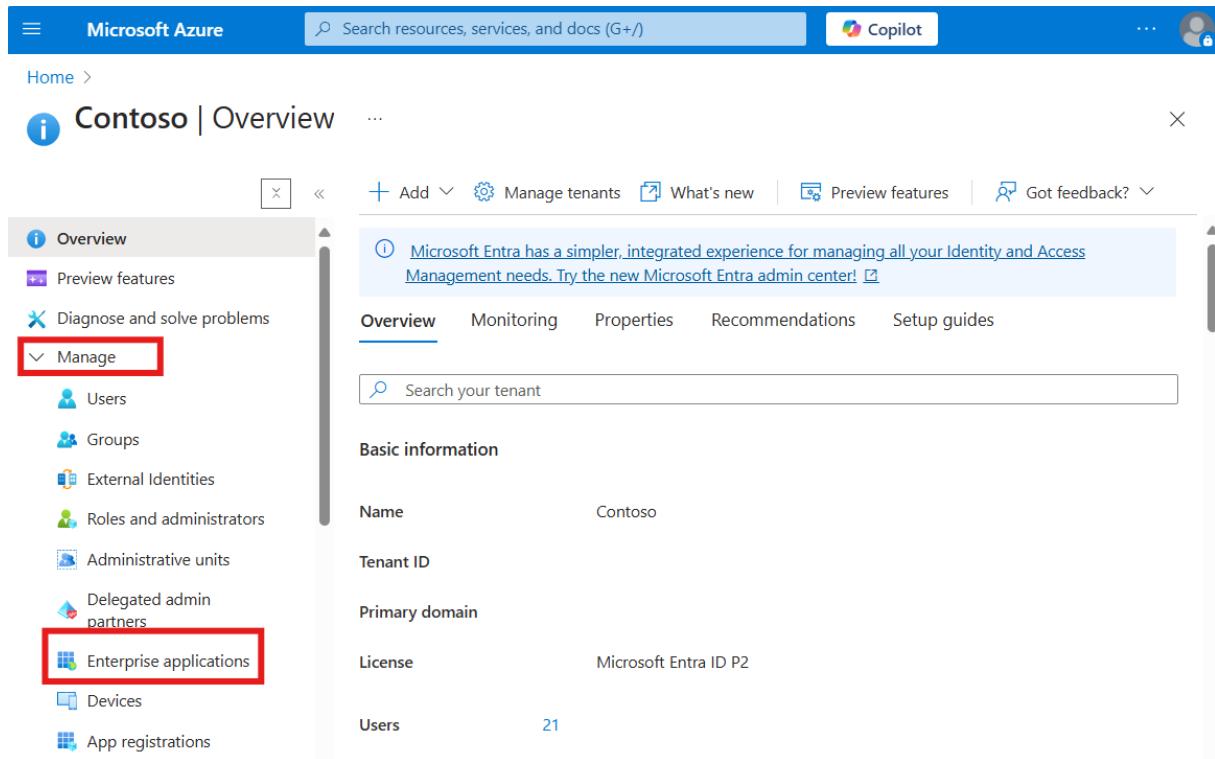
Frequently asked questions (FAQ) about Microsoft Entra, M...

Last Viewed

a minute ago

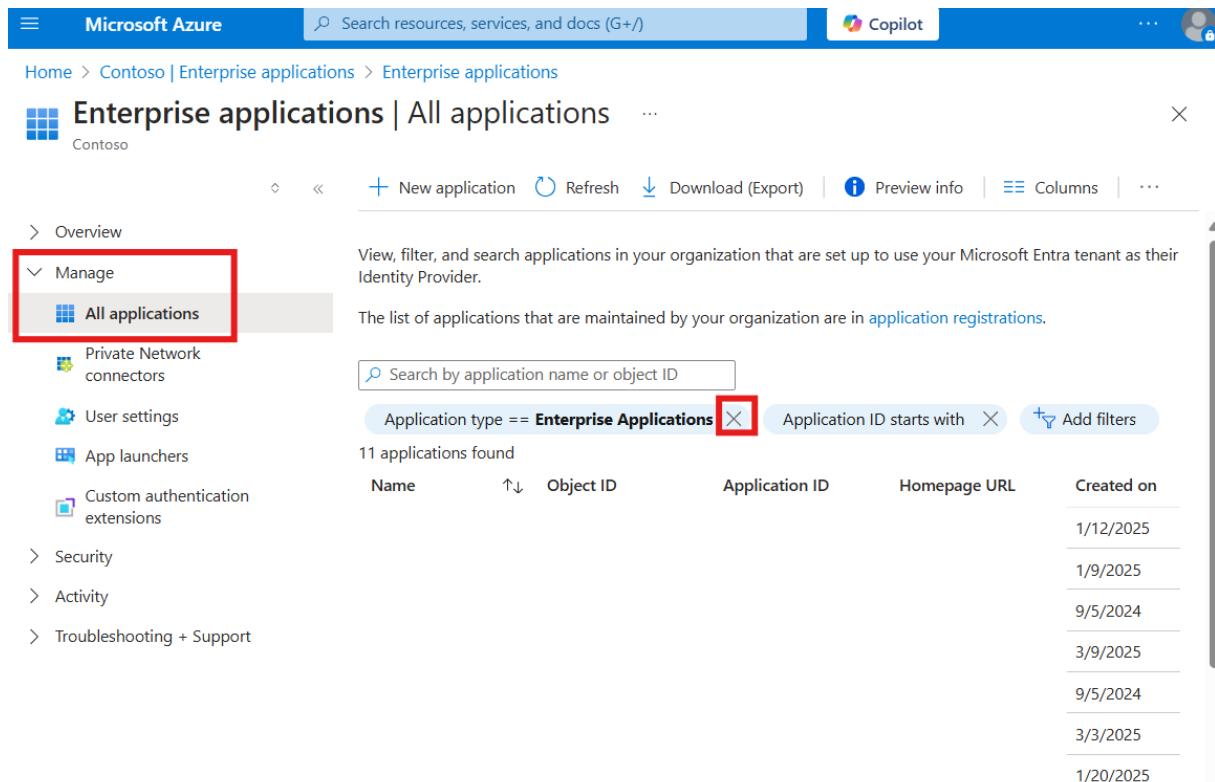
2 hours ago

B. Enterprise Applications를 선택합니다.



The screenshot shows the Microsoft Azure Contoso Overview page. On the left, a navigation menu is open, showing options like Overview, Preview features, Diagnose and solve problems, Manage, and Enterprise applications. The 'Manage' and 'Enterprise applications' items are highlighted with red boxes. The main content area displays basic information for the tenant, including Name (Contoso), Tenant ID, Primary domain, License (Microsoft Entra ID P2), and Users (21). A search bar at the top is labeled 'Search your tenant'.

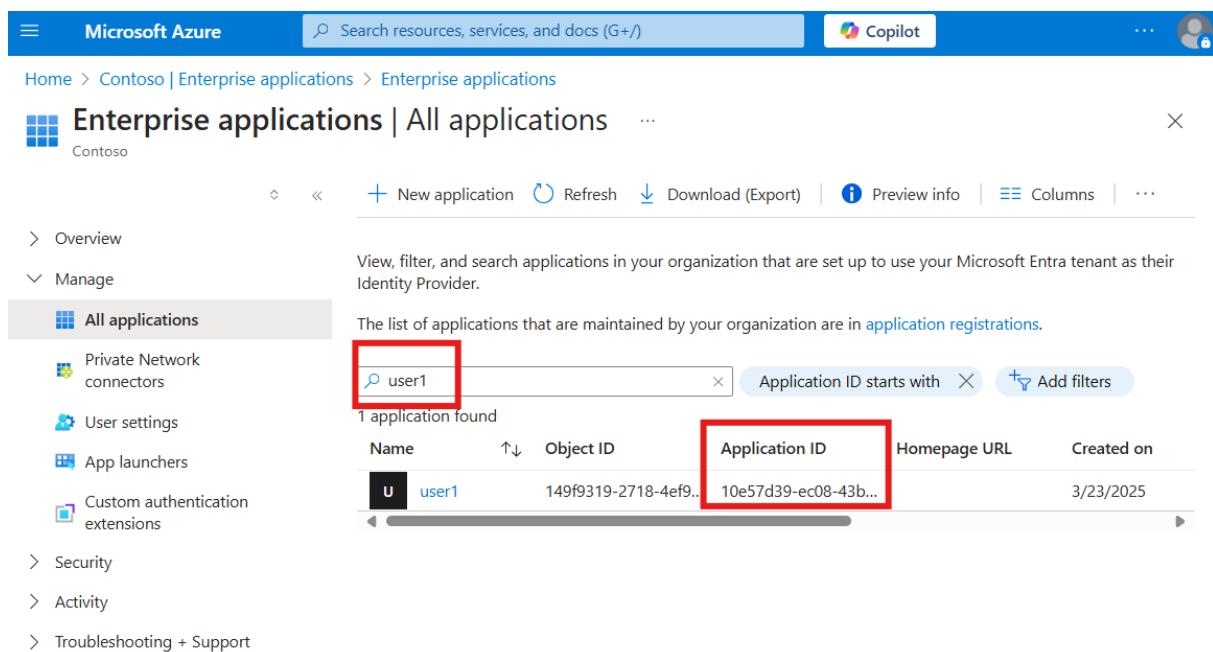
C. All Applications를 선택하고, Application Type 필터를 해제합니다.



The screenshot shows the Microsoft Azure Enterprise applications | All applications page for the Contoso tenant. The left navigation menu is open, showing Overview, Manage, and All applications. The 'All applications' item is highlighted with a red box. The main content area displays a list of applications with a search bar and filters. The 'Application type' filter is set to 'Enterprise Applications' and has a red box around it. Below the table, it says '11 applications found'. The table columns are Name, Object ID, Application ID, Homepage URL, and Created on. The data is as follows:

Name	Object ID	Application ID	Homepage URL	Created on
				1/12/2025
				1/9/2025
				9/5/2024
				3/9/2025
				9/5/2024
				3/3/2025
				1/20/2025

D. 생성한 AI Project의 이름을 검색하고, Application ID를 복사합니다.



The screenshot shows the Microsoft Azure Enterprise applications page for the 'Contoso' tenant. The left sidebar has 'All applications' selected. A search bar at the top right contains 'user1'. The main table lists one application: 'user1' with Application ID '10e57d39-ec08-43b...'. The 'Application ID' column is highlighted with a red box.

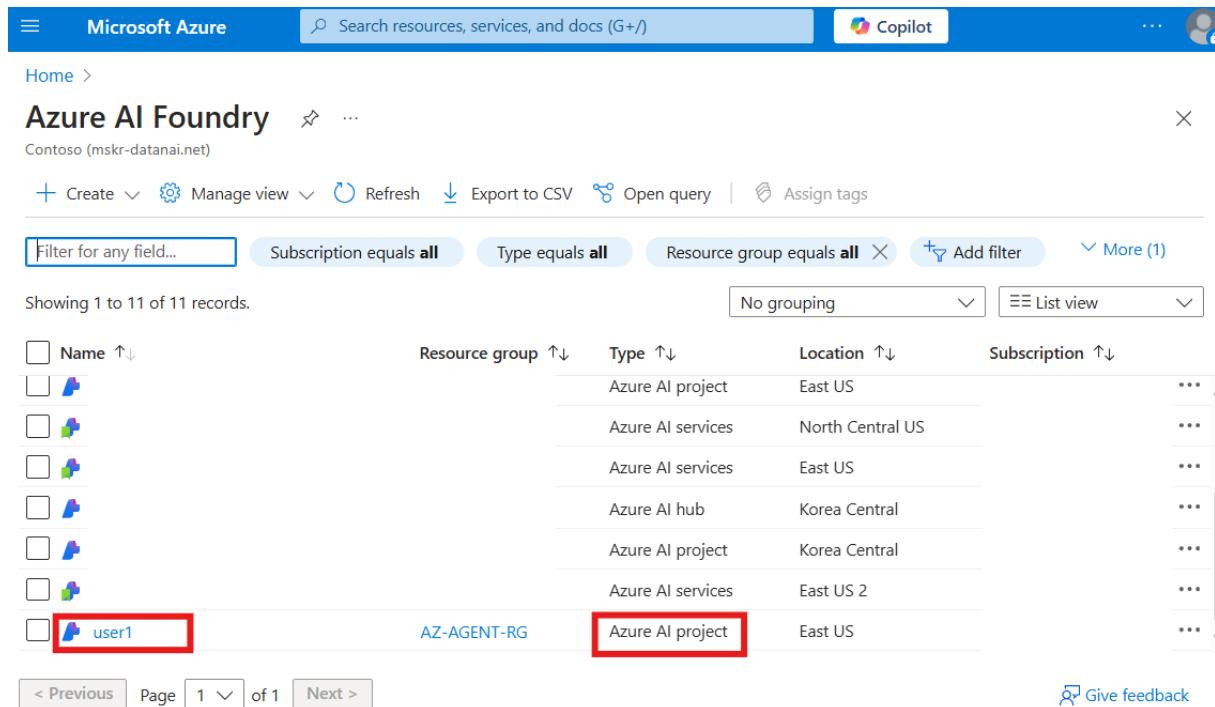
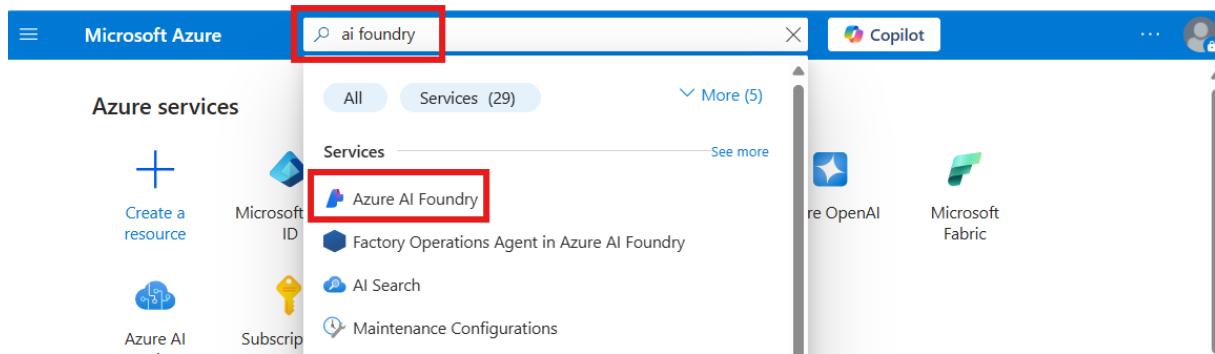
Name	Object ID	Application ID	Homepage URL	Created on
user1	149f9319-2718-4ef9...	10e57d39-ec08-43b...		3/23/2025

E. 클론된 폴더의 AIFoundry\RBAC_CLI 내 user-role-assign.azcli의 ai_project_assignee 정보를 업데이트합니다.

```
AIFoundry > RBAC_CLI > user-role-assign.azcli
1  # Open terminal in Azure portal after login
2  # Assign roles for PromptFlow with following commands
3
4
5
6  $ai_project_assignee = ''
7  $ai_project_workspace_resource = ''
8
9  $storage_account_resource = ''
10 $acr_assignee = ''
11 $compute_instance = ''
12
13
```

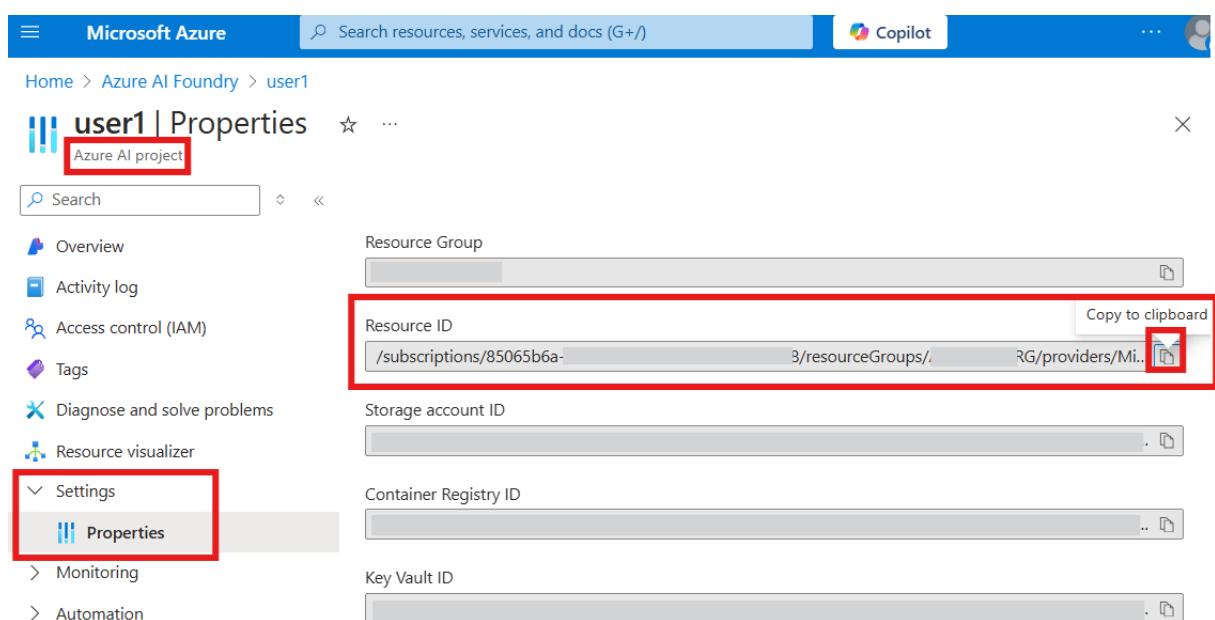
2. 생성한 AI Project의 Workspace ID를 확인하고 CLI를 업데이트합니다.

A. [Azure Portal](#)에서 AI Foundry 서비스로 이동한 후, 생성한 AI Project를 선택합니다.



Name	Resource group	Type	Location	Subscription
user1	AZ-AGENT-RG	Azure AI project	East US	all
		Azure AI services	North Central US	
		Azure AI services	East US	
		Azure AI hub	Korea Central	
		Azure AI project	Korea Central	
		Azure AI services	East US 2	
		Azure AI project	East US	

B. AI Project의 Properties 항목에서 Resource ID를 복사합니다.



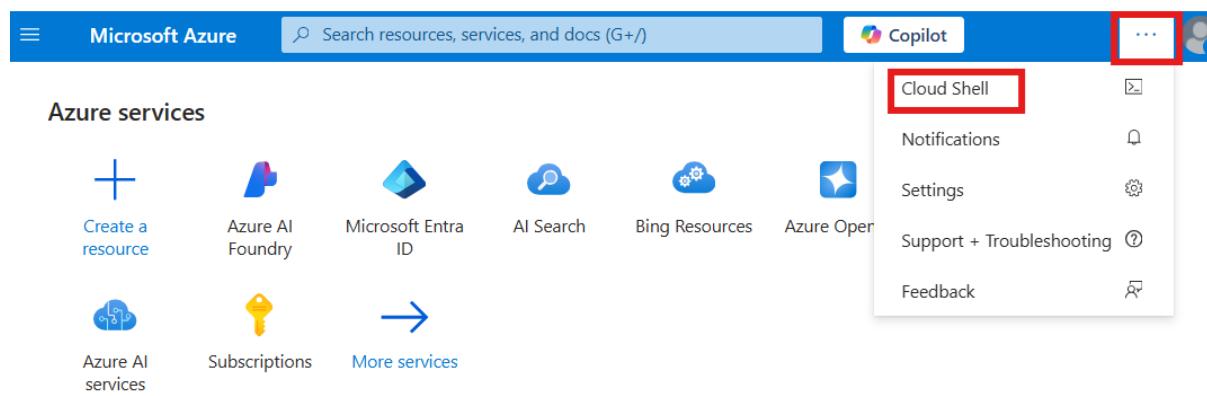
- C. 클론된 폴더의 AIFoundry\RBAC_CLI 내 user-role-assign.azcli의 ai_project_workspace_resource 정보를 업데이트합니다.

```
AIFoundry > RBAC_CLI > ≡ user-role-assign.azcli
1  # Open terminal in Azure portal after login
2  # Assign roles for PromptFlow with following commands
3
4
5
6  $ai_project_assignee = ''
7  $ai_project_workspace_resource = '' [Red box]
8
9  $storage_account_resource = ''
10 $acr_assignee = ''
11 $compute_instance = ''
12
13
```

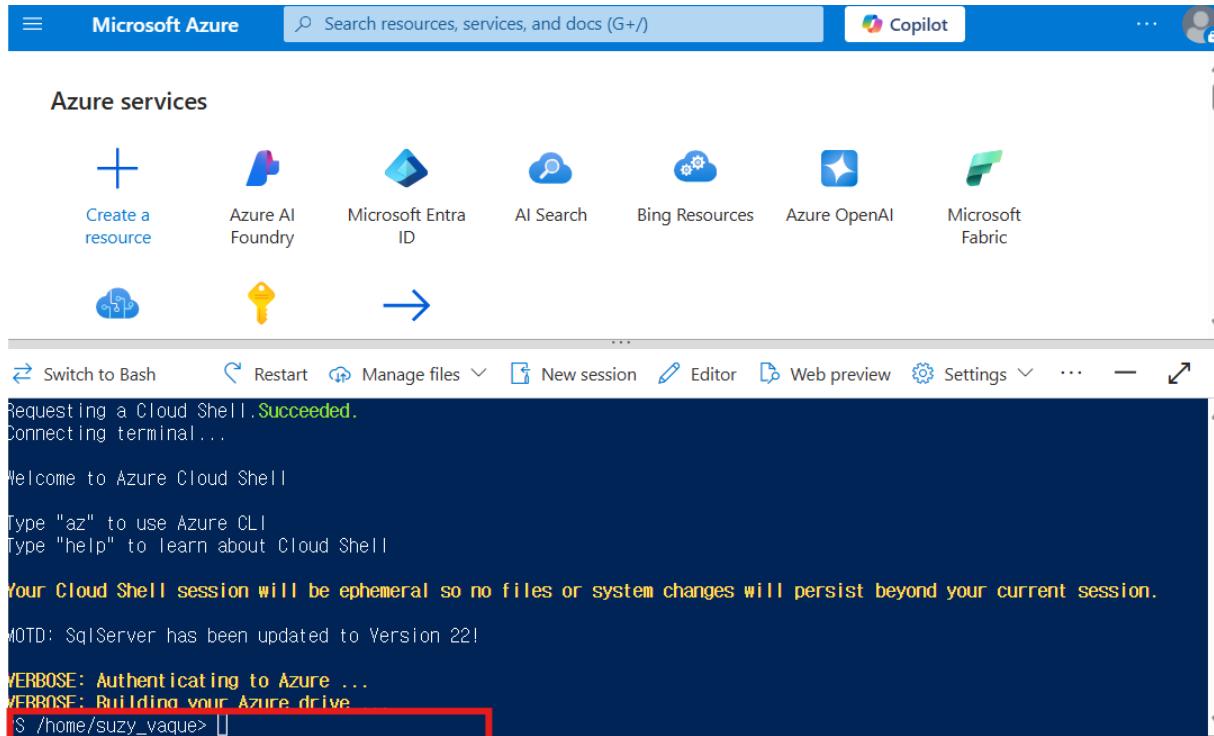
3. 배정받은 리소스의 ID를 확인해 클론된 폴더의 AIFoundry\RBAC_CLI 내 user-role-assign.azcli를 업데이트합니다.

```
AIFoundry > RBAC_CLI > ≡ user-role-assign.azcli
1  # Open terminal in Azure portal after login
2  # Assign roles for PromptFlow with following commands
3
4
5
6  $ai_project_assignee = ''
7  $ai_project_workspace_resource = '' [Red box]
8
9  $storage_account_resource = '' [Red box]
10 $acr_assignee = ''
11 $compute_instance = ''
12
13
```

4. Azure Portal에서 Cloud Shell을 실행합니다.



5. 터미널에 user-role-assign.azcli의 각 커맨드를 실행해 Role Assign을 완료합니다.



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

Copilot

Azure services

Create a resource Azure AI Foundry Microsoft Entra ID AI Search Bing Resources Azure OpenAI Microsoft Fabric

Switch to Bash Restart Manage files New session Editor Web preview Settings

Requesting a Cloud Shell. Succeeded. Connecting terminal...

Welcome to Azure Cloud Shell

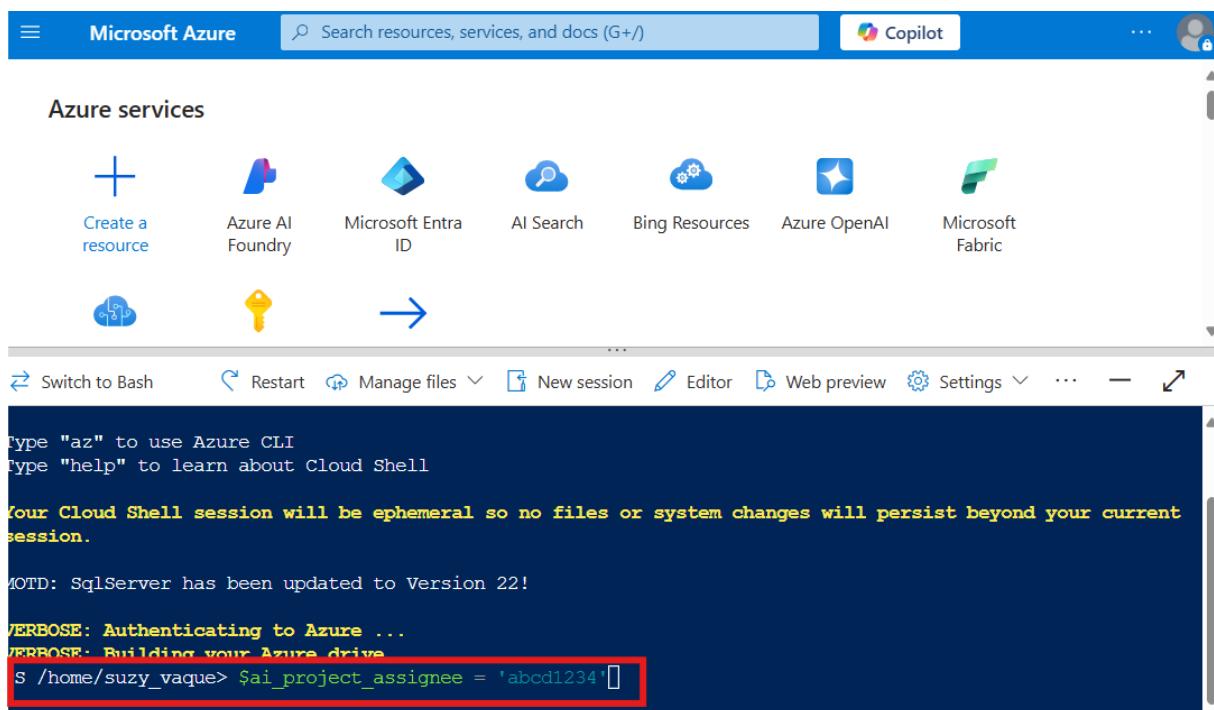
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

MOTD: SqlServer has been updated to Version 22!

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive

```
$ /home/suzy_vaque> 
```



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

Copilot

Azure services

Create a resource Azure AI Foundry Microsoft Entra ID AI Search Bing Resources Azure OpenAI Microsoft Fabric

Switch to Bash Restart Manage files New session Editor Web preview Settings

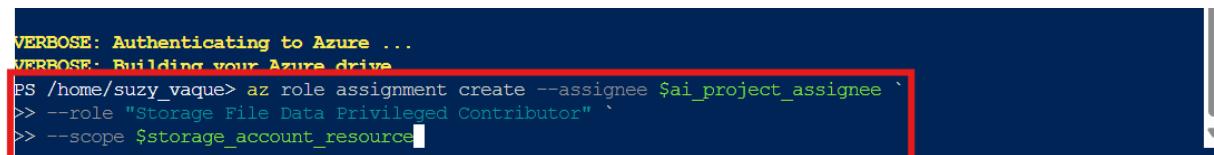
type "az" to use Azure CLI
type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

MOTD: SqlServer has been updated to Version 22!

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive

```
$ /home/suzy_vaque> $ai_project_assignee = 'abcd1234' 
```



VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive

```
$ /home/suzy_vaque> az role assignment create --assignee $ai_project_assignee  
>> --role "Storage File Data Privileged Contributor"  
>> --scope $storage_account_resource 
```

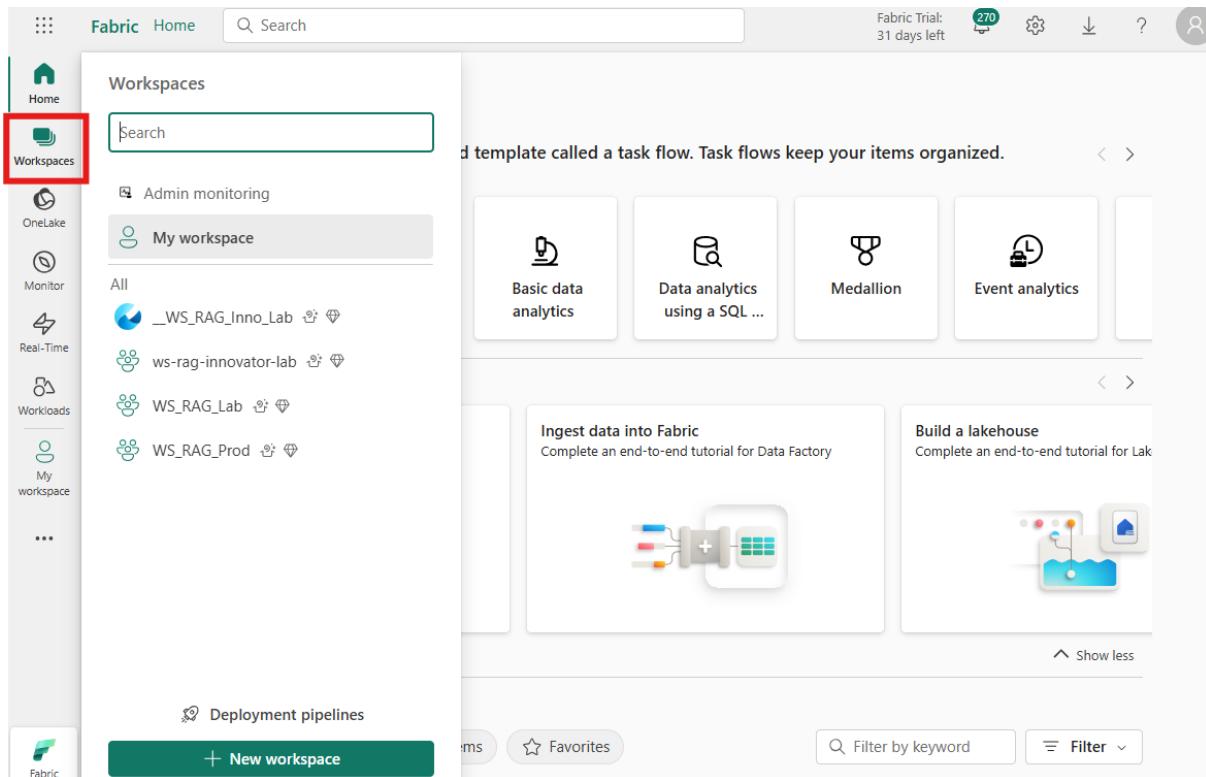
Get Started with Fabric

Import Fabric Notebook

- GitHub Repository를 로컬로 Clone하기 위해, 로컬에서 터미널을 열고 다음 커맨드를 실행합니다.

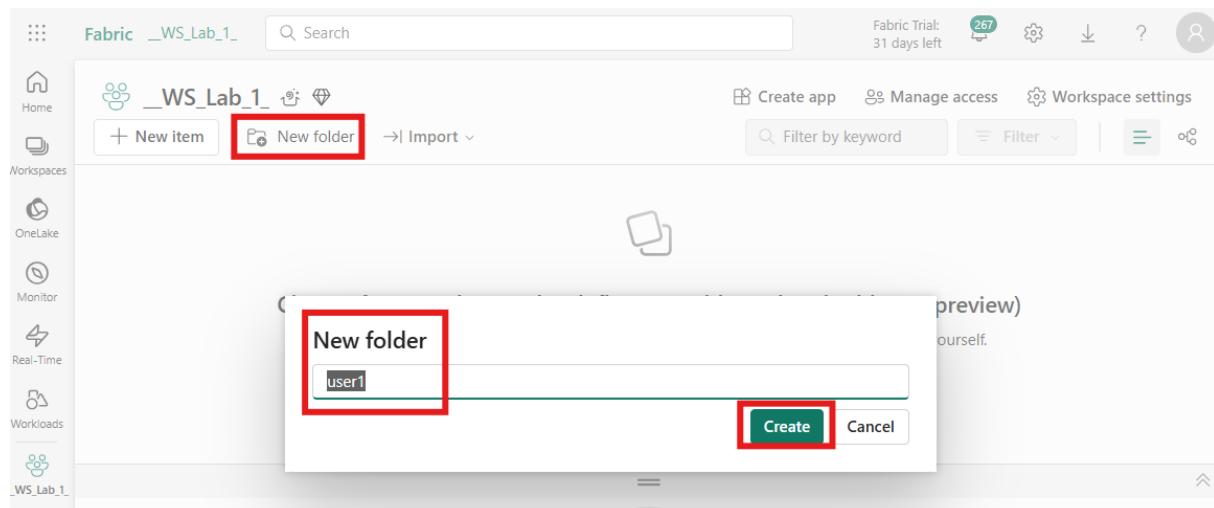
```
git clone https://github.com/KRSTUDataAI/rag-innovator-lab.git
```

6. Fabric Portal에서 배정받은 Workspace를 선택합니다.



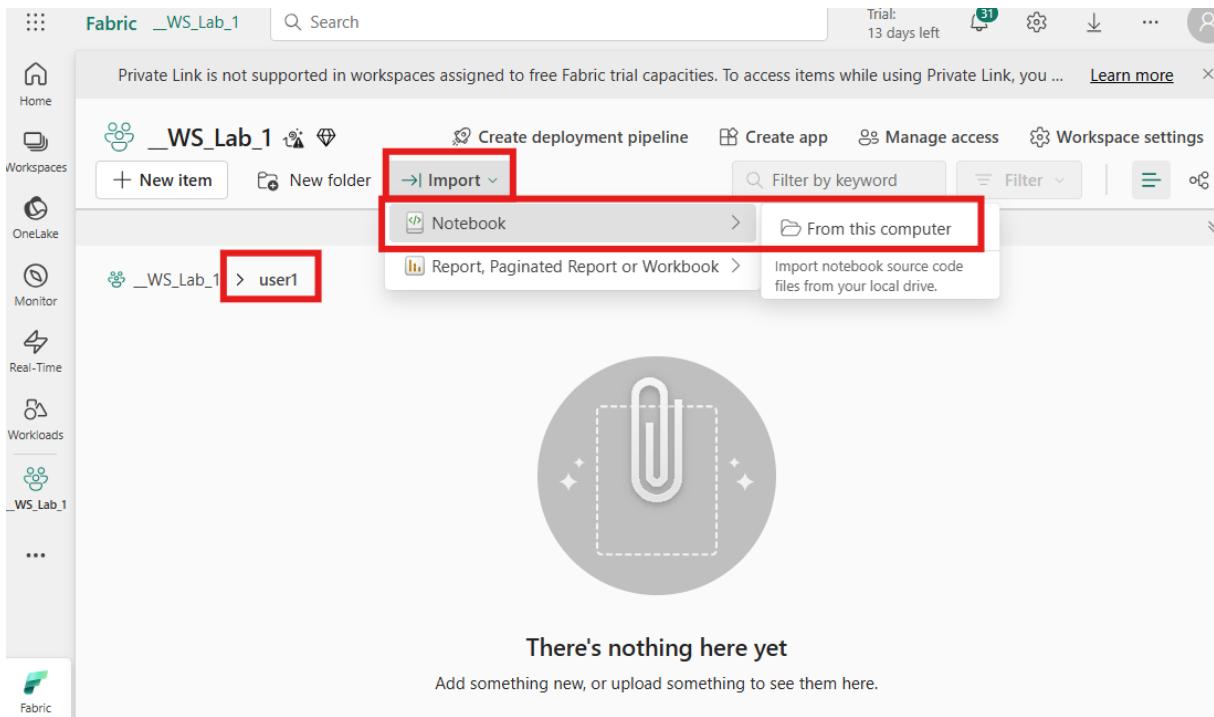
The screenshot shows the Fabric Portal interface. On the left, a sidebar has 'Workspaces' selected and highlighted with a red box. Below it are 'OneLake', 'Monitor', 'Real-Time', 'Workloads', and 'My workspace'. At the bottom of the sidebar is a 'Fabric' icon. The main area is titled 'Workspaces' and shows a search bar. It lists 'My workspace' (selected and highlighted with a red box), 'All' workspaces including '_WS_RAG_Inno_Lab', 'ws-rag-innovator-lab', 'WS_RAG_Lab', and 'WS_RAG_Prod'. Below this is a 'Deployment pipelines' section with a 'New workspace' button. To the right, there are several cards: 'Admin monitoring', 'Basic data analytics', 'Data analytics using a SQL ...', 'Medallion', 'Event analytics', 'Ingest data into Fabric' (with a 'Show less' link), and 'Build a lakehouse' (with a 'Show less' link). At the bottom are 'Filter by keyword' and 'Filter' buttons.

7. 자신의 Custom Suffix 이름으로 폴더를 생성합니다.



The screenshot shows the Fabric Portal interface for a specific workspace named '_WS_Lab_1'. The sidebar shows 'Workspaces' and '_WS_Lab_1'. The main area has a search bar and buttons for 'Create app', 'Manage access', and 'Workspace settings'. Below is a 'New item' button and a 'New folder' button, both of which are highlighted with red boxes. A modal dialog is open, titled 'New folder', with a text input field containing 'user1'. At the bottom of the dialog are 'Create' and 'Cancel' buttons, both of which are highlighted with red boxes. The background shows a preview of the workspace content.

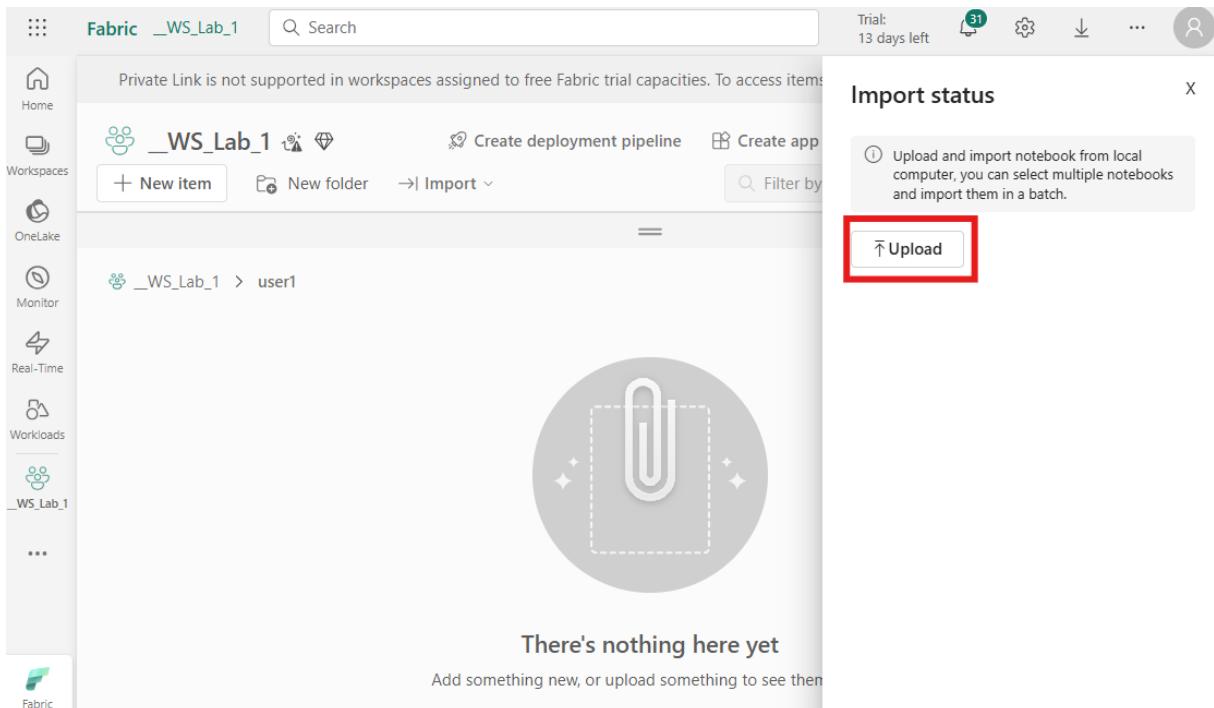
8. 생성한 폴더에 접속한 상태에서 Import Notebook을 진행합니다.



The screenshot shows the Azure Fabric workspace interface. The left sidebar lists 'Workspaces' with '_WS_Lab_1' selected. The main area shows a folder structure: '_WS_Lab_1 > user1'. A red box highlights the 'Import' button in the top navigation bar. A dropdown menu is open, with 'Notebook' selected and another red box highlighting it. The text 'Import notebook source code files from your local drive.' is visible. The center of the screen features a large circular icon with a paperclip and the text 'There's nothing here yet'.

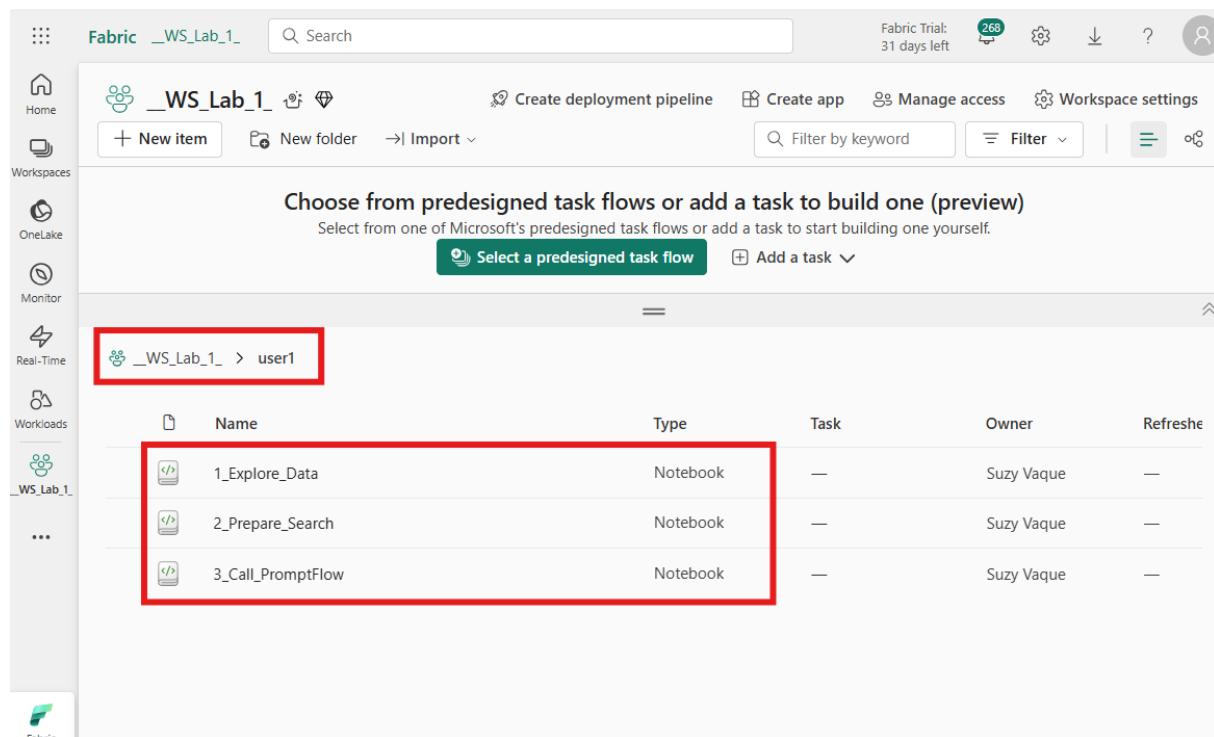
9. 클론된 폴더의 Fabric\Notebooks 내 1_Explore_Data.ipynb, 2_Prepares_Search.ipynb,

3_Call_PromptFlow.ipynb 세 개 노트북을 모두 업로드합니다.



The screenshot shows the Azure Fabric workspace interface. The left sidebar lists 'Workspaces' with '_WS_Lab_1' selected. The main area shows a folder structure: '_WS_Lab_1 > user1'. A red box highlights the 'Import' button in the top navigation bar. A modal dialog titled 'Import status' is open, containing a message: 'Upload and import notebook from local computer, you can select multiple notebooks and import them in a batch.' Below the message is a red box highlighting the 'Upload' button. The center of the screen features a large circular icon with a paperclip and the text 'There's nothing here yet'.

10. Import가 완료되면 Fabric Notebook을 사용할 수 있습니다.

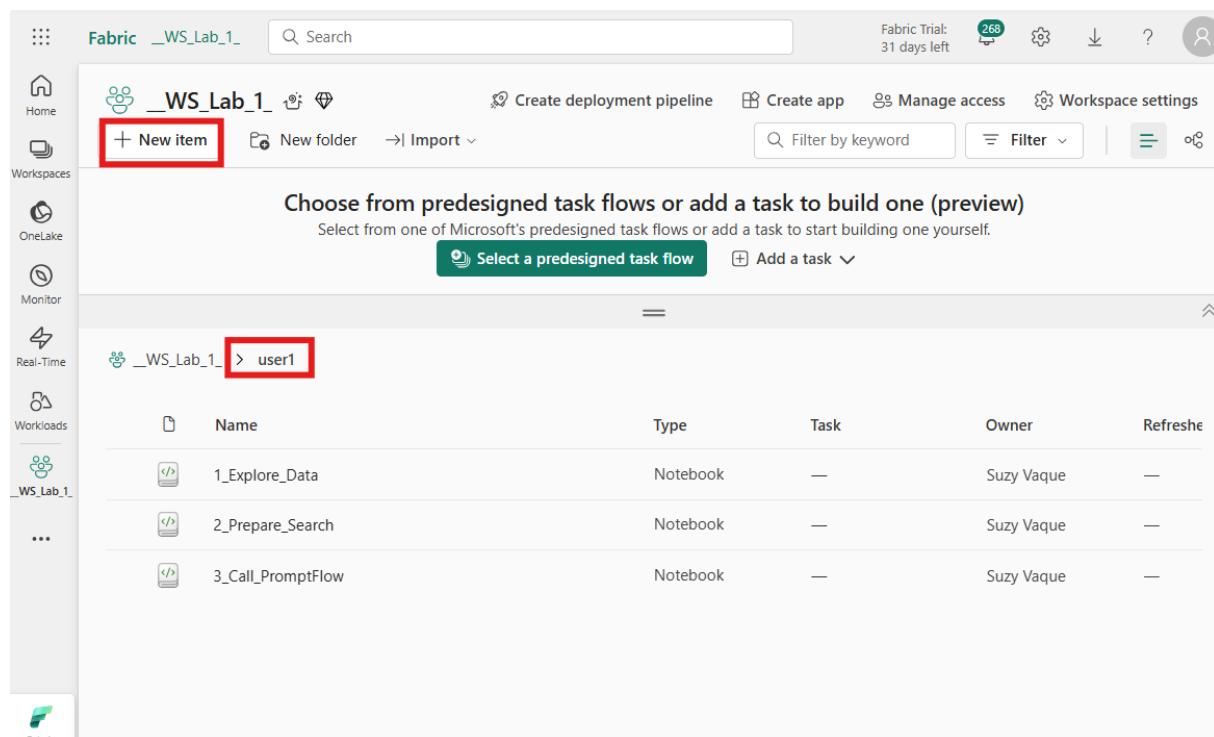


The screenshot shows the Microsoft Fabric workspace interface. The top navigation bar includes 'Fabric _WS_Lab_1', a search bar, and workspace settings. The left sidebar lists 'Workspaces' (OneLake, Monitor, Real-Time, Workloads), with '_WS_Lab_1' selected. The main content area displays a list of items under '_WS_Lab_1 > user1'. A red box highlights the list of three notebooks:

Name	Type	Task	Owner	Refresh
1_Explore_Data	Notebook	—	Suzy Vaque	—
2_Prepares_Search	Notebook	—	Suzy Vaque	—
3_Call_PromptFlow	Notebook	—	Suzy Vaque	—

Create Lakehouse

1. 폴더에 들어온 상태에서, New Item을 선택해 자신의 Custom Suffix 이름으로 Lakehouse를 생성합니다.



The screenshot shows the Microsoft Fabric workspace interface. The top navigation bar includes 'Fabric _WS_Lab_1', a search bar, and workspace settings. The left sidebar lists 'Workspaces' (OneLake, Monitor, Real-Time, Workloads), with '_WS_Lab_1' selected. The main content area displays a list of items under '_WS_Lab_1 > user1'. A red box highlights the '+ New item' button in the top navigation bar.

New item

Store data

Organize, query, and store your ingested data in an easily retrievable format.

Lakehouse (highlighted with a red box)

Provide strategic insights from multiple sources into your business-focused or departmental data.

Eventhouse

Rapidly load structured, unstructured and streaming data for querying.

Sample warehouse

Start a new warehouse with sample data already loaded

Semantic model

Combine data sources in a semantic model to visualize or share it.

Warehouse

Provide strategic insights from multiple sources into your entire business.

Prepare data

Clean, transform, extract, and load your data for analysis and modeling tasks.

Apache Airflow job (preview)

Simplifies the creation and management of Apache Airflow environments on which you can operate end-to-end data pipelines at scale.

Azure Data Factory (preview)

Mount an ADF into Fabric to monitor all your pipelines in one data integration platform.

Copy job

Makes it easy to copy data in Fabric. Includes full copy, incremental copy, and event-based copy modes.

_WS_Lab_1_

Create deployment pipeline **Create app** **Manage access** **Workspace settings**

New item **New folder** **Import**

Choose from predesigned task flows or add a task to build one (preview)

Select from one of Microsoft's predesigned task flows or add a task to start building one yourself.

Select a predesigned task flow **Add a task**

New lakehouse

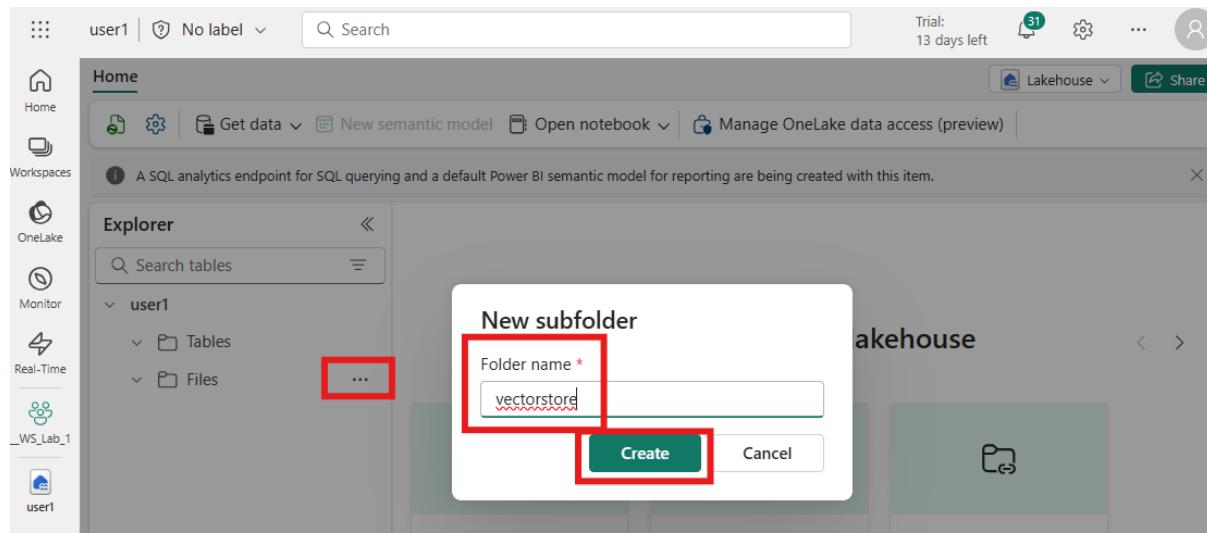
Name (highlighted with a red box)

Lakehouse schemas (Public Preview)

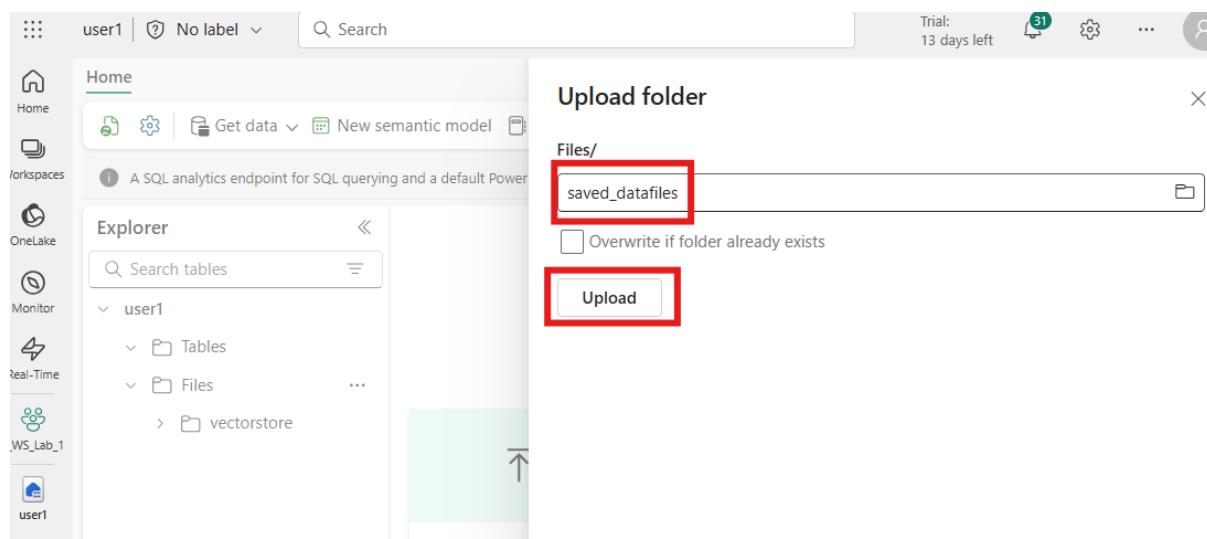
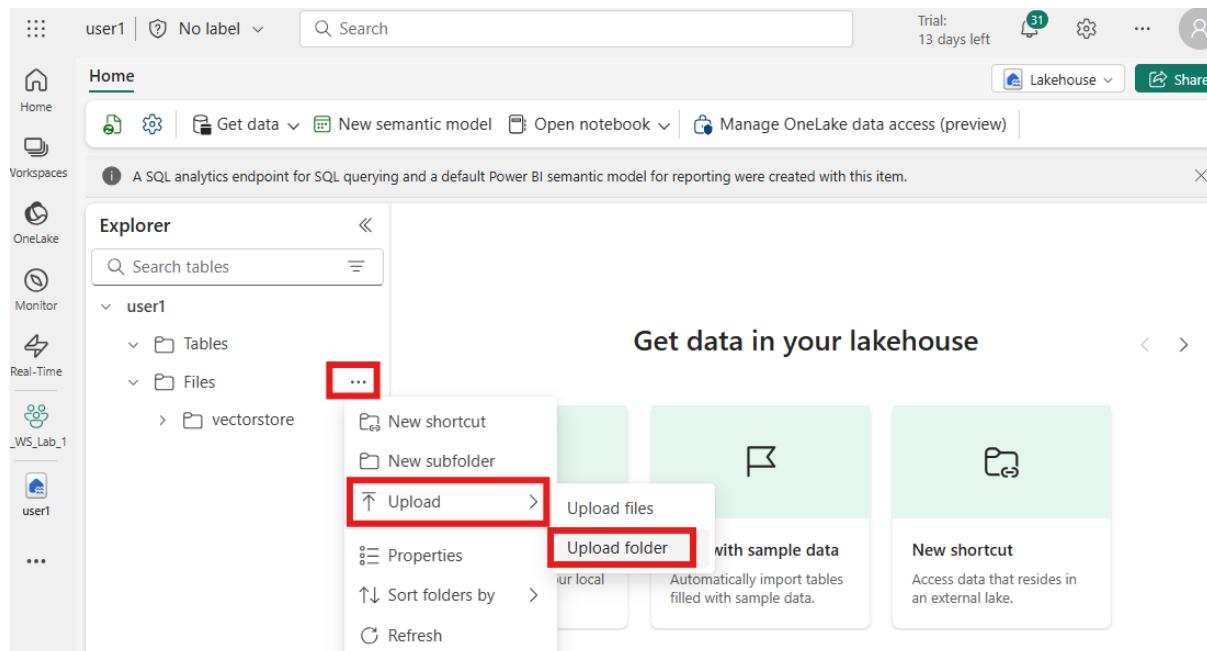
Create **Cancel**

Task	Owner	Refresh
—	Suzy Vaque	—
—	Suzy Vaque	—
—	Suzy Vaque	—

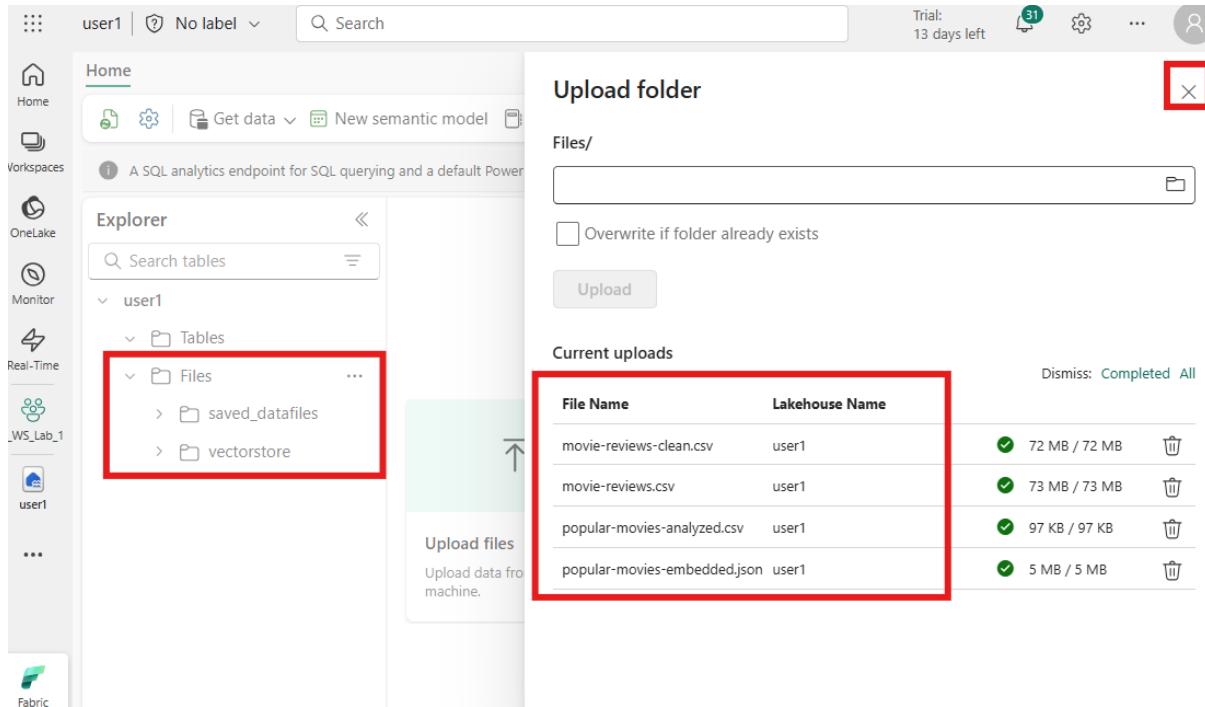
2. Lakehouse 생성이 완료되면 Lakehouse Explorer로 이동합니다.
3. Files에 “vectorstore” 이름의 New Subfolder를 추가합니다.



4. Files에 대해 Upload Folder를 선택해, 앞서 클론해둔 레포지터리의 Fabric\Files\saved_datafiles 폴더를 업로드합니다.



5. 업로드가 완료됐다면 Lakehouse Explorer를 종료합니다.



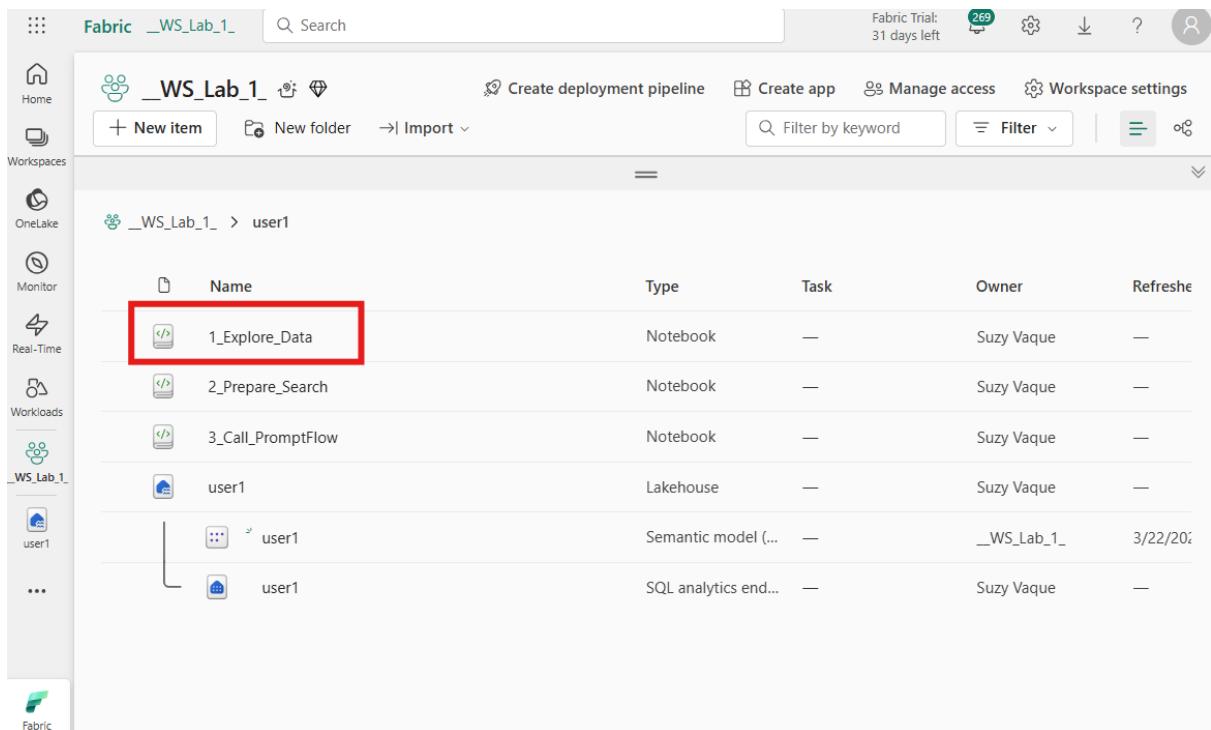
The screenshot shows the Fabric Lakehouse Explorer interface. On the left, the 'Explorer' sidebar shows a tree structure with 'user1' selected. Under 'user1', there are 'Tables' and 'Files' folders. The 'Files' folder is expanded, showing 'saved_datafiles' and 'vectorstore' subfolders, both of which are highlighted with a red box. On the right, a 'Upload folder' dialog is open, with a red box highlighting the 'X' button in the top right corner. Below the dialog, the 'Current uploads' section shows a table of uploaded files:

File Name	Lakehouse Name
movie-reviews-clean.csv	user1
movie-reviews.csv	user1
popular-movies-analyzed.csv	user1
popular-movies-embedded.json	user1

Each row in the table has a green checkmark and a trash bin icon in the last column.

Add Lakehouse to Notebook

1. Workspace 폴더 내의 Fabric Notebook을 선택합니다.

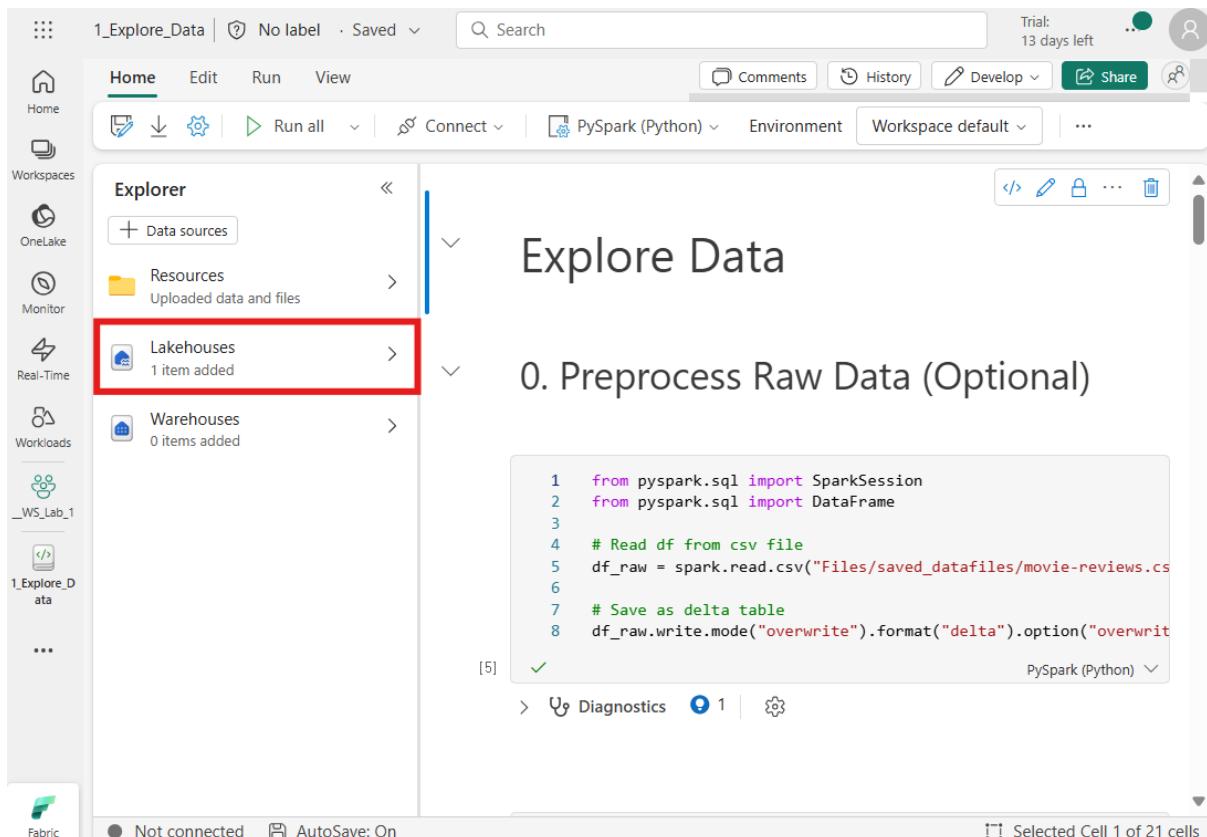


The screenshot shows the Fabric Workspace interface. On the left, the 'Workspaces' sidebar shows a tree structure with 'Fabric' selected. Under 'Fabric', there is a folder named '_WS_Lab_1_'. The '_WS_Lab_1_' folder is expanded, showing a list of items:

Name	Type	Task	Owner	Refresh
1_Explore_Data	Notebook	—	Suzy Vaque	—
2_Prepares_Search	Notebook	—	Suzy Vaque	—
3_Call_PromptFlow	Notebook	—	Suzy Vaque	—
user1	Lakehouse	—	Suzy Vaque	—
user1	Semantic model (...)	—	_WS_Lab_1_	3/22/2024
user1	SQL analytics end...	—	Suzy Vaque	—

The '1_Explore_Data' item is highlighted with a red box. The top navigation bar shows 'Fabric Trial: 31 days left'.

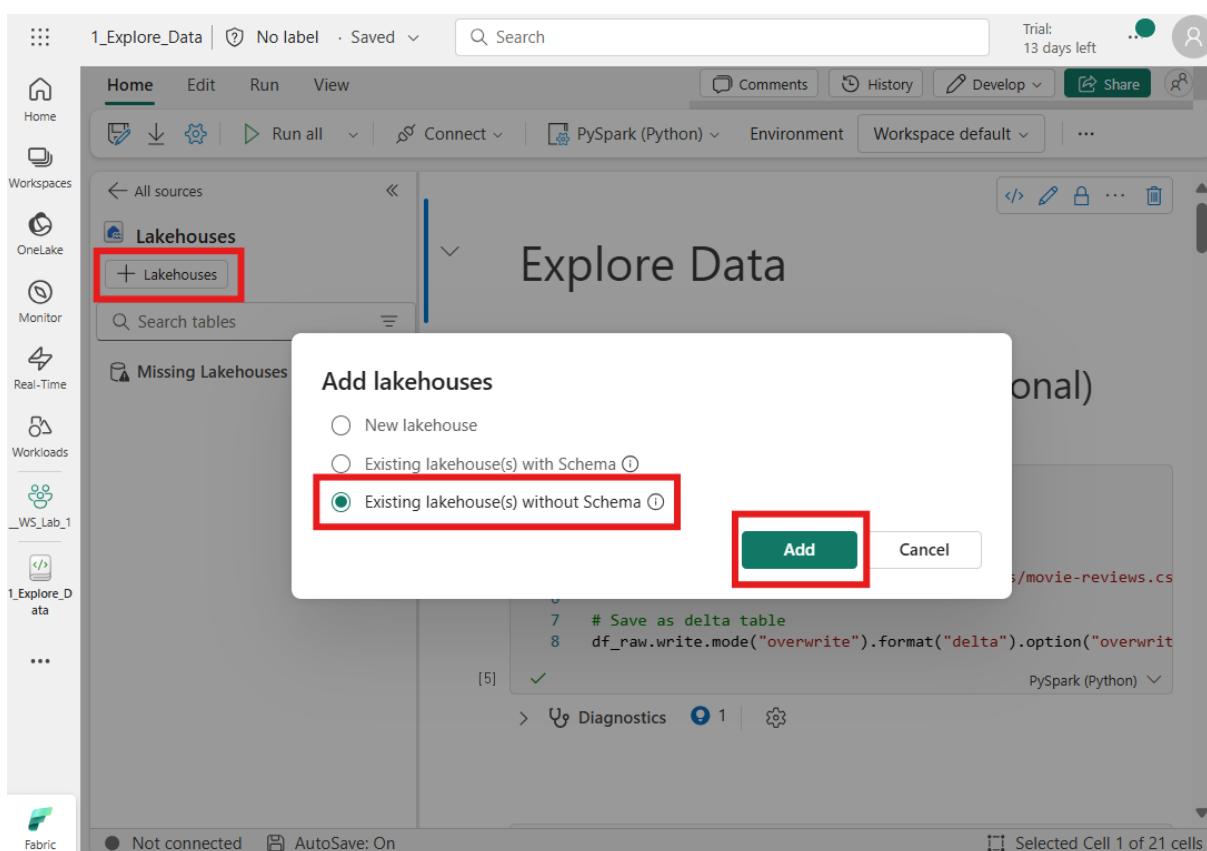
2. Lakehouses 항목을 선택해 앞서 생성한 Lakehouse를 추가합니다.



The screenshot shows the Databricks workspace interface. The left sidebar is titled 'Workspaces' and contains sections for 'Home', 'Monitor', 'Real-Time', 'Workloads', and 'Fabric'. The 'Home' section is currently selected. The 'Explorer' section on the left shows a folder structure: 'Resources' (1 item added), 'Lakehouses' (1 item added, highlighted with a red box), and 'Warehouses' (0 items added). The main area is titled 'Explore Data' and contains the heading '0. Preprocess Raw Data (Optional)'. Below the heading is a code cell in 'PySpark (Python)' showing the following code:

```
1  from pyspark.sql import SparkSession
2  from pyspark.sql import DataFrame
3
4  # Read df from csv file
5  df_raw = spark.read.csv("Files/saved_datafiles/movie-reviews.cs
6
7  # Save as delta table
8  df_raw.write.mode("overwrite").format("delta").option("overwrit
[5]  ✓
```

The code cell has a green checkmark and is labeled 'PySpark (Python)'. At the bottom of the workspace are buttons for 'Diagnostics' and 'Selected Cell 1 of 21 cells'.



The screenshot shows the 'Lakehouses' section in the Explorer sidebar with a red box around the '+ Lakehouses' button. A modal dialog box titled 'Add lakehouses' is open in the center. The dialog contains three radio button options:

- New lakehouse
- Existing lakehouse(s) with Schema ⓘ
- Existing lakehouse(s) without Schema ⓘ

At the bottom right of the dialog are 'Add' and 'Cancel' buttons, with the 'Add' button highlighted by a red box. The background workspace shows the same code cell as the previous screenshot.

Discover data from your org and beyond and use it to create reports

Filter by keyword

All	My data	Endorsed in your org	Favorites	Filter by keyword
user1				
Lakehouse_MovieData				
Lakehouse_Silver				
Lakehouse_RAG_Draft				
Lakehouse_Gold				
Lakehouse_Bronze				

Add Cancel

3. 추가한 Lakehouse를 Default Lakehouse로 변경합니다.

1Explore_Data(1) | No label · Saved · Trial: 13 days left

Home Edit Run View

Comments History Develop Share

Home Workspaces OneLake Monitor Real-Time

Lakehouses + Lakehouses

Search tables

user1

Tables

Explore Data

0. Preprocess Raw Data (Optional)

1Explore_Data(1) | No label · Saved · Trial: 13 days left

Home Edit Run View

Comments History Develop Share

Home Workspaces OneLake Monitor Real-Time

Lakehouses + Lakehouses

Search tables

user1

Missing Lakehouses

61c133de-2946-4026-...

_WS_Lab_1 Set as default lakehouse

user1

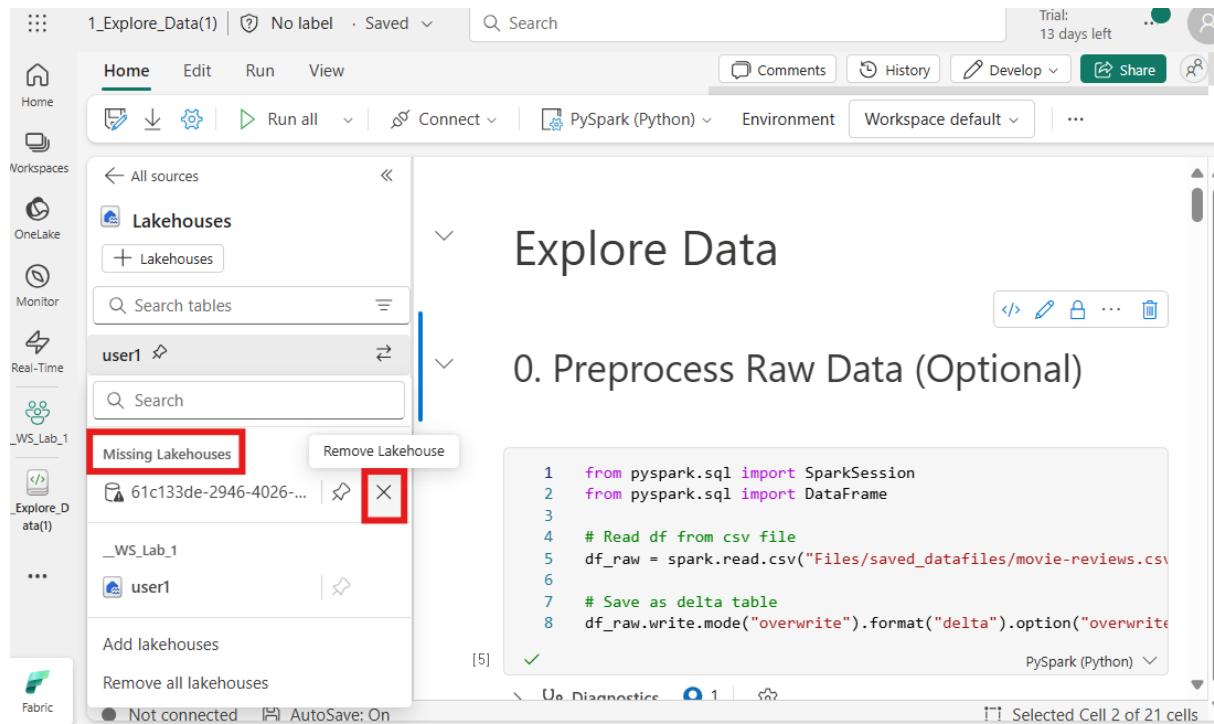
Explore Data

0. Preprocess Raw Data (Optional)

```

1 from pyspark.sql import SparkSession
2 from pyspark.sql import DataFrame
3
4 # Read df from csv file
5 df_raw = spark.read.csv("Files/saved_datafiles/movie-reviews.csv")
6
7 # Save as delta table
  
```

4. Missing Lakehouse는 삭제합니다.



The screenshot shows the Databricks Fabric interface. On the left, the 'Lakehouses' section is open, showing a list of lakehouses. A red box highlights the 'Missing Lakehouses' section, and a red 'X' button is highlighted within it, indicating a delete operation. The main panel displays a notebook cell with the following Python code:

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql import DataFrame
3
4 # Read df from csv file
5 df_raw = spark.read.csv("Files/saved_datafiles/movie-reviews.csv")
6
7 # Save as delta table
8 df_raw.write.mode("overwrite").format("delta").option("overwrit
```

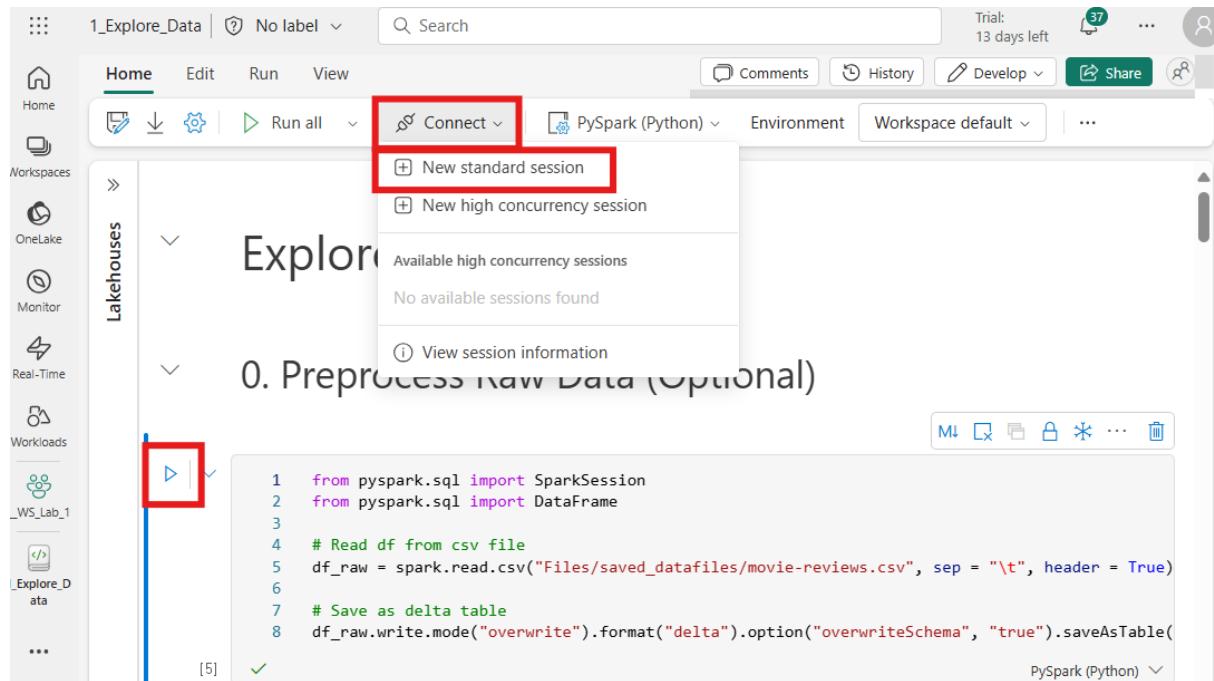
5. 다른 Fabric Notebook에서도 Step 1-4를 반복합니다.

Prepare OneLake Data for Connection

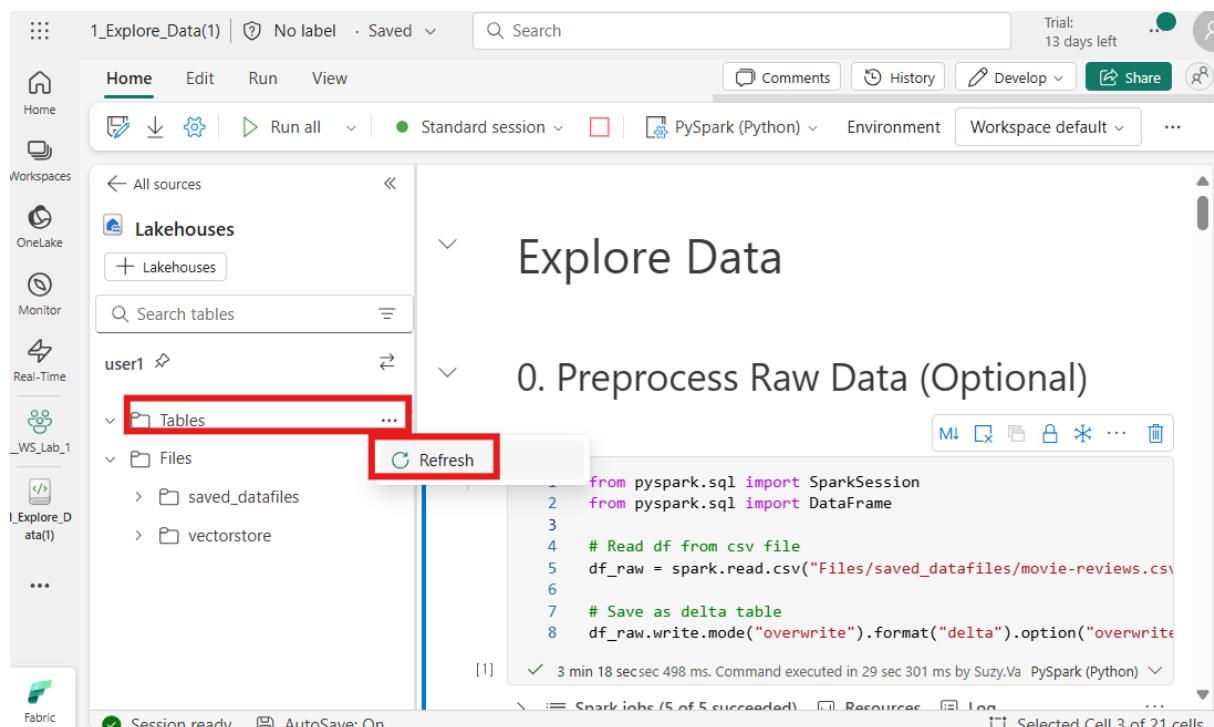
Explore Data

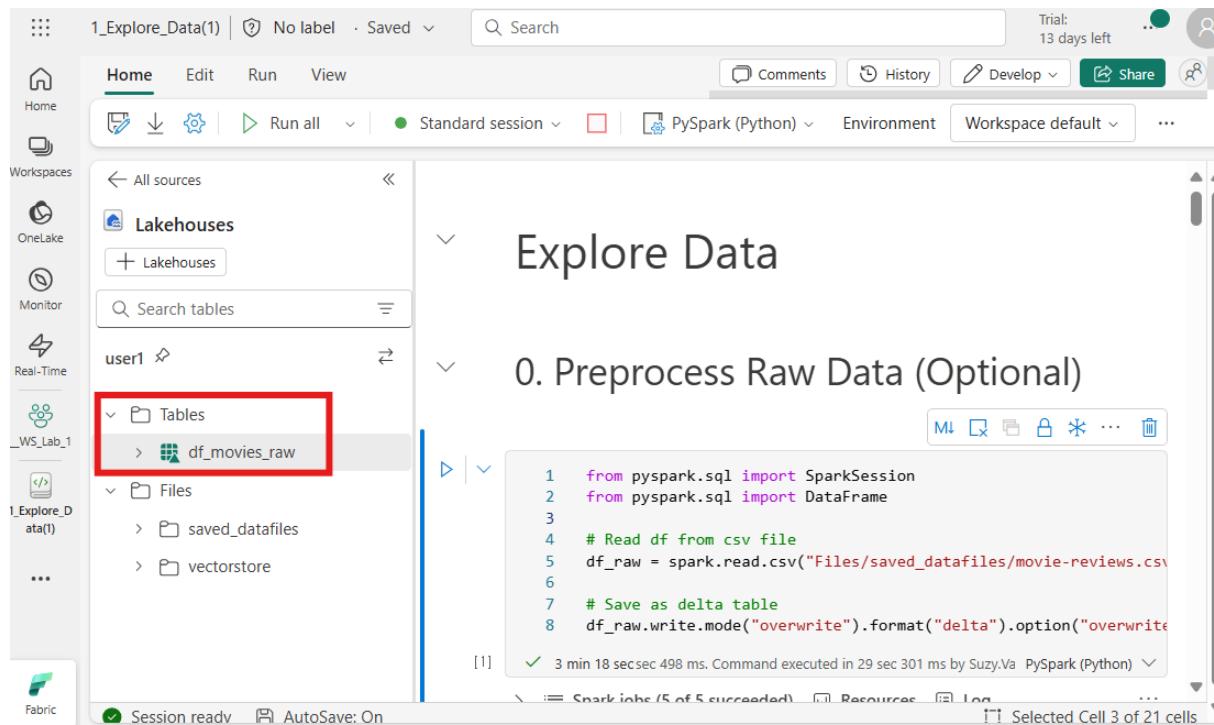
Fabric Notebook 1_Explore_Data를 실행해 데이터를 탐색하고 전처리합니다.

1. 새로운 Spark Session을 연결합니다.



2. 세션이 시작되고 첫번째 셀을 실행하면 Delta Table이 생성된 것을 확인할 수 있습니다.





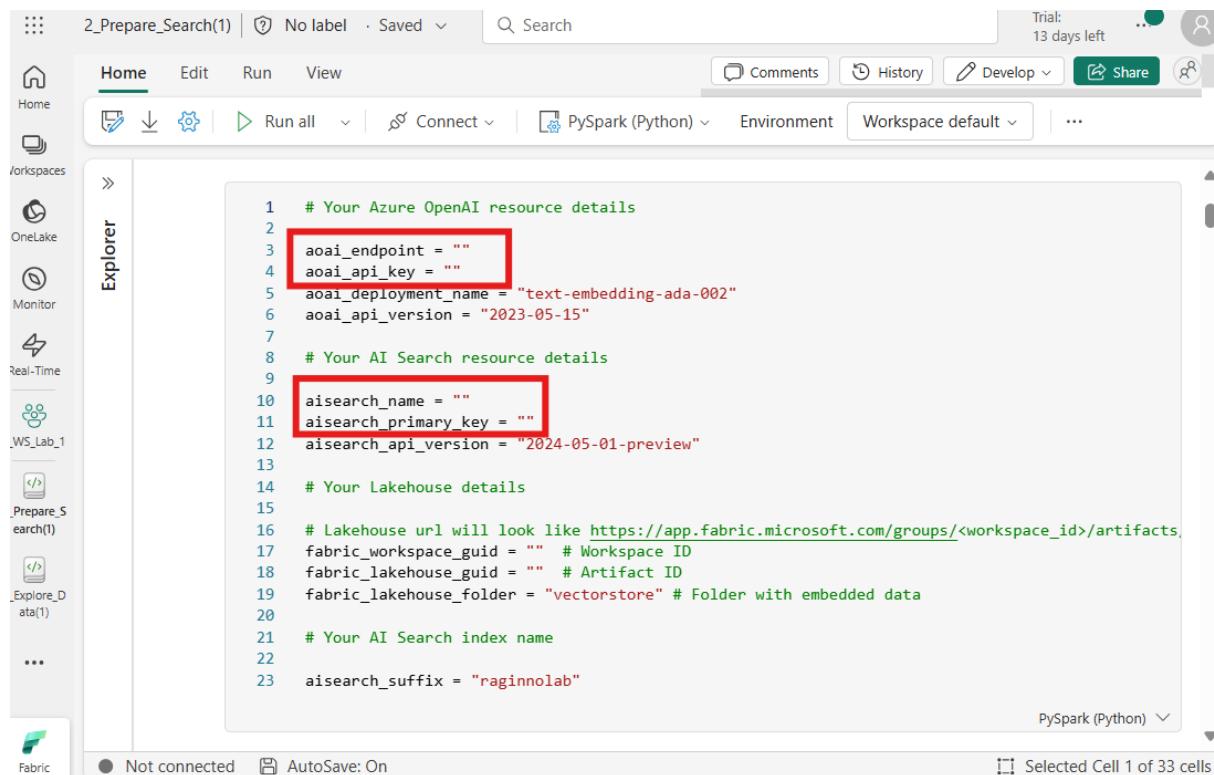
The screenshot shows the Azure Fabric Notebook interface. The left sidebar displays 'Workspaces' with 'OneLake' and 'Monitor' selected. The main area is titled 'Explore Data' and contains a section '0. Preprocess Raw Data (Optional)'. A code cell is shown, with the line 'df_raw.write.mode("overwrite").format("delta").option("overwrit' highlighted with a red box. The status bar at the bottom indicates 'Session ready' and 'AutoSave: On'.

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql import DataFrame
3
4 # Read df from csv file
5 df_raw = spark.read.csv("Files/saved_datafiles/movie-reviews.csv")
6
7 # Save as delta table
8 df_raw.write.mode("overwrite").format("delta").option("overwrit
```

Prepare Search

Fabric Notebook 2_Prepares_Search를 실행해 데이터를 임베딩하고 AI Search 인덱스를 생성합니다.

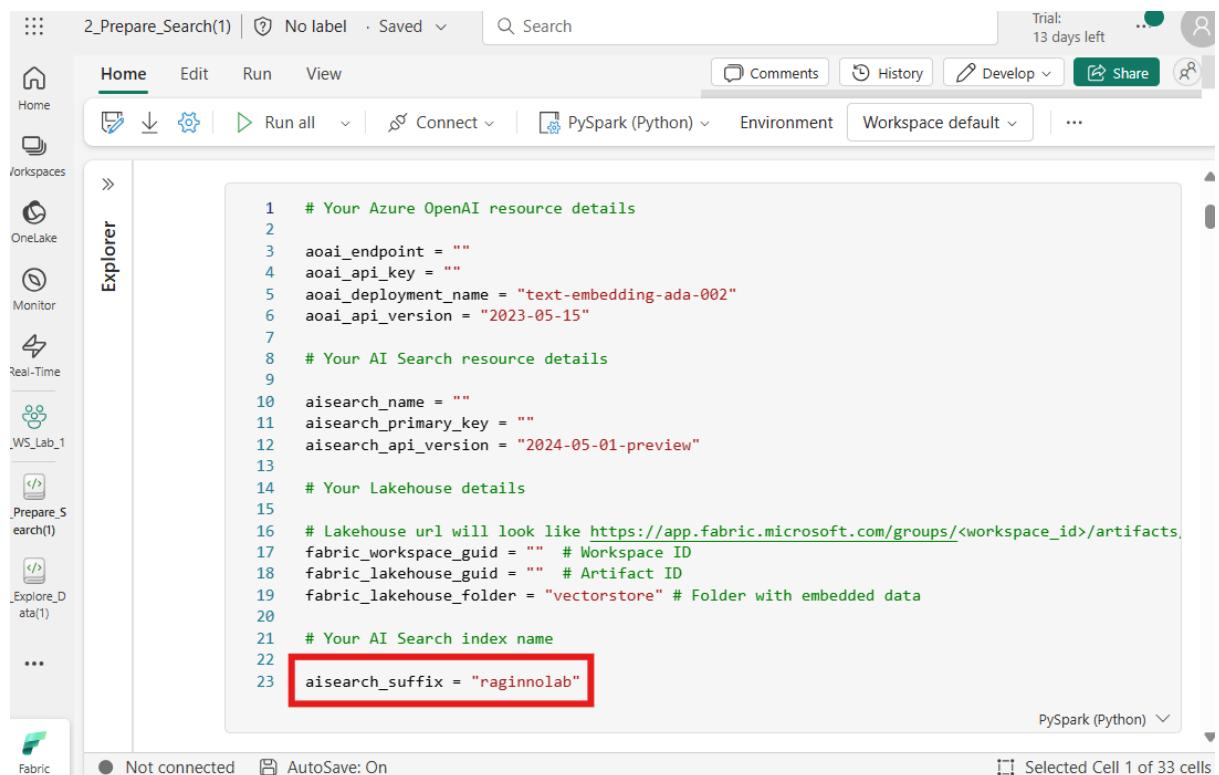
1. User Setting 정보를 바탕으로 Azure OpenAI와 AI Search의 리소스 정보를 추가합니다.



The screenshot shows the Azure Fabric Notebook interface. The left sidebar displays 'Workspaces' with 'OneLake' and 'Monitor' selected. The main area is titled 'Explorer' and contains a code cell with configuration variables. Lines 3 and 10 are highlighted with red boxes. The status bar at the bottom indicates 'Not connected' and 'AutoSave: On'.

```
1 # Your Azure OpenAI resource details
2
3 aoai_endpoint = ""
4 aoai_api_key = ""
5 aoai_deployment_name = "text-embedding-ada-002"
6 aoai_api_version = "2023-05-15"
7
8 # Your AI Search resource details
9
10 aisearch_name = ""
11 aisearch_primary_key = ""
12 aisearch_api_version = "2024-05-01-preview"
13
14 # Your Lakehouse details
15
16 # Lakehouse url will look like https://app.fabric.microsoft.com/groups/<workspace_id>/artifacts
17 fabric_workspace_guid = "" # Workspace ID
18 fabric_lakehouse_guid = "" # Artifact ID
19 fabric_lakehouse_folder = "vectorstore" # Folder with embedded data
20
21 # Your AI Search index name
22
23 aisearch_suffix = "raginnolab"
```

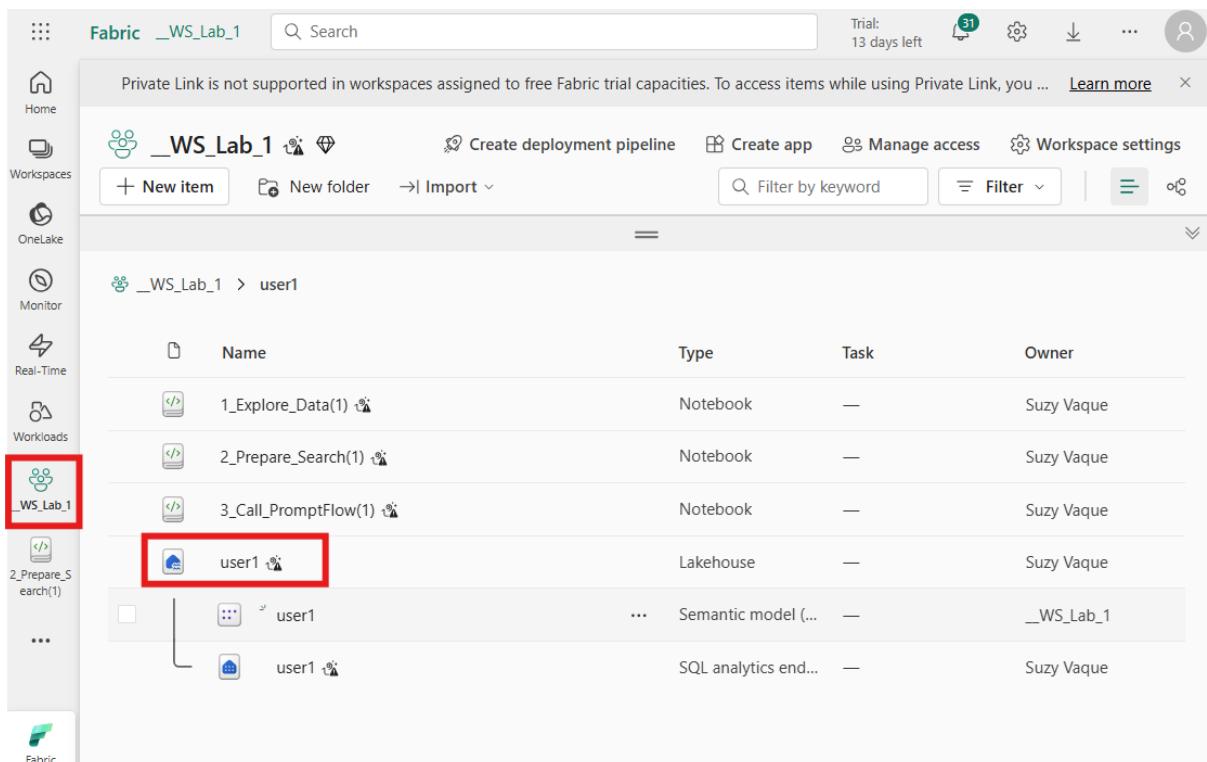
2. AI Search의 Suffix로는 Custom Suffix를 작성합니다.



```
1 # Your Azure OpenAI resource details
2
3 aoai_endpoint = ""
4 aoai_api_key = ""
5 aoai_deployment_name = "text-embedding-ada-002"
6 aoai_api_version = "2023-05-15"
7
8 # Your AI Search resource details
9
10 aistore_name = ""
11 aistore_primary_key = ""
12 aistore_api_version = "2024-05-01-preview"
13
14 # Your Lakehouse details
15
16 # Lakehouse url will look like https://app.fabric.microsoft.com/groups/<workspace_id>/artifacts
17 fabric_workspace_guid = "" # Workspace ID
18 fabric_lakehouse_guid = "" # Artifact ID
19 fabric_lakehouse_folder = "vectorstore" # Folder with embedded data
20
21 # Your AI Search index name
22
23 aistore_suffix = "raginnolab"
```

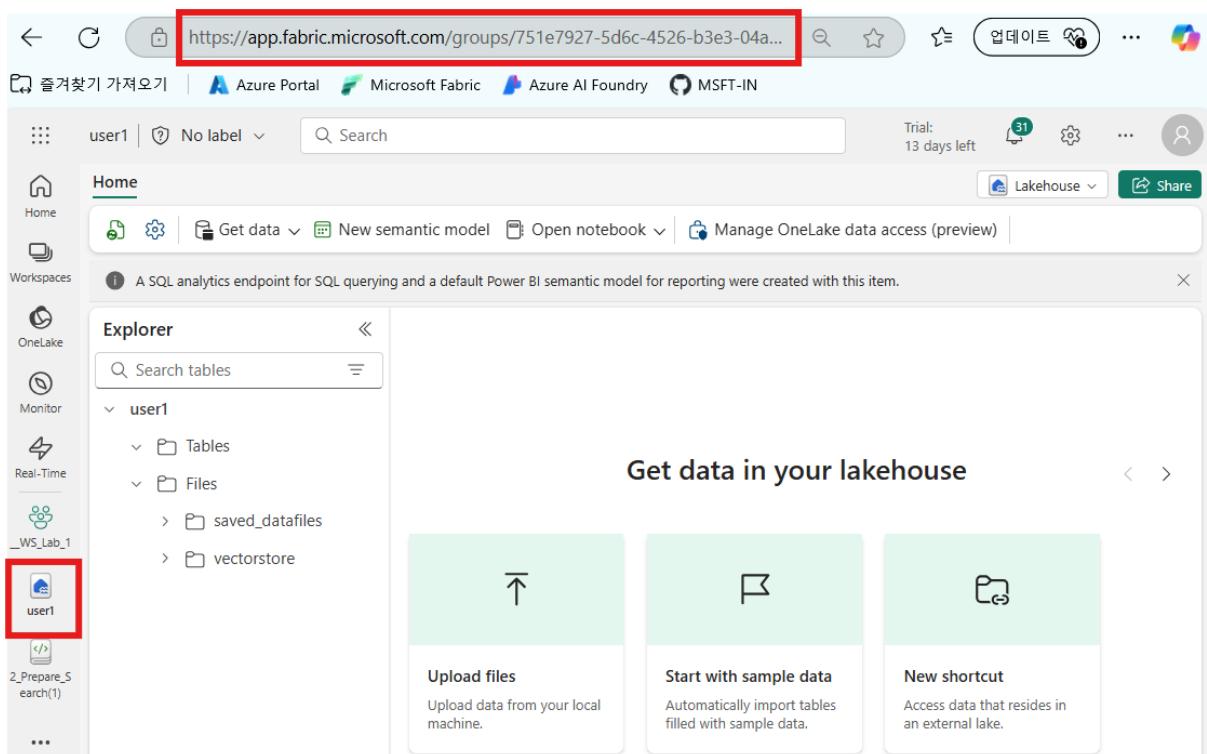
3. Lakehouse 정보를 업데이트합니다.

A. Workspace 폴더의 Lakehouse를 선택합니다.



Name	Type	Task	Owner
1_Explore_Data(1)	Notebook	—	Suzy Vaque
2_Prepares_Search(1)	Notebook	—	Suzy Vaque
3_Call_PromptFlow(1)	Notebook	—	Suzy Vaque
user1	Lakehouse	—	Suzy Vaque
user1	Semantic model (...)	—	_WS_Lab_1
user1	SQL analytics end...	—	Suzy Vaque

B. Lakehouse에 접속한 상태에서 url을 확인합니다.

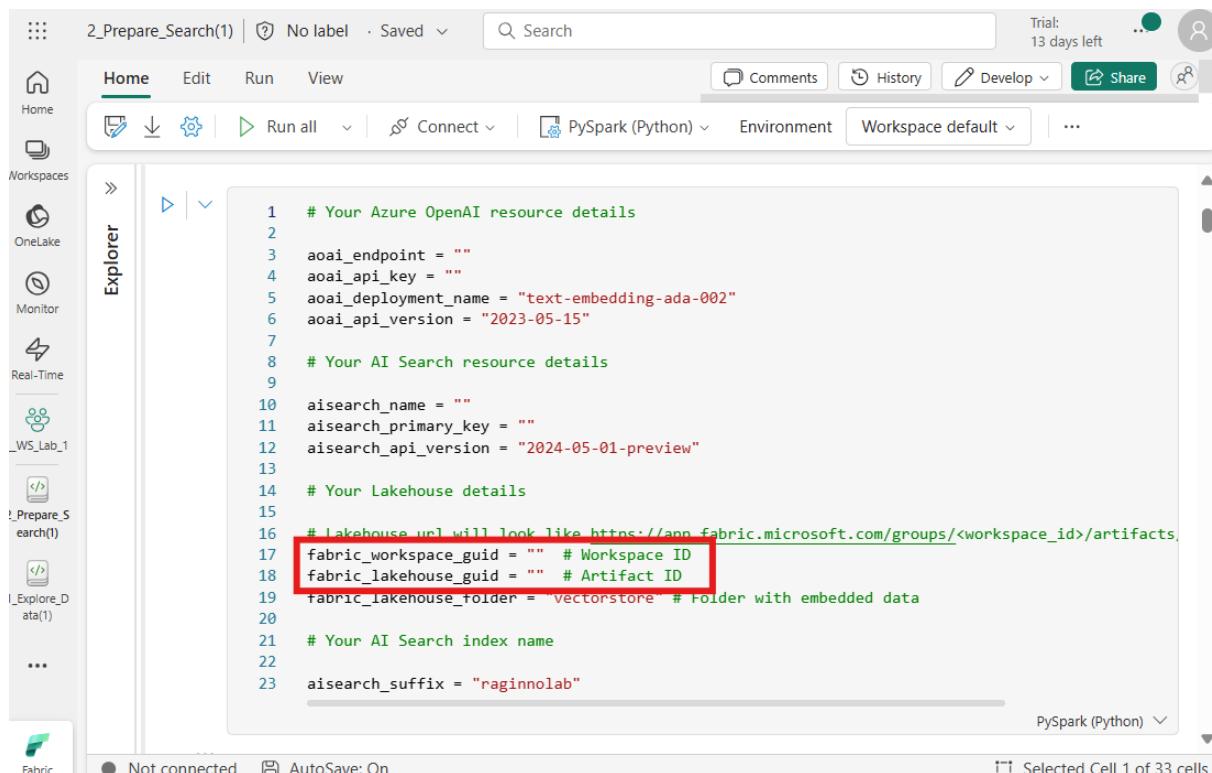


The screenshot shows the Microsoft Fabric Home page. The URL in the browser's address bar is highlighted with a red box: <https://app.fabric.microsoft.com/groups/751e7927-5d6c-4526-b3e3-04a...>. The page displays a sidebar with workspaces like 'user1', 'WS_Lab_1', and '2_Prep...'. The main area shows an 'Explorer' view with 'Tables' and 'Files' under 'user1', and three buttons for 'Upload files', 'Start with sample data', and 'New shortcut'.

C. url의 구조는 다음과 같습니다.

[https://app.fabric.microsoft.com/groups/{WORKSPACE_GUID}/lakehouses/{ARTIFACT_GUID}](https://app.fabric.microsoft.com/groups/{WORKSPACE_GUID}/lakehouses/{ARTIFACT_GUID}?experience=fabric-developer)
?experience=fabric-developer

D. Workspace GUID와 Artifact GUID 값으로 노트북의 정보를 업데이트합니다.



The screenshot shows the Microsoft Fabric Notebook interface. The code in the cell is as follows:

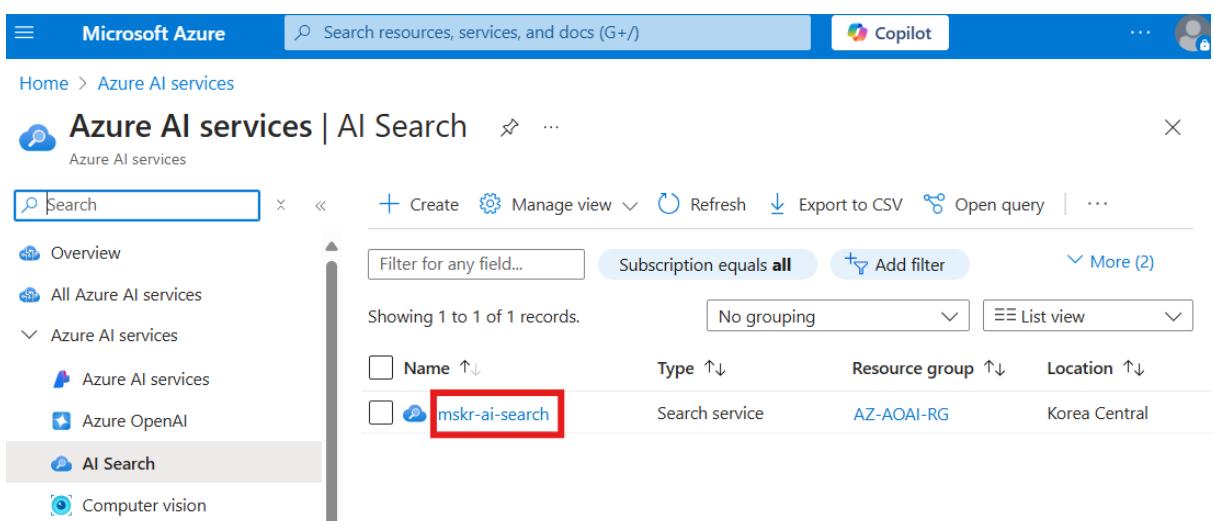
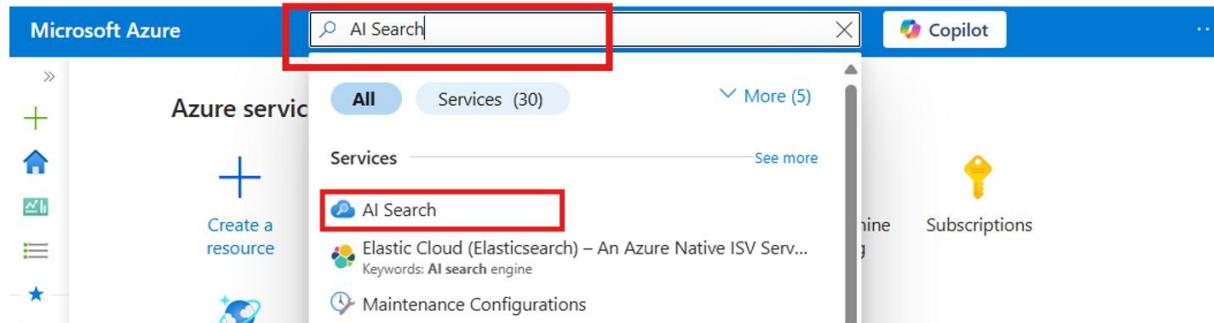
```
1 # Your Azure OpenAI resource details
2
3 aoai_endpoint = ""
4 aoai_api_key = ""
5 aoai_deployment_name = "text-embedding-ada-002"
6 aoai_api_version = "2023-05-15"
7
8 # Your AI Search resource details
9
10 aisearch_name = ""
11 aisearch_primary_key = ""
12 aisearch_api_version = "2024-05-01-preview"
13
14 # Your Lakehouse details
15
16 # Lakehouse url will look like https://app.fabric.microsoft.com/groups/<workspace_id>/artifacts
17 fabric_workspace_guid = "" # Workspace ID
18 fabric_lakehouse_guid = "" # Artifact ID
19 fabric_lakehouse_folder = "vectorstore" # Folder with embedded data
20
21 # Your AI Search index name
22
23 aisearch_suffix = "raginnolab"
```

The line `fabric_lakehouse_guid = "" # Artifact ID` is highlighted with a red box.

Check AI Search Resource

Fabric Notebook 2_Prepares_Search까지 완료했다면, AI Search 리소스에 새로운 Index가 생성됐을 것입니다.

1. [Azure Portal](#)에서 AI Search를 검색하고 자신의 리소스를 선택합니다.



Name	Type	Resource group	Location
mskr-ai-search	Search service	AZ-AOAI-RG	Korea Central

2. 노트북에서 SDK를 통해 생성한 Index, Indexer, Data Source를 확인할 수 있습니다.

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

Home > Azure AI services | AI Search > mskr-ai-search

mskr-ai-search | Resource visualizer

Search service

Search Choose resources Reset diagram Zoom to fit Refresh Export ...

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Resource visualizer

Search management

- Indexes
- Indexers
- Data sources

Aliases

Resource visualizer

Indexes

Indexers

Data sources

Aliases

3. Data Source를 확인해보면, Indexer가 One Lake를 바라보고 있음을 확인할 수 있습니다.

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

Home > mskr-ai-search

mskr-ai-search | Data sources

Search service

Search Add data source Refresh Delete

Filter by name...

Name	Type	Table/Colle...
onelake-datasource-raginnolab	Fabric OneLake files (Preview)	
Azure Data Lake Storage Gen2	pdf2	
Azure Cosmos DB	MovieReviews	
Fabric OneLake files (Preview)		
Fabric OneLake files (Preview)		
Azure Data Lake Storage Gen2	pdf	
Azure Data Lake Storage Gen2	pdf	

Indexes

Indexers

Data sources

Aliases

Skillsets

Create AI Applications on OneLake Data

Get Started with PromptFlow

1. Foundry Portal의 AI Project로 이동합니다.

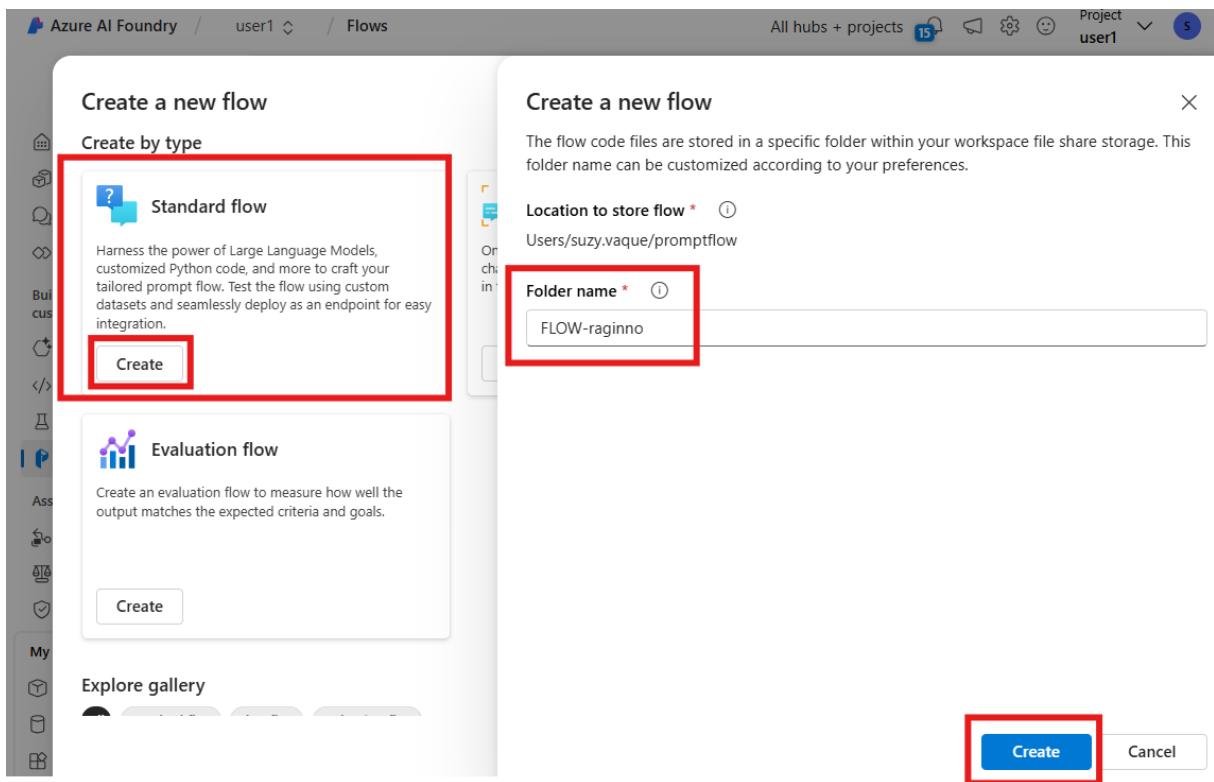
The screenshot shows the Azure AI Foundry interface. At the top, there is a navigation bar with icons for notifications, settings, and user profile. Below the bar, a title 'Jump into a project in Azure AI Foundry' is displayed, along with 'View all projects' and a 'Create project' button. A 'Help' link is also present. The main area is a table listing projects:

Project	Created on	Location	Hub	Description
user1	Mar 23, 2025 10:57 ...	eastus	MSKR-AIHub-EastUS	
MSKR-AIHub-Eastus-Project1	Feb 11, 2025 6:00 PM	eastus	MSKR-AIHub-EastUS	
MSKR-AIHub-Eastus-Project2	Feb 23, 2025 11:35 ...	eastus	MSKR-AIHub-EastUS	
user1	Mar 23, 2025 10:57 ...	eastus	MSKR-AIHub-EastUS	
MSKR-AIHub-Eastus-Project2	Feb 23, 2025 11:35 ...	eastus	MSKR-AIHub-EastUS	

Below the table, a section titled 'Work outside of a project' is shown. It includes a 'Focused on Azure OpenAI Service?' section with a note about building with Azure OpenAI Service models and features. To the right is a 'Chat playground' window with a message: 'When did Mona say that planning project and what is the timeline in Planning Document?'.

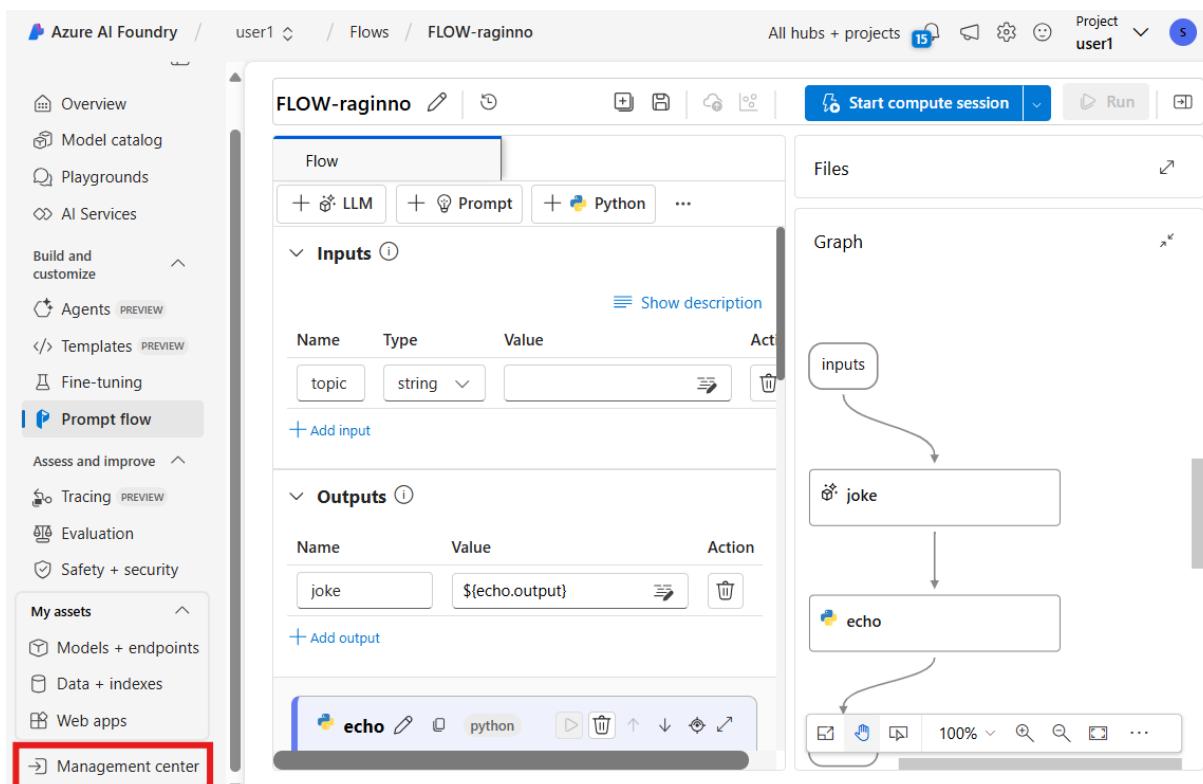
2. Prompt Flow 서비스로 이동해 새로운 Standard Flow를 생성합니다.

The screenshot shows the 'Flows' page in the Azure AI Foundry portal. The left sidebar has a 'Prompt flow' section highlighted with a red box. The main area has a title 'Create, iterate, and debug your orchestration flows' and tabs for 'Flows' and 'Runs'. A 'Create' button is highlighted with a red box. Below the tabs are buttons for 'Refresh', 'Archive', 'Reset view', and checkboxes for 'View only my flows' and 'Include archived'. The central area shows a small diagram of three nodes connected by lines. A message at the bottom says 'You have not created any of your own flows'.

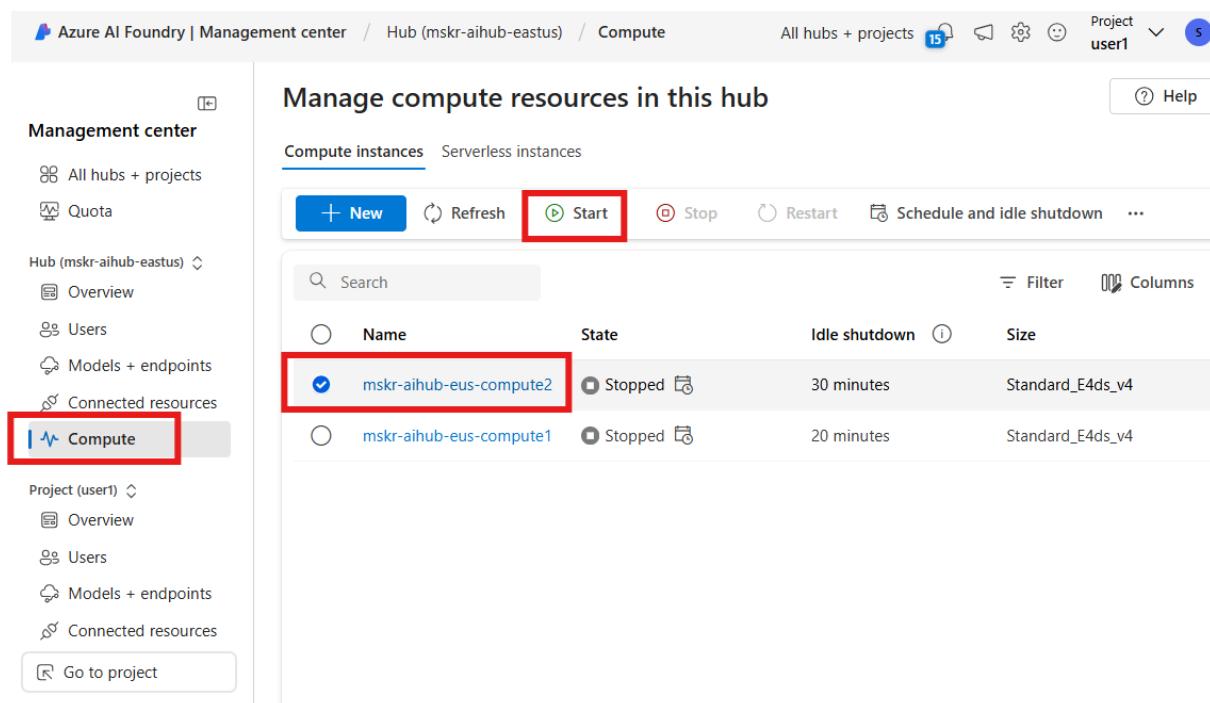


3. Flow를 실행하기 위해 배정받은 Compute Instance를 연결합니다.

A. AI Hub의 Management Center로 이동합니다.



B. 배정받은 Compute를 실행합니다.

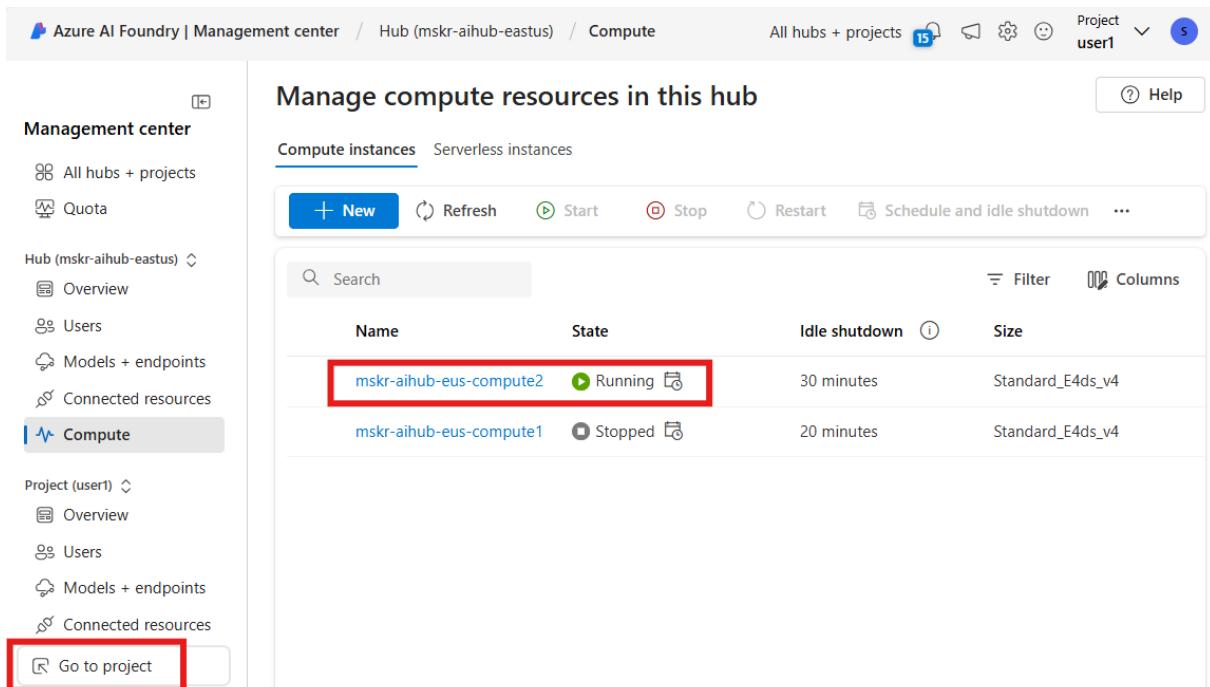


Manage compute resources in this hub

Compute instances Serverless instances

Name	State	Idle shutdown	Size
mskr-aihub-eus-compute2	Stopped	30 minutes	Standard_E4ds_v4
mskr-aihub-eus-compute1	Stopped	20 minutes	Standard_E4ds_v4

C. Compute 실행이 완료되면 다시 AI Project의 Flow로 이동합니다.

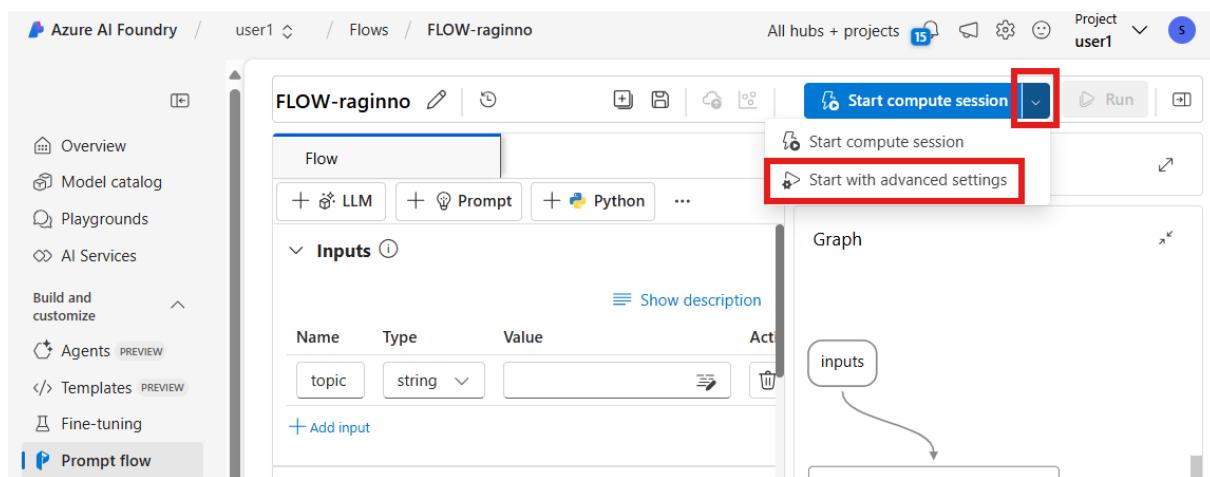


Manage compute resources in this hub

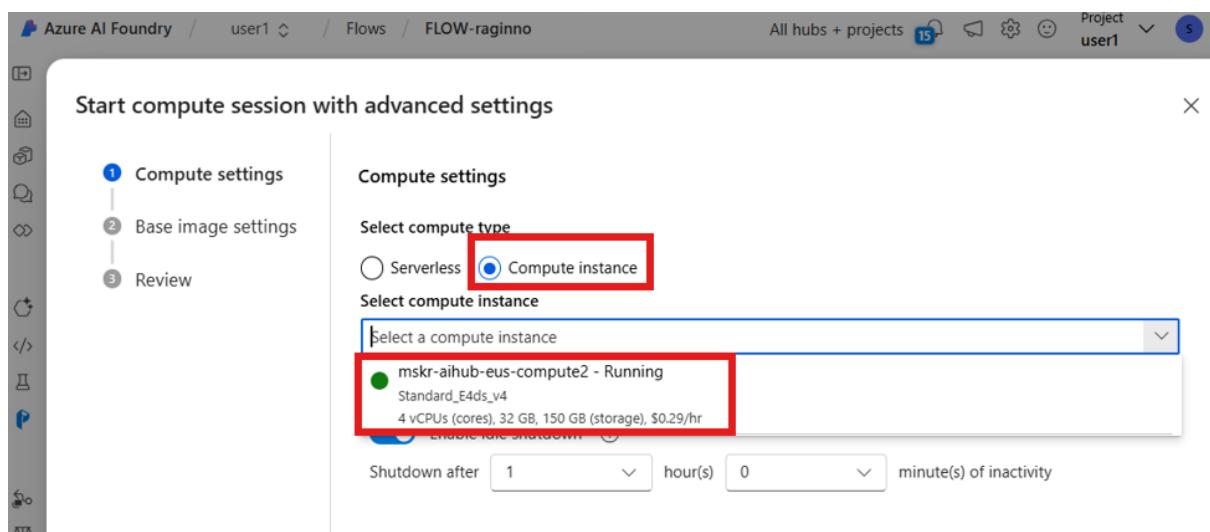
Compute instances Serverless instances

Name	State	Idle shutdown	Size
mskr-aihub-eus-compute2	Running	30 minutes	Standard_E4ds_v4
mskr-aihub-eus-compute1	Stopped	20 minutes	Standard_E4ds_v4

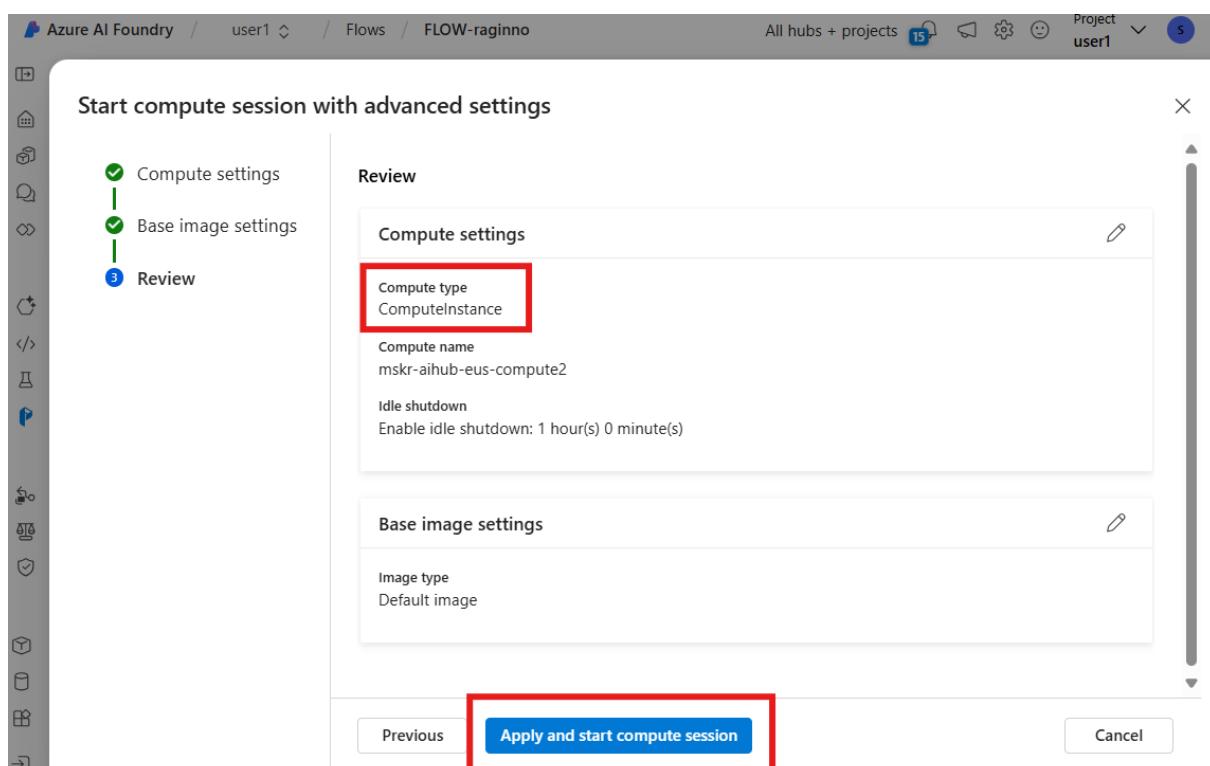
D. Compute Session의 Start with Advanced Settings 옵션을 선택해 Compute Instance를 연결합니다.



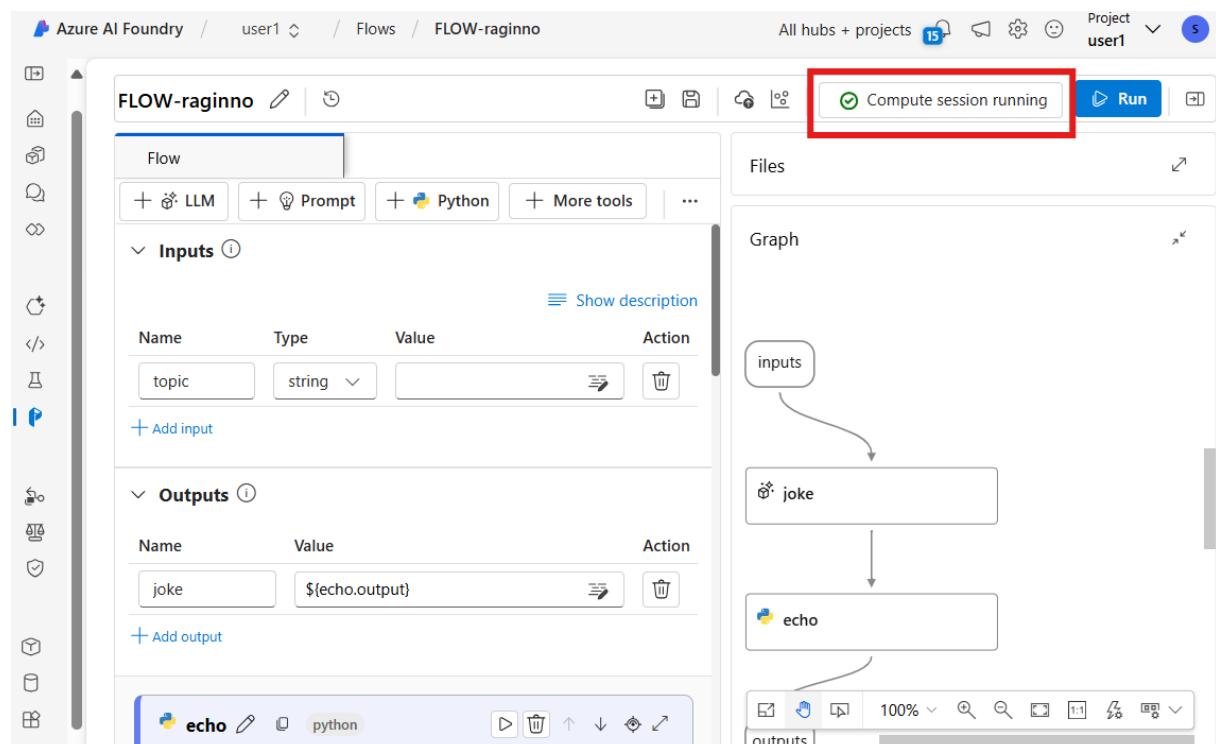
The screenshot shows the Azure AI Foundry interface. In the top right, there are project and user settings. Below the header, there are navigation links for Overview, Model catalog, Playgrounds, AI Services, and a section for Build and customize with Agents, Templates, and Fine-tuning. A sidebar on the left has a 'Prompt flow' tab selected. The main area is titled 'FLOW-raginno' and shows a 'Flow' configuration. At the top right of this area, there are two buttons: 'Start compute session' and 'Start with advanced settings', both of which are highlighted with red boxes. Below the flow configuration, there is an 'Inputs' section with a table and a 'Graph' section showing a flowchart with an 'inputs' node.



The screenshot shows the 'Start compute session with advanced settings' dialog. On the left, a navigation sidebar shows steps 1 (Compute settings), 2 (Base image settings), and 3 (Review). The main area is titled 'Compute settings' and 'Select compute type'. It shows two options: 'Serverless' (radio button not selected) and 'Compute instance' (radio button selected and highlighted with a red box). Below this is a 'Select compute instance' dropdown menu. A list item 'mskr-aihub-eus-compute2 - Running' is selected and highlighted with a red box. The list also includes 'Standard_E4ds_v4' and '4 vCPUs (cores), 32 GB, 150 GB (storage), \$0.29/hr'. At the bottom of the dialog, there is a 'Shutdown after' field set to '1 hour(s) 0 minute(s)'.



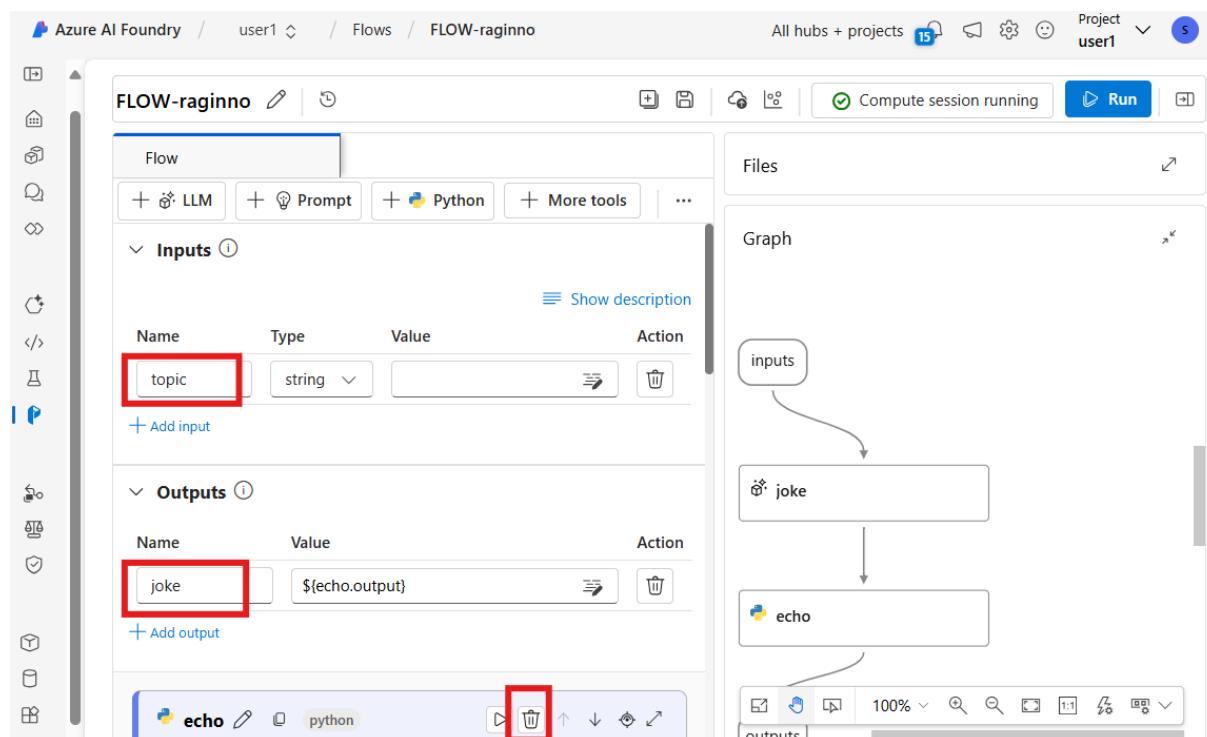
The screenshot shows the 'Start compute session with advanced settings' dialog in the 'Review' step. On the left, the navigation sidebar shows steps 1 (Compute settings), 2 (Base image settings), and 3 (Review), with step 3 highlighted. The main area is titled 'Review' and contains two sections: 'Compute settings' and 'Base image settings'. In the 'Compute settings' section, the 'Compute type' field is highlighted with a red box and shows 'ComputeInstance'. Below it, the 'Compute name' is listed as 'mskr-aihub-eus-compute2' and the 'Idle shutdown' is set to 'Enable idle shutdown: 1 hour(s) 0 minute(s)'. In the 'Base image settings' section, the 'Image type' is listed as 'Default image'. At the bottom of the dialog, there are 'Previous' and 'Apply and start compute session' buttons, with 'Apply and start compute session' highlighted with a red box.



The screenshot shows the Azure AI Foundry interface for creating a flow. The flow is named 'FLOW-raginno'. It has an 'Inputs' section with a single input named 'topic' of type 'string'. It has an 'Outputs' section with one output named 'joke' with the value '\${echo.output}'. The flow graph shows an 'inputs' node connected to a 'joke' node, which then connects to an 'echo' node. The 'echo' node has an 'outputs' section. The 'Compute session running' button is highlighted with a red box.

Create RAG Flow

1. Flow에 기본적으로 생성된 Input, Output, LLM Node의 이름을 적절하게 변경하고, Python Node를 삭제합니다.

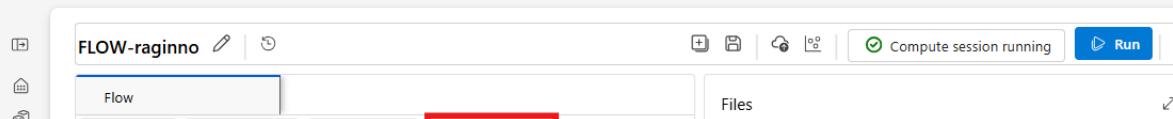


The screenshot shows the Azure AI Foundry interface for creating a flow. The flow is named 'FLOW-raginno'. The 'Inputs' section has a 'topic' node highlighted with a red box. The 'Outputs' section has a 'joke' node highlighted with a red box. The Python node's delete icon is also highlighted with a red box.

The screenshot shows the Azure AI Foundry interface with the following details:

- Header:** Azure AI Foundry / user1 / Flows / FLOW-raginno
- Top Right:** All hubs + projects (15), Notifications, Settings, Project user1, and a blue Run button.
- Left Sidebar:** A vertical toolbar with various icons for file operations, including a plus sign, a document, a folder, a database, a user, a gear, and a search.
- Flow Configuration:**
 - Inputs:** A table with a row for "question" (Type: string). The "question" input is highlighted with a red box.
 - Outputs:** A table with a row for "rag_response" (Value: \${echo.output}). The "rag_response" output is highlighted with a red box.
 - Bottom Bar:** A search bar with "AOAI" and buttons for "llm", "Show variants", and "Generate".
- Graph View:** A diagram showing a flow from an "inputs" node to a "joke" node. The "inputs" node is a rounded rectangle, and the "joke" node is a light blue rounded rectangle with a "joke" icon. A blue arrow points from the "inputs" node to the "joke" node.
- Bottom Right:** A toolbar with icons for file operations, a magnifying glass for search, and a "100%" view.

2. Index Lookup Tool을 추가합니다.



The screenshot shows the Azure AI Foundry interface with the project 'FLOW-raginno' selected. The top navigation bar includes 'All hubs + projects' (15), 'Compute session running', and a 'Run' button. A red box highlights the 'More tools' button in the top navigation bar. A second red box highlights the 'Index Lookup' tool in the dropdown menu that appears when clicking 'More tools'. The main workspace shows the 'Flow' tab selected, with 'Inputs' defined for a 'question' input of type 'string'. A 'Compute session running' status bar at the bottom indicates 'Azure OpenAI GPT-4 Turbo with Vision' and 'Preview'.

The screenshot shows the FLOW-raginno interface. The top bar includes a 'Compute session running' status and 'Run' buttons. The main area has a 'Flow' tab selected, showing a code editor with the following Python script:

```
1  #!/usr/bin/env python3
2  # system:
3  # You are a bot can tell good jokes
4  # user:
5  A joke about {{topic}} please
```

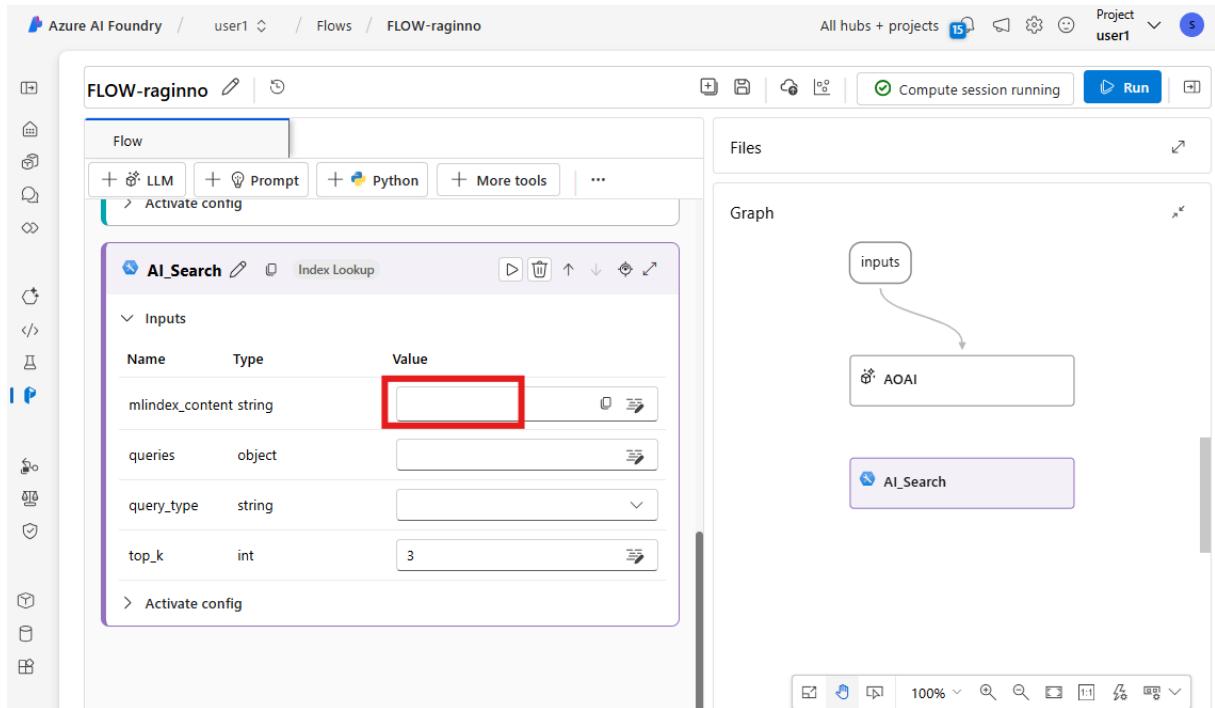
Below the code editor is a 'Inputs' section with a 'Validate and parse input' button. A table lists an input named 'topic' of type 'string' with the value '\$(inputs.topic)'. A 'Activate config' button is also present.

At the bottom, a navigation bar features buttons for 'AI_Search' (highlighted with a red box), 'Index Lookup' (highlighted with a red box), and 'Add'.

The right side of the interface shows a 'Graph' section with a diagram: 'inputs' points to an 'AOAI' block.

3. Index Lookup에 OneLake 기반 AI Search Index를 연결합니다.

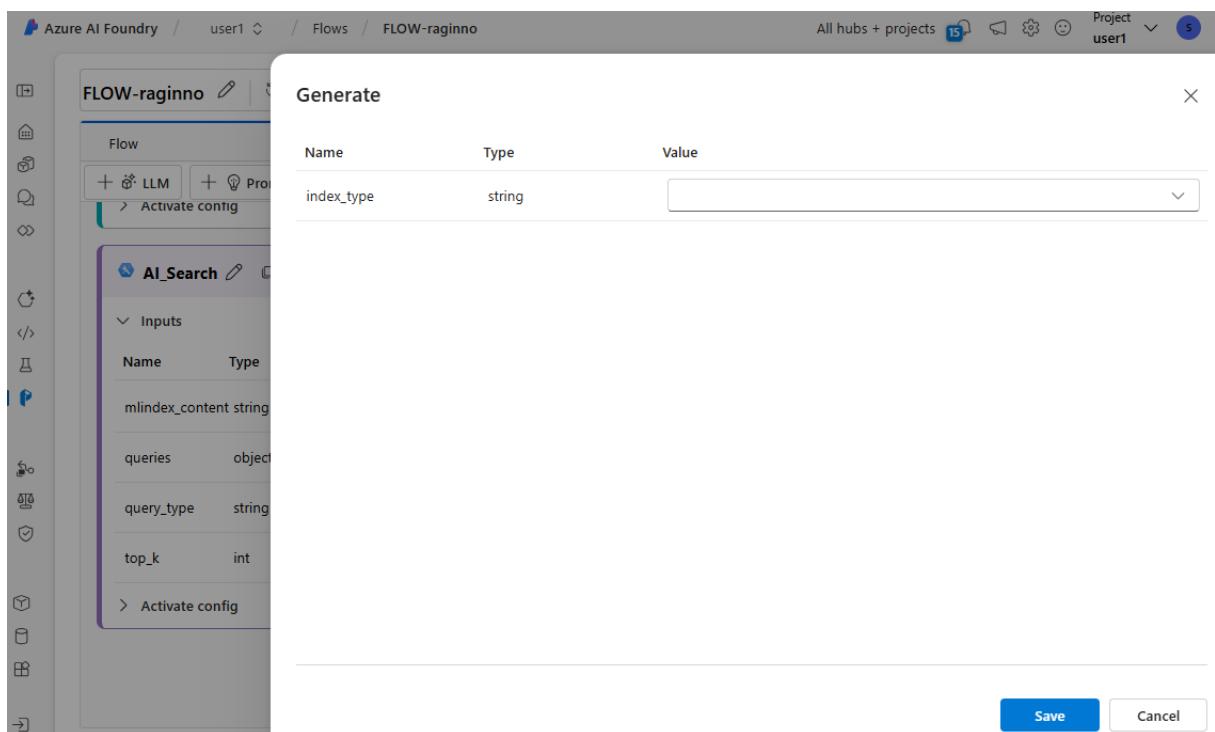
A. Lookup Generate으로 이동합니다.



The screenshot shows the Azure AI Foundry interface with the 'FLOW-raginno' flow selected. The 'AI_Search' component is open, showing its configuration. The 'Inputs' section contains the following fields:

Name	Type	Value
mlindex_content	string	<input type="text"/>
queries	object	<input type="text"/>
query_type	string	<input type="text"/>
top_k	int	3

The 'Value' column for 'mlindex_content' is highlighted with a red box. The 'Graph' panel on the right shows a flow from 'inputs' to 'AOAI' (AI_Search) and then to 'AI_Search'.

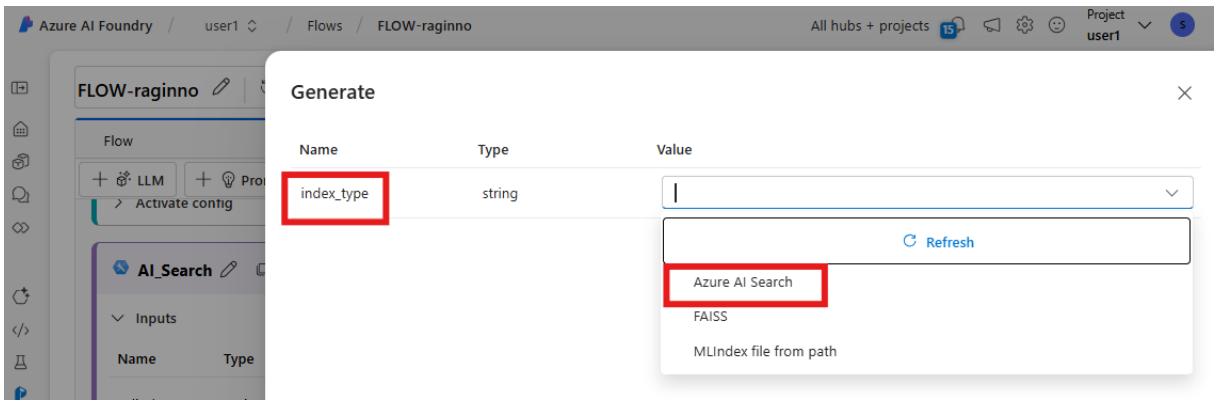


The screenshot shows the 'Generate' dialog box in the Azure AI Foundry interface. It lists an input field:

Name	Type	Value
index_type	string	<input type="text"/>

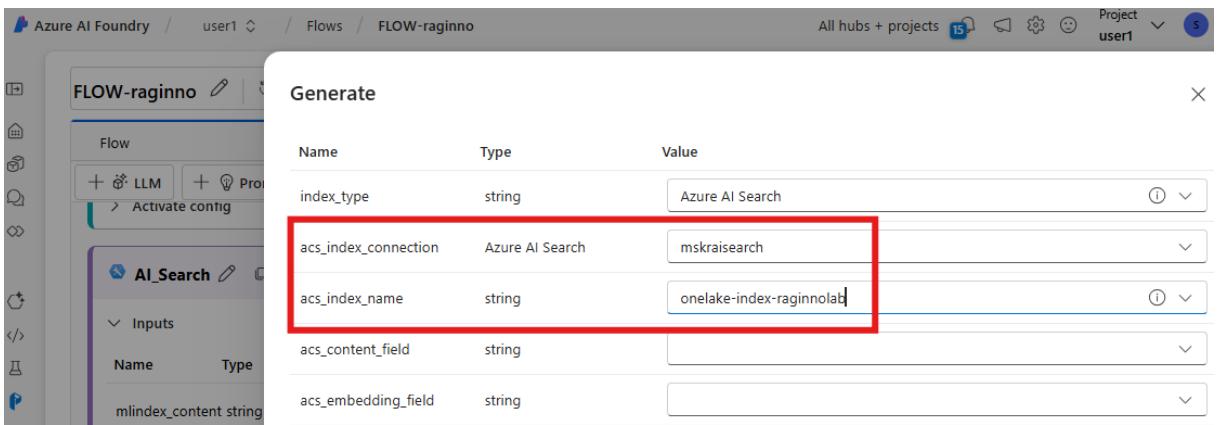
The 'Save' and 'Cancel' buttons are visible at the bottom right of the dialog.

B. Index Type으로 Azure AI Search를 선택합니다.



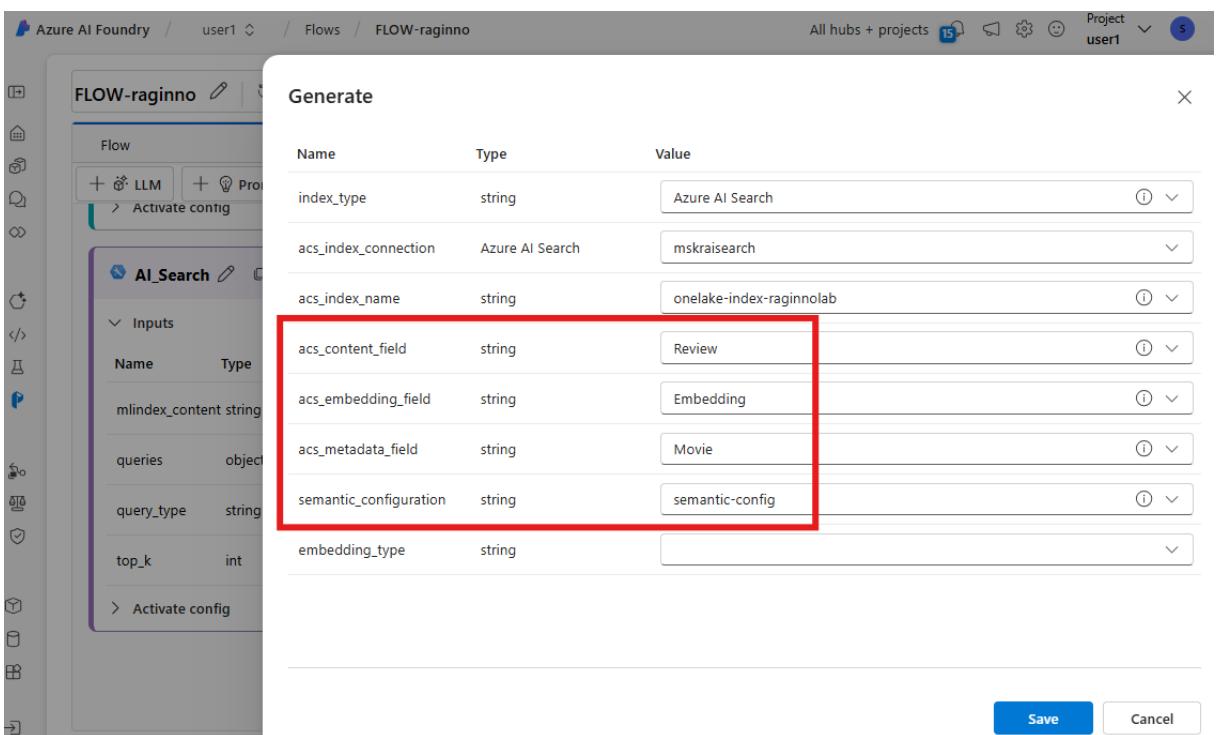
The screenshot shows the 'Generate' dialog in the Azure AI Foundry interface. The 'index_type' field is selected, and a dropdown menu is open, showing 'Azure AI Search' as the chosen option. Other options like 'FAISS' and 'MLIndex file from path' are also visible.

C. 배정받은 AI Search 리소스와 앞서 생성한 Index를 선택합니다.



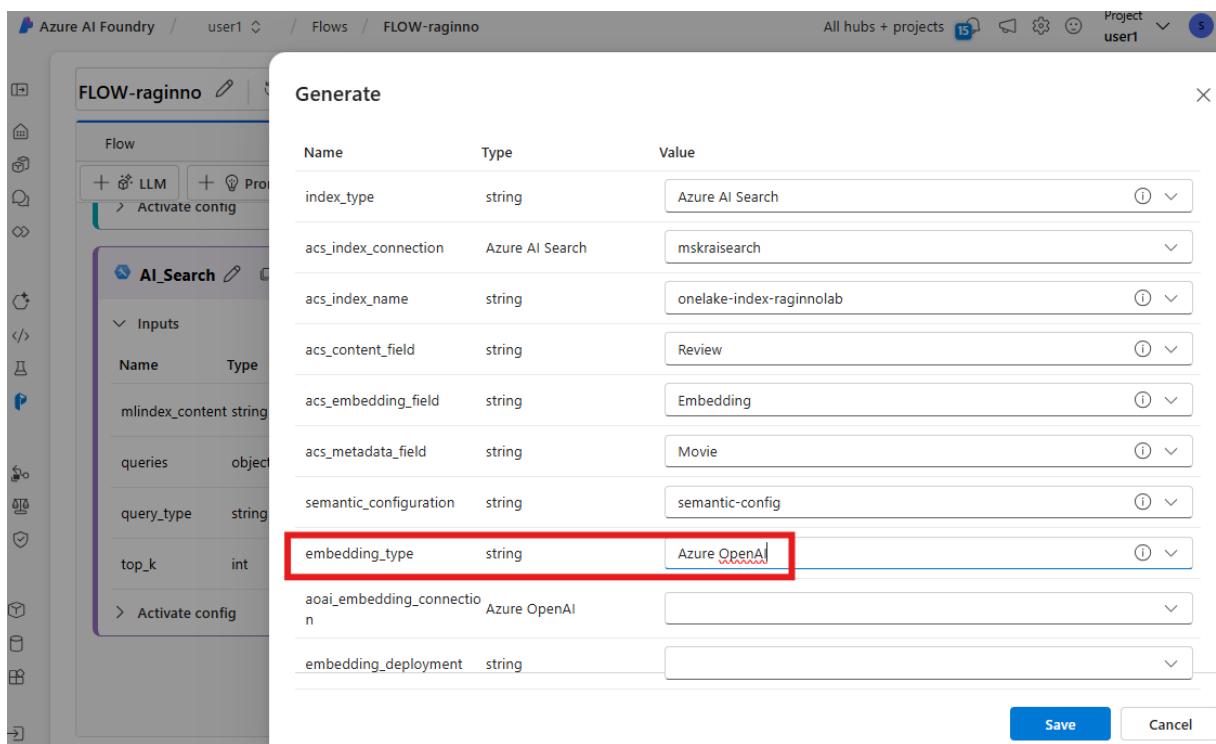
The screenshot shows the 'Generate' dialog with the 'acs_index_connection' and 'acs_index_name' fields selected. Both fields have dropdown menus open, showing 'mskraisearch' and 'onelake-index-raginnolab' as the chosen values respectively. Other fields like 'acs_content_field' and 'acs_embedding_field' are also present but not selected.

D. 필드 정보를 선택합니다.

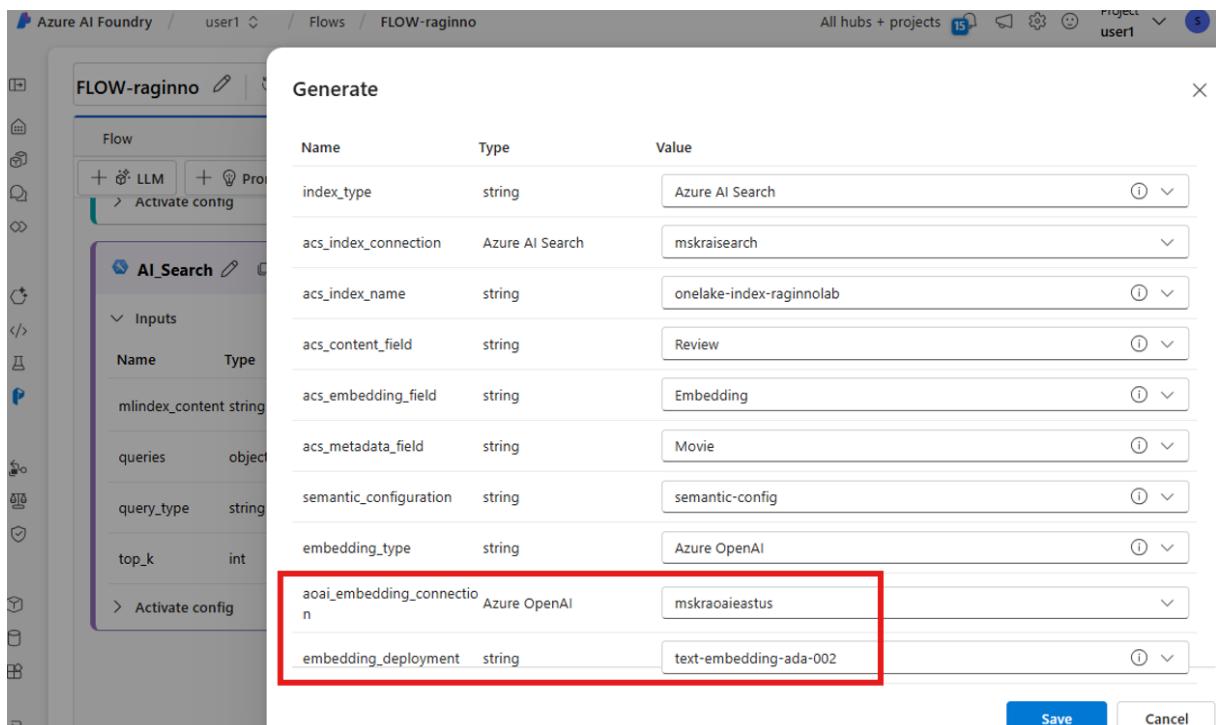


The screenshot shows the 'Generate' dialog with the 'acs_content_field', 'acs_embedding_field', 'acs_metadata_field', and 'semantic_configuration' fields selected. Each of these fields has a dropdown menu open, showing the chosen values: 'Review', 'Embedding', 'Movie', and 'semantic-config' respectively. The 'embedding_type' field is also present but not selected.

E. 쿼리를 임베딩해 Vector Search할 수 있도록 Embedding Type으로 Azure OpenAI를 선택합니다.



F. 배정받은 Azure OpenAI 리소스를 연결합니다.



G. 생성된 Index Lookup Node의 Queries와 Query Type을 지정합니다.

Flow configuration for AI_Search node:

- Inputs:**
 - mlindex_content: string (embeddings: api_base: http)
 - queries: object (Value: \${inputs.question})
 - query_type: string (Value: Hybrid + semantic)
 - top_k: int (Value: 3)

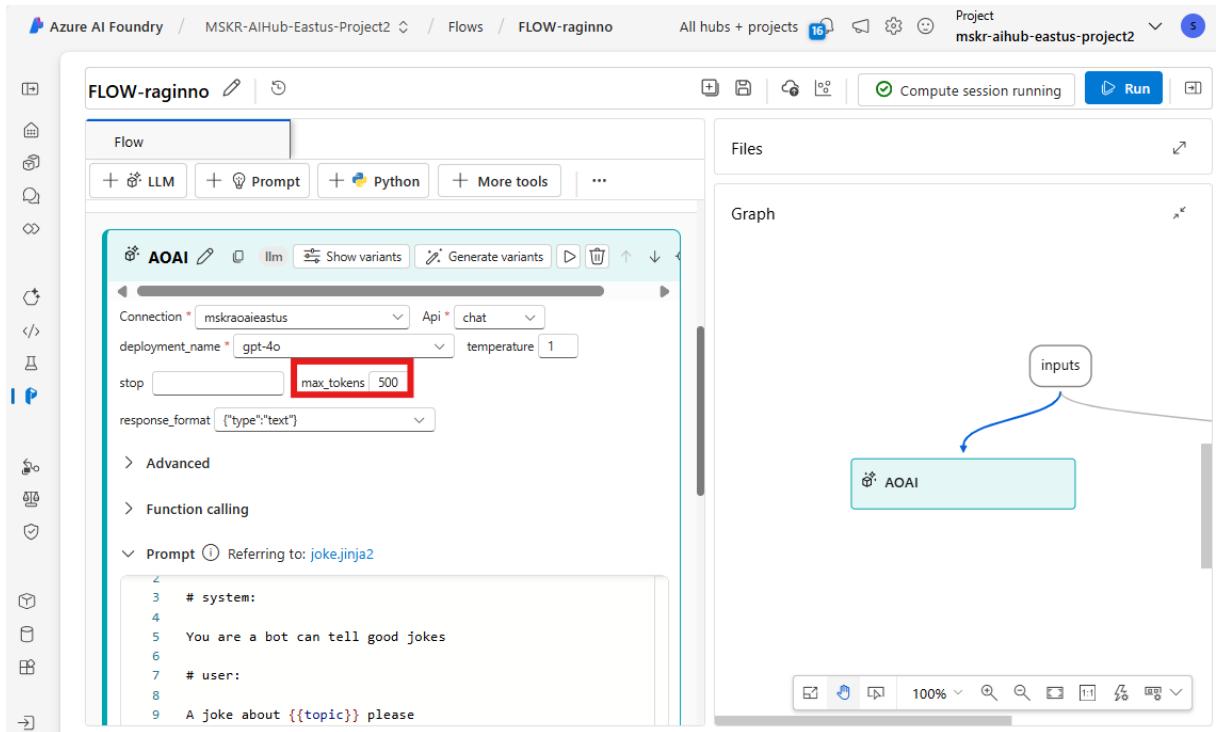
4. LLM Node에 배정받은 Azure OpenAI 리소스를 연결합니다.

Flow configuration for AOAI node:

- Inputs:**
 - Connection: mskraoaieastus
 - deployment_name: gpt-4o

5. LLM Node를 설정합니다.

A. Max Tokens를 수정합니다.



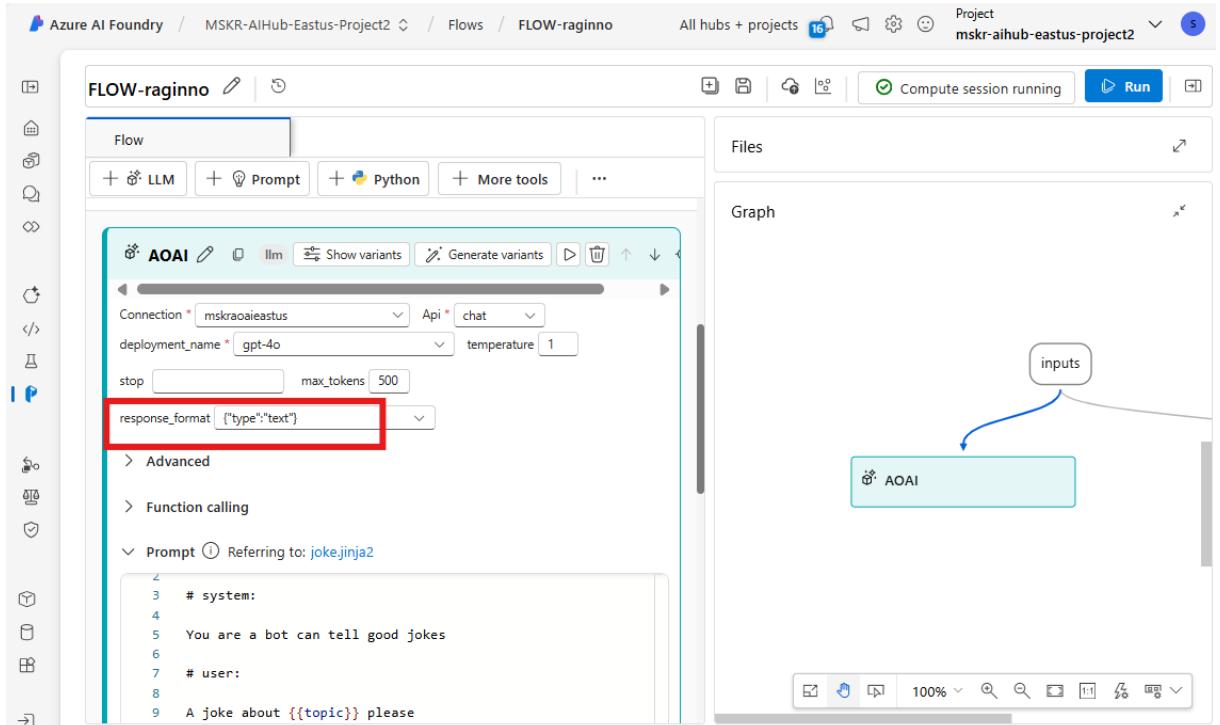
The screenshot shows the Azure AI Foundry interface with the project 'mskr-aihub-eastus-project2'. In the 'Graph' panel, there is a node labeled 'AOAI'. In the 'Flow' panel, the 'AOAI' node is selected. The configuration for the 'AOAI' node is as follows:

- Connection: mskraoaeastus
- Deployment name: gpt-4o
- Temperature: 1
- max_tokens: 500 (highlighted with a red box)
- response_format: [{"type": "text"}]

The 'Prompt' section contains a Jinja2 template:

```
3 # system:  
4  
5 You are a bot can tell good jokes  
6  
7 # user:  
8  
9 A joke about {{topic}} please
```

B. Response Type으로 Text를 선택합니다.



The screenshot shows the Azure AI Foundry interface with the project 'mskr-aihub-eastus-project2'. In the 'Graph' panel, there is a node labeled 'AOAI'. In the 'Flow' panel, the 'AOAI' node is selected. The configuration for the 'AOAI' node is as follows:

- Connection: mskraoaeastus
- Deployment name: gpt-4o
- Temperature: 1
- max_tokens: 500
- response_format: [{"type": "text"}] (highlighted with a red box)

The 'Prompt' section contains a Jinja2 template:

```
3 # system:  
4  
5 You are a bot can tell good jokes  
6  
7 # user:  
8  
9 A joke about {{topic}} please
```

C. 클론된 폴더의 AIFoundry\promptflow_sample.ipynb에 있는 RAG 프롬프트 샘플을 LLM 프롬프트로 사용합니다.

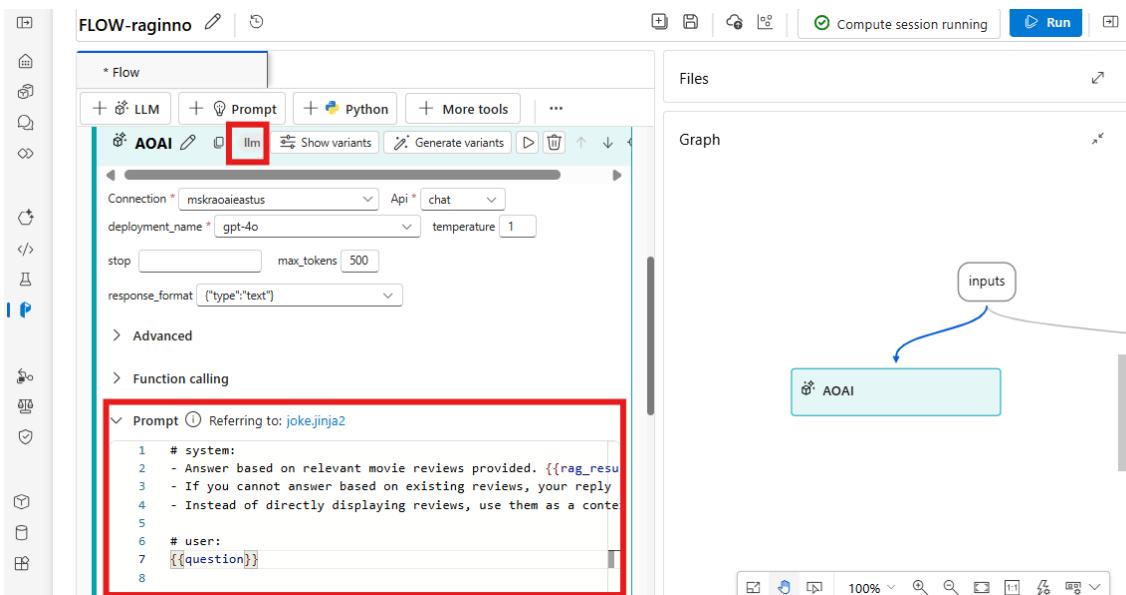
AIFoundry > promptflow_sample.ipynb > ...

Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...

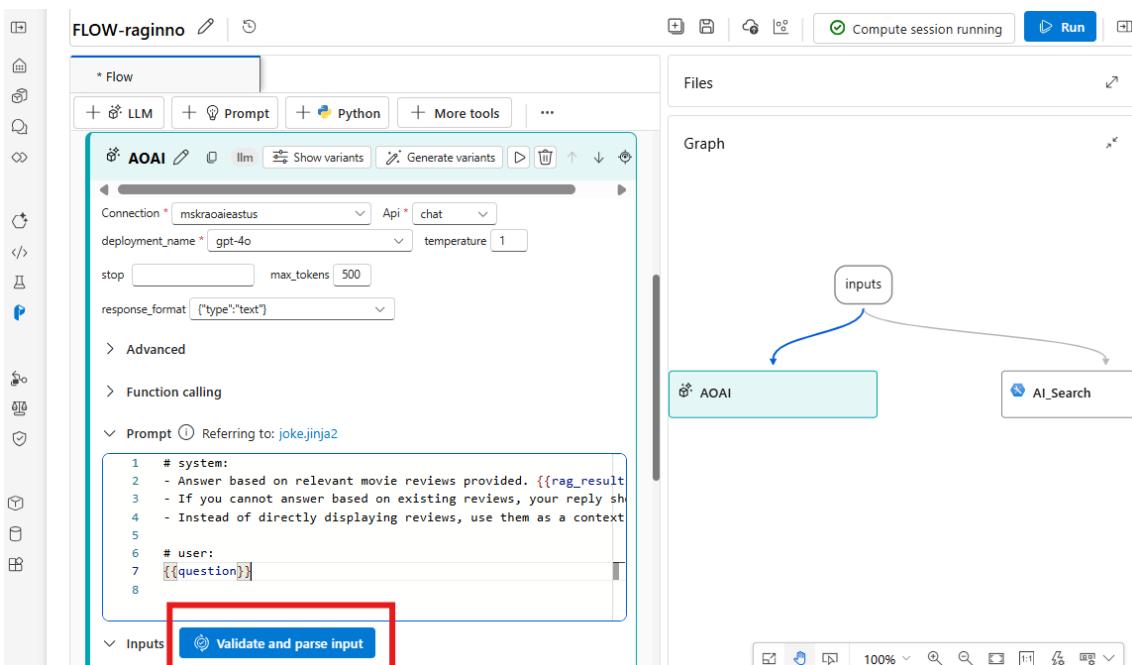
Prompt Guide for PromptFlow

RAG with GPT-4o

```
# system:  
- Answer based on relevant movie reviews provided. {{rag_results}}  
- If you cannot answer based on existing reviews, your reply should start with a disclosure that says so.  
- Instead of directly displaying reviews, use them as a context to create a natural response.  
  
# user:  
{{question}}
```



D. Validate and Parse Input을 선택합니다.



E. Validate가 완료된 후 Input을 적절한 값으로 연결합니다.

The screenshot shows the Azure AI Foundry interface for creating a flow. The main area is titled 'FLOW-raginno'. In the 'Inputs' section, there are two entries: 'question' (string type, value: \${inputs.question}) and 'rag_results' (string type, value: \${AI_Search.output}). The 'Graph' panel to the right shows a flowchart: 'inputs' connects to 'AI_Search', and 'AI_Search' connects to 'AOAI'.

6. Flow의 Output을 수정합니다.

A. LLM Node의 Output을 RAG Response에 연결합니다.

The screenshot shows the Azure AI Foundry interface for creating a flow. The main area is titled 'FLOW-raginno'. In the 'Outputs' section, there are two entries: 'rag_response' (value: \${echo.output}) and 'AOAI' (value: \${AOAI.output}). The 'Graph' panel to the right shows a flowchart: 'AI_Search' connects to 'AOAI'.

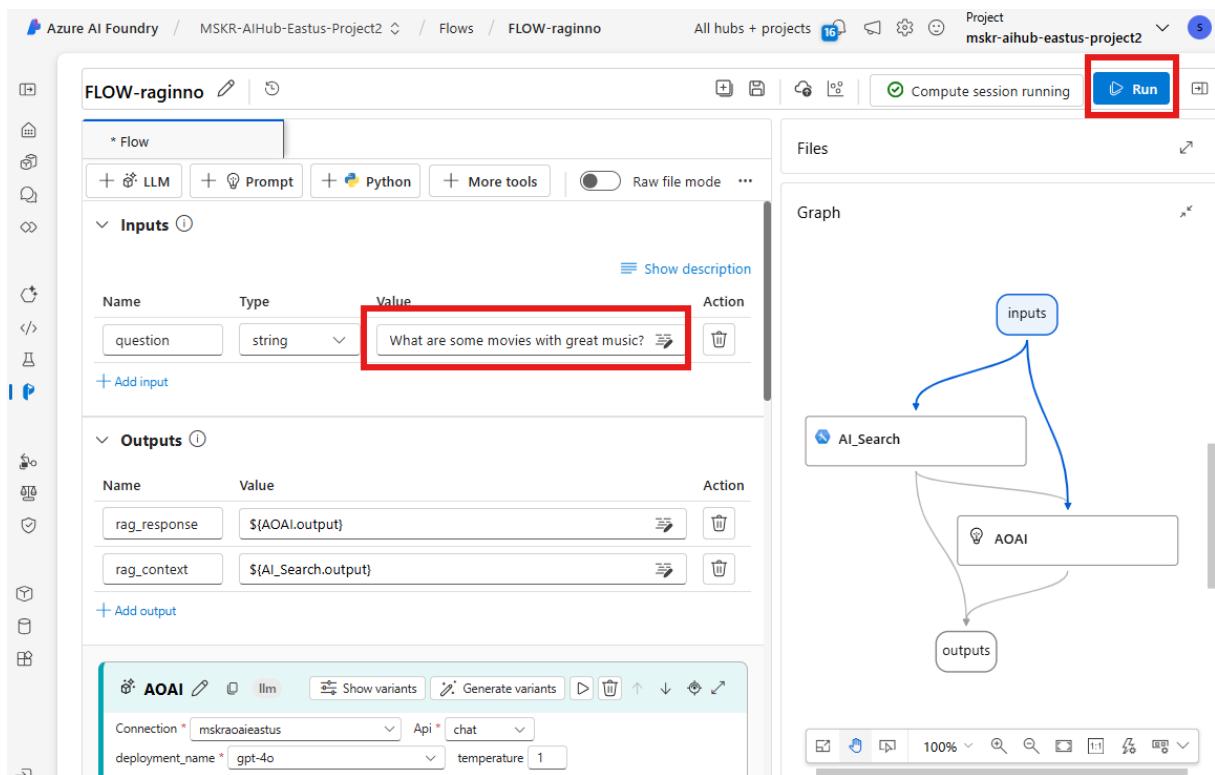
B. Add Output을 선택해 RAG Context를 추가하고, Index Lookup Node의 Output을 연결합니다.

The screenshot shows the Azure AI Foundry interface with the 'FLOW-raginno' flow selected. In the 'Outputs' section, the 'rag_response' output is defined with the value \${AOAI.output}. The 'Add output' button is highlighted with a red box. The Graph panel on the right shows a flow from 'AI_Search' to 'AOAI', then to 'outputs'.

The screenshot shows the Azure AI Foundry interface with the 'FLOW-raginno' flow selected. In the 'Outputs' section, the 'rag_context' output is defined with the value \${AI_Search.output}. The 'rag_context' row is highlighted with a red box. The Graph panel on the right shows a flow from 'AI_Search' to 'AOAI', then to 'outputs'.

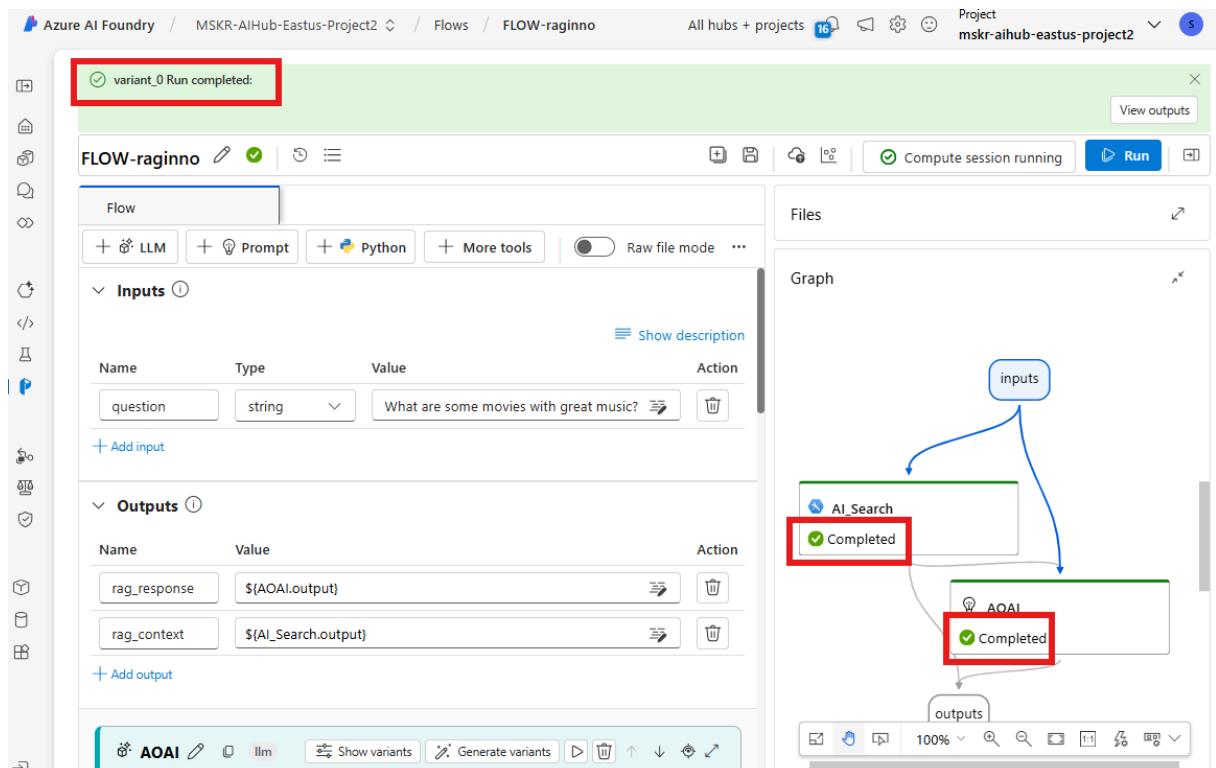
7. Flow 설정이 완료됐으면, Input 질문을 작성하고 Flow를 실행합니다.

- A. Clone한 폴더의 AIFoundry\promptflow_sample.ipynb에서 Input으로 사용 가능한 Sample Questions를 확인할 수 있습니다.



The screenshot shows the Azure AI Foundry interface for creating a flow. The flow is named 'FLOW-raginno'. In the 'Inputs' section, there is a single input named 'question' of type 'string' with the value 'What are some movies with great music?'. In the 'Outputs' section, there are two outputs: 'rag_response' set to '\${AOAI.output}' and 'rag_context' set to '\${AI_Search.output}'. The 'Graph' panel on the right shows a flowchart: 'inputs' connects to 'AI_Search', which then connects to 'AOAI', which finally connects to 'outputs'. The 'Run' button in the top right corner is highlighted with a red box.

8. Flow 실행 결과를 확인합니다.



The screenshot shows the Azure AI Foundry interface after the flow has been run. The top bar indicates 'variant_0 Run completed:'. The 'Graph' panel on the right shows the flowchart with the 'AI_Search' and 'AOAI' boxes both marked with a green checkmark and the text 'Completed', indicating the steps have been successfully executed. The rest of the interface is similar to the previous screenshot, showing the flow configuration and the 'Run' button.

The screenshot shows the Azure AI Foundry interface with the 'Outputs' tab selected. It displays two results:

- rag_response**: A completed response to the question "What are some movies with great music?". The response discusses movies like "A Star Is Born" and Lady Gaga's performance.
- rag_context**: A completed response containing JSON data. It includes metadata for "A STAR IS BORN", additional fields, and a score of 0.02687074989080429.

Call Deployed Endpoint

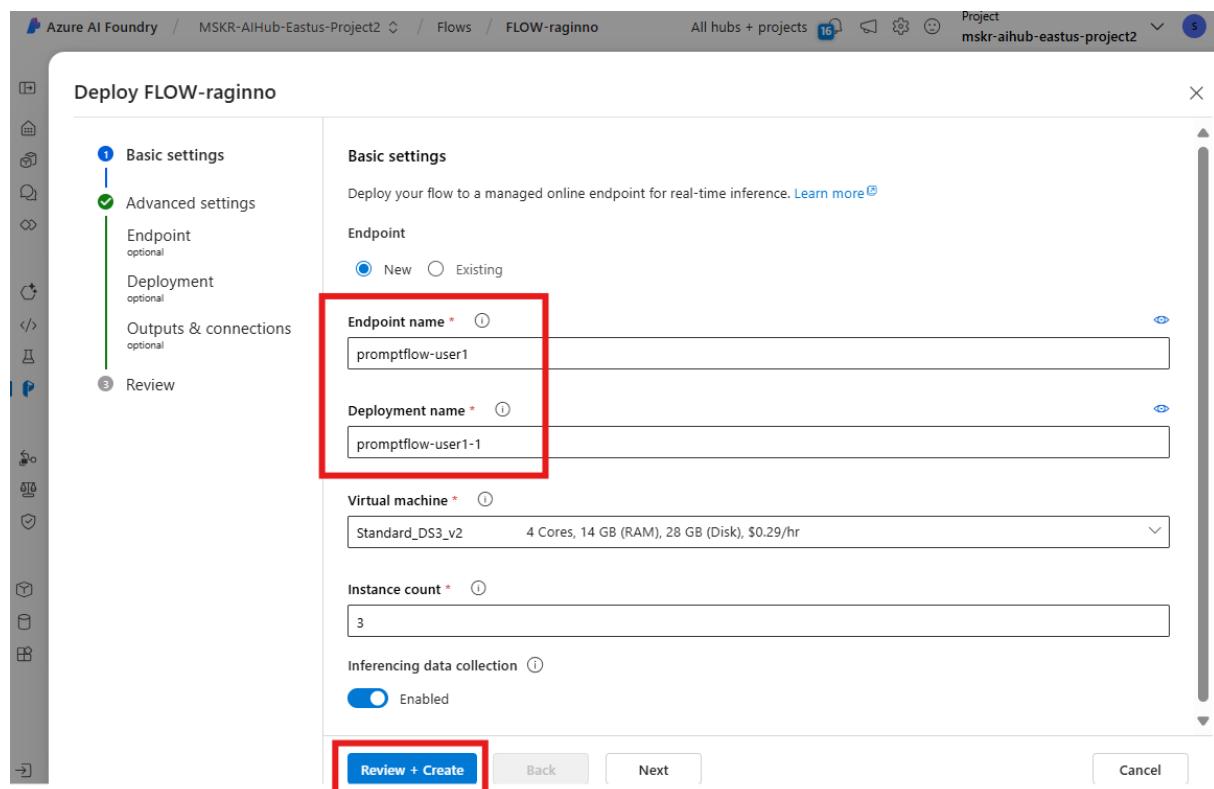
1. Flow 화면에서 Deploy를 선택합니다.

The screenshot shows the Azure AI Foundry Flow interface for the 'FLOW-raginno' flow. The top bar indicates a completed run ('variant_0 Run completed'). The main area shows the flow configuration:

- Inputs**: A single input 'question' of type string with the value "What are some movies with great music?".
- Outputs**: Two outputs: 'rag_response' and 'rag_context', both using the expression `\${AOAI.output}`.
- Graph**: A visual representation of the data flow. It starts with an 'inputs' node, which points to an 'AI_Search' node (status: Completed). The 'AI_Search' node then points to an 'AOAI' node (status: Completed), which finally points to the 'outputs' node.

The 'Compute session running' status is shown in the top right.

2. Custom Suffix를 포함한 이름으로 Endpoint와 Deployment명을 정하고 Create합니다.



Deploy FLOW-raginno

Basic settings

Endpoint name * promptflow-user1

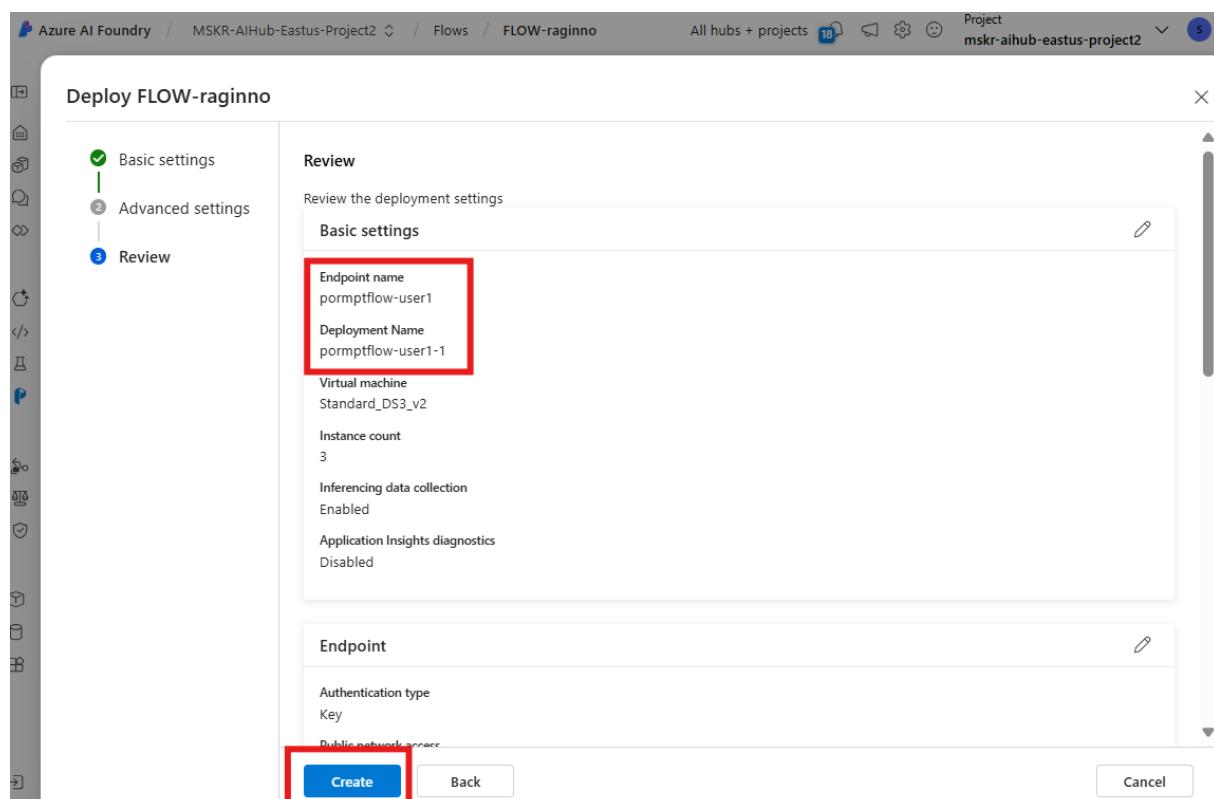
Deployment name * promptflow-user1-1

Virtual machine * Standard_DS3_v2

Instance count * 3

Inferencing data collection Enabled

Review + Create Back Next Cancel



Deploy FLOW-raginno

Review

Basic settings

Endpoint name promptflow-user1

Deployment Name promptflow-user1-1

Virtual machine Standard_DS3_v2

Instance count 3

Inferencing data collection Enabled

Application Insights diagnostics Disabled

Endpoint

Authentication type Key

Create Back Cancel

3. 배포가 완료되면 Endpoint와 Key 값을 확인합니다.

The screenshot shows the Microsoft AI Foundry interface. On the left, a sidebar lists various features: Model catalog, Playgrounds, AI Services, Build and customize, Agents, Templates, Fine-tuning, Prompt flow (selected), Assess and improve, Tracing, Evaluation, Safety + security, My assets (Models + endpoints selected), Data + indexes, Web apps, and Management center. The main workspace displays a flow named "FLOW-raginno" with an "Inputs" section containing a "question" input set to "string" with the value "What are some movies wi...". The "Outputs" section contains "rag_response" and "rag_context" outputs, both set to "\${AOAI.output}". Below the flow is a "AOAI" section with buttons for "Get endpoint", "Show variants", "Generate variants", and "Regenerate". To the right is a "Graph" view showing the flow's structure: "inputs" (blue box) connects to "AI_Search" (green box, status: Completed), which then connects to "AOAI" (green box, status: Completed), which finally outputs the results (blue box). The bottom of the graph view has a toolbar with icons for "Get endpoint", "Update traffic", "Delete deployment", "Regenerate", and "Swagger URL".

The screenshot shows the "Manage deployments" page in Microsoft AI Foundry. The left sidebar includes icons for Home, Models + endpoints (selected), Service endpoints, and a search bar. The main area displays a table of model deployments:

Name	Model name	Model version	State
mskr-agent-svc-eastus_aoai			
mskr-aoai-eastus2			
mskraoiaeastus			
pf-raginnolab			
promptflow-user1			
promptflow-user1-1	promptflow-user1	promptflow-user1	Succeeded

The row for "promptflow-user1-1" is highlighted with a red box around the name. The "State" column for this row also has a red box around the "Succeeded" status.

The screenshot shows the deployment details for "promptflow-user1-1". The left sidebar includes icons for Home, Models + endpoints, Service endpoints, and a search bar. The main area has tabs for Details (selected), Test, Consume, Monitoring, and Logs. The "Deployment info" section shows:

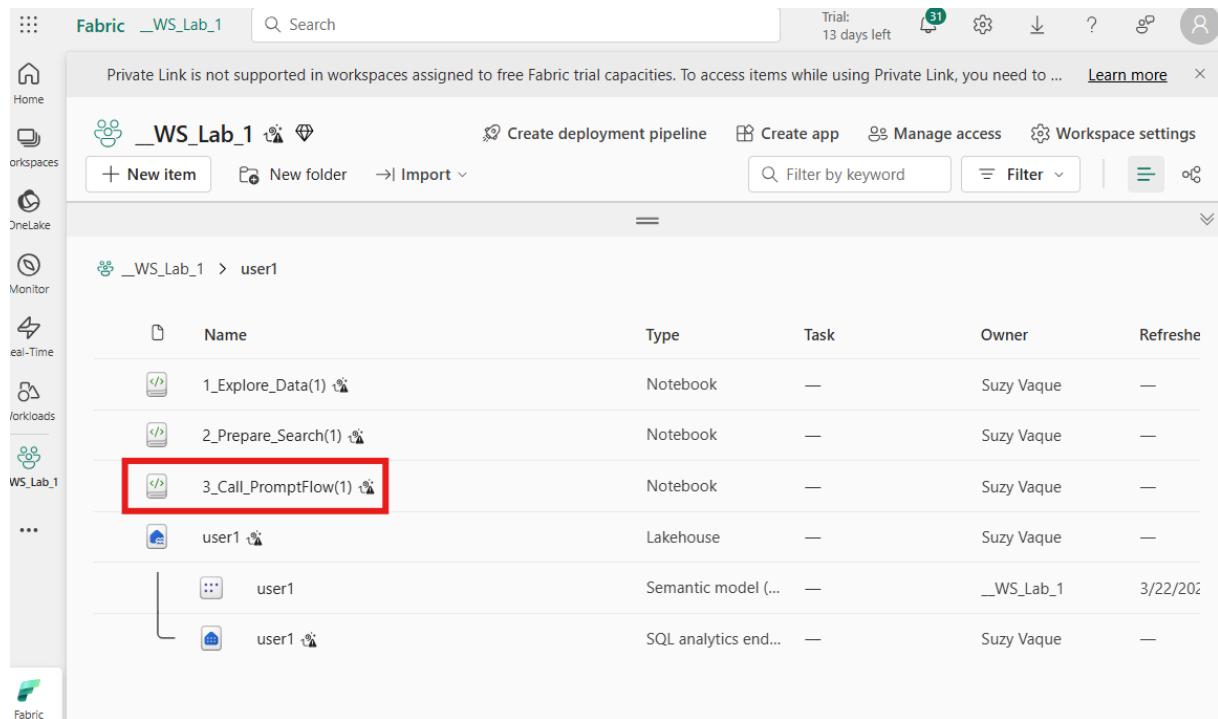
- Name: promptflow-user1-1
- Provisioning state: Succeeded
- Created by: Suzy Vaque
- Traffic allocation: 0%
- Compute type: Dedicated

The "Endpoint" section shows:

- Target URI: <https://promptflow-user1.eastus.inference.ml.azure.com/score>
- Authentication type: Key
- Primary key: (redacted)
- Swagger URL: <https://promptflow-user1.eastus.inference.ml.azure.com/swagger.js...>

Red boxes highlight the "Target URI" and "Primary key" fields.

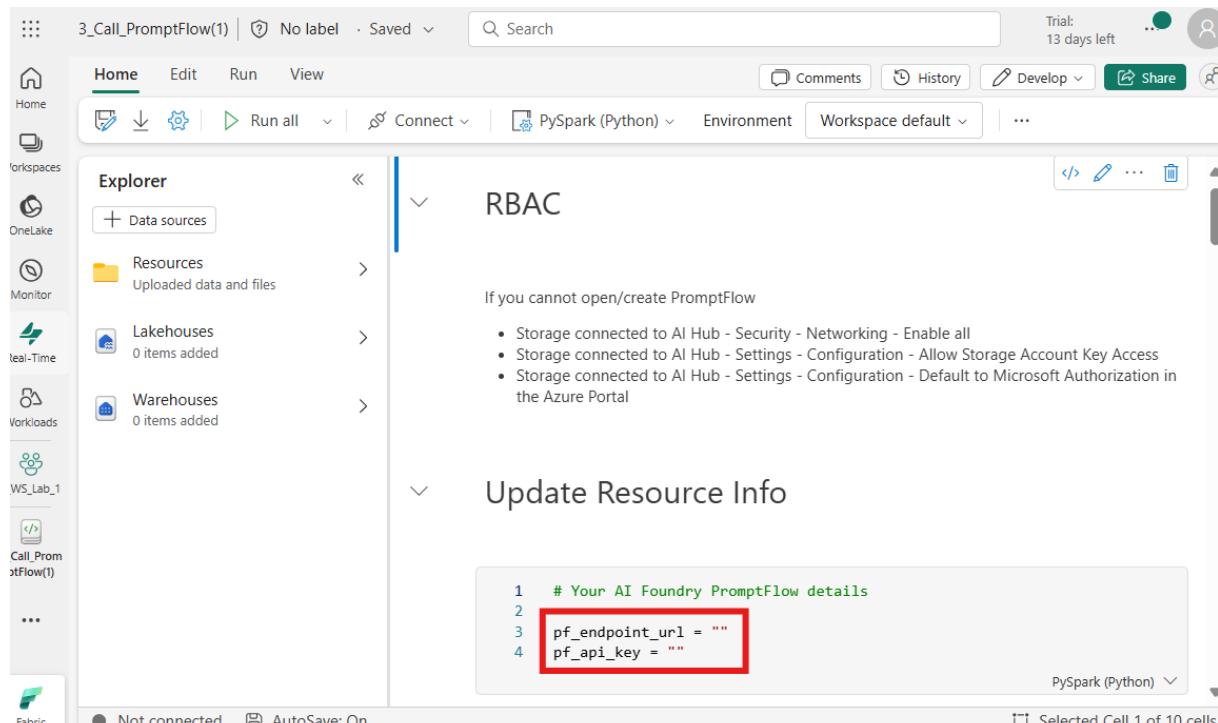
4. Fabric Portal의 3_Call_PromptFlow 노트북으로 이동합니다.



The screenshot shows the Fabric Portal interface. The left sidebar includes icons for Home, Workspaces, OneLake, Monitor, Real-Time, Workloads, and WS_Lab_1. The main area displays a list of items under '_WS_Lab_1 > user1'. The list includes:

Name	Type	Task	Owner	Refresh
1_Explore_Data(1)	Notebook	—	Suzy Vaque	—
2_Prepares_Search(1)	Notebook	—	Suzy Vaque	—
3_Call_PromptFlow(1)	Notebook	—	Suzy Vaque	—
user1	Lakehouse	—	Suzy Vaque	—
user1	Semantic model (...	—	_WS_Lab_1	3/22/2024
user1	SQL analytics end...	—	Suzy Vaque	—

5. Flow의 Endpoint와 Key 값을 업데이트합니다.



The screenshot shows the content of the '3_Call_PromptFlow(1)' notebook. The left sidebar shows the notebook is connected to 'PySpark (Python)'. The main area has tabs for Home, Edit, Run, View, Comments, History, Develop, and Share. The notebook content includes:

RBAC

If you cannot open/create PromptFlow

- Storage connected to AI Hub - Security - Networking - Enable all
- Storage connected to AI Hub - Settings - Configuration - Allow Storage Account Key Access
- Storage connected to AI Hub - Settings - Configuration - Default to Microsoft Authorization in the Azure Portal

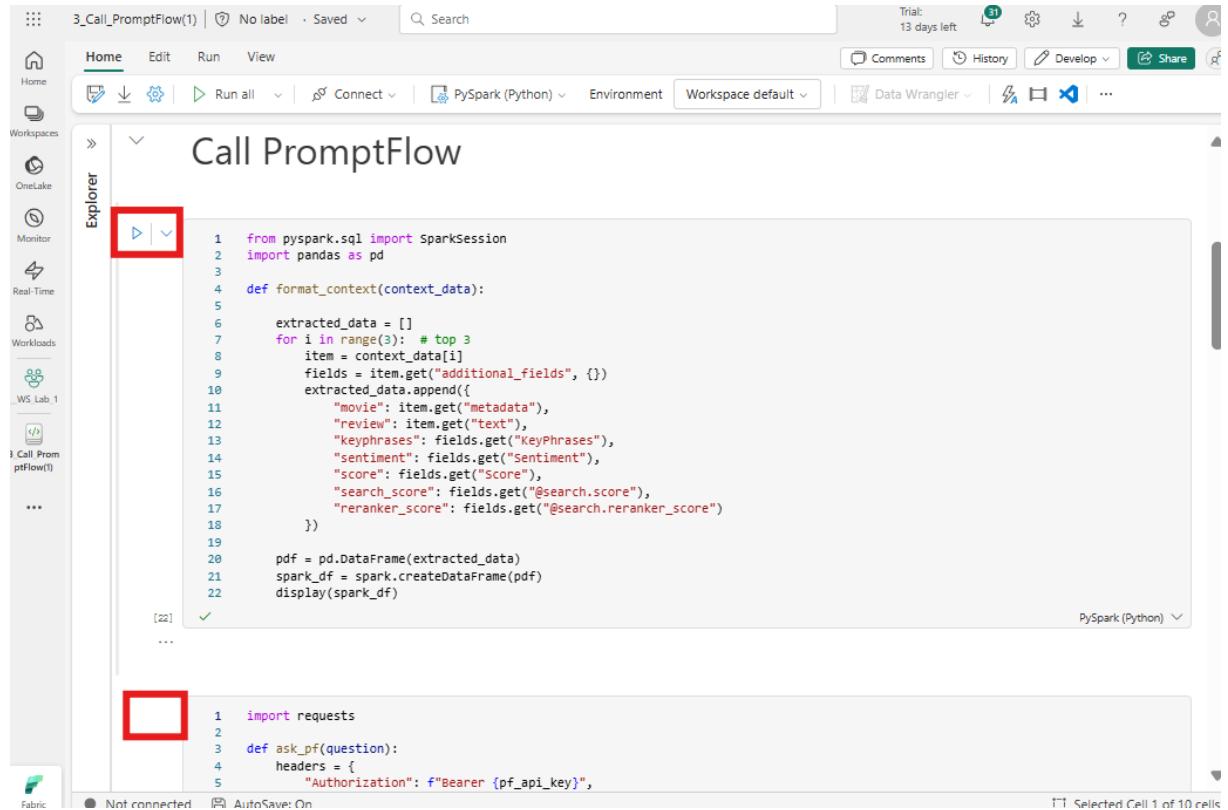
Update Resource Info

```
1  # Your AI Foundry PromptFlow details
2
3  pf_endpoint_url = ""
4  pf_api_key = ""
```

PySpark (Python) Selected Cell 1 of 10 cells

6. Run PromptFlow 항목의 셀들을 실행하면, Fabric Notebook 상에서 REST API 방식으로 Flow를 호출하고

Lakehouse의 데이터를 기반으로 한 RAG 답변을 얻을 수 있습니다.



```
3_Call_PromptFlow(1) | No label · Saved · Search
```

Home Edit Run View

Comments History Develop Share

Call PromptFlow

```
1  from pyspark.sql import SparkSession
2  import pandas as pd
3
4  def format_context(context_data):
5
6      extracted_data = []
7      for i in range(3): # top 3
8          item = context_data[i]
9          fields = item.get("additional_fields", {})
10         extracted_data.append({
11             "movie": item.get("metadata"),
12             "review": item.get("text"),
13             "keyphrases": fields.get("keyPhrases"),
14             "sentiment": fields.get("Sentiment"),
15             "score": fields.get("score"),
16             "search_score": fields.get("@search.score"),
17             "reranker_score": fields.get("@search.reranker_score")
18         })
19
20     pdf = pd.DataFrame(extracted_data)
21     spark_df = spark.createDataFrame(pdf)
22     display(spark_df)
```

```
1  import requests
2
3  def ask_pf(question):
4      headers = {
5          "Authorization": f"Bearer {pf_api_key}",
```

PySpark (Python)

Not connected AutoSave: On

Selected Cell 1 of 10 cells

```
1  response = ask_pf("What are some movies with great music?")
```

From the recent reviews, "A Star Is Born" stands out as a film with remarkable musical numbers. It combines contemporary relevance with old-fashioned movie entertainment and includes an appealing love story between its leads. Lady Gaga's performance in particular is noted for its potential to cement her status as a movie star, largely driven by the impactful musical scenes.

Additionally, "Once Upon a Time... in Hollywood" doesn't focus on musical numbers but is described as exquisitely crafted, which often includes a thoughtfully curated soundtrack typical of Quentin Tarantino's films.

If you're open to a more casual, entertaining experience, "Jurassic World: Fallen Kingdom" is noted for its good mix of entertaining moments, though it isn't particularly highlighted for its music.

For genuinely memorable music in film, "A Star Is Born" is definitely worth checking out based on the current reviews.

[AI Search Context]

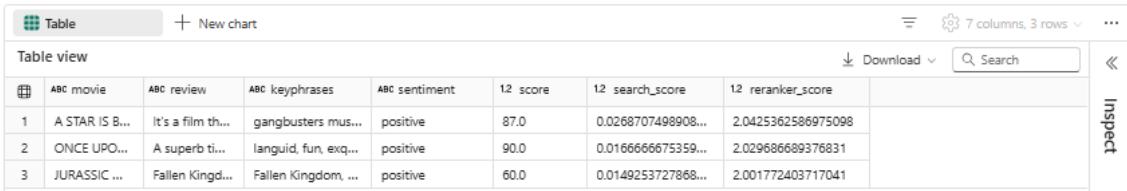


Table view

Download Search

	ABC movie	ABC review	ABC keyphrases	ABC sentiment	12 score	12 search_score	12 reranker_score
1	A STAR IS B...	It's a film th...	gangbusters mus...	positive	87.0	0.0268707498908...	2.0425362586975098
2	ONCE UPO...	A superb ti...	languid, fun, exq...	positive	90.0	0.01666666675359...	2.029686689376831
3	JURASSIC ...	Fallen Kingd...	Fallen Kingdom, ...	positive	60.0	0.0149253727868...	2.001772403717041