

Lab 02 - Performing Sentiment analysis and Text translation with AI Skills in Microsoft Fabric

② Introduction

Azure AI services help developers and organizations rapidly create intelligent, cutting-edge, market-ready, and responsible applications with out-of-the-box and pre-built and customizable APIs and models. In this article, you'll use the various services available in Azure AI services to perform tasks that include: text analytics, translation, document intelligence, vision, image search, speech to text and text to speech conversion, anomaly detection, and data extraction from web APIs.

The goal of Azure AI services is to help developers create applications that can see, hear, speak, understand, and even begin to reason. The catalog of services within Azure AI services can be categorized into five main pillars: Vision, Speech, Language, Web search, and Decision.

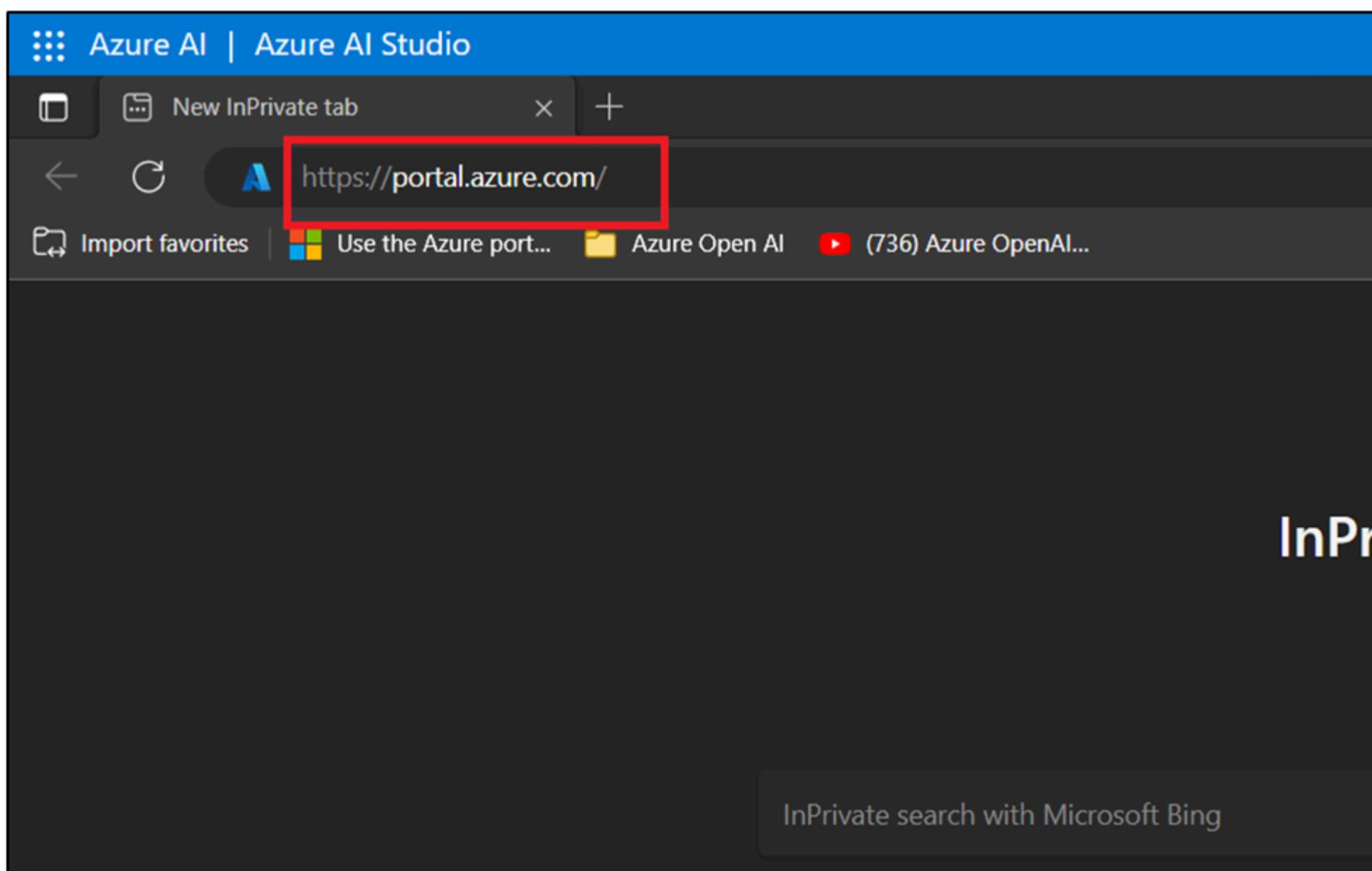
Objective:

- Create a multi-service resource for Azure AI services using the Azure portal.
- Create a workspace and lakehouse in Microsoft Fabric with the Fabric trial enabled.
- Generate a notebook for interactive data exploration.
- Import required libraries and initialize your Spark session
- Utilize the Text Analytics service to analyze the sentiment of input text.
- Utilize the Text Analytics for Health Service to extract and label relevant medical information from unstructured text.
- Utilize the Azure Translator, a cloud-based machine translation service, to translate text from one language to another.
- Extract information from a document into structured data
- Image analysis and tagging using Azure Computer Vision.
- Search for images that are related to a natural language query
- working on a speech-to-text transcription task using Spark and the Speech toText SDK.
- Develop a text-to-speech service that enables applications to convert written text into natural-sounding speech.
- Uses the Anomaly Detector service to find anomalies in entire time series data
- Get information from arbitrary web APIs

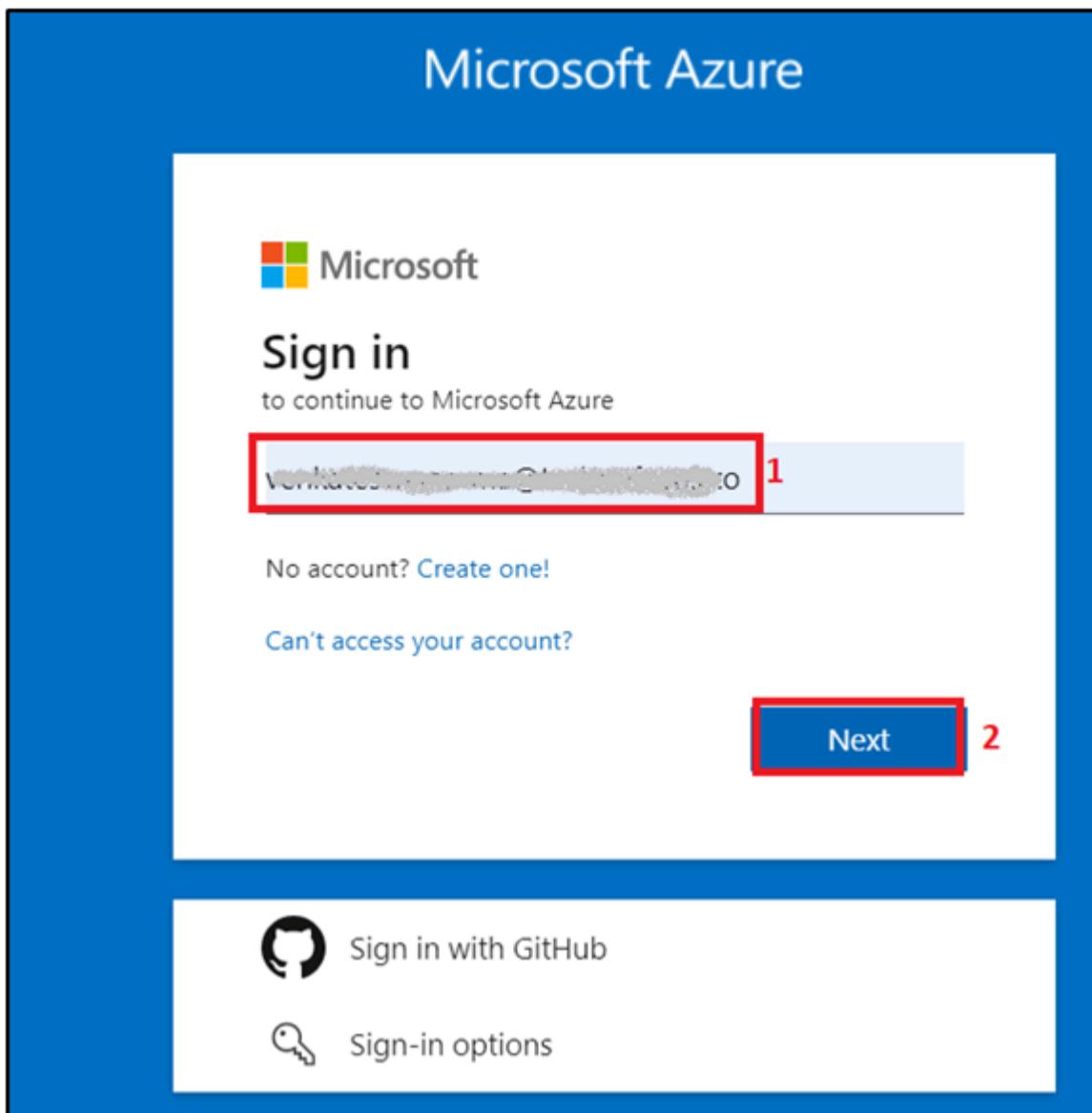
Exercise 1: Setup Lakehouse and create Azure AI service

Task 1: Assign Cognitive Services Contributor roles using the Azure portal

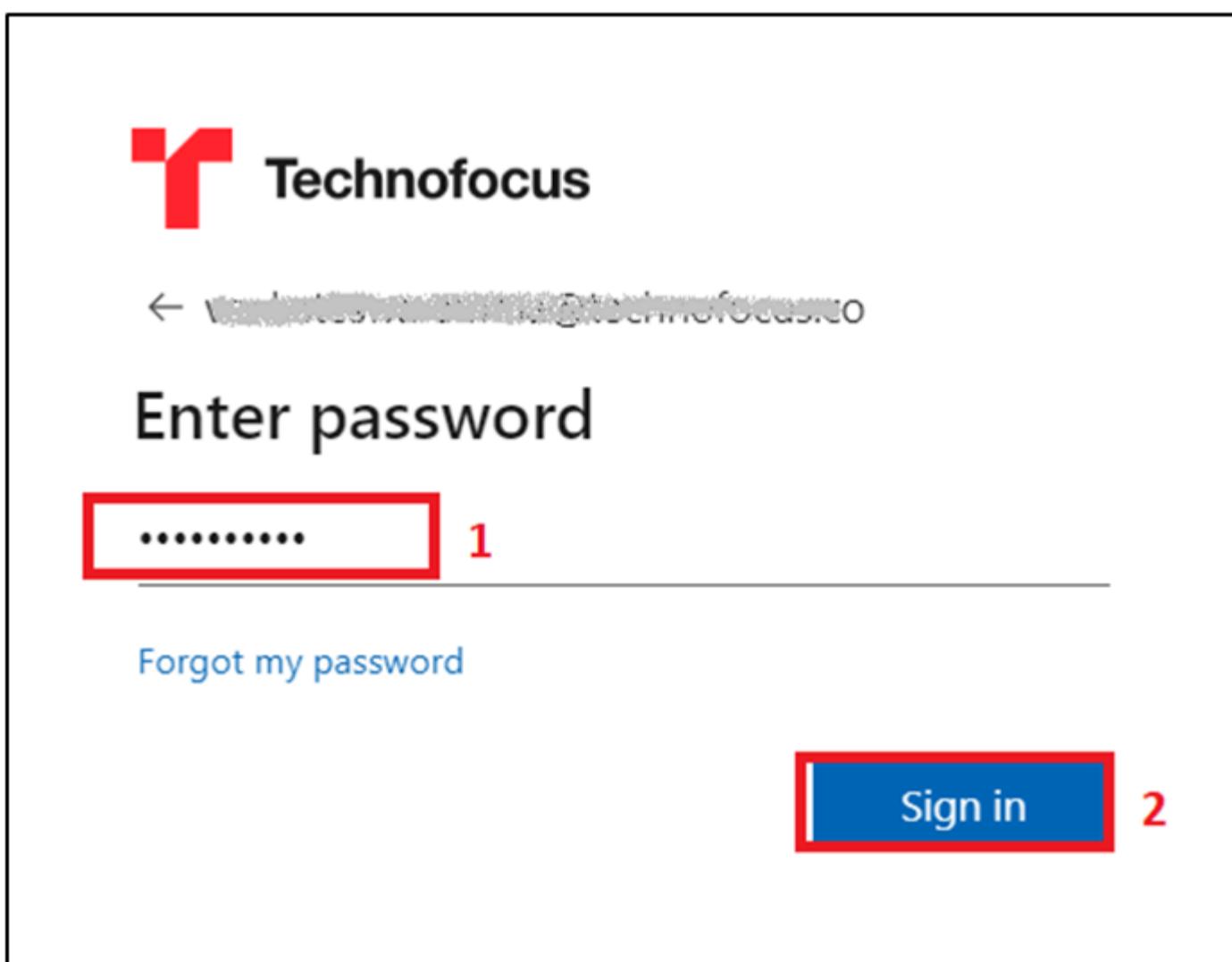
1. Open your browser, navigate to the address bar, and type or paste the following URL: <https://portal.azure.com/>, then press the **Enter** button.



2. In the **Microsoft Azure** window, enter your **Sign-in** credentials, and click on the **Next** button.



3. Then, enter the password and click on the **Sign in** button**.**



4. In **Stay signed in?** window, click on the **Yes** button.



stay signed in? [Technofocus.co](#)

Stay signed in?

Do this to reduce the number of times you are asked to sign in.

Don't show this again

No

Yes

5. Type in **Subscriptions** in the search bar and select **Subscriptions**.

A screenshot of the Microsoft Azure portal. At the top, there's a search bar with the placeholder text 'Subscriptions'. Below the search bar, a navigation bar has several tabs: 'All' (highlighted in blue), 'Services (8)', 'Marketplace (5)', 'Documentation (99+)', 'Resources (0)', and 'Resource Groups (0)'. On the left side, there's a sidebar with sections for 'Services' (containing 'Subscriptions' which is also highlighted with a red box), 'Marketplace', and 'Documentation'. The main content area shows various subscription-related items like 'Billing subscriptions', 'Event Grid Subscriptions', 'Event Grid', 'Quotas', 'Management groups', 'Service Bus', 'Resource groups', and some marketplace items. At the bottom, there's a 'See all' link and a list of documentation articles.

6. Click on your assigned **subscription**.

Home >

Subscriptions

Contoso (M365x59005443.onmicrosoft.com)

+ Add Manage Policies View Requests View eligible subscriptions

Search for any field... Subscriptions == global filter My role == all Status == all + Add filter

Subscription name ↑↓	Subscription ID ↑↓	My role ↑↓	Current cost	Secure Score ↑↓
----------------------	--------------------	------------	--------------	-----------------

Azure Pass - Sponsorship	b41ebf.....2	Account admin	Not available	-
--------------------------	--------------	---------------	---------------	---

7. From the left menu, click on the **Access control(IAM)**.

The screenshot shows the Microsoft Azure Subscriptions page. At the top, there's a search bar and several filter options: 'Subscriptions == global filter', 'My role == all', 'Status == all', and '+ Add filter'. Below the filters, a table lists one subscription: 'Azure Pass - Sponsorship' with Subscription ID 'b41ebf.....2', My role 'Account admin', Current cost 'Not available', and Secure Score '-'. In the left sidebar, there are several sections: Overview, Activity log, Access control (IAM) (which is highlighted with a red box), Tags, Diagnose and solve problems, Security, Events, Cost Management, Cost analysis, Cost alerts, Budgets, Advisor recommendations, Billing, and Invoices. On the right side, there's a summary section with details like Subscription ID, Directory, My role, Offer, Offer ID, and Parent management group. Below that is a chart titled 'Top products by number of resources' showing counts for accounts, serverfarms, storageaccounts, searchservices, and databaseaccounts.

8. On the Access control(IAM) page, Click +Add and select **Add role assignments**.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Subscriptions > Azure Pass - Sponsorship

Azure Pass - Sponsorship | Access control (IAM)

Add role assignment

Add co-administrator

Add custom role

View my access

Check access

Grant access to this resource

View access to this resource

Add role assignment

View

The screenshot shows the Microsoft Azure Access control (IAM) interface for the 'Azure Pass - Sponsorship' subscription. On the left, there's a navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events, Cost Management, Cost analysis, Cost alerts, Budgets, Advisor recommendations, Billing, Invoices, and a 'View my access' button. The 'Access control (IAM)' option is selected. At the top right, there are buttons for 'Add', 'Download role assignments', 'Edit columns', 'Refresh', 'Remove', and a search bar. A prominent red box highlights the 'Add role assignment' button, with a red number '2' placed over it. Below this, there are two sections: 'Grant access to this resource' (with a 'Learn more' link) and 'View access to this resource' (with a 'Learn more' link). A blue 'Add role assignment' button is located at the bottom of the main content area.

9. Type the **Cognitive Services Contributor** in the search box and select it. Click **Next**

Microsoft Azure

Search resources, services, and docs (G+)

Home > Subscriptions > Azure Pass - Sponsorship | Access control (IAM) >

Add role assignment

Role * Members * Conditions Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Job function roles Privileged administrator roles

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

Cognitive Services Contributor 1 X Type : All Category : All

Name ↑	Description ↑↓
Cognitive Services Contributor 2	Lets you create, read, update, delete and manage keys of Cognitive Services.
Cognitive Services Custom Vision Contributor	Full access to the project, including the ability to view, create, edit, or delete projects.
Cognitive Services OpenAI Contributor	Full access including the ability to fine-tune, deploy and generate text
Cognitive Services Speech Contributor	Full access to Speech projects, including read, write and delete all entities, for real-time speech recognition

Showing 1 - 4 of 4 results.

Review + assign Previous Next 3

The screenshot shows the 'Add role assignment' page. At the top, there are tabs for 'Role' (which is selected and has a red asterisk), 'Members', 'Conditions', and 'Review + assign'. Below this, a note says 'A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles.' with a 'Learn more' link. There are two tabs: 'Job function roles' (selected) and 'Privileged administrator roles'. A note below says 'Grant access to Azure resources based on job function, such as the ability to create virtual machines.'. A search bar at the top right contains 'Cognitive Services Contributor' with a red box around it and a red number '1' next to it. To the right of the search bar are 'Type : All' and 'Category : All' buttons. Below the search bar is a table with two columns: 'Name' and 'Description'. The first row shows 'Cognitive Services Contributor' with a red box around it and a red number '2' next to it, followed by its description: 'Lets you create, read, update, delete and manage keys of Cognitive Services.'. Other rows show 'Cognitive Services Custom Vision Contributor' (description: 'Full access to the project, including the ability to view, create, edit, or delete projects.'), 'Cognitive Services OpenAI Contributor' (description: 'Full access including the ability to fine-tune, deploy and generate text'), and 'Cognitive Services Speech Contributor' (description: 'Full access to Speech projects, including read, write and delete all entities, for real-time speech recognition'). At the bottom, there are 'Review + assign', 'Previous', and 'Next' buttons, with the 'Next' button having a red box around it and a red number '3' next to it.

10. In the **Add role assignment** tab, select Assign access to User group or service principal. Under Members, click **+Select members**

The screenshot shows the 'Add role assignment' page in the Microsoft Azure portal. The 'Members' tab is selected, indicated by a blue underline. The 'Selected role' is set to 'Cognitive Services Contributor'. The 'Assign access to' option is set to 'User, group, or service principal' (radio button selected). The 'Members' section contains a button labeled '+ Select members' which is highlighted with a red box. Below this, there is a table with columns 'Name', 'Object ID', and 'Type', showing the message 'No members selected'. The 'Description' field is optional and empty. At the bottom, there are buttons for 'Review + assign', 'Previous', and 'Next'.

11. On the Select members tab , search your Azure OpenAI subscription and click **Select**.

Select members

X

Select ⓘ

aoaitfs006041@technofocus.live

No users, groups, or service principals found.

Selected members:

AT

AOAI TFS 006041

aoaitfs006041@technofocus.live

Remove

Select

Close

12. In the **Add role assignment** page, Click **Review + Assign**, you will get a notification once the role assignment is complete.



Add role assignment

[Role](#) [Members](#) [Conditions](#) [Review + assign](#)**Selected role** Cognitive Services Contributor**Assign access to**
 User, group, or service principal
 Managed identity**Members** [+ Select members](#)

Name	Object ID	Type
AOAI TFS 006041	43f6a3e7-bfa7-4ef3-9e08-8c503ae427e3	User

Description [Review + assign](#)[Previous](#)[Next](#)

Microsoft Azure

Search resources, services, and docs (G)

Home > Subscriptions > Azure Pass - Sponsorship | Access control (IAM) >

Add role assignment

Role Members Conditions **Review + assign**

Role Cognitive Services Contributor

Scope /subscriptions/0c08b20f-9ce4-4949-80d7-79381006f4ae

Members	Name	Object ID
	AOAI TFS 006041	43f6a3e7-bfa7-4ef3-9e08-8c503ae427e3

Description No description

Review + assign Previous Next

13. You will see a notification -- added as Cognitive Services Contributor for Azure Pass-Sponsorship.

Notifications

More events in the activity log → Dismiss all ▾

Added Role assignment ×

AOAI TFS 006041 was added as Cognitive Services Contributor for Azure Pass - Sponsorship.

a few seconds ago

Task 2: Create a multi-service resource for Azure AI services

The multi-service resource is listed under **Azure AI services > Azure AI services multi-service account** in the portal. To create a multi-service resource follow these instructions:

1. Select this link to create a multi-service resource: <https://portal.azure.com/#create/Microsoft.CognitiveServicesAllInOne>
2. On the **Create** page, provide the following information:

Project details	Description
Subscription	Select one of your available Azure subscriptions.
Resource group	Click on Create new > enter AI-FabricXX (XX can be a unique number, you can add more digits after XX to make the name unique)
Region	Select the appropriate region for your CognitiveServices**. **In this lab, we have chosen the East US 2 region.
Name	Cognitive-serviceXXX (XXX can be a unique number, you can add more digits after XXX to make the name unique)

Project details	Description
Pricing tier	Standard S0

3. Configure other settings for your resource as needed, read and accept the conditions (as applicable), and then select **Review + create**.

Microsoft Azure Search resources, services, and documentation

Home > Create Azure AI services

Basics Network Identity Tags Review + create

Get access to Vision, Language, Search, and Speech Azure AI services with a single API key. Quickly connect services together to achieve more insights into your content and easily integrate with other services like Azure Search.

Learn more

Project Details

Subscription * ① Azure Pass - Sponsorship **1**

Resource group * ① **Create new** **2**

A resource group is a container that holds related resources for an Azure solution.

Name * **AI-Fabric879** **3**

OK Cancel **4**

i Location specifies the region only for in regional services. Click here for more details.

Instance Details

Region ①

Name * ①

Pricing tier * ① **The value must not be empty.**

View full pricing details

Previous Next **Review + create**

MICROSOFT Azure

Search resources, services, and docs (Q)

Home >

Create Azure AI services

together to achieve more insights into your content and easily integrate with other services like Azure Search.

[Learn more](#)

Project Details

Subscription * ⓘ ⑤

Resource group * ⓘ ⑥ [Create new](#)

Instance Details

Region ⓘ ⑤

Name * ⓘ ⑥ ✓

i Location specifies the region only for included regional services. This does not specify a region for included non-regional services. [Click here for more details.](#) ⑦

Pricing tier * ⓘ ⑦

[View full pricing details](#)

By checking this box I acknowledge that I have read and understood all the terms below * ⑧

[Previous](#) [Next](#) **Review + create** ⑨

4. In the **Review+submit** tab, once the Validation is Passed, click on the **Create** button.

Microsoft Azure

Search resources, services, and docs

Home > Create Azure AI services ...

Basics Network Identity Tags Review + create

View automation template

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure Pass - Sponsorship
Resource group	AI-Fabric879
Region	East US 2
Name	Cognitive-service237
Pricing tier	Standard S0

Network

Type All networks, including the internet, can access this resource.

Identity

Previous Next **Create**

5. After the deployment is completed, click on the **Go to resource** button.

Microsoft Azure

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

Home > Microsoft.CognitiveServicesAllInOne-20240708121902 | Overview ⚡ ...

Deployment

Search X <> Delete Cancel Redeploy Download Refresh

Overview Deployment name : Microsoft.CognitiveServicesAllInOne-20240708121902 Start time : 7/8/2024, 12:33:41 PM
Subscription : Azure Pass - Sponsorship Correlation ID : fa87fa58-12c9-4c3f-bt
Resource group : AI-Fabric879

Inputs Outputs Template

Your deployment is complete

Deployment details

Next steps

Go to resource

Give feedback

Tell us about your experience with deployment

6. In your **Azure AI service** window, navigate to the **Resource Management** section, and click on **Keys and Endpoints**.

The screenshot shows the Microsoft Azure portal interface. At the top, it says "Microsoft Azure" and "Search resources, s...". Below that, the URL is "Home > Microsoft.CognitiveServicesAllInOne-20240708121902 | Overview >". The main title is "Cognitive-service237" with a cloud icon. Below it, it says "Azure AI services multi-service account". On the left, there's a sidebar with "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Resource Management" (which is expanded), "Keys and Endpoint" (which is highlighted with a red box), "Pricing tier", "Networking", "Identity", "Cost analysis", "Properties", and "Locks". On the right, under "Essentials", it shows: Resource group (move) : AI-Fabric879, Status : Active, Location : East US 2, Subscription (move) : Azure Pass - Sponsorship, Subscription ID : 0c08b20f-9ce4-4949-80d7-7938100, and Tags (edit) : Add tags. Below that, there are tabs for "Get Started" (which is selected), "Decision", "Language", "Speech", and "Visio".

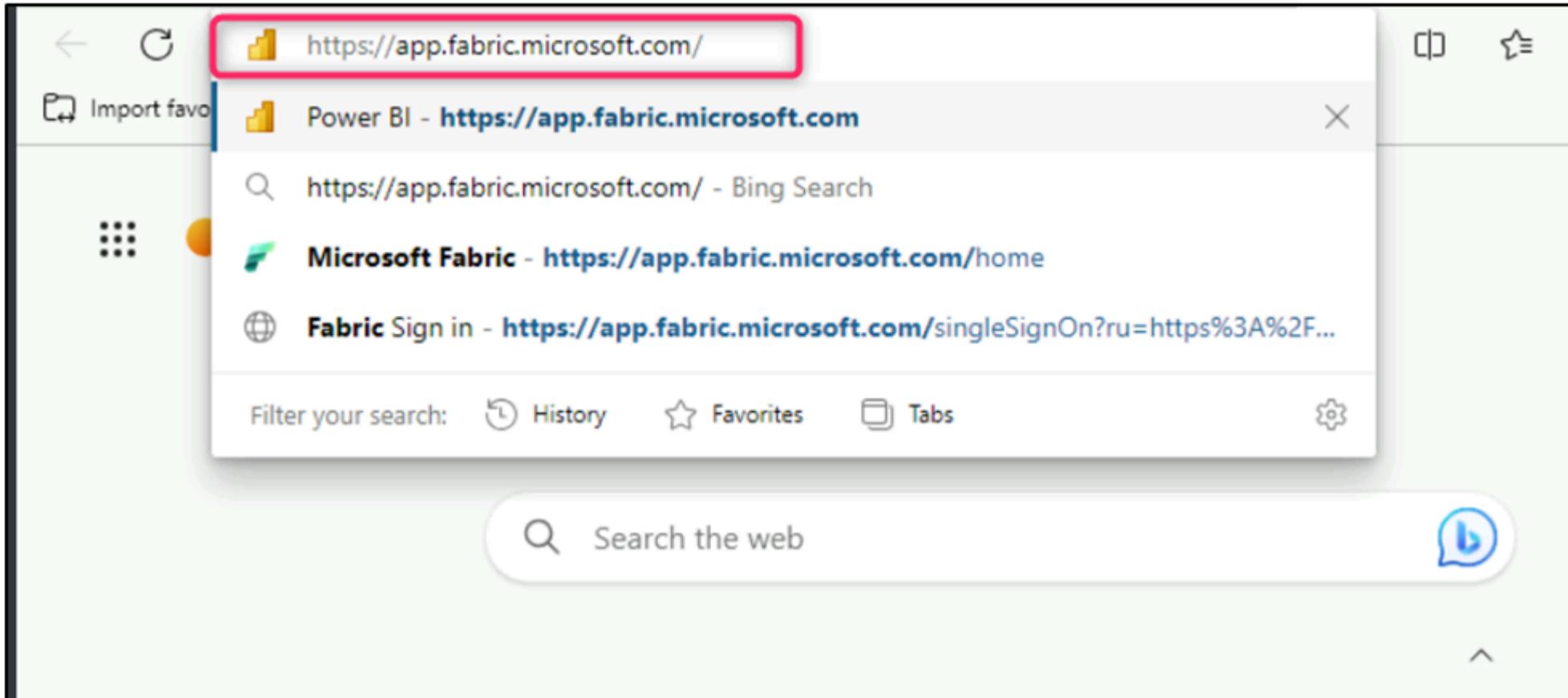
7. In **Keys and Endpoints** page, copy **KEY1**, **KEY 2**, **Location/Region** and **Endpoint** values and paste them in a notepad as shown in the below image, then **Save** the notepad to use the information in the upcoming lab.

The screenshot shows the "Cognitive-service237 | Keys and Endpoint" page. At the top, it says "Microsoft Azure" and "Search resources, services, and docs (G+/-)". Below that, the URL is "Home > Microsoft.CognitiveServicesAllInOne-20240708121902 | Overview > Cognitive-service237". The main title is "Cognitive-service237 | Keys and Endpoint" with a key icon. Below it, it says "Azure AI services multi-service account". On the left, there's a sidebar with "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Resource Management" (which is expanded), "Keys and Endpoint" (which is highlighted with a red box), "Pricing tier", "Networking", "Identity", "Cost analysis", "Properties", and "Locks". On the right, there are two "Regenerate Key" buttons. A callout box points to the "Show Keys" button with the text: "These keys are used to access your Azure AI services API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service." Below that, there are fields for "KEY 1" (with a "Copy to clipboard" button), "KEY 2" (with a "Copy to clipboard" button), "Location/Region" (set to "eastus2"), and "Endpoint" (set to "https://cognitive-service237.cognitiveservices.azure.com/", with a "Copy to clipboard" button highlighted with a red box).

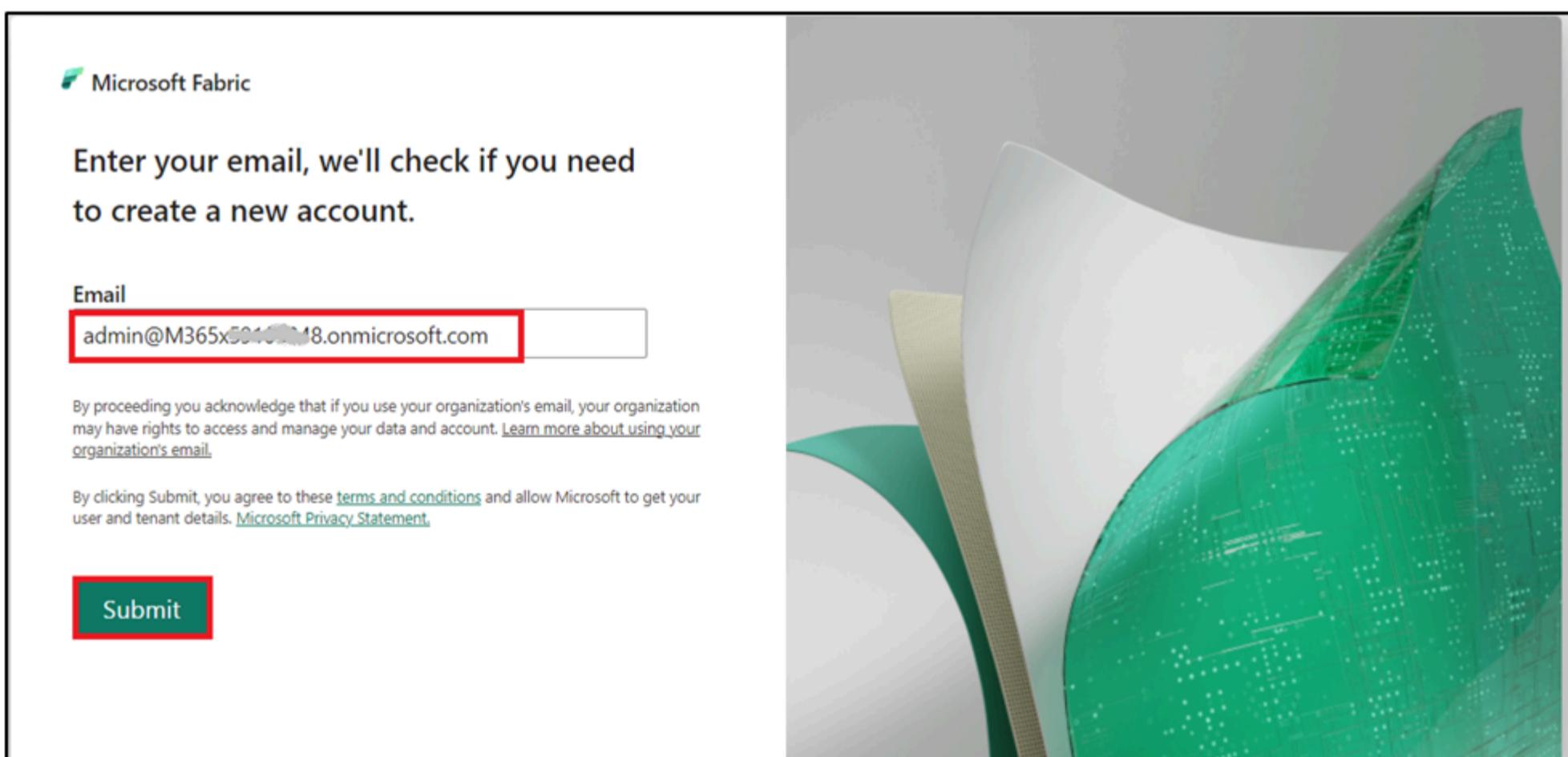
Task 3: Create a Fabric workspace

- ② In this task, you create a Fabric workspace. The workspace contains all the items needed for this lakehouse tutorial, which includes lakehouse, dataflows, Data Factory pipelines, the notebooks, Power BI datasets, and reports.

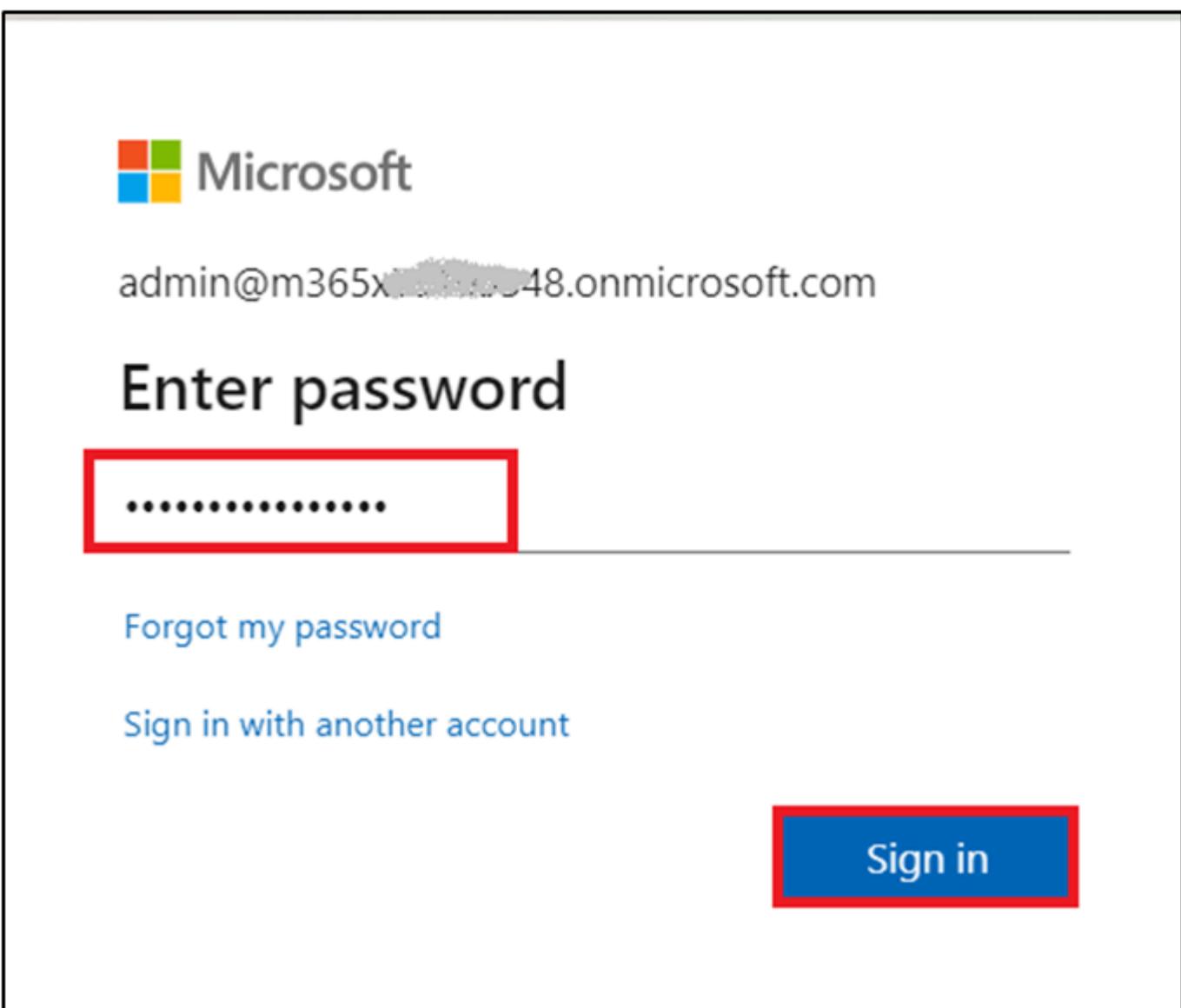
1. Open your browser, navigate to the address bar, and type or paste the following URL:  <https://app.fabric.microsoft.com/> then press the **Enter** button.



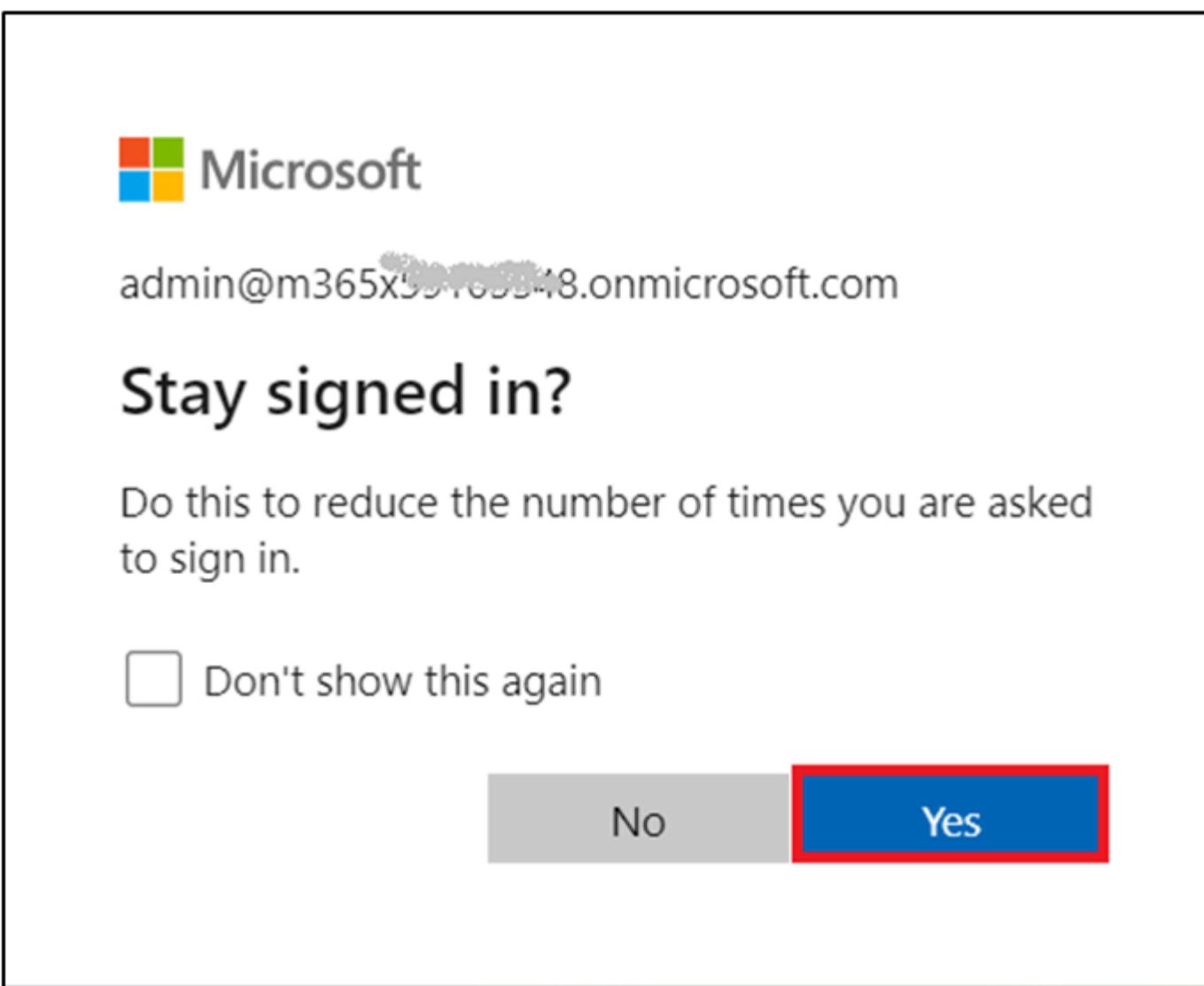
2. In the **Microsoft Fabric** window, enter your **Microsoft 365** credentials, and click on the **Submit** button.



3. Then, In the **Microsoft** window enter the password and click on the **Sign in** button**.**



4. In **Stay signed in?** window, click on the **Yes** button.



5. You'll be directed to Power BI Home page.

Microsoft Fabric Home

Home

Microsoft Fabric

All your data. In one location. Organize. Collaborate. Create.

Explore the experience



Power BI
Find insights, track progress, and make decisions faster using rich visualizations.



Data Factory
Solve complex data ingestion, transformation, and orchestration scenarios using cloud-scale data movement and data transformation services.



Data Activator
Detect patterns and conditions in your Power BI reports and streaming data, and then take actions such as alert users or kick-off workflows.



Synapse Data Engineering
Create a lakehouse and operationalize your workflow to build, transform, and share your data estate.



Synapse Data Science
Unlock powerful insights using AI and machine learning technology.



Synapse Data Warehouse
Scale up your insights by storing and analyzing data in a secure SQL warehouse. Benefit from top-tier performance at petabyte scale in an open-data format.

Microsoft Fabric

6. Go back to **Power BI** window. On the left side navigation menu of Power BI Home page, navigate and click on **Workspaces**.

Power BI Home

Join us at FabCon Europe in Stockholm, Sweden, from September 25 to 27, 2024

+ New report

Recommended

Explore basic Power BI concepts
Getting started with Power BI

The diagram illustrates the data pipeline in Power BI. It starts with a green icon representing a data source (e.g., database). An arrow points from this source to a yellow bar chart icon, representing the transformation or loading of data into Power BI. From the bar chart, an arrow points to a blue cube icon, representing the data being used to generate reports. Finally, an arrow points from the cube to an orange document icon, representing the final report output.

Open

Explore the 100 most useful productivity tips

Explore this data story

THE DEFINITIVE
100
MOST USEFUL PRODUCTIVITY TIPS

Discover all of the 100 practical productivity tips shared by the folks over in their PDF report. Now it's time to see if we can convert the 100 pages of the PDF into an interactive Power BI report. Check out the project to learn more now! See [here](#).

DiscoverIt's Top 5 Tips for visualizing quantitative data in Power BI:

1. Use filters to filter your data
2. Use Card Visuals instead of tables for creating a nice readable gallery
3. Use icons, images, and video to make everything look cool in BI!

1. Productivity tools
2. Data visualization
3. Business intelligence

Open

Recent

Favorites

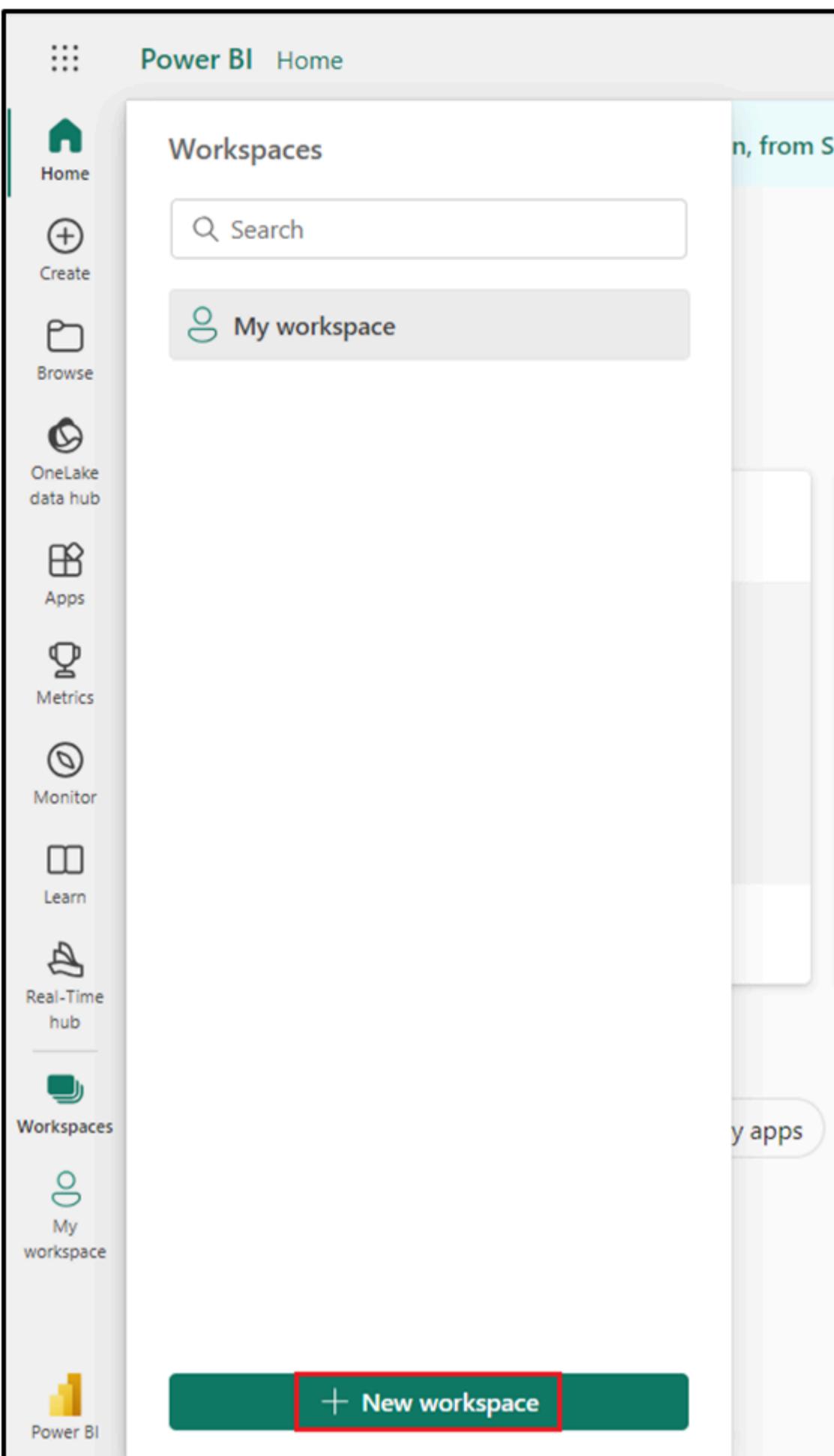
My apps

Workspaces

My workspace

Power BI

7. In the Workspaces pane, click on + New workspace button**.**



8. In the **Create a workspace** pane that appears on the right side, enter the following details, and click on the **Apply** button.

|Name | **AI-Fabric-XXX** | (XXX can be a unique number) (here, we entered AI*-Fabric-XX** *-7891) | |Advanced |Under **License mode**, select **Trial** |
|Default storage format |Small dataset storage format | |Template apps |Check the Develop template apps |

Create a workspace

X

Name *

AI-Fabric-789

1

This name is available

Description

Describe this workspace

Domain ⓘ

Assign to a domain (optional)

▼

[Learn more about workspace settings](#)

Workspace image



Upload

Reset

Advanced ^

Contact list * ⓘ



aoaitfs006041 (Owner)

Enter users and groups

License mode ⓘ

Pro

Select Pro to use basic Power BI features and collaborate on reports, dashboards, and scorecards. To access a Pro workspace, users need Pro per-user licenses. [Learn more](#)

Trial

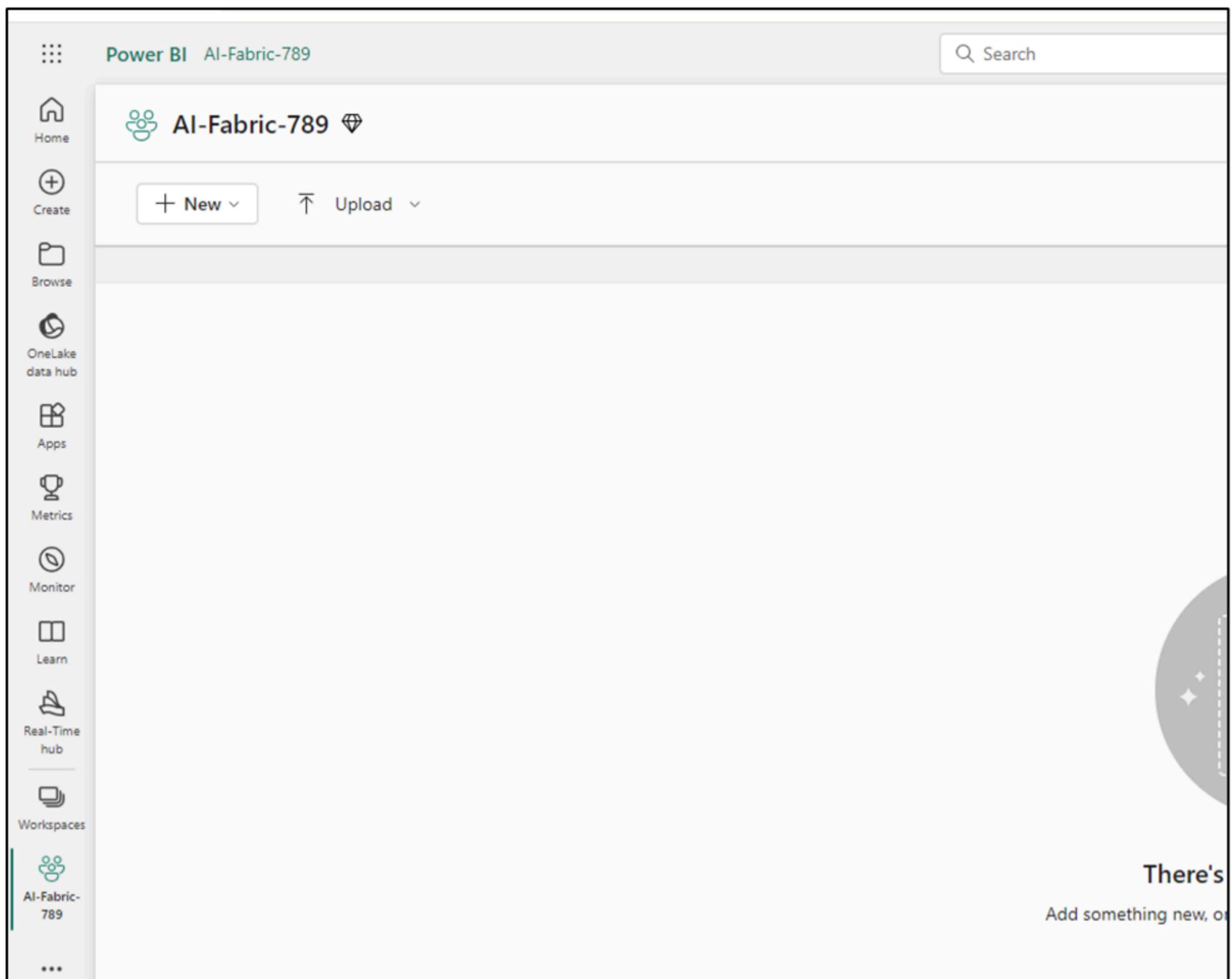
Select the free trial per-user license to try all the new features and experiences in Microsoft Fabric for 60 days. A Microsoft Fabric trial license allows users to create Microsoft Fabric items and collaborate with others in a Microsoft Fabric trial capacity. Explore new

2

Apply

Cancel

9. Wait for the deployment to complete. It takes 2-3 minutes to complete.



Task 4: Create a lakehouse and create a notebook

1. In the **AI-Fabric-XXX** page, click on the **Power BI** icon located at the bottom left and select **Data Engineering** under Synapse.

Power BI Document Intelligence-Fabric34

Document Intelligence-Fabric34 ⚡

+ New ▾ Upload ▾

Microsoft Fabric →

- Power BI
- Data Factory
- Data Activator
- Industry Solutions
- Real-Time Intelligence
- Synapse
- Data Engineering**
- Data Science
- Data Warehouse

2. In the **Synapse Data Engineering** Home page, select **Lakehouse** to create a lakehouse.

Synapse Data Engineering Home

Join us at FabCon Europe in Stockholm, Sweden, from September 25 to 27, 2024 for the ultimate Power BI, Fabric, SQL, and AI learning event. Use code MSCUST

New

Current workspace: AI-Fabric-789

Items will be saved to this workspace.

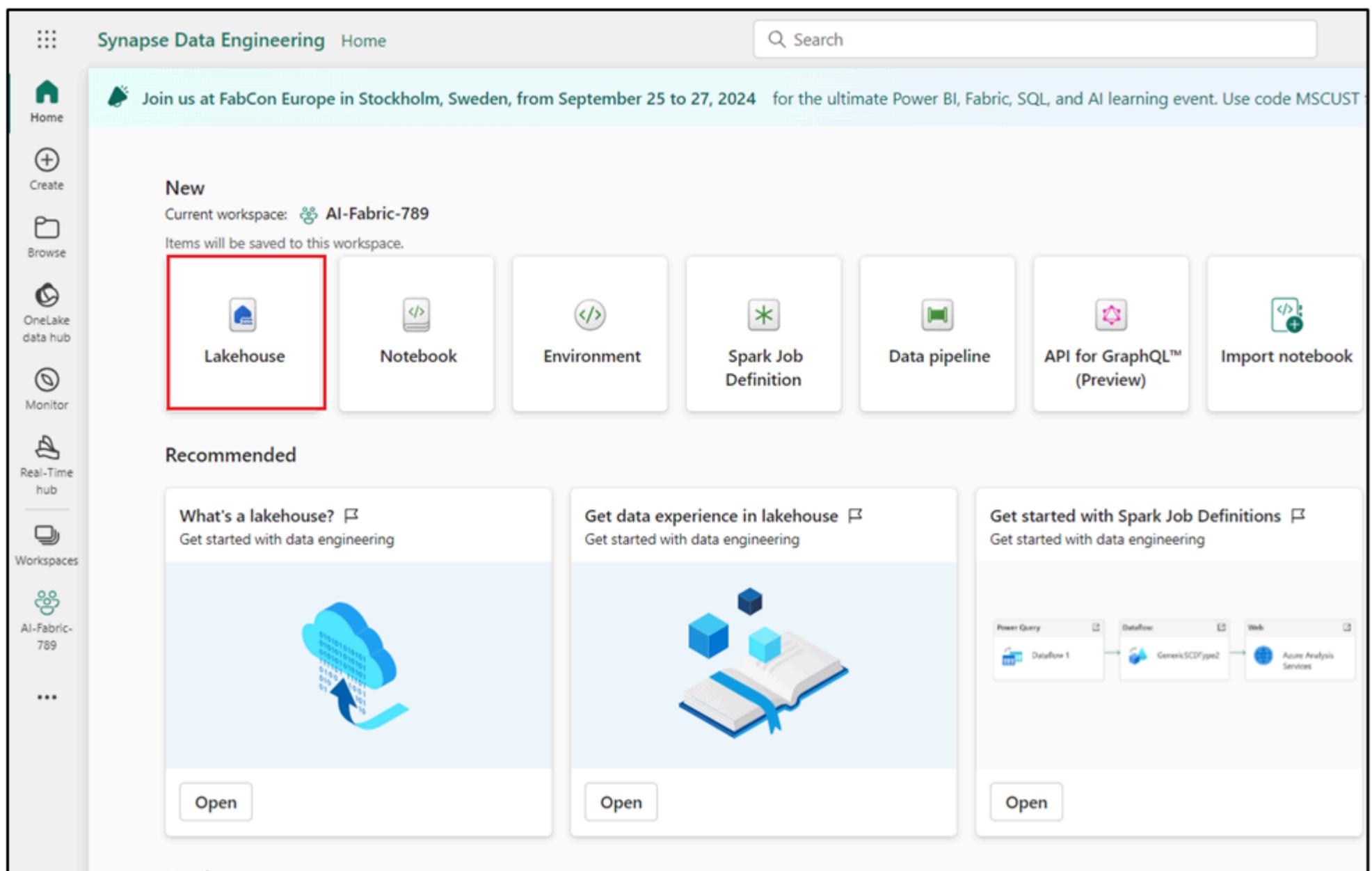
Lakehouse Notebook Environment Spark Job Definition Data pipeline API for GraphQL™ (Preview) Import notebook

Recommended

What's a lakehouse? Get started with data engineering Open

Get data experience in lakehouse Get started with data engineering Open

Get started with Spark Job Definitions Get started with data engineering Open



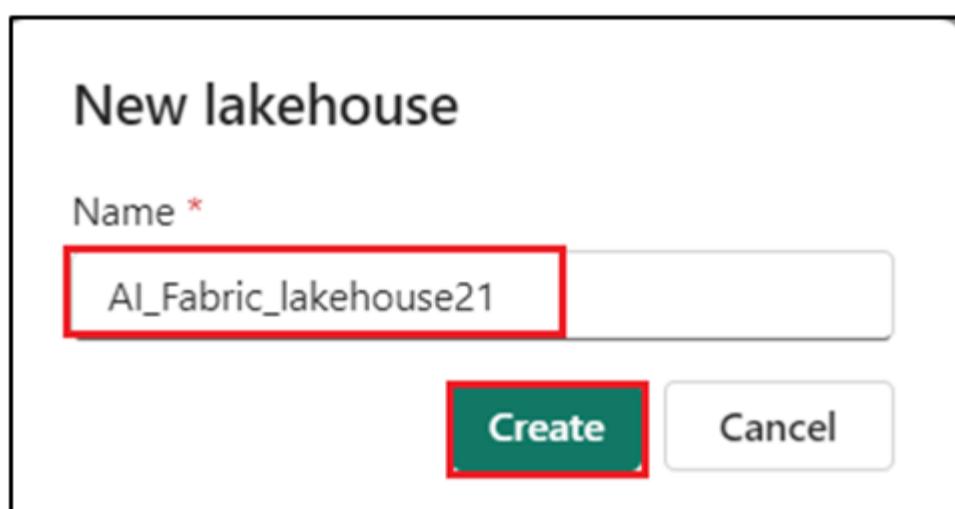
3. In the **New lakehouse** dialog box, enter **AI_Fabric_lakehouseXX** in the **Name** field, click on the **Create** button and open the new lakehouse.

⚠ Note: Ensure to remove space before **AI_Fabric_lakehouseXX**.

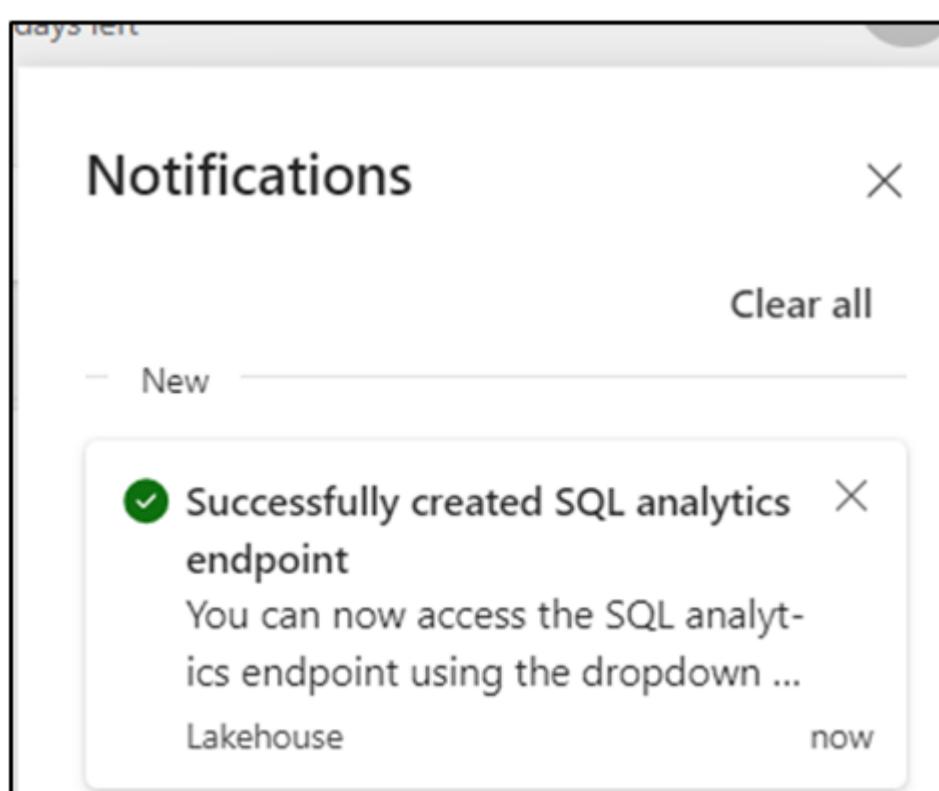
New lakehouse

Name *

Create **Cancel**



4. You will see a notification stating **Successfully created SQL endpoint**.



AI_Fabric_lakehouse21

Home

Get data New semantic model Open notebook Manage OneLake data access (preview)

A SQL analytics endpoint for SQL querying and a default Power BI semantic model for faster reporting were created and will be updated with any tables added to the lakehouse

Explorer

AI_Fabric_lakehouse21

- Tables
- Files

Get data in you

Upload files

New Dataflow Gen2

New data pipeline

5. In the **Lakehouse** page, navigate and click on **Open notebook** drop in the command bar, then select **New notebook**.

AI_Fabric_lakehouse21

Home

Get data New semantic model Open notebook Manage OneLake data access (preview)

A SQL analytics endpoint for SQL querying and a default Power BI semantic model for faster reporting were created and will be updated with any tables added to the lakehouse

Explorer

AI_Fabric_lakehouse21

- Tables
- Files

Get

Existing notebook

New notebook

Exercise 2: Use Azure AI services with SynapseML in Microsoft Fabric

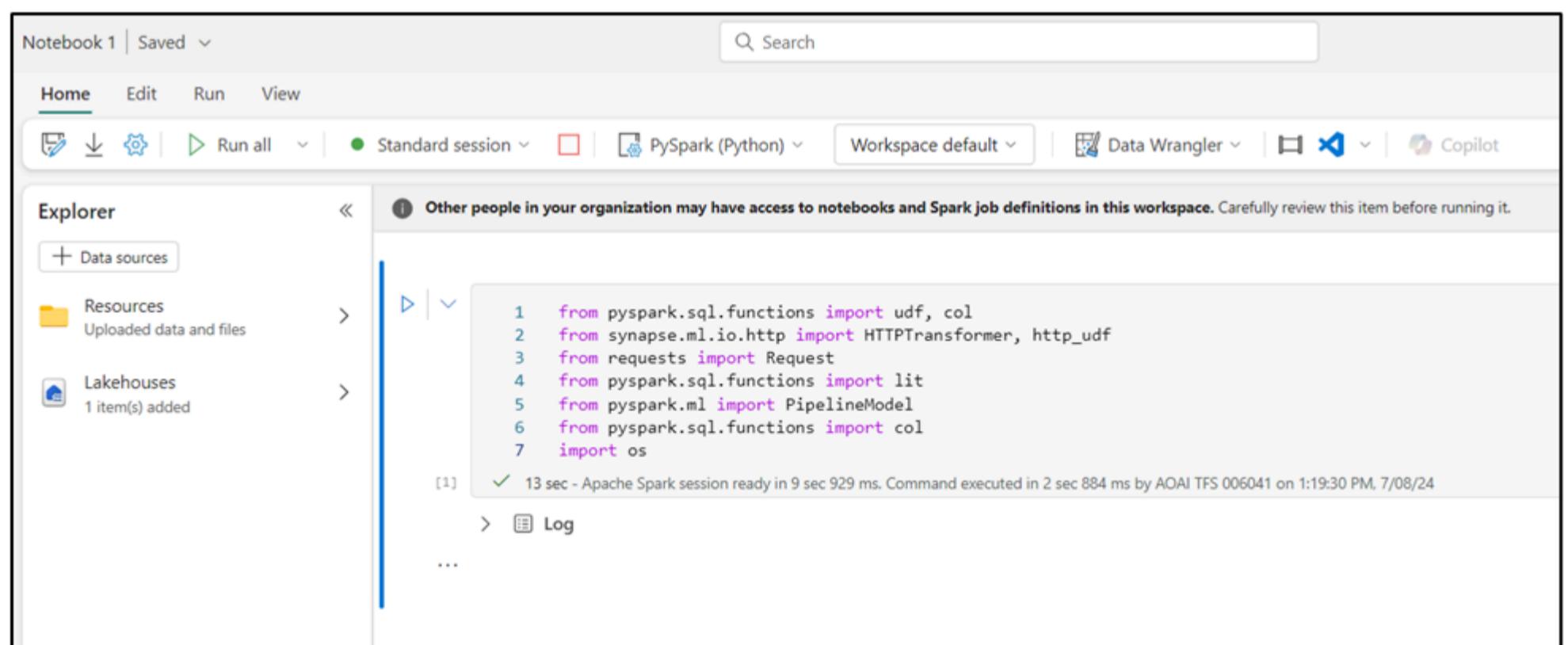
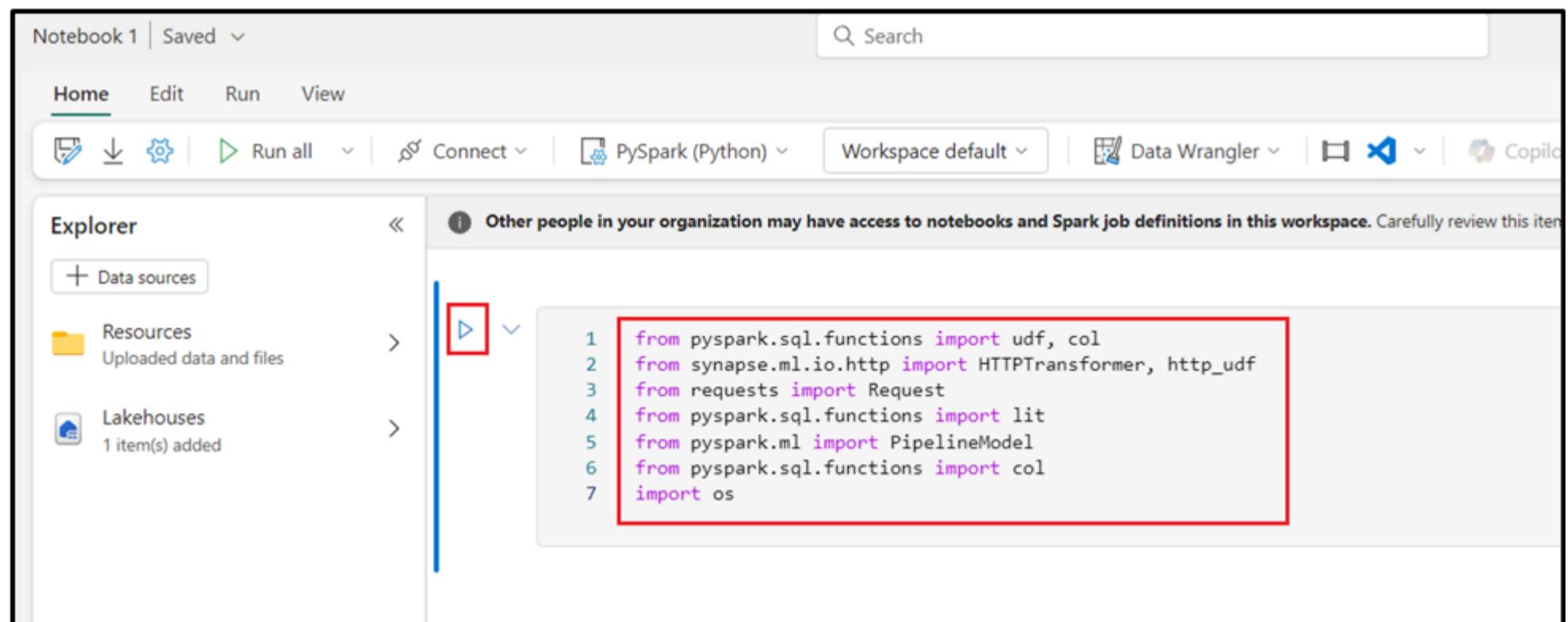
Task 1: Import required libraries and initialize Spark session.

To begin, import required libraries and initialize your Spark session.

1. In the query editor, paste the following code to import required libraries

Copy

```
from pyspark.sql.functions import udf, col  
  
from synapse.ml.io.http import HTTPTransformer, http_udf  
  
from requests import Request  
  
from pyspark.sql.functions import lit  
  
from pyspark.ml import PipelineModel  
  
from pyspark.sql.functions import col  
  
import os
```



2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook, and enter the following code in it. Click on **Run cell**** button and review the output

Copy

```
from pyspark.sql import SparkSession  
  
from synapse.ml.core.platform import *  
  
# Bootstrap Spark Session  
  
spark = SparkSession.builder.getOrCreate()
```

Standard session | PySpark (Python) | Workspace default | Data Wrangler

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before you share it.

```
1 from pyspark.sql.functions import udf, col
2 from synapse.ml.io.http import HTTPTransformer, http_udf
3 from requests import Request
4 from pyspark.sql.functions import lit
5 from pyspark.ml import PipelineModel
6 from pyspark.sql.functions import col
7 import os
```

[1] ✓ 13 sec - Apache Spark session ready in 9 sec 929 ms. Command executed in 2 sec 884 ms by AOAI TFS 006041 on 1:19:30 PM, 7/08/24

> Log

...

+ Code + Markdown

D [2]

```
1 from pyspark.sql import SparkSession
2 from synapse.ml.core.platform import *
3
4 # Bootstrap Spark Session
5 spark = SparkSession.builder.getOrCreate()
```

✓ 1 sec - Command executed in 370 ms by AOAI TFS 006041 on 1:20:33 PM, 7/08/24

...

- 3. Import Azure AI services libraries and replace the keys and locations in the following code snippet with your Azure AI services key and location.
- 4. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 5. Replace your **Azure AI services key** and **location**.

Copy

```
from synapse.ml.cognitive import *

# A general Azure AI services key for Text Analytics, Vision and Document Intelligence (or use separate keys that belong to each service)
service_key = "<YOUR-KEY-VALUE>" # Replace <YOUR-KEY-VALUE> with your Azure AI service key, check prerequisites for more details
service_loc = "eastus"

# A Bing Search v7 subscription key
bing_search_key = "<YOUR-KEY-VALUE>" # Replace <YOUR-KEY-VALUE> with your Bing v7 subscription key, check prerequisites for more detail
# An Anomaly Detector subscription key
anomaly_key = "<YOUR-KEY-VALUE>" # Replace <YOUR-KEY-VALUE> with your anomaly service key, check prerequisites for more details
anomaly_loc = "westus2"

# A Translator subscription key
translator_key = "<YOUR-KEY-VALUE>" # Replace <YOUR-KEY-VALUE> with your translator service key, check prerequisites for more details
translator_loc = "eastus"

# An Azure search key
search_key = "<YOUR-KEY-VALUE>" # Replace <YOUR-KEY-VALUE> with your search key, check prerequisites for more details
```

The screenshot shows a PySpark notebook interface with several tabs at the top: Standard session, PySpark (Python), Workspace default, Data Wrangler, and Copilot. A message bar at the top states: "Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it." Below this, there is a code cell containing Python code for initializing Azure AI services. The code imports `synapse.ml.cognitive` and defines variables for service keys and locations for various services: Text Analytics, Bing Search v7, Anomaly Detector, Translator, and Azure search. The code cell has a red border and a green checkmark indicating successful execution. The output shows deprecation warnings from the `synapse.ml.cognitive` module.

```
1 from synapse.ml.cognitive import *
2
3 # A general Azure AI services key for Text Analytics, Vision and Document Intelligence (or use separate keys that belong to each service)
4 service_key = "3de4a0cc4e014f618ff3afa080a10c92" # Replace <YOUR-KEY-VALUE> with your Azure AI service key, check prerequisites for more details
5 service_loc = "eastus2"
6
7 # A Bing Search v7 subscription key
8 bing_search_key = "3de4a0cc4e014f618ff3afa080a10c92" # Replace <YOUR-KEY-VALUE> with your Bing v7 subscription key, check prerequisites for more details
9
10 # An Anomaly Detector subscription key
11 anomaly_key = "3de4a0cc4e014f618ff3afa080a10c92" # Replace <YOUR-KEY-VALUE> with your anomaly service key, check prerequisites for more details
12 anomaly_loc = "eastus2"
13
14 # A Translator subscription key
15 translator_key = "3de4a0cc4e014f618ff3afa080a10c92" # Replace <YOUR-KEY-VALUE> with your translator service key, check prerequisites for more details
16 translator_loc = "eastus2"
17
18 # An Azure search key
19 search_key = "3de4a0cc4e014f618ff3afa080a10c92" # Replace <YOUR-KEY-VALUE> with your search key, check prerequisites for more details
```

[3] ✓ 1 sec - Command executed in 1 sec 25 ms by AOAI TFS 006041 on 1:53:12 PM, 7/08/24

```
... /home/trusted-service-user/cluster-env/trident_env/lib/python3.10/site-packages/synapse/ml/internal_utils/import_hooks/__init__.py:303: DeprecationWarning: Importing 'HasElasticNetParam' from 'synapse.ml.cognitive' is deprecated. Use 'synapse.ml.services' instead.
    module = self.loader.load_module(fullname)
/home/trusted-service-user/cluster-env/trident_env/lib/python3.10/site-packages/synapse/ml/internal_utils/import_hooks/__init__.py:303: DeprecationWarning: Importing 'HasFeaturesCol' from 'synapse.ml.cognitive' is deprecated. Use 'synapse.ml.services' instead.
    module = self.loader.load_module(fullname)
/home/trusted-service-user/cluster-env/trident_env/lib/python3.10/site-packages/synapse/ml/internal_utils/import_hooks/__init__.py:303: DeprecationWarning: Importing 'HasLabelCol' from 'synapse.ml.cognitive' is deprecated. Use 'synapse.ml.services' instead.
```

Task 2: Perform sentiment analysis on text

?

The **Text Analytics** service provides several algorithms for extracting intelligent insights from text. For example, you can use the service to find the sentiment of some input text. The service will return a score between 0.0 and 1.0, where low scores indicate negative sentiment and high scores indicate positive sentiment.

- 1. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 2. Enter the following code in it. Click on **> Run cell** button and review the output

Copy

```
# Create a dataframe that's tied to it's column names
df = spark.createDataFrame(
[
    ("I am so happy today, its sunny!", "en-US"),
    ("I am frustrated by this rush hour traffic", "en-US"),
    ("The cognitive services on spark aint bad", "en-US"),
],
["text", "language"],
)

# Run the Text Analytics service with options
sentiment = (
    TextSentiment()
    .setTextCol("text")
    .setLocation(service_loc)
    .setSubscriptionKey(service_key)
    .setOutputCol("sentiment")
    .setErrorCol("error")
    .setLanguageCol("language")
)

# Show the results of your text query in a table format
display(
    sentiment.transform(df).select(
        "text", col("sentiment.document.sentiment").alias("sentiment")
    )
)
```

Standard session | PySpark (Python) | Workspace default | Data Wrangler | Copilot

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

/home/trusted-service-user/cluster-env/trident_env/lib/python3.10/site-packages/synapse/ml/internal_utils/import_hooks/__init__.py:303:...

Show all (43.8 kB)

+ Code -1 Markdown

3

```

1 # Create a dataframe that's tied to its column names
2 df = spark.createDataFrame(
3     [
4         ("I am so happy today, its sunny!", "en-US"),
5         ("I am frustrated by this rush hour traffic", "en-US"),
6         ("The cognitive services on spark aint bad", "en-US"),
7     ],
8     ["text", "language"],
9 )
10
11 # Run the Text Analytics service with options
12 sentiment = (
13     TextSentiment()
14     .setTextCol("text")
15     .setLocation(service_loc)
16     .setSubscriptionKey(service_key)
17     .setOutputCol("sentiment")
18     .setErrorCol("error")
19     .setLanguageCol("language")
20 )
21
22 # Show the results of your text query in a table format
23 display(
24     sentiment.transform(df).select(
25         "text", col("sentiment.document.sentiment").alias("sentiment")
26     )
27 )

```

2

```

17     .setOutputCol("sentiment")
18     .setErrorCol("error")
19     .setLanguageCol("language")
20 )
21
22 # Show the results of your text query in a table format
23 display(
24     sentiment.transform(df).select(
25         "text", col("sentiment.document.sentiment").alias("sentiment")
26     )
27 )

```

[4] ✓ 23 sec - Command executed in 22 sec 519 ms by AOAI TFS 006041 on 1:56:26 PM, 7/08/24

> Spark jobs (3 of 3 succeeded) Resources Log

Table		Chart	Download	Showing rows 1 - 3
ABC	text	ABC	sentiment	
1	I am so happy today, its sunny!		positive	
2	I am frustrated by this rush hour traffic		negative	
3	The cognitive services on spark aint bad		neutral	

Task 3: Perform text analytics for health data

? The **Text Analytics for Health Service** extracts and labels relevant medical information from unstructured text such as doctor's notes, discharge summaries, clinical documents, and electronic health records.

- 1. The following code sample analyzes and transforms text from doctors notes into structured data.
- 2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on **> Run cell** button and review the output

Copy

```

df = spark.createDataFrame(
    [
        ("20mg of ibuprofen twice a day",),
        ("1tsp of Tylenol every 4 hours",),
        ("6-drops of Vitamin B-12 every evening",),
    ],
    ["text"],
)

healthcare = (
    AnalyzeHealthText()
    .setSubscriptionKey(service_key)
    .setLocation(service_loc)
    .setLanguage("en")
    .setOutputCol("response")
)
display(healthcare.transform(df))

```

1 Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

	ABC text	ABC sentiment
1	I am so happy today, its sunny!	positive
2	I am frustrated by this rush hour t...	negative
3	The cognitive services on spark ai...	neutral

2

3

Code **1** **Markdown**

```

1 df = spark.createDataFrame(
2     [
3         ("20mg of ibuprofen twice a day",),
4         ("1tsp of Tylenol every 4 hours",),
5         ("6-drops of Vitamin B-12 every evening",),
6     ],
7     ["text"],
8 )
9
10 healthcare = (
11     AnalyzeHealthText()
12     .setSubscriptionKey(service_key)
13     .setLocation(service_loc)
14     .setLanguage("en")
15     .setOutputCol("response")
16 )
17
18 display(healthcare.transform(df))

```

[5] ✓ 27 sec - Command executed in 27 sec 3 ms by AOAI TFS 006041 on 4:03:08 PM, 7/08/24

The screenshot shows a PySpark notebook interface. At the top, there's a message: "Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it." Below this is a code cell containing Python code for processing a DataFrame and using the AnalyzeHealthText service. The code is as follows:

```

1 df = spark.createDataFrame(
2     [
3         ("20mg of ibuprofen twice a day",),
4         ("1tsp of Tylenol every 4 hours",),
5         ("6-drops of Vitamin B-12 every evening",),
6     ],
7     ["text"],
8 )
9
10 healthcare = (
11     AnalyzeHealthText()
12     .setSubscriptionKey(service_key)
13     .setLocation(service_loc)
14     .setLanguage("en")
15     .setOutputCol("response")
16 )
17
18 display(healthcare.transform(df))

```

Below the code cell, a status message indicates: "✓ 27 sec - Command executed in 27 sec 3 ms by A0AI TFS 006041 on 4:03:08 PM, 7/08/24". Underneath the code cell, there are tabs for "Spark jobs (3 of 3 succeeded)", "Resources", and "Log".

At the bottom, there's a table view showing the results of the analysis. The table has three columns: "ABC text", "ANY AnalyzeHealthText_37dfcbc90a...", and "ANY response". The data is as follows:

	ABC text	ANY AnalyzeHealthText_37dfcbc90a...	ANY response
1	20mg of ibuprofen t...	NULL	{"document":{"id":0,"entities":[{"confidenceScore":0.99,"text":"20mg","offset":0,"category":"Dosage","length":4}, {"confidenceScore":1,"text":"ibuprof..."]}}
2	1tsp of Tylenol ever...	NULL	{"document":{"id":0,"entities":[{"confidenceScore":1,"text":"1tsp","offset":0,"category":"Dosage","length":4}, {"confidenceScore":1,"text":"Tylenol","of..."]}}
3	6-drops of Vitamin ...	NULL	{"document":{"id":0,"entities":[{"confidenceScore":0.95,"text":"6","offset":0,"category":"Dosage","length":1}, {"confidenceScore":0.62,"text":"drops",...}]}]

Task 4: Translate text into a different language

Translator is a cloud-based machine translation service and is part of the Azure AI services family of cognitive APIs used to build intelligent apps. Translator is easy to integrate in your applications, websites, tools, and solutions. It allows you to add multi-language user experiences in 90 languages and dialects and can be used for text translation with any operating system.

- 1. The following code sample does a simple text translation by providing the sentences you want to translate and target languages you want to translate them to.
- 2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on **Run cell** button and review the output

Copy

```
from pyspark.sql.functions import col, flatten

# Create a dataframe including sentences you want to translate

df = spark.createDataFrame(
    [(["Hello, what is your name?", "Bye"],)],
    [
        "text",
    ],
)

# Run the Translator service with options

translate = (
    Translate()
    .setSubscriptionKey(translator_key)
    .setLocation(translator_loc)
    .setTextCol("text")
    .setToLanguage(["zh-Hans"])
    .setOutputCol("translation")
)

# Show the results of the translation.

display(
    translate.transform(df)
    .withColumn("translation", flatten(col("translation.translations")))
    .withColumn("translation", col("translation.text"))
    .select("translation")
)
```

Standard session ▾ | PySpark (Python) ▾ | Workspace default ▾ | Data Wrangler ▾ | Copilot

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

Code 1+ Markdown

3

```
1 from pyspark.sql.functions import col, flatten
2
3 # Create a dataframe including sentences you want to translate
4 df = spark.createDataFrame(
5     [{"Hello, what is your name?", "Bye"},],
6     [
7         "text",
8     ],
9 )
10
11 # Run the Translator service with options
12 translate = (
13     Translate()
14     .setSubscriptionKey(translator_key)
15     .setLocation(translator_loc)
16     .setTextCol("text")
17     .setToLanguage(["zh-Hans"])
18     .setOutputCol("translation")
19 )
20
21 # Show the results of the translation.
22 display(
23     translate.transform(df)
24     .withColumn("translation", flatten(col("translation.translations")))
25     .withColumn("translation", col("translation.text"))
26     .select("translation")
27 )
```

2

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```
9 )
10
11 # Run the Translator service with options
12 translate = (
13     Translate()
14     .setSubscriptionKey(translator_key)
15     .setLocation(translator_loc)
16     .setTextCol("text")
17     .setToLanguage(["zh-Hans"])
18     .setOutputCol("translation")
19 )
20
21 # Show the results of the translation.
22 display(
23     translate.transform(df)
24     .withColumn("translation", flatten(col("translation.translations")))
25     .withColumn("translation", col("translation.text"))
26     .select("translation")
27 )
```

[6] ✓ 4 sec - Command executed in 4 sec 73 ms by AOAI TFS 006041 on 4:07:35 PM, 7/08/24

> Spark jobs (3 of 3 succeeded) Resources Log

...

Table Chart | Download ▾ | Showing rows 1 - 1

	ANY translation
1	["你好，你叫什么名字？", "再见"]

Task 5: Extract information from a document into structured data

Azure AI Document Intelligence is a part of Azure AI services that lets you build automated data processing software using machine learning technology. With Azure AI Document Intelligence, you can identify and extract text, key/value pairs, selection marks, tables, and structure from your documents. The service outputs structured data that includes the relationships in the original file, bounding boxes, confidence and more.

1. The following code sample analyzes a business card image and extracts its information into structured data.

- 2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on **> Run cell** button and review the output

Copy

```
 from pyspark.sql.functions import col, explode

# Create a dataframe containing the source files

imageDf = spark.createDataFrame(
    [
        (
            "https://mmlspark.blob.core.windows.net/datasets/FormRecognizer/business_card.jpg",
        )
    ],
    [
        "source",
    ],
)

# Run the Form Recognizer service

analyzeBusinessCards = (
    AnalyzeBusinessCards()
    .setSubscriptionKey(service_key)
    . setLocation(service_loc)
    . setImageUrlCol("source")
    . setOutputCol("businessCards")
)

# Show the results of recognition.

display(
    analyzeBusinessCards.transform(imageDf)
    .withColumn(
        "documents", explode(col("businessCards.analyzeResult.documentResults.fields"))
    )
    .select("source", "documents")
)
```

Standard session | [File](#) | [View](#) | [Run](#) | [PySpark \(Python\)](#) | [Workspaces](#) | [Data Wrangler](#) | [Help](#) | [Copilot](#)

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

[+ Code](#) [Markdown](#)

3

```
1 from pyspark.sql.functions import col, explode
2
3 # Create a dataframe containing the source files
4 imageDf = spark.createDataFrame(
5     [
6         (
7             "https://mmlspark.blob.core.windows.net/datasets/FormRecognizer/business_card.jpg",
8         )
9     ],
10    [
11        "source",
12    ],
13)
14
15 # Run the Form Recognizer service
16 analyzeBusinessCards = (
17     AnalyzeBusinessCards()
18     .setSubscriptionKey(service_key)
19     .setLocation(service_loc)
20     .setImageUrlCol("source")
21     .setOutputCol("businessCards")
22 )
23
24 # Show the results of recognition.
25 display(
26     analyzeBusinessCards.transform(imageDf)
27     .withColumn(
28         "documents", explode(col("businessCards.analyzeResult.documentResults.fields"))
29     )
30     .select("source", "documents")
31 )
```

2

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```
11     "source",
12 ],
13 )
14
15 # Run the Form Recognizer service
16 analyzeBusinessCards = (
17     AnalyzeBusinessCards()
18     .setSubscriptionKey(service_key)
19     .setLocation(service_loc)
20     .setImageUrlCol("source")
21     .setOutputCol("businessCards")
22 )
23
24 # Show the results of recognition.
25 display(
26     analyzeBusinessCards.transform(imageDf)
27     .withColumn(
28         "documents", explode(col("businessCards.analyzeResult.documentResults.fields"))
29     )
30     .select("source", "documents")
31 )
```

[7] ✓ 9 sec - Command executed in 8 sec 866 ms by AOAI TFS 006041 on 4:11:31 PM, 7/08/24

> [Spark jobs \(3 of 3 succeeded\)](#) [Resources](#) [Log](#)

...

	Table	Chart	Download	Showing rows 1 - 1
ABC	source			ANY documents
1	https://mmlspark.blob.core.windows.net/datasets/FormRecognizer/business_card.jpg			{"Addresses":{"type": "...}}

Task 6: Analyze and tag images

Computer Vision analyzes images to identify structure such as faces, objects, and natural-language descriptions.

- 1. The following code sample analyzes images and labels them with *tags*. Tags are one-word descriptions of things in the image, such as recognizable objects, people, scenery, and actions.
- 2. Use the [+ Code](#) icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on [Run cell](#) button and review the output

[Copy](#)

```
# Create a dataframe with the image URLs  
  
base_url = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/"  
  
df = spark.createDataFrame(  
    [  
        (base_url + "objects.jpg",),  
        (base_url + "dog.jpg",),  
        (base_url + "house.jpg",),  
    ],  
    ["image"],  
)  
  
# Run the Computer Vision service. Analyze Image extracts information from/about the images.  
  
analysis = (  
    AnalyzeImage()  
    .setLocation(service_loc)  
    .setSubscriptionKey(service_key)  
    .setVisualFeatures(  
        ["Categories", "Color", "Description", "Faces", "Objects", "Tags"]  
    )  
    .setOutputCol("analysis_results")  
    .setImageUrlCol("image")  
    .setErrorCol("error")  
)  
  
# Show the results of what you wanted to pull out of the images.  
  
display(analysis.transform(df).select("image", "analysis_results.description.tags"))
```

Standard session | PySpark (Python) | Workspace default | Data Wrangler | Copilot

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

+ Code 1+ Markdown

3

```
1 # Create a dataframe with the image URLs
2 base_url = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images"
3 df = spark.createDataFrame([
4     [
5         (base_url + "objects.jpg"),
6         (base_url + "dog.jpg"),
7         (base_url + "house.jpg"),
8     ],
9     [
10        "image",
11    ],
12 )
13
14 # Run the Computer Vision service. Analyze Image extracts information from/about the images.
15 analysis = (
16     AnalyzeImage()
17     .setLocation(service_loc)
18     .setSubscriptionKey(service_key)
19     .setVisualFeatures(
20         ["Categories", "Color", "Description", "Faces", "Objects", "Tags"]
21     )
22     .setOutputCol("analysis_results")
23     .setImageUrlCol("image")
24     .setErrorCol("error")
25 )
26
27 # Show the results of what you wanted to pull out of the images.
28 display(analysis.transform(df).select("image", "analysis_results.description.tags"))
```

Standard session | PySpark (Python) | Workspace default | Data Wrangler | Copilot

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```
16     AnalyzeImage()
17     .setLocation(service_loc)
18     .setSubscriptionKey(service_key)
19     .setVisualFeatures(
20         ["Categories", "Color", "Description", "Faces", "Objects", "Tags"]
21     )
22     .setOutputCol("analysis_results")
23     .setImageUrlCol("image")
24     .setErrorCol("error")
25 )
26
27 # Show the results of what you wanted to pull out of the images.
28 display(analysis.transform(df).select("image", "analysis_results.description.tags"))
```

[8] ✓ 6 sec - Command executed in 5 sec 576 ms by AOAI TFS 006041 on 4:20:46 PM, 7/08/24

PySpark

> Spark jobs (3 of 3 succeeded) Resources Log

ABC image	ANY tags
1 https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/objects.jpg	["skating", ...]
2 https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/dog.jpg	["dog", "out..."]
3 https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVision/Images/house.jpg	["outdoor", ...]

Task 7: Search for images that are related to a natural language query

② **Bing Image Search** searches the web to retrieve images related to a user's natural language query.

- 1. The following code sample uses a text query that looks for images with quotes. The output of the code is a list of image URLs that contain photos related to the query.
- 2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on **▷ Run cell** button and review the output

Copy

```
# Number of images Bing will return per query

imgsPerBatch = 10

# A list of offsets, used to page into the search results

offsets = [(i * imgsPerBatch,) for i in range(100)]

# Since web content is our data, we create a dataframe with options on that data: offsets

bingParameters = spark.createDataFrame(offsets, ["offset"])

# Run the Bing Image Search service with our text query

bingSearch = (

    BingImageSearch()

    .setSubscriptionKey(bing_search_key)

    .setOffsetCol("offset")

    .setQuery("Martin Luther King Jr. quotes")

    .setCount(imgsPerBatch)

    .setOutputCol("images")

)

# Transformer that extracts and flattens the richly structured output of Bing Image Search into a simple URL column

getUrls = BingImageSearch.getUrlTransformer("images", "url")

# This displays the full results returned, uncomment to use

# display(bingSearch.transform(bingParameters))

# Since we have two services, they are put into a pipeline

pipeline = PipelineModel(stages=[bingSearch, getUrls])

# Show the results of your search: image URLs

display(pipeline.transform(bingParameters))
```

Standard session | PySpark (Python) | Workspace default | Data Wrangler | Copilot

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

Code **Markdown**

Run 3

```

1 # Number of images Bing will return per query
2 imgsPerBatch = 10
3 # A list of offsets, used to page into the search results
4 offsets = [(i * imgsPerBatch,) for i in range(100)]
5 # Since web content is our data, we create a dataframe with options on that data: offsets
6 bingParameters = spark.createDataFrame(offsets, ["offset"])
7
8 # Run the Bing Image Search service with our text query
9 bingSearch = (
10     BingImageSearch()
11     .setSubscriptionKey(bing_search_key)
12     .setOffsetCol("offset")
13     .setQuery("Martin Luther King Jr. quotes")
14     .setCount(imgsPerBatch)
15     .setOutputCol("images")
16 )
17
18 # Transformer that extracts and flattens the richly structured output of Bing Image Search into a simple URL column
19 getUrl = BingImageSearch.getUrlTransformer("images", "url")
20
21 # This displays the full results returned, uncomment to use
22 # display(bingSearch.transform(bingParameters))
23
24 # Since we have two services, they are put into a pipeline
25 pipeline = PipelineModel(stages=[bingSearch, getUrl])
26
27 # Show the results of your search: image URLs
28 display(pipeline.transform(bingParameters))|
```

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```

7
8 # Run the Bing Image Search service with our text query
9 bingSearch = (
10     BingImageSearch()
11     .setSubscriptionKey(bing_search_key)
12     .setOffsetCol("offset")
13     .setQuery("Martin Luther King Jr. quotes")
14     .setCount(imgsPerBatch)
15     .setOutputCol("images")
16 )
17
18 # Transformer that extracts and flattens the richly structured output of Bing Image Search into a simple URL column
19 getUrl = BingImageSearch.getUrlTransformer("images", "url")
20
21 # This displays the full results returned, uncomment to use
22 # display(bingSearch.transform(bingParameters))
23
24 # Since we have two services, they are put into a pipeline
25 pipeline = PipelineModel(stages=[bingSearch, getUrl])
26
27 # Show the results of your search: image URLs
28 display(pipeline.transform(bingParameters))|
```

[9] ✓ 3 sec - Command executed in 2 sec 877 ms by AOAI TFS 006041 on 4:26:06 PM, 7/08/24

Spark jobs (3 of 3 succeeded) Resources Log

ID	Description	Status	Stages	Tasks	Duration	Processed	Data re
Job 26	getRowsInJsonString at Display.scala:470	Succeeded	1/1	3/3 succeeded	341 ms	0 rows	0 B
Job 25	getRowsInJsonString at Display.scala:470	Succeeded	1/1	4/4 succeeded	339 ms	0 rows	0 B
Job 24	getRowsInJsonString at Display.scala:470	Succeeded	1/1	1/1 succeeded	513 ms	0 rows	0 B

Task 8: Transform speech to text

The **Speech-to-text** service converts streams or files of spoken audio to text.

- 1. The following code sample transcribes one audio file to text.
- 2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on **Run cell** button and review the output

Copy

```
# Create a dataframe with our audio URLs, tied to the column called "url"
df = spark.createDataFrame(
    [("https://mmlspark.blob.core.windows.net/datasets/Speech/audio2.wav",)], ["url"]
)

# Run the Speech-to-text service to translate the audio into text
speech_to_text = (
    SpeechToTextSDK()
    .setSubscriptionKey(service_key)
    .setLocation(service_loc)
    .setOutputCol("text")
    .setAudioDataCol("url")
    .setLanguage("en-US")
    .setProfanity("Masked")
)

# Show the results of the translation
display(speech_to_text.transform(df).select("url", "text.DisplayText"))
```

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** Standard session, PySpark (Python), Workspace default, Data Wrangler, Copilot.
- Notebook Content:**
 - A warning message: "Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it."
 - A code cell containing the provided Python code for creating a DataFrame and using the SpeechToTextSDK.
 - The code cell has a red border and is labeled with a red number "3" in the top-left corner.
 - The output of the code cell is visible below, showing a table with one row of data.
 - The output table has a red border and is labeled with a red number "2" in the top-right corner.
- Table Output:**

ABC url	ABC DisplayText
1 https://mmlspark.blob.core.windows.net/datasets/Speech/audio2....	Custom Speech provides tools that allow you to visually inspect the recognition quality of a model by comparing audio data

Task 9: Transform text to speech

② **Text to speech** is a service that allows you to build apps and services that speak naturally, choosing from more than 270 neural voices across 119 languages and variants.

- 1. The following code sample transforms text into an audio file that contains the content of the text.
- 2. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 3. Enter the following code in it. Click on **Run cell** button and review the output

Copy

```

from synapse.ml.cognitive import TextToSpeech

fs = ""

if running_on_databricks():
    fs = "dbfs:"

elif running_on_synapse_internal():
    fs = "Files"

# Create a dataframe with text and an output file location

df = spark.createDataFrame(
    [
        (
            "Reading out loud is fun! Check out aka.ms/spark for more information",
            fs + "/output.mp3",
        )
    ],
    ["text", "output_file"],
)

tts = (
    TextToSpeech()
    .setSubscriptionKey(service_key)
    .setTextCol("text")
    . setLocation(service_loc)
    .setVoiceName("en-US-JennyNeural")
    .setOutputFileCol("output_file")
)

# Check to make sure there were no errors during audio creation

display(tts.transform(df))

```

Task 10: Detect anomalies in time series data

? [Anomaly Detector](#) is great for detecting irregularities in your time series data. The following code sample uses the Anomaly Detector service to find anomalies in entire time series data.

- 1. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 2. Enter the following code in it. Click on **> Run cell** button and review the output

```

# Create a dataframe with the point data that Anomaly Detector requires

df = spark.createDataFrame(
[

    ("1972-01-01T00:00:00Z", 826.0),
    ("1972-02-01T00:00:00Z", 799.0),
    ("1972-03-01T00:00:00Z", 890.0),
    ("1972-04-01T00:00:00Z", 900.0),
    ("1972-05-01T00:00:00Z", 766.0),
    ("1972-06-01T00:00:00Z", 805.0),
    ("1972-07-01T00:00:00Z", 821.0),
    ("1972-08-01T00:00:00Z", 20000.0),
    ("1972-09-01T00:00:00Z", 883.0),
    ("1972-10-01T00:00:00Z", 898.0),
    ("1972-11-01T00:00:00Z", 957.0),
    ("1972-12-01T00:00:00Z", 924.0),
    ("1973-01-01T00:00:00Z", 881.0),
    ("1973-02-01T00:00:00Z", 837.0),
    ("1973-03-01T00:00:00Z", 9000.0),
],
["timestamp", "value"],
).withColumn("group", lit("series1"))

# Run the Anomaly Detector service to look for irregular data

anomaly_detector = (
    SimpleDetectAnomalies()
    .setSubscriptionKey(anomaly_key)
    . setLocation(anomaly_loc)
    .setTimestampCol("timestamp")
    .setValueCol("value")
    .setOutputCol("anomalies")
    .setGroupbyCol("group")
    .setGranularity("monthly")
)

# Show the full results of the analysis with the anomalies marked as "True"

display(
    anomaly_detector.transform(df).select("timestamp", "value", "anomalies.isAnomaly")
)

```

Standard Session | [trySpark \(Python\)](#) | [Workspace Default](#) | [Data Wrangler](#) | [Copy](#)

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

Code **1** **Markdown**

3

```
1 # Create a dataframe with the point data that Anomaly Detector requires
2 df = spark.createDataFrame(
3     [
4         ("1972-01-01T00:00:00Z", 826.0),
5         ("1972-02-01T00:00:00Z", 799.0),
6         ("1972-03-01T00:00:00Z", 890.0),
7         ("1972-04-01T00:00:00Z", 900.0),
8         ("1972-05-01T00:00:00Z", 766.0),
9         ("1972-06-01T00:00:00Z", 805.0),
10        ("1972-07-01T00:00:00Z", 821.0),
11        ("1972-08-01T00:00:00Z", 20000.0),
12        ("1972-09-01T00:00:00Z", 883.0),
13        ("1972-10-01T00:00:00Z", 898.0),
14        ("1972-11-01T00:00:00Z", 957.0),
15        ("1972-12-01T00:00:00Z", 924.0),
16        ("1973-01-01T00:00:00Z", 881.0),
17        ("1973-02-01T00:00:00Z", 837.0),
18        ("1973-03-01T00:00:00Z", 9000.0),
19    ],
20    ["timestamp", "value"],
21 ).withColumn("group", lit("series1"))
22
23 # Run the Anomaly Detector service to look for irregular data
24 anomaly_detector = (
25     SimpleDetectAnomalies()
26     .setSubscriptionKey(anomaly_key)
27     .setLocation(anomaly_loc)
28     .setTimestampCol("timestamp")
29     .setValueCol("value")
30     .setOutputCol("anomalies")
31     .setGroupbyCol("group")
```

2

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```
35 # Show the full results of the analysis with the anomalies marked as "True"
36 display(
37     anomaly_detector.transform(df).select("timestamp", "value", "anomalies.isAnomaly")
38 )
[11] ✓ 4 sec - Command executed in 4 sec 23 ms by AOAI TFS 006041 on 4:36:45 PM, 7/08/24
```

> [Spark jobs \(2 of 2 succeeded\)](#) [Resources](#) [Log](#)

...

[Table](#) [Chart](#) [Download](#) Showing rows 1 - 15

	ABC timestamp	1.2 value	0/1 isAnomaly
1	1972-01-01T00:00:00Z	826.0	NULL
2	1972-02-01T00:00:00Z	799.0	NULL
3	1972-03-01T00:00:00Z	890.0	NULL
4	1972-04-01T00:00:00Z	900.0	NULL
5	1972-05-01T00:00:00Z	766.0	NULL
6	1972-06-01T00:00:00Z	805.0	NULL
7	1972-07-01T00:00:00Z	821.0	NULL
8	1972-08-01T00:00:00Z	20000.0	NULL
9	1972-09-01T00:00:00Z	883.0	NULL
10	1972-10-01T00:00:00Z	898.0	NULL
11	1972-11-01T00:00:00Z	957.0	NULL
12	1972-12-01T00:00:00Z	924.0	NULL
13	1973-01-01T00:00:00Z	881.0	NULL
14	1973-02-01T00:00:00Z	837.0	NULL
15	1973-03-01T00:00:00Z	9000.0	NULL

Task 11: Get information from arbitrary web APIs

- With HTTP on Spark, you can use any web service in your big data pipeline. The following code sample uses the [World Bank API](#) to get information about various countries around the world.

- 1. Use the **+ Code** icon below the cell output to add a new code cell to the notebook
- 2. Enter the following code in it. Click on **> Run cell** button and review the output

Copy

```
# Use any requests from the python requests library

def world_bank_request(country):

    return Request(
        "GET", "http://api.worldbank.org/v2/country/{}?format=json".format(country)
    )

# Create a dataframe with specifies which countries we want data on

df = spark.createDataFrame([("br",), ("usa",)], ["country"]).withColumn(
    "request", http_udf(world_bank_request)(col("country"))
)

# Much faster for big data because of the concurrency :)

client = (
    HTTPTransformer().setConcurrency(3).setInputCol("request").setOutputCol("response")
)

# Get the body of the response

def get_response_body(resp):

    return resp.entity.content.decode()

# Show the details of the country data returned

display(
    client.transform(df).select(
        "country", udf(get_response_body)(col("response")).alias("response")
    )
)
```

Standard session ▾ | PySpark (Python) ▾ | Workspace default ▾ | Data Wrangler ▾ | Copilot

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

Code **Markdown**

3

```

1 # Use any requests from the python requests library
2
3
4 def world_bank_request(country):
5     return Request(
6         "GET", "http://api.worldbank.org/v2/country/{}?format=json".format(country)
7     )
8
9
10 # Create a dataframe with specifies which countries we want data on
11 df = spark.createDataFrame([("br",), ("usa",)], ["country"]).withColumn(
12     "request", http_udf(world_bank_request)(col("country")))
13 )
14
15 # Much faster for big data because of the concurrency :)
16 client = (
17     HTTPTransformer().setConcurrency(3).setInputCol("request").setOutputCol("response")
18 )
19
20 # Get the body of the response
21
22
23 def get_response_body(resp):
24     return resp.entity.content.decode()
25
26
27 # Show the details of the country data returned
28 display(
29     client.transform(df).select(
30         "country", udf(get_response_body)(col("response")).alias("response"))
31     )
32 )

```

2

Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.

```

18 )
19
20 # Get the body of the response
21
22
23 def get_response_body(resp):
24     return resp.entity.content.decode()
25
26
27 # Show the details of the country data returned
28 display(
29     client.transform(df).select(
30         "country", udf(get_response_body)(col("response")).alias("response"))
31     )
32 )

```

[12] ✓ 9 sec - Command executed in 8 sec 868 ms by AOAI TFS 006041 on 4:43:18 PM, 7/08/24

PySpa

> Spark jobs (3 of 3 succeeded) Resources Log

Table Chart Download Showing rows 1 - 2

	ABC country	ABC response
1	br	[{"page":1,"pages":1,"per_page":50,"total":1},[{"id":"BRA","iso2Code":"BR","name":"Brazil","region":[{"id":"LCN","iso2Code":"ZJ","value":"Latin America & Caribbean"}]]]
2	usa	[{"page":1,"pages":1,"per_page":50,"total":1},[{"id":"USA","iso2Code":"US","name":"United States","region":[{"id":"NAC","iso2Code":"XU","value":"North America"}]]]

Inspect Search

3. To save notebook, select the drop down for **N****otebook1** and enter **Use Azure AI services with SynapseML******.

The screenshot shows the Azure AI services with SynapseML notebook interface. The top navigation bar includes back, forward, and refresh buttons, and the URL app.fabric.microsoft.com/groups/79824608-a1fe-4f7a-a6c1-e4fc314. The title bar says "Notebook 1 | Saved" with a red box around it and a red number "1" indicating a save operation. The left sidebar has icons for Home, Create, Browse, OneLake data hub, Monitor, Real-Time hub, Workspaces, and AI-Fabric-789, with "AI-Fabric-789" selected. The main content area shows a "Name" field with "Use Azure AI services with SynapseML" (red box), a "Location" field with "AI-Fabric-789", an "Owner" field with "AOAI TFS 006041", a "Save status" field showing "Last saved: 7/8/2024, 4:48:43 PM", a "Description" field with "New notebook", and a "Show more" link. On the right, there's a code editor with Python code for PySpark, a log section showing a successful Apache Spark session read, and another code editor section below.

```
from pyspark.sql import *
from synapse.ml.io import *
from requests import *
from pyspark.sql import *
from pyspark.ml import *
from pyspark.sql import *
import os
```

[1] ✓ 14 sec - Apache Spark session ready

> Log

```
from pyspark.sql import *
from synapse.ml.core import *
```

Task 12: Delete the resources

- ① To avoid incurring unnecessary Azure costs, you should delete the resources you created in this quickstart if they're no longer needed. To manage resources, you can use the [Azure portal](#).
- 1. To delete the storage account, navigate to **Azure portal Home** page, click on **Resource groups**.

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and several navigation icons. Below the header, there's a section titled "Azure services" with various service icons: Create a resource, Resource groups (which is highlighted with a red box), Azure OpenAI, Quickstart Center, Virtual machines, App Services, Storage accounts, and SQL databases. Below this, there's a "More services" button. The main area is titled "Resources" and has tabs for "Recent" and "Favorite". Under "Recent", there are two items: "aoai210705_RG" (Resource group, last viewed 10 minutes ago) and "Azure-openai-test90" (Azure OpenAI, last viewed 17 hours ago). There's also a "See all" link. At the bottom of the resources section, there are links for "Subscriptions", "Resource groups" (with a red box around it), and "All resources".

2. Click on the assigned resource group.

The screenshot shows the "Resource groups" page in Microsoft Azure. The top navigation bar includes "Home >" and the title "Resource groups". The page shows a single record: "Contoso" under "Subscription equals all" and "Location equals all". The table has columns for "Name" and "Actions". One row is visible, containing a checkbox, the name "AI-FabricRG89", and a delete icon. The "AI-FabricRG89" entry is highlighted with a red box.

3. In the **Resource group** home page, select the **delete resource group**

Microsoft Azure

Home > Resource groups >

AI-FabricRG89

Resource group

Search Create Manage view Delete resource group Refresh

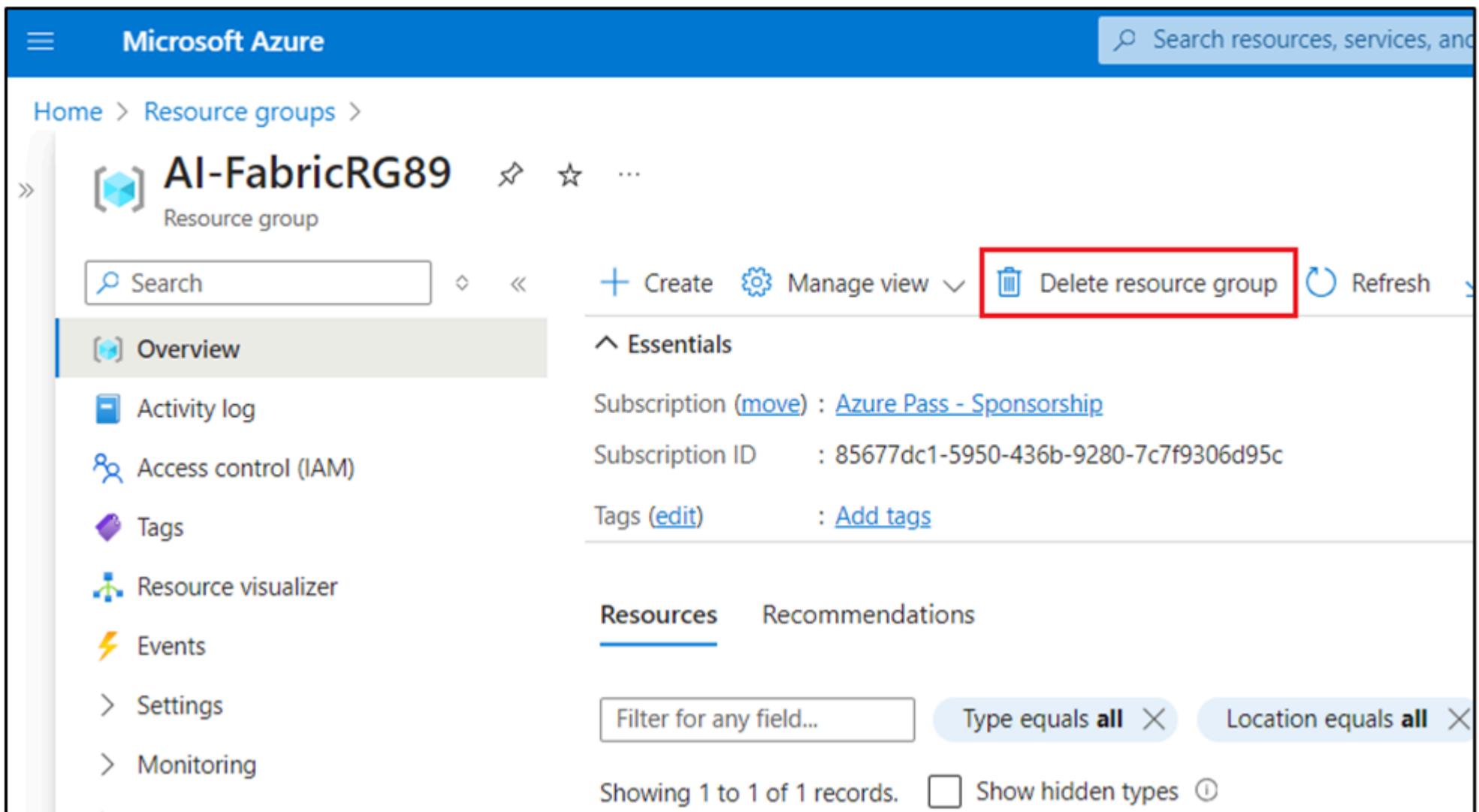
Overview Activity log Access control (IAM) Tags Resource visualizer Events Settings Monitoring

Subscription (move) : Azure Pass - Sponsorship
Subscription ID : 85677dc1-5950-436b-9280-7c7f9306d95c
Tags (edit) : Add tags

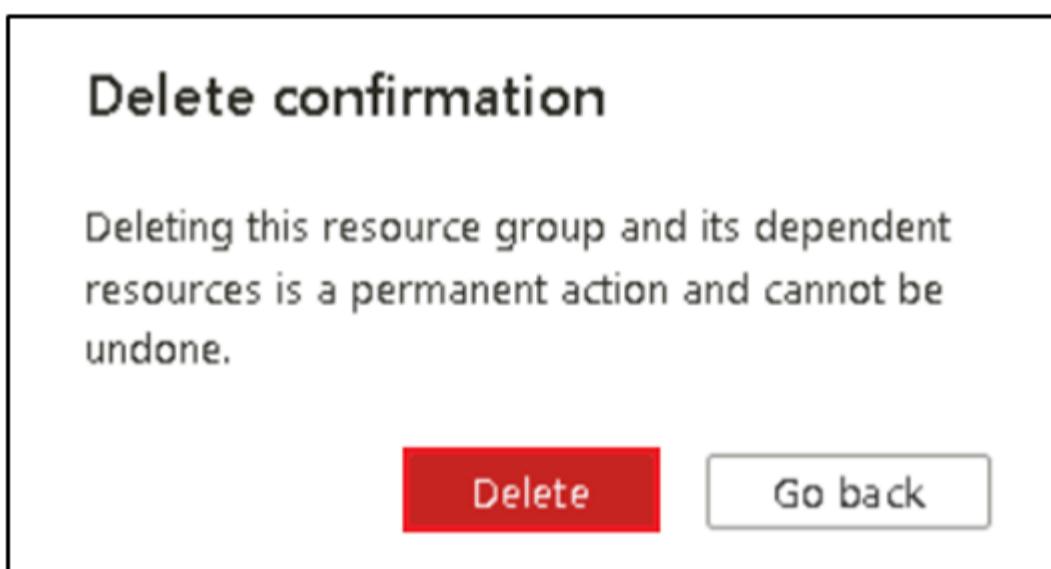
Resources Recommendations

Filter for any field... Type equals all Location equals all

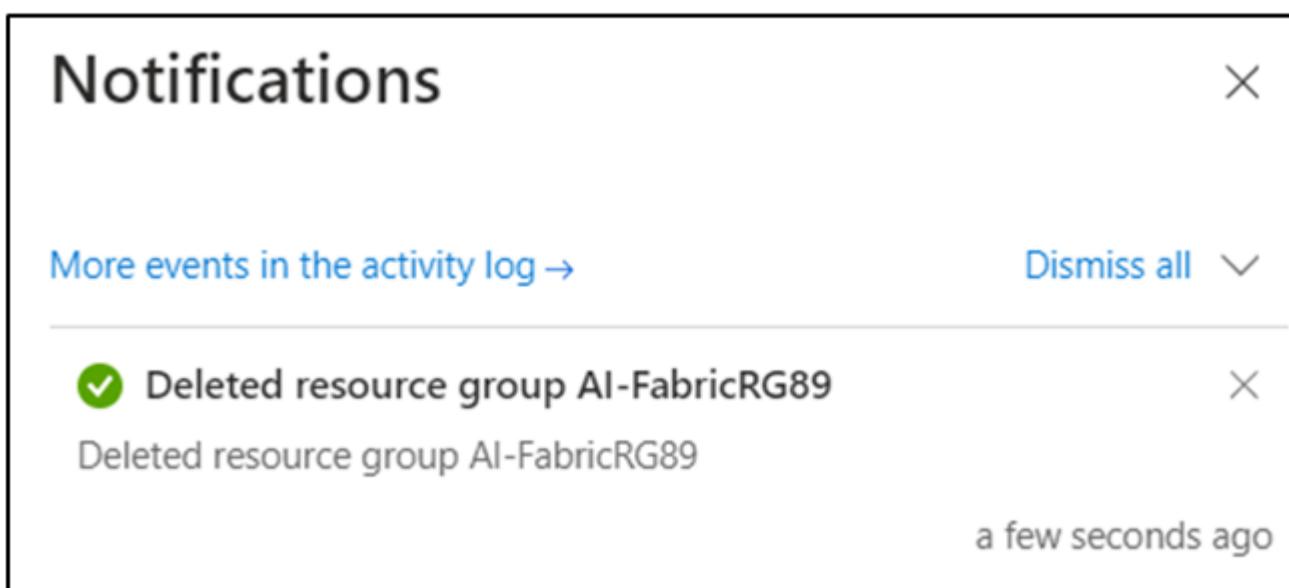
Showing 1 to 1 of 1 records. Show hidden types



- 4. In the **Delete Resources** pane that appears on the right side, navigate to **Enter "resource group name" to confirm deletion** field, then click on the **Delete** button.
- 5. On **Delete confirmation** dialog box, click on **Delete** button.



- 6. Click on the bell icon, you'll see the notification --**Deleted resource group AOAI-RG8**



- 7. Open your browser, navigate to the address bar, and type or paste the following URL: `++https://app.fabric.microsoft.com/++` then press the **Enter** button.

Workspaces

Search

My workspace

All

AI-Fabric-675

Workspaces

AI-Fabric-675

...

Deployment pipelines

+ New workspace

8. Select the ... option under the workspace name and select **Workspace settings**.

Name	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in a
AI_Fabric_Lakehouse78	Lakehouse	—	AOAI TFS 006...	—	—	—	—	
AI_Fabric_Lakehouse78	Semantic model	—	AI-Fabric-675	7/15/24, 10:50:...	N/A	—	—	
AI_Fabric_Lakehouse78	SQL analytics	—	AI-Fabric-675	—	N/A	—	—	

9. Select **General **and click on **Remove this workspace**.

Workspace settings

General

- License info
- Azure connections
- System storage
- Git integration
- OneLake
- Workspace identity
- Network security

Power BI

- Data
- Engineering/Science
- Data Factory

Notifications

Workspace contacts

Add the people you'll collaborate with in this workspace. Workspace contacts will get notifications about issues affecting the workspace. You must add at least one person.

AT AOA TFS 006351 X

OneDrive

Workspace OneDrive

Enter groups

Manage

Delete workspace

This workspace will be permanently deleted after a retention period of 7 days, as defined by your Fabric administrator. Contact your Fabric administrator if you need to restore the workspace and recover the items in it during this retention period.

 Remove this workspace

10. Click on **Delete** in the warning that pops up.

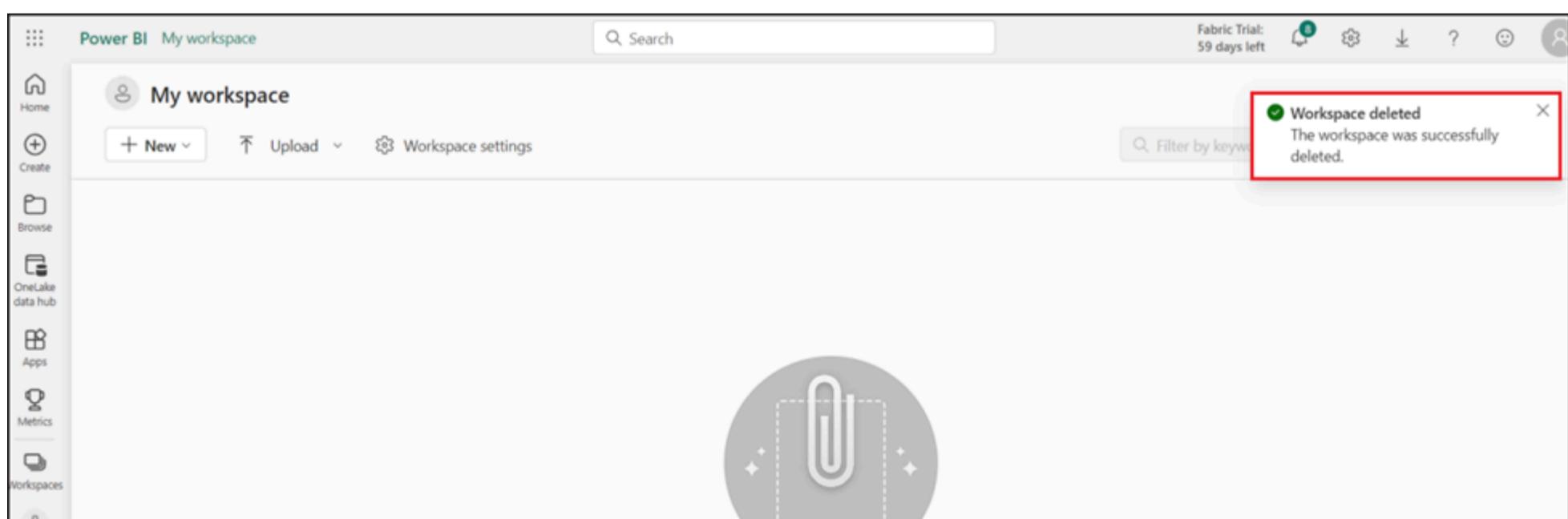
Delete workspace?

You're about to delete this workspace. Are you sure you want to continue?

Delete

Cancel

11. Wait for a notification that the Workspace has been deleted, before proceeding to the next lab.



The screenshot shows the Power BI 'My workspace' interface. The top navigation bar includes 'Power BI' and 'My workspace'. The left sidebar has options like 'Home', 'Create', 'Browse', 'OneLake data hub', 'Apps', 'Metrics', and 'Workspaces'. The main area is titled 'My workspace' and contains buttons for '+ New', 'Upload', and 'Workspace settings'. A search bar and a 'Fabric Trial: 59 days left' status are also present. A prominent message box in the bottom right corner states: 'Workspace deleted. The workspace was successfully deleted.' with a checkmark icon. The entire message box is enclosed in a red border.