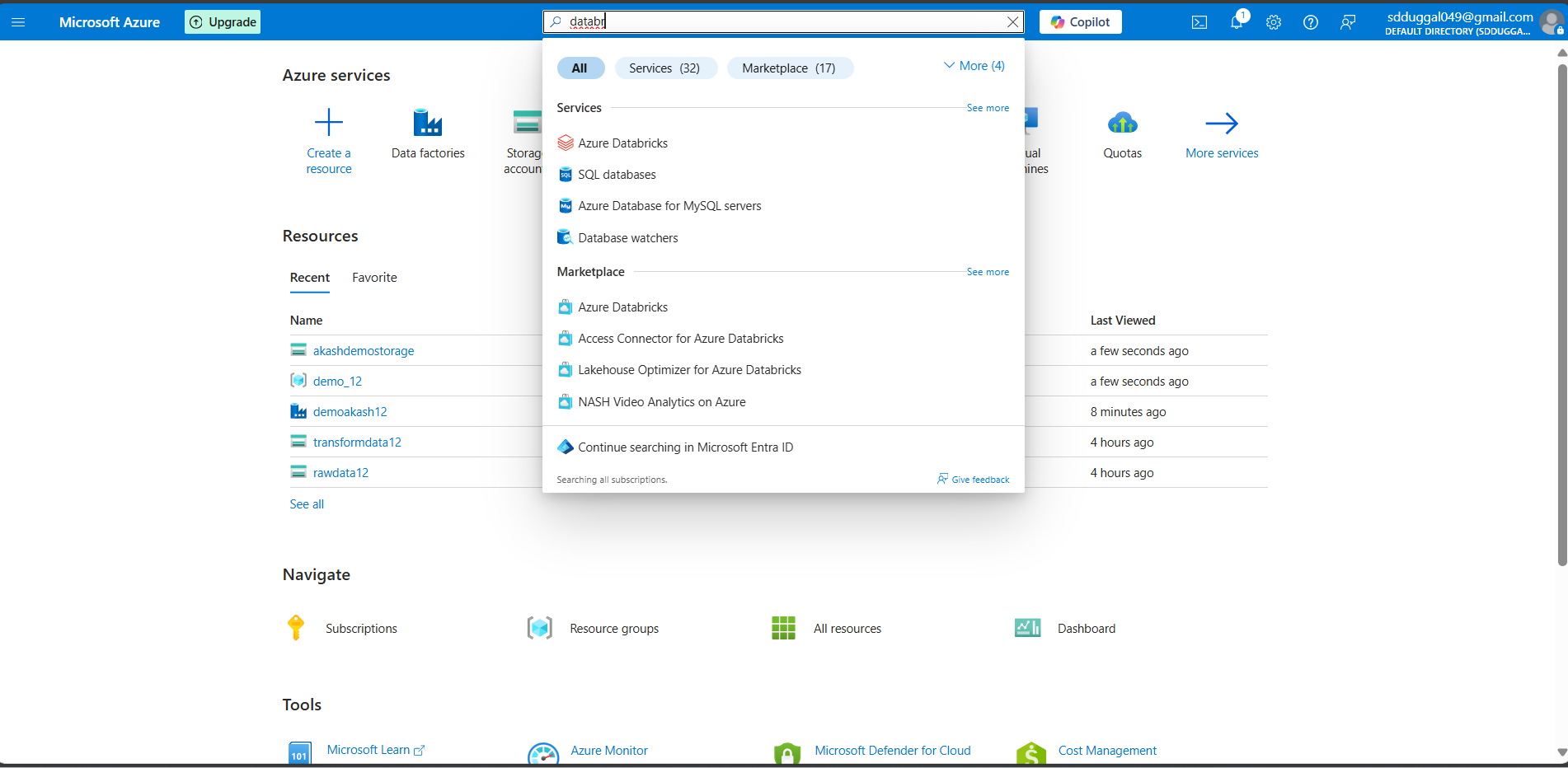
Will create a Databricks service for the transformation

1st we will create a Databricks service

Search for Databricks



Click on create databricks service

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

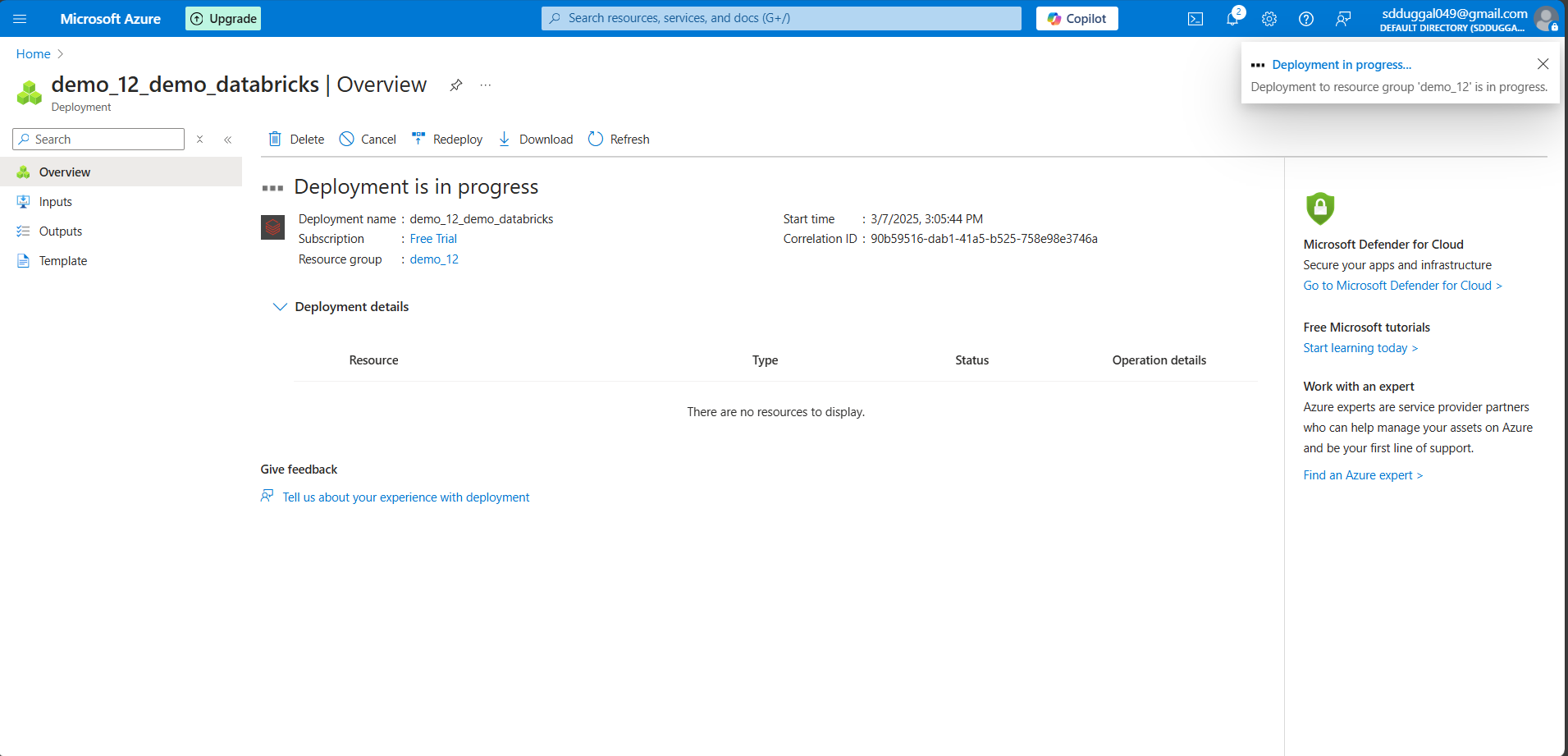
Just select the

Resource group and name the workspace and select region

Same as we have done in last 2 services

Click on review + create

It will take time



Once deployment is done

A screenshot of a computer

AI-generated content may be incorrect.

Click on Go to resource

A screenshot of a computer

AI-generated content may be incorrect.

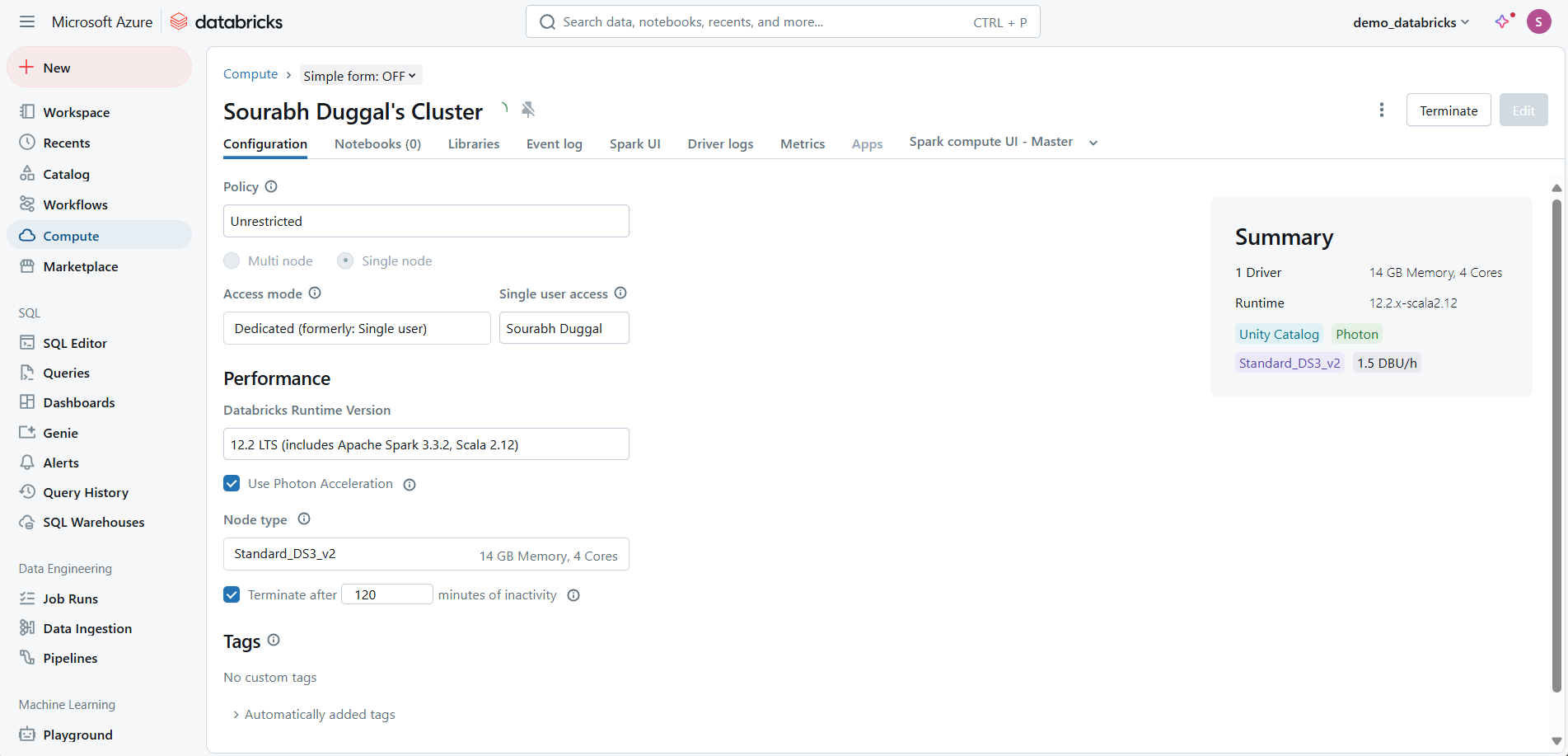
Click on launch resource

You will be redirected to Databricks workspace

Now we have to create a compute for running our spark code

A screenshot of a computer

AI-generated content may be incorrect.

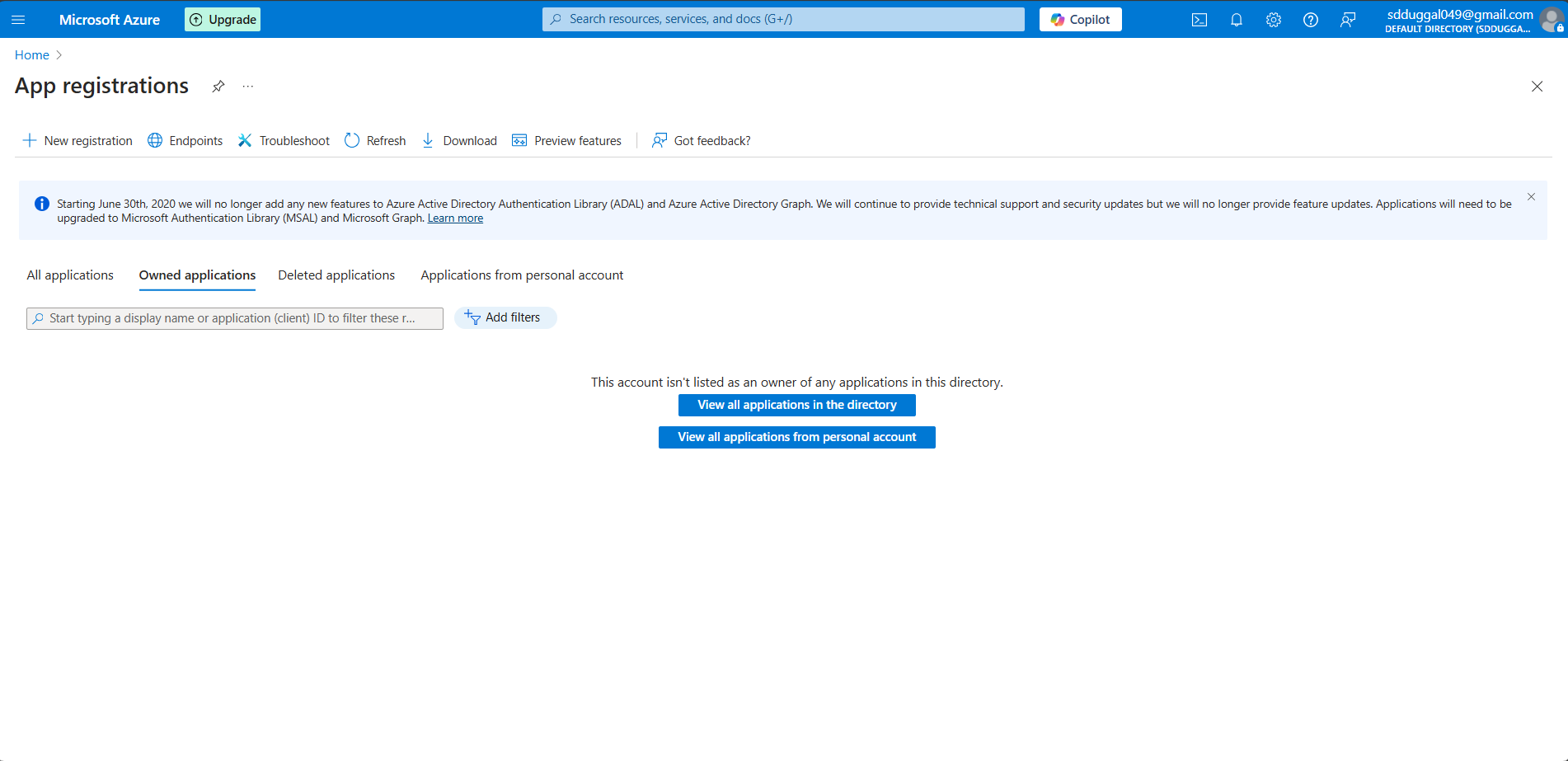


we have created the compute   
now we will open the notebook in that and will do basic transformations

Create a connection between storage to Databricks notebook

We have to mount the storage to the notebook

For that we have to create connection through application



Give the name and register the app

A screenshot of a computer

AI-generated content may be incorrect.

Once its register we will land up on this screen

A computer screen shot of a computer

AI-generated content may be incorrect.

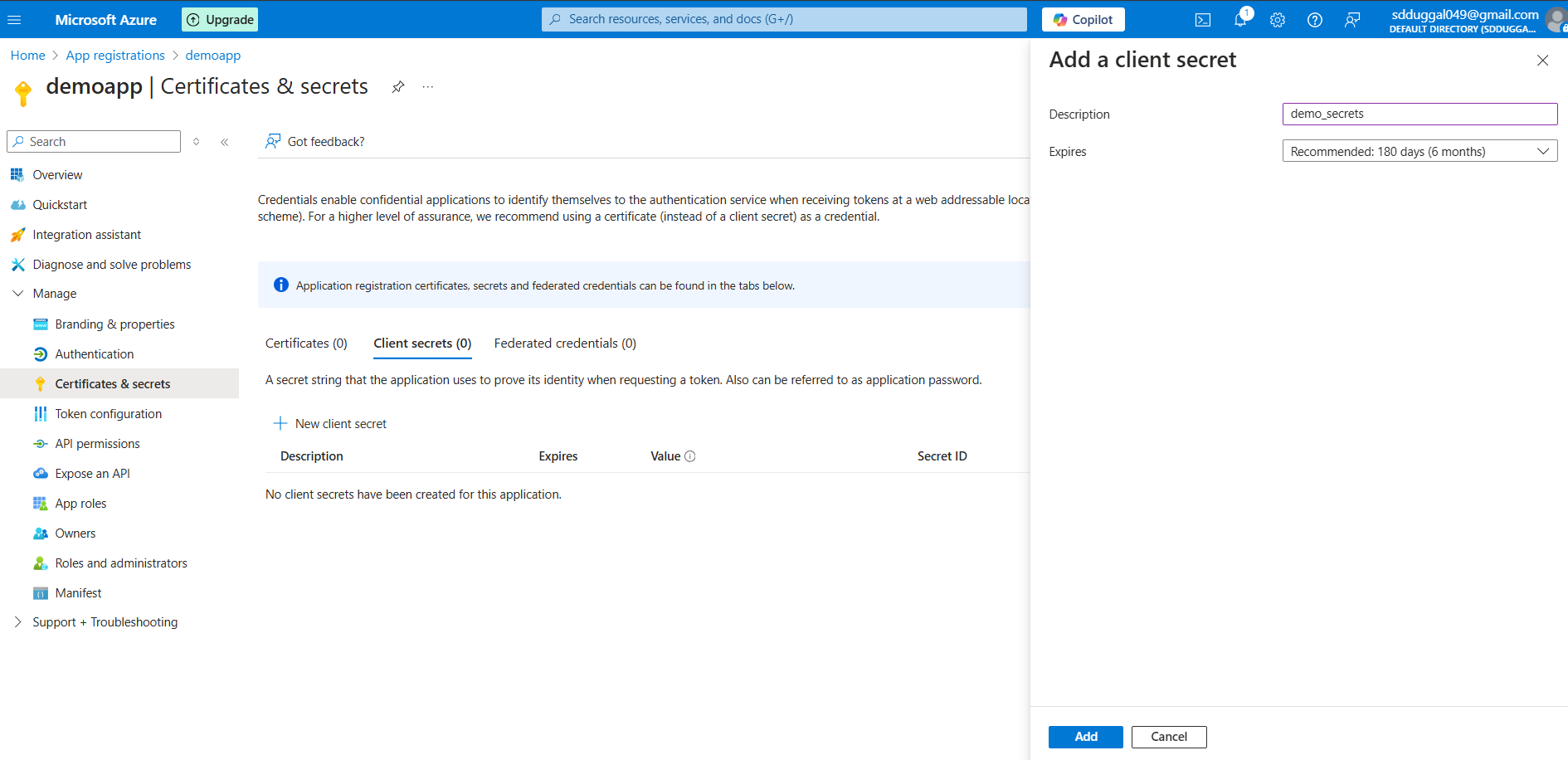
From this screen we need to things

One is

ClientID

TennantID

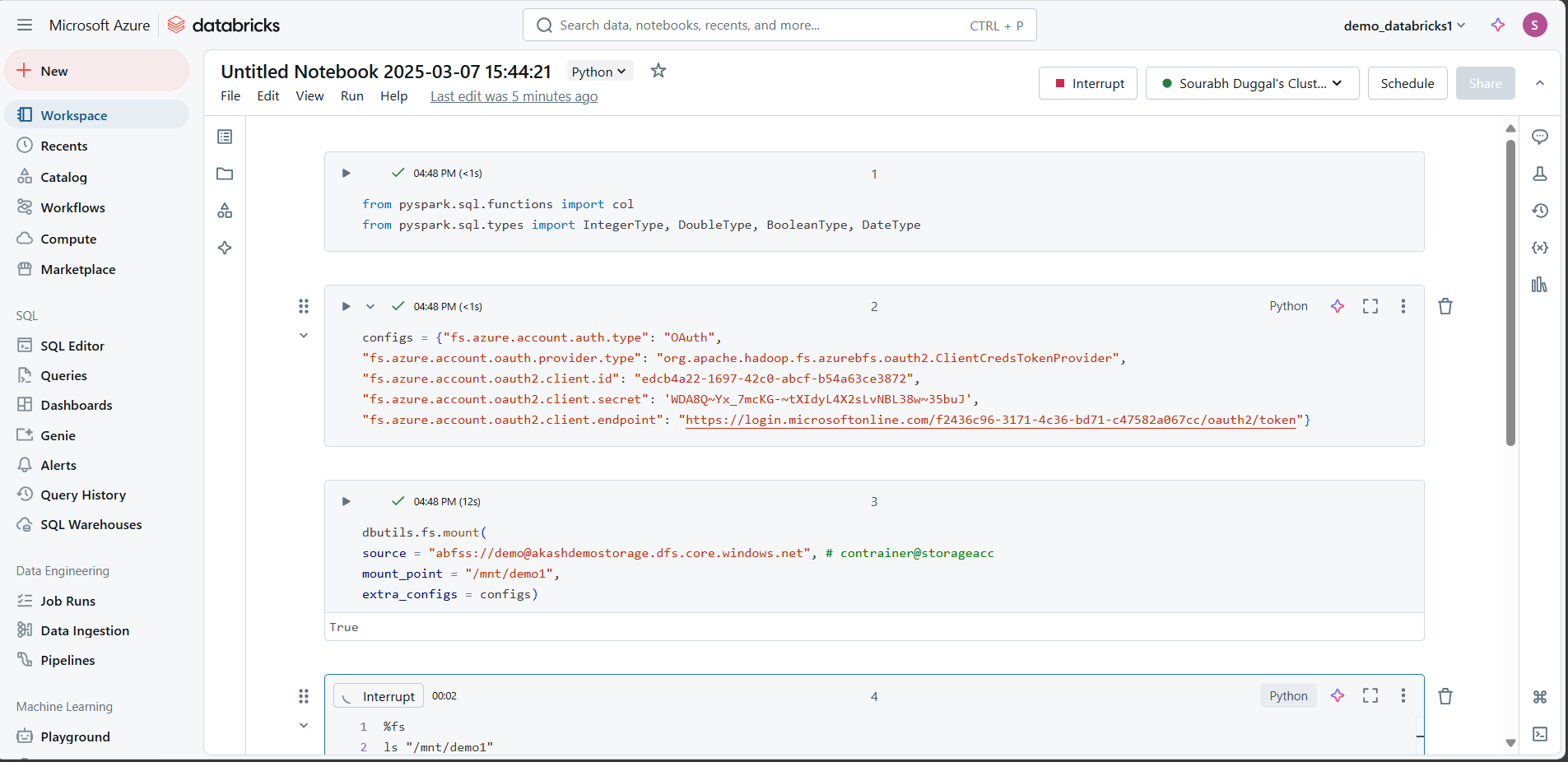
Now we have to create client secrets



Click on the add   
client secrete will gets added

Copy that information as well

Now open the notebook



Execute this first

from pyspark.sql.functions import col

from pyspark.sql.types import IntegerType, DoubleType, BooleanType, DateType

then execute this

configs = {"fs.azure.account.auth.type": "OAuth",

"fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",

"fs.azure.account.oauth2.client.id": "",

"fs.azure.account.oauth2.client.secret": '',

"fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/tanent\_id/oauth2/token"}

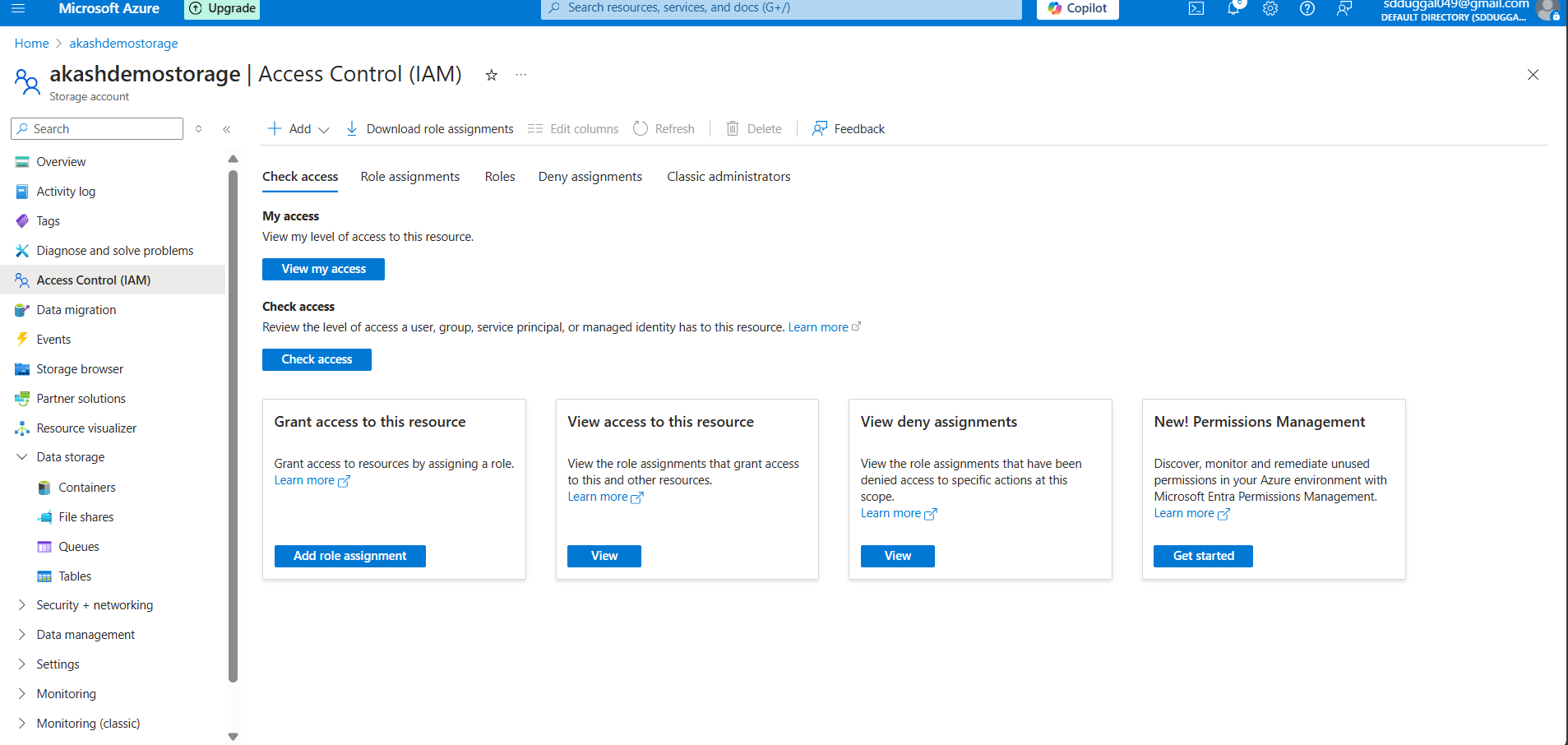
dbutils.fs.mount(

source = "abfss:// *contrainer* @ *storageacc*.dfs.core.windows.net", *# contrainer@storageacc*

mount\_point = "/mnt/demo ",

extra\_configs = configs)

Before executing this we have to give permission to app to access storage account



Give storage blob data contributor access

A screenshot of a computer

AI-generated content may be incorrect.

Select members

A screenshot of a computer

AI-generated content may be incorrect.

Now app has access to read/write this storage account

Once its done now you can run this

%fs

ls "/mnt/demo1"

A screenshot of a computer

AI-generated content may be incorrect.

Will start with basics transformation and will store this data into the storage account