Mounting ADLS in Azure Databricks (Access ADLS files from Azure Databricks) – Using Service Principal:

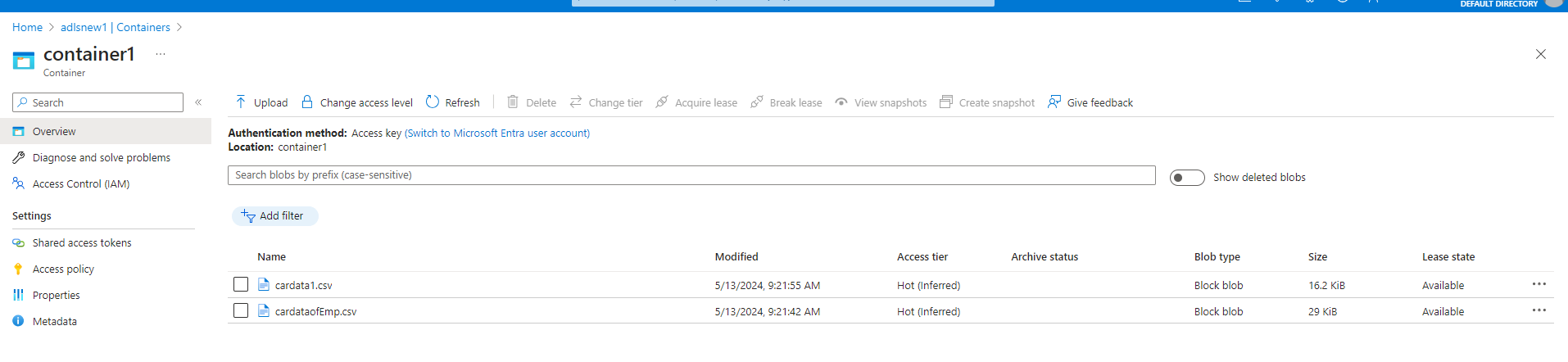
Different ways to access are:

1. Service principal – Microsoft recommends this method due to efficiency.
2. SAS key
3. Account access keys

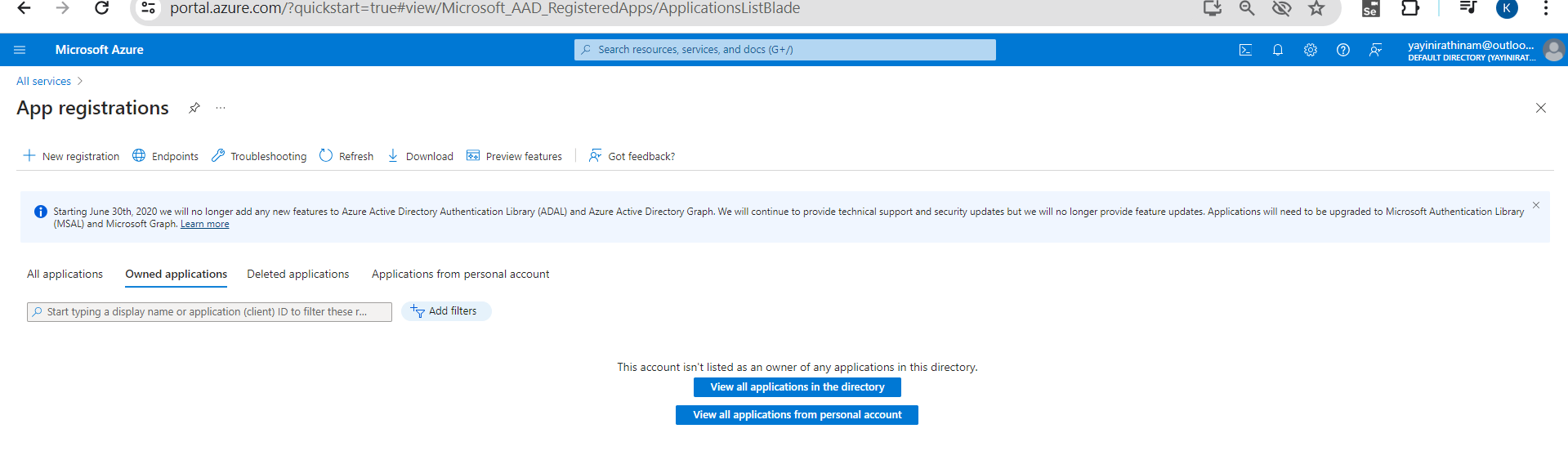
Steps to do that:

1. Create App/ service Principal – App registration – Create a pointer to Datalake from databricks to access files
2. Store Key vault
3. Create Scope
4. Mount DL

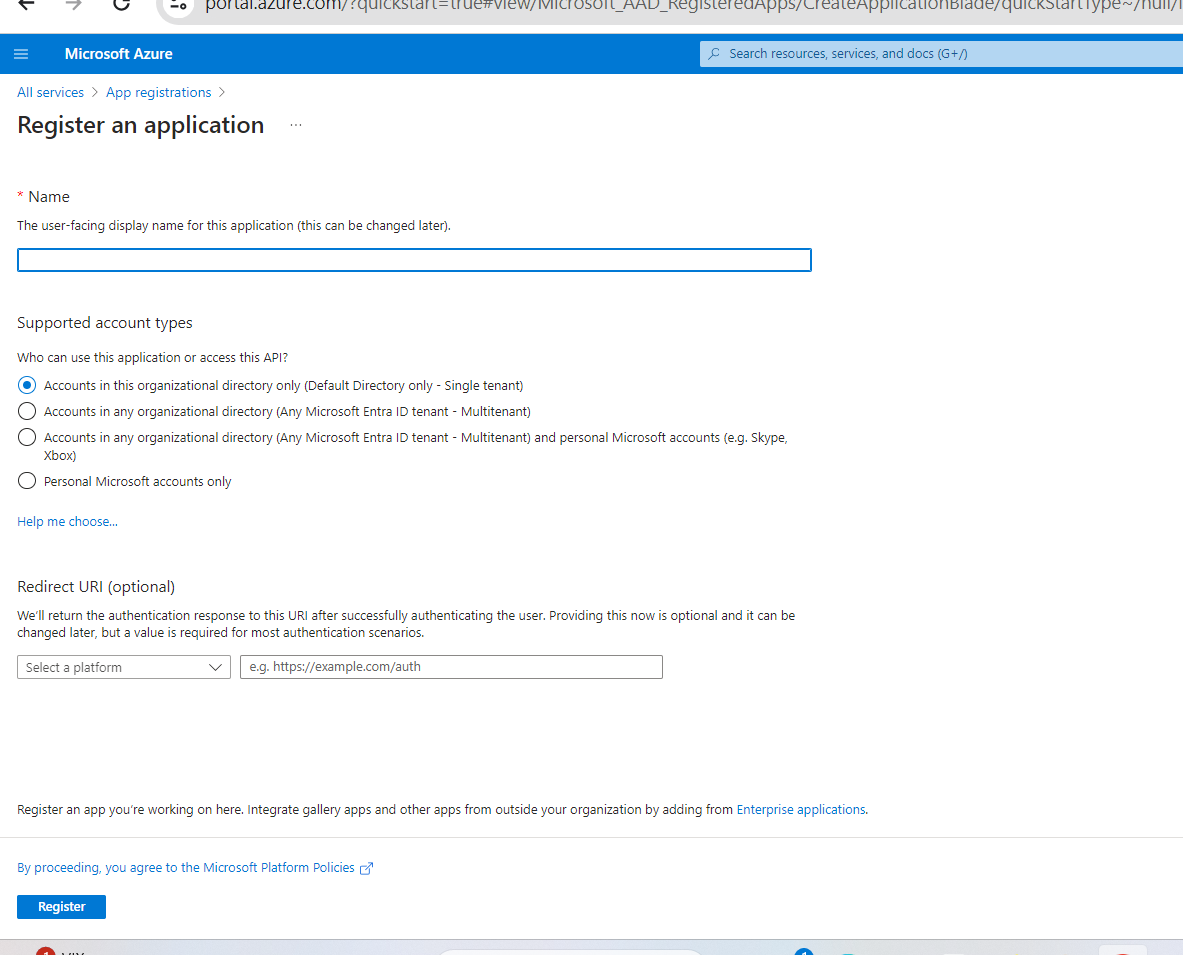
**Created ADLS and upload files:**



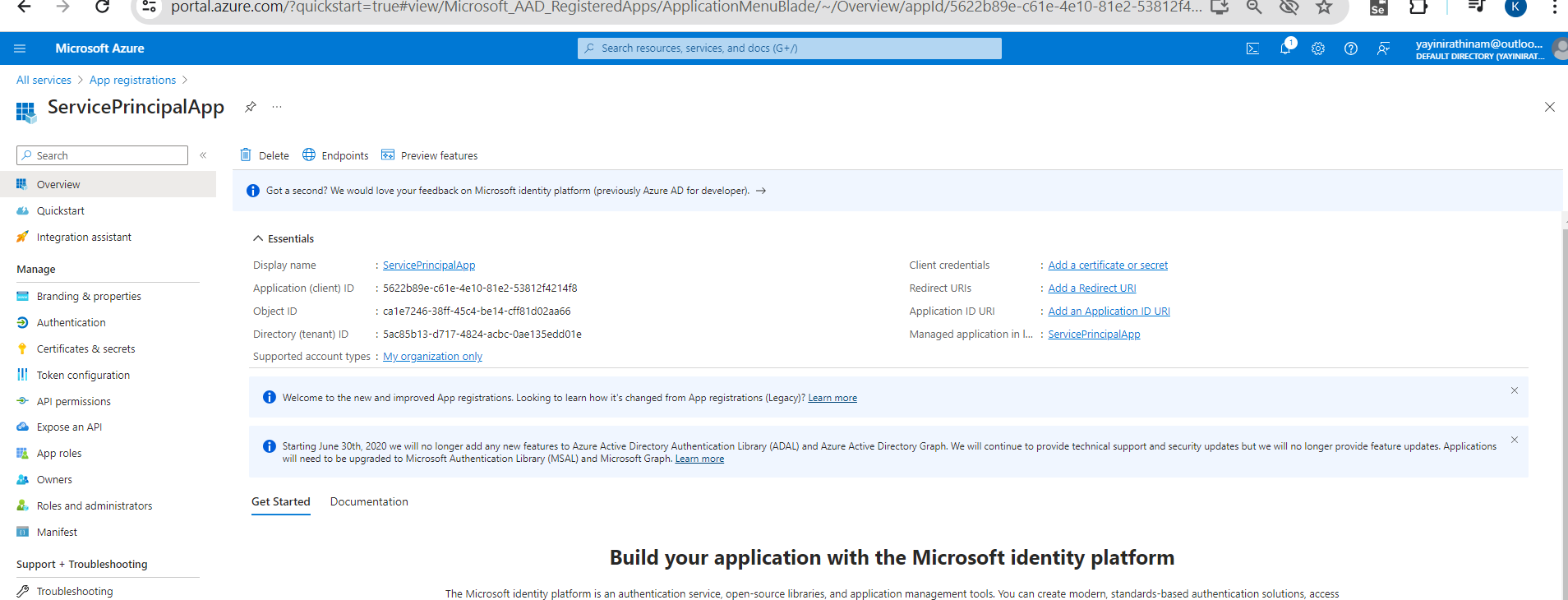
**Search for App registration in the home page:**



**New Registration:**

****

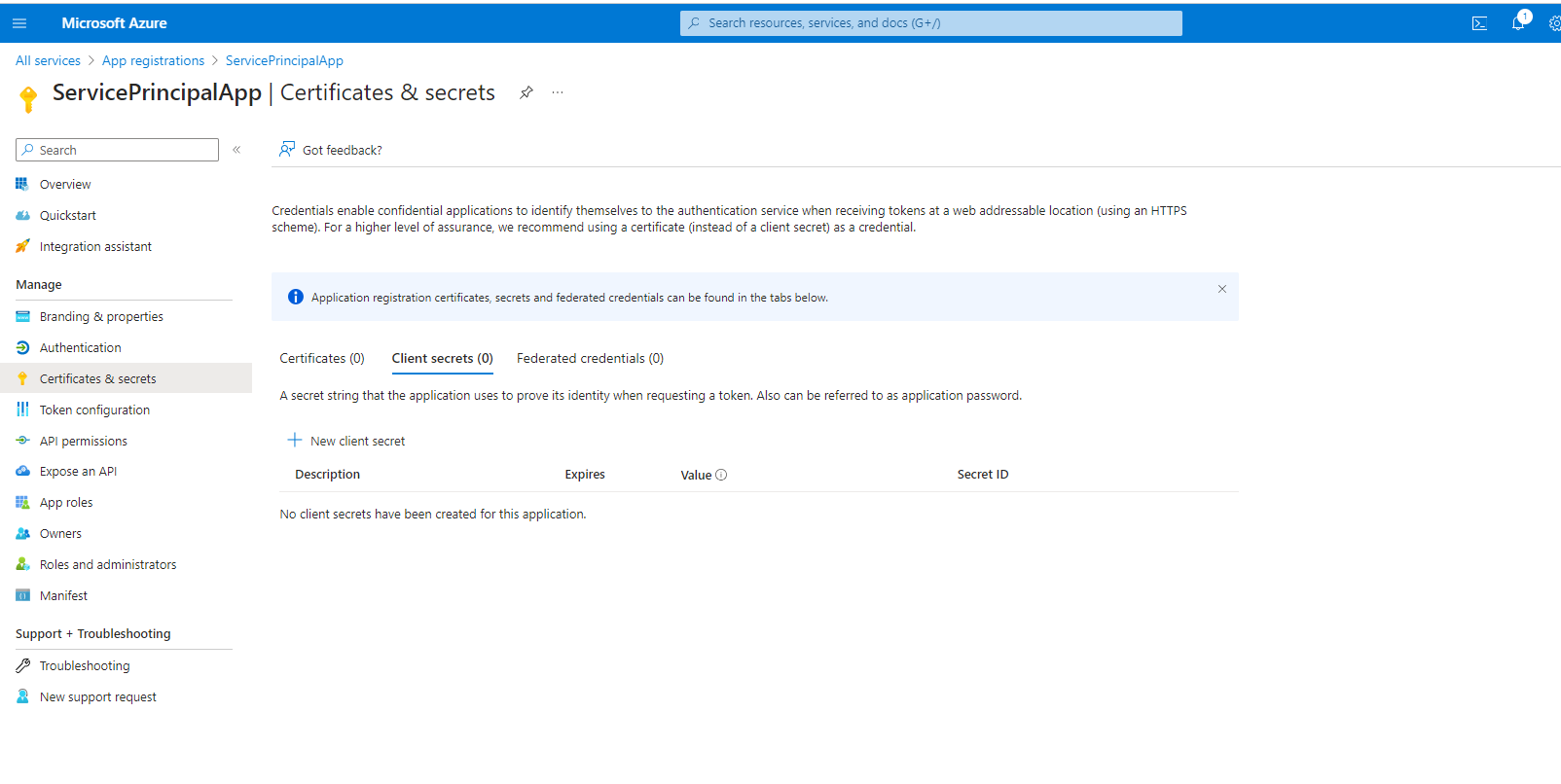
Created the ServicePrincipal/ Application:

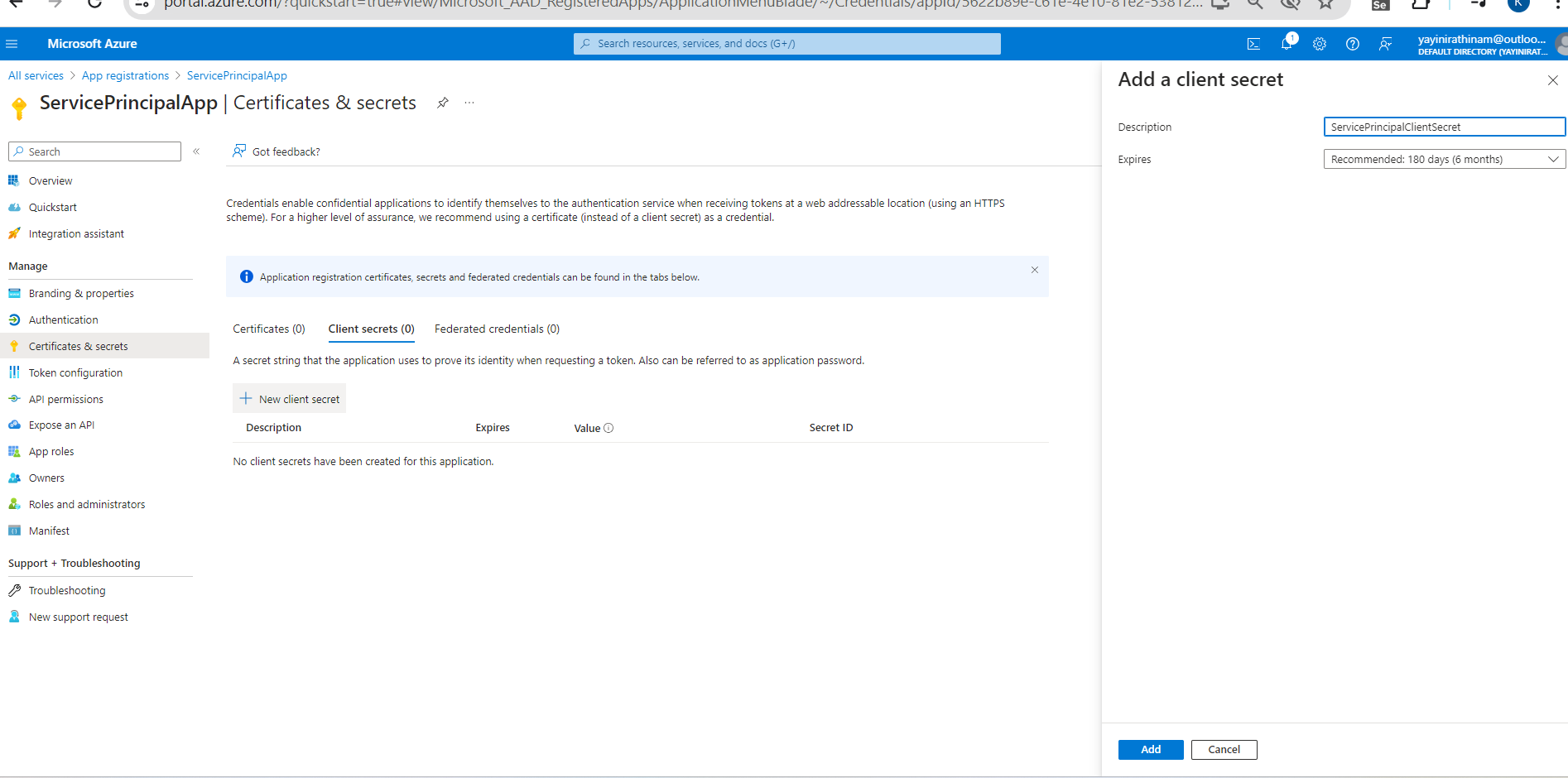


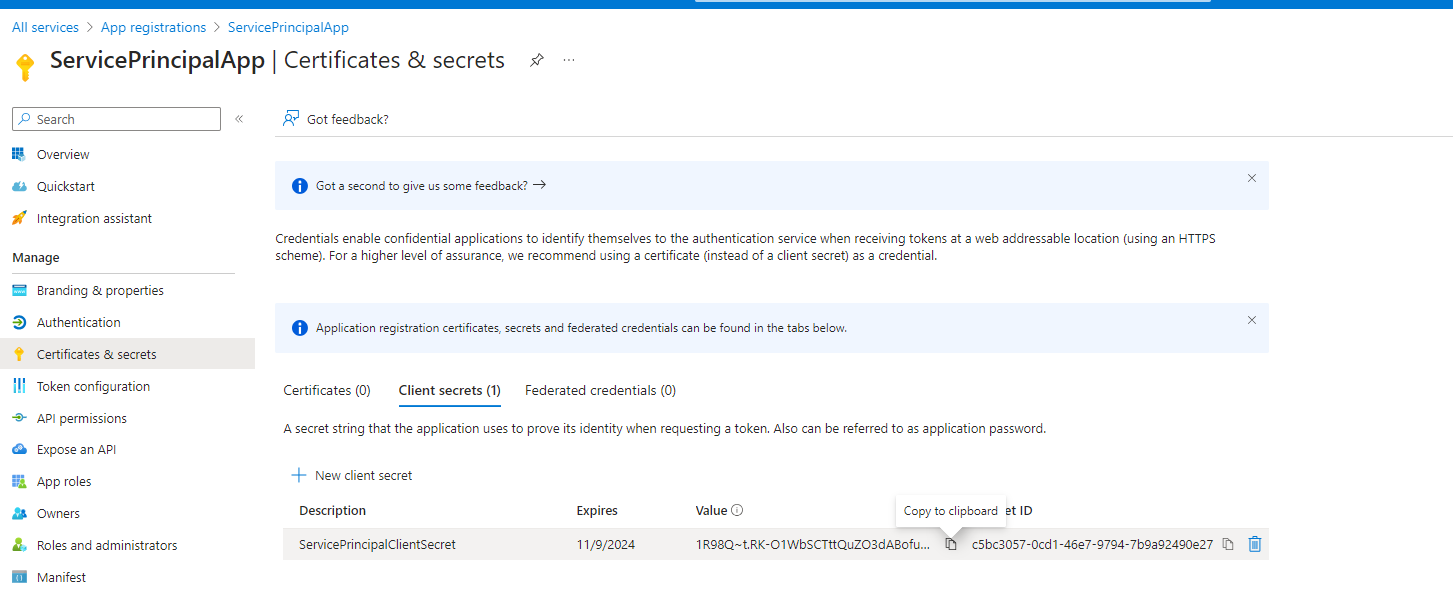
Copy Client Id, Tenant Id:

Client ID : 5622b89e-c61e-4e10-81e2-53812f4214f8

Tenant Id: 5ac85b13-d717-4824-acbc-0ae135edd01e







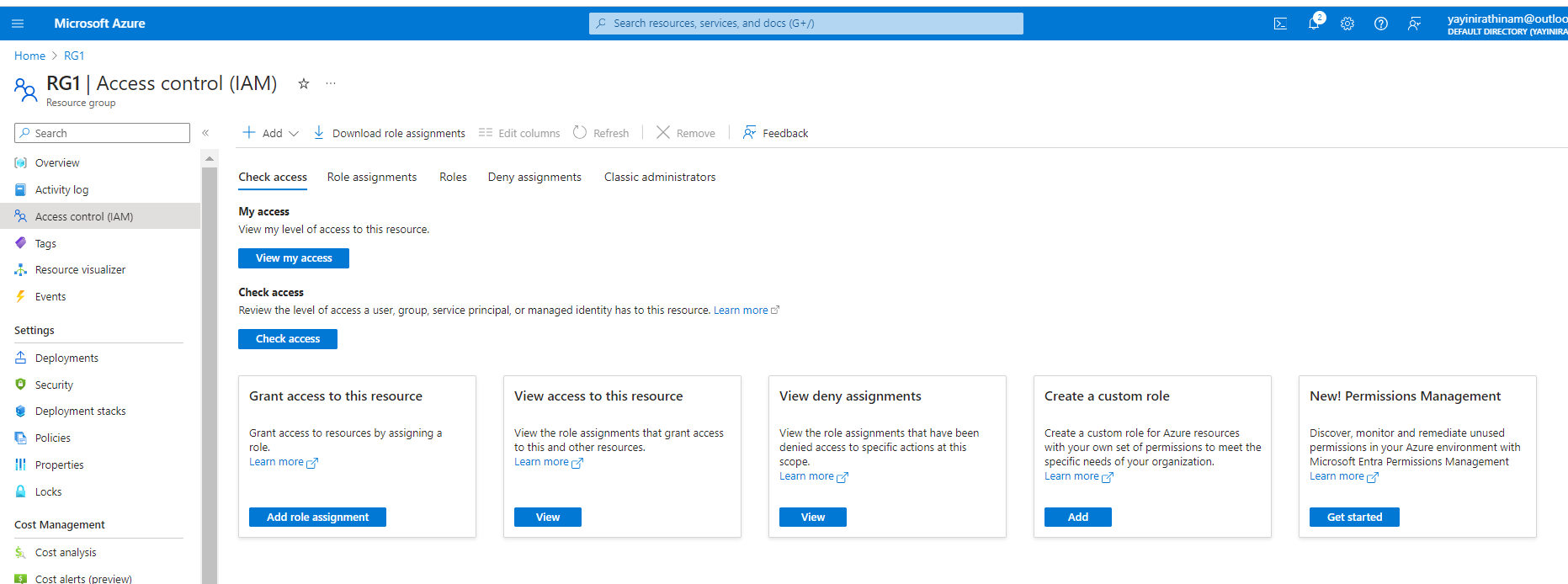
Make a note of the Client Secret Key:

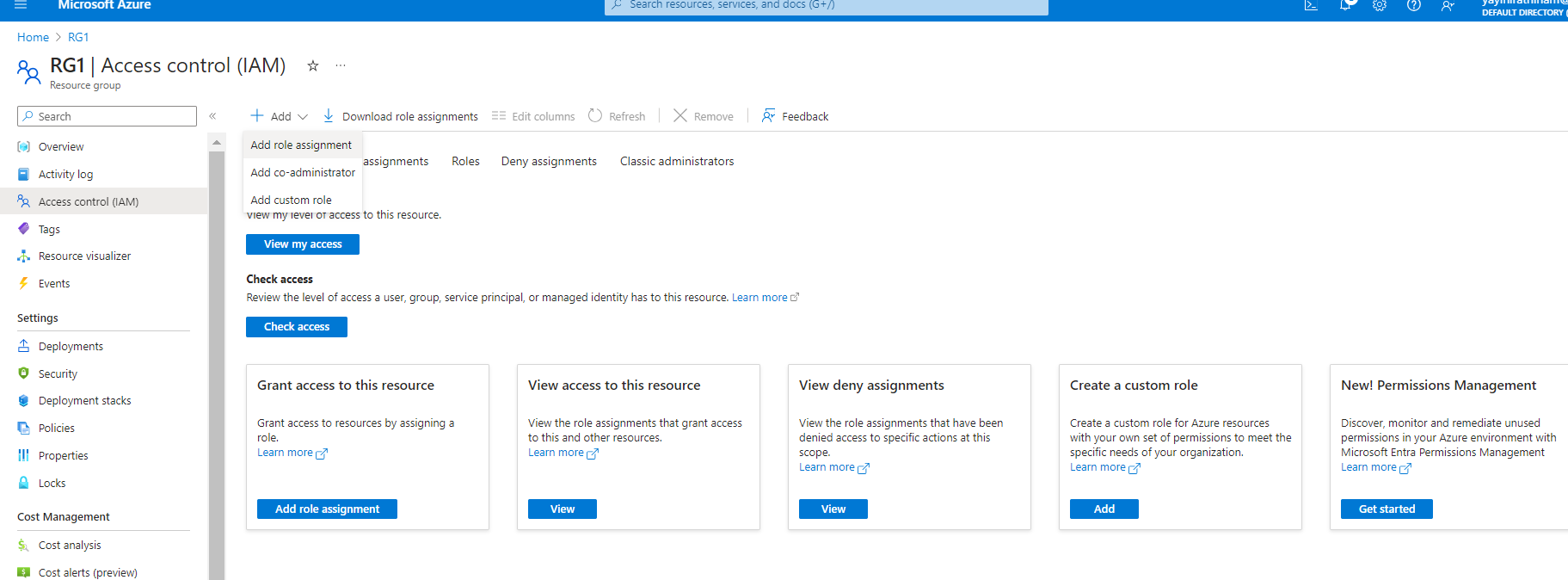
Client Secret key:

1R98Q~t.RK-O1WbSCTttQuZO3dABofumIGHU5aiM

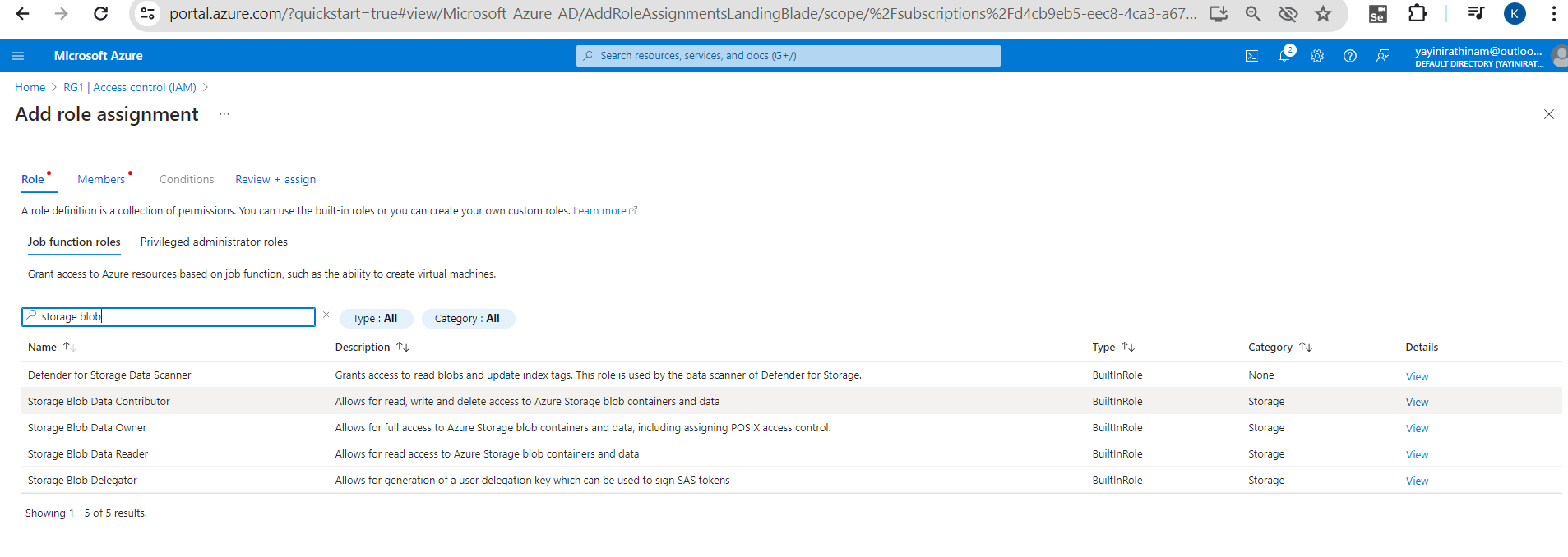
1R98Q~t.RK-O1WbSCTttQuZO3dABofumIGHU5aiM

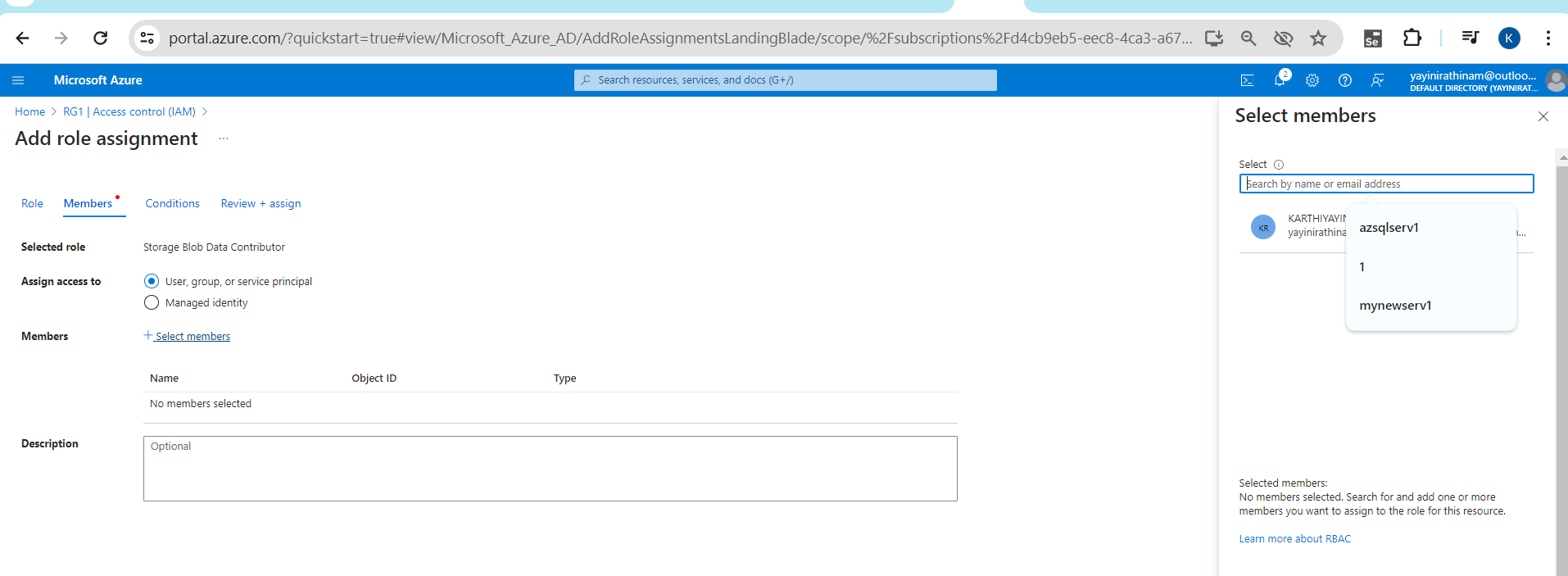
Go to resource group where the ADLS is present and create IAM access to provide access for the service principal to access ADLS.



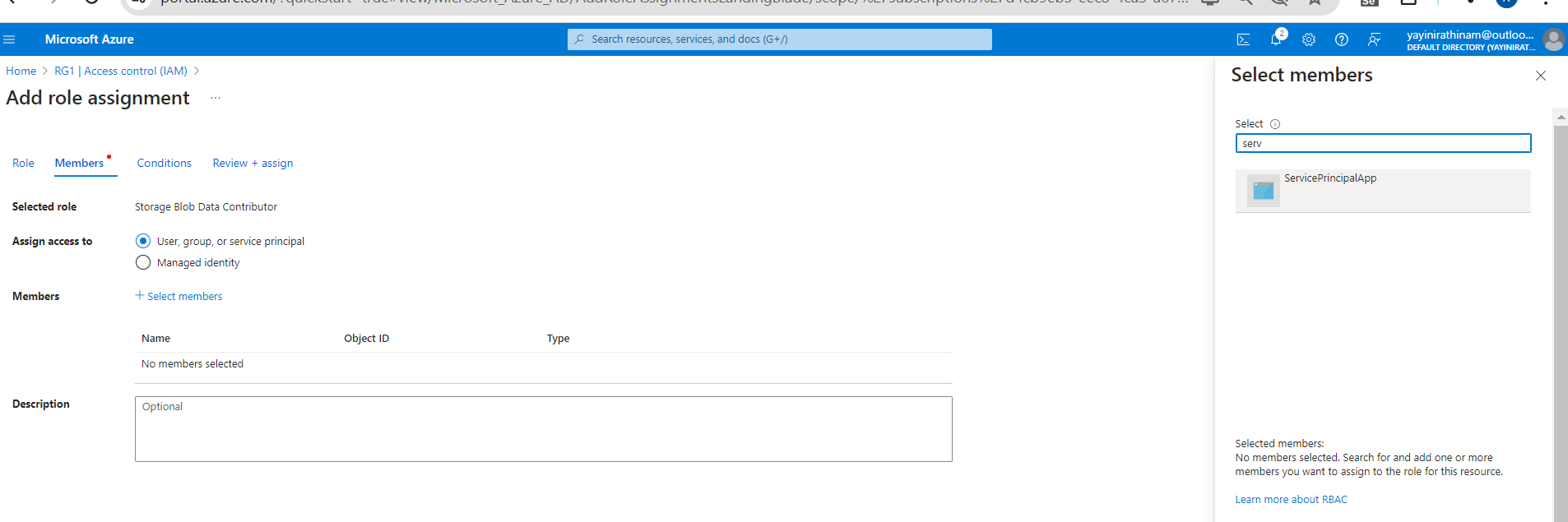


Search for Storage Contributor

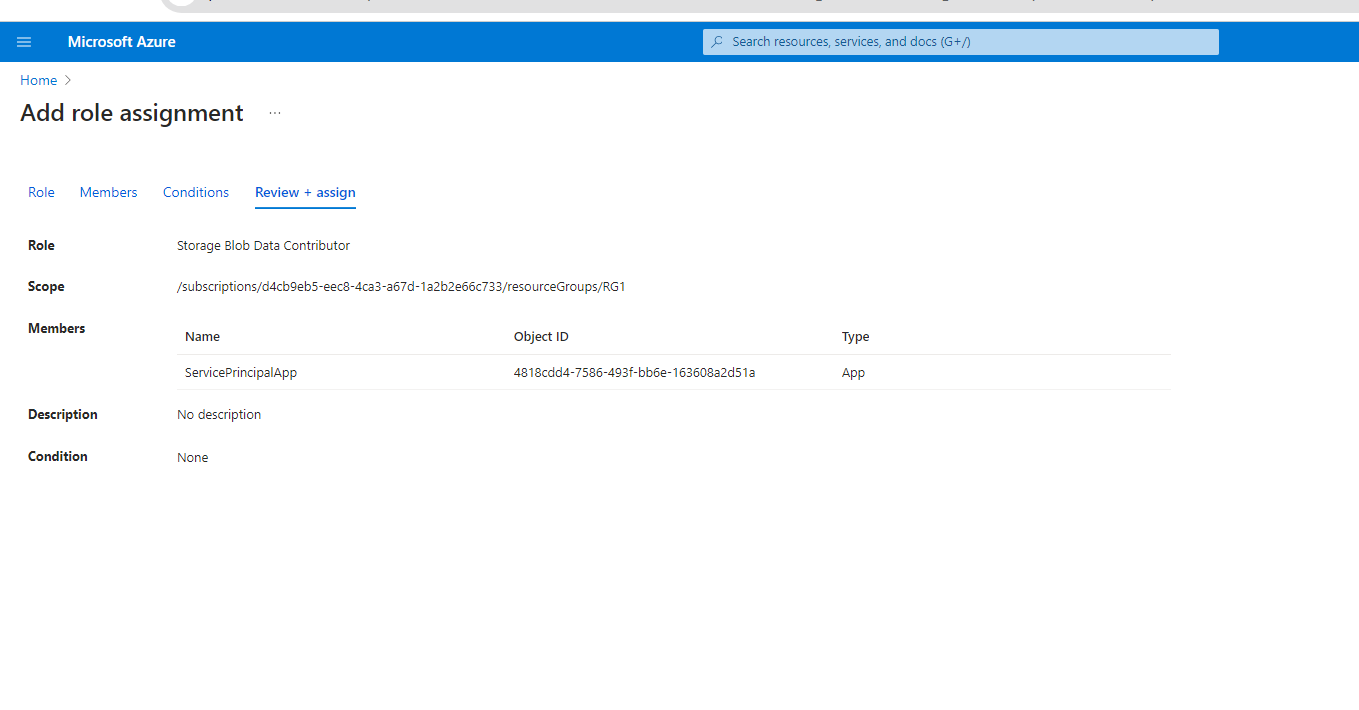




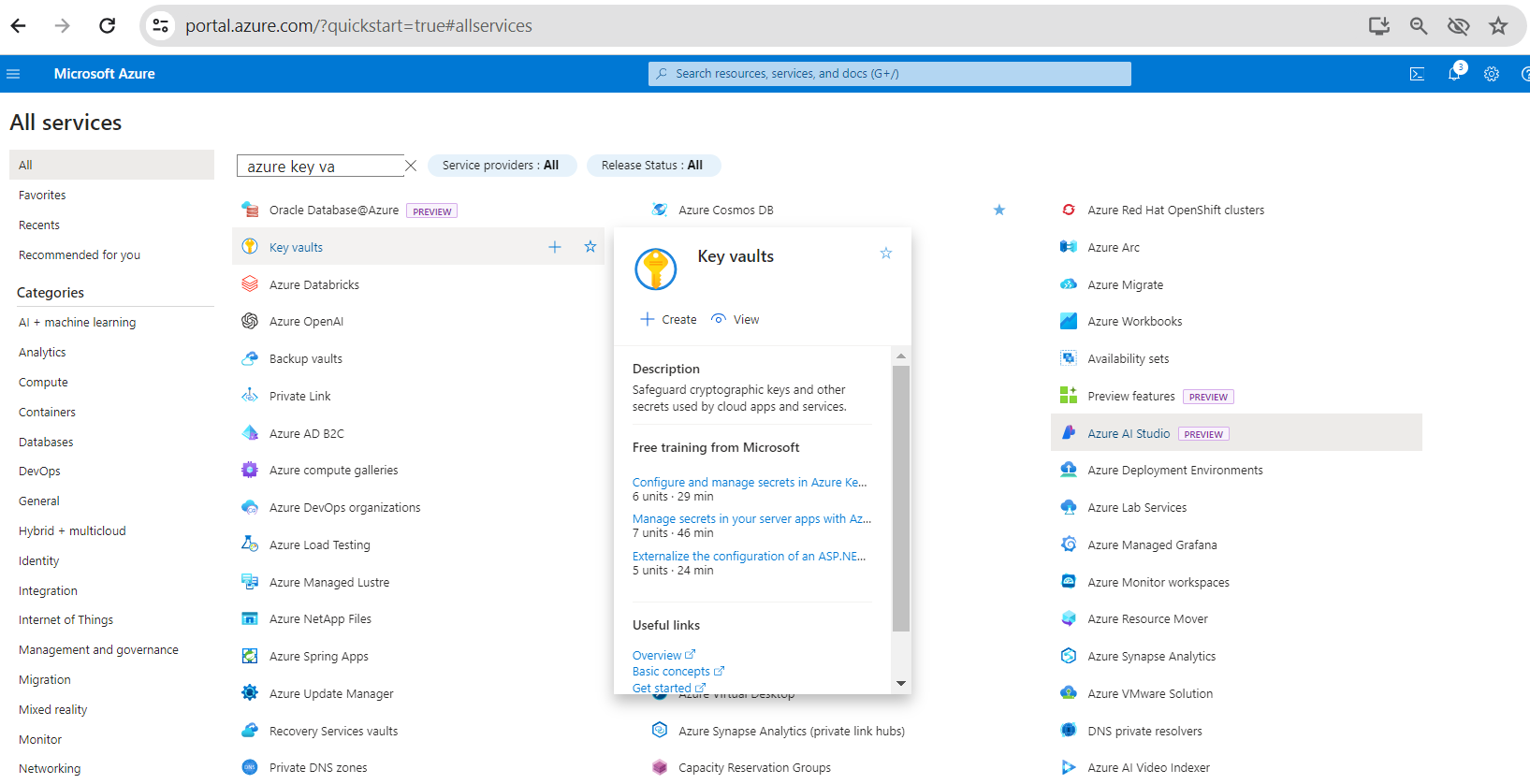
Search for the service principal and assign the contributor role:

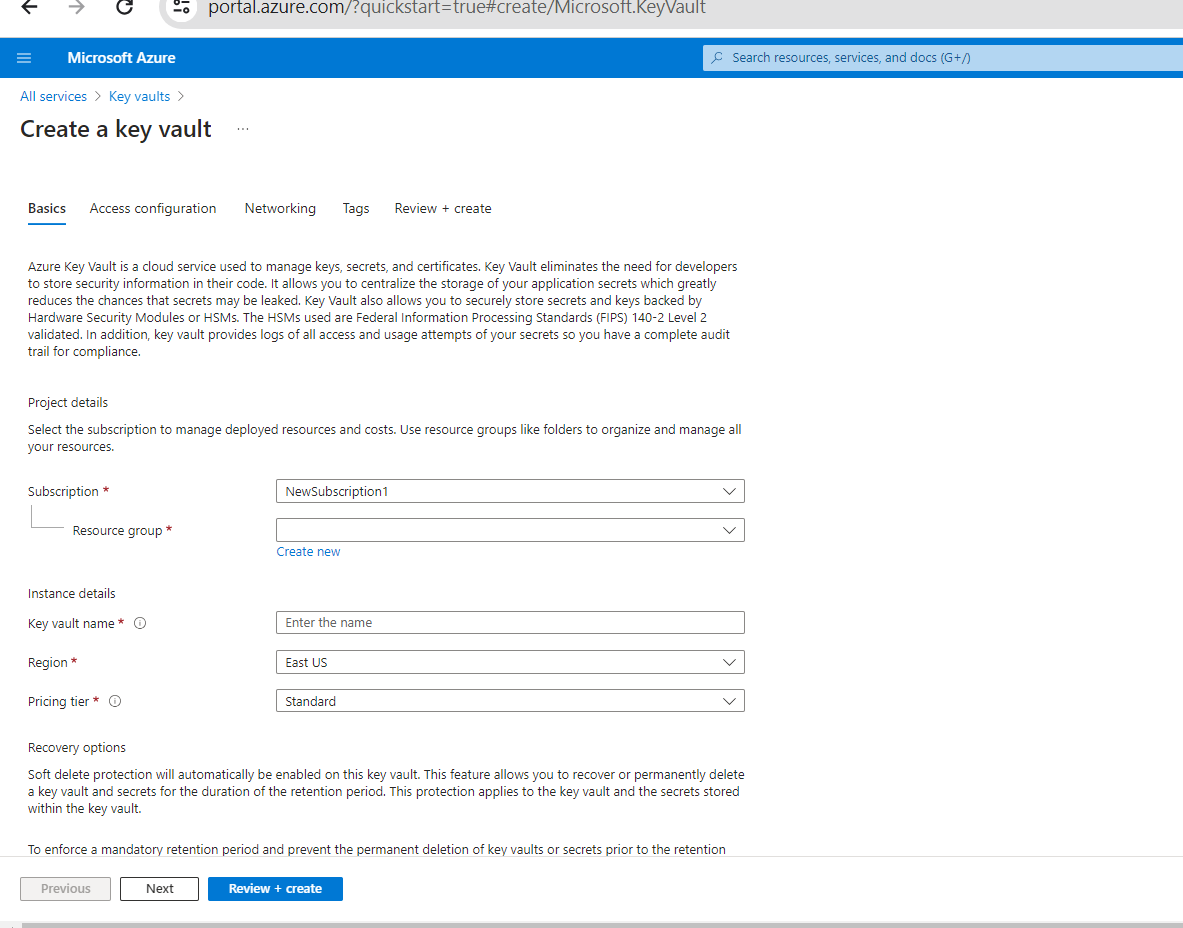


Create Azure Databricks:

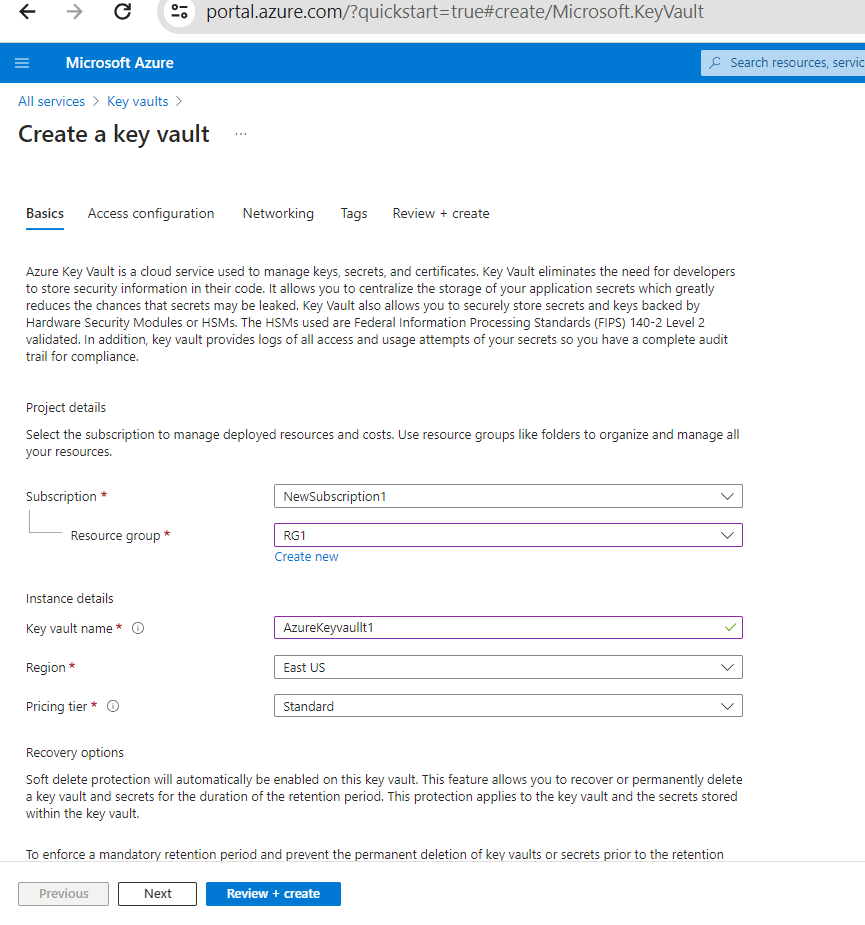


Creating Azure Key Vaults

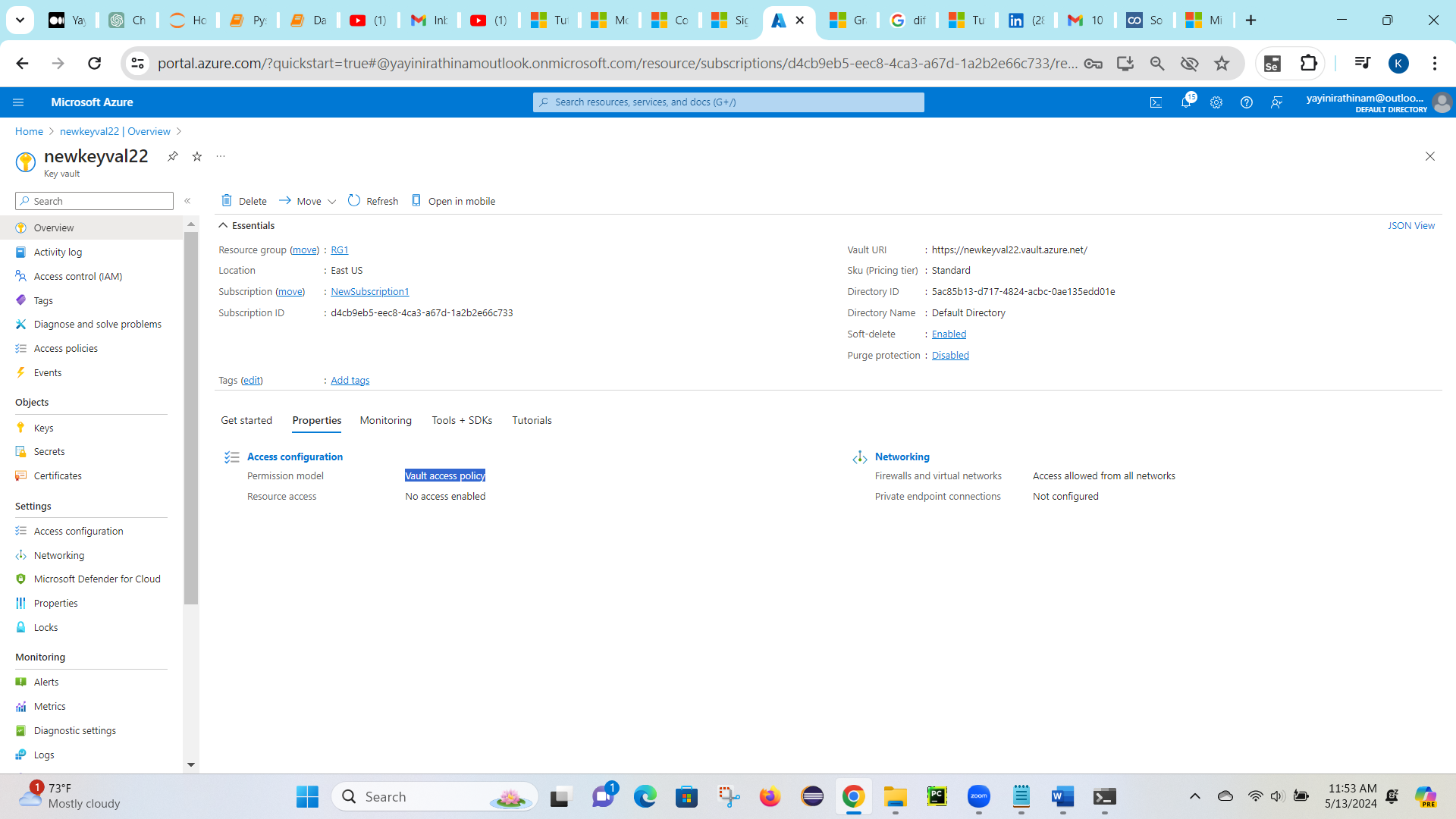




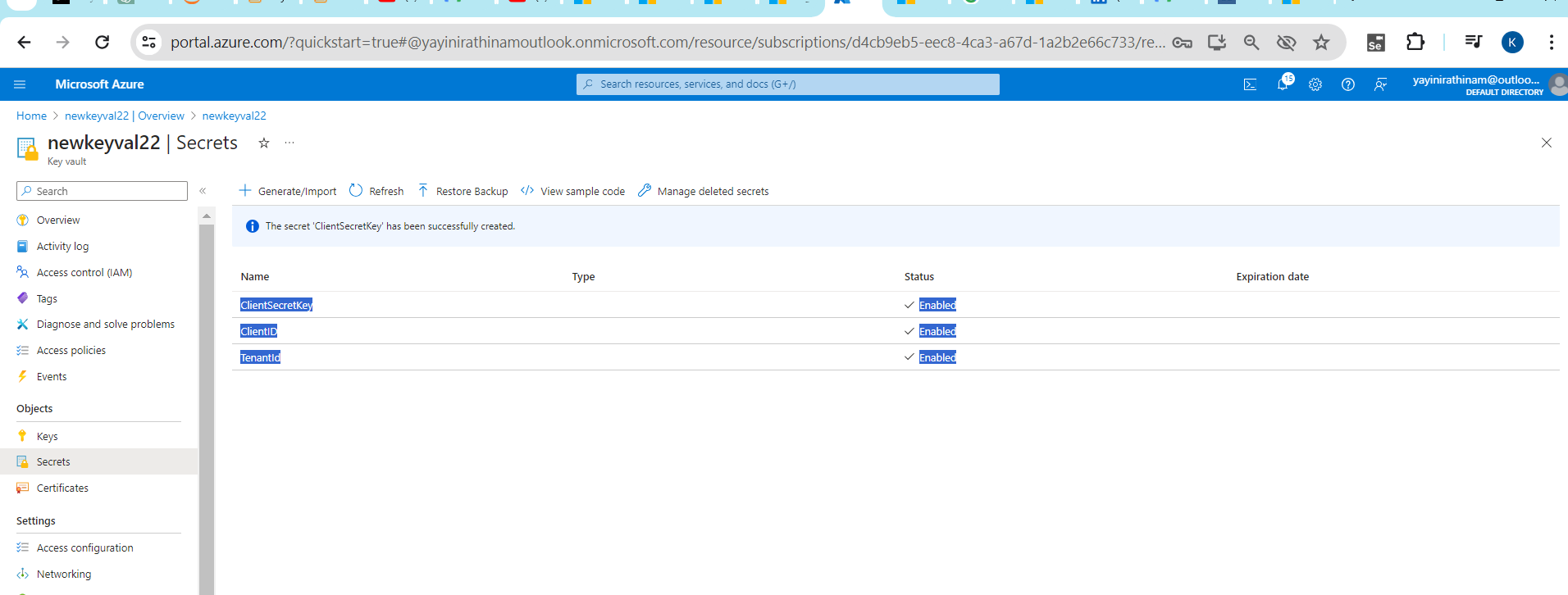
Creating a key vault:



Ensure you select this while creating key vault not the RBAC one



3 secrtes are created for client id, tenenat id and clientsecret

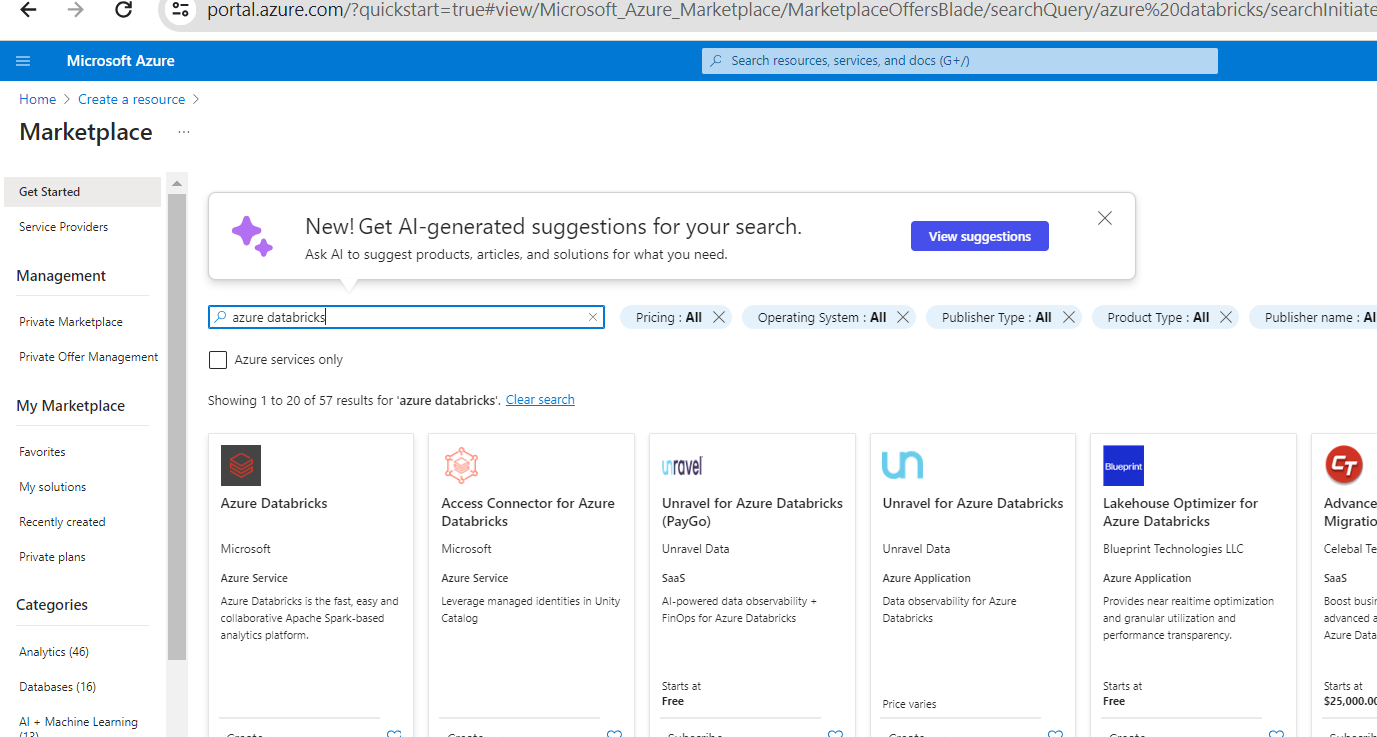


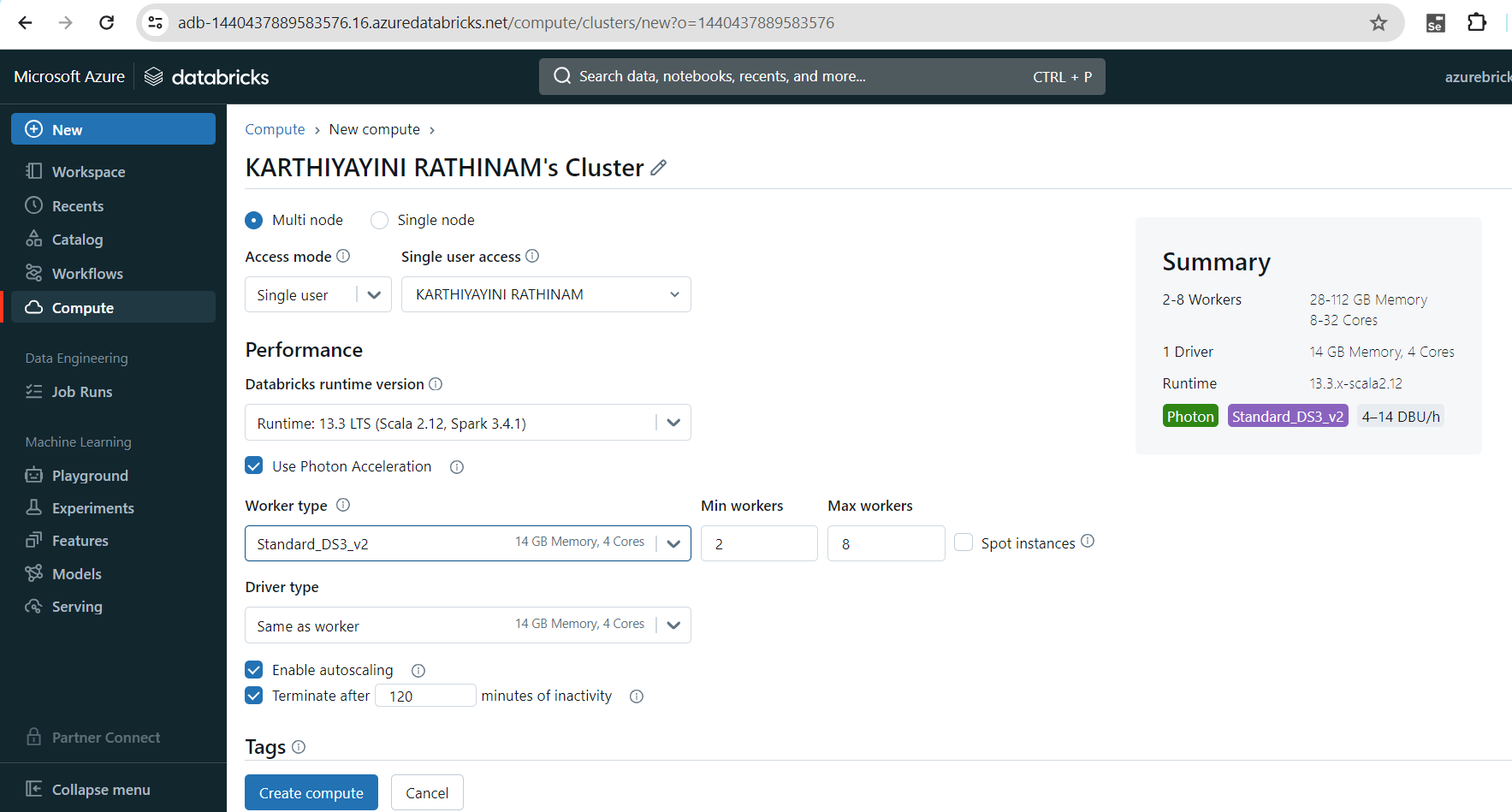
Copy the URL of Key vault and Resource ID

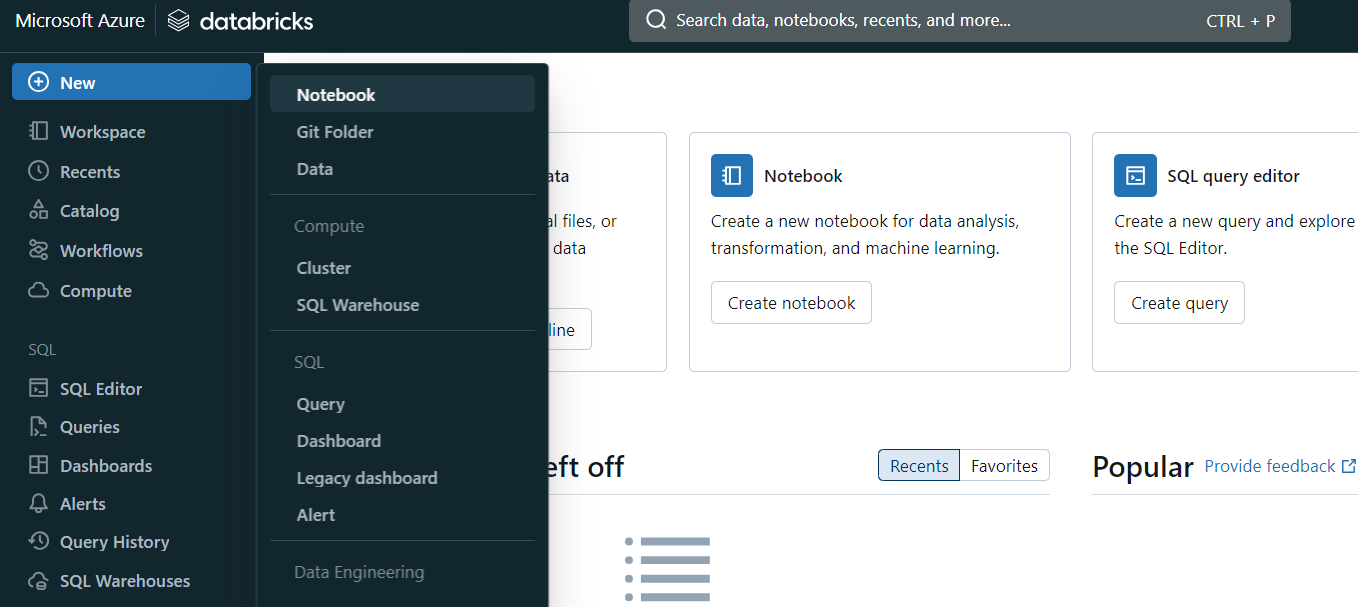
<https://newkeyval22.vault.azure.net/>

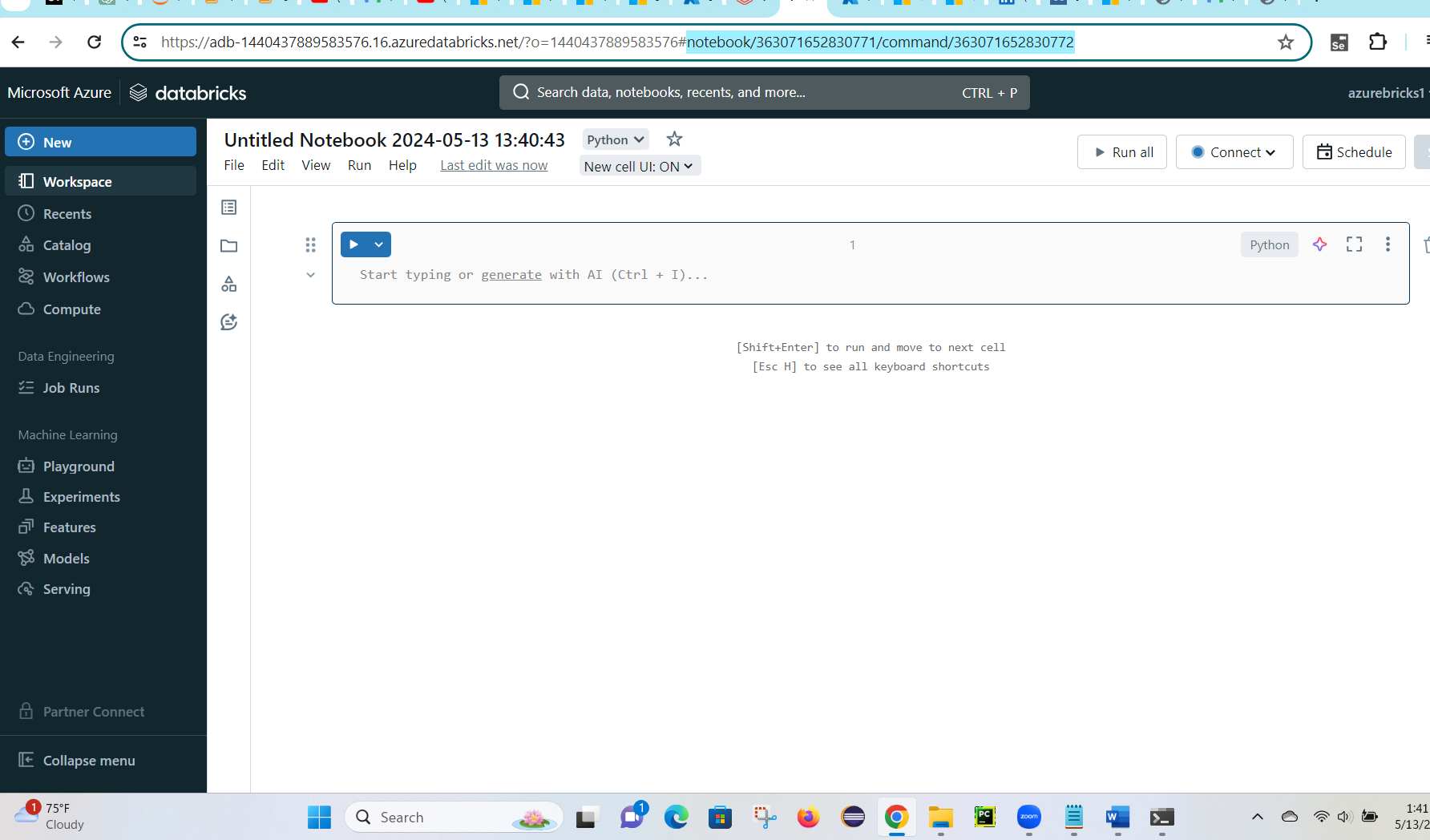
**/subscriptions/d4cb9eb5-eec8-4ca3-a67d-1a2b2e66c733/resourceGroups/RG1/providers/Microsoft.KeyVault/vaults/newkeyval22**

**secretscope**

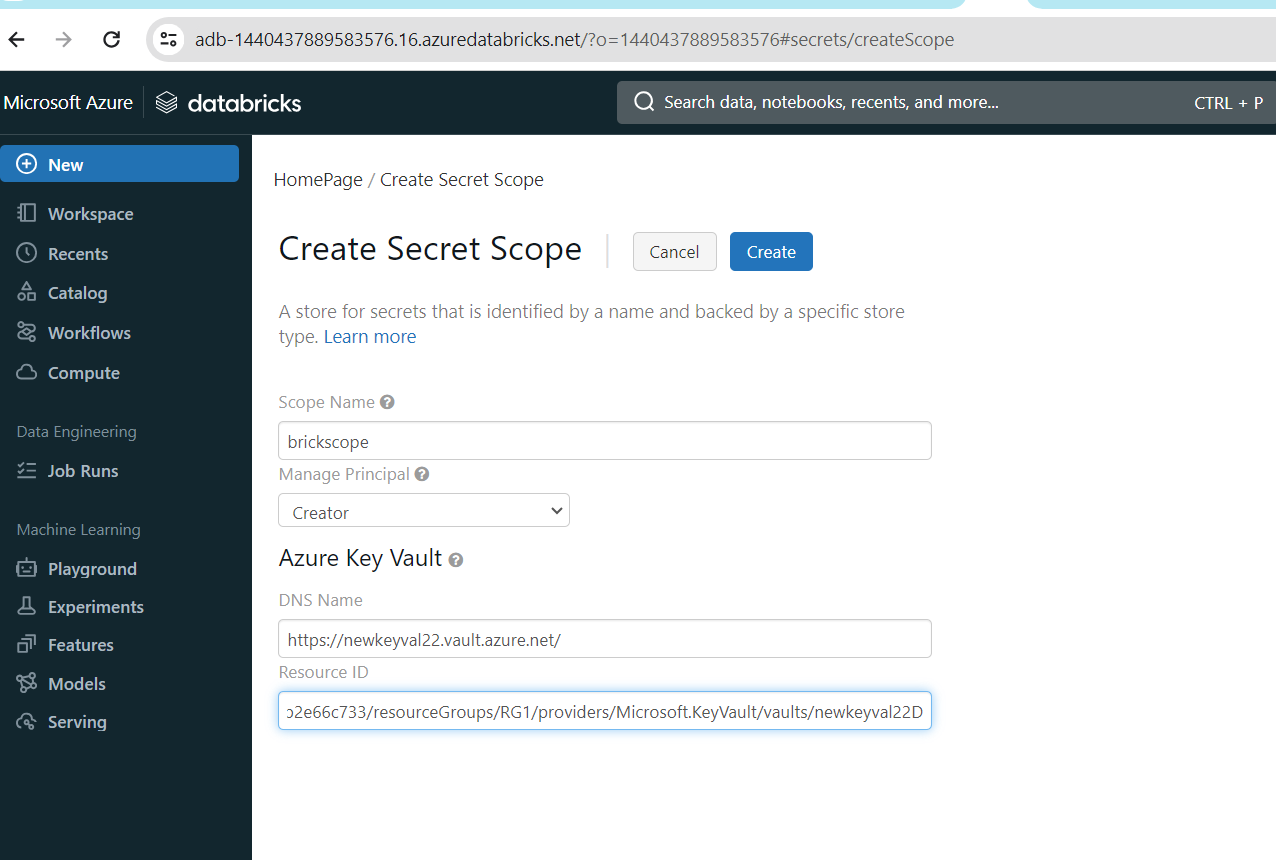


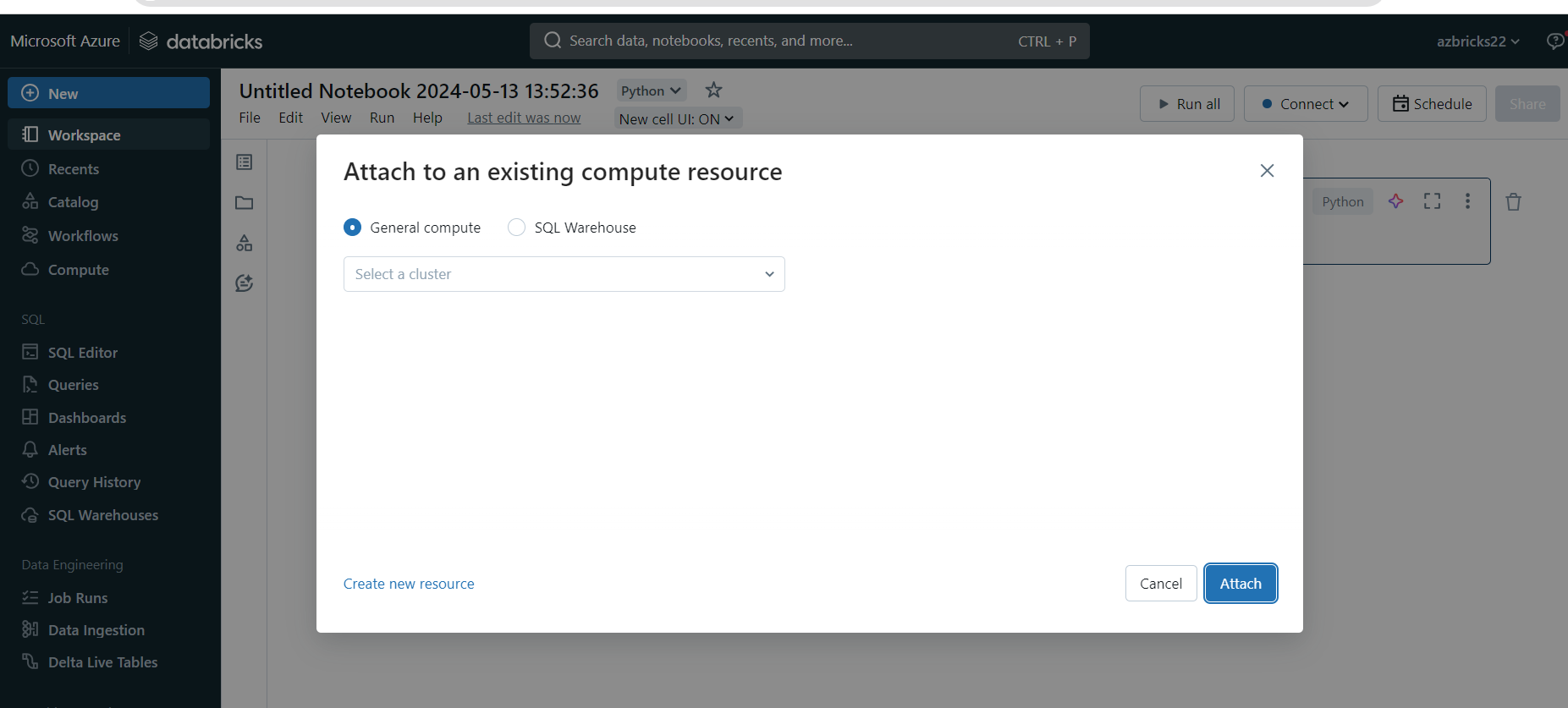
****

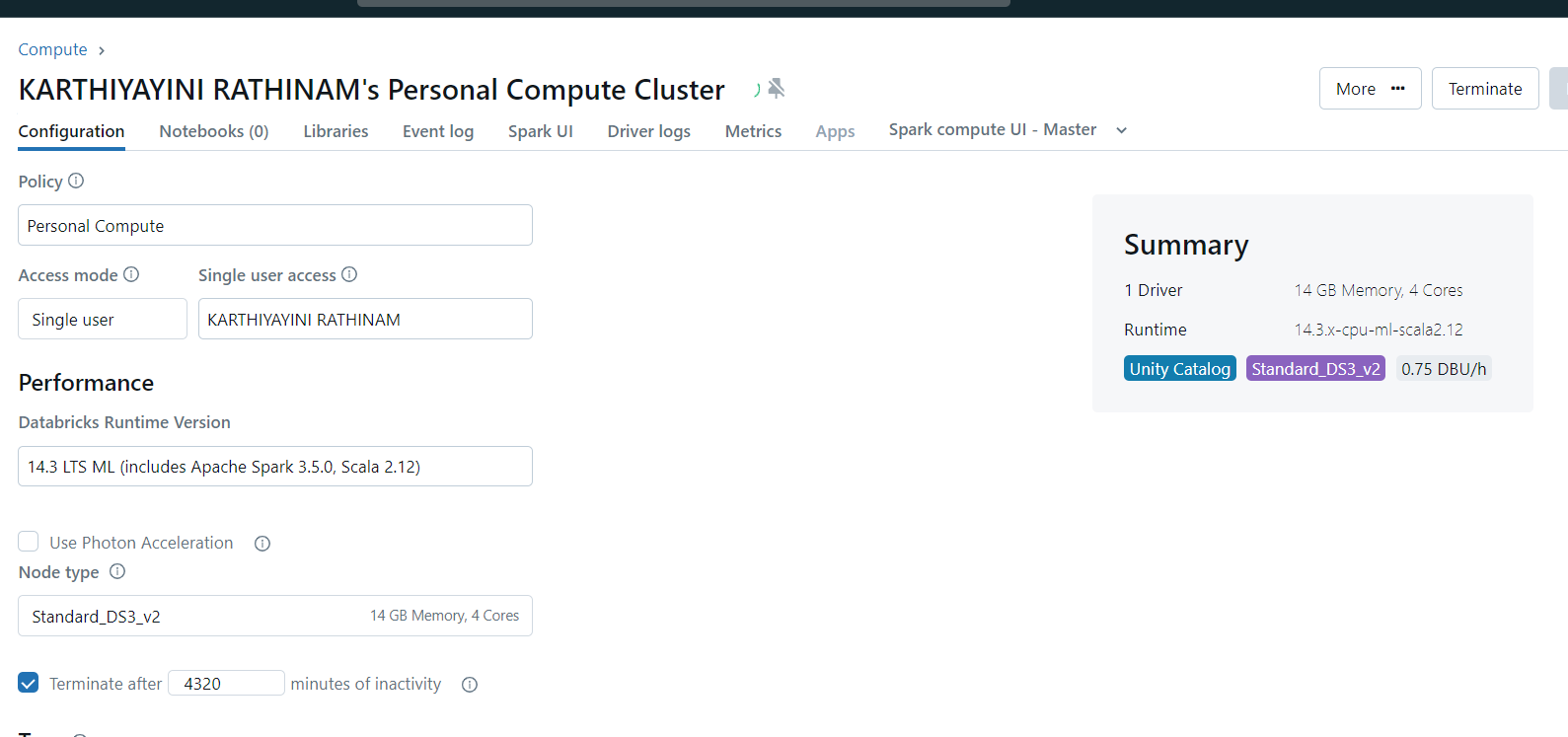
****

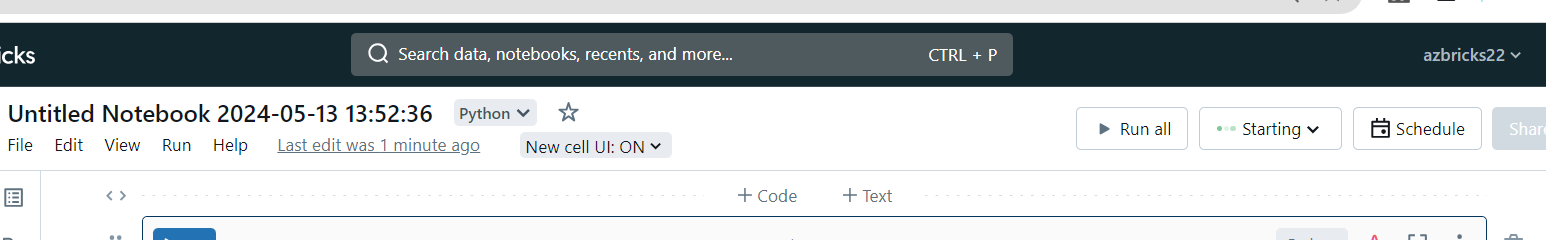
****

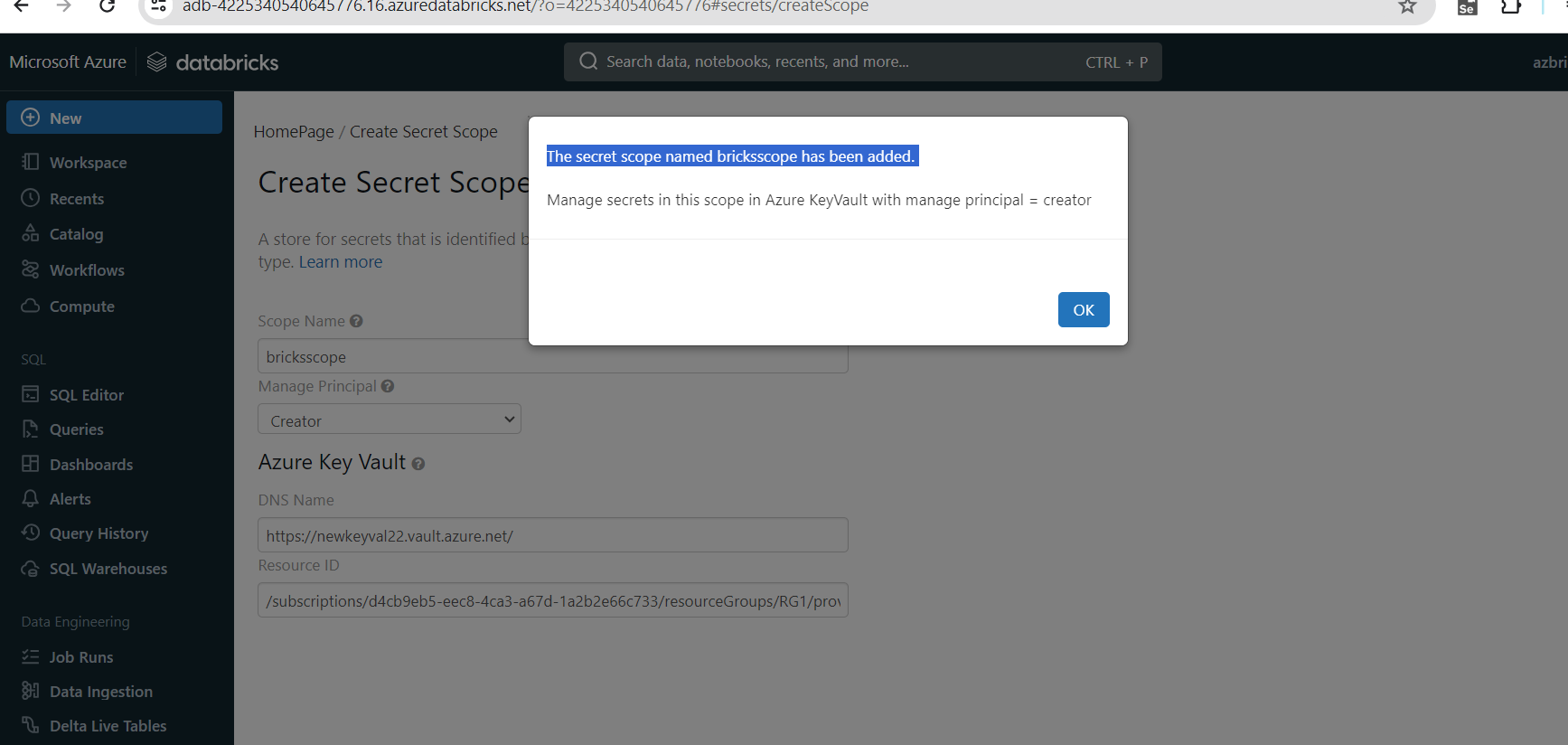
**Go to create scope page and provide key vault url and resource type copied**

****

****

****

****

****

**Bricksscope is the name of the scope**

sp\_clientid = dbutils.secrets.get(scope="brickscope",key="ClientID")

sp\_clientsecret\_value = dbutils.secrets.get(scope="brickscope",key="ClientSecretKey")

tenantid = dbutils.secrets.get(scope="brickscope",key="TenantId")

oauth\_endpoint = "https://login.microsoftonline.com/"+tenantid+"/oauth2/token"

adls\_container\_name = "container1"

adls\_account\_name = "adlsnew1"

adls\_source = "abfss://{}@{}.dfs.core.windows.net/".format(adls\_container\_name,adls\_account\_name)

mount\_point\_name = "/mnt/sample-datasets"

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": sp\_clientid,

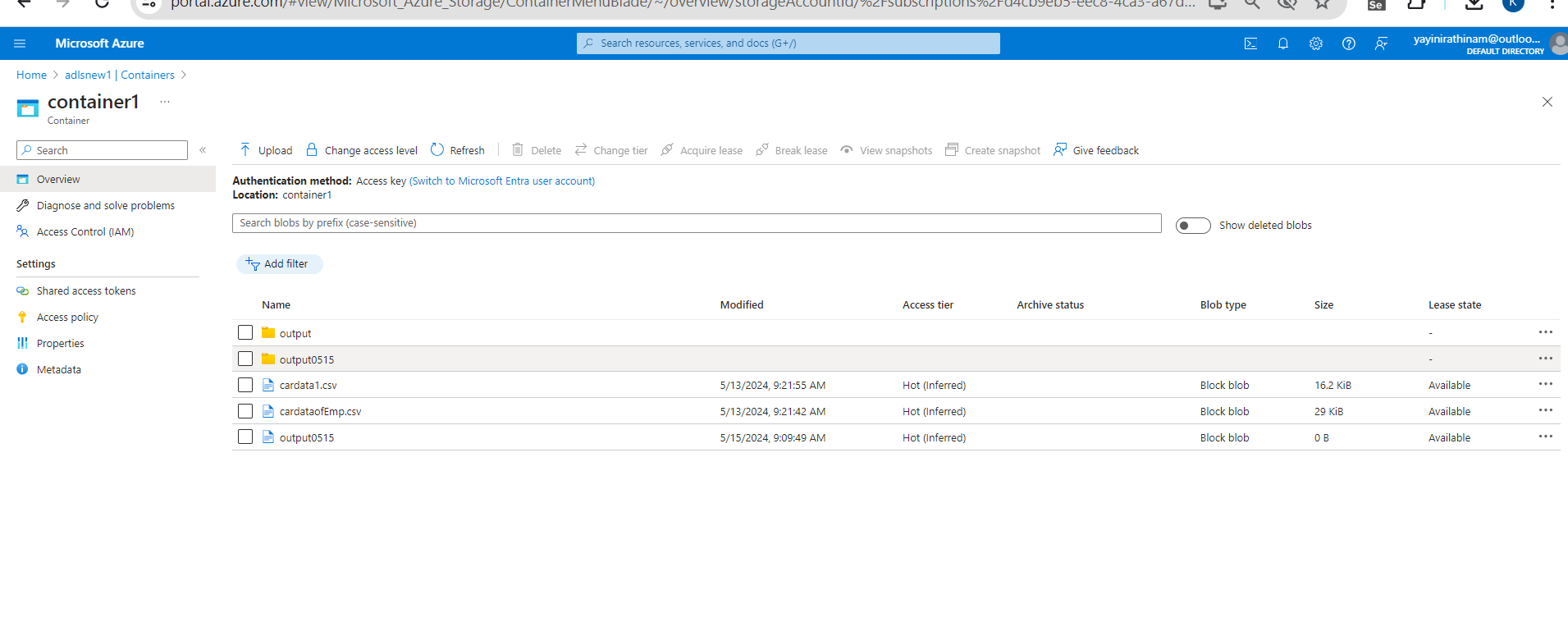
          "fs.azure.account.oauth2.client.secret": sp\_clientsecret\_value,

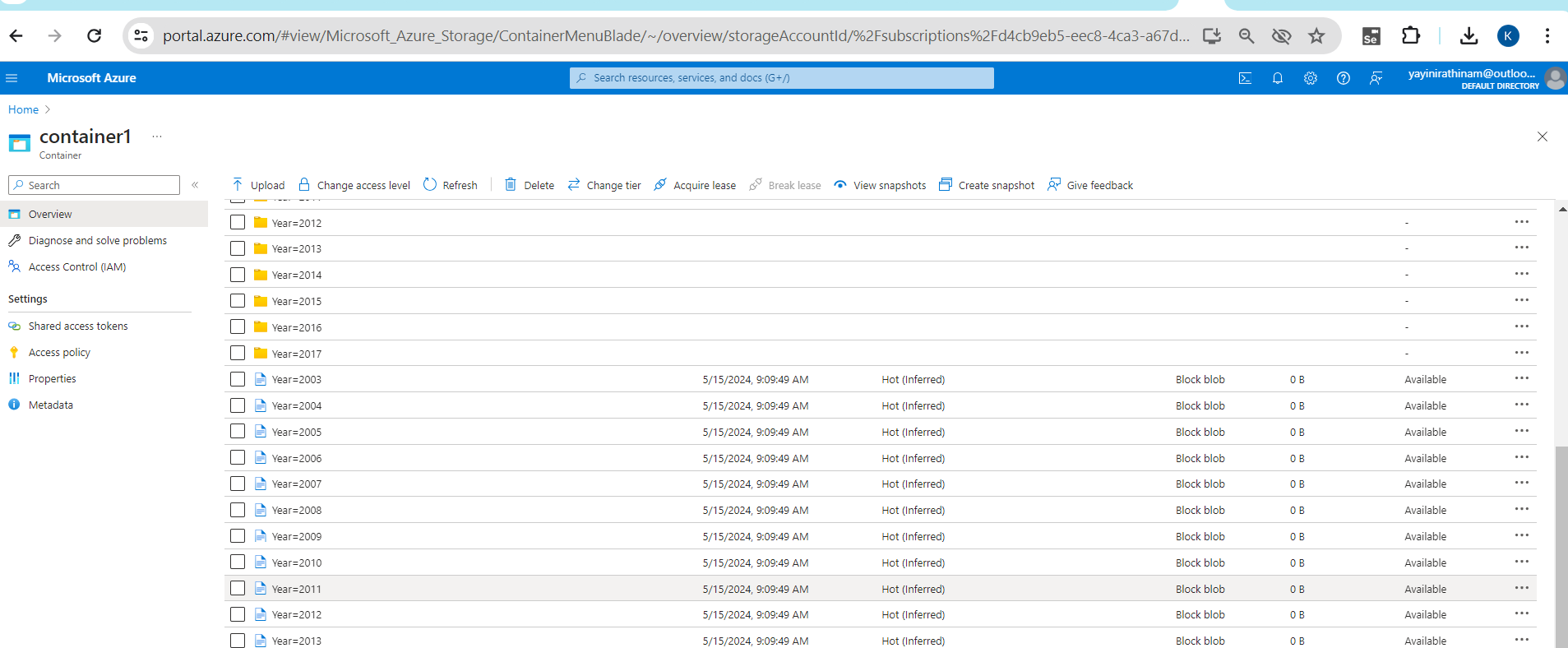
          "fs.azure.account.oauth2.client.endpoint": oauth\_endpoint}

if not any(mount.mountPoint == mount\_point\_name for mount in dbutils.fs.mounts()):

    dbutils.fs.mount(source = adls\_source,mount\_point = mount\_point\_name,extra\_configs = configs)

    print("mount succeeded!")

****

****

**Refer the coding file.**