1

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

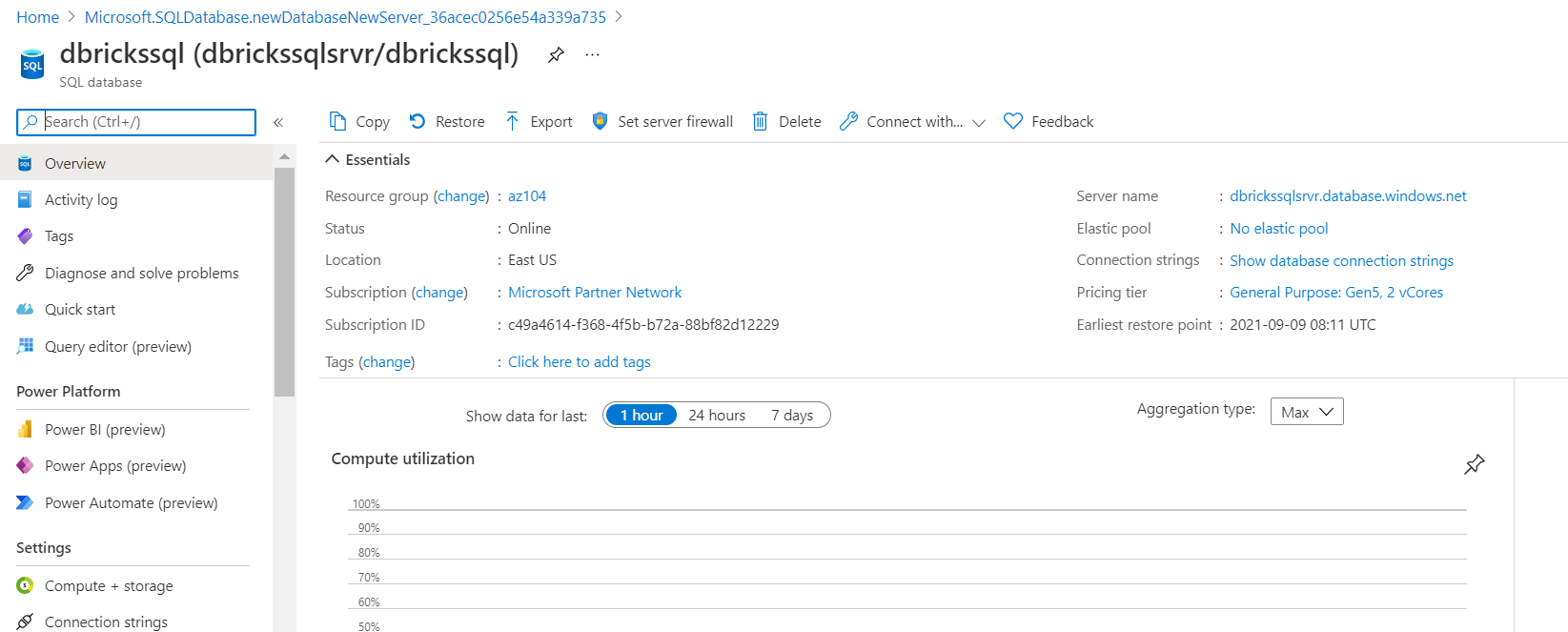
Now review and Create

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



Noite down the server addfress

Graphical user interface, text, application, email

Description automatically generated

Create New Table and upload the CSV file

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

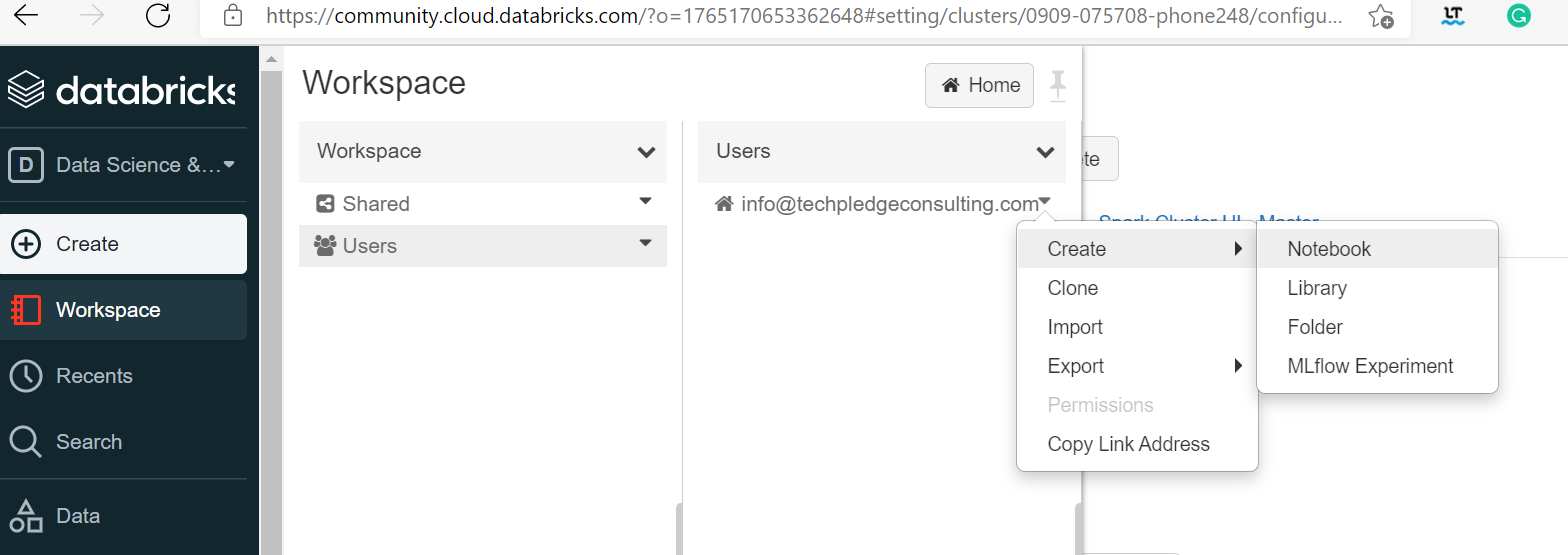
Description automatically generated

Notedown the oath

Graphical user interface, text, application, email

Description automatically generated

Create new Notebook



Graphical user interface, text, application, email

Description automatically generated

Background pattern

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Launch workspace

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Create Cluster

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Upload data

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email, website

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Type the script and cliek run to execute it\

from pyspark.sql import \*

import pandas as pd

jdbcHostname = "dbrickssqlsrvr.database.windows.net"

jdbcPort =1433

jdbcDatabase ="dbrickssql"

properties = {

"user": "dbadmin",

"password": "Password@123"

}

url = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)

filedf = sqlContext.read.csv("/FileStore/tables/BDCS\_DIY\_Sales.csv", header=True)

Graphical user interface, text, application, email

Description automatically generated



Now Run the script



Now add additional Line

from pyspark.sql import \*

import pandas as pd

jdbcHostname = "dbrickssqlsrvr.database.windows.net"

jdbcPort =1433

jdbcDatabase ="dbrickssql"

properties = {

"user": "dbadmin@dbrickssqlsrvr",

"password": "Password@123"

}

url = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)

**filedf = sqlContext.read.csv("/FileStore/tables/BDCS\_DIY\_Sales.csv", header=True)**

**filedf = filedf.withColumnRenamed("\_c0", "id")**

**filedf1 = DataFrameWriter(filedf)**

Graphical user interface, text, application, email

Description automatically generated

Now add last line to write to database

from pyspark.sql import \*

import pandas as pd

jdbcHostname = "dbrickssqlsrvr.database.windows.net"

jdbcPort =1433

jdbcDatabase ="dbrickssql"

properties = {

"user": "dbadmin@dbrickssqlsrvr",

"password": "Password@123",

"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"

}

url = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)

filedf = sqlContext.read.csv("/FileStore/tables/BDCS\_DIY\_Sales.csv", header=True)

filedf = filedf.withColumnRenamed("\_c0", "id")

filedf1 = DataFrameWriter(filedf)

**filedf1.jdbc(url=url, table="sales", mode="overwrite", properties=properties)**

Graphical user interface, text, application, email

Description automatically generated

**Now go to sql and qyery editir**

Graphical user interface, application

Description automatically generated

**Add the up to dfirewall**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Lab 2**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Click Revuew abd Create

Graphical user interface, text, application, email

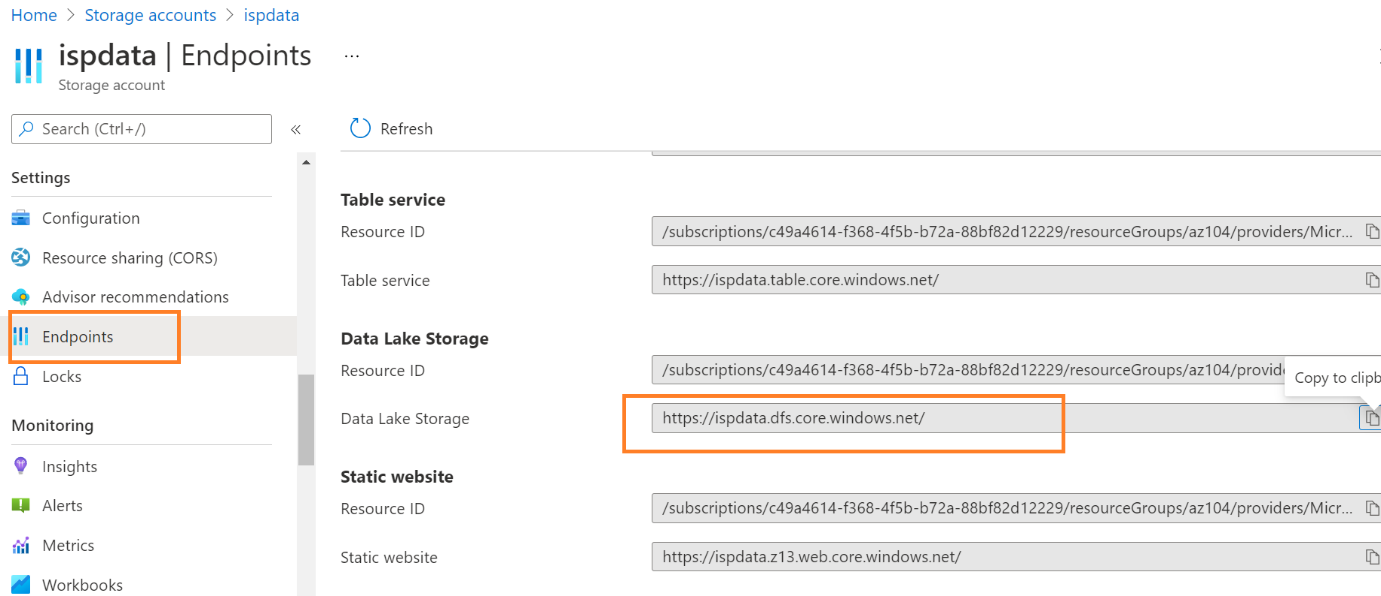
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Notedown the dfatalake URL , change the **https to adl**

adl://ispdata.dfs.core.windows.net



Go to azure ad

Graphical user interface, text, application, email

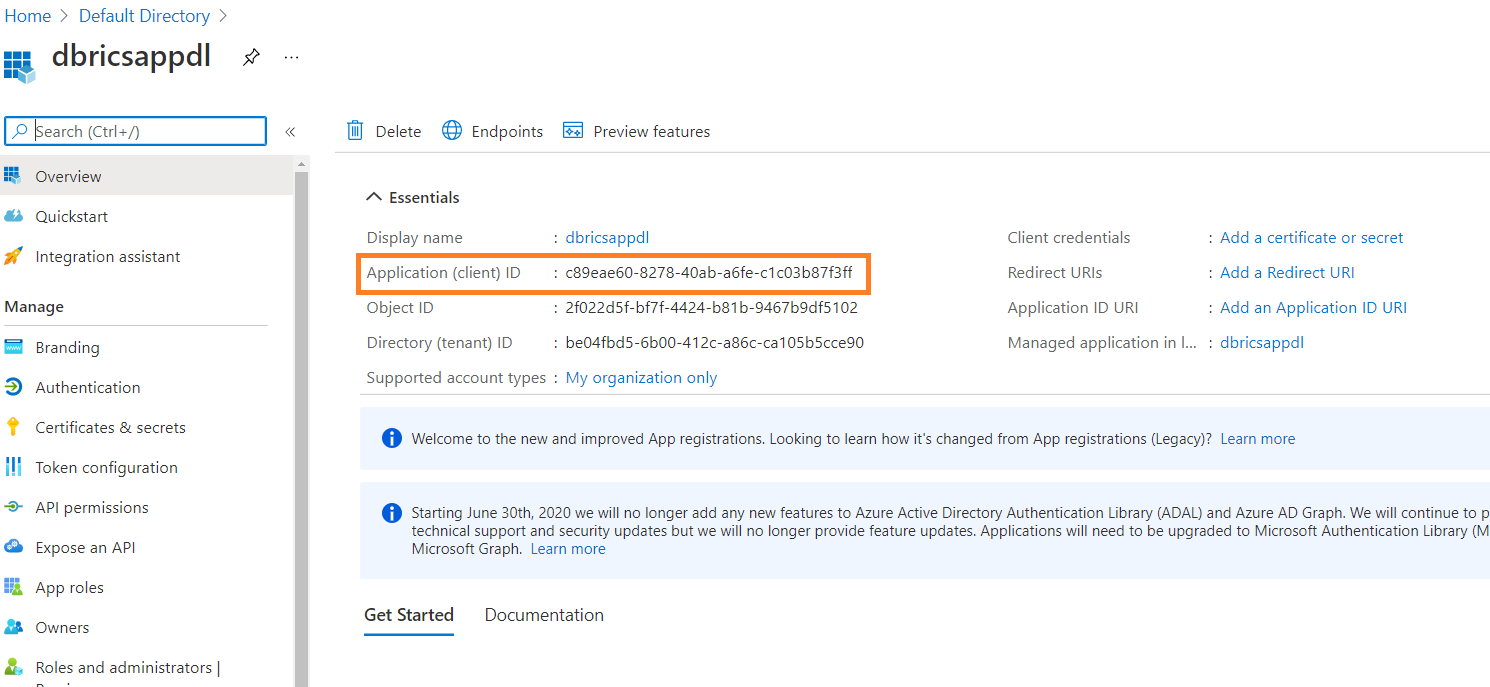
Description automatically generated

Graphical user interface, text, application, email

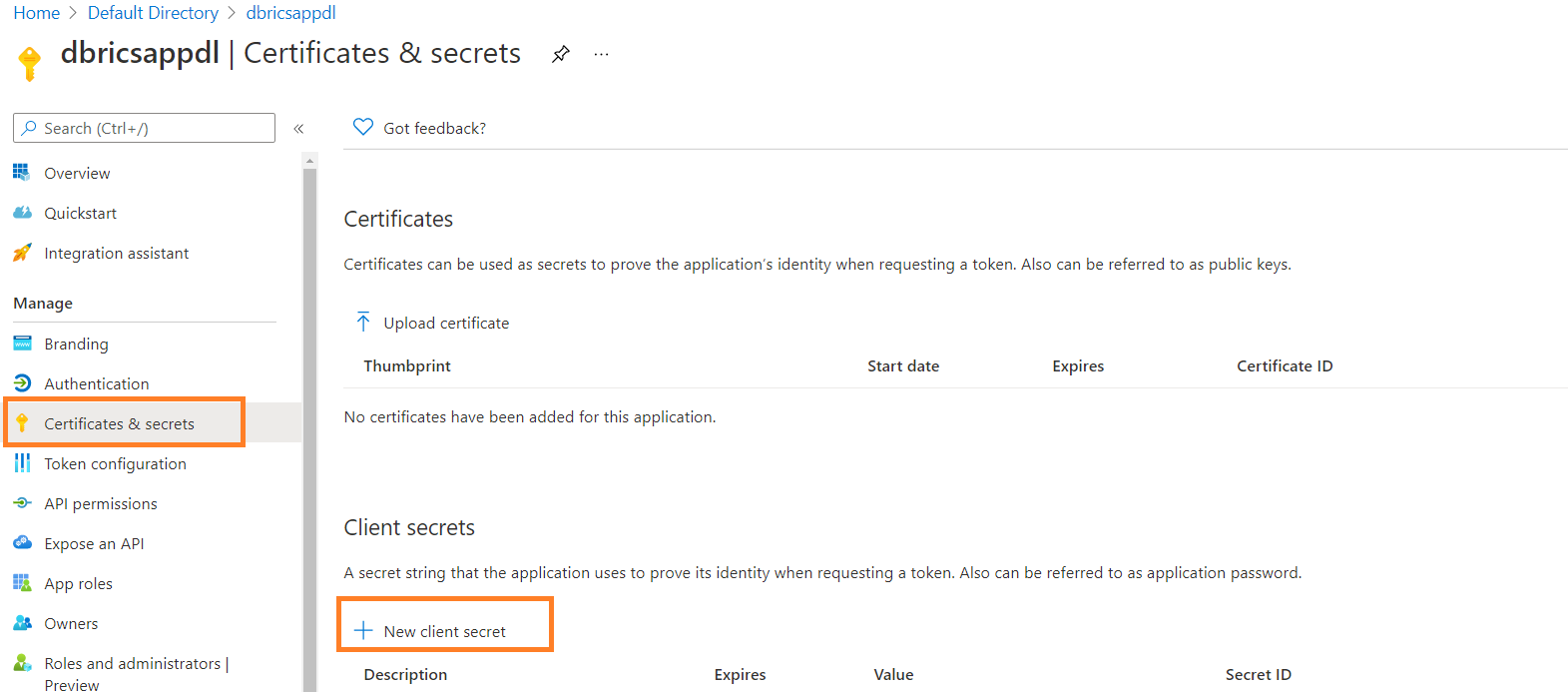
Description automatically generated

Graphical user interface, text, application

Description automatically generated

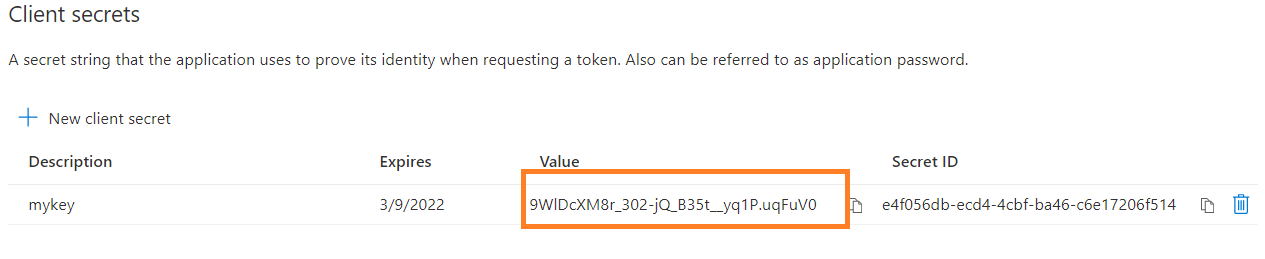


Notedown app id : c89eae60-8278-40ab-a6fe-c1c03b87f3ff



Graphical user interface, text, application

Description automatically generated



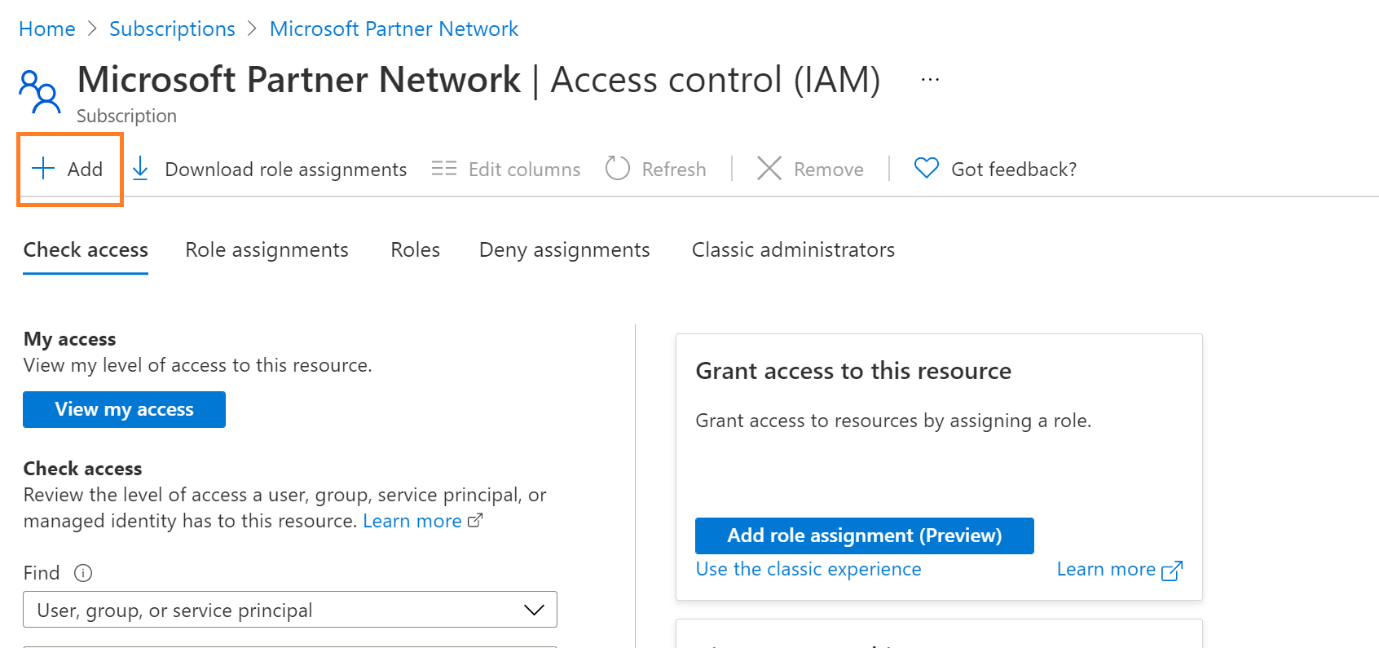
Copy the key

9WlDcXM8r\_302-jQ\_B35t\_\_yq1P.uqFuV0

Get the tenant ID

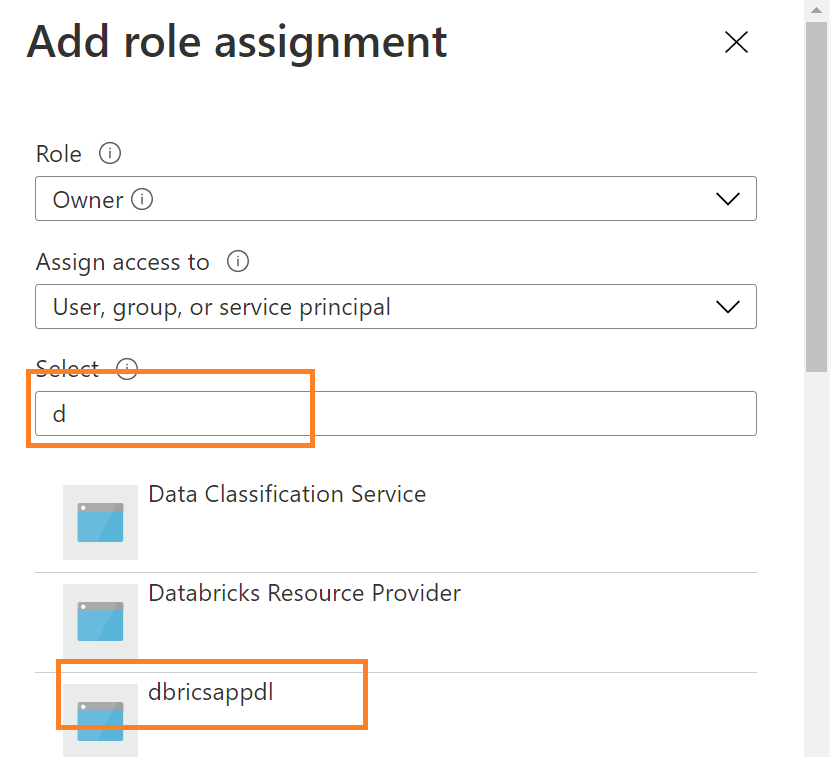
Graphical user interface, text, application, email

Description automatically generated



Graphical user interface, application

Description automatically generated



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

dbutils.fs.mkdirs("/mnt/mountdatalake")

configs = {"dfs.adls.oauth2.access.token.provider.type":"ClientCredential",

"dfs.adls.oauth2.client.id": "c89eae60-8278-40ab-a6fe-c1c03b87f3ff",

"dfs.adls.oauth2.credential": "G2vtQpSq8G2djgf.W.U3PTi0fnXV-\_oF.e",

"dfs.adls.oauth2.refresh.url": "https://login.microsoftonline.com/be04fbd5-6b00-412c-a86c-ca105b5cce90/oauth2/token"}

dbutils.fs.mount(

source = "adl://ispdata.dfs.core.windows.net/mountdatalake",

mount\_point = "/mnt/mountdatalake",

extra\_configs = configs )

Graphical user interface, text, application, email

Description automatically generated

Now the softlibnk created , lets now write the data from sql top datalkaken

\

\

Create new nioreb book

Graphical user interface, text, application, email

Description automatically generated

Now write below code firsty tp creaed from sql

from pyspark.sql import \*

import pandas as pd

jdbcHostname = "dbrickssqlsrvr.database.windows.net"

jdbcPort =1433

jdbcDatabase ="dbrickssql"

properties = {

"user": "dbadmin@dbrickssqlsrvr",

"password": "Password@123",

"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"

}

url = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)

pushdown\_query = "(select \* from dbo.sales) as sales"

print(pushdown\_query)

# This Panda read the data from sql

df2 = spark.read.jdbc(url=url, table=pushdown\_query, properties=properties)

Graphical user interface, text, application, email

Description automatically generated

You should see the outrput as

Table

Description automatically generated

Now add new line to read data from sql and write to datalake

from pyspark.sql import \*

import pandas as pd

jdbcHostname = "dbrickssqlsrvr.database.windows.net"

jdbcPort =1433

jdbcDatabase ="dbrickssql"

properties = {

"user": "dbadmin@dbrickssqlsrvr",

"password": "Password@123",

"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"

}

url = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)

pushdown\_query = "(select \* from dbo.sales) as sales"

print(pushdown\_query)

# This Panda read the data from sql

df2 = spark.read.jdbc(url=url, table=pushdown\_query, properties=properties)

# This Panda read the write data to datalake

**df2.toPandas().to\_csv('/dbfs/mnt/mountdatalake/sales.csv', header='True', encoding="utf-8")**

Graphical user interface, text, application

Description automatically generated