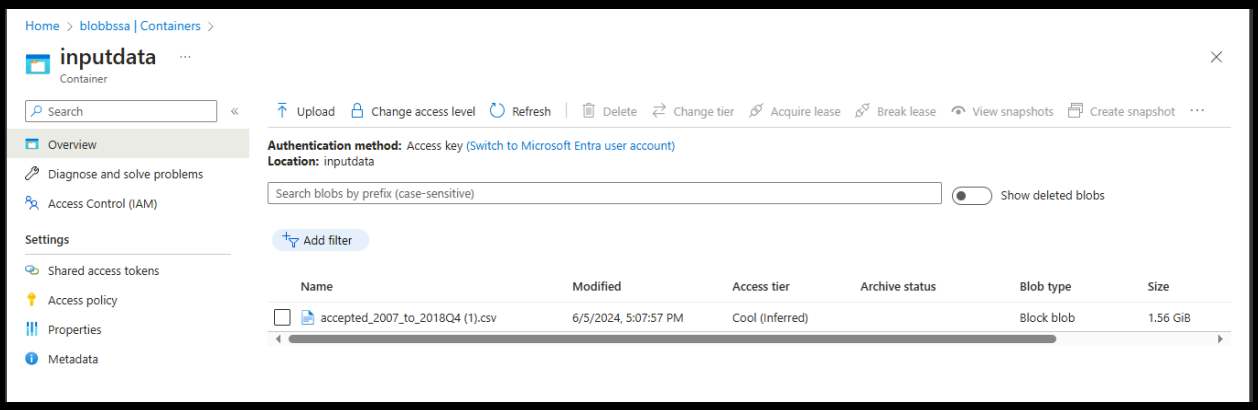
Pyspark Project : Blob Storage → Azure Databricks → Azure SQL Database :

Services required :

Azure Blob Storage
Azure Databricks
Azure SQL Database
Azure key vault

Creation of azure blob storage :

→ The creation of azure blob storage is shown in the document below.
https://docs.google.com/document/d/1vZCMfM9ieAlTQm6Jdwl2JzxcMkwnyvkSJHCrry/edit2kVIo?usp=sharing
→ Create a container : inputData
→ Upload bike data in the inputData container



Creation of Azure SQL Database.
→ Creation of Azure SQL Database is shown in the document below.
https://docs.google.com/document/d/16iB1EsGKHc6-bcgTPSfqkK6BVf3n8_fpbat42uNOvXc/edit?usp=sharing

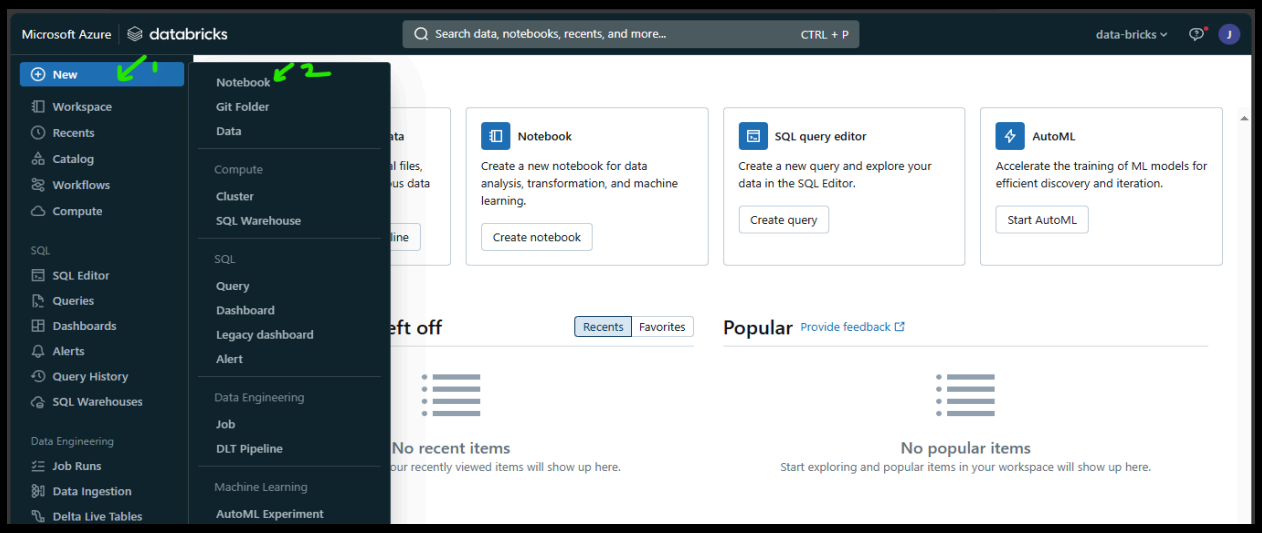
Creation of Azure Databricks
→ Creation of Azure SQL Databricks is shown in the document below.
https://docs.google.com/document/d/1s7Vqs4gcZbGMPRk7vEWGFBXIMC5AooOvC-E57qILwRw/edit?usp=sharing

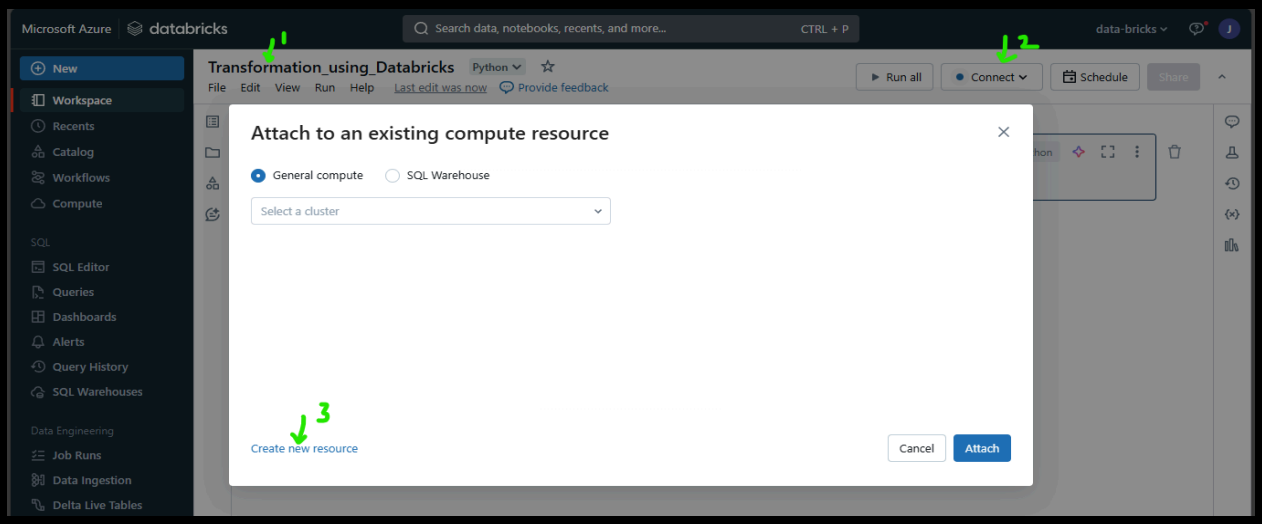
Creation of azure key vault :

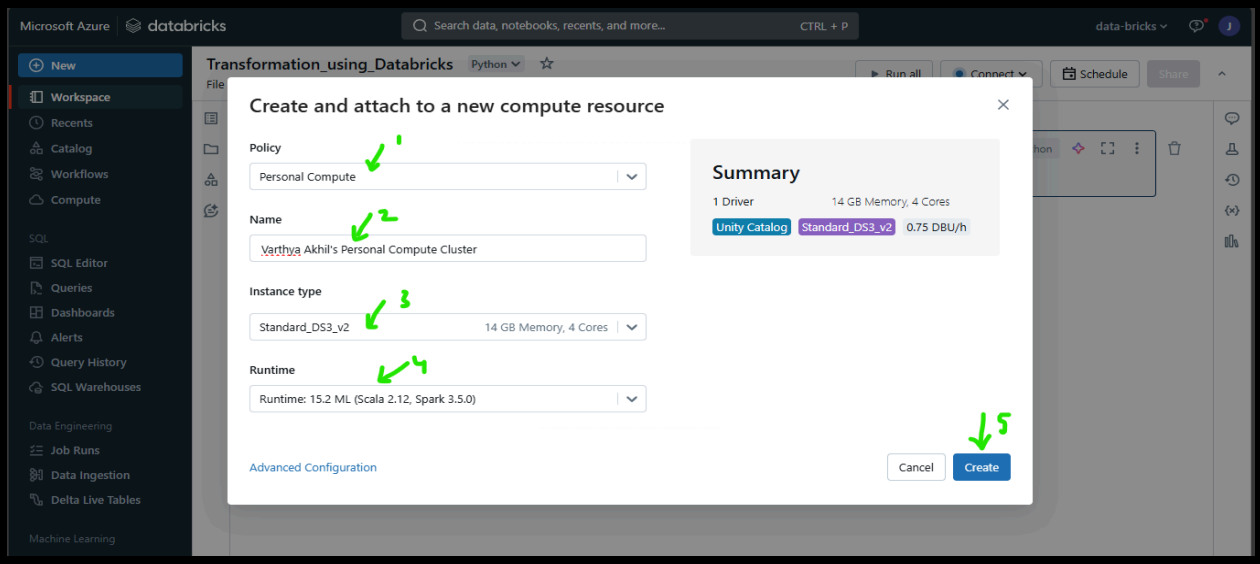
→ Creation of Azure Key Vault is shown in the document below.

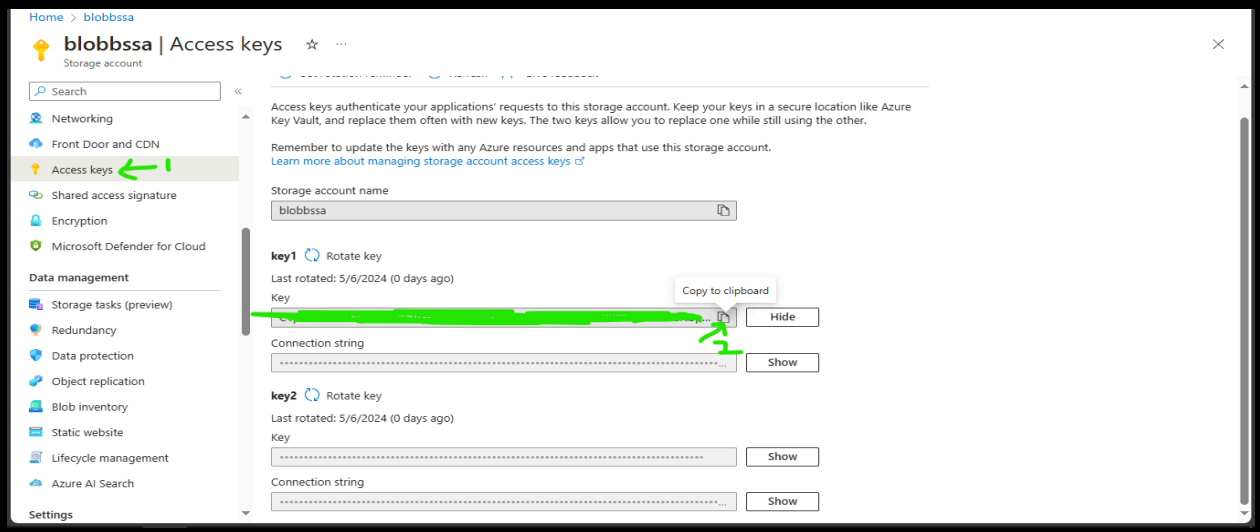
https://docs.google.com/document/d/1ScQ42B5c5ZuRnFjtLsdJpnV9067pLZkpcDf4WyKjaVw/edit?usp=sharing

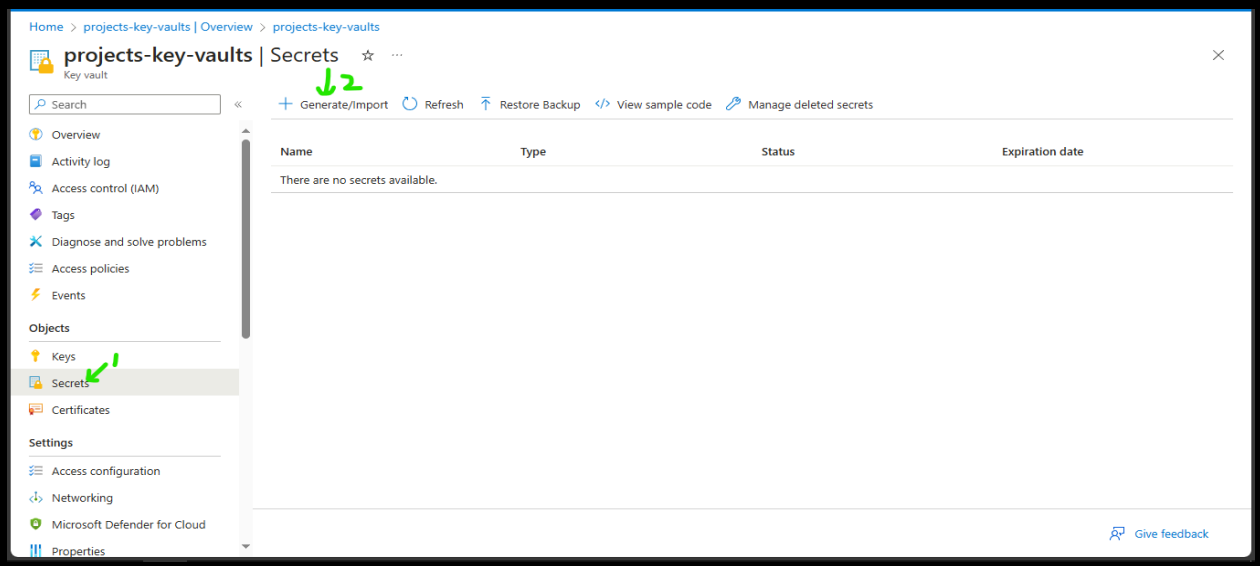
In Databricks workspace →

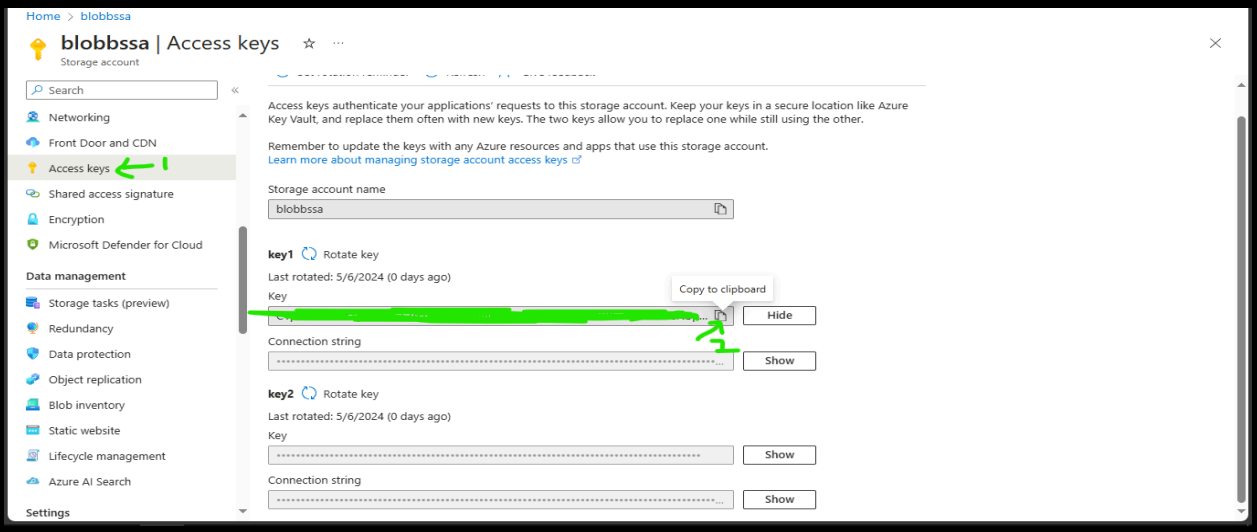




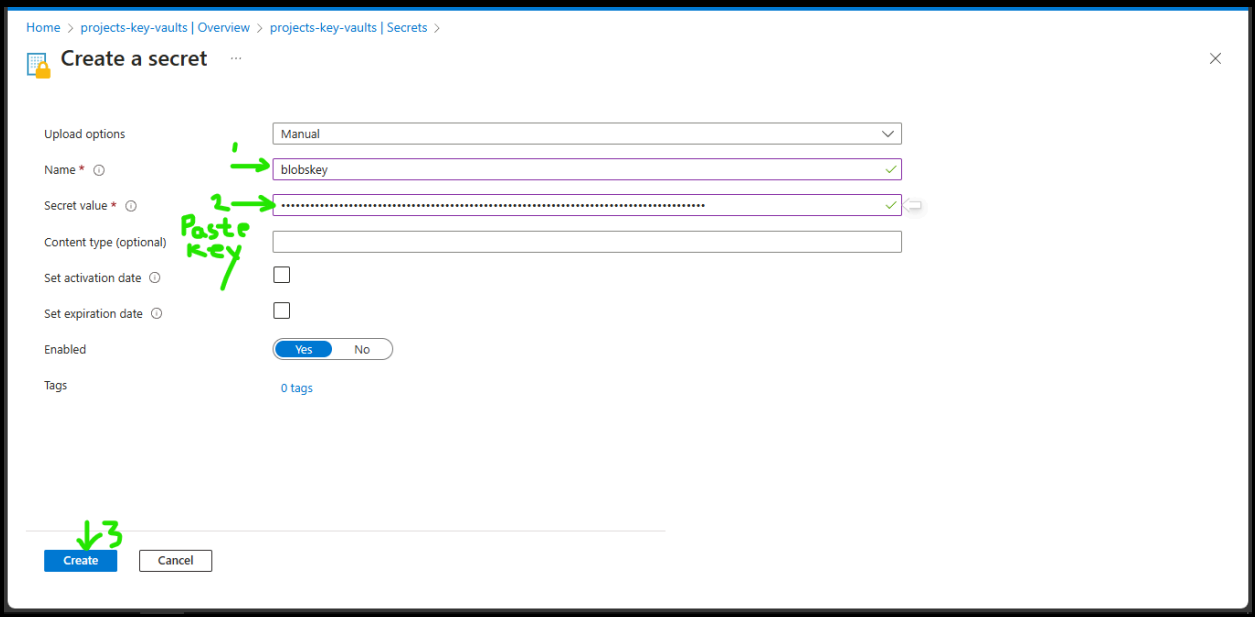

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
data-bricks
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Transformation_using_Databricks
Python
File
Run all
Connect
Schedule
Share
Create and attach to a new compute resource
Policy
Personal Compute
Name
Varthya Akhil's Personal Compute Cluster
Instance type
Standard_DS3_v2
14 GB Memory, 4 Cores
Runtime
Runtime: 15.2 ML (Scala 2.12, Spark 3.5.0)
Advanced Configuration
Summary
1 Driver
14 GB Memory, 4 Cores
Unity Catalog
Standard_DS3_v2
0.75 DBU/h
Cancel
Create


Home > blobbssa
blobbssa | Access keys
Storage account
Search
Networking
Front Door and CDN
Access keys
Shared access signature
Encryption
Microsoft Defender for Cloud
Data management
Storage tasks (preview)
Redundancy
Data protection
Object replication
Blob inventory
Static website
Lifecycle management
Azure AI Search
Settings
Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.
Remember to update the keys with any Azure resources and apps that use this storage account.
Learn more about managing storage account access keys
Storage account name
blobbssa
key1
Rotate key
Last rotated: 5/6/2024 (0 days ago)
Key
Copy to clipboard
Hide
Connection string
Show
key2
Rotate key
Last rotated: 5/6/2024 (0 days ago)
Key
Show
Connection string
Show


Home > projects-key-vaults | Overview > projects-key-vaults
projects-key-vaults | Secrets
Key vault
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Access policies
Events
Objects
Keys
Secrets
Certificates
Settings
Access configuration
Networking
Microsoft Defender for Cloud
Properties
Generate/Import
Refresh
Restore Backup
View sample code
Manage deleted secrets
Name
Type
Status
Expiration date
There are no secrets available.
Give feedback



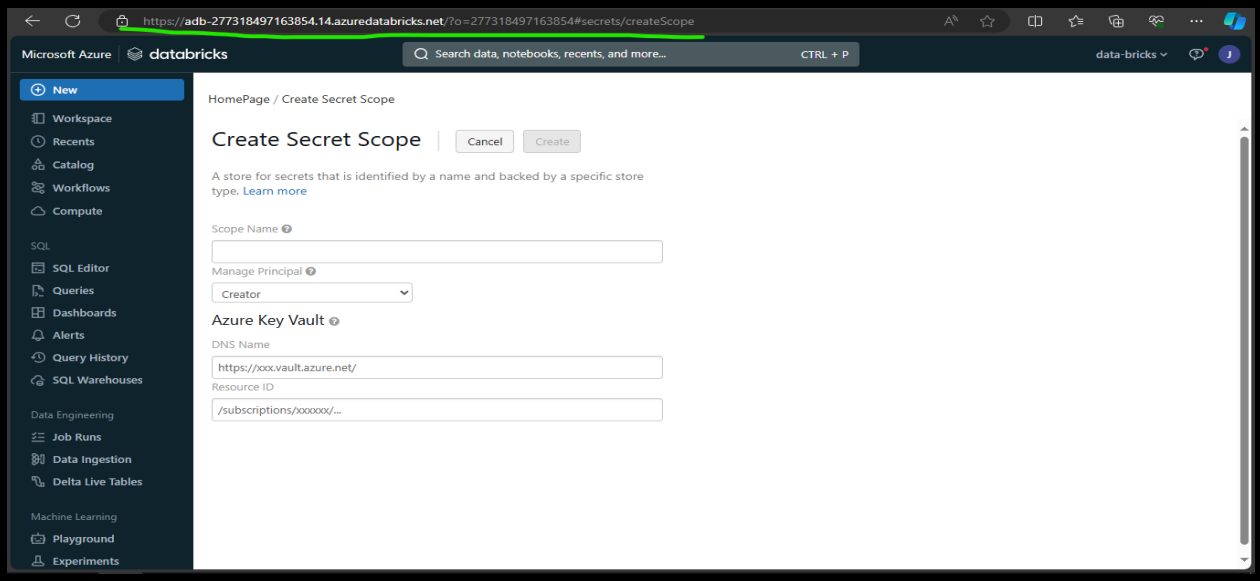
Copy the access key from blob storage and paste in the secret value.

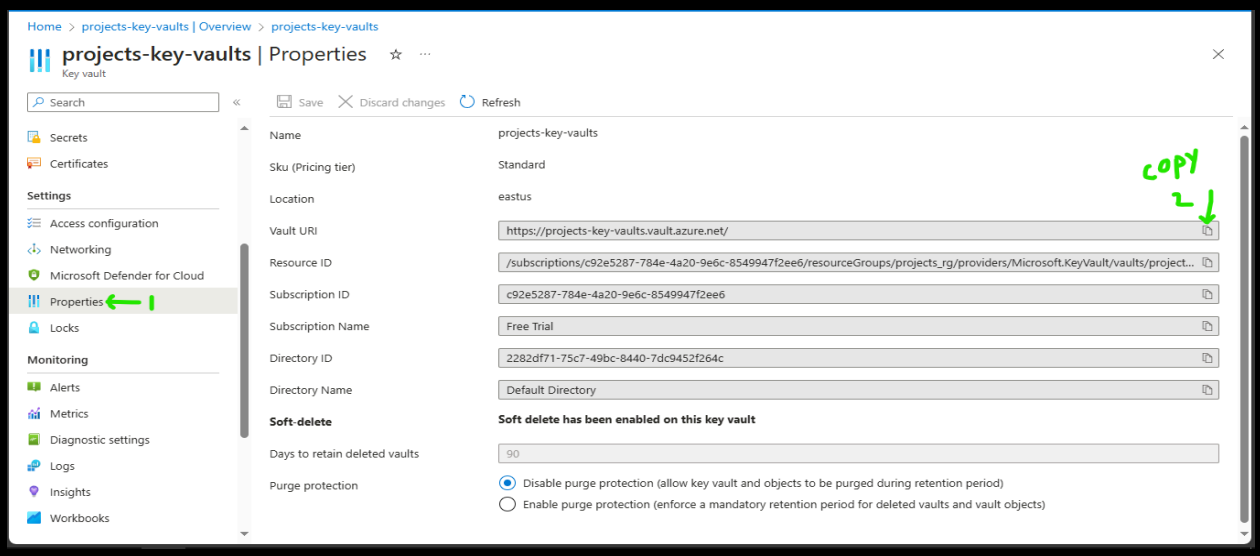


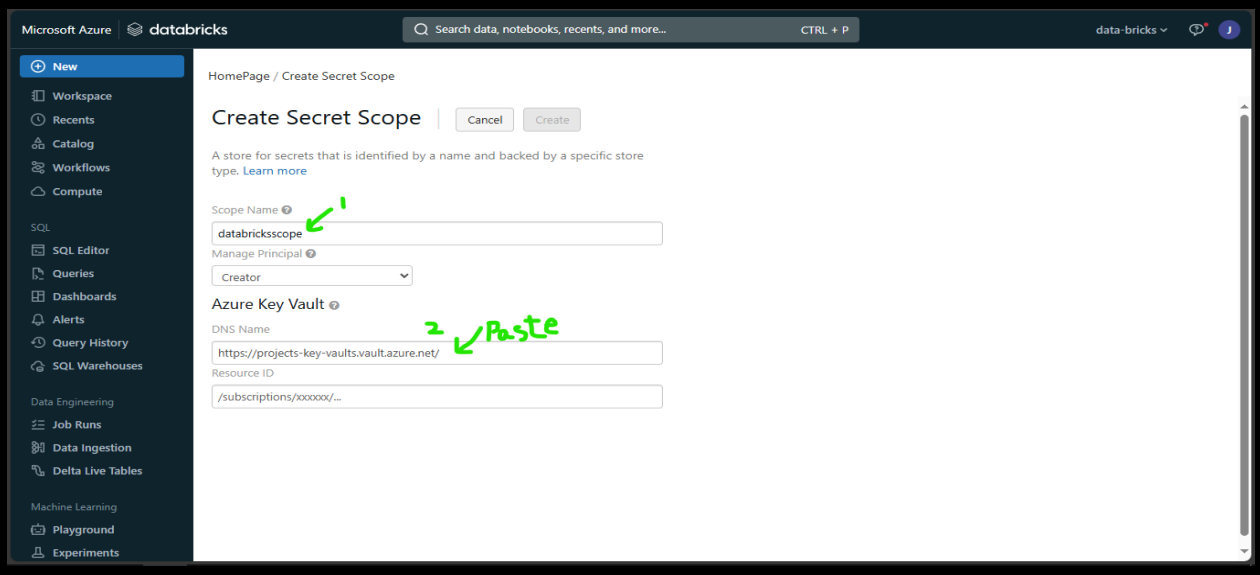
Let's create a scope in databricks.

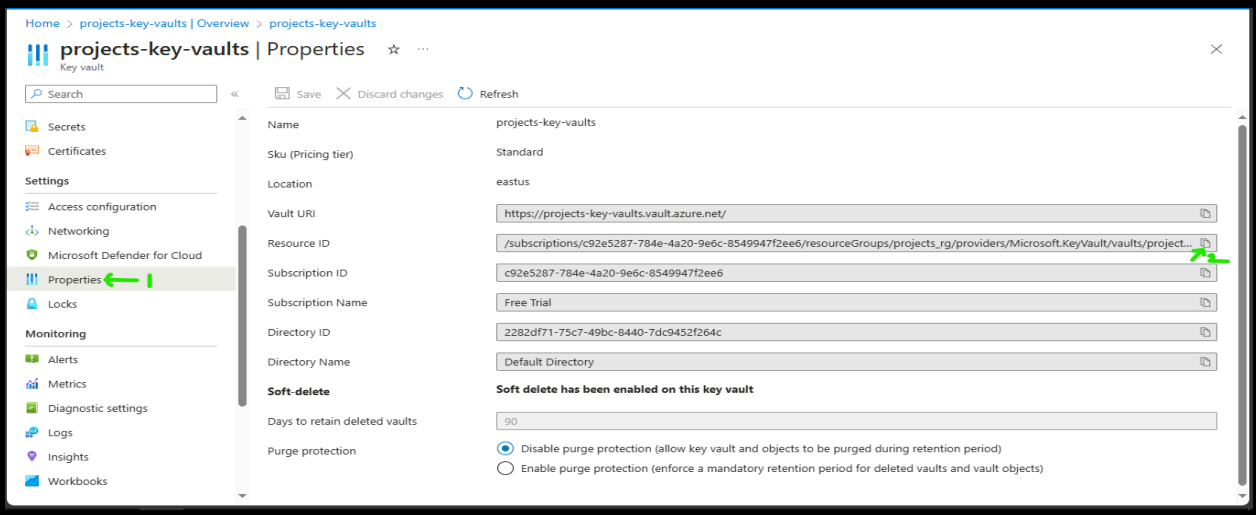
Using this scope we will import the key.

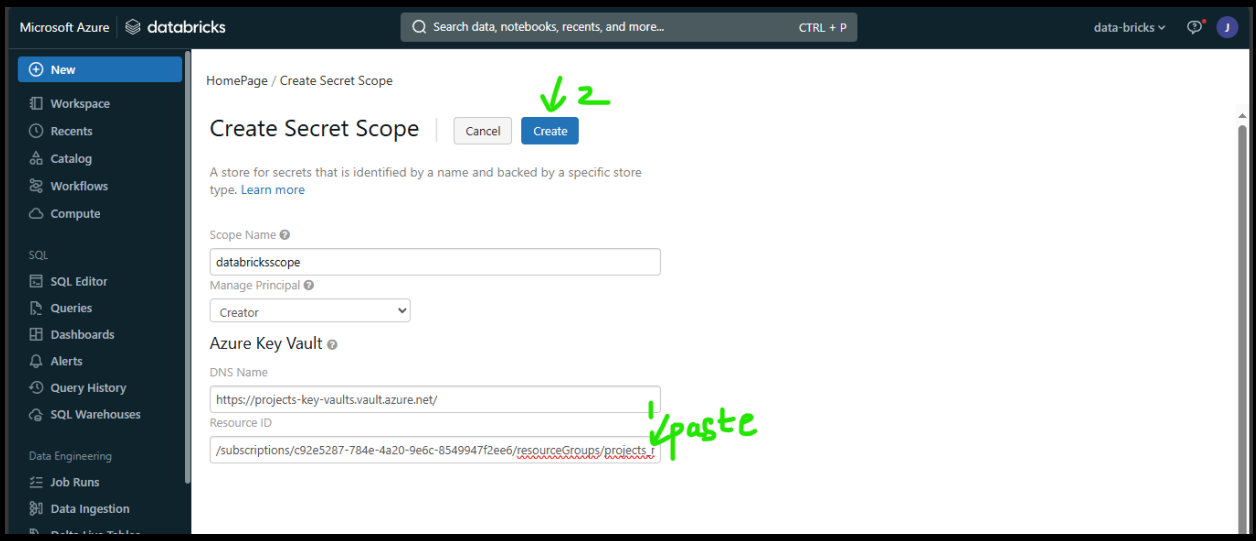
Copy the url, paste in the new tab and remove the characters after # and give secrets/createscope in the url.

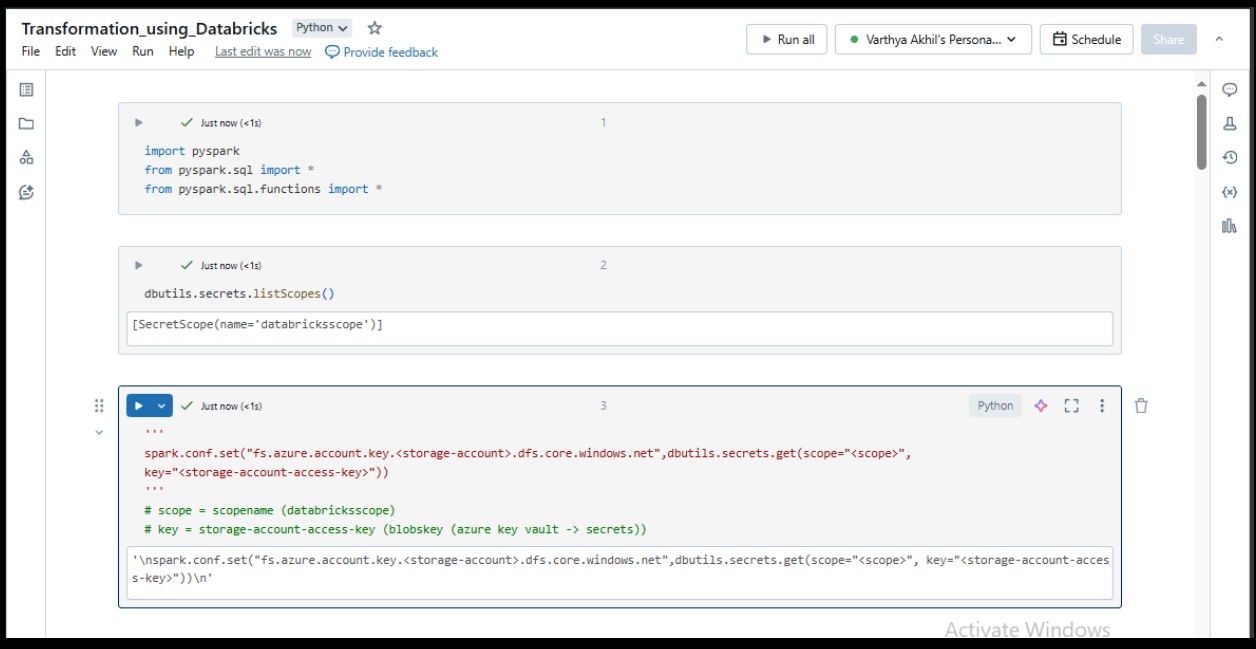

https://adb-2773184971638 54.14.azuredatabricks.net/?o=2773184971638 54#secrets/createScope
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
data-bricks
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
HomePage / Create Secret Scope
Create Secret Scope
Cancel
Create
A store for secrets that is identified by a name and backed by a specific store type. Learn more
Scope Name
Manage Principal
Creator
Azure Key Vault
DNS Name
https://xxx.vault.azure.net/
Resource ID
/subscriptions/xxxxxx/...

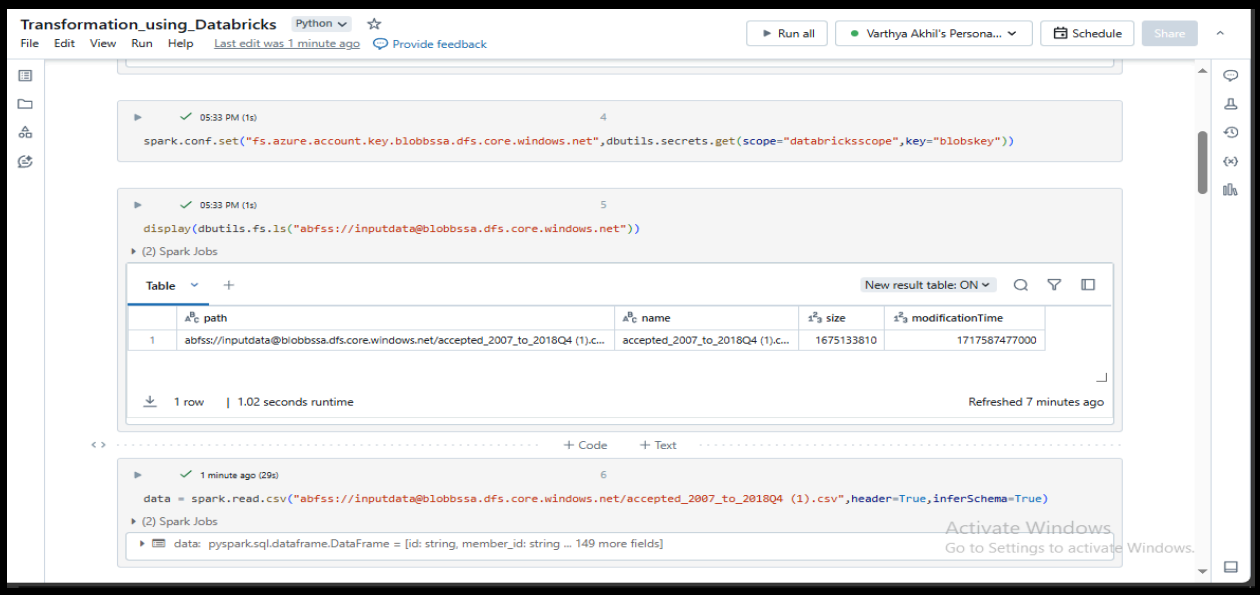

Home > projects-key-vaults | Overview > projects-key-vaults
projects-key-vaults | Properties
Key vault
Search
Save
Discard changes
Refresh
Secrets
Certificates
Settings
Access configuration
Networking
Microsoft Defender for Cloud
Properties
Locks
Monitoring
Alerts
Metrics
Diagnostic settings
Logs
Insights
Workbooks
Name
projects-key-vaults
Sku (Pricing tier)
Standard
Location
eastus
Vault URI
https://projects-key-vaults.vault.azure.net/
Resource ID
/subscriptions/c92e5287-784e-4a20-9e6c-8549947f2ee6/resourceGroups/projects_rg/providers/Microsoft.KeyVault/vaults/project...
Subscription ID
c92e5287-784e-4a20-9e6c-8549947f2ee6
Subscription Name
Free Trial
Directory ID
2282df71-75c7-49bc-8440-7dc9452f264c
Directory Name
Default Directory
Soft-delete
Soft delete has been enabled on this key vault
Days to retain deleted vaults
90
Purge protection
Disable purge protection (allow key vault and objects to be purged during retention period)
Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)
copy
2
1


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
data-bricks
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
HomePage / Create Secret Scope
Create Secret Scope
Cancel
Create
A store for secrets that is identified by a name and backed by a specific store type. Learn more
Scope Name
databricksscope
1
Manage Principal
Creator
Azure Key Vault
DNS Name
2
Paste
https://projects-key-vaults.vault.azure.net/
Resource ID
/subscriptions/xxxxxx/...


Home > projects-key-vaults | Overview > projects-key-vaults
projects-key-vaults | Properties
Key vault
Search
Save
Discard changes
Refresh
Secrets
Certificates
Settings
Access configuration
Networking
Microsoft Defender for Cloud
Properties
Locks
Monitoring
Alerts
Metrics
Diagnostic settings
Logs
Insights
Workbooks
Name
projects-key-vaults
Sku (Pricing tier)
Standard
Location
eastus
Vault URI
https://projects-key-vaults.vault.azure.net/
Resource ID
/subscriptions/c92e5287-784e-4a20-9e6c-8549947f2ee6/resourceGroups/projects_rg/providers/Microsoft.KeyVault/vaults/project...
Subscription ID
c92e5287-784e-4a20-9e6c-8549947f2ee6
Subscription Name
Free Trial
Directory ID
2282df71-75c7-49bc-8440-7dc9452f264c
Directory Name
Default Directory
Soft-delete
Soft delete has been enabled on this key vault
Days to retain deleted vaults
90
Purge protection
Disable purge protection (allow key vault and objects to be purged during retention period)
Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

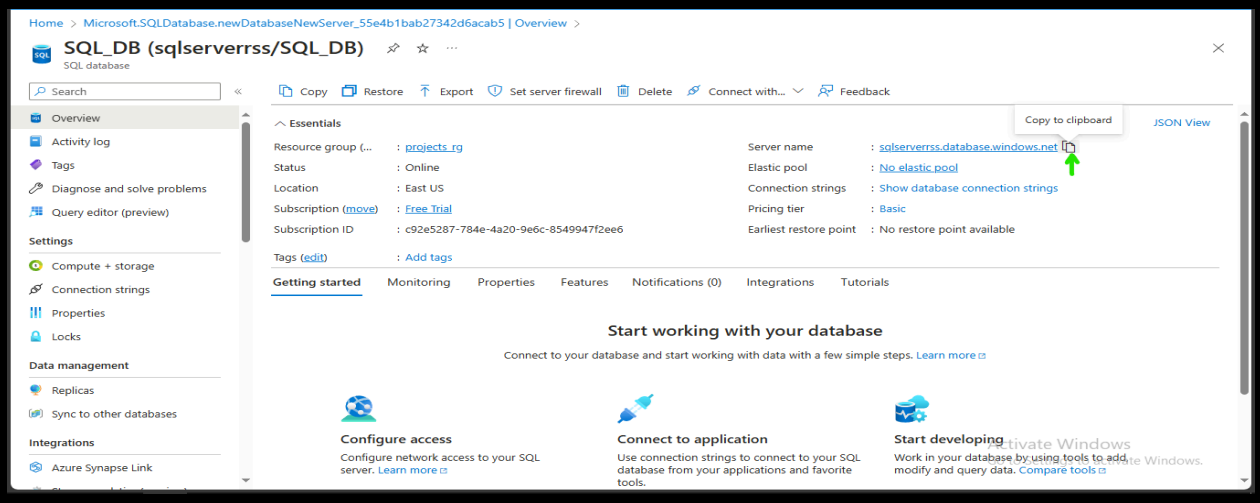

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
data-bricks
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
HomePage / Create Secret Scope
Create Secret Scope
Cancel
Create
A store for secrets that is identified by a name and backed by a specific store type. Learn more
Scope Name
databricksscope
Manage Principal
Creator
Azure Key Vault
DNS Name
https://projects-key-vaults.vault.azure.net/
Resource ID
/subscriptions/c92e5287-784e-4a20-9e6c-8549947f2ee6/resourceGroups/projects_r
paste


Transformation_using_Databricks
Python
File Edit View Run Help
Last edit was now
Provide feedback
Run all
Varthya Akhil's Persona...
Schedule
Share
Just now (<1s)
1
import pyspark
from pyspark.sql import *
from pyspark.sql.functions import *
Just now (<1s)
2
dbutils.secrets.listScopes()
[SecretScope(name='databricksscope')]
Just now (<1s)
3
Python
'''
spark.conf.set("fs.azure.account.key.<storage-account>.dfs.core.windows.net",dbutils.secrets.get(scope="<scope>",
key="<storage-account-access-key>"))
'''
scope = scopename (databricksscope)
key = storage-account-access-key (blobskey (azure key vault -> secrets))
'\nspark.conf.set("fs.azure.account.key.<storage-account>.dfs.core.windows.net",dbutils.secrets.get(scope="<scope>", key="<storage-account-access-key>"))\n'
Activate Windows



Transformations are shown in the document below.
https://drive.google.com/file/d/17VodR4MyYiuFt-MtgTD1qa0wEVScSZMG/view?usp=sharing

Loading dataframe into sql database.
Boilerplate code to connect sql database.
→ jdbc_url = "jdbc:sqlserver://<your_server_name>.database.windows.net:1433;database=<your_database_name>"
Copy the servername from azure sql database overview



Paste the server name in the boilerplate code.

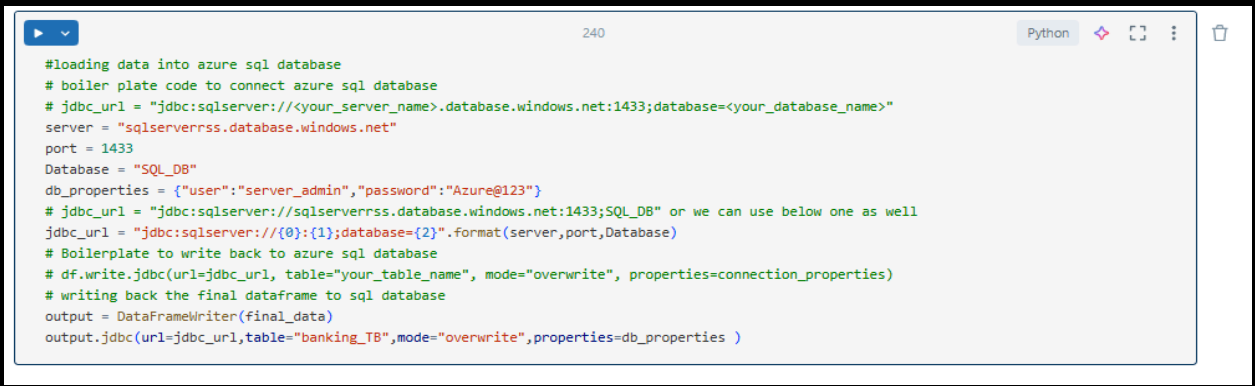
```
21
#loading data into azure sql database
# boiler plate code to connect azure sql database
# jdbc_url = "jdbc:sqlserver://<your_server_name>.database.windows.net:1433;database=<your_database_name>"
```

```
1 minute ago (<1s)   22
server = "sqlserverrss.database.windows.net"
port = 1433
Database = "SQL_DB"
db_properties = {"user":"server_admin","password":"[redacted]"}
```

```
1 minute ago (<1s)   23
# jdbc_url = "jdbc:sqlserver://sqlserverrss.database.windows.net:1433;SQL_DB" or we can use below one as well
jdbc_url = "jdbc:sqlserver://{0}:{1};database={2}".format(server,port,Database)
```

Boilerplate code to writeback to sql server database

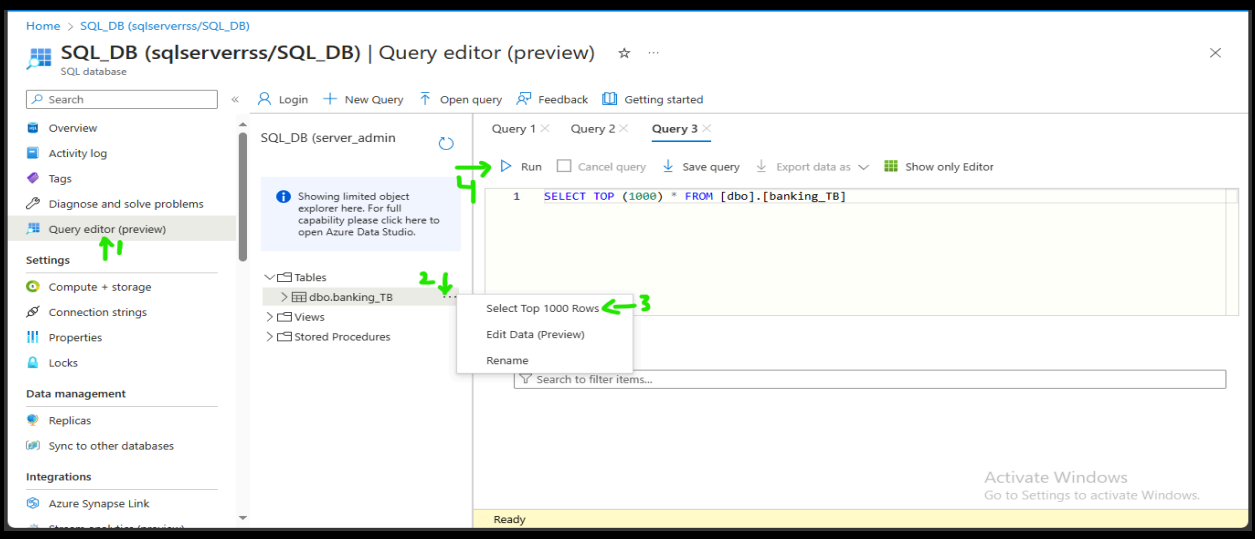
→ df.write.jdbc(url=jdbc_url, table="your_table_name", mode="overwrite", properties=connection_properties)

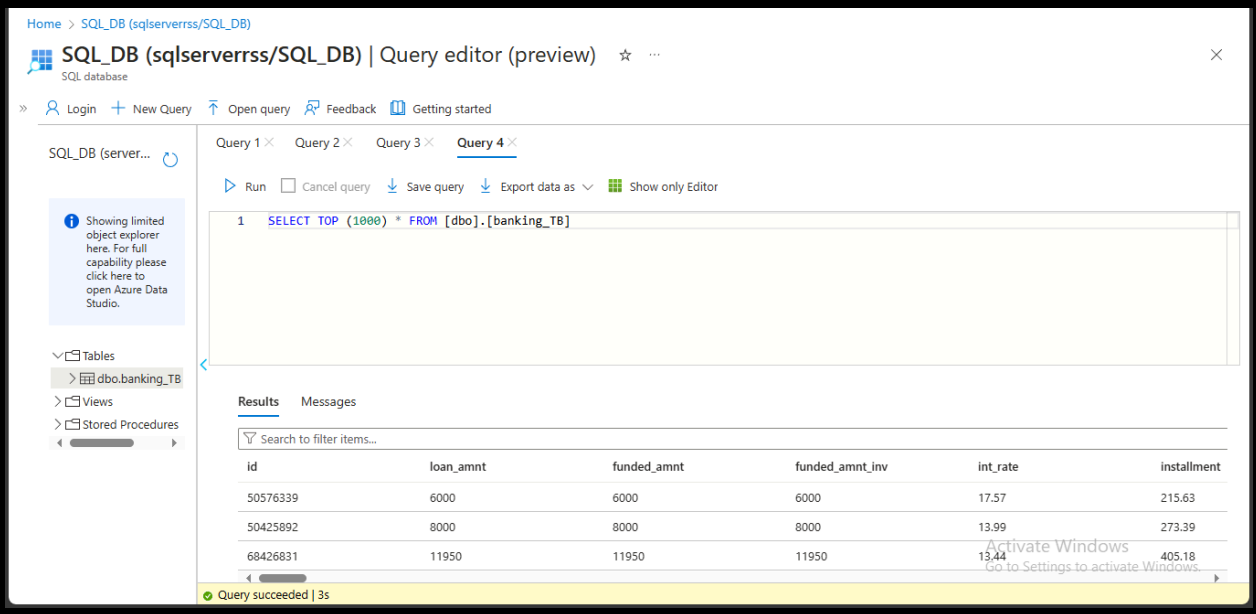


```
240   Python
#loading data into azure sql database
# boiler plate code to connect azure sql database
# jdbc_url = "jdbc:sqlserver://<your_server_name>.database.windows.net:1433;database=<your_database_name>"
server = "sqlserverrss.database.windows.net"
port = 1433
Database = "SQL_DB"
db_properties = {"user":"server_admin","password":"Azure@123"}
# jdbc_url = "jdbc:sqlserver://sqlserverrss.database.windows.net:1433;SQL_DB" or we can use below one as well
jdbc_url = "jdbc:sqlserver://{0}:{1};database={2}".format(server,port,Database)
# Boilerplate to write back to azure sql database
# df.write.jdbc(url=jdbc_url, table="your_table_name", mode="overwrite", properties=connection_properties)
# writing back the final dataframe to sql database
output = DataFrameWriter(final_data)
output.jdbc(url=jdbc_url,table="banking_TB",mode="overwrite",properties=db_properties )
```

Now data has been successfully loaded into an azure sql database.

Lets see data in azure sql database by querying .




Home > SQL_DB (sqlserverrss/SQL_DB)
SQL_DB (sqlserverrss/SQL_DB) | Query editor (preview)
SQL database
Login
New Query
Open query
Feedback
Getting started
SQL_DB (server...
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
dbo.banking_TB
Views
Stored Procedures
Query 1
Query 2
Query 3
Query 4
Run
Cancel query
Save query
Export data as
Show only Editor
1 SELECT TOP (1000) * FROM [dbo].[banking_TB]
Results
Messages
Search to filter items...
id
loan_amnt
funded_amnt
funded_amnt_inv
int_rate
installment
50576339
6000
6000
6000
17.57
215.63
50425892
8000
8000
8000
13.99
273.39
68426831
11950
11950
11950
13.44
405.18
Activate Windows
Go to Settings to activate Windows.
Query succeeded | 3s