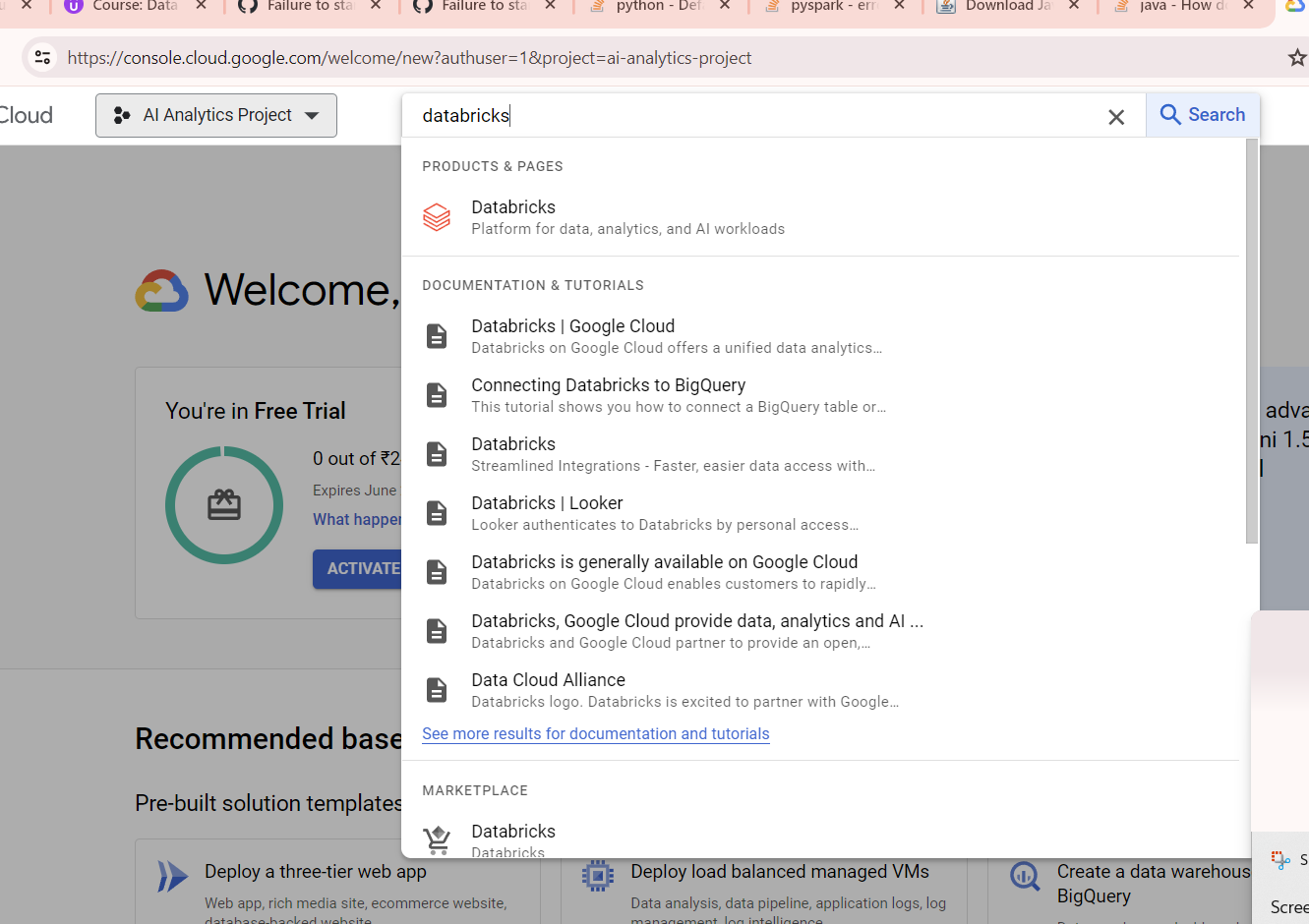
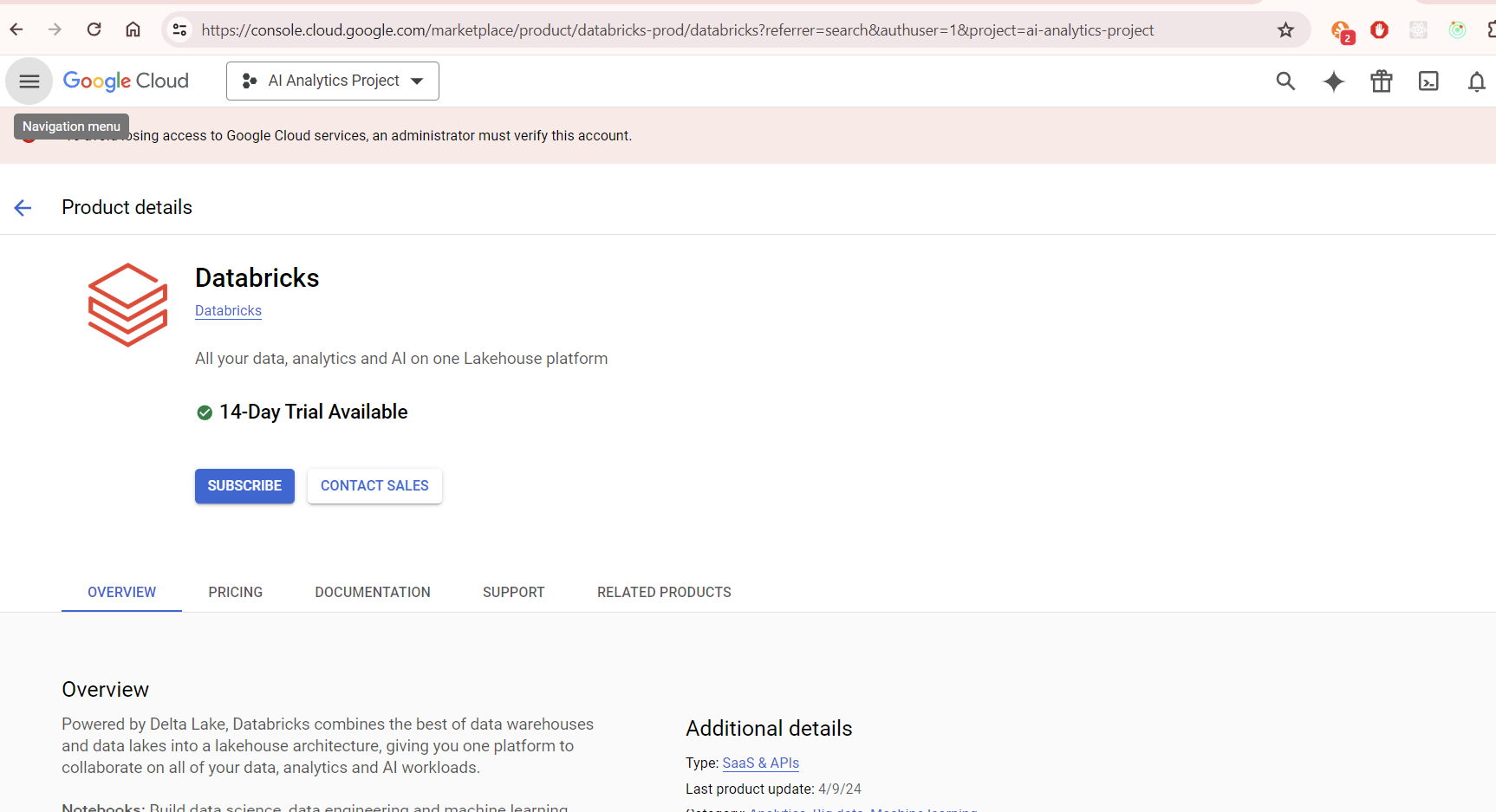
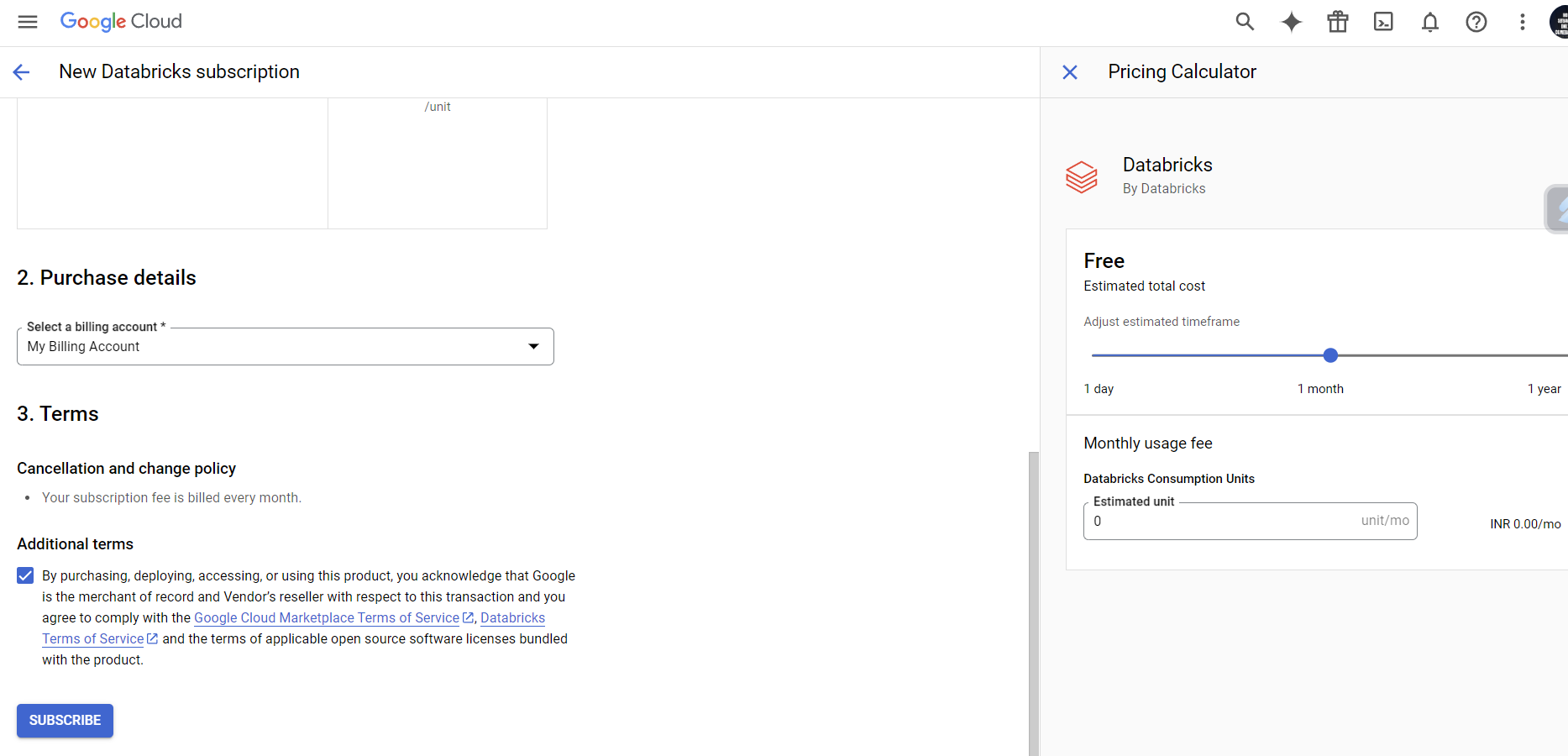
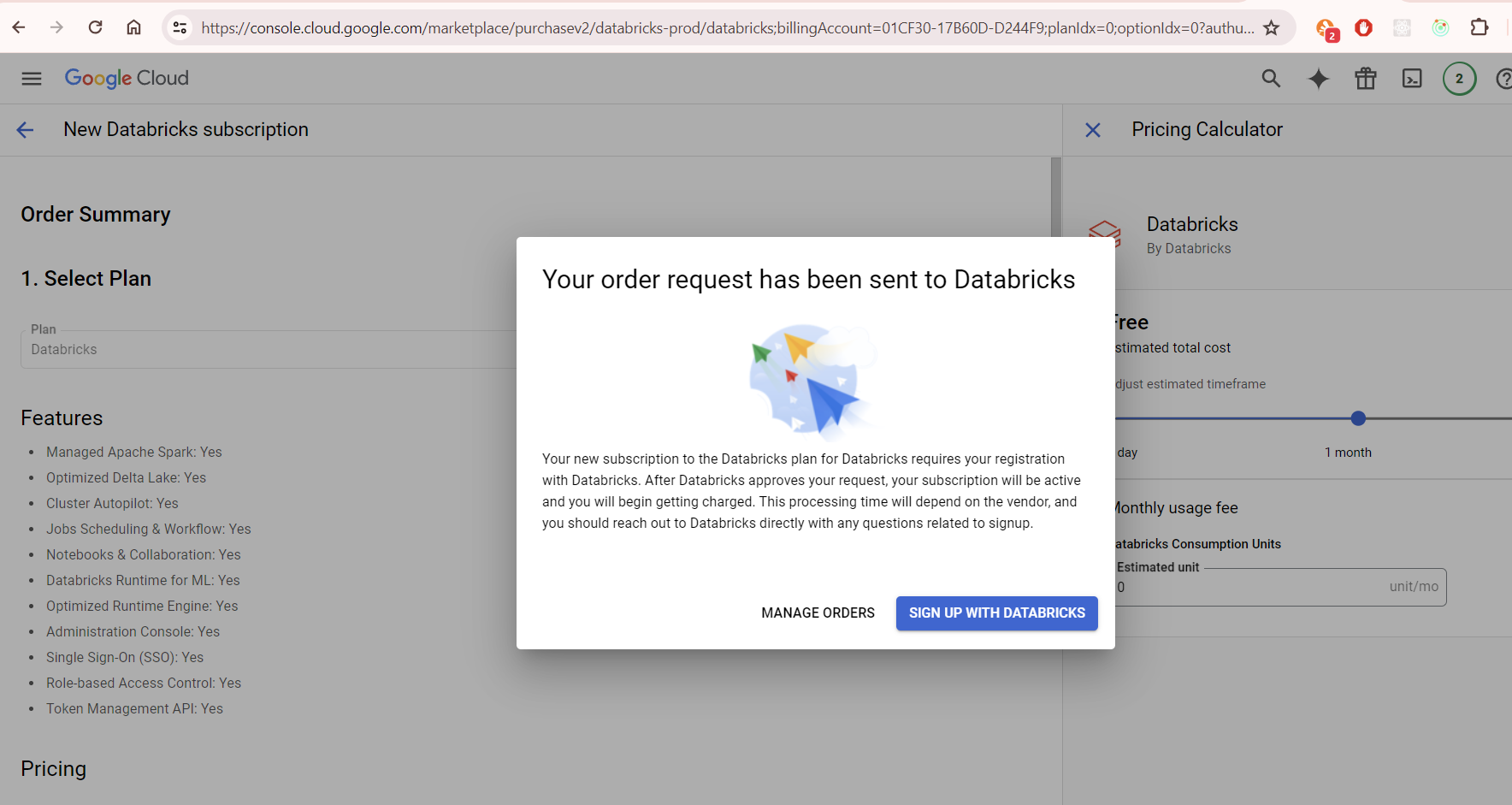
**Databricks** is an industry-leading, cloud-based data engineering tool used for processing and transforming massive quantities of data and exploring the data through machine learning models.





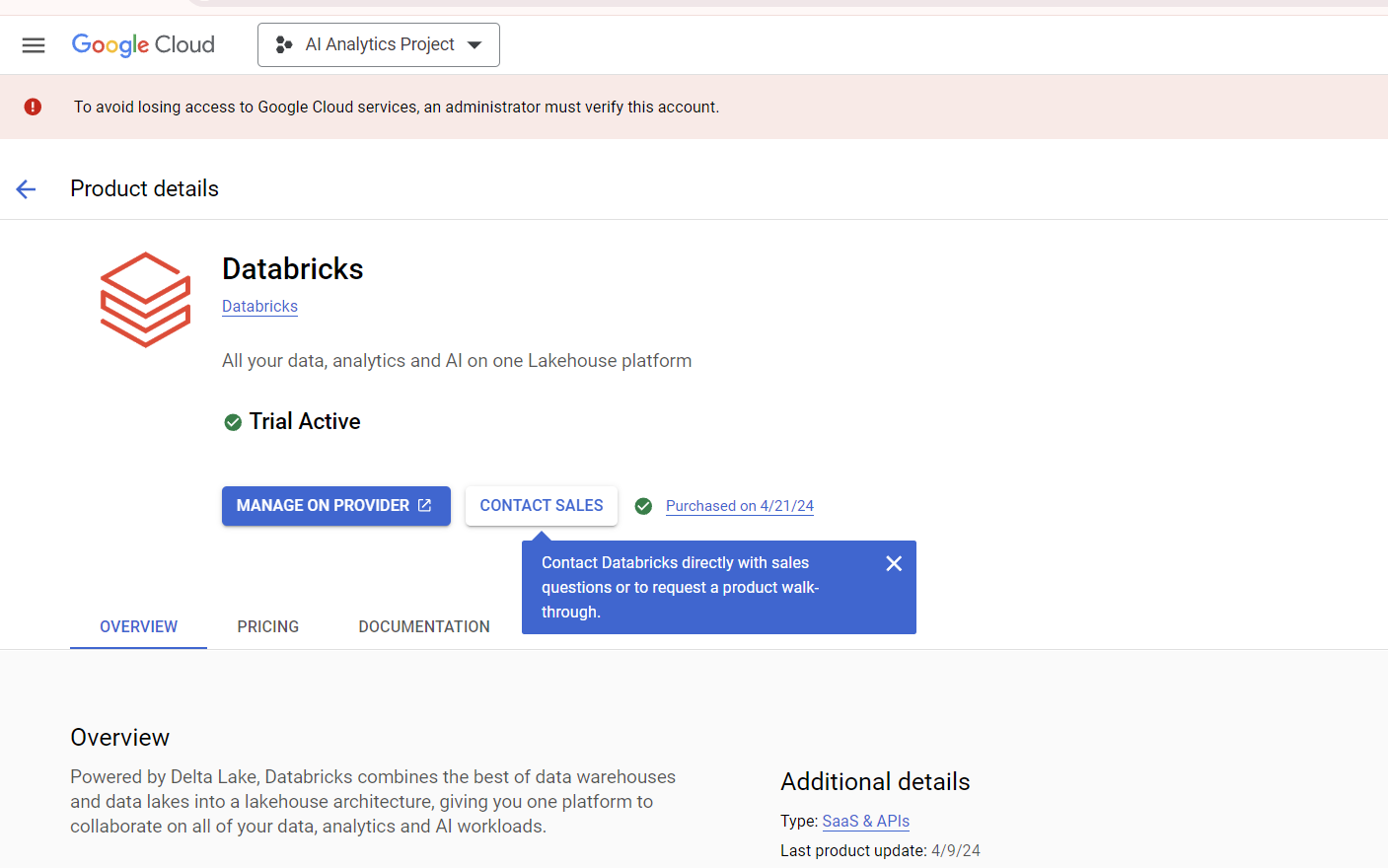


Again subscribe

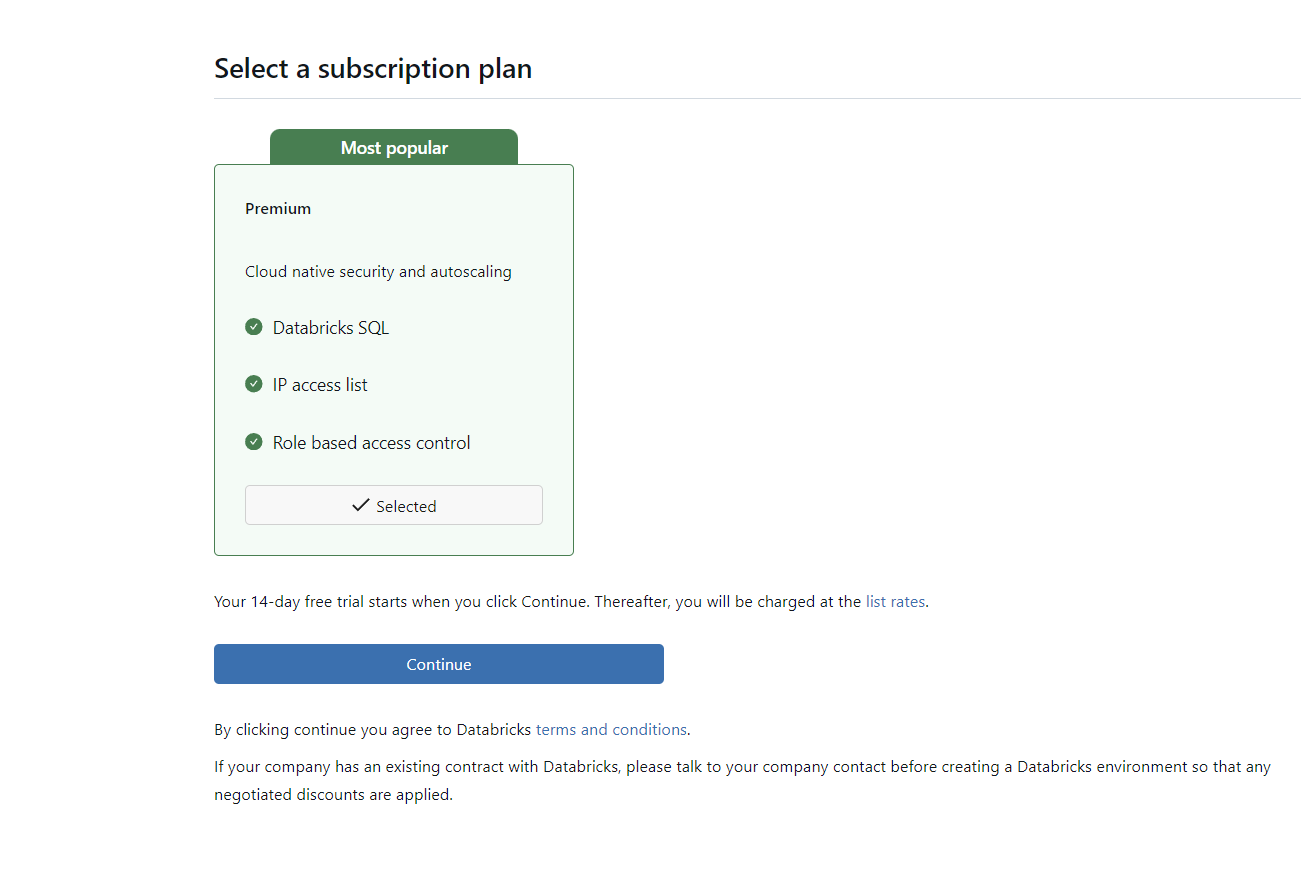


Sign up

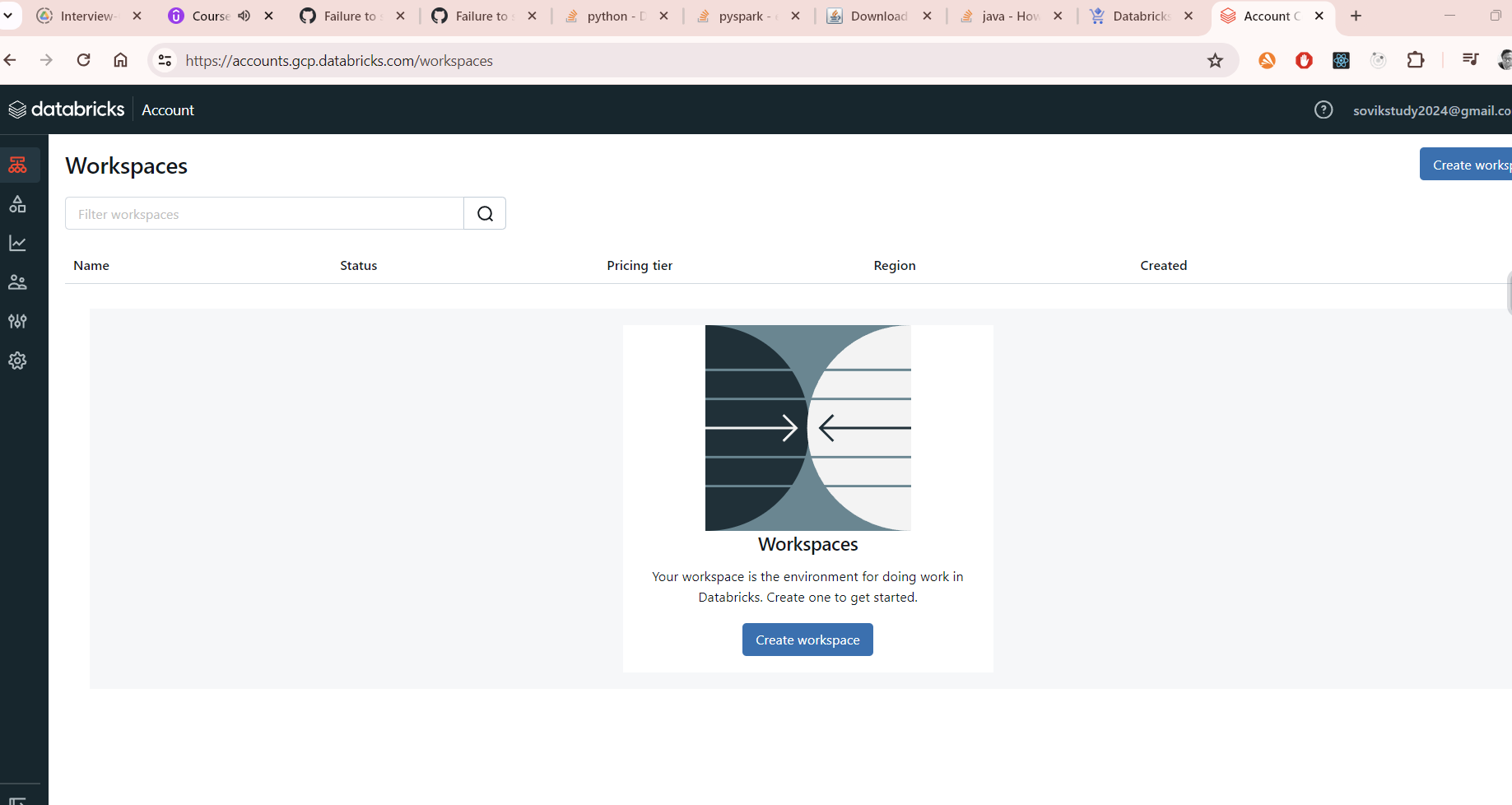
Now after that Manage the mother page

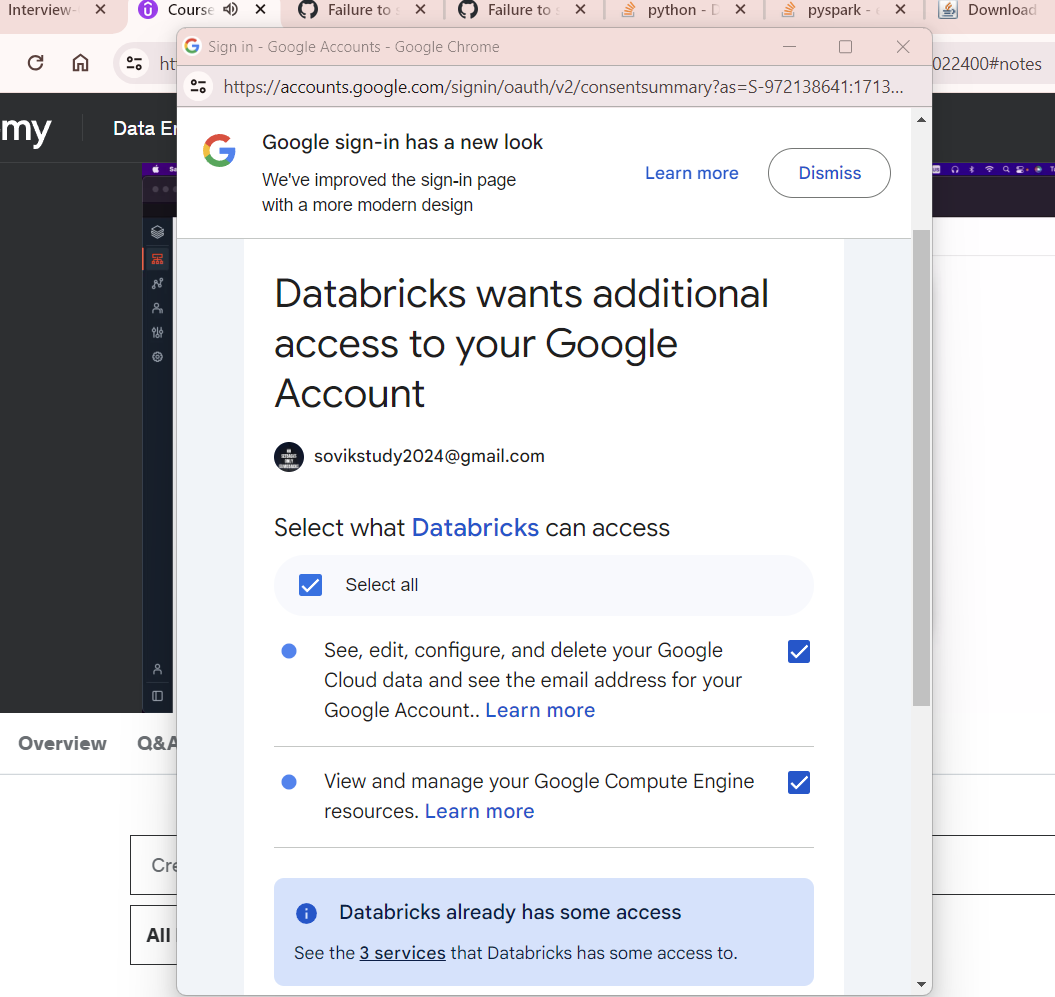


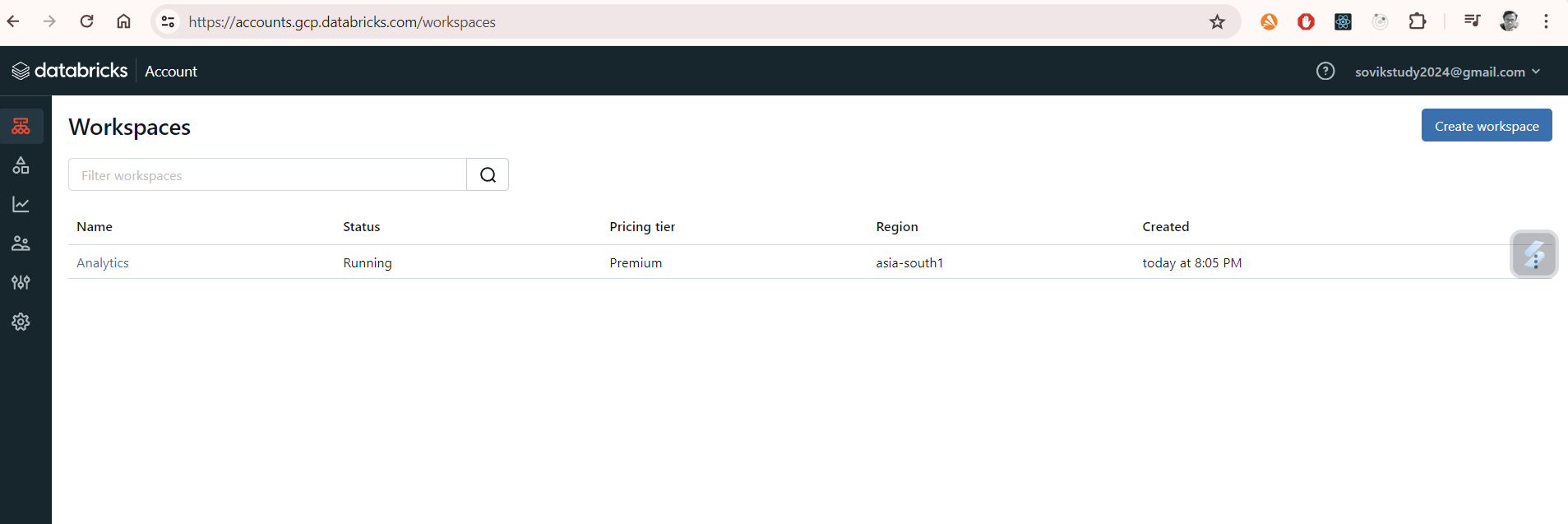
Click on Manage on Provider



Create Workspace

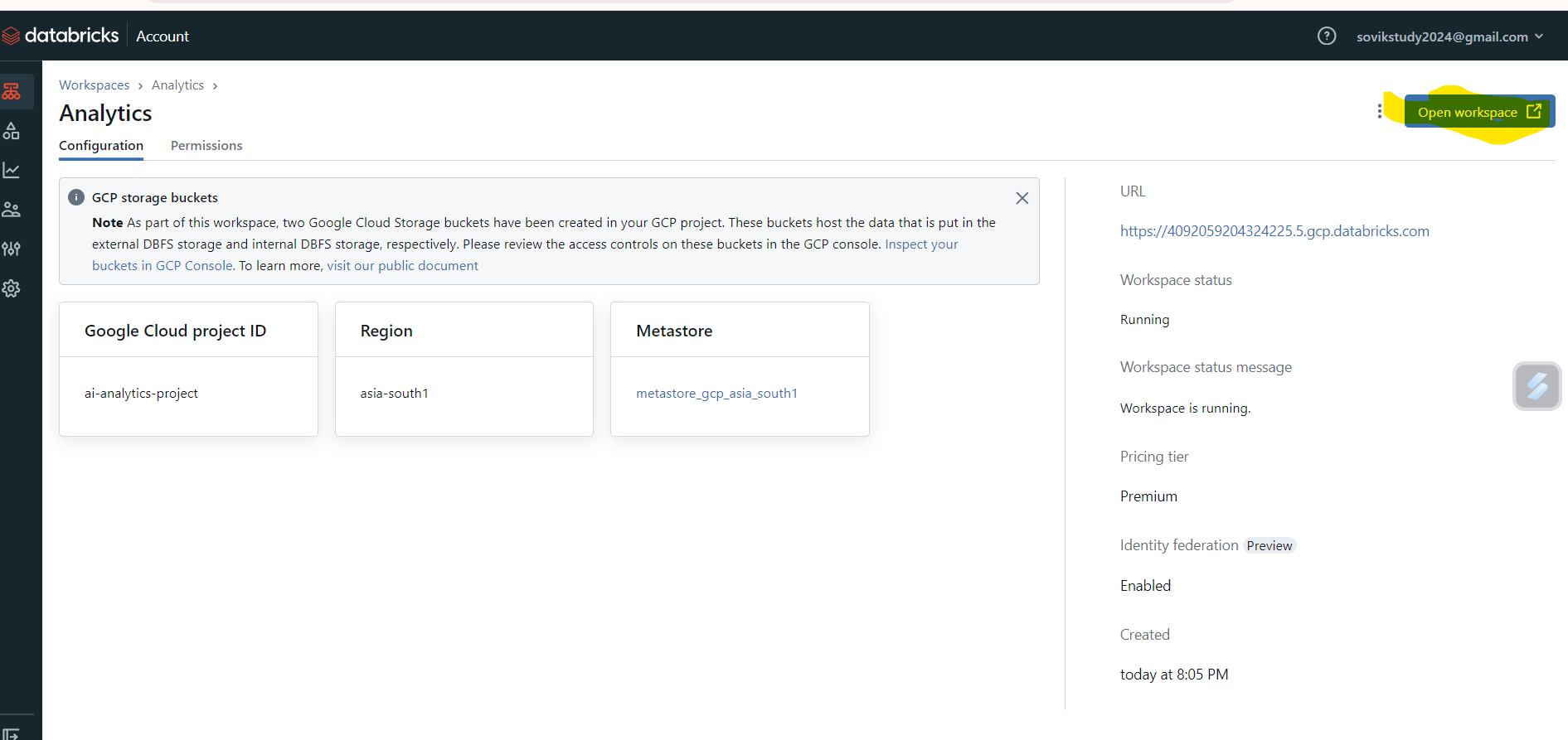


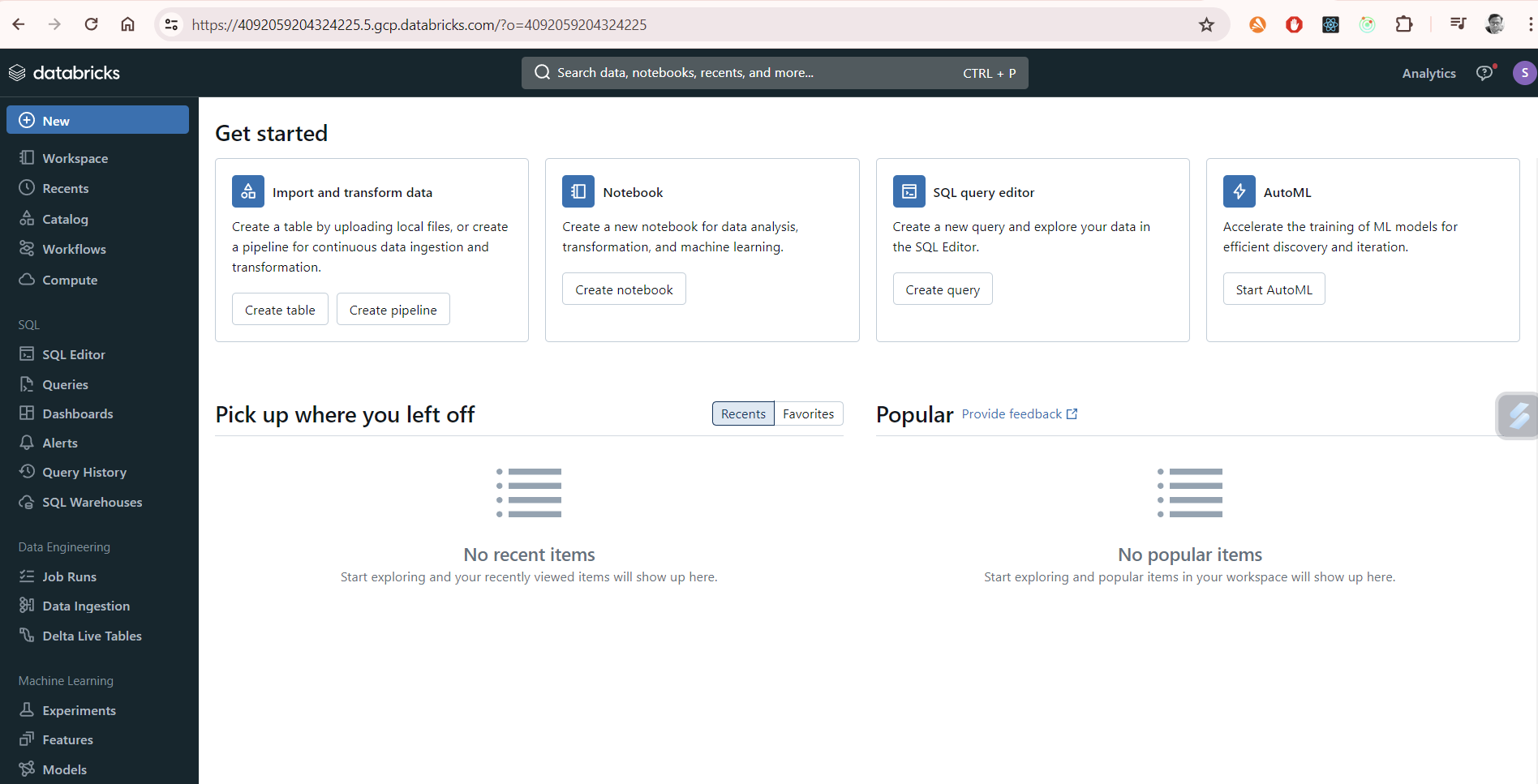




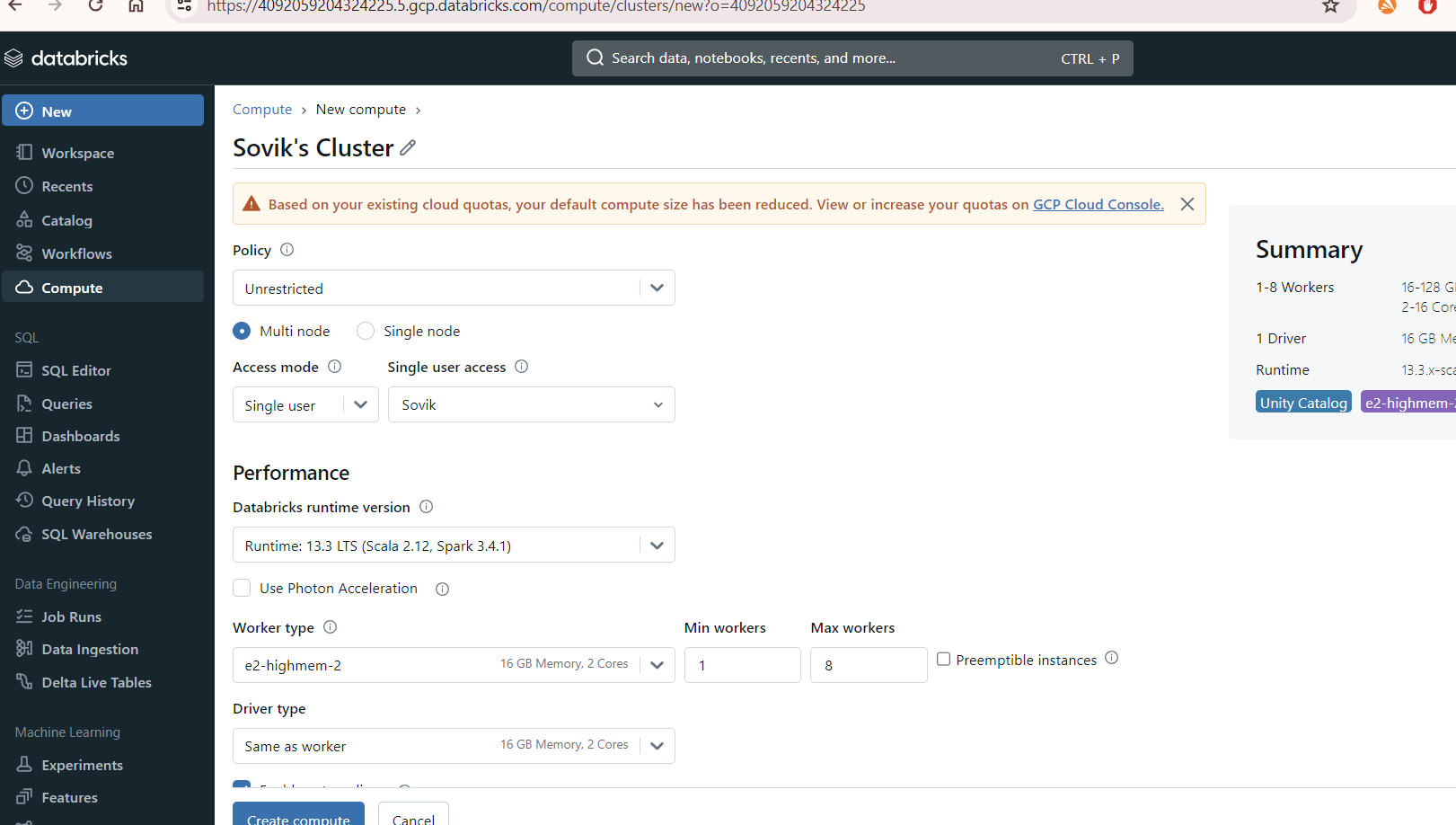
Click on View

## 

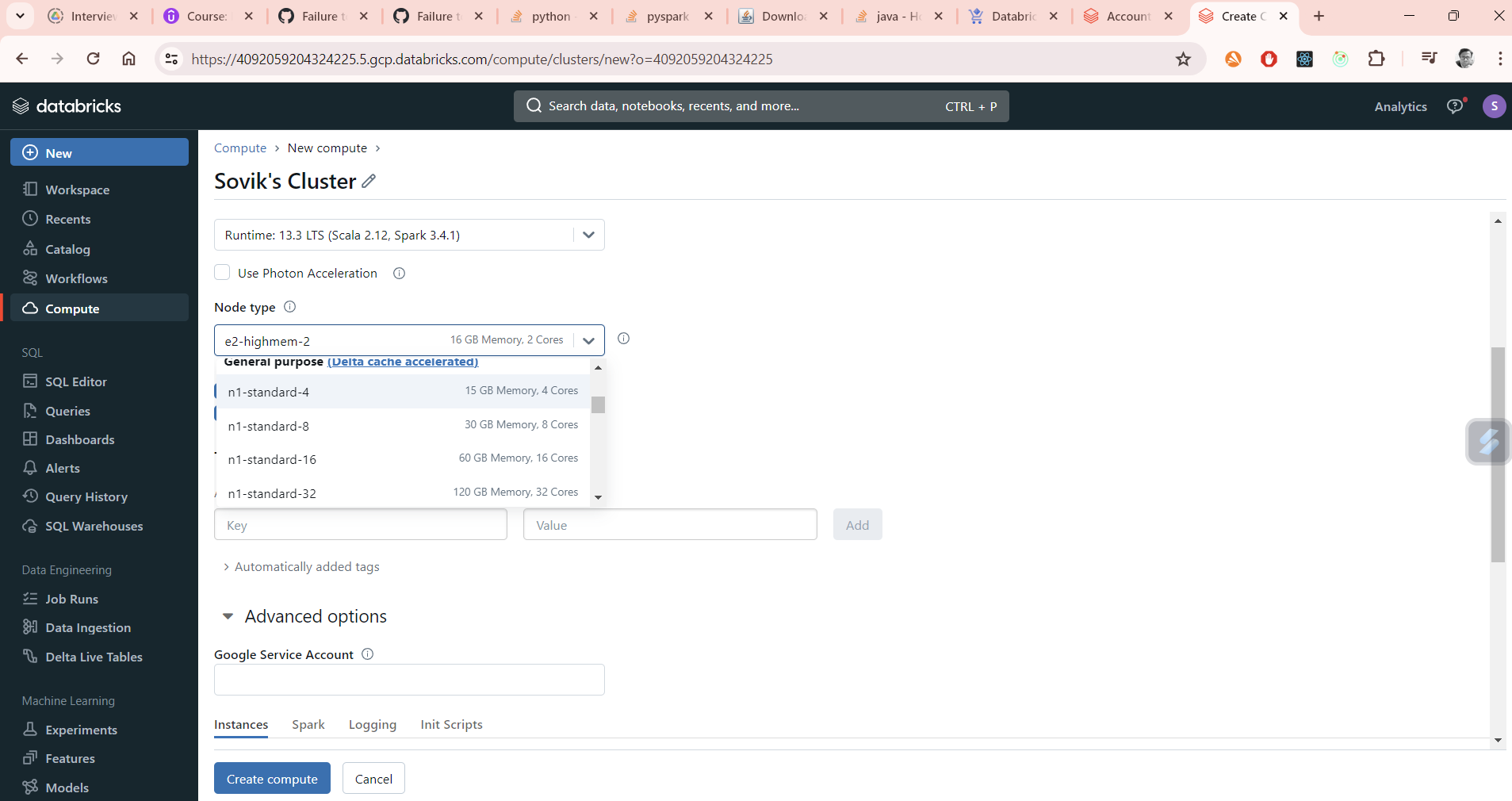


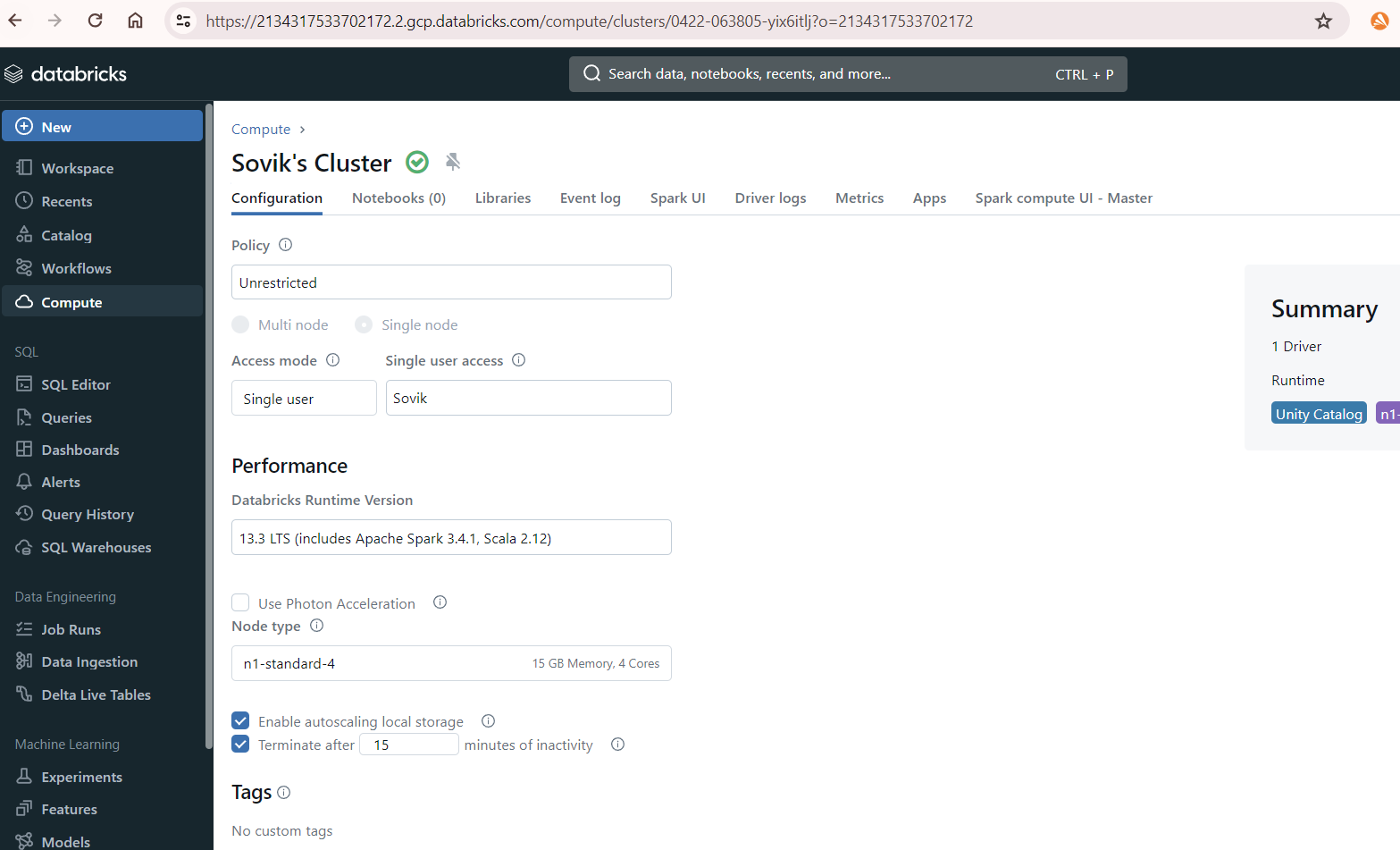


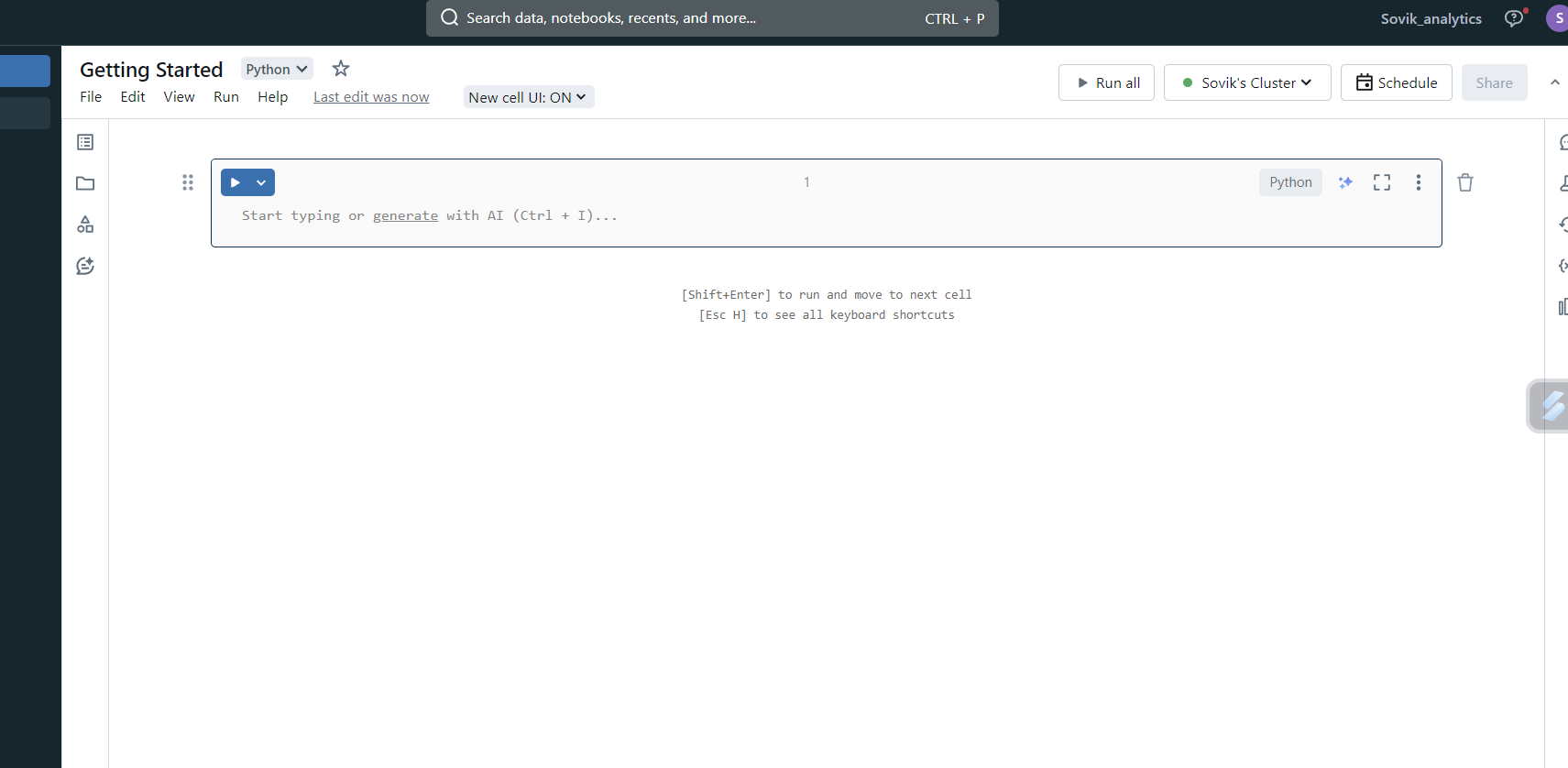
New->Cluster

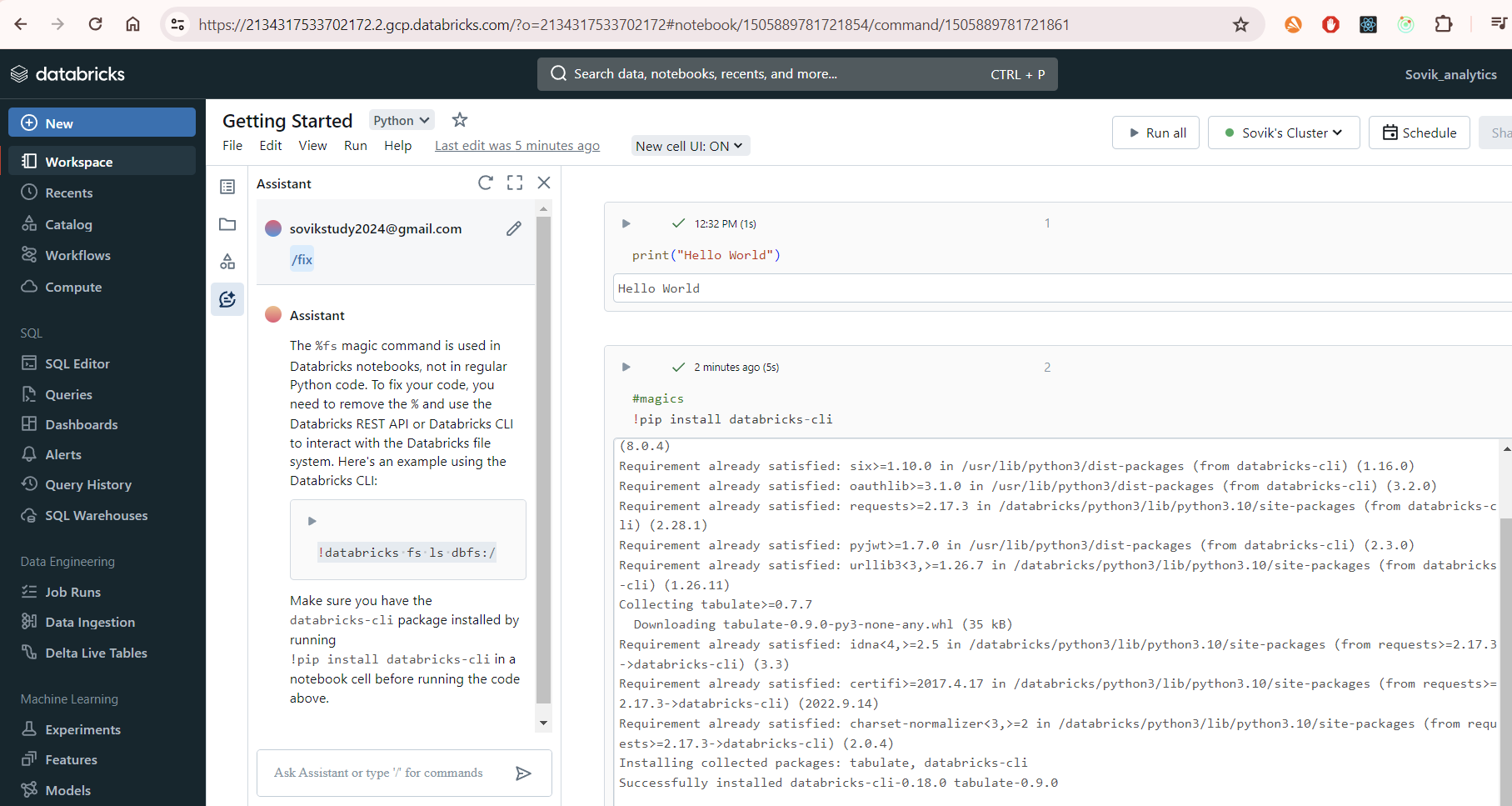


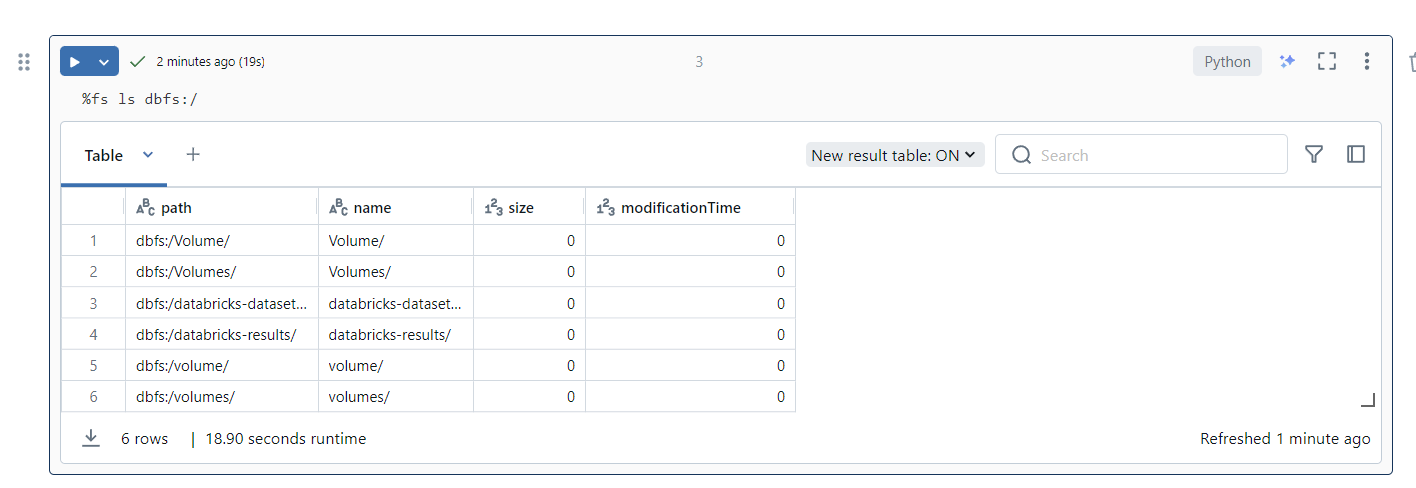
Single node:











**All-purpose clusters**, such as ad hoc analysis, data exploration, and development, are designed for collaborative use. Multiple users can share them. On the other hand, **job clusters** are specifically for running automated jobs. They terminate once the job is completed, reducing resource usage and cost.

A **pool** consists of both idle instances kept ready for new clusters and instances in use by running clusters. All of these instances are of the same instance provider type, selected when creating a pool. A pool's instance type cannot be edited.

Features:

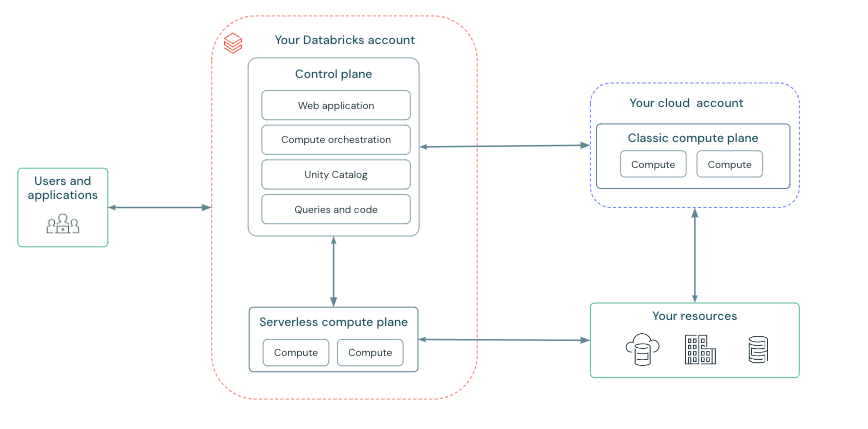
* A unified platform for data, analytics and AI.
* Data reliability, security and performance.
* Sharing. An open, secure, zero-copy sharing for all data.
* ETL and orchestration for batch and streaming data.
* Governance. ...
* Real-time analytics, AI and applications made simple.
* Artificial Intelligence. ...
* Data Engineering.

## **High-level architecture**

Databricks operates out of a **control plane** and a **compute plane**.

* The **control plane** includes the backend services that Databricks manages in your Databricks account. The web application is in the control plane.
* The **compute plane** is where your data is processed. There are two types of compute planes depending on the compute that you are using.
  + For serverless compute, the serverless compute resources run in a serverless compute plane in your Databricks account.
  + For classic Databricks compute, the compute resources are in your AWS account in what is called the classic compute plane. This refers to the network in your AWS account and its resources.

The following diagram describes the overall Databricks architecture.



## **Serverless compute plane**

In the serverless compute plane, Databricks compute resources run in a compute layer within your Databricks account. Databricks creates a serverless compute plane in the same AWS region as your workspace’s classic compute plane.

To protect customer data within the serverless compute plane, serverless compute runs within a network boundary for the workspace, with various layers of security to isolate different Databricks customer workspaces and additional network controls between clusters of the same customer.

To learn more about networking in the serverless compute plane, [Serverless compute plane networking](https://docs.databricks.com/en/security/network/serverless-network-security/index.html).

## **Classic compute plane**

In the classic compute plane, Databricks compute resources run in your AWS account. New compute resources are created within each workspace’s virtual network in the customer’s AWS account.

A classic compute plane has natural isolation because it runs in each customer’s own AWS account. To learn more about networking in the classic compute plane, see [Classic compute plane networking](https://docs.databricks.com/en/security/network/classic/index.html).

For regional support, see [Databricks clouds and regions](https://docs.databricks.com/en/resources/supported-regions.html).

Was this article helpful?

**In VS code:**

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE> **cd Databricks\_demo**

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> **pip install databricks-cli**

Collecting databricks-cli

Downloading databricks\_cli-0.18.0-py2.py3-none-any.whl.metadata (4.0 kB)

Requirement already satisfied: click>=7.0 in c:\data engineering\de udemy\python\_for\_de\.conda\lib\site-packages (from databricks-cli) (8.1.7)

Collecting pyjwt>=1.7.0 (from databricks-cli)

Downloading PyJWT-2.8.0-py3-none-any.whl.metadata (4.2 kB)

Collecting oauthlib>=3.1.0 (from databricks-cli)

Downloading oauthlib-3.2.2-py3-none-any.whl.metadata (7.5 kB)

Collecting requests>=2.17.3 (from databricks-cli)

Downloading requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)

Collecting tabulate>=0.7.7 (from databricks-cli)

Downloading tabulate-0.9.0-py3-none-any.whl.metadata (34 kB)

Requirement already satisfied: six>=1.10.0 in c:\users\sovik\appdata\roaming\python\python310\site-packages (from databricks-cli) (1.16.0)

Requirement already satisfied: urllib3<3,>=1.26.7 in c:\data engineering\de udemy\python\_for\_de\.conda\lib\site-packages (from databricks-cli) (2.2.1)

Requirement already satisfied: colorama in c:\users\sovik\appdata\roaming\python\python310\site-packages (from click>=7.0->databricks-cli) (0.4.6)

Collecting charset-normalizer<4,>=2 (from requests>=2.17.3->databricks-cli)

Downloading charset\_normalizer-3.3.2-cp310-cp310-win\_amd64.whl.metadata (34 kB)

Collecting idna<4,>=2.5 (from requests>=2.17.3->databricks-cli)

Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)

Collecting certifi>=2017.4.17 (from requests>=2.17.3->databricks-cli)

Downloading certifi-2024.2.2-py3-none-any.whl.metadata (2.2 kB)

Downloading databricks\_cli-0.18.0-py2.py3-none-any.whl (150 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 150.3/150.3 kB 2.2 MB/s eta 0:00:00

Downloading oauthlib-3.2.2-py3-none-any.whl (151 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 151.7/151.7 kB 9.4 MB/s eta 0:00:00

Downloading PyJWT-2.8.0-py3-none-any.whl (22 kB)

Downloading requests-2.31.0-py3-none-any.whl (62 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.6/62.6 kB ? eta 0:00:00

Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)

Downloading certifi-2024.2.2-py3-none-any.whl (163 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 163.8/163.8 kB 9.6 MB/s eta 0:00:00

Downloading charset\_normalizer-3.3.2-cp310-cp310-win\_amd64.whl (100 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100.3/100.3 kB ? eta 0:00:00

Downloading idna-3.7-py3-none-any.whl (66 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.8/66.8 kB 3.8 MB/s eta 0:00:00

Installing collected packages: tabulate, pyjwt, oauthlib, idna, charset-normalizer, certifi, requests, databricks-cli

Successfully installed certifi-2024.2.2 charset-normalizer-3.3.2 databricks-cli-0.18.0 idna-3.7 oauthlib-3.2.2 pyjwt-2.8.0 requests-2.31.0 tabulate-0.9.0

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> databricks --version

Version 0.18.0

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> databricks --help

Usage: databricks [OPTIONS] COMMAND [ARGS]...

Options:

-v, --version 0.18.0

--debug Debug Mode. Shows full stack trace on error.

--profile TEXT CLI connection profile to use. The default profile is

"DEFAULT".

-h, --help Show this message and exit.

Commands:

cluster-policies Utility to interact with Databricks cluster policies.

clusters Utility to interact with Databricks clusters.

configure Configures host and authentication info for the CLI.

fs Utility to interact with DBFS.

groups Utility to interact with Databricks groups.

instance-pools Utility to interact with Databricks instance pools.

jobs Utility to interact with jobs.

libraries Utility to interact with libraries.

pipelines Utility to interact with Databricks Delta Live Tables

Pipelines.

repos Utility to interact with Repos.

runs Utility to interact with the jobs runs.

secrets Utility to interact with Databricks secret API.

stack [Beta] Utility to deploy and download Databricks resource

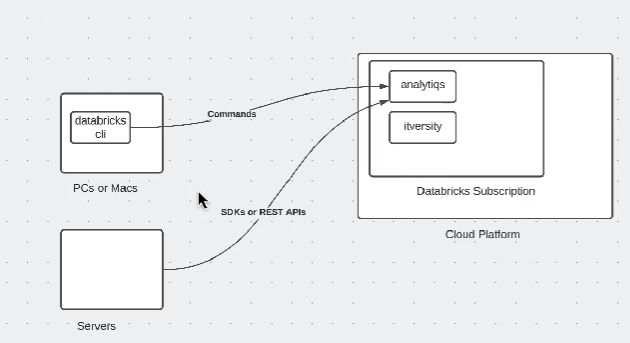
stacks.

tokens Utility to interact with Databricks tokens.

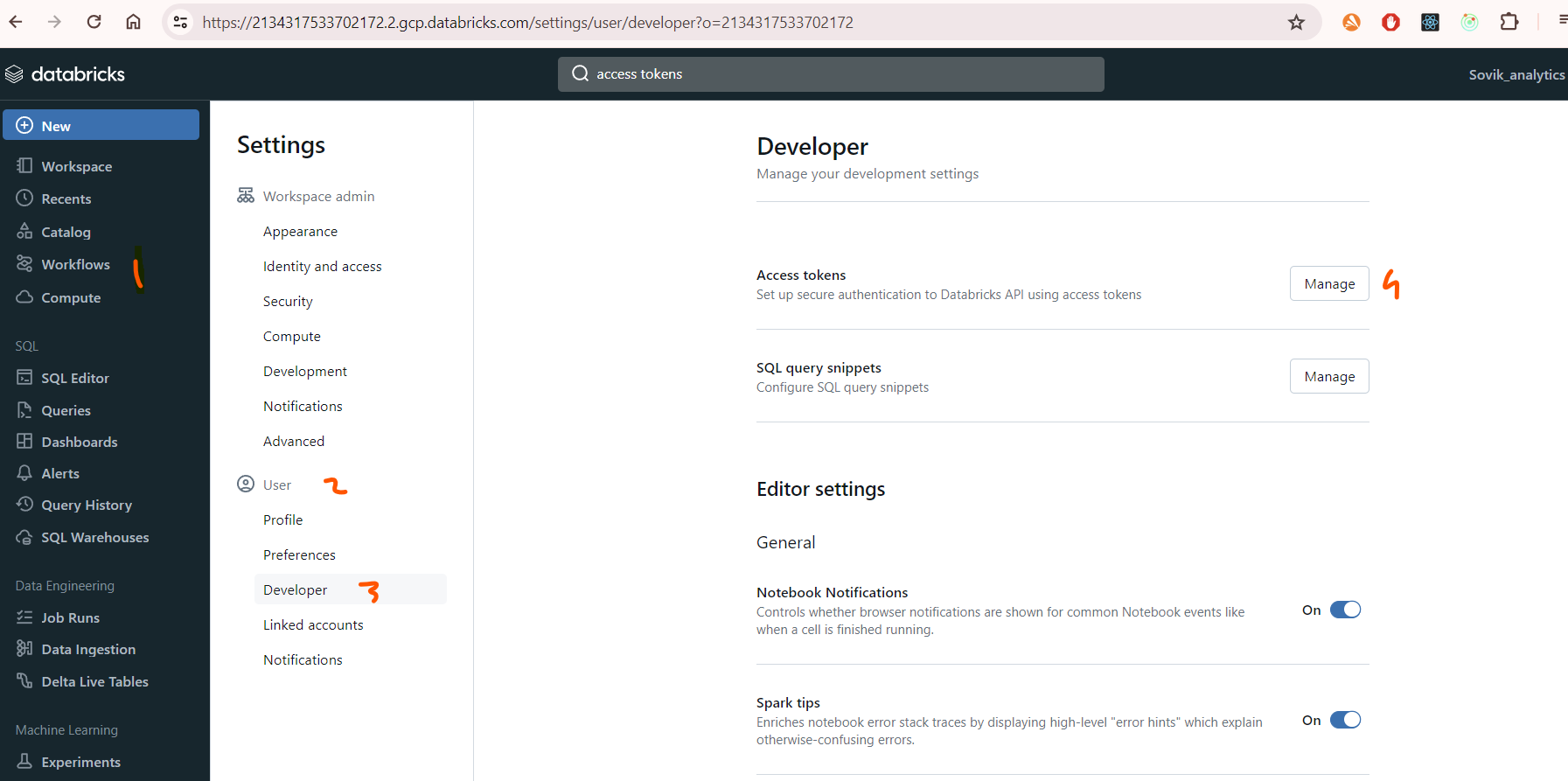
unity-catalog Utility to interact with Databricks Unity Catalog.

workspace Utility to interact with the Databricks workspace.

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo>



Acess token



Now access from vs code:

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> databricks configure --host https://2134317533702172.2.gcp.databricks.com/ --token --profile sovikstudy2024@gmail.com

Token:

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> **databricks fs ls --profile sovikstudy2024@gmail.com**

Volume

Volumes

databricks-datasets

databricks-results

volume

volumes

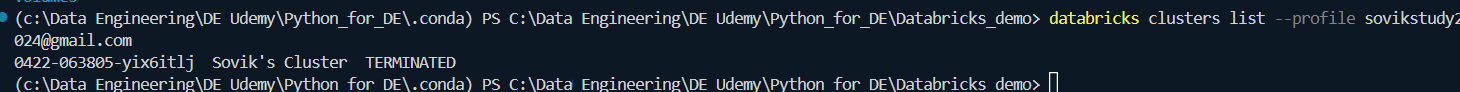
(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo>

Command to configure:

**databricks configure --host https://2134317533702172.2.gcp.databricks.com/ --token --profile sovikstudy2024@gmail.com**

Token: **dapi9343092366c5e19cec14a05a20ed0065**

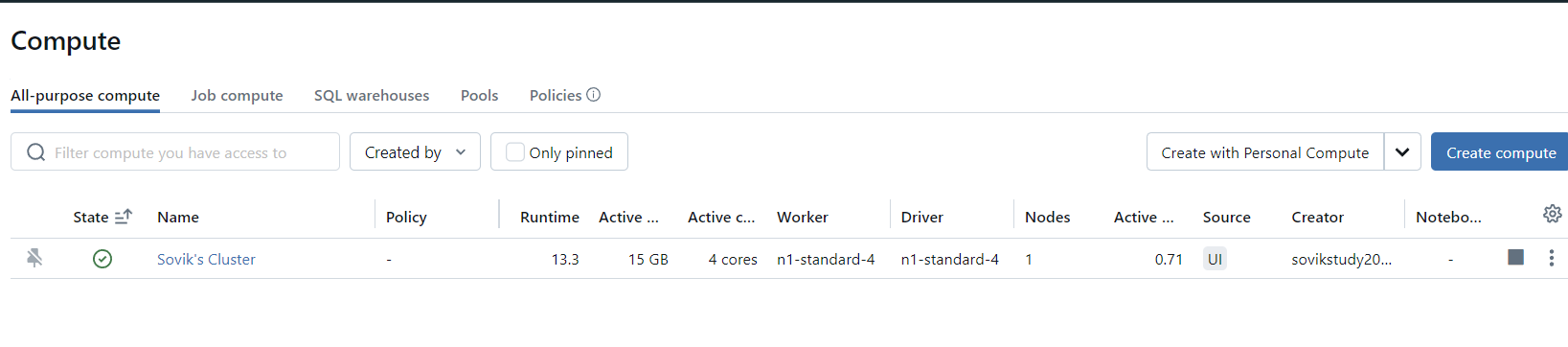
Cli -commands for Databricks:



Command: databricks clusters list --profile sovikstudy2024@gmail.com

To start the cluster;

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo**> databricks clusters start --cluster-id 0422-063805-yix6itlj --profile** [**sovikstudy2024@gmail.com**](mailto:sovikstudy2024@gmail.com)



To delete:

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> **databricks clusters delete --cluster-id 0422-063805-yix6itlj --profile** [**sovikstudy2024@gmail.com**](mailto:sovikstudy2024@gmail.com)

Does not work for gcp but we can delete from the GUI

Copy folder from local:

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> **databricks fs cp --recursive data/retail\_db dbfs:/public/retail\_db --profile sovikstudy2024@gmail.com**

data/retail\_db\categories\part-00000 -> dbfs:/public/retail\_db/categories/part-00000

data/retail\_db\create\_db.sql -> dbfs:/public/retail\_db/create\_db.sql

data/retail\_db\create\_db\_tables\_pg.sql -> dbfs:/public/retail\_db/create\_db\_tables\_pg.sql

data/retail\_db\customers\part-00000 -> dbfs:/public/retail\_db/customers/part-00000

data/retail\_db\departments\part-00000 -> dbfs:/public/retail\_db/departments/part-00000

data/retail\_db\load\_db\_tables\_pg.sql -> dbfs:/public/retail\_db/load\_db\_tables\_pg.sql

data/retail\_db\orders\part-00000 -> dbfs:/public/retail\_db/orders/part-00000

data/retail\_db\orders\_csv\_op -> dbfs:/public/retail\_db/orders\_csv\_op

data/retail\_db\orders\_json\_op -> dbfs:/public/retail\_db/orders\_json\_op

data/retail\_db\order\_items\part-00000 -> dbfs:/public/retail\_db/order\_items/part-00000

data/retail\_db\products\part-00000 -> dbfs:/public/retail\_db/products/part-00000

data/retail\_db\README.md -> dbfs:/public/retail\_db/README.md

data/retail\_db\schemas.json -> dbfs:/public/retail\_db/schemas.json

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo>

Check if the files are uploaded:

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo> **databricks fs ls dbfs:/public/retail\_db --profile sovikstudy2024@gmail.com**

README.md

categories

create\_db.sql

create\_db\_tables\_pg.sql

customers

departments

load\_db\_tables\_pg.sql

order\_items

orders

orders\_csv\_op

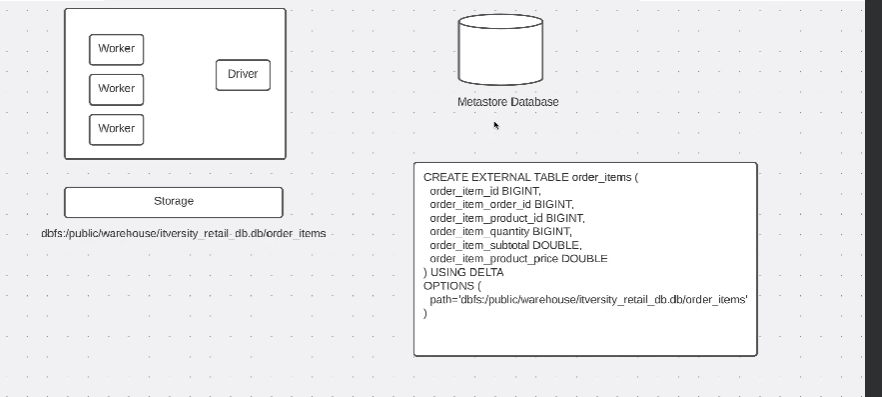
orders\_json\_op

products

schemas.json

(c:\Data Engineering\DE Udemy\Python\_for\_DE\.conda) PS C:\Data Engineering\DE Udemy\Python\_for\_DE\Databricks\_demo

Spark overview:



### Managed Tables

1. **Definition**: Managed tables (also known as internal tables) are tables where Databricks manages the data's lifecycle. This means Databricks is responsible for storing, organizing, and managing the data associated with the table.
2. **Storage Location**: When you create a managed table, Databricks automatically stores the data in a designated location within the managed storage (typically in a managed database or catalog). You don't have to specify the storage path.
3. **Data Lifecycle**: Databricks takes care of the data lifecycle, including cleanup, when you drop a managed table. The data associated with the table is also deleted.
4. **CREATE Statement**: When creating a managed table, you don't specify the **LOCATION** option. The storage path is managed by Databricks.
5. **Use Cases**: Managed tables are useful when you want Databricks to handle data storage and management.

### External Tables

1. **Definition**: External tables (also known as unmanaged tables) are tables where the data is stored outside the control of Databricks, in a specified storage location.
2. **Storage Location**: When you create an external table, you must specify the **LOCATION** option to define the storage location of the data (e.g., in DBFS, S3, Azure Blob Storage, etc.).
3. **Data Lifecycle**: The lifecycle of the data is not managed by Databricks. When you drop an external table, only the metadata is removed, but the underlying data remains in the specified location.
4. **CREATE Statement**: When creating an external table, you specify the **LOCATION** option to define where the data is stored.
5. **Use Cases**: External tables are useful when you want to manage data independently of Databricks or if you want to share data across multiple systems or environments.

### Conclusion

The choice between managed and external tables depends on your use case and data management preferences:

* **Use Managed Tables** when you want Databricks to handle data storage and lifecycle management.
* **Use External Tables** when you want control over where your data is stored and how it is managed, or if you want to share data across multiple systems.

Let me know if you have any more questions or if there's anything else I can help you with.