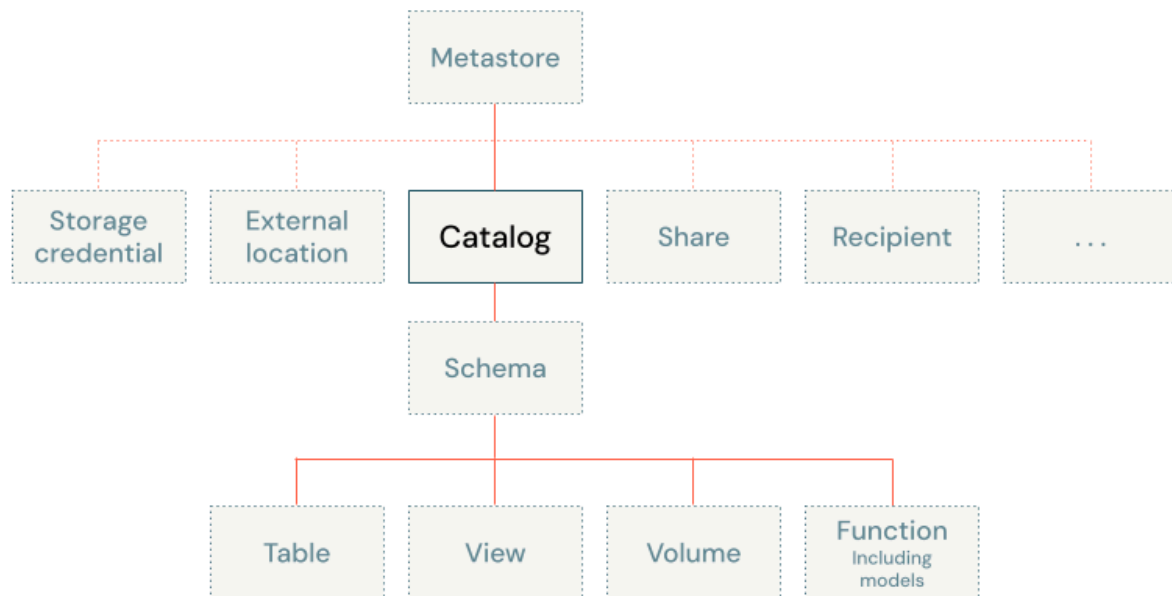


Catalog-Schema-View-Vol

Create Catalog

[Create Catalog](#)

Catalogs? (เพิ่มเติม [here](#))



เป็นหน่วยหลักในการจัดระเบียบข้อมูลของ Unity Catalog ใน Databricks เป็นระดับที่สูงสุดในโครงสร้างชื่อข้อมูลแบบ 3 ระดับ (catalog.schema.table-etc)

type

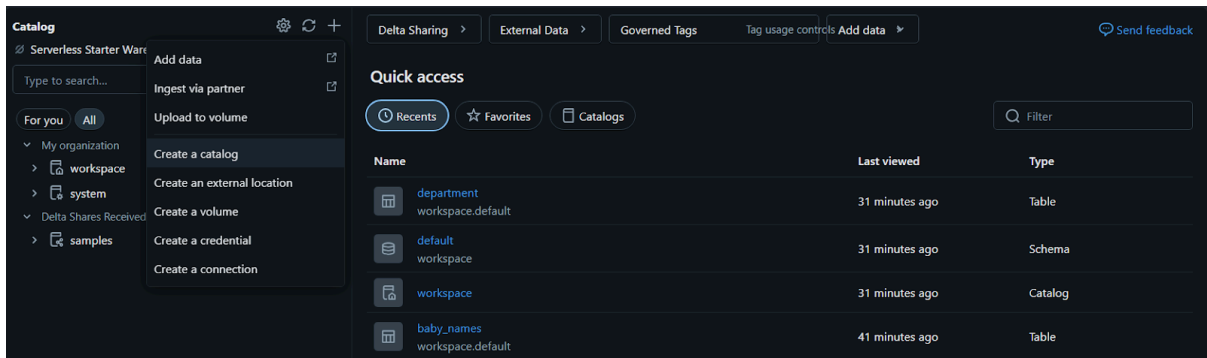
- shared catalog
- standard catalog : ปกติทั่วไปสำหรับการเก็บข้อมูล ที่เราใช้กันคือตัวนี้ รายละเอียด catalog แบบอื่นใน docs
- foreign catalog

มีวิธีการสร้าง catalog (ที่แนะนำ 2 แบบ)

- ใช้ UI ของ Databricks
- ใช้ SQL

ใช้ UI ของ Databricks (Catalog Explorer)

- ในหน้า Catalog คลิก '+' และเลือก 'Create a catalog'



- ตั้งชื่อและประเภทให้เรียบร้อย ในที่นี้เราเลือกแบบ Standard

Catalog name*

example

Type*

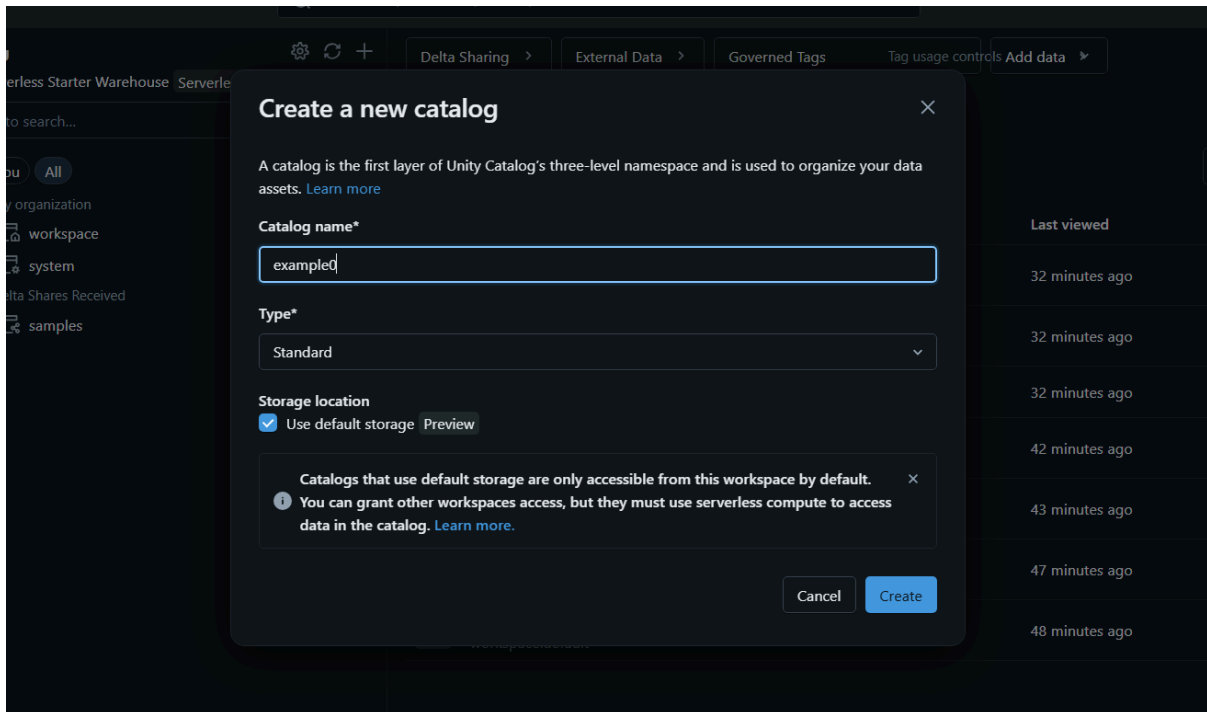
Standard

✓ **Standard**
Contains tables, views, functions and other objects. Data stored in cloud storage

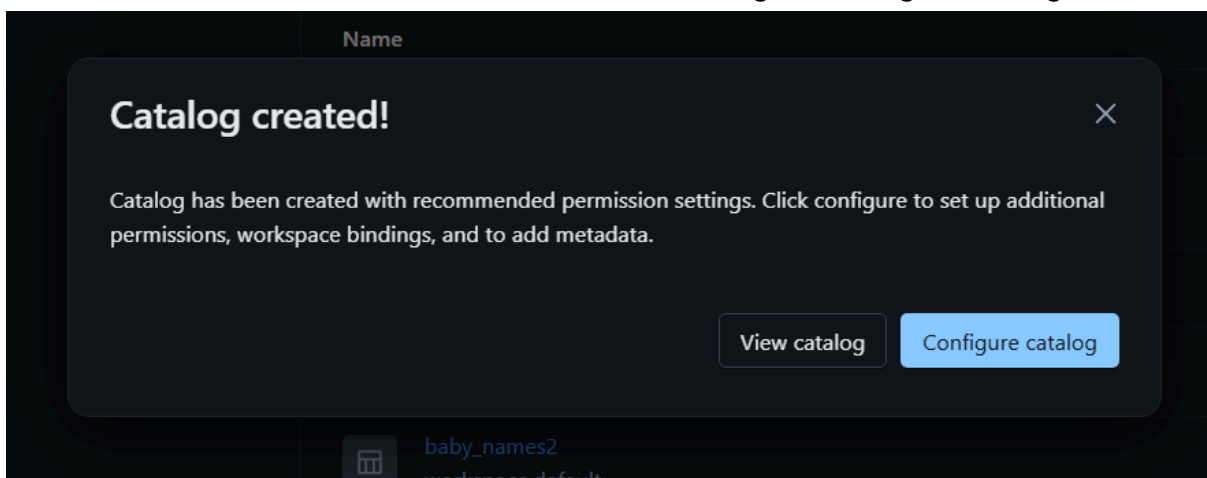
Foreign
Mount and query foreign data such as MySQL, Postgres, SQL Server and more

Shared
Mount and query data shared with your organization via Delta Sharing

Cancel Create



- หลังจากสร้างเสร็จแล้ว จะมีให้เลือกระหว่าง View Catalog หรือ Configure Catalog



- เลือก Configure Catalog จะสามารถจัดการ 2 ส่วนนี้เบื้องต้นได้ คือ Access (Permissions) และ Metadata ในภาพถัดไปจะเป็นการจัดการ Access กรณีที่เลือก workspace ทั้งหมดสามารถเข้าถึงได้

1 Access

Step 1

Access
Specify the users, groups, and service principals that have privileges on this catalog. Additionally, set up workspace-catalog bindings to isolate user data access.

Workspaces
Limit the workspaces in which users can access this catalog. [Learn more](#)

☒ All workspaces have access

Owner
Assign a user or group who will be able to manage access to all objects in the catalog

65070213@kmitl.ac.th

Privileges
Choose which users or groups can access this catalog. All account users are granted **BROWSE** by default. [Learn more](#)

[Grant](#) [Revoke](#) Type to filter by principal

Principal	Privilege	Object
All account users	BROWSE	example00

- ภาพด้านล่างคือกรณีที่เร ไม่ได้เลือก All workspaces have access เราจะสามารถ Assign ไปยัง workspace ต่างๆได้ แต่จะมี workspace หลักของเราตั้งต้นมาให้ และไม่สามารถลบได้

1 Access

Step 1

Access
Specify the users, groups, and service principals that have privileges on this catalog. Additionally, set up workspace-catalog bindings to isolate user data access.

Workspaces
Limit the workspaces in which users can access this catalog. [Learn more](#)

☐ All workspaces have access

[Assign to workspaces](#) [Manage Access Level](#) [Remove](#)

Workspace name	Workspace id	Access Level
workspace	4210069894755587	Read & Write

- ส่วนที่ 2 คือการใส่ Metadata ใน catalog เพื่อช่วยในการค้นหาให้หาเจอได้ง่าย

> Tags : เก็บในรูปแบบ Key-Value สามารถเพิ่มเติม Tags ได้ในหน้านี้

Catalog Explorer >

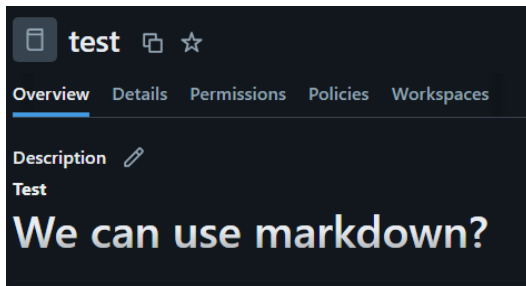
Governed Tags

[Governed Tags](#) [Account Permissions](#)

20 governed tags [Create governed tag](#)

Type	Tag key	Description	Allowed values
	class.credit_card	A credit card number containing 12-19 digits.	None
	class.de_id_card	A German National ID Card number (Personalausweis) containing ...	None
	class.de_passport	A German passport number containing 9 alphanumeric characters.	None
	class.de_ssnr	A German Social Security Number (Sozialversicherungsnummer) c...	None
	class.de_tax_id	A German Tax Identification Number (Steueridentifikationsnumme...	None

> Comment : รายละเอียด คำอธิบายเบื้องต้นของ catalog (ที่แจ้งคือ รองรับการใส่ markdown ด้วย! (ลองมาแล้ว))



ใส่ข้อมูลให้เรียบร้อย หรือจะไม่ได้ใส่ก็ได้ แล้ว Save

Configure catalog

1 Access

2 Metadata

Step 2

Metadata

Improve discoverability and governance by adding a markdown comment and tags. [Learn more](#)

Tags

Governed tags follow rules enforced by tag policies. [Manage governed tags](#)

Key	Value
Select a key	Select a value

Comment

Cancel Back Save

- สุดท้ายหลังจากสร้างเสร็จ จะได้หน้าตาประมาณนี้

Catalog

Serverless Starter Warehouse Serverless 2XS

Type to search...

For you All

My organization

workspace

system

example0

example00

Delta Shares Received

samples

Catalog Explorer

example00

Overview Details Permissions Policies Workspaces

Description

AI generate Add

Filter schemas 2 schemas

Name	Owner	Created at
default	65070213@kmitl.ac.th	Dec 06, 2025, 09:53 PM
information_schema	System user	Dec 06, 2025, 09:53 PM

About this catalog

Owner 65070213@kmitl.ac.th

Tags

Add tags

Policies

New policy

ใช้ SQL

Query :

```
CREATE CATALOG [ IF NOT EXISTS ] <catalog-name>  
  [ MANAGED LOCATION '<location-path>' ]  
  [ COMMENT <comment> ];
```

โดยในที่นี้ ยกตัวอย่างเป็น catalog ชื่อ example

```
CREATE CATALOG IF NOT EXISTS example;
```

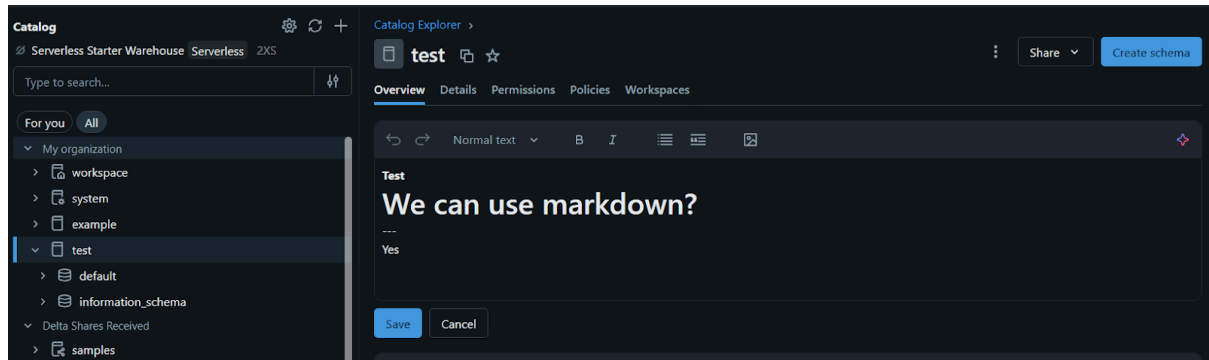
แก้ไข permissions เพิ่มเติมในหน้า catalog

Create Schema

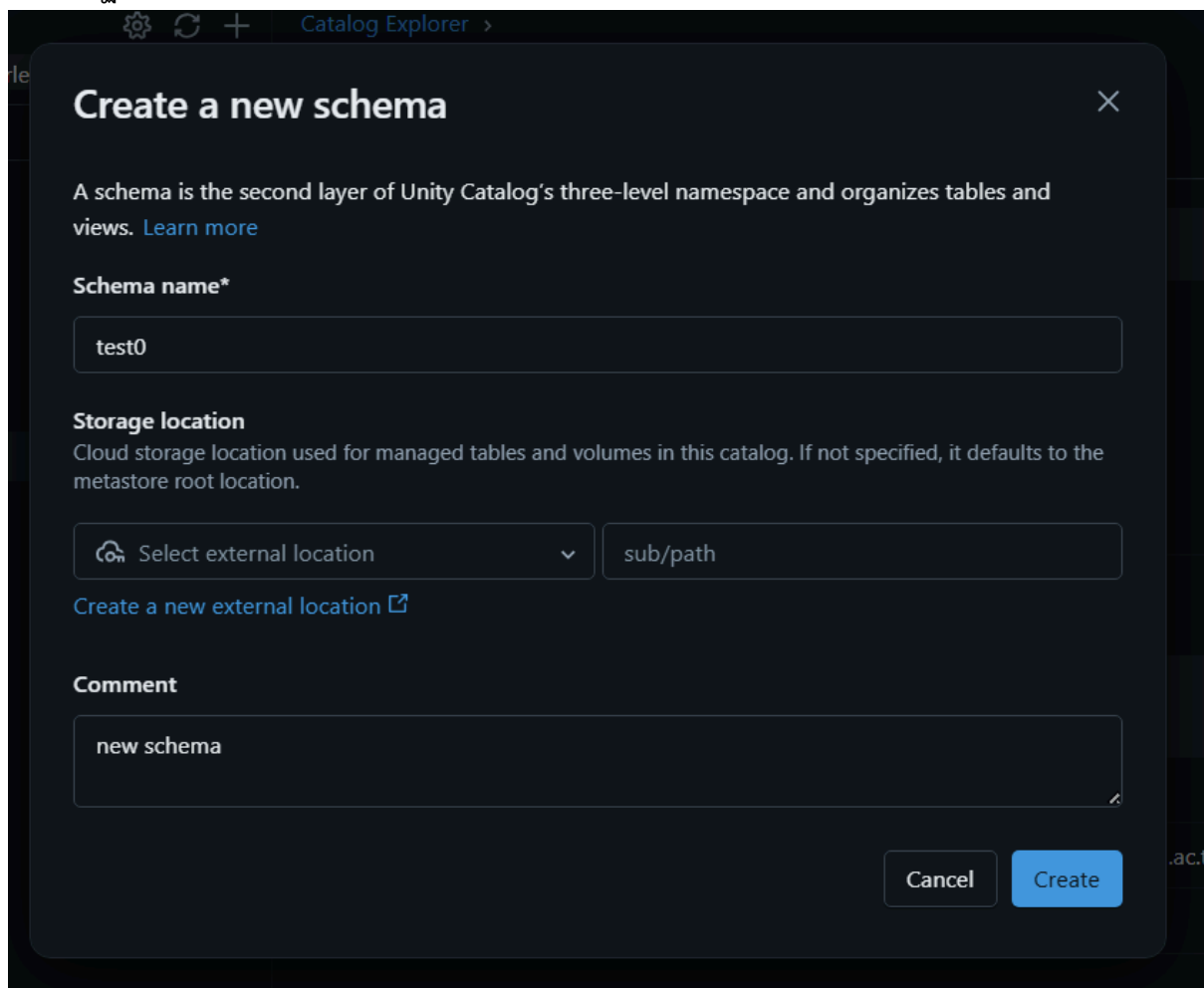
[Create Schema](#)

Schema = Database

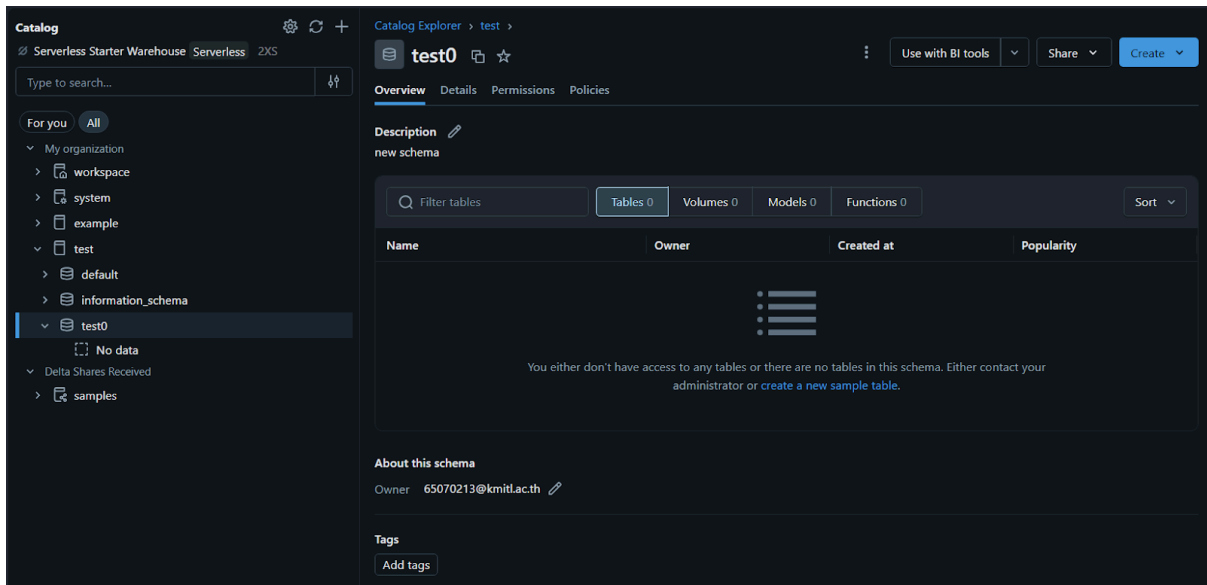
เลือก catalog ที่ต้องการสร้าง schema > คลิกปุ่ม 'Create Schema'



จะปรากฏหน้าต่างให้ใส่รายละเอียดเบื้องต้น แล้วคลิก 'Create'



เราจะเห็น Schema ใหม่ปรากฏ



นอกจากสร้างใน Catalog Explorer แล้ว ยังใช้ SQL ได้เช่นกัน

Query :

```
CREATE { DATABASE | SCHEMA } [ IF NOT EXISTS ] <catalog-name>.<schema-name>
  [ MANAGED LOCATION '<location-path>' | LOCATION '<location-path>' ]
  [ COMMENT <comment> ]
  [ WITH DBPROPERTIES ( <property-key = property_value [ , ... ]> ) ];
```

ที่อยู่ใน [] = optional

โดยชื่อ schema ที่เราอยากตั้งให้แทน **<schema-name>** ได้เลย

ที่สามารถตั้งเพิ่มเติมได้ เช่น

- ตั้ง location ที่ระบุว่ามีไฟล์จริงอยู่ที่ไหน
- comment คือ คำอธิบาย schema
- property-key เป็นการตั้งค่าเฉพาะของ spark (ถ้ามีอะไรพิเศษจริงๆ ค่อยยุ่งกับตรงนี้)

Create View(s)

[Create Views](#)

ก่อนที่จะเราจะสร้าง View

- ต้องมีสิทธิ์ใช้ catalog รวมไปถึงการใช้ schema, และการสร้างตารางด้วย
- ต้องมีสิทธิ์อ่านข้อมูลในตาราง
- มีอีก 2 ข้อ (เกี่ยวกับ workspace-local Hive metastore และ Delta Sharing) แต่ไม่น่าลึกขนาดนั้น เบื้องต้นน่าจะมีสิทธิ์หมดแล้ว

ในที่นี้เราจะใช้ SQL

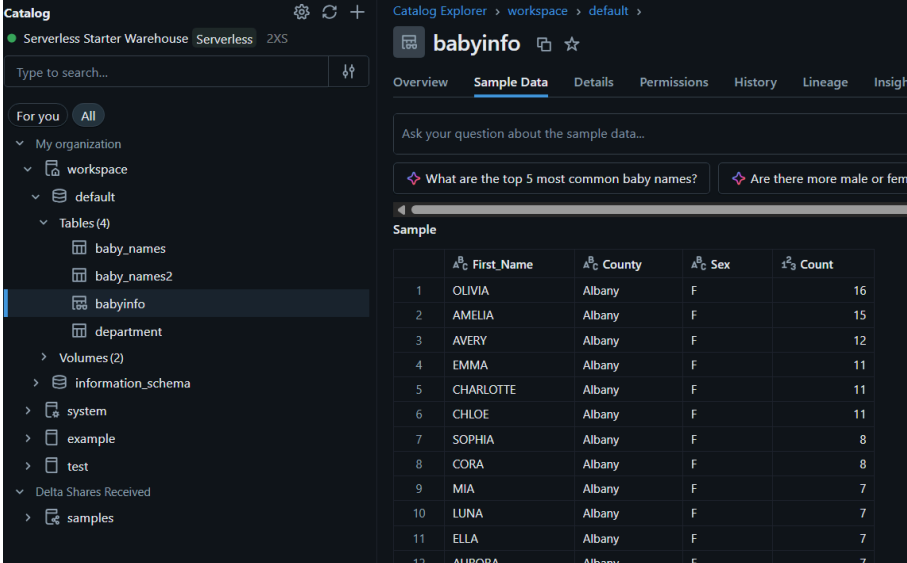
Query :

```
CREATE VIEW <catalog-name>.<schema-name>.<view-name> AS  
SELECT <query>;
```

ตัวอย่างที่ใช้กับงานที่เราลองทำไปแล้ว

```
CREATE VIEW workspace.default.babyinfo AS  
SELECT  
  First_Name,  
  County,  
  Sex,  
  Count  
FROM workspace.default.baby_names;
```

จะเห็นว่า มี view babyinfo งอกมาแล้ว



The screenshot shows the Databricks Catalog Explorer interface. On the left, the 'Catalog' sidebar shows the hierarchy: 'workspace' > 'default' > 'Tables (4)'. The 'babyinfo' view is selected and highlighted. The main panel displays the 'babyinfo' view details, including a 'Sample Data' tab. Below the 'Sample Data' tab, a table shows the first 12 rows of data:

	First_Name	County	Sex	Count
1	OLIVIA	Albany	F	16
2	AMELIA	Albany	F	15
3	AVERY	Albany	F	12
4	EMMA	Albany	F	11
5	CHARLOTTE	Albany	F	11
6	CHLOE	Albany	F	11
7	SOPHIA	Albany	F	8
8	CORA	Albany	F	8
9	MIA	Albany	F	7
10	LUNA	Albany	F	7
11	ELLA	Albany	F	7
12	AURORA	Albany	F	7

ถ้าต้องการลบ view ใช้ query :

```
DROP VIEW IF EXISTS catalog_name.schema_name.view_name;
```

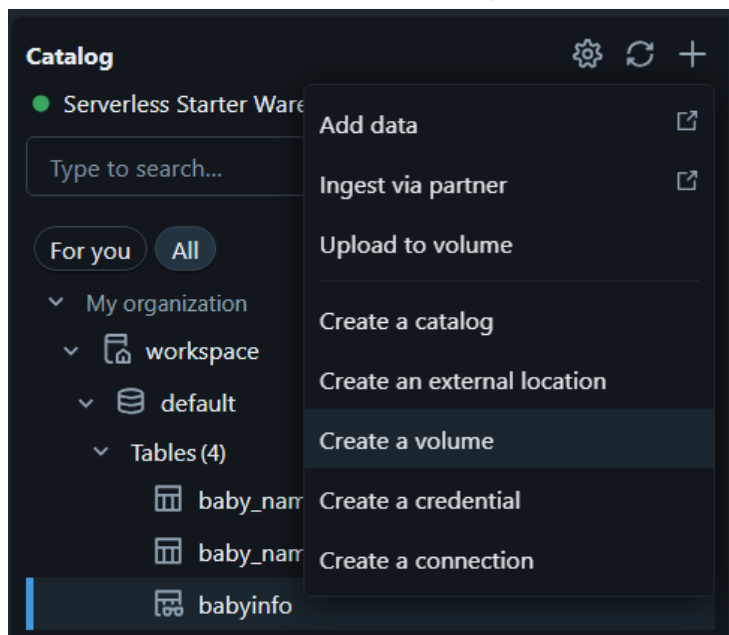

Volumes

[What are Unity Catalog volumes?](#)

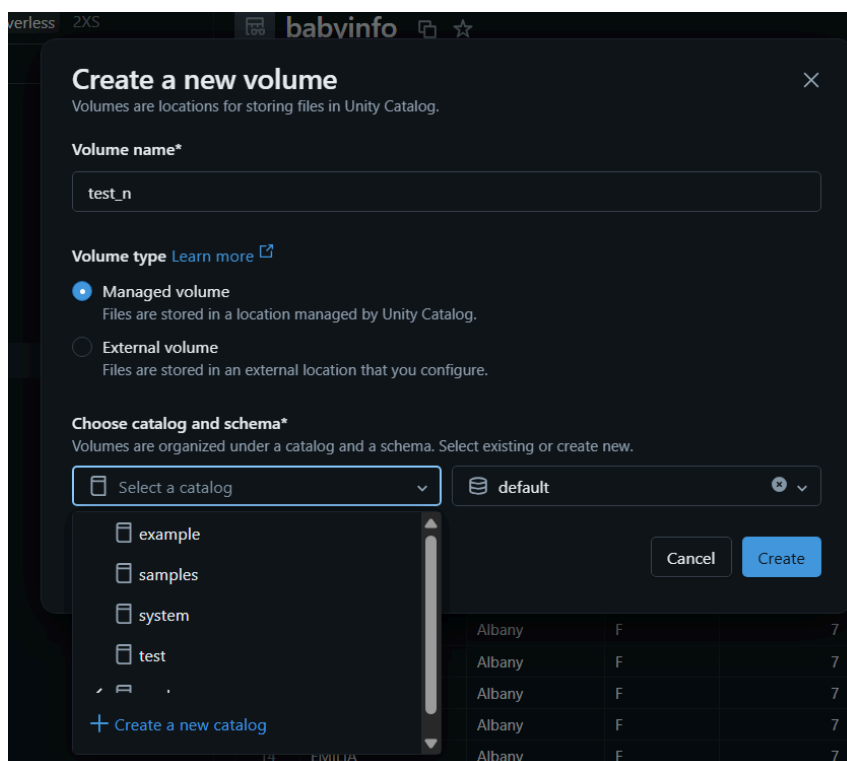
volume จะเก็บไฟล์ได้หลากหลายประเภท เช่น รูปภาพ เสียง หรือนานาจิตตังที่ไม่ใช่ tabular-data

จริงๆถ้าเรื่องการสร้าง volumes ได้แอบแสดงวิธีทำไปแล้วในเอกสารอีกชุดละ (แต่ในกรณีนั้นคือการใช้ SQL)

ในที่นี้จะสร้าง Volume โดยใช้ Catalog Explorer



หลังจากเลือก create volume แล้ว จะเดิ้งหน้าต่า่งนี้ขึ้นมา และมี option ให้เลือก



ใส่ชื่อ volume ให้เรียบร้อย

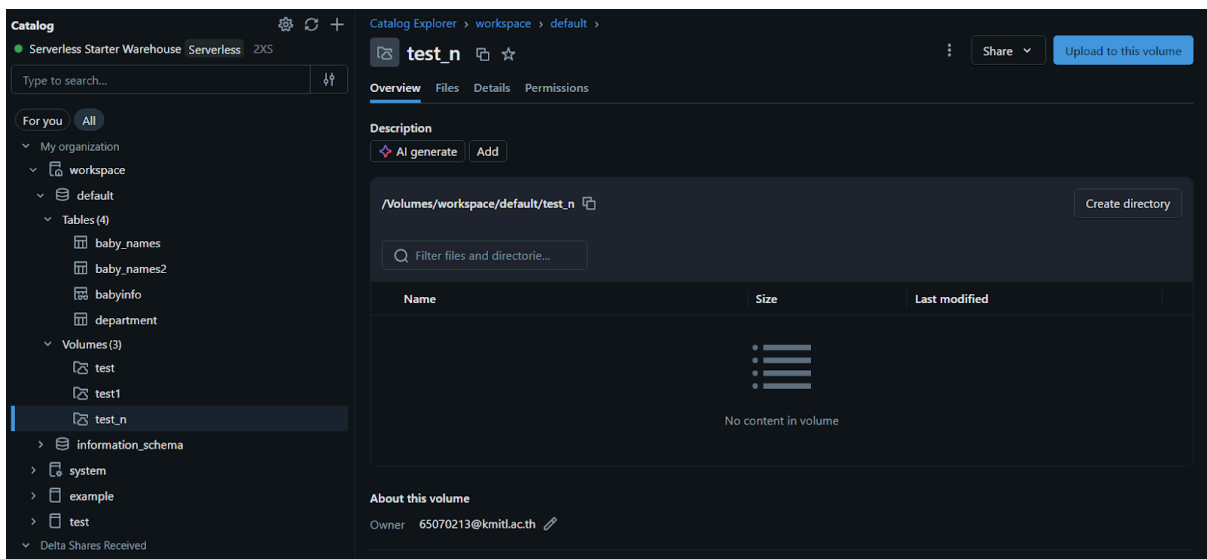
ในส่วนของ Volume Type

- Managed Volume
จะเป็นที่เก็บข้อมูลแบบพื้นฐานใน Databricks เราจะใช้ตัวนี้

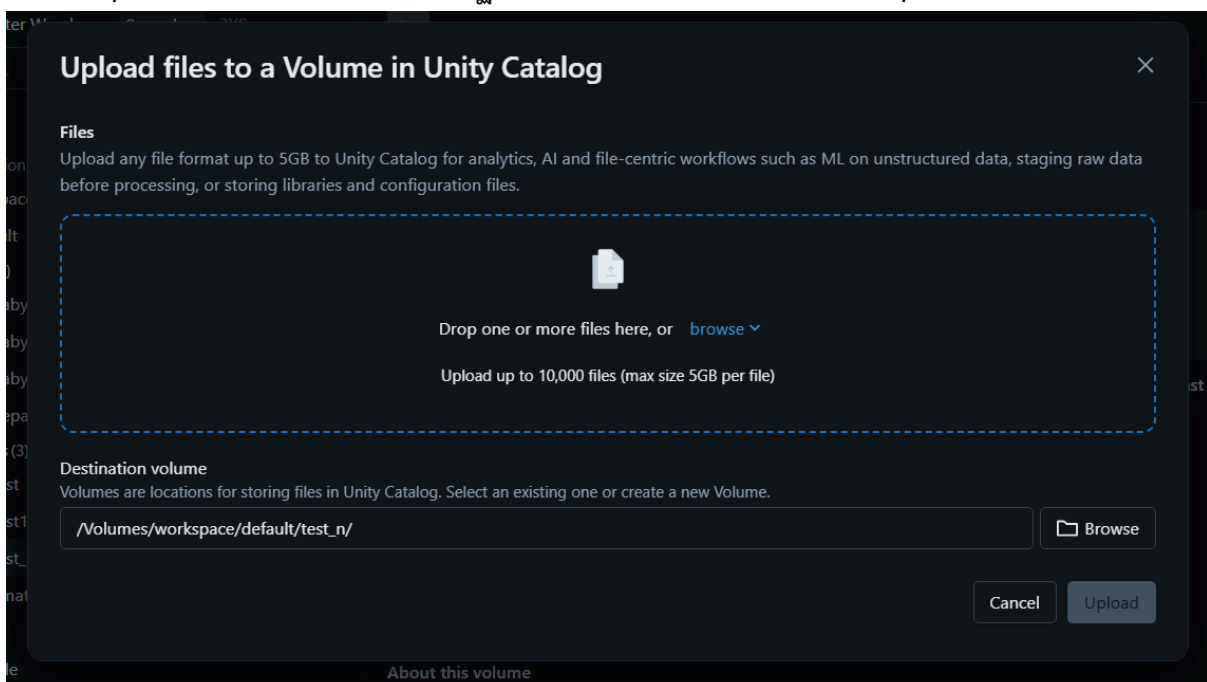
- External Volume
เชื่อมต่อกับข้างนอก พวก cloud

เลือก catalog, schema แล้วคลิก create

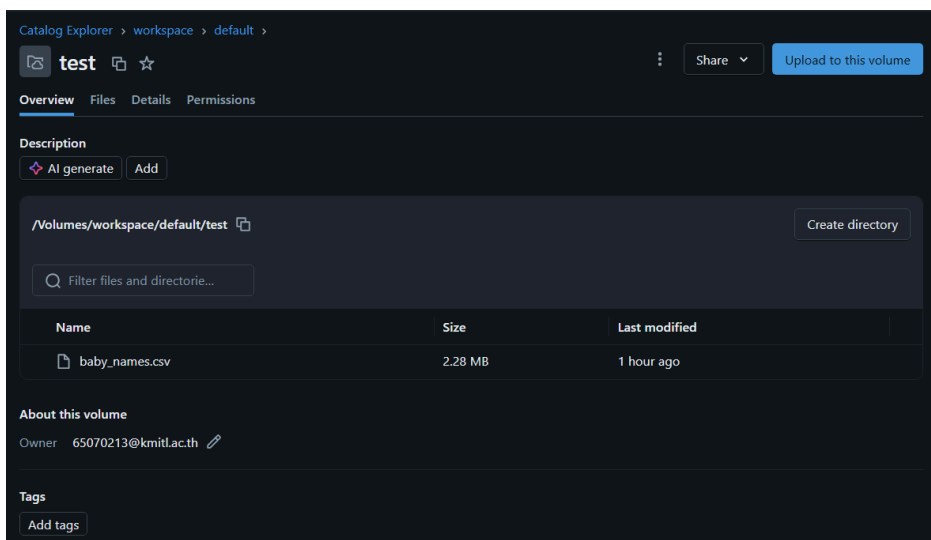
หลังจากสร้างแล้ว จะได้หน้าตา Volume ประมาณนี้ 🙋



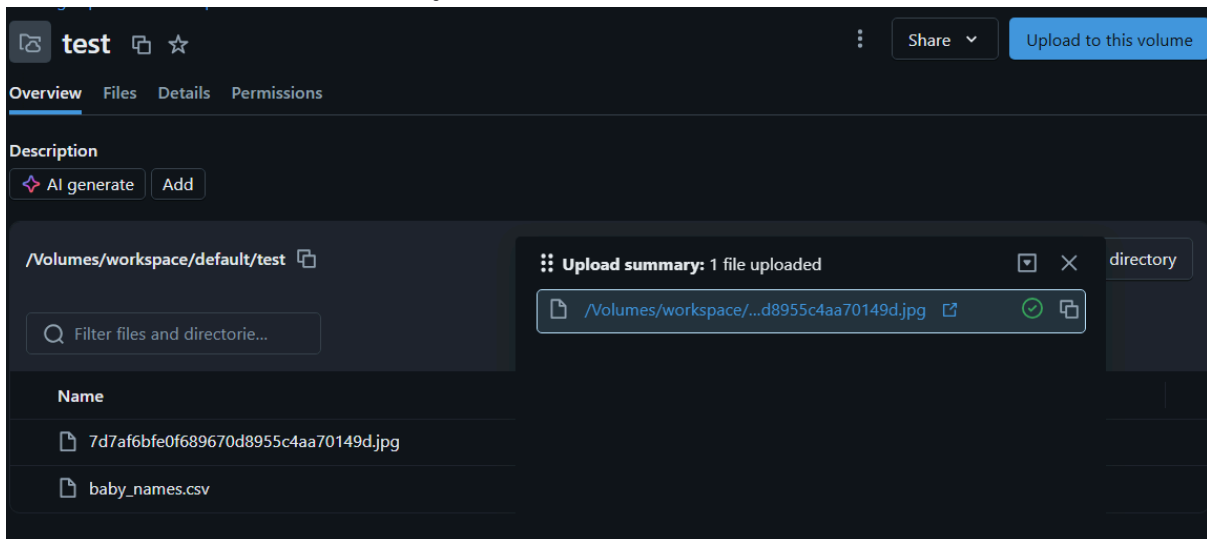
ถ้าคลิก Upload to this volume จะปรากฏหน้าต่างแบบข้างล่างขึ้นมา ให้เรา upload file เข้าไปได้



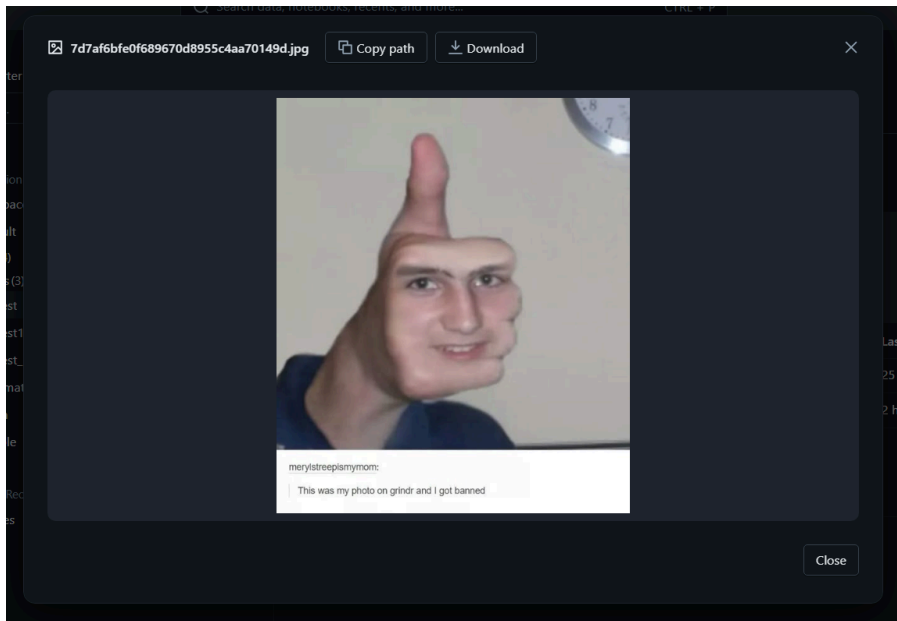
ตัวอย่าง volume ที่มีไฟล์แล้ว เช่น



สมมติลองอัปไฟล์เข้าไป ตัวอย่างเป็นรูปภาพ



หลังจากอัปเข้าไปแล้วสามารถลองคลิกดูไฟล์ได้



ซึ่ง path ของ volume สามารถ copy จากหน้า volume ได้เลย
โดยปกติ path ของ volume จะอยู่ในรูปแบบนี้

`/Volumes/<catalog>/<schema>/<volume>/<path>/<file-name>`

ถ้าใช้กับ Apache spark รูปแบบนี้ก็ใช้งานได้

`dbfs:/Volumes/<catalog>/<schema>/<volume>/<path>/<file-name>`