Unity Catalog

Session 1

===========

Unity catalog is a unified governance solution for all the data assets in your lakehouse.

Data governance means setting data policies..

1. tables (Database objects)
2. users (100 users)
3. access policies

Basically we have objects & users & we grant permissions to users on various objects.

All of this is centralized and not just limited to individual workspaces.

Key features of unity catalog

================================

=> unity catalog allows you to implement a centralized metadata layer. so our databases, tables and views can be shared across multiple workspaces.

=> unity catalog security model is based on Standard ANSI SQL and alllows admins to

grant permissions in their existing data lake using familiar syntax.

=> Built in auditing - captures user level audit logs..

Unity catalog capures an audit log of actions performed against the metastore. This

enables admins to access fine-grained details about who accessed a given dataset

and what actions they performed.

It offers more features, such as

Delta Sharing - you can easily share existing data in Delta lake and parquet formats to

any computing platform.

Data Lineage -

t1

t3 - total_sales

t2

Data Discovery - a open seach capability and we can search for anything there.

=============

Metastore - tables, views

User Management - Users

Access controls

Multiple databricks workspaces are linked as it is centralized

without unity catalog, your metastore, user management and access control are

defined at the workspace level. Data objects such as databases and tables which are

created in one workspace will not be visible in other workspace.

Session 2

===========

How to configure the Metastore for the Unity catalog

Step 1: Configure Storage for your Metastore - ADLS gen2

table - Data + Metadata

1. Created a storage account for ADLS Gen2.

This storage will keep the metadata related to unity catalog metastore & the objects

inside it. For the managed tables this will keep the data also.

Storage account name - ttunitycatalogsa

2. Create a storage container that will hold your unity catalog metastore's metadata

and managed data.

container - unityroot

3. In Azure, create a databricks access connector that holds a managed identity and

give it access to the storage container.

=> create databricks access connector

ttdatabricksaccesscon

go to your storage account and click on access control (IAM)

click on add -> role assignment -> storage blob data contributor -> to the above

created connector

Step 2: you will go to your Databricks UI & there we will create the metastore. we can

then link multiple workspaces to this metastore.

created a premium databricks workspace - ttdatabricks101ws

ttdatabricks102ws - standard

log in to workspace as an account admin

clicked on data icon

clicked on create metastore ->

-> region should be the same where you have the storage account and the

workspaces

-> storage account path: unityroot@ttunitycatalogsa.dfs.core.windows.net/

-> access connector id:

/subscriptions/6a5c4c05-4619-4d46-a93e-f2354a39e238/resourceGroups/tt-a
zure-datab

ricks-rg/providers/Microsoft.Databricks/accessConnectors/ttdatabricksaccessc
on

we attached the premium workspace with this metastore.

Session 3

==========

we saw how to create a metastore.

The unity catalog object model

=> Metastore: The top level container for metadata. Each metastore it
exposes a 3

level namespace. This is how the data is organized.

This is used to store all the metadata and data for managed tables.

catalog.schema.table

Catalog: First layer in the object hierarchy.(Dev, Stage, Prod)

can be based on logical grouping like environment, team etc..

users can see all catalogs on which they have been assigned the usage data

permissions..

Schema: Also known as a database, schema are second layer in the object
hierarchy.

the schema organizes table and views. To access (or list) a table or view
inside the

schema, then user must have usage data permission on the schema and also
its

parent catalog. Also I should have select permission on the table catalog - prod

database name: retaildb

table - sales

Tables - prod.retaildb.sales

it is the lowest level in the object hierarchy.

tables can be external or managed

External - the data is stored in external locations in your cloud storage of your choice.

If we drop an external table only the metadata is deleted but not the acutal data. The

data is stored outside root storage location. if you require direct access to data which

other tools are also using, then you should create external tables.

Managed - The data is stored in the root storage. These are the default way to create

tables in unity catalog. Use Delta format to store the underlying data. when a

managed table is dropped, the underlying data is deleted from your cloud within 30

days.

Also we can create read only views from tables.

To create a table, a user must have CREATE and USAGE Permissions on the Schema.

To query a table, users must have the Select permission on the table, and they must

have the usage permission on its parent catalog and schema.

Views

======

A read only object created from one or more tables.

Session 4

==========

Metastore

Catalog -> Schema -> Tables

In all cases the metadata is stored in Metastore root

but only in case of managed tables the data is stored in metastore root.

How to add user

click on user tab -> add users

Azure AD would have user details

you would wish to sync the user details from Azure AD to Databricks Unity Catalog.

This sync between Azure AD -> Unity catalog is possible using SCIM

It is a open standard that allows you to automate user provisioning between multiple

systems.

scim token - dsapide9c10d2ac52fd812fdead54bb4390c9

scim url -

https://accounts.azuredatabricks.net/api/2.0/accounts/cdf10f57-8ae5-4ea7-8fc2-86d1a3

ef237e/scim/v2

Azure AD -> Enterprise applications -> SCIM connector

Provisioning -> automatic ->

Session 5

==========

how to sync the users from Azure active directory to the unity catalog

SCIM connector.

Admin roles for unity catalog

===============================

1. Account admin

2. Metastore admin

3. Workspace admin

workspace users

The one who creates the unity catalog enabled databricks becomes the account

admin

sachinm@trendytechsumitoutlook.onmicrosoft.com

Gowu8050

manasan@trendytechsumitoutlook.onmicrosoft.com

Sotu0615

kapild@trendytechsumitoutlook.onmicrosoft.com

Polu6644

satishb@trendytechsumitoutlook.onmicrosoft.com

Woja0753

ramd@trendytechsumitoutlook.onmicrosoft.com

Ruqu2661

nehas@trendytechsumitoutlook.onmicrosoft.com

Fata5965

I want to make sachin the account admin

https://accounts.azuredatabricks.net/

Session 6

==========

Sumit created the databricks account which has unity catalog enabled so Sumit

became the account admin during his first login.

we added 6 users

I made Sachin the account admin.

1. Account admin (group cannot be created)

has been assigned to Sachin (individual person)

2. Metastore admin

trendytech-metastore-admins - Sumit, Sachin, Manasa

3. Workspace admin

trendytech-workspace-admins - kapil

Let us create a group for developers - workspace users

trendytech-developer-grp

kapil, neha, ram, satish

account admin - Sumit, Sachin

Metastore admin - Sumit, Sachin, Manasa

Workspace admin - Kapil

Developers - Kapil, Neha, Ram, Satish

change the admin of metastore to the group trendytech-metastore-admins

As a account admin

====================

1. Created the databricks workspaces

2. created the metastore

3. Link databricks workspace to unity catalog

4. delegate the account admin role to someone else

5. add users

6. create groups

7. create metastore admins

8. create workspace admins

Session 7

==========

Account admin - Sumit, Sachin

Metastore admin - Manasa, Sumit, Sachin (trendytech-metastore-admins)

trendytech unity catalog metastore - trendytech-metastore-admins

workspace admin - Kapil

ttdatabricks101ws ->

admin - trendytech-workspace-admins

user - trendytech-developer-grp

workspace developers - Kapil, Neha, Ram, Satish

I logged in as Manasa (who is the metastore admin)

Metastore -> catalog -> Schema -> Tables & views

we created a new catalog with the name customer360-development

catalog customer360-development

schema retaildb

ideally the database (Schema) & the objects inside the database (tables & views) are

created by developers.

The one who creates the catalog becomes the owner.

trendytech-metastore-admins (manasa,sumit,sachin)

we have changed the owner to trendytech-metastore-admins

I have assigned all privileges (create database, table, views etc.. on the catalog

customer360-development)

as a metastore admin we -

1. created the catalog

2. gave permissions to objects in metastore (developer group)

Session 8

==========

https://adb-3427263434701955.15.azuredatabricks.net/

workspace admin - Kapil

workspace user - Neha

We logged in as Neha (a normal user - developer for the workspace)

she wont be able to see an option to create a cluser

when I log in as Kapil (workspace admin)

we see the option to create a cluster

If I want to manage the workspace I should click on admin console on the top right.

workspace admin can create a cluster but a developer by default cannot create it.

there are 3 modes

=================

workspace admins can give below 3 kind of permissions to the developers.

1. unrestricted cluster creation policy - a developer will be able to create a cluster of

their choice.

2. a workspace admin can create a cluster and share it with all the developers. A

shared cluster.

3. the workspace admin creates cluster policy to restrict the kind of cluster which a

developer can create.

Session 9

==========

1. unrestricted cluster creation -> we logged in as workspace admin (Kapil) and we

gave neha the permission of unrestricted cluster creation in the users tab.

2. Shared cluster that can be used

the workspace admin will create a shared cluster and then he will give attach to

permission for the developer group.

the workspace admin (Kapil created a shared cluster and gave attach to permssions

to the developers)

3. restricted cluster with limited options

Session 10

===========

{

"node_type_id" : {

"type" : "allowlist",

"values" : ["Standard_DS3_v2", "Standard_DS4_v2"],

"defaultValue" : "Standard_DS4_v2"

},

"spark_version" : {

"type" : "unlimited",

```
"defaultValue" : "11.3.x-scala2.12"
},
"runtime_engine" : {
"type" : "fixed",
"value" : "STANDARD",
"hidden" : true
},
"num_workers" : {
"type" : "fixed",
"value" : 0,
"hidden" : true
},
"data_security_mode" : {
"type" : "fixed",
"value" : "SINGLE_USER",
"hidden" : true
},
"driver_instance_pool_id" : {
"type" : "forbidden",
"hidden" : true
},
"cluster_type" : {
"type" : "fixed",
"value" : "all-purpose"
},
"instance_pool_id" : {
```

```
"type" : "forbidden",
"hidden" : true
},
"azure_attributes.availability" : {
"type" : "fixed",
"value" : "ON_DEMAND_AZURE",
"hidden" : true
},
"spark_conf.spark.databricks.cluster.profile" : {
"type" : "fixed",
"value" : "singleNode",
"hidden" : true
},
"autotermination_minutes" : {
"type" : "unlimited",
"defaultValue" : 60,
"isOptional" : true
}
}
```

Sumit - account admin

I made sachin as the account admin

we created 6 users

metastore admin group - manasa, sumit, sachin

developer group - kapil, ram, neha, satish

workspace admin group - kapil

workspace - users (developer group)

workspace - workspace admin (workspace admin)

Manasa (metastore admin)

we made metastore-admin-group as the owner of metastore

catalog inside the metastore

workspace User - could not see the option to create a cluster (Neha)

workspace admin - is the one who manages the workspace, can add and delete users,

can create cluster (Kapils)

1. unrestricted

2. shared cluster

3. restricted cluster creation based on policies