Microsoft

Microsoft Partner Project Ready

Implement with Impact

Modern Data Platform with Azure Databricks

<Speaker name or subtitle>

<Date>

Day 3 of 3

Technofocus

Course Plan and Learning Objectives

Day 1

Module 1 - Introduction to Azure Databricks
- Azure Databricks: A Data Intelligent Platform
- Why Azure Databricks
- Decision guide: Azure Databricks vs. Microsoft Fabric

Module 2 - Migration to Azure Databricks
- Microsoft Cloud Adoption Framework for Azure
- Migration strategies
- Data landing zones
- Migration scenarios

Interactive Simulated Lab Experience
- End-to-End Streaming Pipeline with Lakeflow Declarative Pipelines in Azure Databricks

Day 2

Module 3 - Integration with Azure
- Seamless integration with Microsoft Azure services
- Connect to Azure Data Lake Storage (ADLS) Gen2 and Blob Storage
- Leverage Azure Databricks for Azure Cosmos DB Operations
- Secret management with Azure Key Vault
- Connect Azure Databricks to Azure Event Hubs

Module 4 - Integration with Microsoft Fabric and Power BI
- Data Intelligence with Azure Databricks and Microsoft Fabric
- Connect Power BI to Azure Databricks
- Integration with Azure Data Factory
- Mirroring Azure Databricks Unity Catalog

Interactive Simulated Lab Experience
- Setup and use Unity Catalog for Data Management in Azure Databricks
- Real-Time Streaming with Azure Databricks and Azure Event Hubs

Day 3

Module 5 - Integration with Azure AI Foundry
- Azure Databricks connector in Azure AI Foundry
- Mosaic AI and machine learning on Azure Databricks
- Query Generative AI model serving endpoints
- Databricks Assistant, AI/BI Genie and AI Functions on Azure Databricks
- Chat with LLMs and prototype GenAI apps using AI Playground
- Build and optimize agents on your data with Agent Bricks

Module 6 - Security and Governance
- Integrate Azure Databricks with Microsoft Purview
- Integration of Azure Databricks Unity Catalog with Microsoft Purview

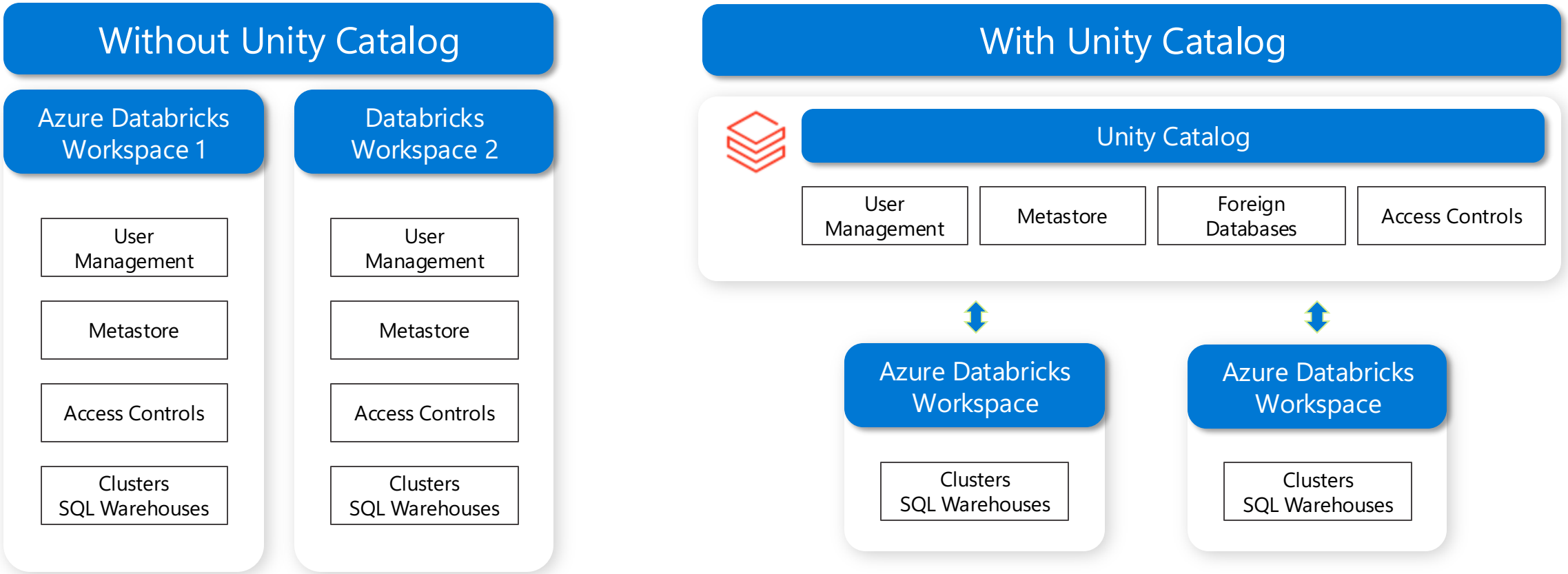
Module 7 - Well-architected for Azure Databricks
- Lakehouse implementation: Principles and best practices
- Azure Databricks well-architected framework

Interactive Simulated Lab Experience
- Responsible AI with Large Language Models using Azure Databricks and Azure OpenAI
- Connect to and manage Azure Databricks in Microsoft Purview

06
Security and Governance

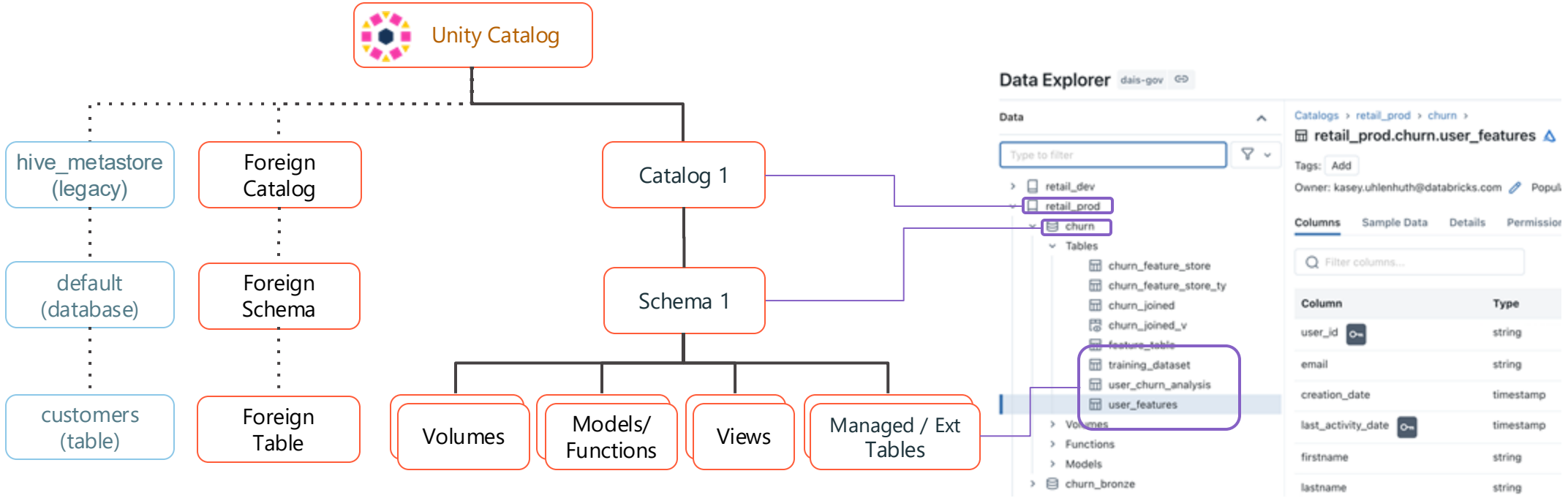
All your metadata, in one place

One metadata layer across file and database sources superpowers governance



Governed namespace across file and database sources

Access legacy metastore and foreign databases powered by Lakehouse Federation




```
SELECT * FROM main.paul.red_wine;  -- <catalog>.<database>.<table>

SELECT * FROM hive_metastore.default.customers;

SELECT * FROM snowflake_warehouse.some_schema.some_table;
```

Centralized Access Controls

Centrally grant and manage access permissions across workloads

Using ANSI SQL DCL

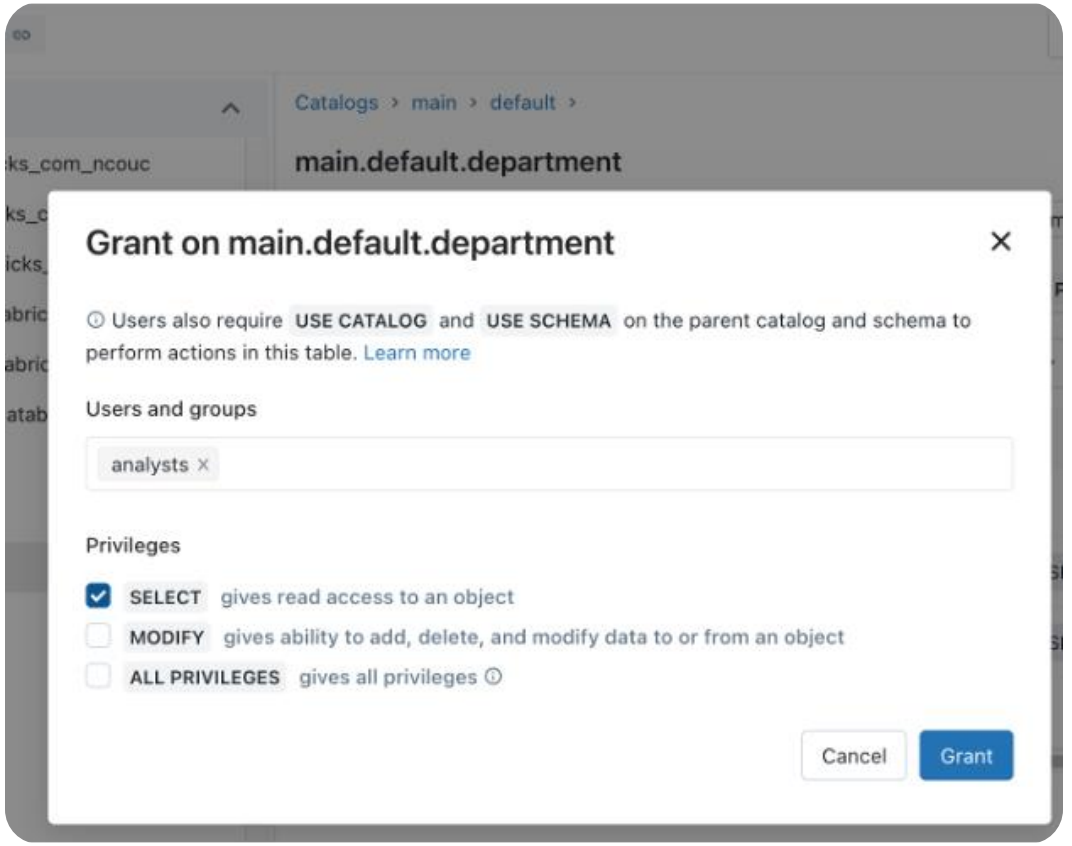
```
GRANT <privilege> ON <securable_type>
<securable_name> TO `<principal>`

GRANT SELECT ON iot.events TO engineers
```

Choose permission level

'Table'= collection of files in S3/ADLS

Sync groups from your identity provider

Using UI



Row Level Security and Column Level Masking

Provide differential fine-grained access to datasets

Only show specific rows

```
CREATE FUNCTION <name> ( <parameter_name >
<parameter_type> .. )
RETURN {filter clause whose output must be a boolean}

CREATE FUNCTION us_filter(region STRING)
RETURN IF(IS_MEMBER('admin'), true, region="US");

ALTER TABLE sales SET ROW FILTER us_filter ON region;
```

Test for group membership

Assign reusable filter to table

Specify filter predicates

Mask or redact sensitive columns

```
CREATE FUNCTION <name> (<parameter_name>, <parameter_type>, [,
<column>...])
RETURN {expression with the same type as the first parameter}

CREATE FUNCTION ssn_mask(ssn STRING)
RETURN IF(IS_MEMBER('admin'), ssn, "****");

ALTER TABLE users ALTER COLUMN table_ssn SET MASK ssn_mask;
```

Test for group membership

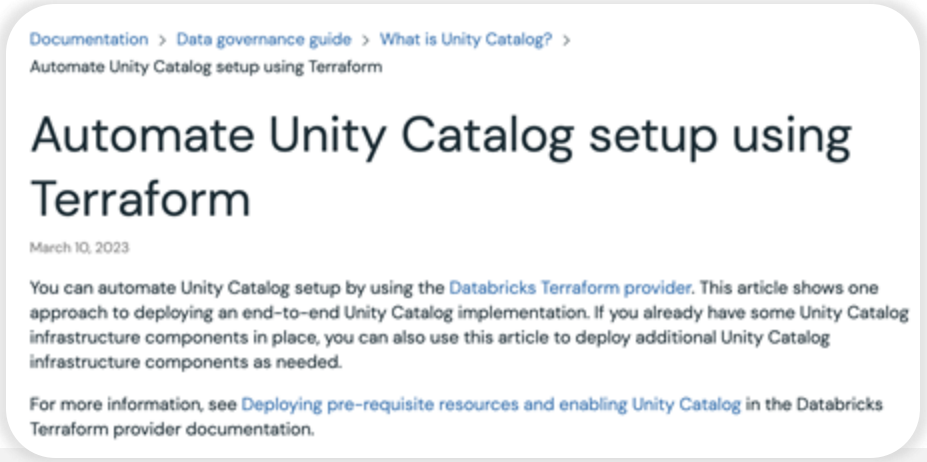
Assign reusable mask to column

Specify mask or function to mask

Work with Terraform & APIs

Use data-sec-ops, policies as code patterns to scale your efforts

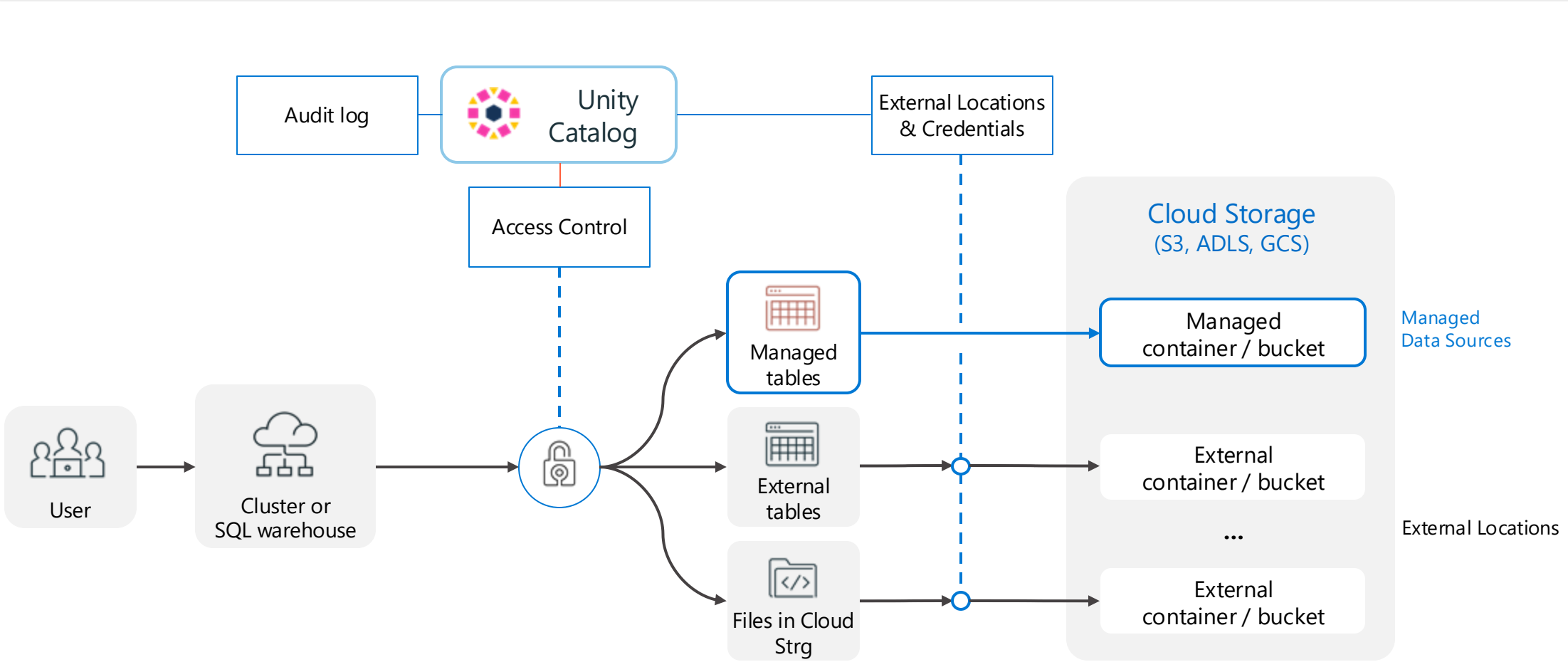
- Privileges for UC objects can be managed programmatically using our Terraform provider, especially for teams already using Terraform
- This will pair naturally with the management of the UC objects (Metastore, Catalog, Assignments etc.) themselves.
- (If not already using Terraform, maybe now is a good time!)




```
resource "databricks_grants" "sandbox" {
  provider = databricks.workspace
  catalog = databricks_catalog.sandbox.name
  grant {
    principal = "Data Scientists"
    privileges = ["USAGE", "CREATE"]
  }
  grant {
    principal = "Data Engineers"
    privileges = ["USAGE"]
  }
}
```

Managed Data Sources & External Locations

Simplify data access management across clouds



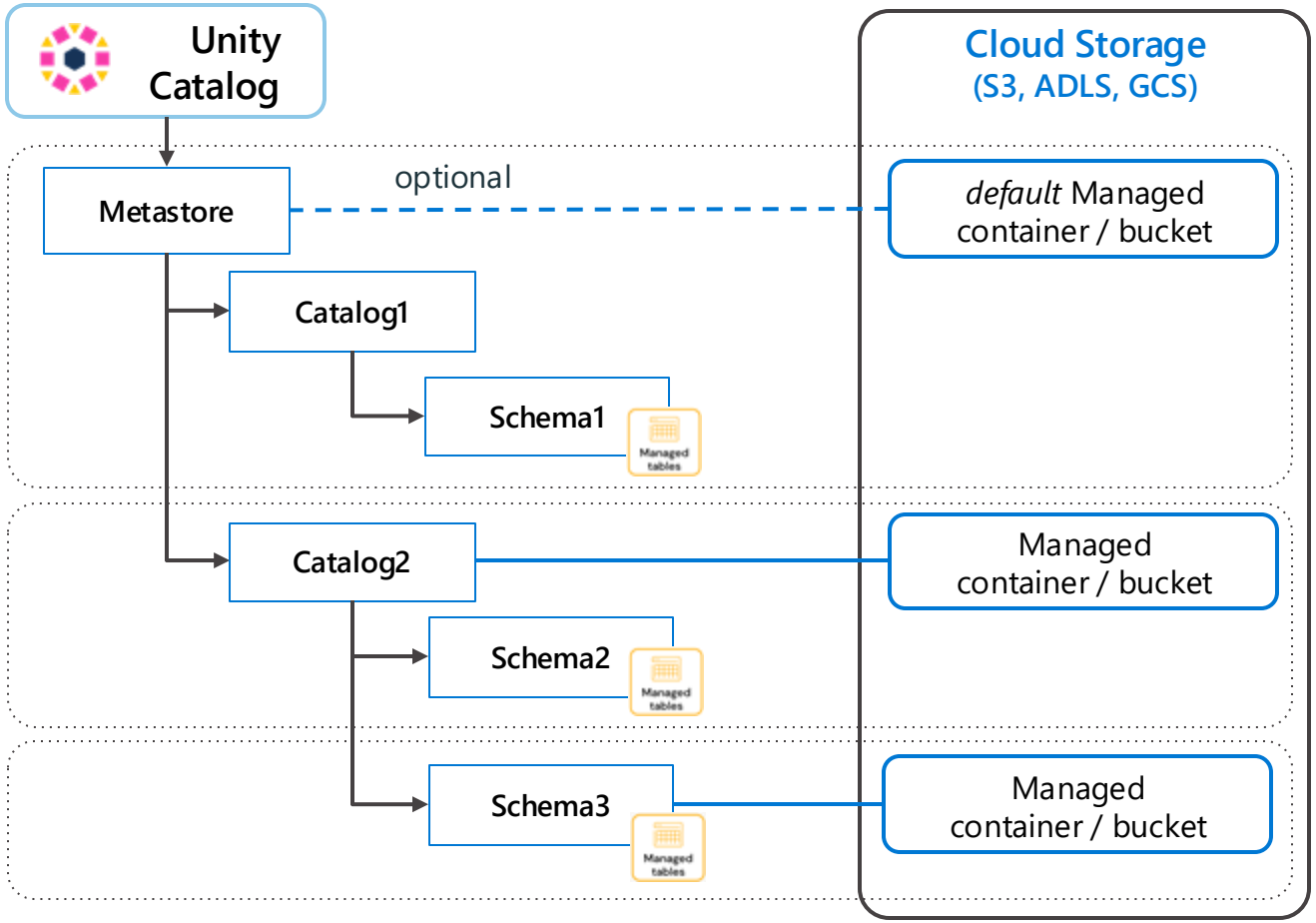
Default access to storage by catalog or schema

Use managed data sources for data isolation or cost allocation

1) Default storage associated with the **metastore**

2) Define at the **catalog**

3) Define at the **schema**



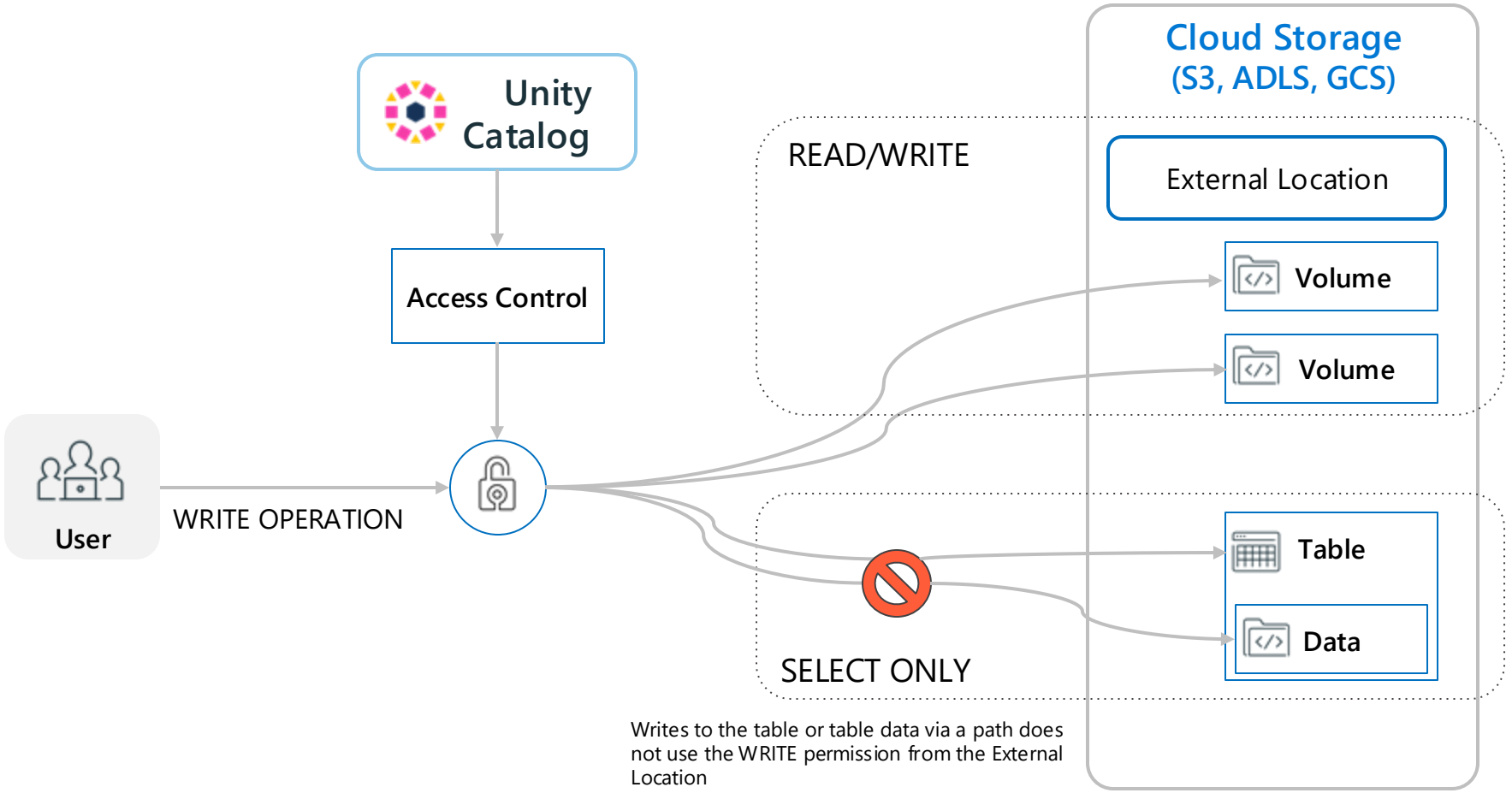
strong recommendation: do *not* store catalogs/schemas in the metastore-level default location

.
.
.

rather, specify a separate managed storage location for each catalog (which becomes the default location for schemas/tables w/in that catalog)

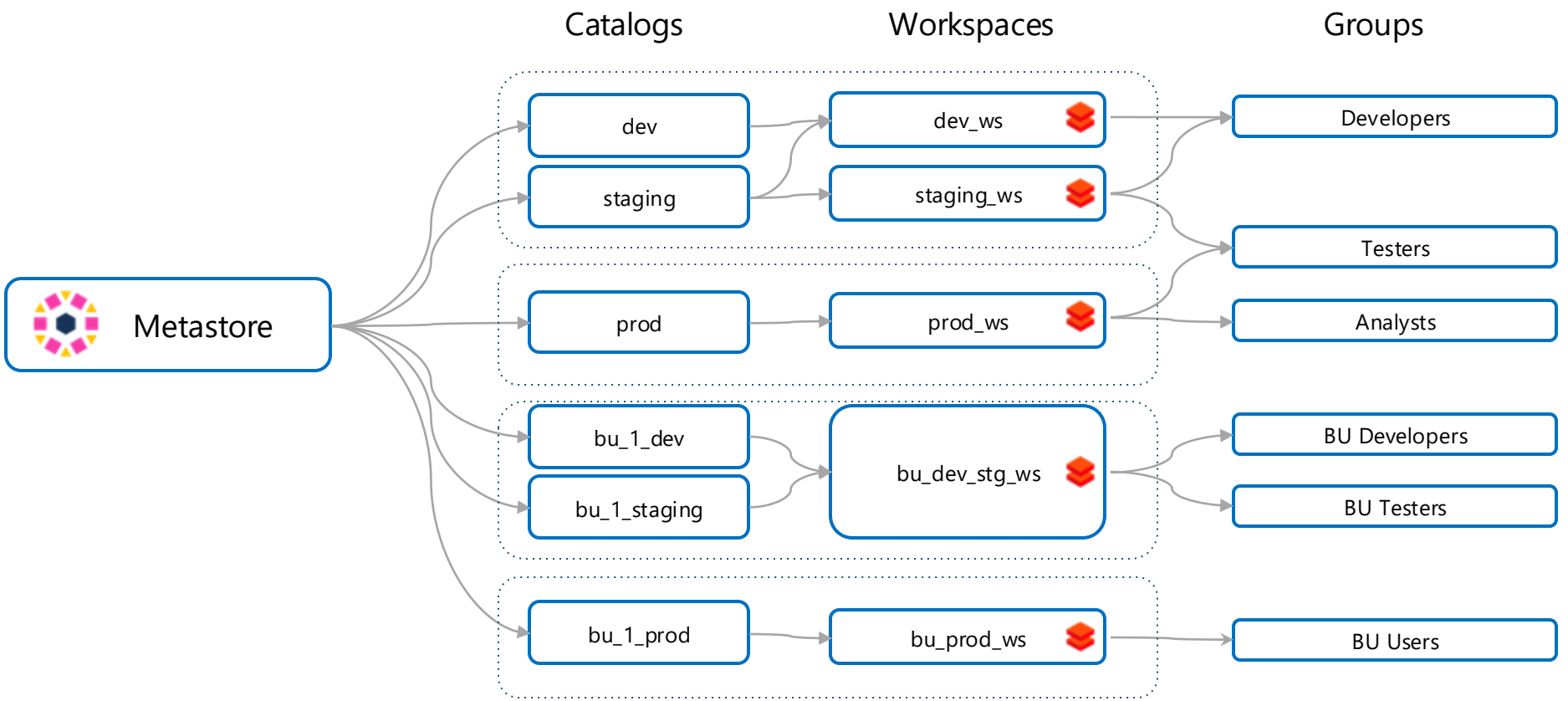
Govern filesystems and objects distinctly

Govern external tables and filesystem access separately



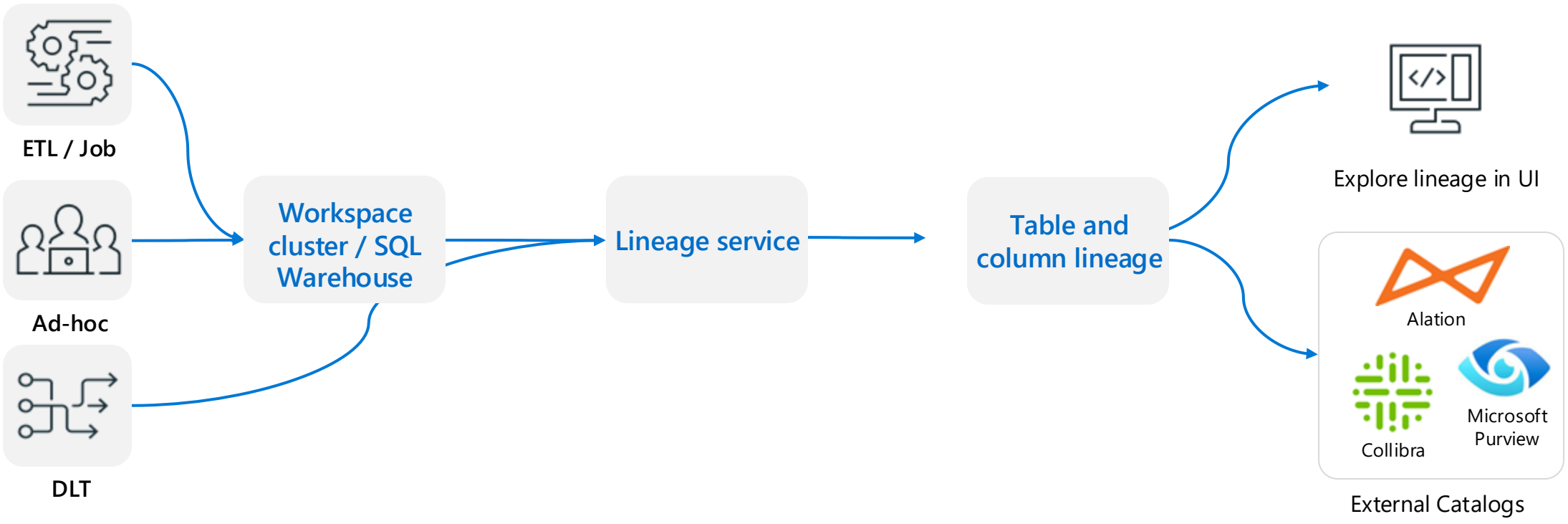
Access data from specified environments only

Restrict data access by environment or purpose



Access to data and availability of data can be isolated across workspaces and groups

Lineage flow

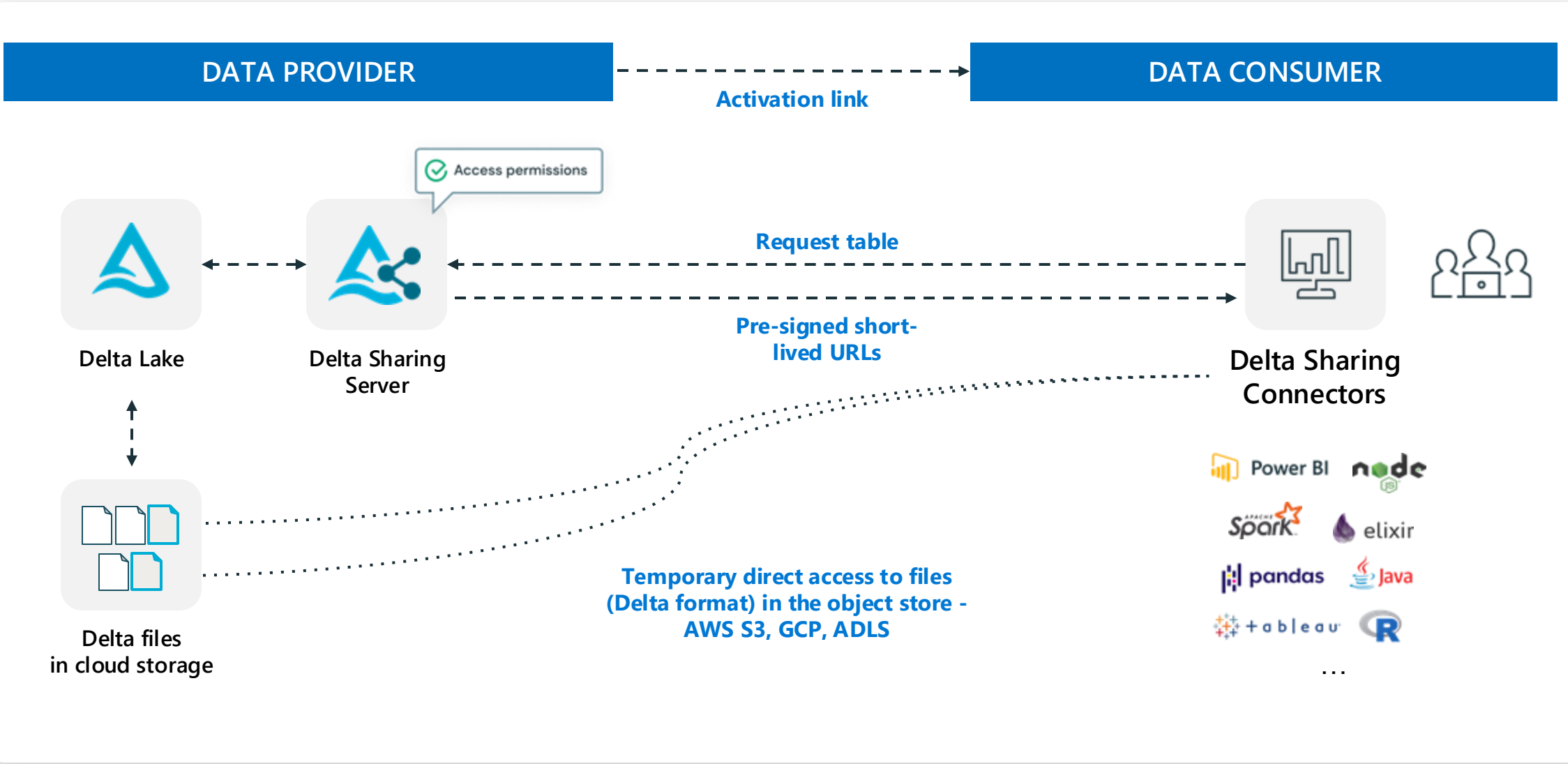


Code (any language) is submitted to a cluster or SQL warehouse or DLT* executes data flow

- Lineage service analyzes logs emitted from the cluster, and pulls metadata from DLT
- Assembles column and table level lineage

- Presented to the end user graphically in Databricks
- Lineage can be exported via API and imported into other tool

Delta Sharing



Lakehouse Federation and Unity Catalog

- Lakehouse Federation is the query federation platform for Azure Databricks
- Azure Databricks uses Unity Catalog to manage query federation

SQL

```
CREATE CONNECTION <connection-name> TYPE databricks
OPTIONS (
  host '<workspace-instance>',
  httpPath '<sql-warehouse-path>',
  personalAccessToken '<personal-access-token>'
);
```

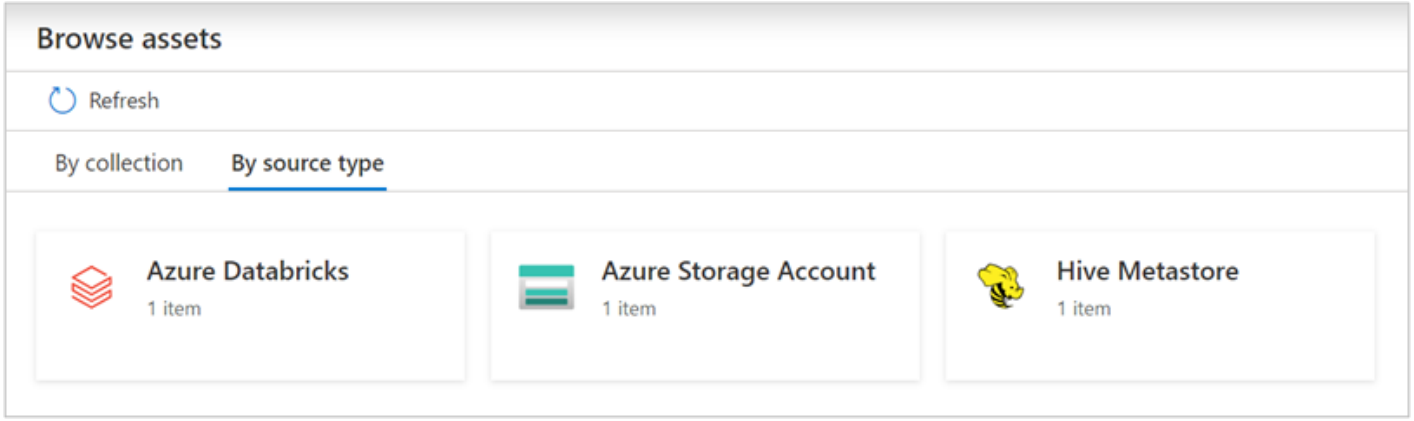
SQL Copy

```
CREATE FOREIGN CATALOG [IF NOT EXISTS] <catalog-name> USING CONNECTION <connection-name>
OPTIONS (database '<database-name>');
```

Integrate Azure Databricks with Microsoft Purview

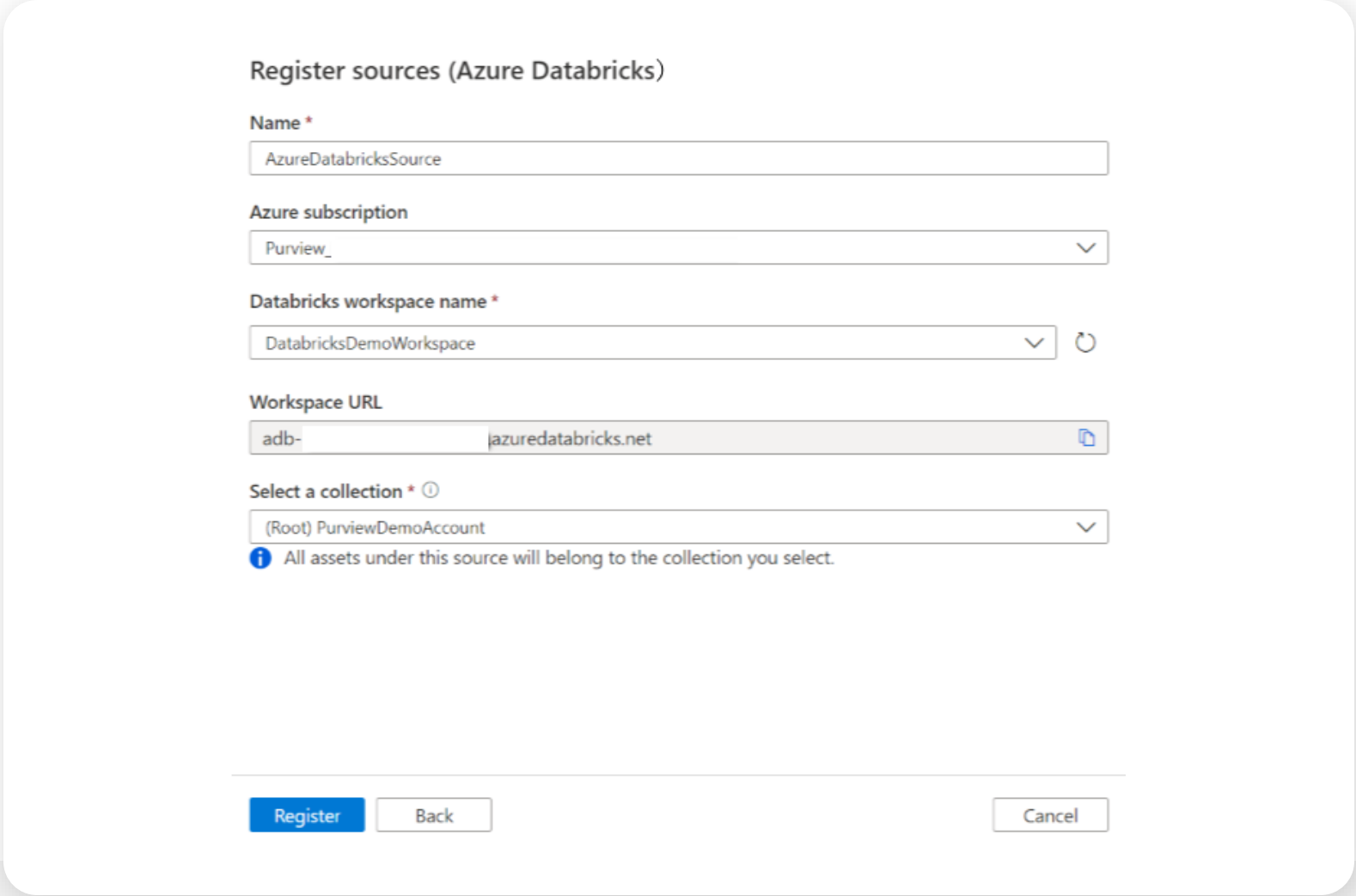
Integrate Azure Databricks with Microsoft Purview

- Extract technical metadata including:
  - Azure Databricks workspace
  - Hive server
  - Databases
  - Tables including the columns, foreign keys, unique constraints, and storage description
  - Views including the columns and storage description
- Fetch relationship between external tables and Azure Data Lake Storage Gen2/Azure Blob assets
- Fetch static lineage between tables and views based on the view definition



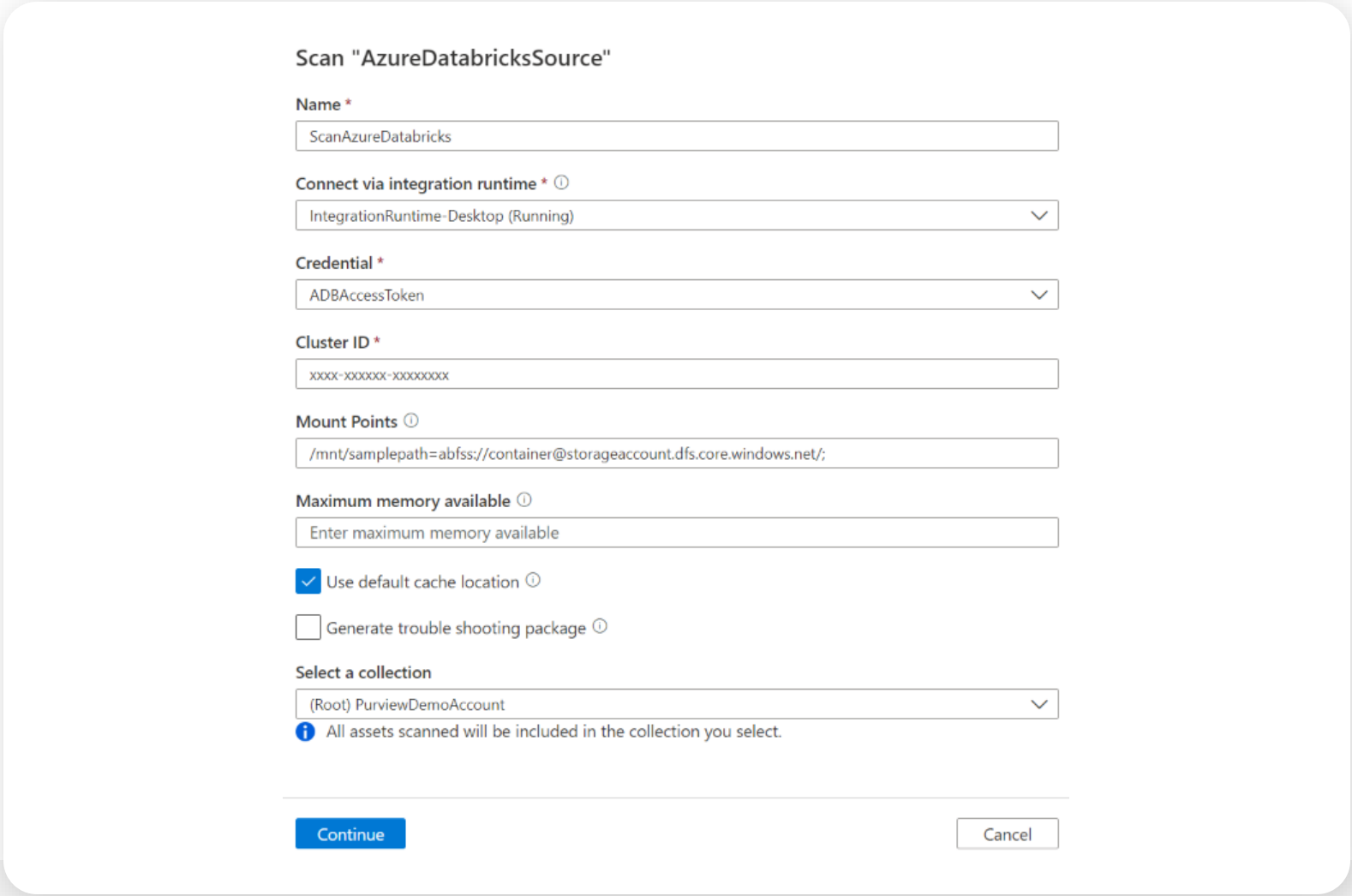
Register an Azure Databricks workspace in Microsoft Purview

- Register an Azure Databricks workspace in Microsoft Purview by using the Microsoft Purview governance portal



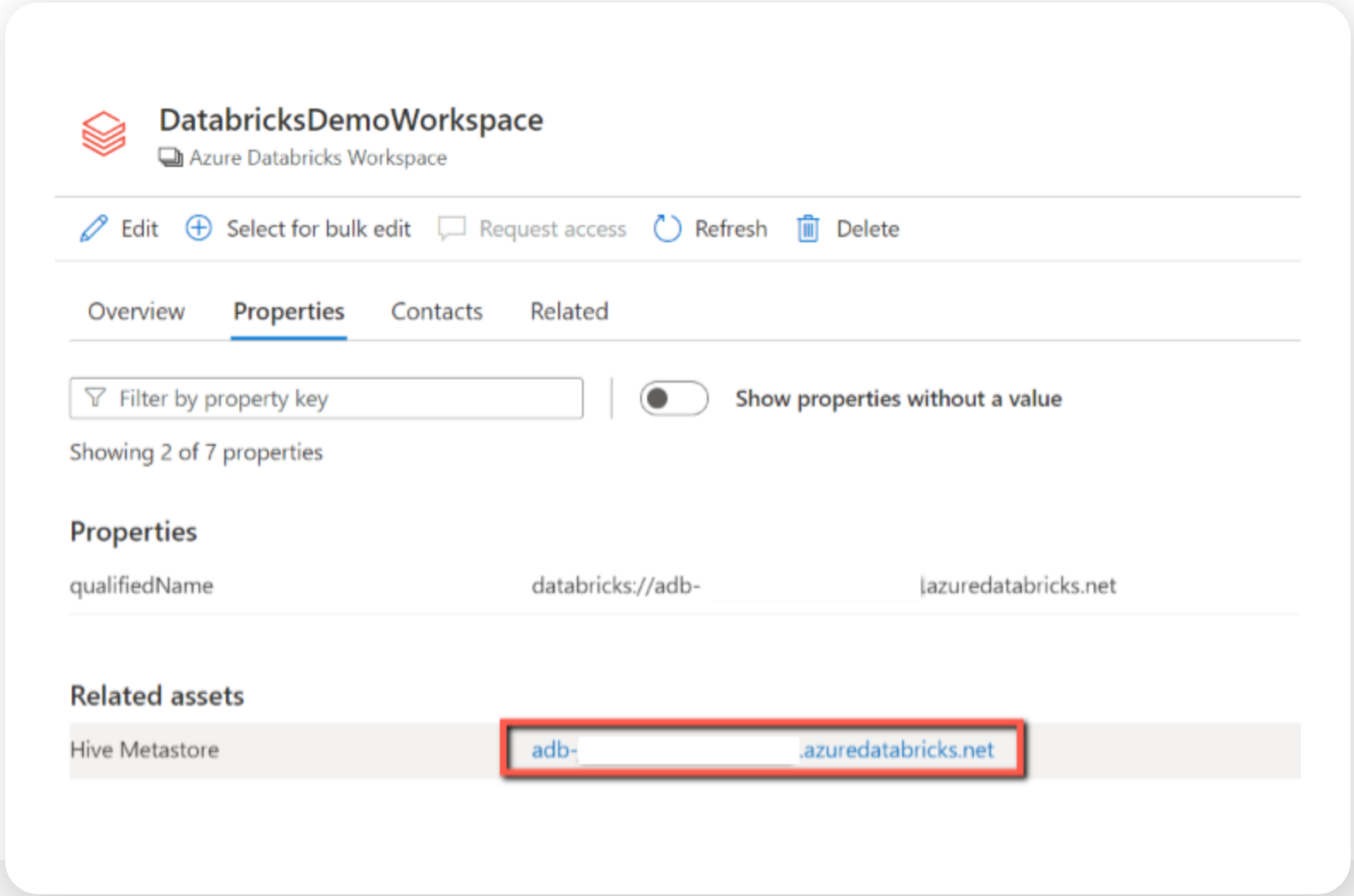
Scan Azure Databricks to automatically identify assets

- Scanning captures metadata from data sources and brings it to Microsoft Purview



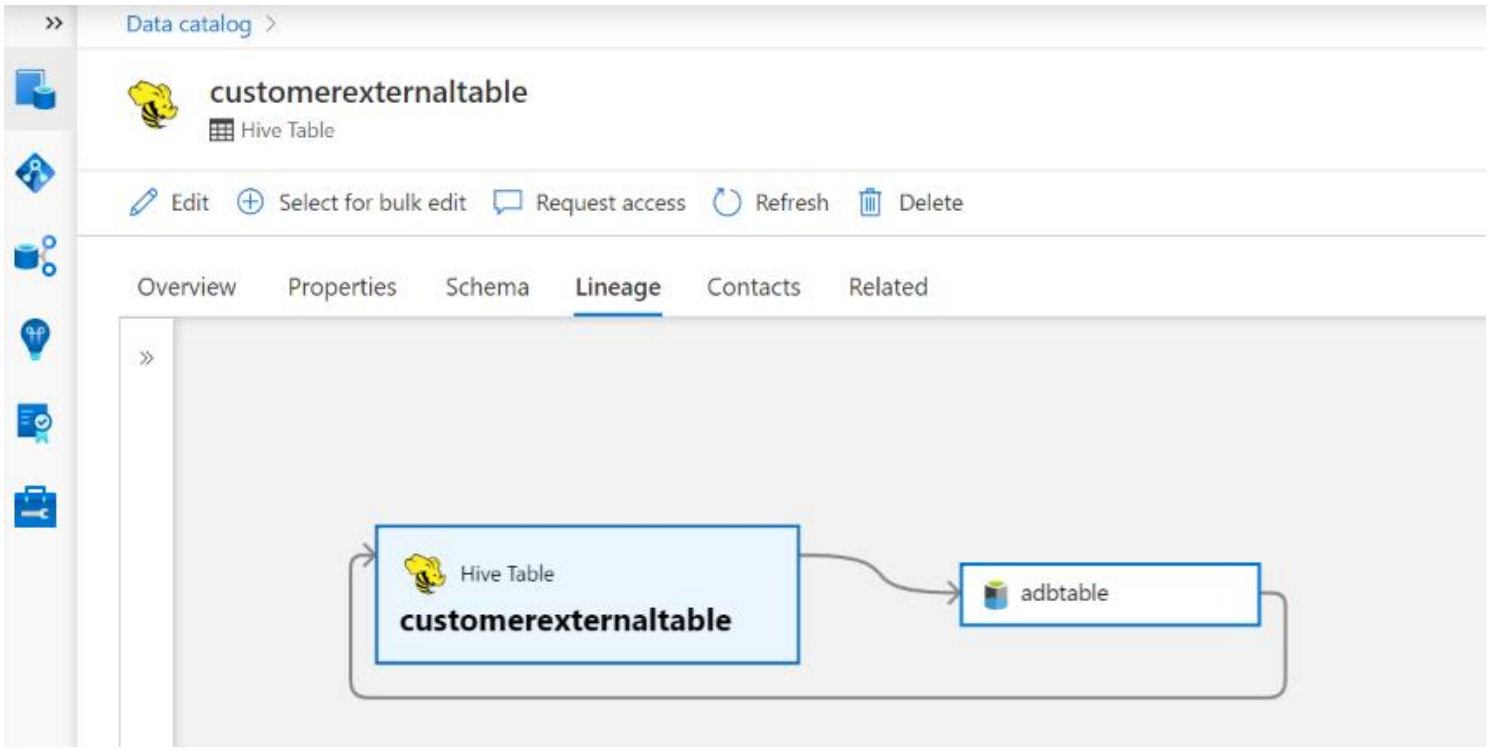
Browse and search assets

- After scanning your Azure Databricks, you can browse or search Unified Catalog to view the asset details



Data Lineage

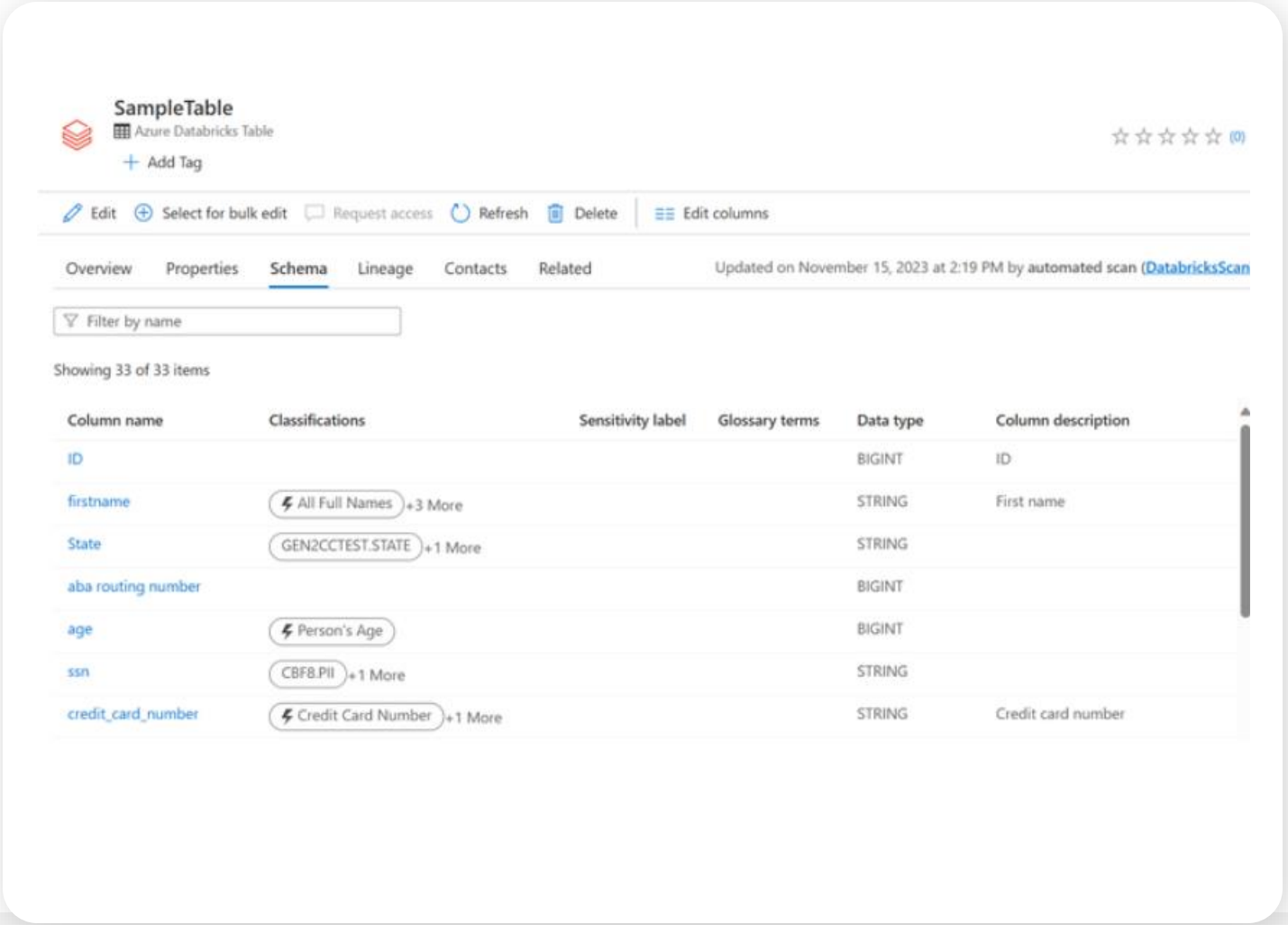
- Track data flow across Azure Databricks notebooks
- Improve the ability to audit, monitor, and manage data movement



Integration of Azure Databricks Unity Catalog with Microsoft Purview

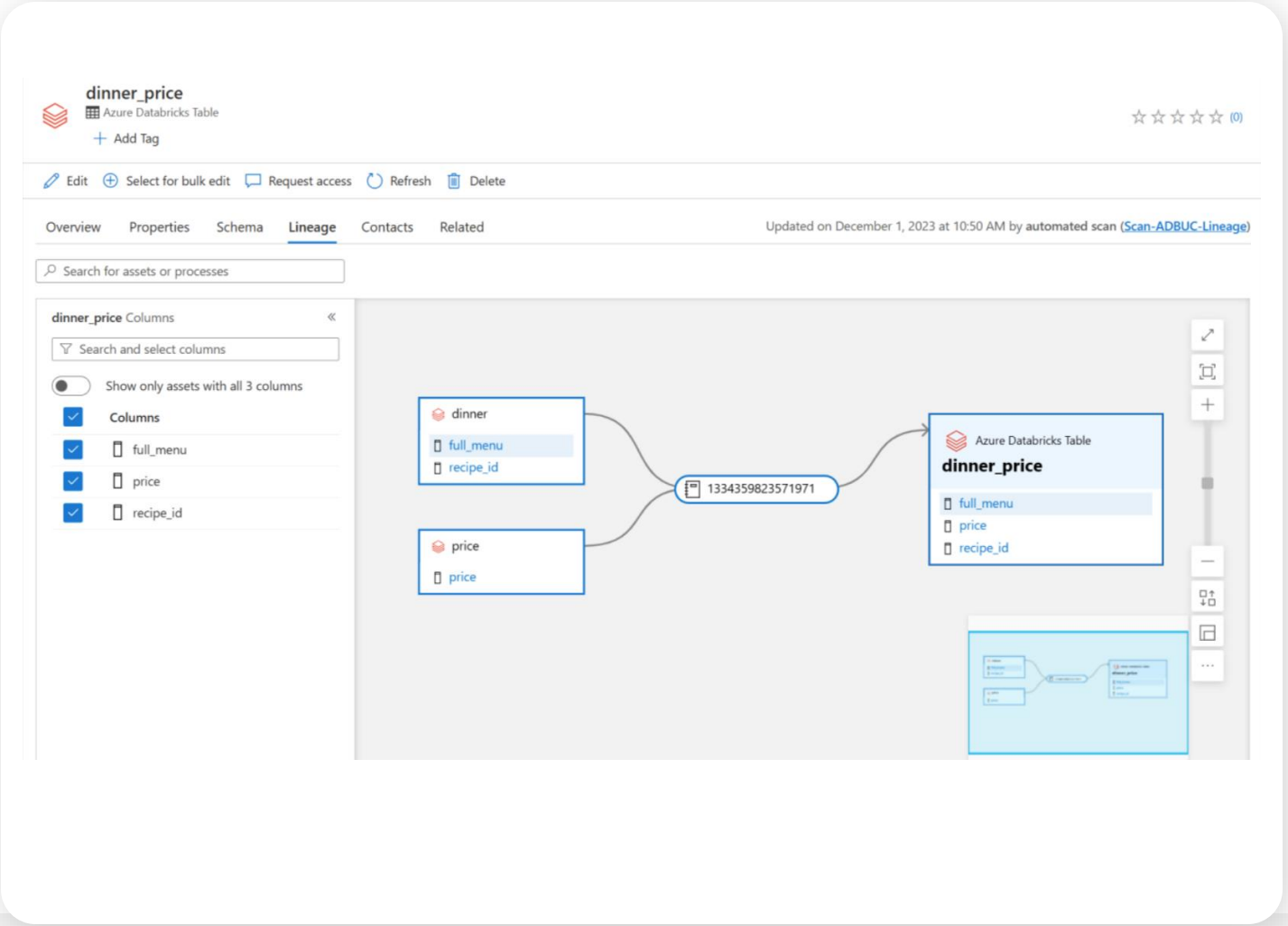
Better together: Microsoft Purview and Azure Databricks Unity Catalog

- Extract technical metadata including:
  - Metastore
  - Catalogs
  - Schemas
  - Tables including the columns
  - Views including the columns
- Fetch lineage on assets relationships between tables, views, columns during notebook runs



Data Lineage for Azure Databricks Unity Catalog in Microsoft Purview

- Track data flow across Azure Databricks notebooks
- Improve the ability to audit, monitor, and manage data movement
- Improve transparency
- Diagnose errors
- Ensure compliance with data governance policies



Demo

Connect to Azure Databricks Unity Catalog in Microsoft Purview

Coming up next...

Day 1

Module 1 - Introduction to Azure Databricks

- Azure Databricks: A Data Intelligent Platform
- Why Azure Databricks
- Decision guide: Azure Databricks vs. Microsoft Fabric

Module 2 - Migration to Azure Databricks

- Microsoft Cloud Adoption Framework for Azure
- Migration strategies
- Data landing zones
- Migration scenarios

Interactive Simulated Lab Experience

- End-to-End Streaming Pipeline with Lakeflow Declarative Pipelines in Azure Databricks

Day 2

Module 3 - Integration with Azure

- Seamless integration with Microsoft Azure services
- Connect to Azure Data Lake Storage (ADLS) Gen2 and Blob Storage
- Leverage Azure Databricks for Azure Cosmos DB Operations
- Secret management with Azure Key Vault
- Connect Azure Databricks to Azure Event Hubs

Module 4 - Integration with Microsoft Fabric and Power BI

- Data Intelligence with Azure Databricks and Microsoft Fabric
- Connect Power BI to Azure Databricks
- Integration with Azure Data Factory
- Mirroring Azure Databricks Unity Catalog

Interactive Simulated Lab Experience

- Setup and use Unity Catalog for Data Management in Azure Databricks
- Real-Time Streaming with Azure Databricks and Azure Event Hubs

Day 3

Module 5 - Integration with Azure AI Foundry

- Azure Databricks connector in Azure AI Foundry
- Mosaic AI and machine learning on Azure Databricks
- Query Generative AI model serving endpoints
- Databricks Assistant, AI/BI Genie and AI Functions on Azure Databricks
- Chat with LLMs and prototype GenAI apps using AI Playground
- Build and optimize agents on your data with Agent Bricks

Module 6 - Security and Governance

- Integrate Azure Databricks with Microsoft Purview
- Integration of Azure Databricks Unity Catalog with Microsoft Purview

Module 7 - Well-architected for Azure Databricks

- Lakehouse implementation: Principles and best practices
- Azure Databricks well-architected framework

Interactive Simulated Lab Experience

- Responsible AI with Large Language Models using Azure Databricks and Azure OpenAI
- Connect to and manage Azure Databricks in Microsoft Purview

Microsoft

Technofocus

Thank You!