

Ansh Ranjan

Azure Data

EXERCISE 5: DataBricks for Data Engineering

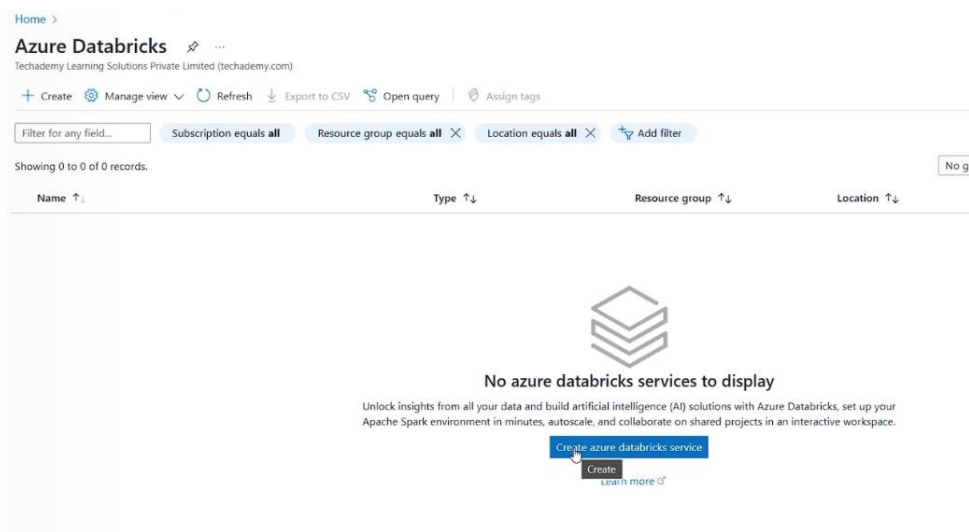
TASK 1: Deploy a DataBricks workspace

1. Sign in to the Azure Portal:

- Go to [Azure Portal](#) and log in with your credentials.

2. Create a New Resource:

- Click **Create a resource** on the left-hand menu.
- Search for **Azure Databricks** in the search bar.
- Select **Azure Databricks** from the results and click **Create**.



3. Configure the Workspace:

- Resource Group:** Choose an existing resource group or create a new one.
- Workspace Name:** Provide a unique name for your Databricks workspace.
- Region:** Select the Azure region closest to your users or data source for better performance.
- Pricing Tier:** Select a pricing tier based on your requirements (Standard, Premium, or Trial).

A screenshot of the 'Create an Azure Databricks workspace' form. The form has tabs for 'Basics', 'Networking', 'Encryption', 'Security & compliance', 'Tags', and 'Review + create'. The 'Basics' tab is selected. Under 'Project Details', there is a section for 'Subscription' and 'Resource group'. The 'Subscription' dropdown is set to 'MML Learners' and the 'Resource group' dropdown is set to 'rg-azuser2415_mmllocal-ga0V5'. Under 'Instance Details', there is a section for 'Workspace name', 'Region', and 'Pricing Tier'. The 'Workspace name' is set to 'anshdatabrick', the 'Region' is set to 'Central India', and the 'Pricing Tier' is set to 'Premium (+ Role-based access controls)'. At the bottom, there is a field for 'Managed Resource Group name' with the placeholder text 'Enter name for managed resource group'. A notification message at the bottom states: 'We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.'

4. Networking (Optional):

- Configure network settings if required, such as deploying the workspace in a Virtual Network (VNet).

5. Review + Create:

- Click **Review + Create** to validate your configuration.
- Once validation is successful, click **Create** to deploy the workspace.

6. Wait for Deployment:

- The deployment process might take a few minutes. Monitor the progress in the **Notifications** section of the portal.

✓ Your deployment is complete

Deployment name : rg-azuser2415_mmllocal-ga0V5_anshdatabrick1
Subscription : MML Learners
Resource group : rg-azuser2415_mmllocal-ga0V5

Start time : 11/28/2024, 11:11:43 AM
Correlation ID : 7fb0992d-e44f-47b0-98b7-d07e1f91fd9e

> Deployment details
v Next steps

[Go to resource](#)

Give feedback
[Tell us about your experience with deployment](#)

7. Access the Workspace:

- Once deployment is complete, go to the **Resource** to access the Databricks workspace.
- Click the **Launch Workspace** button to open the Azure Databricks environment.

TASK 2: Process a big sample dataset

1. Go to your DataBricks Workspace > Create > New Notebook and initialize your spark session. You will need a cluster to run queries

```
1 from pyspark.sql import SparkSession
2 spark = SparkSession.builder.appName("MovieLens").getOrCreate()
```

2. Extraction

```
1 df_data=spark.read.load("dbfs:/FileStore/shared_uploads/ranjanansh2002@gmail.com/u.data", format="csv", sep="\t", inferSchema="true")
2
```

(2) Spark Jobs

df_data: pyspark.sql.dataframe.DataFrame = [_c0: integer, _c1: integer ... 2 more fields]

```
1 df_data.display()
```

(1) Spark Jobs

	i^2_{c0}	i^2_{c1}	i^2_{c2}	i^2_{c3}
1	196	242	3	881250949
2	186	302	3	891717742
3	22	377	1	878887116
4	244	51	2	880606923
5	166	346	1	886397596

5. Since no column names are provided, we need to add them manually

```
1 column_mapping = {
2     "_c0": "UserId",
3     "_c1": "movieId",
4     "_c2": "rating",
5     "_c3": "timeStamp"
6 }
7
8 for col_old, col_new in column_mapping.items():
9     df_data=df_data.withColumnRenamed(col_old, col_new)
```

6. Getting Schema of the dataframe

```
1 df_data.printSchema()

root
 |-- UserId: integer (nullable = true)
 |-- movieId: integer (nullable = true)
 |-- rating: integer (nullable = true)
 |-- timeStamp: integer (nullable = true)
```

7. Repeating same steps for other dataframe and creating Views

```
1 df_data.createOrReplaceTempView('Data')
2 df_user.createOrReplaceTempView('User')
```

8. Deriving Analytics from the data

Gender wise user breakdown

```
1 spark.sql("select gender, count(*) from User group by gender").show()

└─ (2) Spark Jobs ─┘

+-----+-----+
|gender|count(1)|
+-----+-----+
| F |      273|
| M |      670|
+-----+-----+
```

Give the top 5 movies which are reviewed maximum number of times

```
1 spark.sql("select movieID, count(*) as Num_times from Data group by movieID order by Num_times desc limit 5").show()

└─ (2) Spark Jobs ─┘

+-----+-----+
|movieID|Num_times|
+-----+-----+
| 50 |      583|
| 258 |      509|
| 100 |      508|
| 181 |      507|
| 294 |      485|
+-----+-----+
```

List the top 10 movies which received highest number of 5 star ratings

```
1 spark.sql("select movieID, count(*) as Num_5star_reviews from Data where rating = 5 group by movieID order by Num_5star_reviews desc limit 5").show()

└─ (2) Spark Jobs ─┘

+-----+-----+
|movieID|Num_5star_reviews|
+-----+-----+
| 50 |          325|
| 100 |          227|
| 127 |          214|
| 174 |          202|
| 56 |          188|
+-----+-----+
```

TASK 3: DataBricks Key Features and use cases

Key Features of Databricks

- Unified platform for **data engineering, analytics, and ML**
- Built on **Apache Spark** for fast, distributed processing
- Supports **Delta Lake** with ACID transactions and schema enforcement
- Collaborative **notebooks** with multi-language support (Python, SQL, etc.)
- Built-in **visualizations** and **MLflow** for ML lifecycle
- **Auto-scaling clusters**, CI/CD, and cloud integration (Azure, AWS, GCP)

Use Cases

- **ETL & Data Pipelines:** Ingest, transform, and clean large datasets
- **Data Lakehouse:** Unified storage and analytics using Delta Lake
- **Machine Learning:** Build, train, and deploy ML models
- **Real-Time Analytics:** Process streaming data
- **Business Intelligence:** Connect with Power BI/Tableau for reporting