

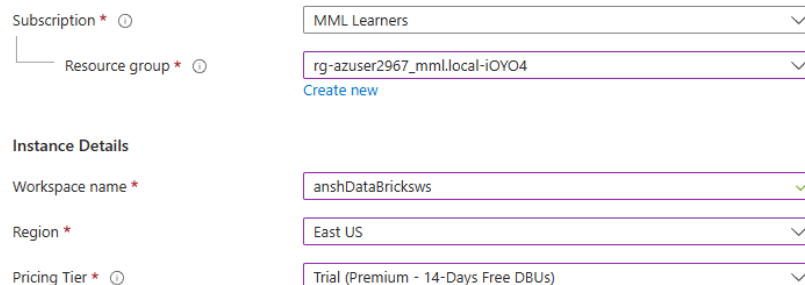
# Ansh Ranjan

## Azure Databricks

### Exercise 1 – Settings up DataBricks and Spark Basics

#### TASK 1: Create a new Azure DataBricks workspace

1. Go to Azure Portal > Azure DataBricks > Create > Enter details > Review and Create



Subscription \* ⓘ MML Learners

Resource group \* ⓘ rg-azuser2967\_mml.local-iOYO4  
[Create new](#)

**Instance Details**

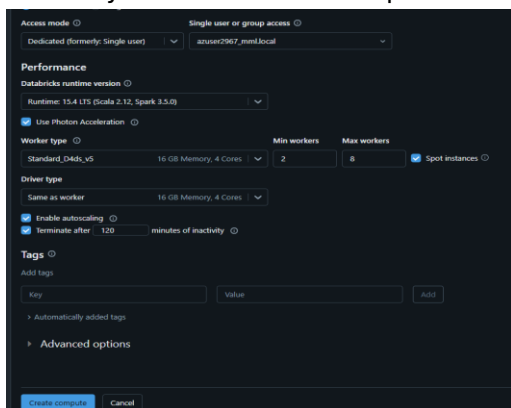
Workspace name \* anshDataBricksws ✓

Region \* East US ✓

Pricing Tier \* ⓘ Trial (Premium - 14-Days Free DBUs) ✓

#### TASK 2: Launch a spark cluster and explore databricks interface

1. Launch your Databricks workspace > Computer side tab > Create Compute



Access mode ⓘ Single user or group access ⓘ

Dedicated (formerly: Single user) azuser2967\_mml.local

**Performance**

Databricks runtime version ⓘ

Runtime: 13.4 LTS (Scala 2.12, Spark 3.5.0)

☒ Use Photon Acceleration ⓘ

Worker type ⓘ

Standard\_D4ds\_v5 16 GB Memory, 4 Cores

Min workers 2 Max workers 8 ☒ Spot instances ⓘ

Driver type

Same as worker 16 GB Memory, 4 Cores

☒ Enable autoscaling ⓘ

☒ Terminate after 120 minutes of inactivity ⓘ

**Tags** ⓘ

Add tags

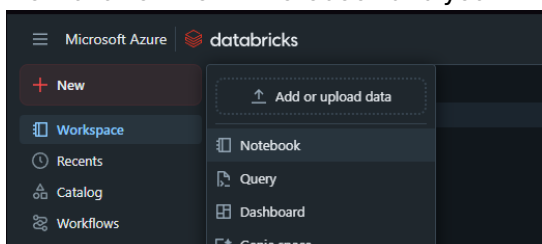
Key Value Add

Automatically added tags

Advanced options

Create compute Cancel

2. Now click on New > Notebook and you will have your notebook ready in your workspace



The **Azure Databricks workspace** provides a unified environment for data engineering, data science, and machine learning. The main parts of the interface include:

#### 1. Workspace

- Organize notebooks, libraries, and workflows.
- Create folders and share them with users or groups.

#### 2. Notebooks

- Interactive notebooks supporting **Python, SQL, Scala,** and **R.**
- Run code in cells and visualize data easily.

### 3. Clusters

- Spin up Spark clusters for running jobs or interactive analysis.
- Choose autoscaling and runtime version (with Delta, ML, or GPU support).

### 4. Jobs

- Schedule notebooks or workflows.
- Automate ETL, ML training, or batch jobs.

### 5. Data

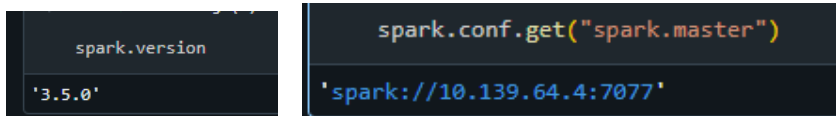
- Browse databases, tables, and files.
- Supports Unity Catalogue (if enabled) for secure, centralized governance.

### 6. Repos (Git Integration)

- Connect to GitHub or Azure DevOps to version-control notebooks and code.

## TASK 3: Run Basic Spark commands

1. Running spark.version command in first cell



The image shows two side-by-side screenshots of a Jupyter notebook interface. The left screenshot shows a code cell with the command `spark.version` and its output `'3.5.0'`. The right screenshot shows a code cell with the command `spark.conf.get("spark.master")` and its output `'spark://10.139.64.4:7077'`.

Now that we know our databricks workspace is ready we can start performing ETL tasks