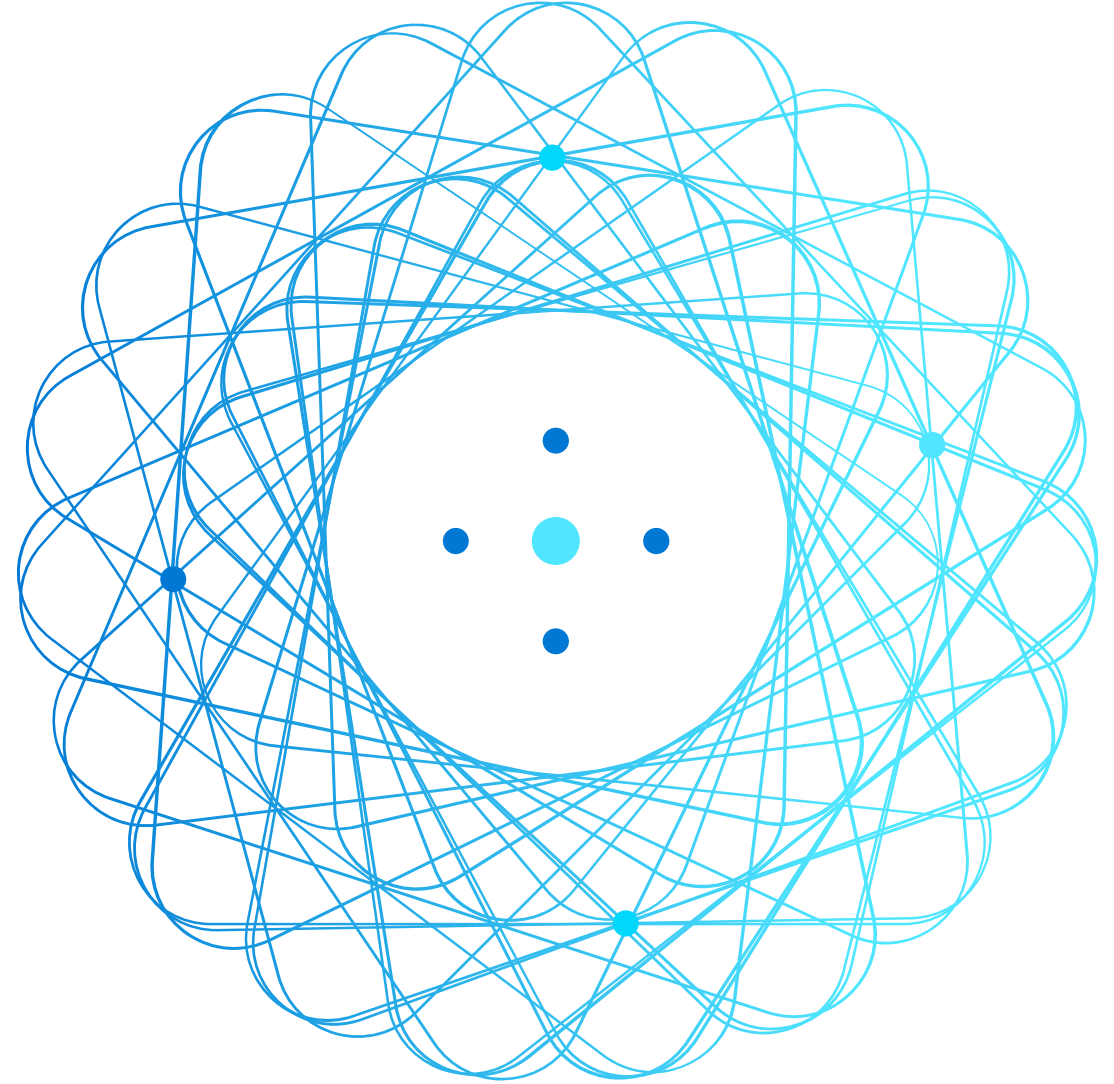


# Data engineering with Azure Databricks



# Agenda



Explore Azure Databricks



Use Apache Spark in Azure Databricks



Run Azure Databricks notebooks in Azure Data Factory

# Explore Azure Databricks



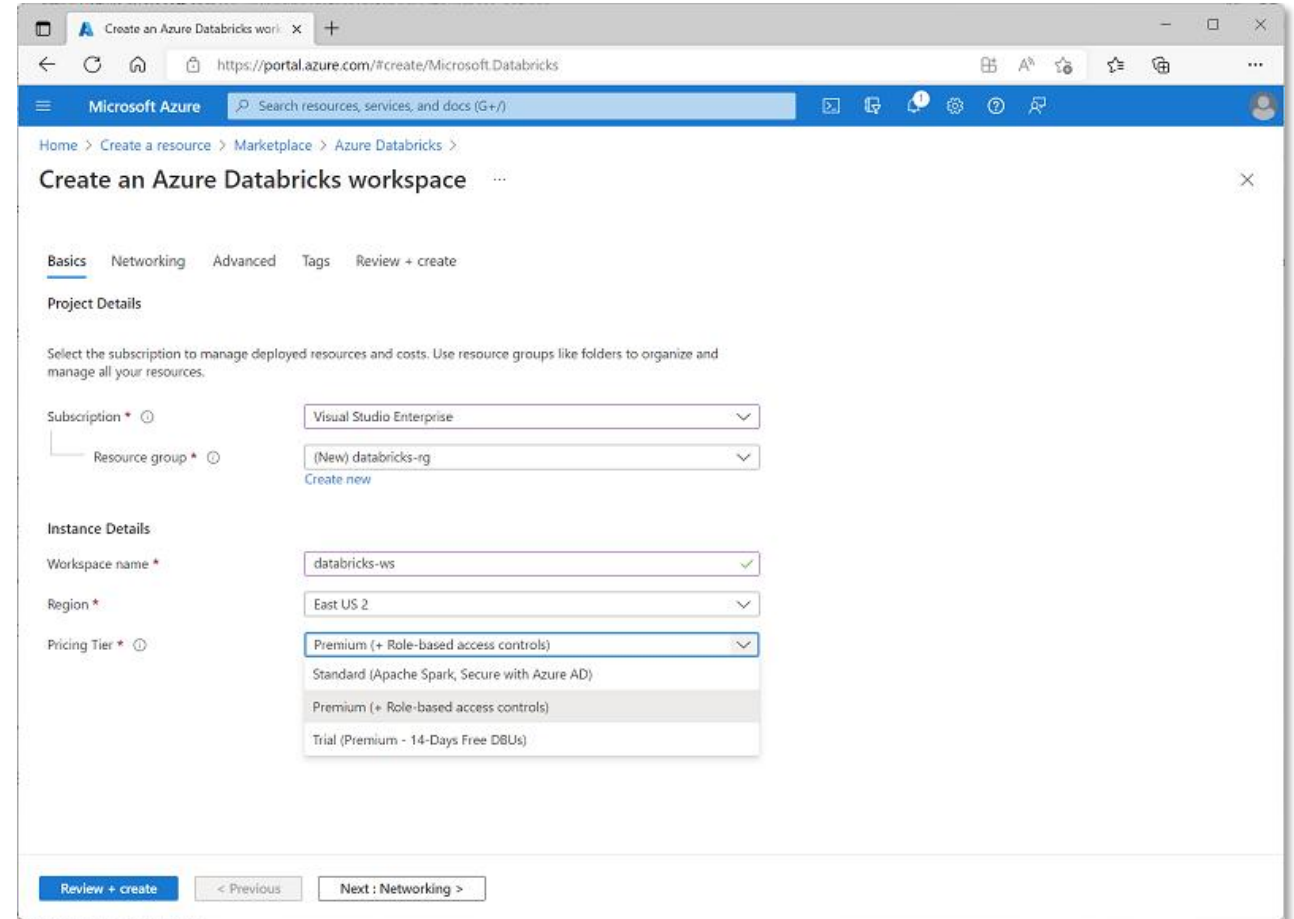
# What is Azure Databricks?

**Fully managed, cloud-based data analytics platform**

- Built on Apache Spark
- Web-based portal

**Provisioned as an Azure resource**

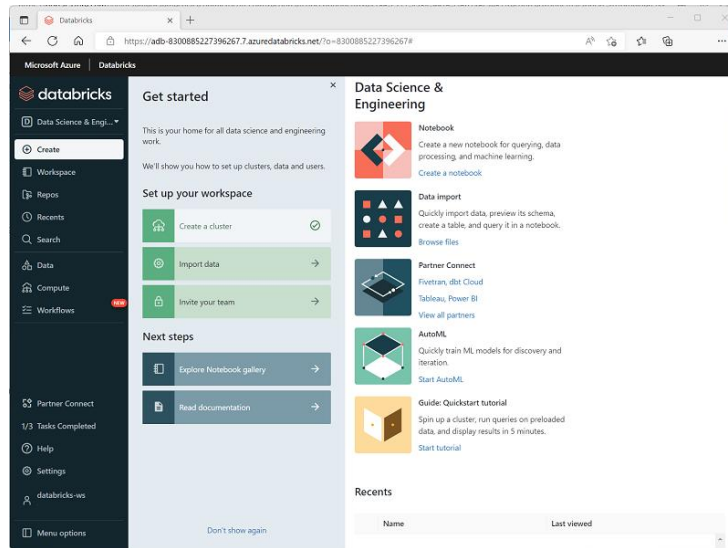
- Standard tier
- Premium tier
- Trial



The screenshot shows the 'Create an Azure Databricks workspace' page in the Azure portal. The page is titled 'Create an Azure Databricks workspace' and has a breadcrumb trail: Home > Create a resource > Marketplace > Azure Databricks >. The page is divided into tabs: Basics, Networking, Advanced, 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 'Visual Studio Enterprise'. The 'Resource group' dropdown is set to '(New) databricks-rg' with a 'Create new' link below it. Under 'Instance Details', there is a section for 'Workspace name', 'Region', and 'Pricing Tier'. The 'Workspace name' is 'databricks-ws'. The 'Region' is 'East US 2'. The 'Pricing Tier' dropdown is open, showing options: 'Premium (+ Role-based access controls)' (selected), 'Standard (Apache Spark, Secure with Azure AD)', 'Premium (+ Role-based access controls)', and 'Trial (Premium - 14-Days Free DBUs)'. At the bottom, there are three buttons: 'Review + create', '< Previous', and 'Next : Networking >'.

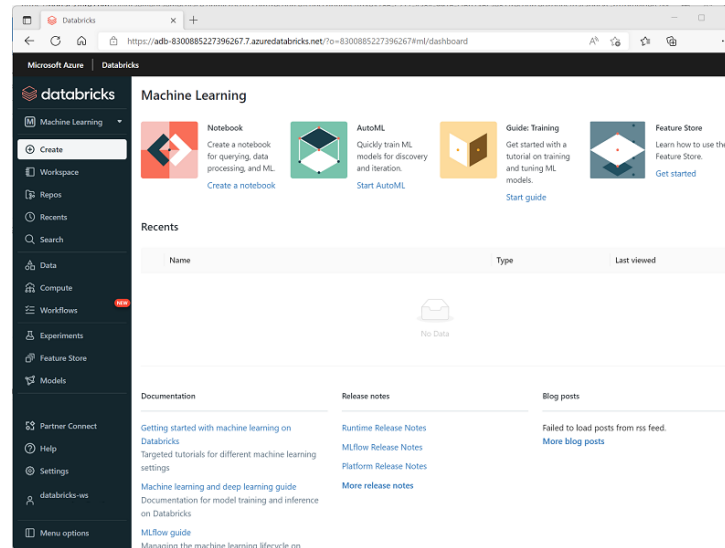
# Azure Databricks workloads

## Data Science and Engineering



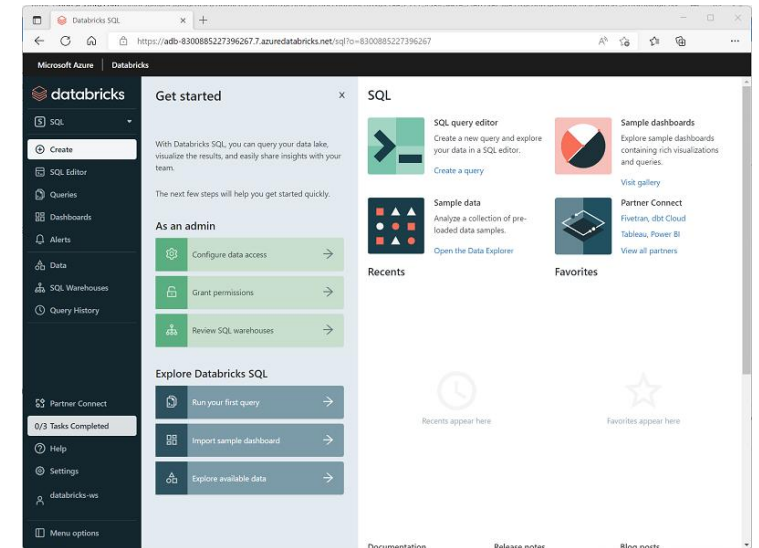
Use notebooks to run Apache Spark code to manipulate and explore data

## Machine Learning



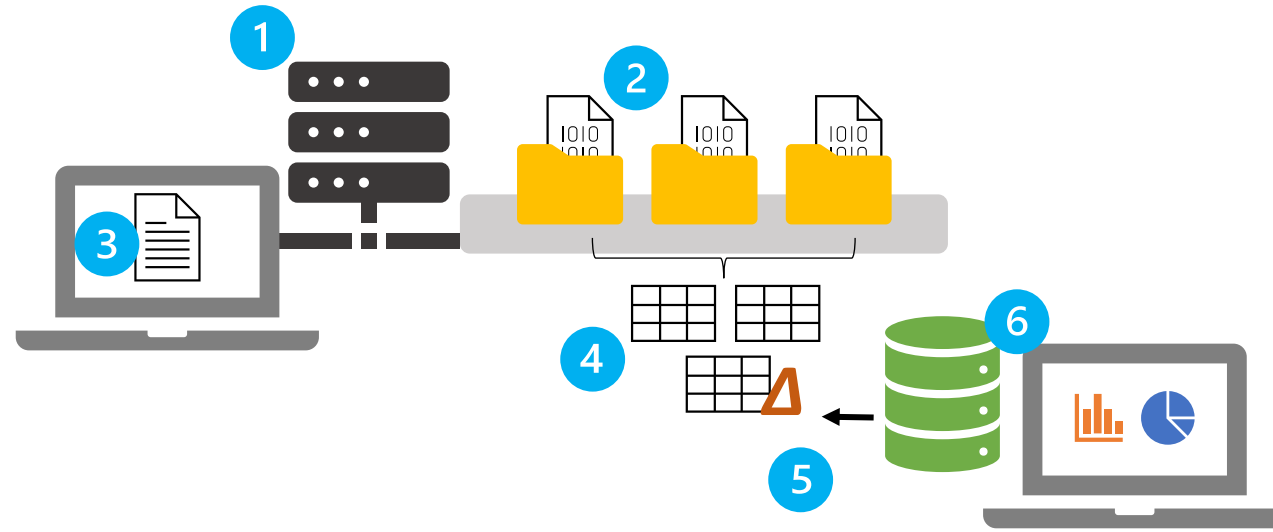
Train predictive models using SparkML and other machine learning frameworks

## SQL



Store and query data in relational tables using SQL  
Only available in *Premium* tier workspaces

# Key concepts



1. **Apache Spark clusters** provide highly scalable parallel compute for distributed data processing
2. **Databricks File System (DBFS)** provides distributed shared storage for data lakes
3. **Notebooks** provide an interactive environment for combining code, notes, and images
4. **Metastore** provides a relational abstraction layer, enabling you to define tables based on data in files
5. **Delta Lake** builds on the metastore to enable common relational database capabilities
6. **SQL Warehouses** provide relational compute endpoints for querying data in tables

# Demo: Explore Azure Databricks

You can try this for yourself later  
by following the instructions at the  
link below:

<https://aka.ms/mslearn-adb>



# Knowledge check



You plan to create an Azure Databricks workspace and use the SQL persona view in the Azure Databricks portal. Which of the following pricing tiers can you select?

- ☐ Enterprise
  - ☐ Standard
  - ☒ Premium
- 



You need to use Spark to process data in files, preparing it for analysis. Which persona view should you use in the Azure Databricks portal?

- ☒ Data Science and Engineering
  - ☐ Machine Learning
  - ☐ SQL
- 



You've created an Azure Databricks workspace in which you plan to use code in notebooks to process data. What must you create in the workspace?

- ☐ A SQL Warehouse
- ☒ A Spark cluster
- ☐ A Windows Server virtual machine



# Use Apache Spark in Azure Databricks



# Create a Spark cluster

## Create a cluster in the Azure Databricks portal, specifying:

- Cluster name
- Cluster mode (standard, high-concurrency, or single-node)
- Databricks Runtime version
- Worker and driver node VM configuration
- Autoscaling and automatic shutdown

The screenshot displays the 'Create Cluster' interface in the Azure Databricks portal. The page is titled 'My Cluster' and includes a sidebar with navigation icons. The main content area contains several configuration sections:

- Cluster mode:** A dropdown menu set to 'Standard'.
- Performance:**
  - Databricks runtime version:** A dropdown menu set to 'Runtime: 10.4 LTS (Scala 2.12, Spark 3.2.1)'.
  - ☐ Use Photon Acceleration
  - Worker type:** A dropdown menu set to 'Standard\_DS3\_v2' with a sub-menu showing '14 GB Memory, 4 Cores'. To its right are input fields for 'Min workers' (set to 2) and 'Max workers' (set to 8), and a checkbox for 'Spot instances'.
  - Driver type:** A dropdown menu set to 'Same as worker' with a sub-menu showing '14 GB Memory, 4 Cores'.
  - ☒ Enable autoscaling
  - ☒ Terminate after: 120 minutes of inactivity
- Tags:** A section with 'Add tags' input fields for 'Key' and 'Value', and an 'Add' button. Below it, a link says '> Automatically added tags'.

On the right side, a 'Summary' panel provides a quick overview of the cluster configuration:

Summary	
2-8 Workers	28-112 GB Memory 8-32 Cores
1 Driver	14 GB Memory, 4 Cores
Runtime	10.4.x-scala2.12
Standard_DS3_v2	2.25 - 6.75 DBU/h

At the bottom of the page, there are 'Create Cluster' and 'Cancel' buttons.

# Use Spark in notebooks

## Interactive notebooks

- Syntax highlighting and error support
- Code auto-completion
- Interactive data visualizations
- The ability to export results

The screenshot displays the Databricks web interface. On the left is a dark sidebar with navigation options: Data Science & Engi..., Create, Workspace (highlighted), Repos, Recents, Search, Data, Compute, Workflows (marked with a 'NEW' badge), Partner Connect, 1/3 Tasks Completed, Help, Settings, databricks-ws, and Menu options. The main area is titled 'Workspace' and shows a tree view with 'Workspace', 'Shared', and 'Users' folders. A 'My Notebook' is listed under the 'Shared' folder. The notebook itself is titled 'My Notebook' and is in 'Python' mode. It contains a code cell with the following code:

```
1 df1 = spark.read.format("csv").option("header",  
2 "true").load("dbfs:/FileStore/products.csv")  
3 display(df1)
```

Below the code, it shows '(2) Spark Jobs' and a summary of the first job: 'df1: pyspark.sql.dataframe.DataFrame = [ProductID: string, ProductName: string ... 2 more fields]'. A 'Table' tab is selected, displaying a table with 7 rows and 4 columns: ProductID, ProductName, and Ca. The table data is as follows:

	ProductID	ProductName	Ca
1	771	Mountain-100 Silver, 38	Mc
2	772	Mountain-100 Silver, 42	Mc
3	773	Mountain-100 Silver, 44	Mc
4	774	Mountain-100 Silver, 48	Mc
5	775	Mountain-100 Black, 38	Mc
6	776	Mountain-100 Black, 42	Mc
7	777	Mountain-100 Black, 44	Mc

At the bottom, it indicates 'Showing all 295 rows.' and 'Command took 0.88 seconds'.

# Use Spark to work with data files

## Dataframe API

```
%pyspark
df=spark.read.load('/data/products.csv',
    format='csv',
    header=True
)
display(df.limit(10))
```

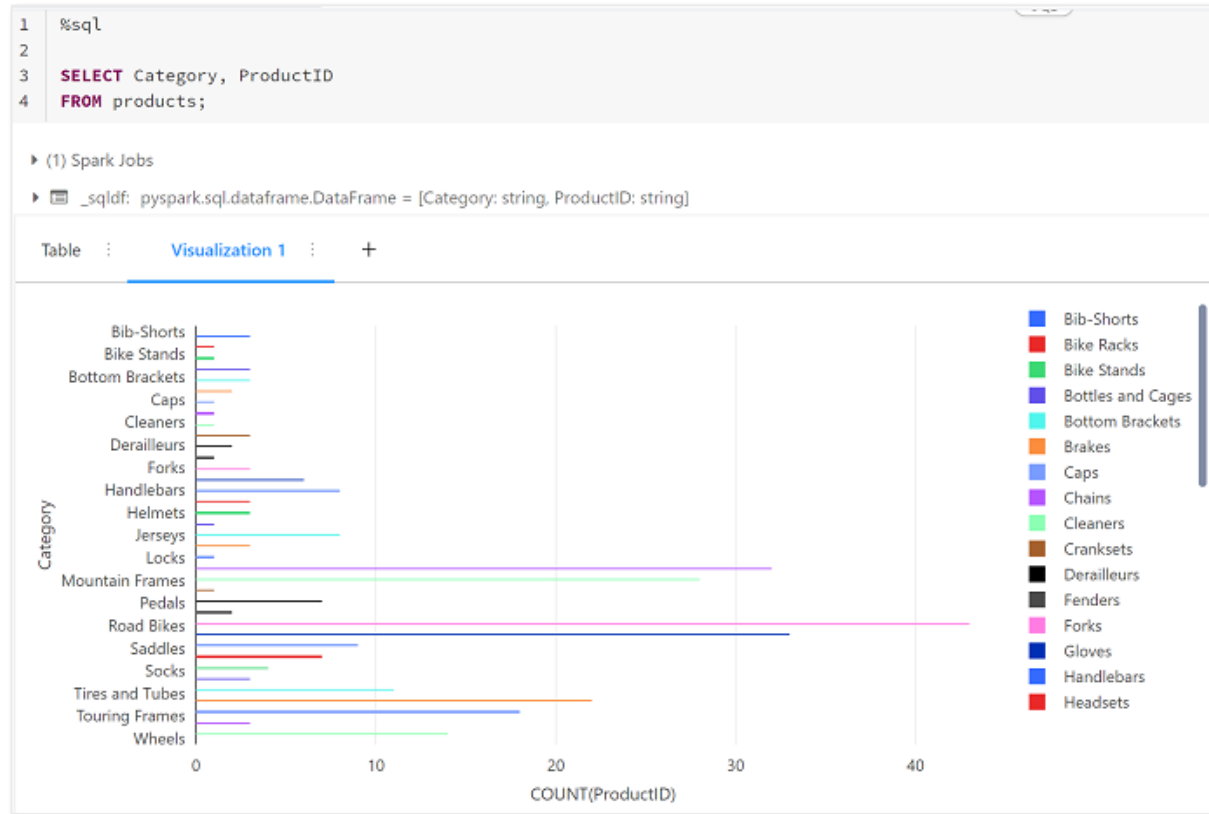
## Spark SQL API

```
%pyspark
df.createOrReplaceTempView("products")
```

```
%sql
SELECT Category, COUNT(ProductID) AS ProductCount
FROM products
GROUP BY Category
ORDER BY Category
```

# Visualize data

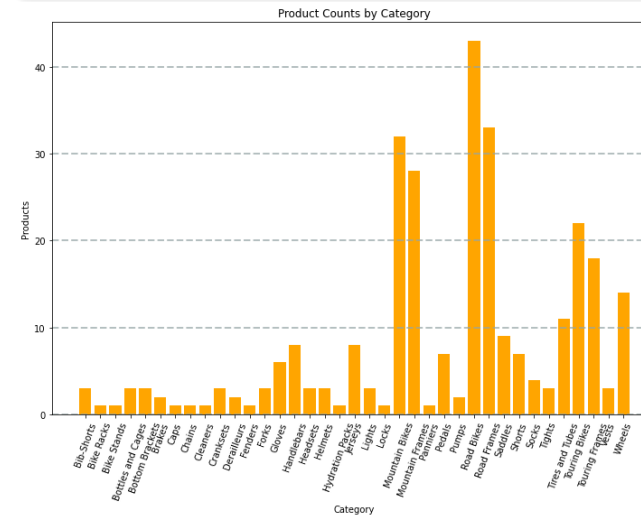
## Built-in charts



## Graphics packages

```
from matplotlib import pyplot as plt

fig = plt.figure(figsize=(12,8))
plt.bar(x=data['Category'],
        height=data['ProductCount'],
        color='orange')
plt.show()
```



# Exercise: Use Spark in Azure Databricks

Use the hosted lab environment provided, or view the lab instructions at the link below:

<https://aka.ms/mslearn-databricks-spark>



# Knowledge check



**Which definition best describes Apache Spark?**

- ☐ A highly scalable relational database management system
  - ☐ A virtual server with a Python runtime
  - ☒ A distributed platform for parallel data processing using multiple languages
- 



**You need to use Spark to analyze data in a parquet file. What should you do?**

- ☒ Load the parquet file into a dataframe
  - ☐ Import the data into a table in a serverless SQL pool
  - ☐ Convert the data to CSV format
- 



**You want to write code in a notebook cell that uses a SQL query to retrieve data from a view in the Spark catalog. Which magic should you use?**

- ☐ %spark
- ☐ %pyspark
- ☒ %sql

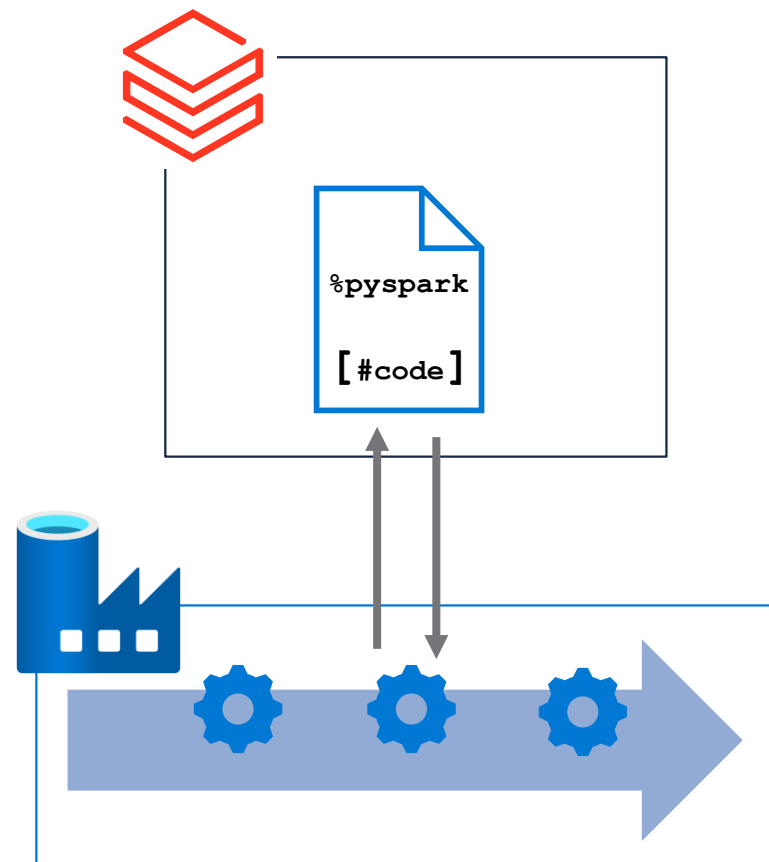
# Run Azure Databricks notebooks in Azure Data Factory



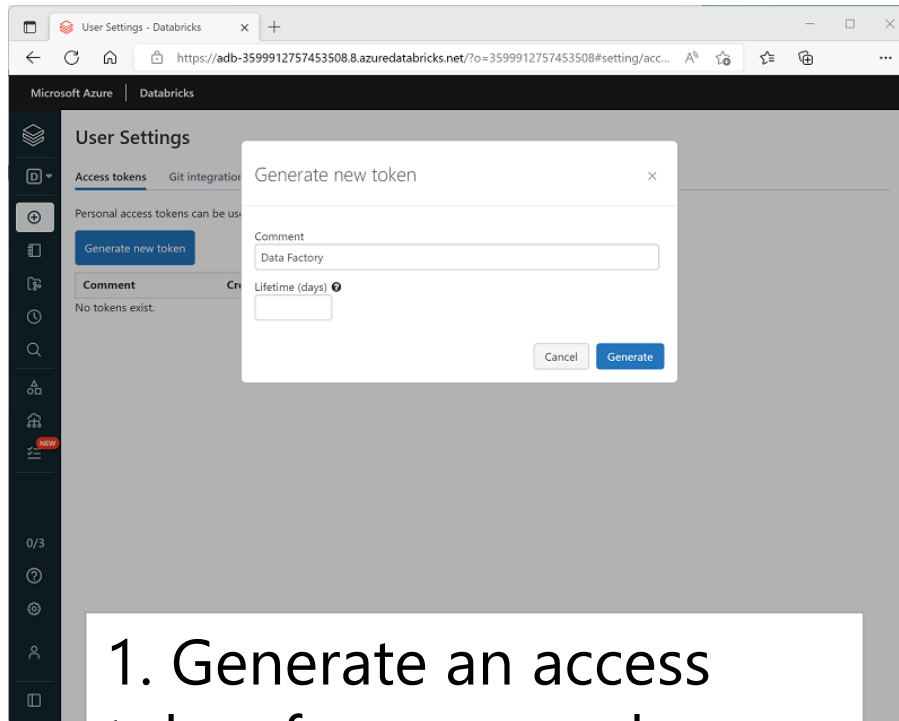


# Azure Databricks notebooks and pipelines

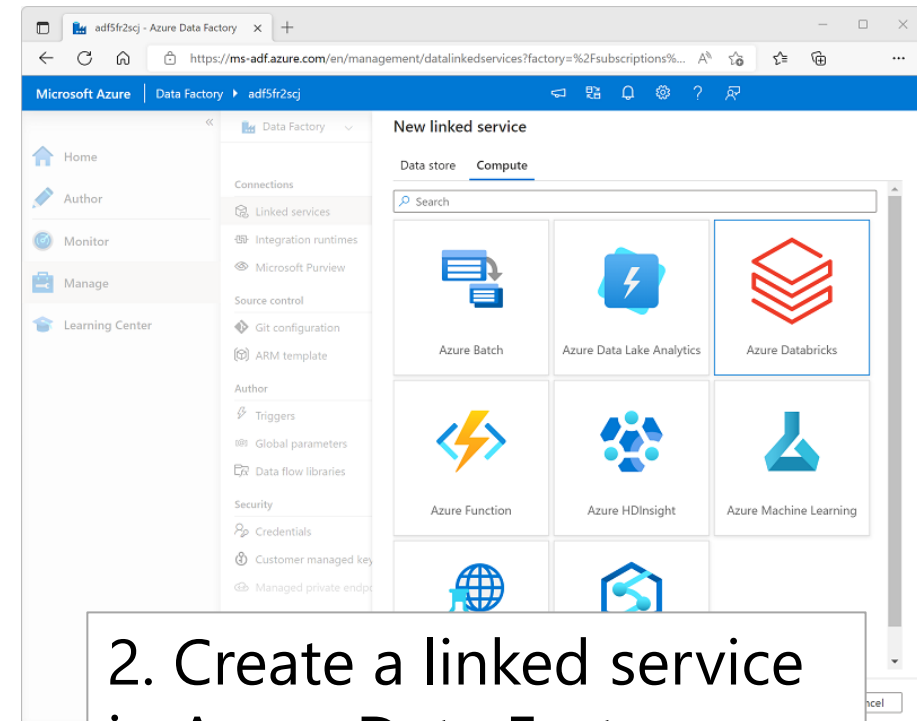
- Use notebooks to develop and test data transformation code
- Incorporate notebooks into data ingestion and transformation pipelines using Azure Data Factory or Azure Synapse Analytics
- Run pipelines on-demand, at scheduled times, or in response to events



# Create a linked service for Azure Databricks



1. Generate an access token for your workspace

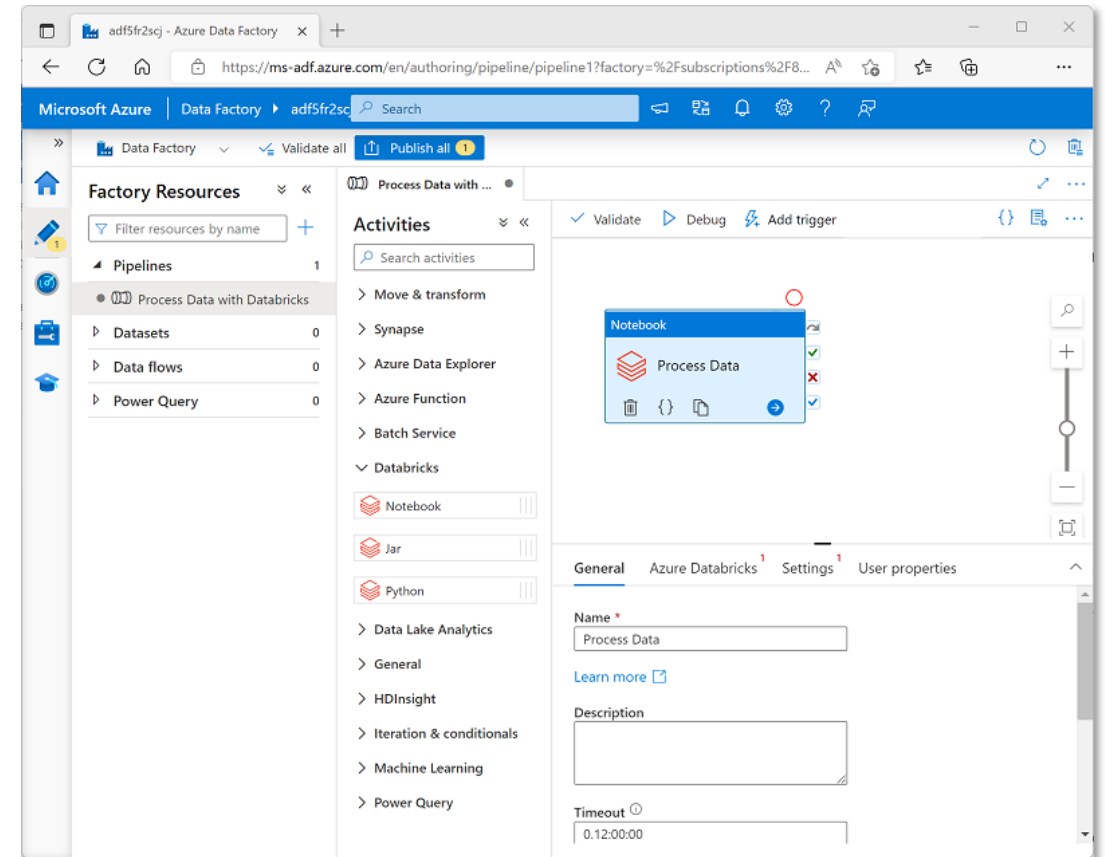


2. Create a linked service in Azure Data Factory

# Use a Notebook activity in a pipeline

Add a *Notebook* activity to a pipeline, specifying:

- **General** properties such as name, timeout, and number of retries
- **Azure Databricks** properties - the linked service for your workspace
- **Settings**, such as the notebook path and parameter details
- **User properties** to define custom configuration values



# Use parameters in a notebook

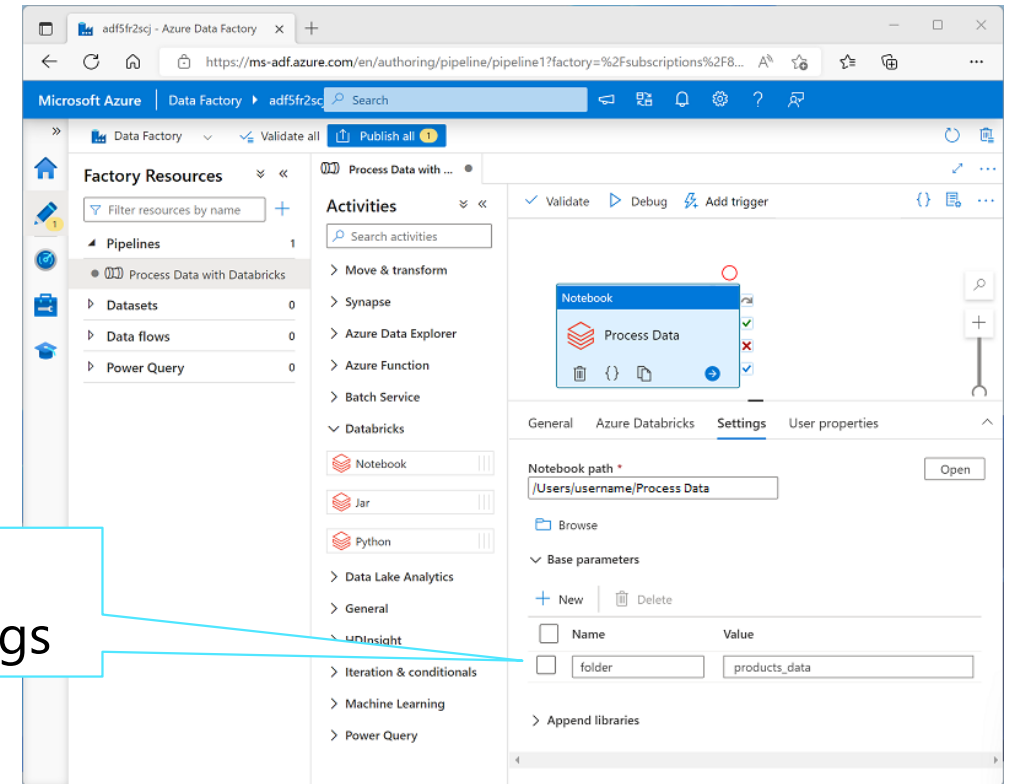
Use the ***dbutils*** library to work with parameters

```
# define parameter with default value
dbutils.widgets.text("folder", "data")

# Get input parameter value if passed
folder = dbutils.widgets.get("folder")
...

# Return output parameter
path = "dbfs://{0}/products.csv".format(folder)
dbutils.notebook.exit(path)
```

Set parameter values in notebook activity settings



# Demo: Run an Azure Databricks Notebook with Azure Data Factory

You can try this for yourself later by following the instructions at the link below:

<https://aka.ms/mslearn-databricks-factory>



# Knowledge check



You want to connect to an Azure Databricks workspace from Azure Data Factory. What must you define in Azure Data Factory?

- ☐ A global parameter
  - ☒ A linked service
  - ☐ A customer managed key
- 



You need to run a notebook in the Azure Databricks workspace referenced by a linked service. What type of activity should you add to a pipeline?

- ☒ Notebook
  - ☐ Python
  - ☐ Jar
- 



You need to use a parameter in a notebook. Which library should you use to define parameters with default values and get parameter values that are passed to the notebook?

- ☐ notebook
- ☐ argparse
- ☒ dbutils.widget

# Further reading



Data engineering with Azure Databricks  
<https://aka.ms/mslearn-azure-databricks>