

Title



Enjoy the most efficient handwriting experience! Taking notes on the go, whether for inspiration, ideas, knowledge learning, business insights, or even sketches...



Creative notes

By reading we enrich the mind;
by writing we polish it.

 Created by Notein

- Apache Spark Coordinates work across multiple processing nodes in a cluster, known in Microsoft Fabric as Spark Pool
- The process of distributing tasks and collating results is handled by Spark

Spark Pools

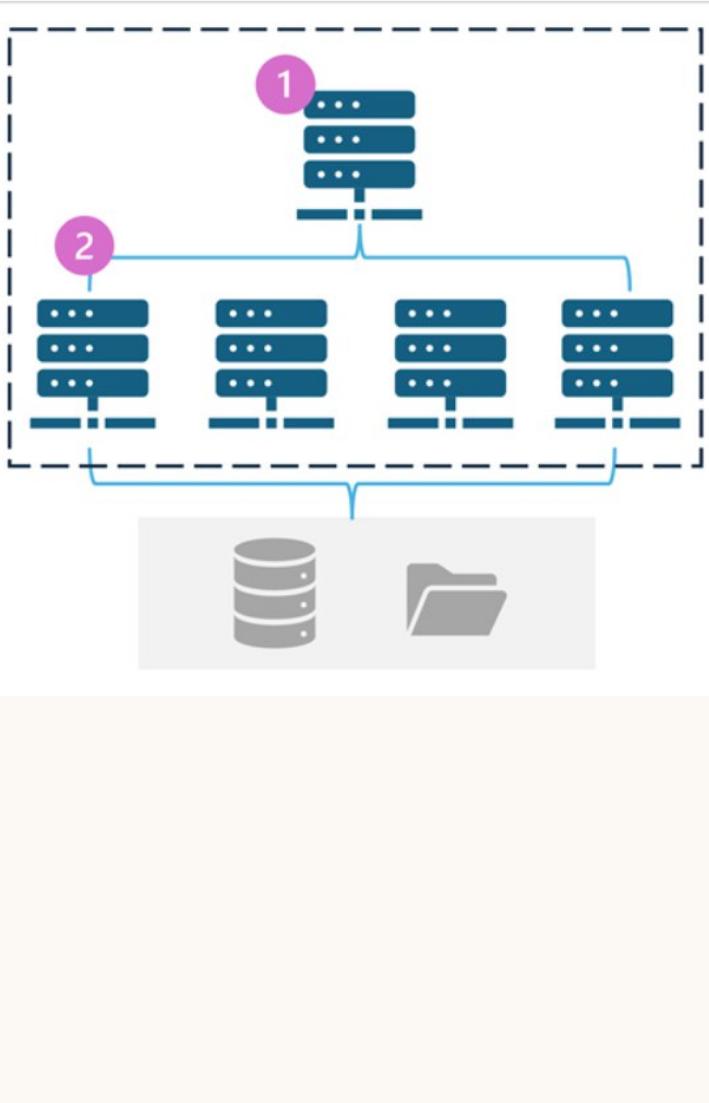
- Spark Pools consists of Compute nodes that distribute data processing tasks. It contains 2 kinds of node:
 - (1) head node → Coordinates distributed process through driver program
 - (2) multiple worker nodes → on which actual executors perform the actual data processing tasks.

Types of Spark Pools

Microsoft fabric provider:

- (1) Starter pool in each workspace enabling spark jobs to be started & run quickly with minimal setup and configuration (Prehydrated Compute resource)
- (2) Custom pool by your own customization of specific node configurations (provided on demand)

Note The ability to customize Spark pool settings can be disabled by Fabric administrator at The fabric Capacity level



You can manage settings for the starter pool and create new Spark pools in the **Admin portal** section of the workspace settings, under **Capacity settings**, then **Data Engineering/Science Settings**.

Microsoft Fabric

Admin portal

Tenant settings New

Usage metrics

Users

Premium Per User

Audit logs

Domains New

Workloads

Tags New

Capacity settings

- Refresh summary
- Embed Codes
- Organizational visuals
- Azure connections
- Workspaces
- Custom branding
- Fabric identities
- Featured content
- Help + support

Create new pool

Spark pool name *

StarterPool

Node family

Memory optimized

Node size

Small

Autoscale

If enabled, your Apache Spark pool will automatically scale up and down based on the amount of activity.

Enable autoscale

1 5

Dynamically allocate executors

Enable dynamic allocation

1 4

Create **Cancel**

Open Spark settings

- Each workspace is assigned a spark cluster
- workspace admin can manage settings for the spark cluster in the workspace settings

- Configuration settings of spark pools:

Node Family: Type of virtual machines used for spark cluster nodes (e.g. memory optimized nodes)
↳ dependent on subscription)

Autoscale: provide nodes as needed (scale up/down) automatically. Initial & max nodes allocated to the pool can be set

Dynamic allocation: allocate executor process on the worker nodes dynamic based on the data volumes

Native execution engine: in microsoft fabric is a vectorized processing engine that run spark operations directly on lakehouse infrastructure

Runtime Version

Spark properties

It improve performance of queries used on large datasets in parquet or delta file formats

To enable the native execution engine for a specific script or notebook, you can set these configuration properties at the beginning of your code, like this:

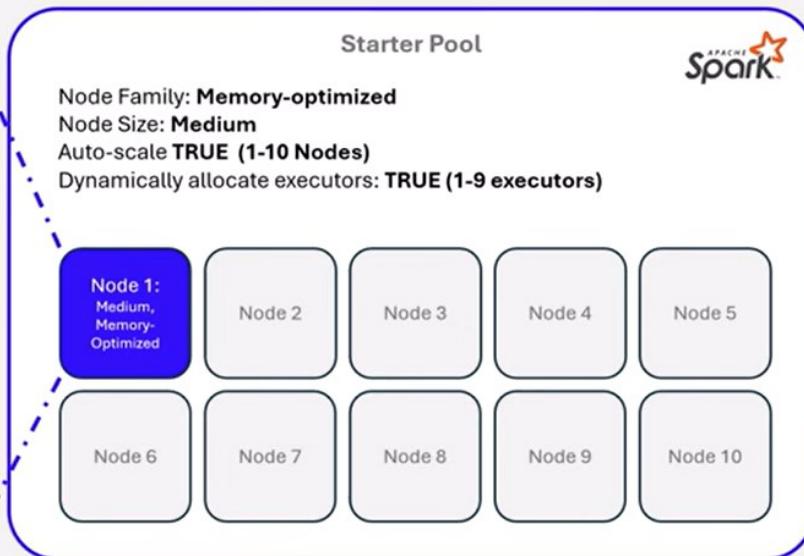
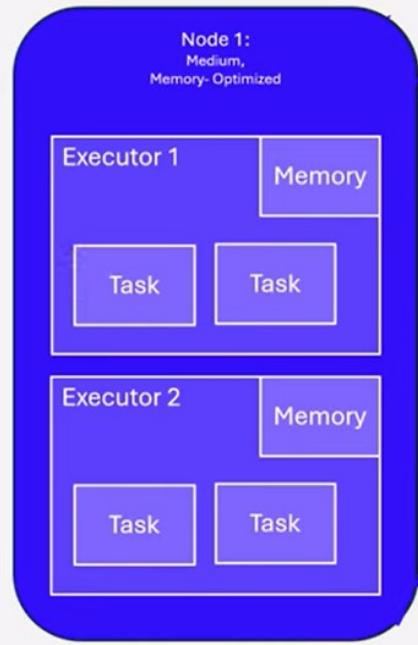
```
JSON Copy
%%configure
{
  "conf": {
    "spark.native.enabled": "true",
    "spark.shuffle.manager": "org.apache.spark.shuffle.sort.ColumnarShuffleManager"
  }
}
```

High Concurrency mode: share the spark session across multiple concurrent users or process

(1) Isolation between notebooks

(2) Same session to increase efficiency

A Node is a commodity server. Executors are JVM processes that performs Tasks.



So instead of a Starter Pool, you want to create a Custom Spark Pool

Beware:

Starter Pool

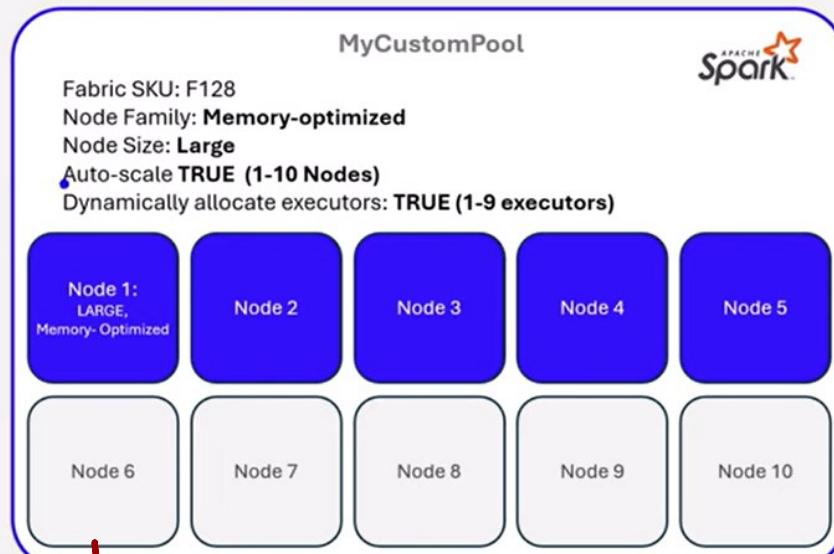
Spark Session startup time:
~10 seconds

Custom Spark Pool –

Spark Session startup times:
~2-3 minutes

why we create it?

- Consume less Capacity -



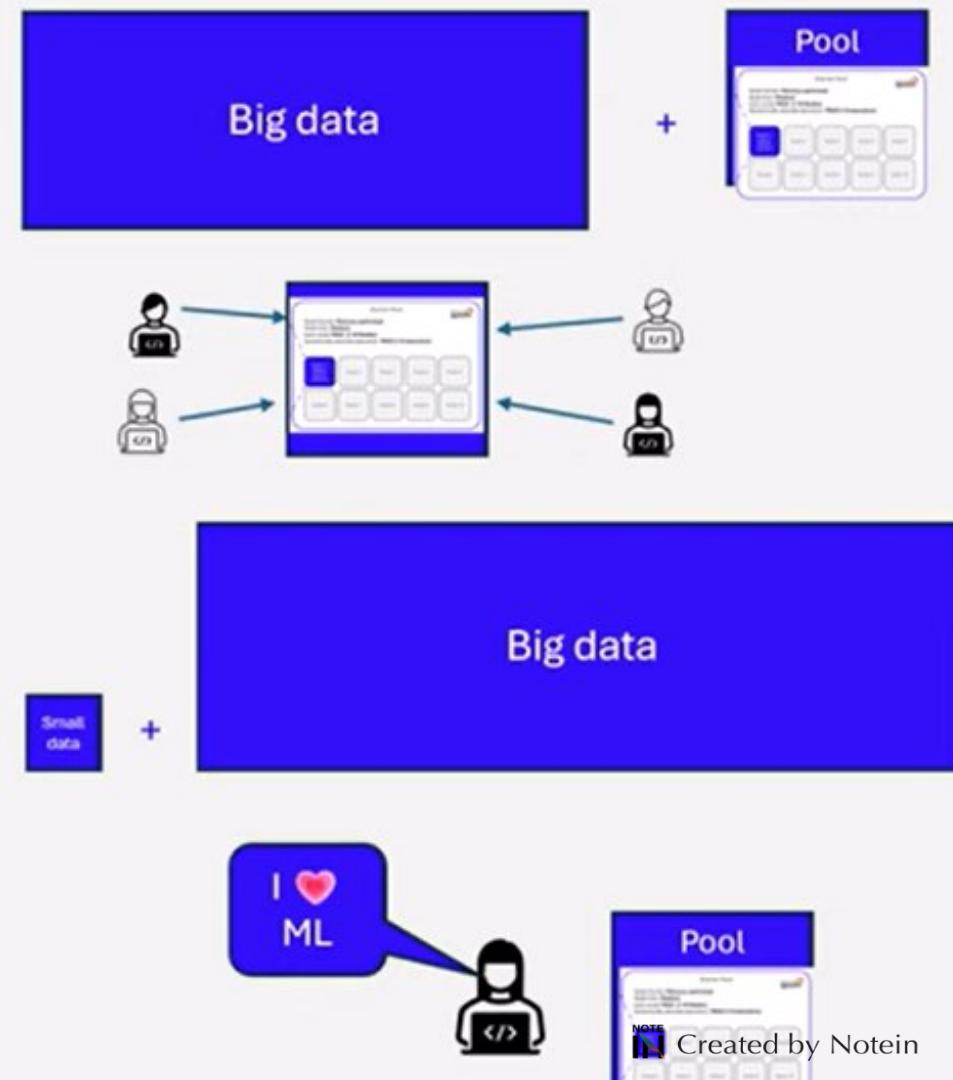
SKU name	Capacity units	Spark vCores	Node size	Default max nodes	Max number of nodes
F2	2	4	Medium	1	1
F4	4	8	Medium	1	1
F8	8	16	Medium	2	2
F16	16	32	Medium	3	4
F32	32	64	Medium	8	8
F64	64	128	Medium	10	16
(Trial Capacity)	64	128	Medium	10	16
F128	128	256	Medium	10	32
F256	256	512	Medium	10	64
F512	512	1024	Medium	10	128
F1024	1024	2048	Medium	10	200
F2048	2048	4096	Medium	10	200

- Data requires more Compute
- Dividing processing to different Pools so, They won't eat the same resource when they run



The limits of the Starter Pool, and when you might need to create a Custom Pool

- **Data size:** if you are processing very large datasets, that the Starter Pool is struggling to process, then a custom Pool with increased node sizing can help.
- **Concurrency:** Spark is using a limited set of resources, based on your Fabric Capacity, and if you have multiple users all spinning up their own Spark Sessions, this will eat up your resources. Therefore, when you have multiple users all trying to use Spark (or multiple jobs all scheduled to use the capacity at the same time), you might need a custom Spark configuration to accommodate this.
- **Variance in job resource requirements:** When you have a high variance in the resources required for certain jobs that you are running, it can be more efficient to create custom configurations for each job. For example, some very large jobs, and some very small jobs.
- **Job Type:** our focus currently is on Data Engineering, but Data Scientists/ AI Engineers might also be using your capacity. Training machine learning models will nearly always require a customized Spark configuration.



DownSide

Don't support CI/CD and is tied to The workspace

So where can I create and manage custom Spark Pools?

Option 1: In the Workspace Settings

Spark settings
Configure and manage settings for Spark workloads and the default environment for the workspace.

Pool Environment Jobs High concurrency Automatic log
Pool

Default pool for workspace
Use the automatically created starter pool or create custom pools for workspaces and items in the capacity. If the setting Customize compute configurations for items is turned off, this pool will be used for all environments in this workspace.
StarterPool

Pool details
Node family: Memory optimized Node size: Medium Number of nodes: 1 - 10

Create new pool

Spark pool name *
 Name should not be empty.

Node family
Memory optimized

Node size
Small

Autoscale
If enabled, your Apache Spark pool will automatically scale up and down based on the amount of activity.
 Enable autoscale
1 10

Dynamically allocate executors
 Enable dynamic allocation
1 9

Create **Cancel**

All notebooks and Spark Jobs in the Workspace will use these Settings (by default)

- ✗ No git integration
- ✗ No pool re-use across workspace

Customize compute configurations for items

When turned on, users can adjust compute configuration for individual items such as notebooks and Spark job definitions. Learn more about Customize compute configurations for items

On

So where can I create and manage custom Spark Pools?

Option 2: a Default Environment in your Workspace Settings

Spark settings
Configure and manage settings for Spark workloads and the default environment for the workspace.

Pool Environment Jobs High concurrency Automatic log

Set default environment
The default environment will provide Spark properties, libraries, and developer settings for notebooks and Spark job definitions in this workspace when users don't select a different environment. Learn more about Set default environment

Workspace default ▾
Available environments
No environments available
New environment

Environment pool
Default pool

Pool details
Node family Auto (Memory optimized) Node size Medium Number of nodes 1 - 10

Compute
Spark driver core 8
Spark driver memory 56GB
Spark executor core 8
Spark executor memory 56GB
Dynamically allocate executors Enable dynamic allocation
Spark executor instances 1 - 9

All notebooks and Spark Jobs in the Workspace will use these Settings (by default)

✓ Allows Git integration (through Environment)
✗ No pool re-use across workspace • •

- One workspace can have multiple environments (one is default env.)

Spark Job Settings

- ◆ Reserve maximum cores for active Spark jobs
 - Spark estimates the minimum and maximum number of cores it needs to complete a job and then does the reservations based on that
 - Turning on gives more reliability but could be less optimal usage
 - Turning off reduces reliability but uses the compute more optimally
- ◆ Spark session timeout
 - Time after inactive Spark sessions are terminated
 - Defaults to 20 minutes

Pool Environment **Jobs** High concurrency Automatic log

Reserve maximum cores for active Spark jobs Off

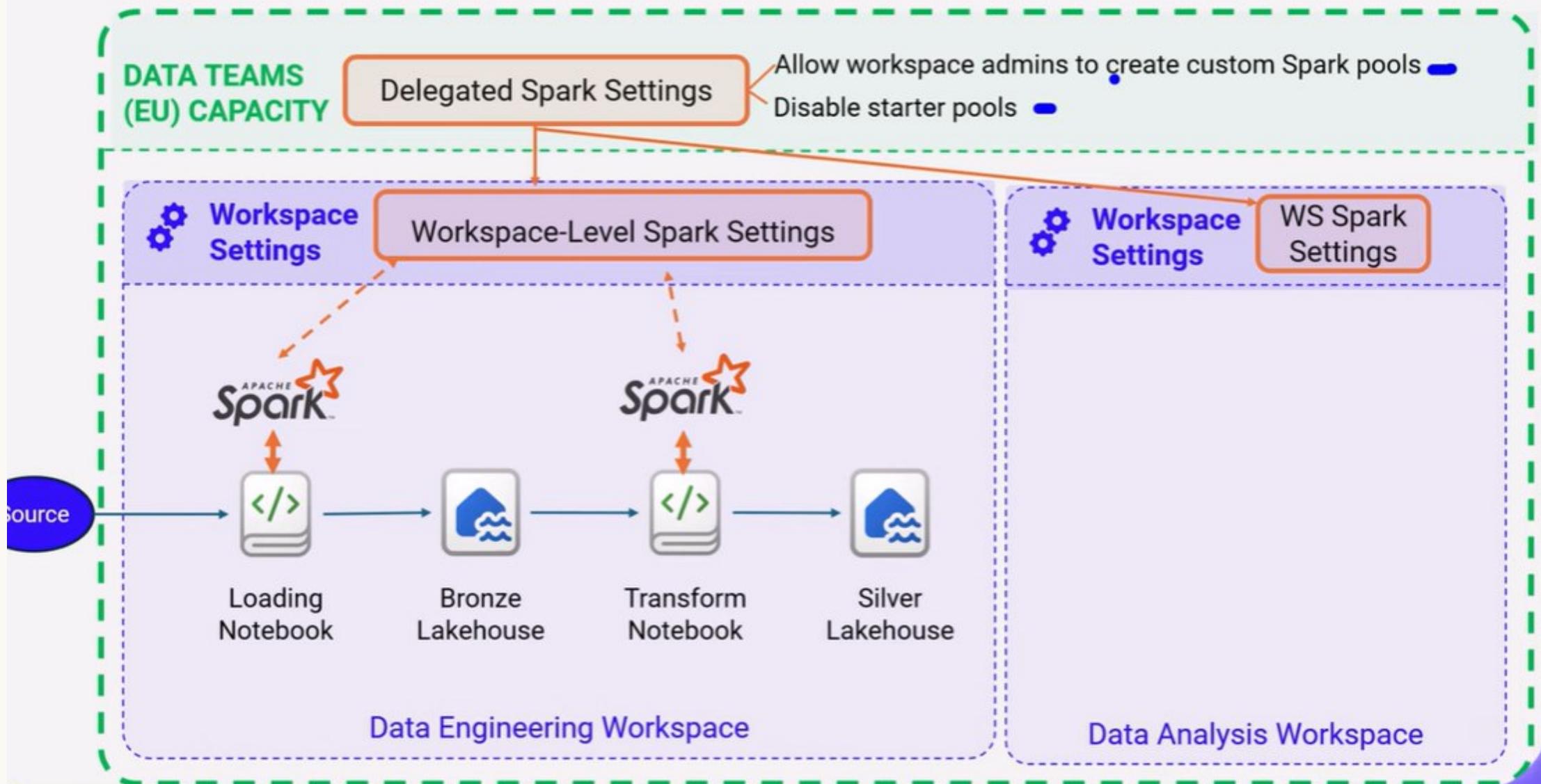
When this setting is on, your Fabric capacity reserves the maximum number of cores needed for active Spark jobs, ensuring job reliability by making sure that cores are available if a job scales up. When this setting is off, jobs are started based on the minimum number of cores needed, letting more jobs run at the same time. [Learn more about reserving maximum cores](#)

Set Spark session timeout
Specify a time to terminate inactive Spark sessions. [Learn more about session expiry](#)

20 minutes

Reset to default time

Configuring Spark Settings – in the Capacity Admin Settings



- Due to massive growth in data, organizations are shifting from traditional IT centric data architectures (where data is governed and managed centrally) —to→ more federated models organized according to business needs.
- This federated models are called Data Mesh
 -
- Is a decentralized data architecture that organizes data by specific business domains (e.g. sales, HR.)
- Data mesh architecture primarily supports organizing data into domains and enabling data consumers to filter and find content by domain
- It also enables federated governance, which means that some governance currently controlled at the tenant level can be **delegated to domain-level control**, enabling customised rules and restrictions for different domains.
- Domain is way of logically together all the data in an organization that is relevant to particular area or field.
- When a workspace is associated with the domain, all the items in the workspace are also associated with the domain, and they receive a domain attribute as part of their metadata.

Note Domain assignment does not affect item visibility or accessibility for tenant users. Item discovery, visibility, and access depend on such things as workplace role and item permissions, but not domain assignment

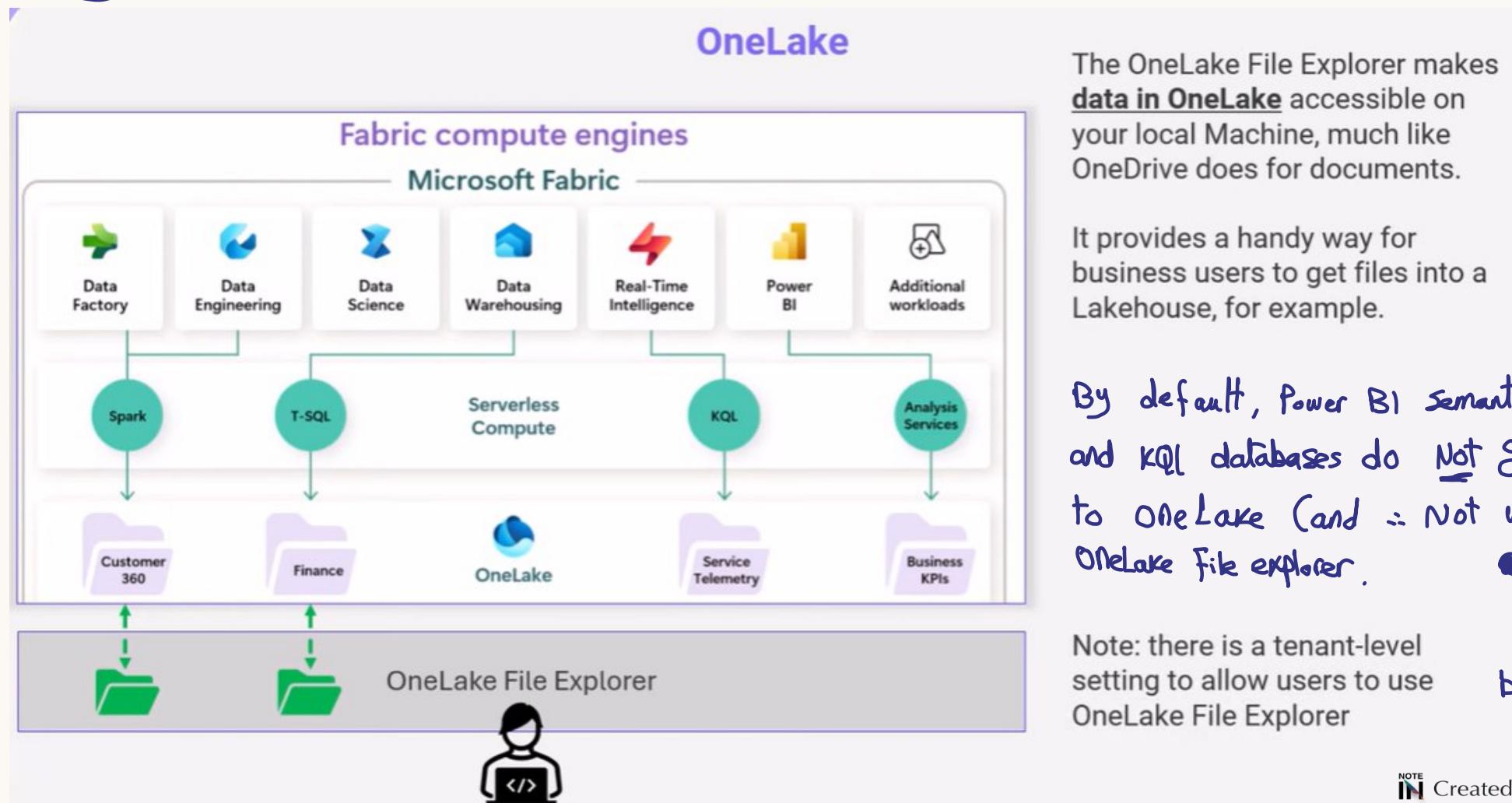
So, what can we currently do with Domains (by Role)?

Task	Fabric Admin	Domain Admin	Domain Contributor (Workspace Admin)
Create & delete domains	✓		
Appoint Domain Admins	✓		
View and edit all domains	✓		
Appoint Domain Contributors	✓	✓	
View and edit their own domains	✓	✓	
Override delegated tenant settings*	✓	✓	
Assign workspaces to domains (bulk, in Domain Settings)	✓	✓	
Assign workspaces to domains (singular, in Workspace Settings)	✓	✓	✓

- * The following tenant-level settings can be delegated to the Domain-level:
- Domain-level default sensitivity label
 - Certification Settings

Default domain

- is a domain that has been defined as the default domain for specified users and/or security group
- The system scans the organization's workspace, when it finds a workspace admin
 - if workspace is assigned to some domain → nothing happens
 - else it's assigned to default domain



The OneLake File Explorer makes **data in OneLake** accessible on your local Machine, much like OneDrive does for documents.

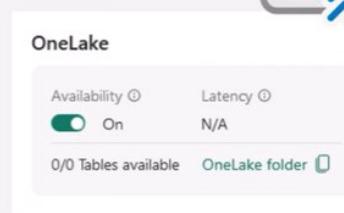
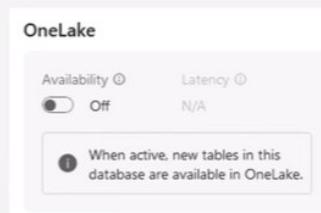
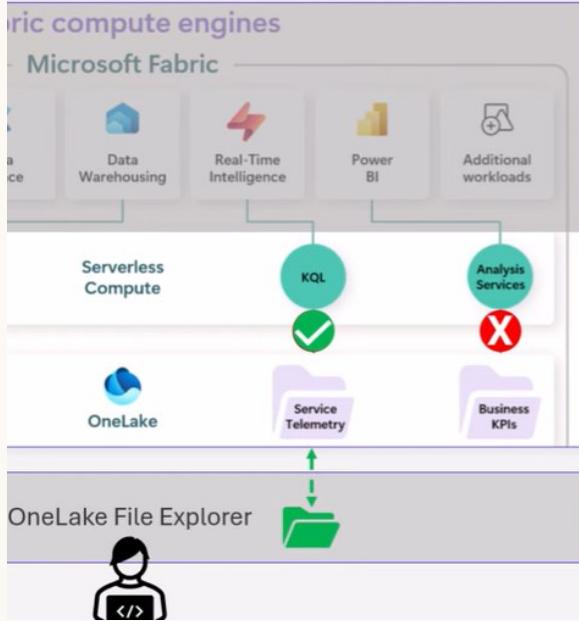
It provides a handy way for business users to get files into a Lakehouse, for example.

By default, Power BI semantic models and KQL databases do Not get written to OneLake (and ∵ Not written to OneLake file explorer).

Note: there is a tenant-level setting to allow users to use OneLake File Explorer



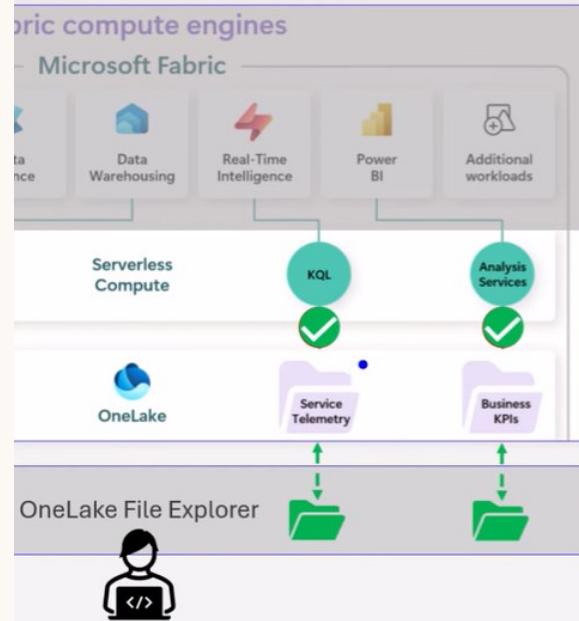
Enabling OneLake integration for KQL Database



Step 1:
Within the UI of your KQL Database, turn on this
'Availability' button.

Note:
When you turn on OneLake integration in a KQL Database, it only works for new tables. It doesn't backfill existing data.

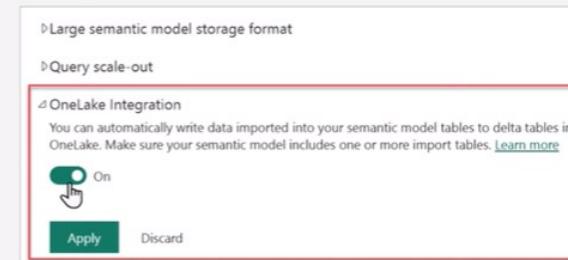
Enabling OneLake integration for Power BI Semantic Models



Note: only applies to Import mode semantic models/ tables

Step 1: Enable this functionality in the Tenant-level Admin Settings

Step 2: within the individual semantic model, enable OneLake Integration



Final note: when the delta tables for Semantic Models, and KQL Databases are written to OneLake like this, it opens up their use in other experiences.

→ Maybe Not any More

Shortcuts & Shortcut Caching

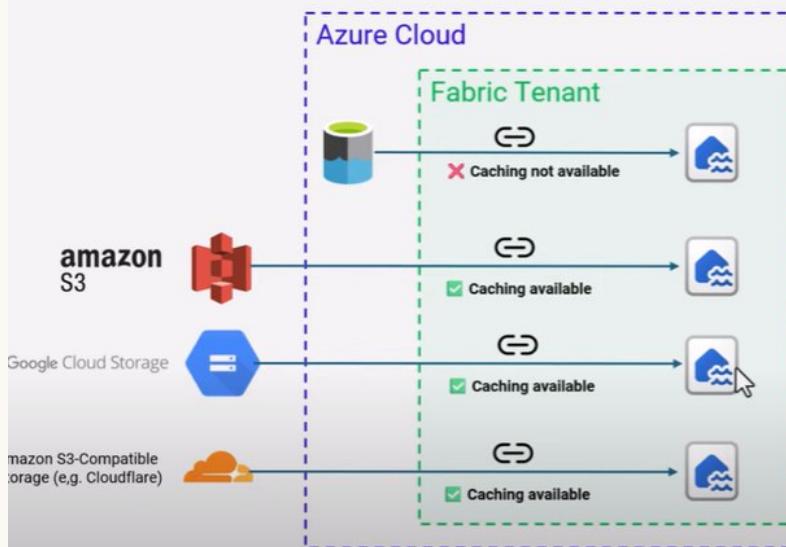
Shortcuts allow you to access external data in other cloud platforms from within Fabric, without the need for ETL.

Shortcut caching can be used to reduce egress costs associated with this cross-cloud data access.

Enable Shortcut Caching for data accessed through this Workspace:

The screenshot shows the 'OneLake Settings' page under 'Workspace settings'. The left sidebar lists various sections: General, License info, Azure connections, System storage, Git integration, OneLake (which is selected and highlighted with a red box), Workspace identity, Network security, Monitoring, Power BI, Delegated Settings, Data Engineering/Science, and Data Factory. The main content area is titled 'OneLake Settings' and describes how to configure and manage settings for OneLake in the workspace. It includes a link to learn more about OneLake and a section for 'OneLake File Explorer' which integrates OneLake with Windows File Explorer. A 'Download OneLake app' button is also present. Below this, there's a 'Shortcut Settings' section with a toggle switch labeled 'Enable cache for shortcuts' (which is off) and a descriptive text explaining that data accessed through shortcuts will be cached in OneLake. A red box highlights this section, and a cursor icon is visible at the bottom right.

Shortcuts & Shortcut Caching



Key caching rules to be aware of:

24 hour rule: If a file hasn't been accessed for more than 24 hours, it's purged from the cache

1GB rule: Individual files greater than 1 GB in size aren't cached

Apache Airflow Jobs (aka Data Workflows)

←

Create new pool

Name
MyAirflowRuntime

Compute node size
Small

Enable autoscale

Extra nodes 0

Create **Cancel**

The Apache Airflow job (previously known as the Data Workflow) is a (preview) Fabric item that allows you to configure and runtime Apache Airflow DAGs (directed acyclic graphs).

These are orchestration jobs. They normally require a server (or hosted platform) to run a DAG, but Fabric provides a mechanism to run your DAG using a Pool.



Notebooks

Parameters

- ◆ One cell can be toggled as a parameter cell
 - Define default values as variables
- ◆ Different types of parameters are supported
 - String, Integer, Float and Boolean
- ◆ Execution engine adds a new cell beneath the parameter cell with input parameters to overwrite the default values

The screenshot shows a Jupyter Notebook interface with two code cells and a context menu.

Cell 1:

```
1 param1 = "default value for param1"
2 param2 = 1
3 param3 = True
```

Cell 2:

```
1 param1 = "default value for param1"
2 param2 = 1
3 param3 = True
```

Context Menu (over Cell 1):

- M4
- New comment thread
- Move cell up
- Move cell down
- Hide input
- Hide output
- @] Toggle parameter cell**

Execution Results:

[1]: ✓ - Command executed in 312 ms on 2:49:09 PM, 3/13/25

[2]: ✓ - Command executed in 292 ms on 2:49:09 PM, 3/13/25

Input parameter

Content ↑	Type	Value
param1	string	new value
param2	int	2
param3	bool	False

NotebookUtils (MsSparkUtils)

- ◆ Notebookutils
 - An integrated package in Microsoft Fabric Notebooks designed to streamline common tasks such as file system operations, environment variable access, notebook chaining, and secret management
- ◆ File System utilities (notebookutils.fs)
 - Interact with file systems like ADLS and OneLake
 - Create directories, move and delete files etc.
- ◆ Notebook utilities (notebookutils.notebook)
 - Run notebooks, set exit values, create notebooks etc.
- ◆ Credential utilities (notebookutils.credentials)
 - Get access tokens and manage secrets in an Azure Key Vault
- ◆ Other utilities
 - Lakehouse, Session and Runtime

utility for dealing
with lakehouse

utilities for getting some information and
managing the current session

```
1 notebookutils.fs.mkdirs("Files/new_folder")
2 notebookutils.fs.ls("Files/new_folder")
3 notebookutils.fs.rm('Files/new_folder', True)
```

```
1 notebookutils.notebook.run("nb_example_1", 90)
2 notebookutils.notebook.exit("value string")
```

```
1 notebookutils.credentials.getSecret('https://<name>.vault.azure.net/', \
2 'secret name')
```

```
1 notebookutils.lakehouse.list()
2 notebookutils.runtime.context
3 notebookutils.session.stop()
```

- Cell magic Commands are on Cell level and changes what the entire Cell does
- Change the Cell default language to another

line commands

- do operations on a single line of a cell
- There can not be anything else in the cell than those commands.

◆ Cell magic commands

- The most relevant commands: `%sql`, `%pyspark`, `%spark`, `%sparkr`, `%bash`

◆ Line magic commands

- The most relevant commands: `%pip`, `%run`

```
1 %%sql
2 -- This cell is using Spark SQL
3 SELECT 'ABC' AS col1
```

```
1 %%pyspark
2 # This cell is using PySpark
3 python_var == "Python Variable!"
```

```
1 %%spark
```

```
1 %pip install pandas
✓ 12 sec - Command executed in 12 sec 852 ms by Aleksi Partanen on 12:54:31 PM, 3/14/25
```

```
Requirement already satisfied: pandas in /home/trusted-service-user/cluster-env/trident_env/lib/python3.11/site-packages (2.1.4)
Requirement already satisfied: numpy<2,>=1.23.2 in /home/trusted-service-user/cluster-env/trident_env/lib/python3.11/site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/trusted-service-user/cluster-env/trident_env/lib/python3.11/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2028.1 in /home/trusted-service-user/cluster-env/trident_env/lib/python3.11/site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in /home/trusted-service-user/cluster-env/trident_env/lib/python3.11/site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in /home/trusted-service-user/cluster-env/trident_env/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: python -m pip install --upgrade pip
```

```
2 // This cell is using Spark Scala  
3 var scala_var = "Scala Variable"
```

Note: you may need to restart the kernel to use updated packages.
Warning: PySpark kernel has been restarted to use updated packages.

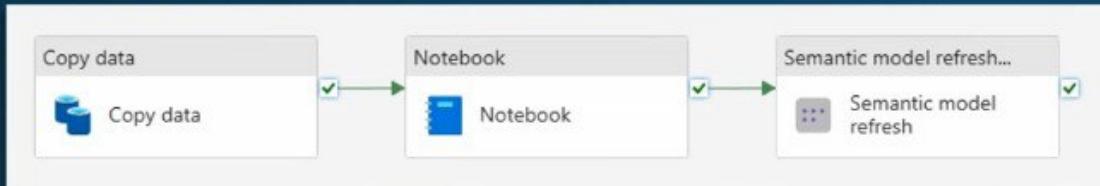
```
1 %%sparkr  
2 # This cell is using Spark R  
3 r_var <- "R Variable"
```

```
1 %%bash  
2 # This cell is using Bash  
3 bash_var="Bash Variable"
```

```
1 %run nb_example_1
```

Running Notebooks

- ◆ Run manually
 - Run cell by cell or run the whole notebook
- ◆ Run notebooks from notebook
 - Use %run magic command
 - NotebookUtils library
 - notebookutils.notebook.run
 - notebookutils.notebook.runMultiple
- ◆ Run in a data pipeline



```
1 %run nb_example_1  
  
1 notebookutils.notebook.run("nb_example_1", 90)  
  
1 notebookutils.notebook.runMultiple(["nb_example_1", "nb_example_2"])  
  
1 DAG = {  
2     "activities": [  
3         {  
4             "name": "nb_example_1",  
5             "path": "nb_example_1",  
6             "timeoutPerCellInSeconds": 90,  
7             "args": {"param_1": "some value"},  
8             "retry": 1,  
9             "retryIntervalInSeconds": 10  
10        },  
11        {  
12            "name": "nb_example_2",  
13            "path": "nb_example_2",  
14            "args": {"param_1": "some value"}  
15        },  
16        {  
17            "name": "nb_example_3",  
18            "path": "nb_example_3",  
19            "timeoutPerCellInSeconds": 90,  
20            "args": {"param_1": "some value"},  
21            "retry": 1,  
22            "retryIntervalInSeconds": 10,  
23            "dependencies": ["nb_example_1", "nb_example_2"]  
24        },  
25        {  
26            "timeoutInSeconds": 300,  
27            "concurrency": 3  
28        }  
29    exit_values_dag = notebookutils.notebook.runMultiple(DAG, {"displayDAGViaGraphviz": True})
```

↓
run → run 1 notebook

runMultiple → run multiple notebooks in any order.
If we use dag with it, we can notebooks
in specific order we need

