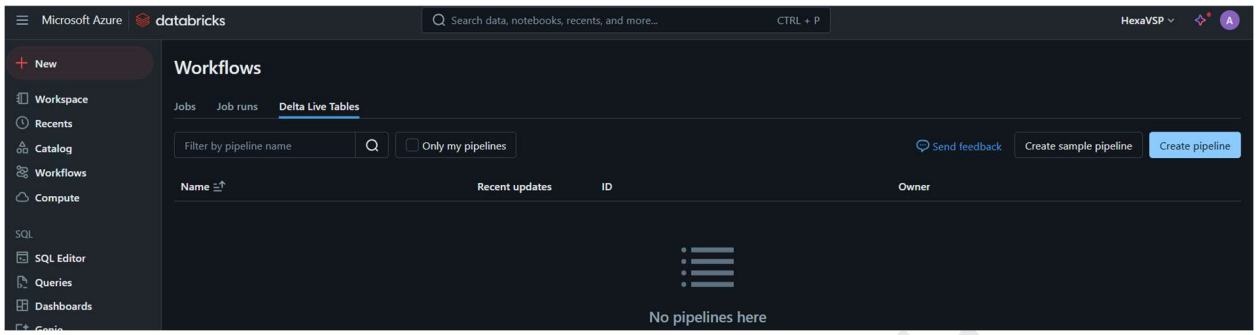
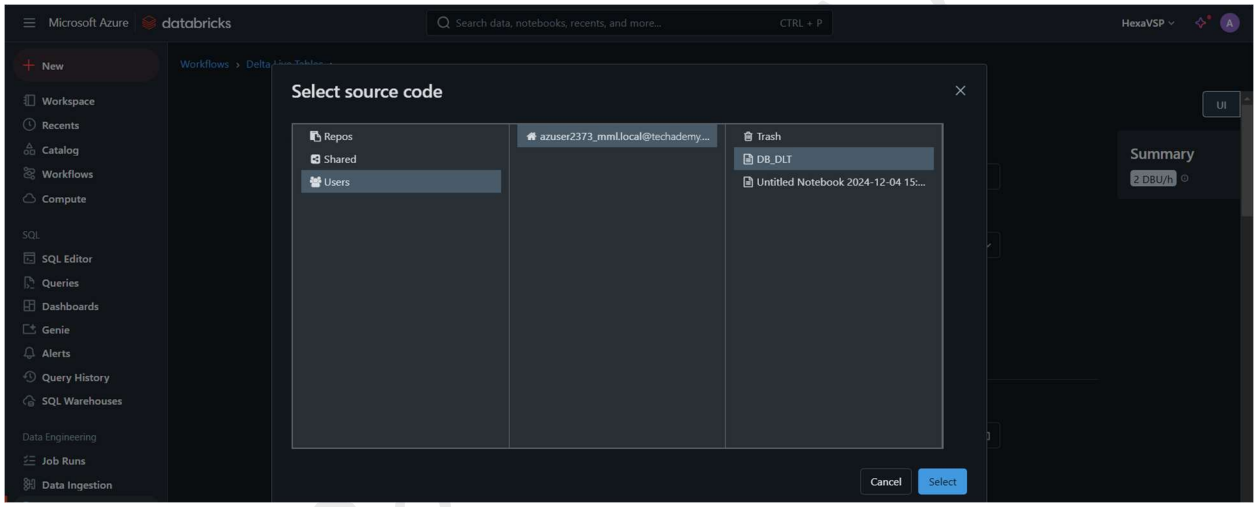
Create pipeline for DLT

Use Databricks premium resource to access this service.

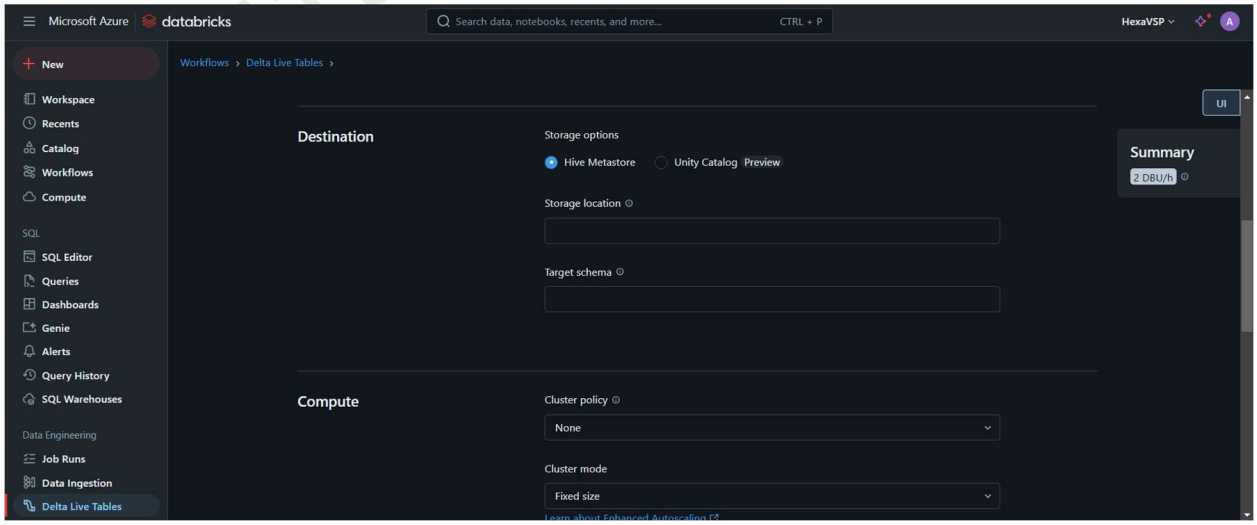
Under Workflow, click the Delta live tables



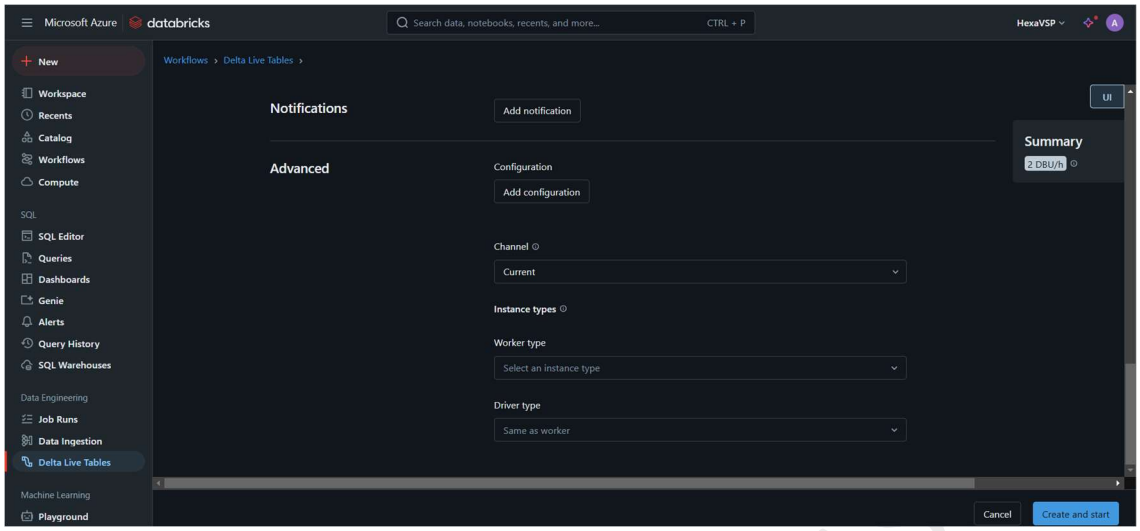
Now provide name, and other necessary details. Choose the source code file

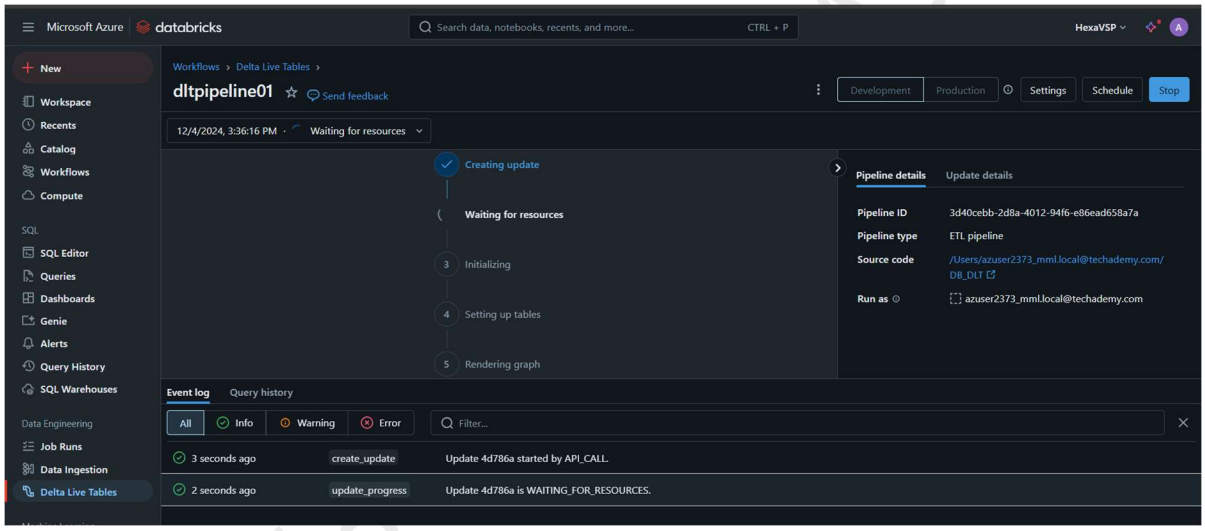


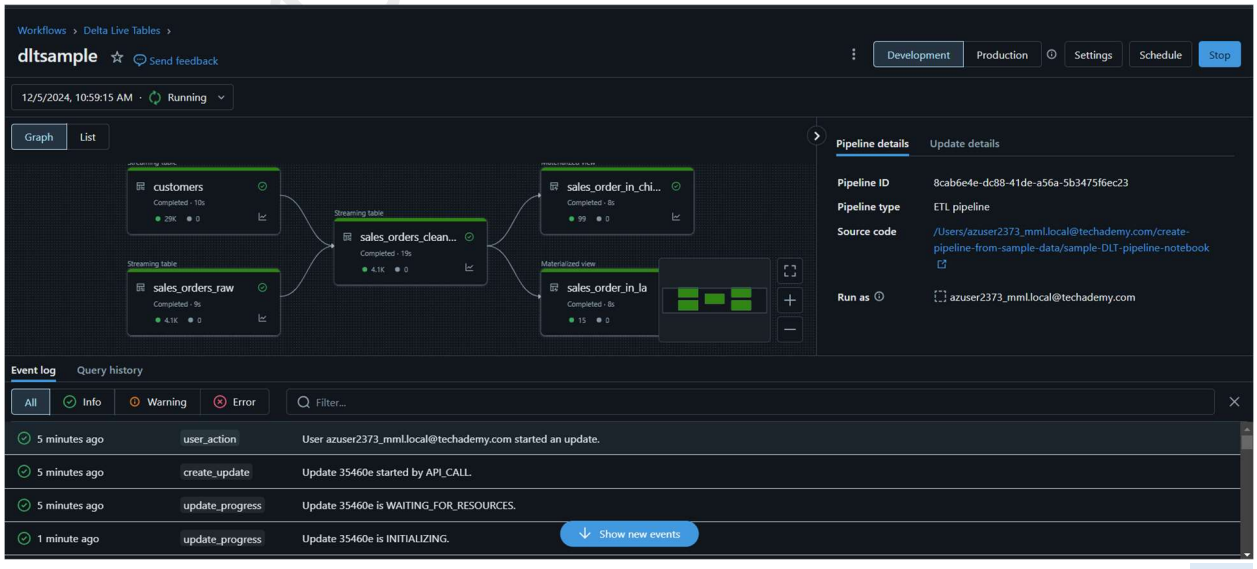
Choose the worker, worker type and all other necessary info like min worker, max worker.

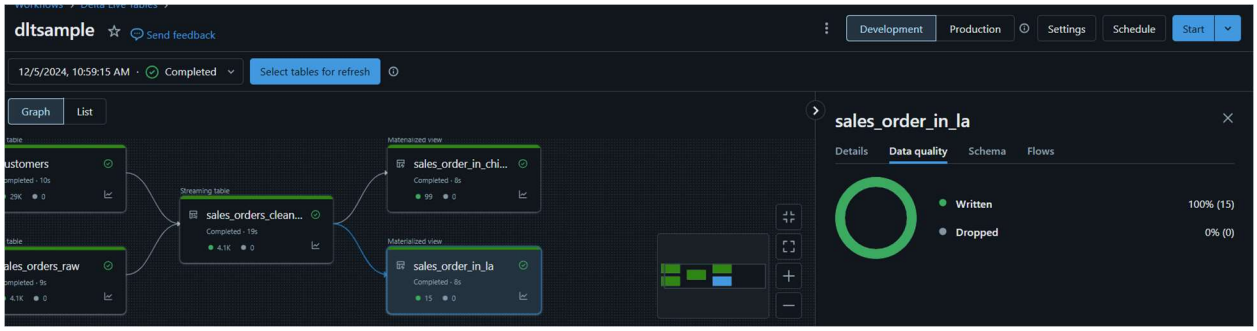


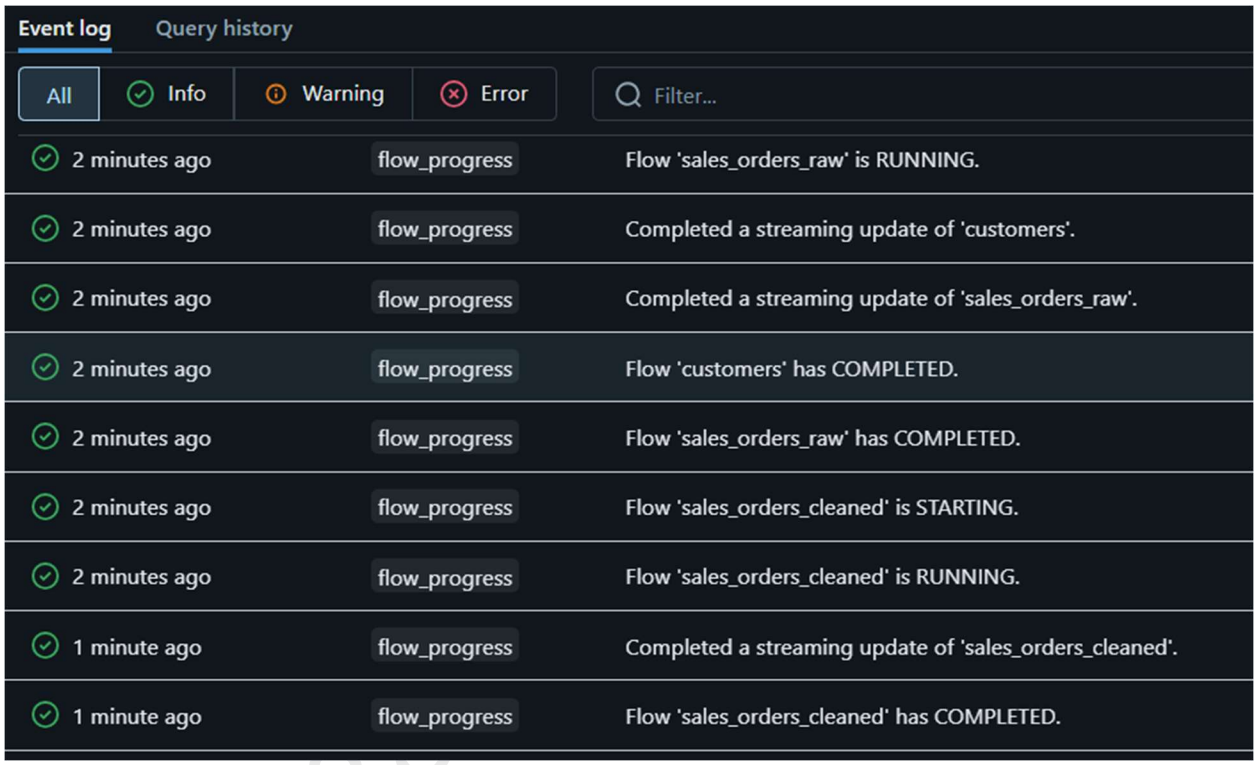
We can add the notification.

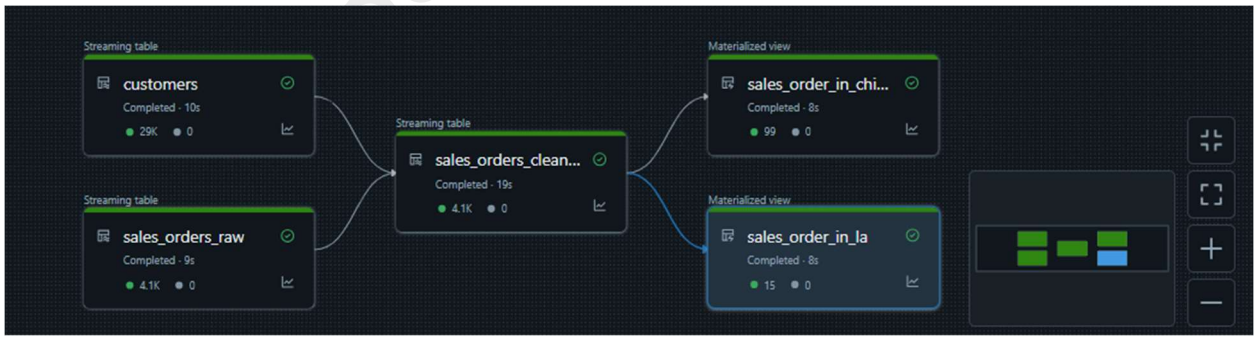




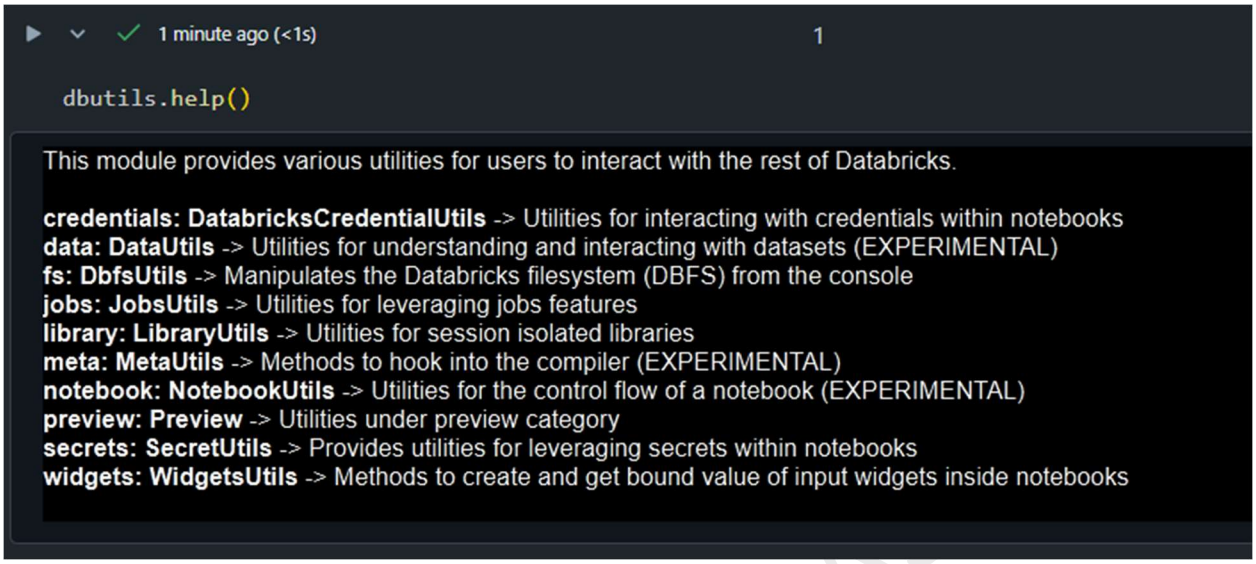



dltsample
Send feedback
Development
Production
Settings
Schedule
Start
12/5/2024, 10:59:15 AM · Completed
Select tables for refresh
Graph
List
customers
Completed · 10s
29K 0
sales_orders_raw
Completed · 9s
4.1K 0
Streaming table
sales_orders_clean...
Completed · 19s
4.1K 0
Materialized view
sales_order_in_chi...
Completed · 8s
99 0
Materialized view
sales_order_in_la
Completed · 8s
15 0
sales_order_in_la
Details
Data quality
Schema
Flows
Written
100% (15)
Dropped
0% (0)


Event log
Query history
All
Info
Warning
Error
Filter...
2 minutes ago flow_progress Flow 'sales_orders_raw' is RUNNING.
2 minutes ago flow_progress Completed a streaming update of 'customers'.
2 minutes ago flow_progress Completed a streaming update of 'sales_orders_raw'.
2 minutes ago flow_progress Flow 'customers' has COMPLETED.
2 minutes ago flow_progress Flow 'sales_orders_raw' has COMPLETED.
2 minutes ago flow_progress Flow 'sales_orders_cleaned' is STARTING.
2 minutes ago flow_progress Flow 'sales_orders_cleaned' is RUNNING.
1 minute ago flow_progress Completed a streaming update of 'sales_orders_cleaned'.
1 minute ago flow_progress Flow 'sales_orders_cleaned' has COMPLETED.


Streaming table
customers
Completed · 10s
29K 0
Streaming table
sales_orders_raw
Completed · 9s
4.1K 0
Streaming table
sales_orders_clean...
Completed · 19s
4.1K 0
Materialized view
sales_order_in_chi...
Completed · 8s
99 0
Materialized view
sales_order_in_la
Completed · 8s
15 0

Dbutils

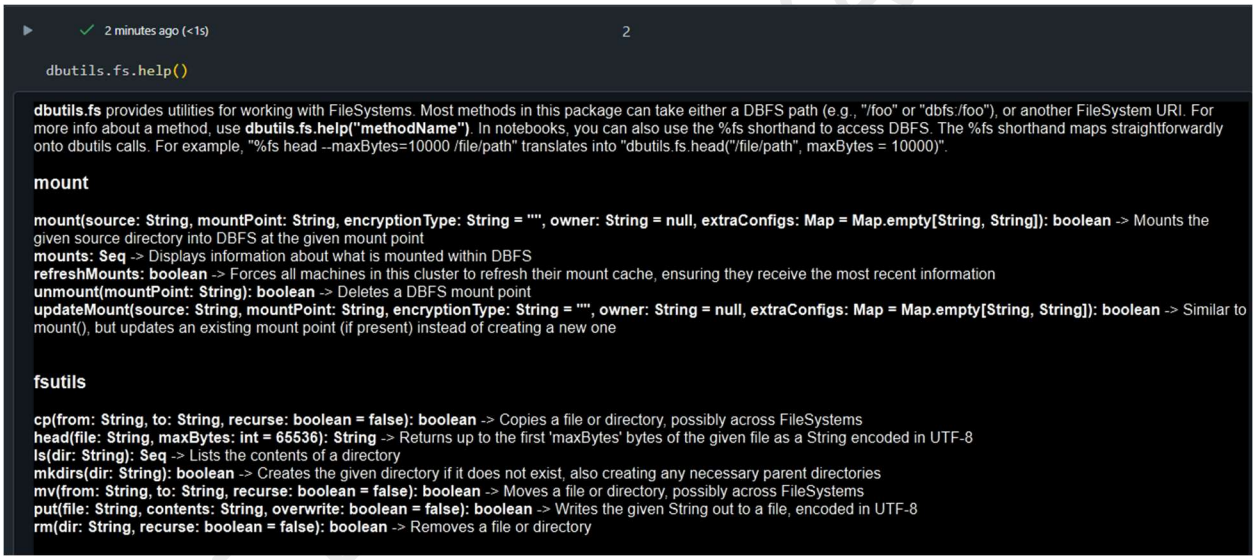


1 minute ago (<1s) 1

```
dbutils.help()
```

This module provides various utilities for users to interact with the rest of Databricks.

credentials: DatabricksCredentialUtils -> Utilities for interacting with credentials within notebooks
data: DataUtils -> Utilities for understanding and interacting with datasets (EXPERIMENTAL)
fs: DbfsUtils -> Manipulates the Databricks filesystem (DBFS) from the console
jobs: JobsUtils -> Utilities for leveraging jobs features
library: LibraryUtils -> Utilities for session isolated libraries
meta: MetaUtils -> Methods to hook into the compiler (EXPERIMENTAL)
notebook: NotebookUtils -> Utilities for the control flow of a notebook (EXPERIMENTAL)
preview: Preview -> Utilities under preview category
secrets: SecretUtils -> Provides utilities for leveraging secrets within notebooks
widgets: WidgetsUtils -> Methods to create and get bound value of input widgets inside notebooks



2 minutes ago (<1s) 2

```
dbutils.fs.help()
```

dbutils.fs provides utilities for working with FileSystems. Most methods in this package can take either a DBFS path (e.g., "/foo" or "dbfs:/foo"), or another FileSystem URI. For more info about a method, use **dbutils.fs.help("methodName")**. In notebooks, you can also use the %fs shorthand to access DBFS. The %fs shorthand maps straightforwardly onto dbutils calls. For example, "%fs head --maxBytes=10000 /file/path" translates into "dbutils.fs.head("/file/path", maxBytes = 10000)".

mount

mount(source: String, mountPoint: String, encryptionType: String = "", owner: String = null, extraConfigs: Map = Map.empty[String, String]): boolean -> Mounts the given source directory into DBFS at the given mount point
mounts: Seq -> Displays information about what is mounted within DBFS
refreshMounts: boolean -> Forces all machines in this cluster to refresh their mount cache, ensuring they receive the most recent information
unmount(mountPoint: String): boolean -> Deletes a DBFS mount point
updateMount(source: String, mountPoint: String, encryptionType: String = "", owner: String = null, extraConfigs: Map = Map.empty[String, String]): boolean -> Similar to mount(), but updates an existing mount point (if present) instead of creating a new one

fsutils

cp(from: String, to: String, recurse: boolean = false): boolean -> Copies a file or directory, possibly across FileSystems
head(file: String, maxBytes: int = 65536): String -> Returns up to the first 'maxBytes' bytes of the given file as a String encoded in UTF-8
ls(dir: String): Seq -> Lists the contents of a directory
mkdirs(dir: String): boolean -> Creates the given directory if it does not exist, also creating any necessary parent directories
mv(from: String, to: String, recurse: boolean = false): boolean -> Moves a file or directory, possibly across FileSystems
put(file: String, contents: String, overwrite: boolean = false): boolean -> Writes the given String out to a file, encoded in UTF-8
rm(dir: String, recurse: boolean = false): boolean -> Removes a file or directory

2 minutes ago (4s) 4

```
%fs
ls /databricks-datasets
```

Table

	path	name	size	modificationTime
9	dbfs:/databricks-datasets/atlas_higgs/	atlas_higgs/	0	1733289606806
10	dbfs:/databricks-datasets/bikeSharing/	bikeSharing/	0	1733289606806
11	dbfs:/databricks-datasets/cctvVideos/	cctvVideos/	0	1733289606806
12	dbfs:/databricks-datasets/credit-card-fraud/	credit-card-fraud/	0	1733289606806
13	dbfs:/databricks-datasets/cs100/	cs100/	0	1733289606806
14	dbfs:/databricks-datasets/cs110x/	cs110x/	0	1733289606806

2 minutes ago (<1s) 5

```
dbutils.fs.ls("dbfs:/databricks-datasets/COVID/")
```

```
[FileInfo(path='dbfs:/databricks-datasets/COVID/.DS_Store', name='.DS_Store', size=6148, modificationTime=1615901208000),
 FileInfo(path='dbfs:/databricks-datasets/COVID/CORD-19/', name='CORD-19/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/CSSEGISandData/', name='CSSEGISandData/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/ESRI_hospital_beds/', name='ESRI_hospital_beds/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/IHME/', name='IHME/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/USAFacts/', name='USAFacts/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/coronavirusdataset/', name='coronavirusdataset/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/covid-19-data/', name='covid-19-data/', size=0, modificationTime=1733289613327),
 FileInfo(path='dbfs:/databricks-datasets/COVID/download_daily_covid-19_datasets.sh', name='download_daily_covid-19_datasets.sh', size=1083, modifi
cationTime=1609189123000),
 FileInfo(path='dbfs:/databricks-datasets/COVID/estimated_patient_impact_and_hospital_capacity_by_state/', name='estimated_patient_impact_and_hospi
tal_capacity_by_state/', size=0, modificationTime=1733289613327)]
```

2 minutes ago (<1s) 6

```
dbutils.fs.ls("databricks-datasets")
```

```
 FileInfo(path='dbfs:/databricks-datasets/online_retail/', name='online_retail/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/overlap-join/', name='overlap-join/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/power-plant/', name='power-plant/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/retail-org/', name='retail-org/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/rwe/', name='rwe/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/sai-summit-2019-sf/', name='sai-summit-2019-sf/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/sample_logs/', name='sample_logs/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/samples/', name='samples/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/sfo_customer_survey/', name='sfo_customer_survey/', size=0, modificationTime=1733289615620),
 FileInfo(path='dbfs:/databricks-datasets/sms_spam_collection/', name='sms_spam_collection/', size=0, modificationTime=1733289615620),
```

2 minutes ago (<1s) 8

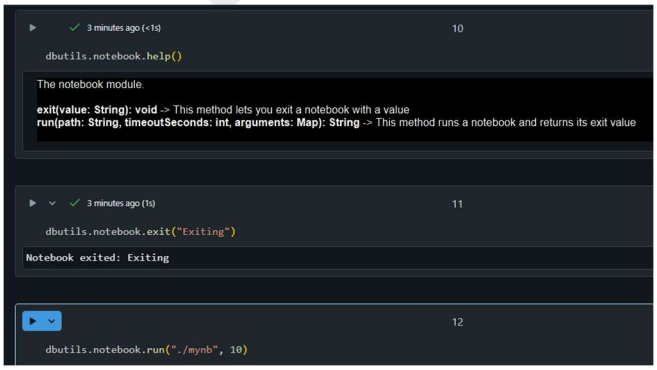
```
for foldername in dbutils.fs.ls("databricks-datasets/"):
    print(foldername.name)
```

```
nyctaxi-with-zipcodes/
online_retail/
overlap-join/
power-plant/
retail-org/
rwe/
sai-summit-2019-sf/
sample_logs/
samples/
sfo_customer_survey/
sms_spam_collection/
songs/
structured-streaming/
timeseries/
tpch/
travel_recommendations_realtime/
warmup/
weather/
wiki/
wikipedia-datasets/
wine-quality/
```

2 minutes ago (<1s) 9 Python

```
for foldername in dbutils.fs.ls("databricks-datasets/"):
    print(foldername.modificationTime)
```

```
1733289624100
1733289624100
1733289624100
1733289624100
1733289624100
```



3 minutes ago (<1s) 10

```
dbutils.notebook.help()
```

The notebook module.

exit(value: String): void -> This method lets you exit a notebook with a value
run(path: String, timeoutSeconds: int, arguments: Map): String -> This method runs a notebook and returns its exit value

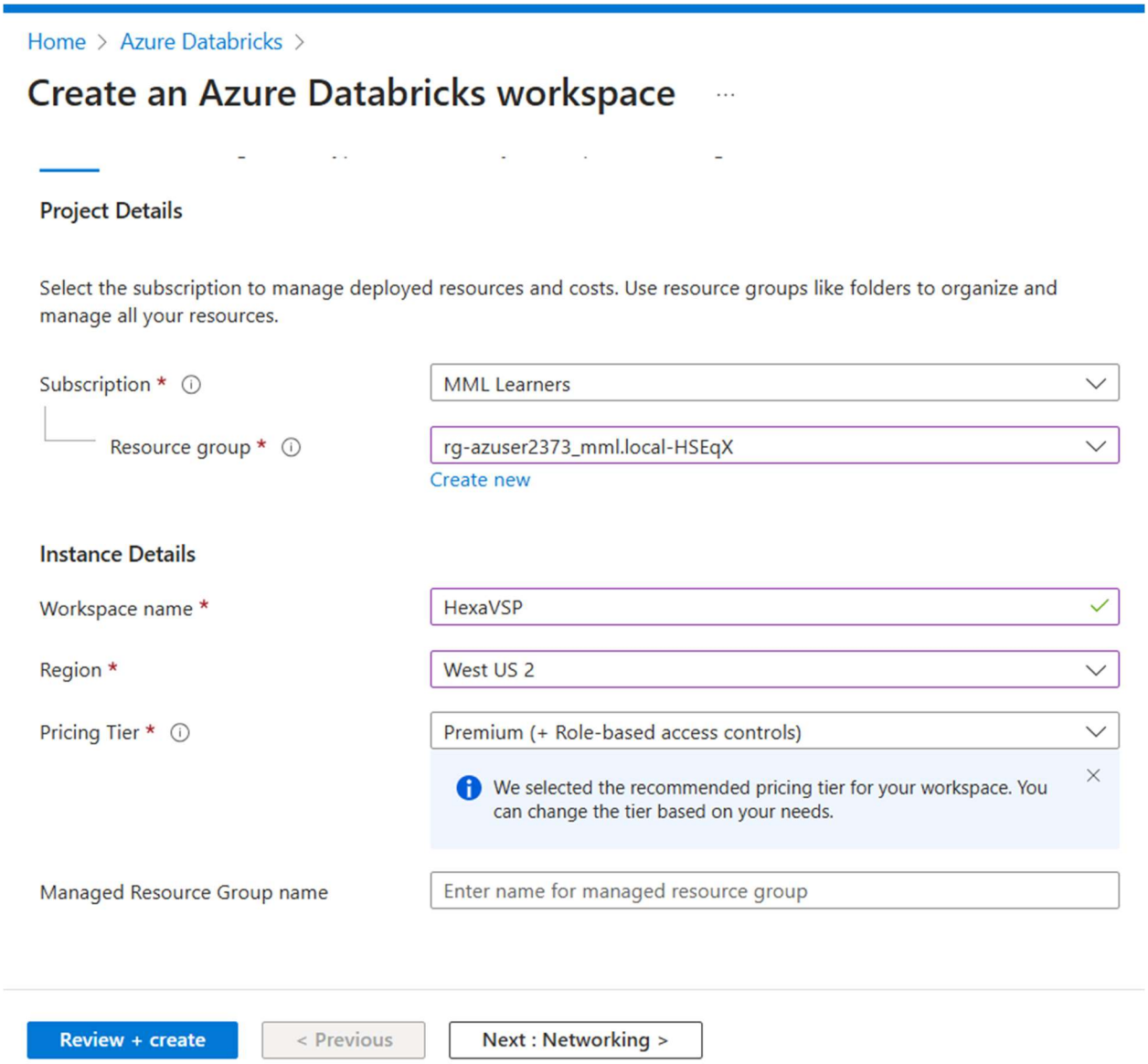
3 minutes ago (1s) 11

```
dbutils.notebook.exit("Exiting")
```

```
Notebook exited: Exiting
```

12

```
dbutils.notebook.run("./mynb", 10)
```


Home > Azure Databricks >
Create an Azure Databricks workspace
Project Details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
Subscription *
MML Learners
Resource group *
rg-azuser2373_mml.local-HSEqX
Create new
Instance Details
Workspace name *
HexaVSP
Region *
West US 2
Pricing Tier *
Premium (+ Role-based access controls)
We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.
Managed Resource Group name
Enter name for managed resource group
Review + create
< Previous
Next : Networking >