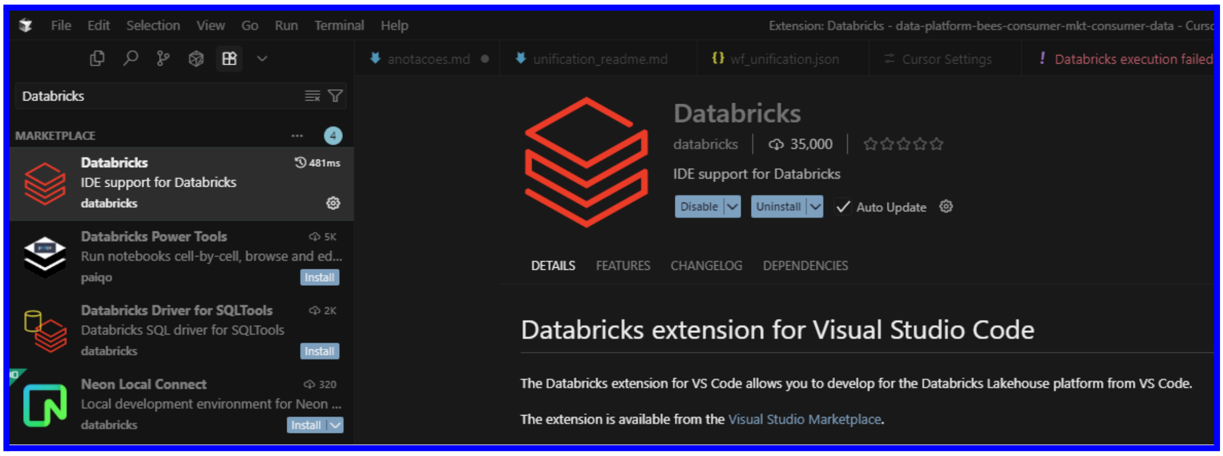
Databricks extension for Visual Studio Code and Cursor

fabio.reisclasso@emeal.nttdata.com

1.Install the Databricks extension


File Edit Selection View Go Run Terminal Help
Extension: Databricks - data-platform-bees-consumer-mkt-consumer-data - Curs
anotacoes.md
unification_readme.md
wf_unification.json
Cursor Settings
Databricks execution failed
Databricks
MARKETPLACE
Databricks
IDE support for Databricks
databricks
481ms
Databricks Power Tools
Run notebooks cell-by-cell, browse and ed...
paiqo
5K
Install
Databricks Driver for SQLTools
Databricks SQL driver for SQLTools
databricks
2K
Install
Neon Local Connect
Local development environment for Neon ...
databricks
320
Install
Databricks
databricks
35,000
IDE support for Databricks
Disable
Uninstall
Auto Update
DETAILS
FEATURES
CHANGELOG
DEPENDENCIES

Databricks extension for Visual Studio Code

The Databricks extension for VS Code allows you to develop for the Databricks Lakehouse platform from VS Code.

The extension is available from the Visual Studio Marketplace.

https://learn.microsoft.com/en-us/azure/databricks/dev-tools/vscode-ext/install

Install and open the extension

1. In **Visual Studio Code/Cursor**, open the Extensions view (View > Extensions from the main menu).

2. In Search Extensions in the Marketplace, enter Databricks.

a. https://marketplace.visualstudio.com/items?itemName=databricks.databricks

3. Click the Databricks extension and then click Install.

4. Click Reload Required, or restart Visual Studio Code.

5. To open the extension, on the sidebar, click the Databricks icon.

6. Configure the Databricks extension for Visual Studio Code to connect to your Azure Databricks workspace.

See Configure your Databricks project using the Databricks extension for Visual Studio Code.

2.Configure your Databricks project using the Databricks

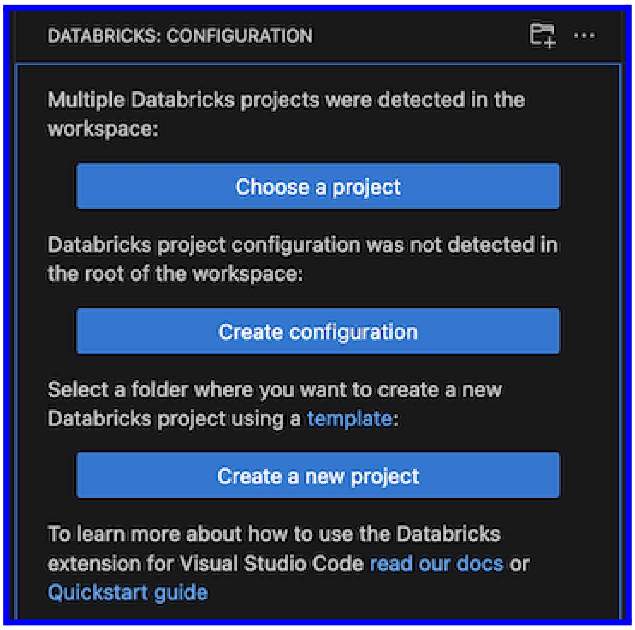
2.1 Configure the Databricks profile for the project

When you create a Databricks project or convert a project to be a Databricks project, you configure a profile that includes authentication settings used to connect to Databricks. If you want to change the authentication profile used, click the gear icon associated with AuthType in the Configuration view.

For more information Databricks extension for Visual Studio Code authentication, see Set up authorization for the Databricks extension for Visual Studio Code

Opening the Databricks extension you will find three possibilities:

1. Choose a project
2. Create configuration
3. Create a new project



DATABRICKS: CONFIGURATION

Multiple Databricks projects were detected in the workspace:

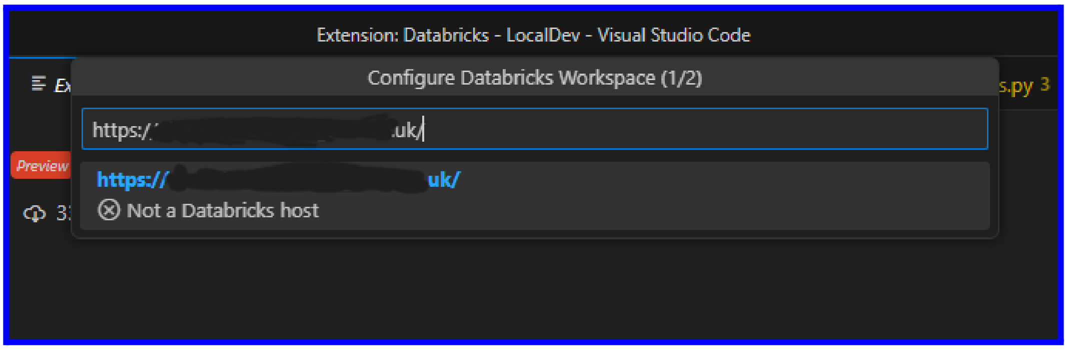
Databricks project configuration was not detected in the root of the workspace:

Select a folder where you want to create a new Databricks project using a template:

To learn more about how to use the Databricks extension for Visual Studio Code read our docs or Quickstart guide

Choose your project

2.1.1 Choosing a project



Extension: Databricks - LocalDev - Visual Studio Code

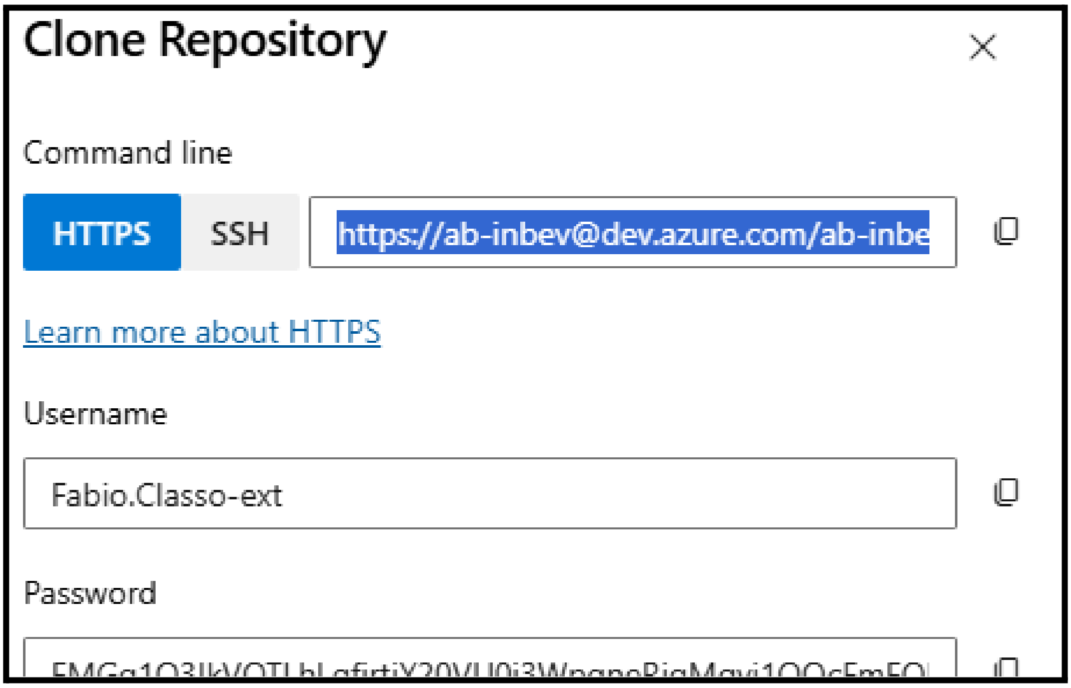
Configure Databricks Workspace (1/2)

https:/[redacted].uk/

https://[redacted]uk/

ⓧ Not a Databricks host

Add you WorkSapce url like:
https://adb-3985425310945749.9.azuredatabricks.net/browse/folders/1627126173437543



Clone Repository

Command line

https://ab-inbev@dev.azure.com/ab-inbe

Learn more about HTTPS

Username

Fabio.Classo-ext

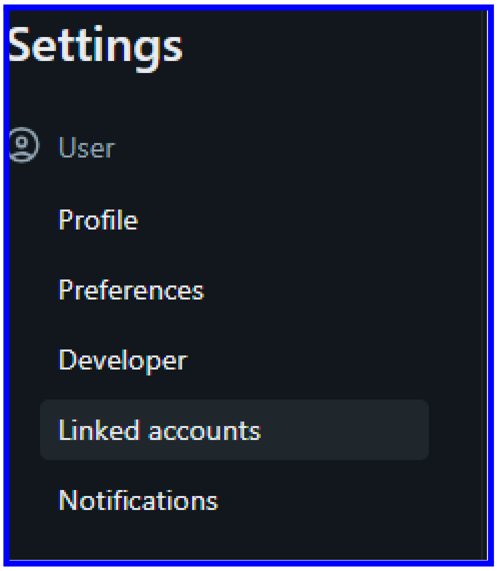
Password

FMGq1O3lkVOTLbLqfirtiY20VU0i3WpqnoBiqMqvj1OOcFmFO

You will need to provide your Azure DevOps password.

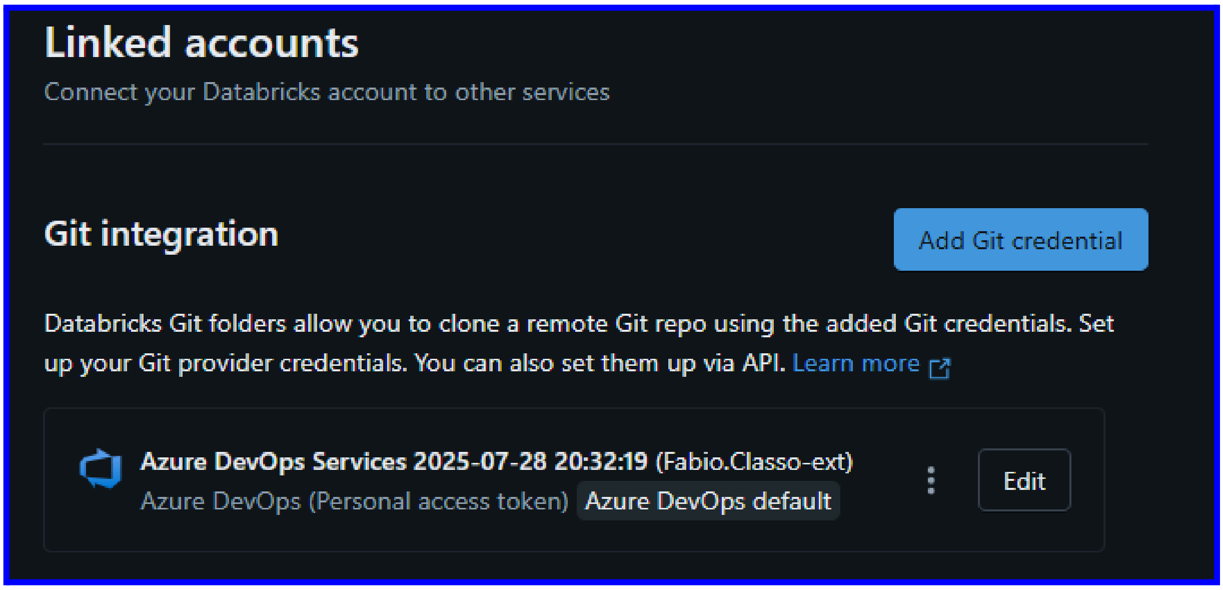
2.1.2 Linked accounts

Connect your Databricks account to other services



Settings

User



Linked accounts

Connect your Databricks account to other services

Git integration

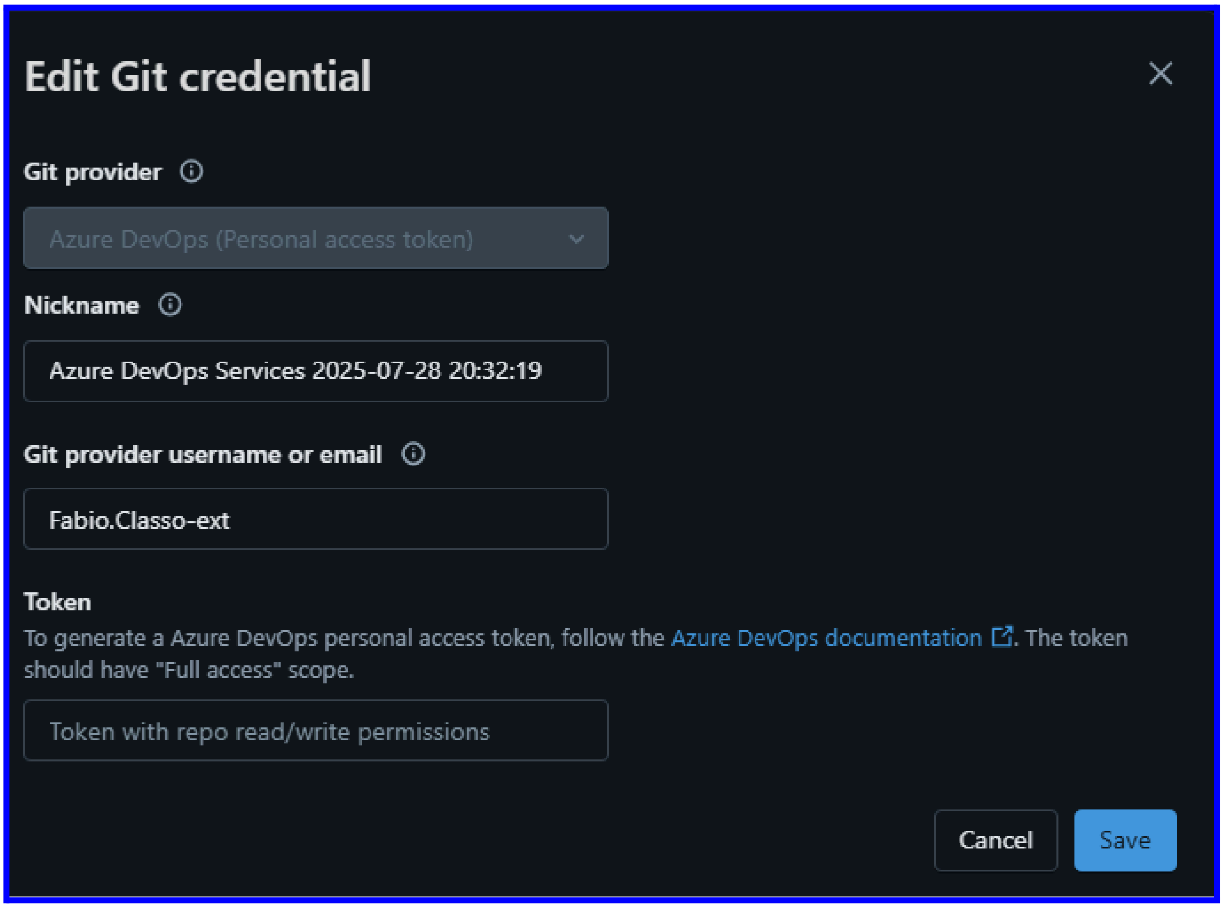
Databricks Git folders allow you to clone a remote Git repo using the added Git credentials. Set up your Git provider credentials. You can also set them up via API. Learn more

Azure DevOps Services 2025-07-28 20:32:19 (Fabio.Classo-ext)

Azure DevOps (Personal access token) Azure DevOps default

2.1.3 Git integration

Databricks Git folders allow you to clone a remote Git repo using the added Git credentials. Set up your Git provider credentials. You can also set them up via AP.



Edit Git credential

Git provider: Azure DevOps (Personal access token)

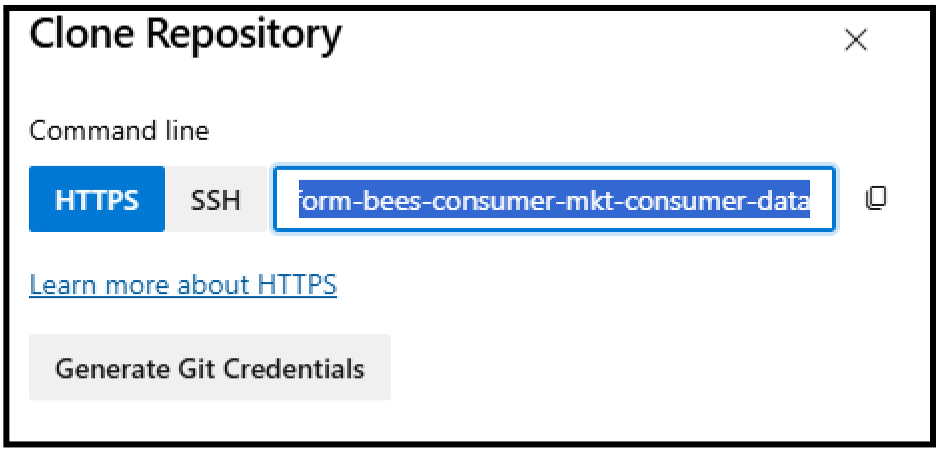
Nickname: Azure DevOps Services 2025-07-28 20:32:19

Git provider username or email: Fabio.Classo-ext

Token

To generate a Azure DevOps personal access token, follow the Azure DevOps documentation. The token should have "Full access" scope.

Token with repo read/write permissions

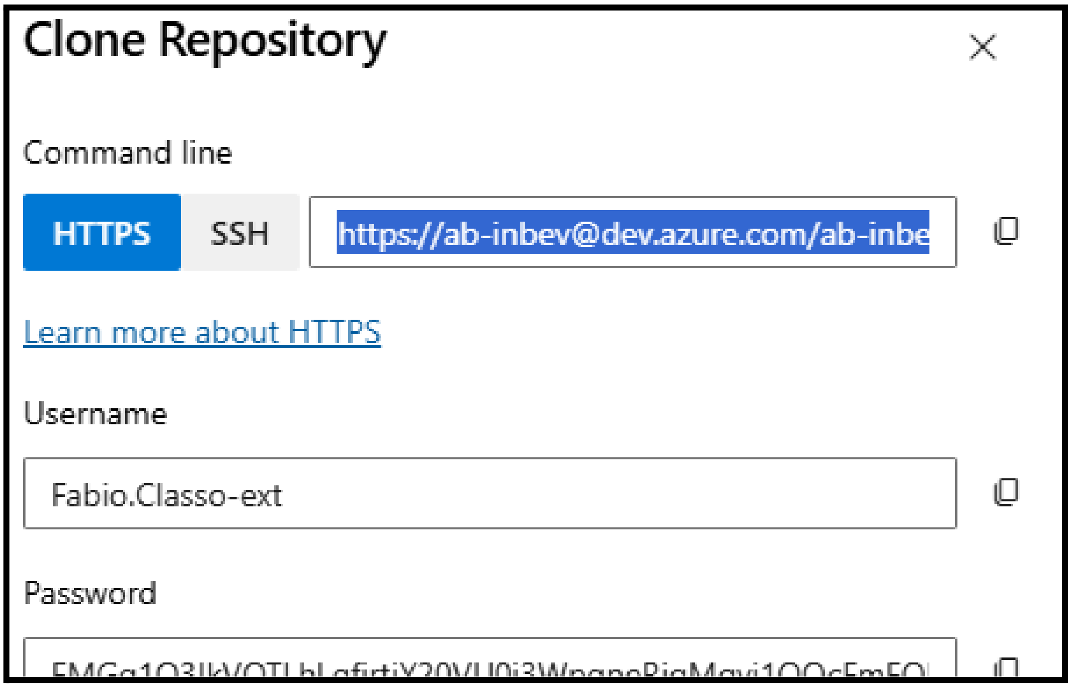


Clone Repository

Command line

HTTPS | SSH | form-bees-consumer-mkt-consumer-data

Learn more about HTTPS



Clone Repository

Command line

HTTPS SSH https://ab-inbev@dev.azure.com/ab-inbe

Learn more about HTTPS

Username

Fabio.Classo-ext

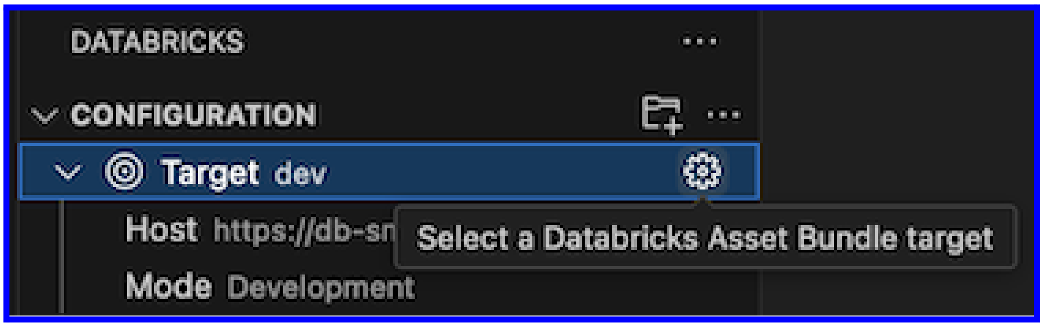
Password

FMGq1O3IkVOTLbLqfirtiX20VU0i3WpqnoPiqMqvi1OOcFmFO

2.1.4 Change the target deployment workspace

To select or switch the deployment target for your Databricks project (for example, to switch from a dev target to a prod target):

1. In the Configuration view of the Databricks extension panel, click the gear icon (Select a Databricks Asset Bundle target) associated with Target.



Target dev

Host https://db-sn

Mode Development

Select a Databricks Asset Bundle target

2. In the Command Palette, select the desired deployment target.

https://learn.microsoft.com/en-us/azure/databricks/dev-tools/vscode-ext/configure

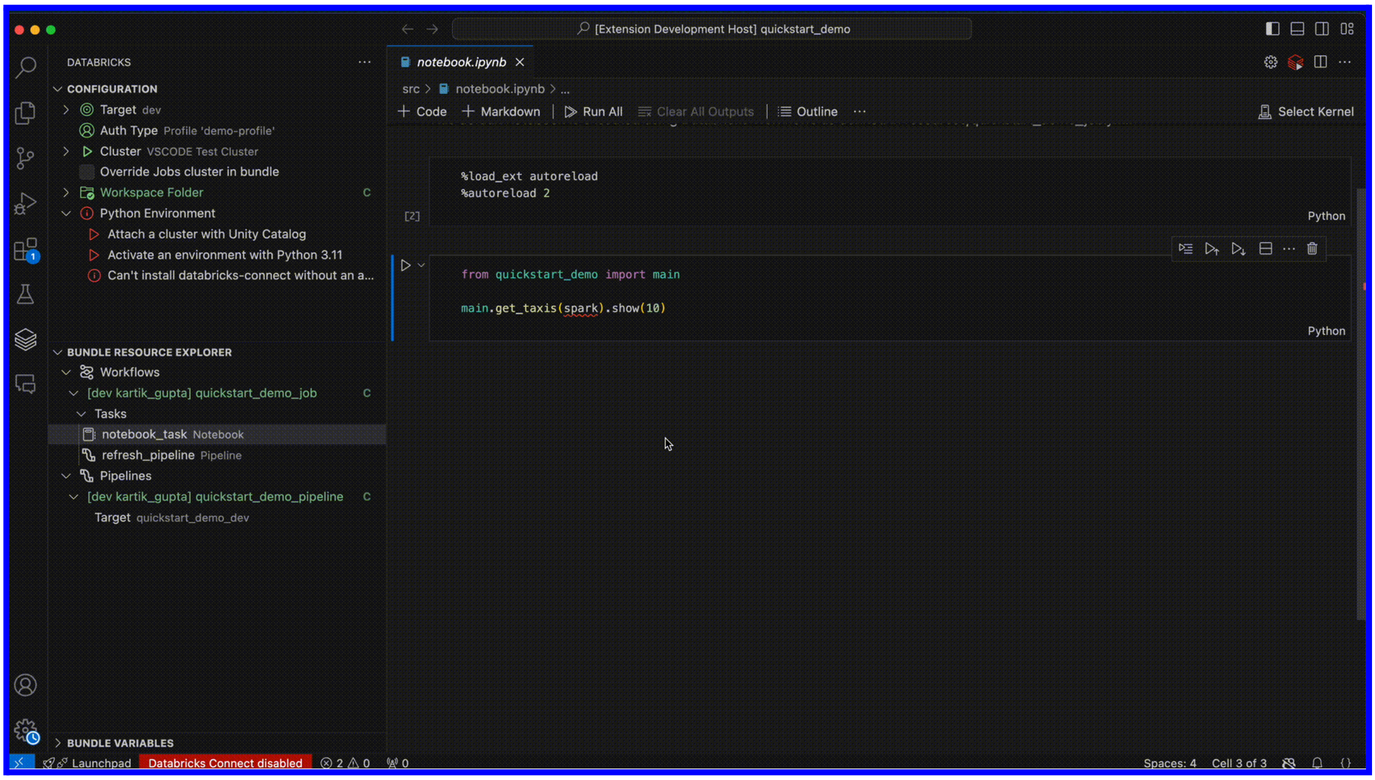
2.1.5 Creating configuration (Databricks Asset Bundles)

What are Databricks Asset Bundles?

Databricks Asset Bundles make it possible to describe Databricks resources such as jobs, pipelines, and notebooks as source files. These source files provide an end-to-end definition of a project, including how it should be structured, tested, and deployed, which makes it easier to collaborate on projects during active development. For more information, see Databricks Asset Bundles.

Deploying Databricks Asset Bundles

1. In the Databricks extension panel, find the "Bundle Resource Explorer" view.
2. Click on the "Deploy" button.
3. You can monitor the deployment status in the log output window.



```
%load_ext autoreload
%autoreload 2
```

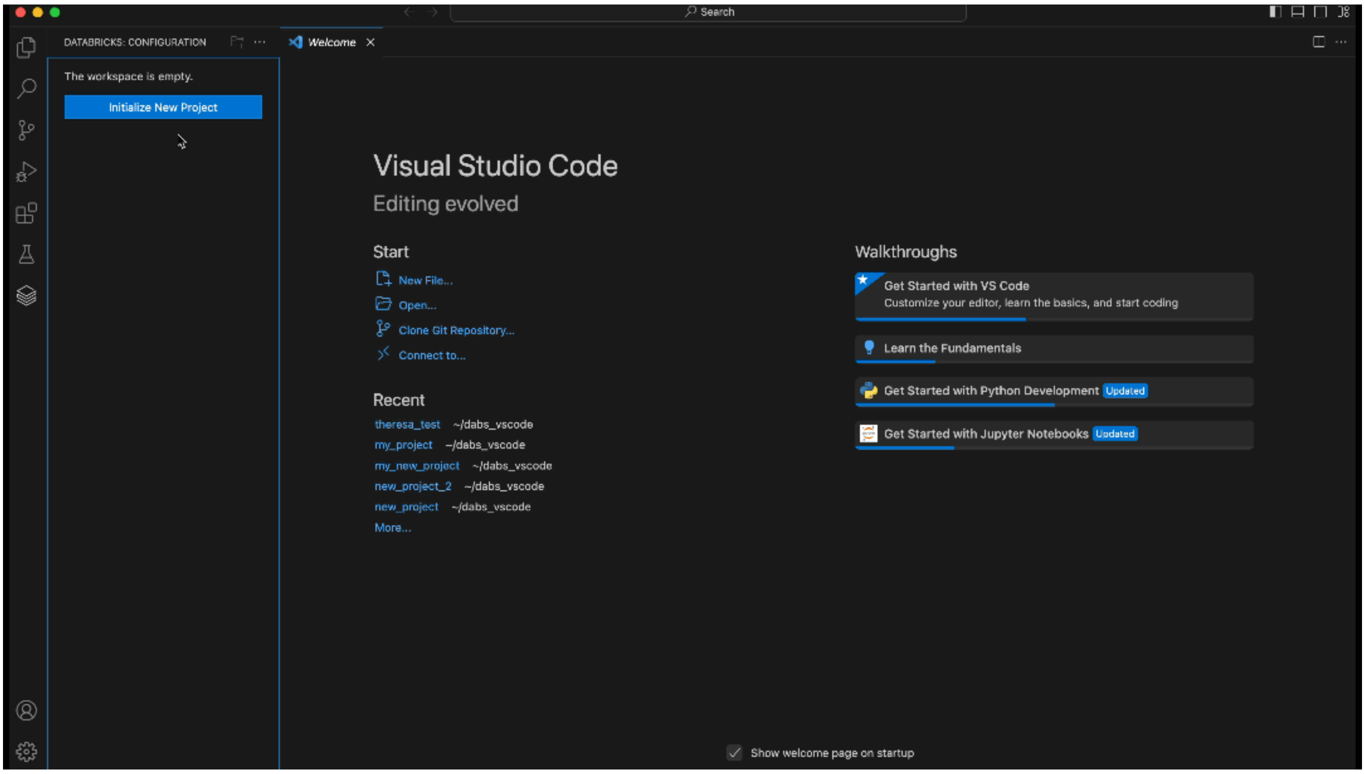
```
from quickstart_demo import main

main.get_taxis(spark).show(10)
```

CONFIGURATION
- Target dev
- Auth Type Profile 'demo-profile'
- Cluster VSCODE Test Cluster
- Override Jobs cluster in bundle
- Workspace Folder
- Python Environment
  - Attach a cluster with Unity Catalog
  - Activate an environment with Python 3.11
  - Can't install databricks-connect without an a...

BUNDLE RESOURCE EXPLORER
- Workflows
  - [dev kartik_gupta] quickstart_demo_job
    - Tasks
      - notebook_task Notebook
      - refresh_pipeline Pipeline
- Pipelines
  - [dev kartik_gupta] quickstart_demo_pipeline
    - Target quickstart_demo_dev

2.1.6 Create a new project

1. Open the Databricks extension panel by clicking on the Databricks icon on the left sidebar.
2. Click on the "Create a new Databricks project" button.
3. Follow the selection dialogs to create a Databricks configuration profile or select an existing one.
4. Select a folder to create your project in.
5. Follow the selection dialogs to create a new Databricks project.
6. Select the newly created project to open it, using the selector that appears.
7. VS Code will reopen with the new project loaded, and the extension will automatically login using the selected profile.



The workspace is empty.

Search
DATABRICKS: CONFIGURATION
Welcome
Initialize New Project
Visual Studio Code
Editing evolved
Start
New File...
Open...
Clone Git Repository...
Connect to...
Recent
theresa_test ~/dabs_vscode
my_project ~/dabs_vscode
my_new_project ~/dabs_vscode
new_project_2 ~/dabs_vscode
new_project ~/dabs_vscode
More...
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
Get Started with Python Development Updated
Get Started with Jupyter Notebooks Updated
Show welcome page on startup

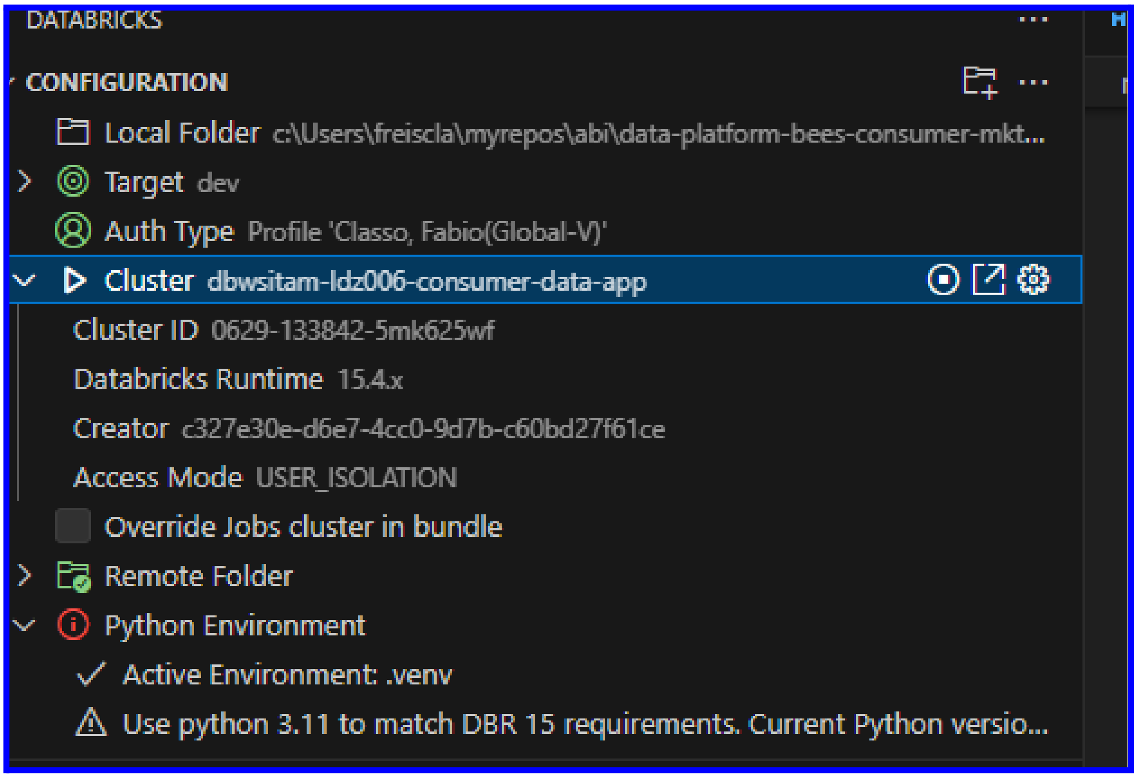
3.Select compute for running code and jobs

Using the Databricks extension for Visual Studio Code, you can select serverless, select an existing Azure Databricks cluster, or create a new Azure Databricks cluster, for running your code and jobs. Once you have connected to compute, a cluster's ID, Databricks Runtime version, creator, state, and access mode are displayed. You can also start and stop a cluster, and navigate directly to the cluster's page details.

3.1 Use an existing cluster

If you have an existing Azure Databricks cluster that you want to use:

1. In the Configuration view, next to Cluster, click Select a cluster or the gear (Configure cluster) icon.
2. In the Command Palette, select the cluster that you want to use.



DATABRICKS

CONFIGURATION

Local Folder c:\Users\freiscla\myrepos\abi\data-platform-bees-consumer-mkt...

Target dev

Auth Type Profile 'Classo, Fabio(Global-V)'

Cluster dbwsitam-ldz006-consumer-data-app

Cluster ID 0629-133842-5mk625wf

Databricks Runtime 15.4.x

Creator c327e30e-d6e7-4cc0-9d7b-c60bd27f61ce

Access Mode USER_ISOLATION

□ Override Jobs cluster in bundle

Remote Folder

Python Environment

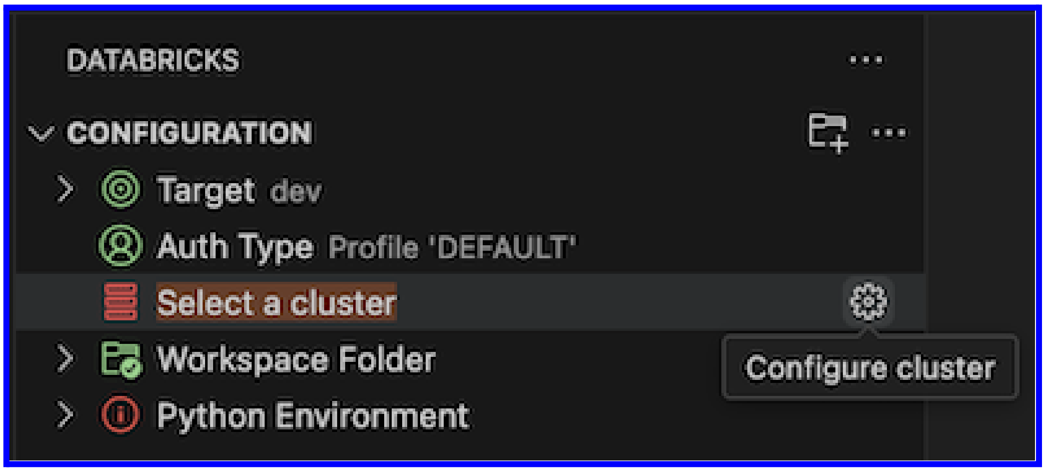
✓ Active Environment: .venv

⚠ Use python 3.11 to match DBR 15 requirements. Current Python versio...

3.2 Use serverless

Serverless compute is managed by Azure Databricks. When you run workloads on serverless compute, Azure Databricks automatically allocates and manages the necessary compute resources.

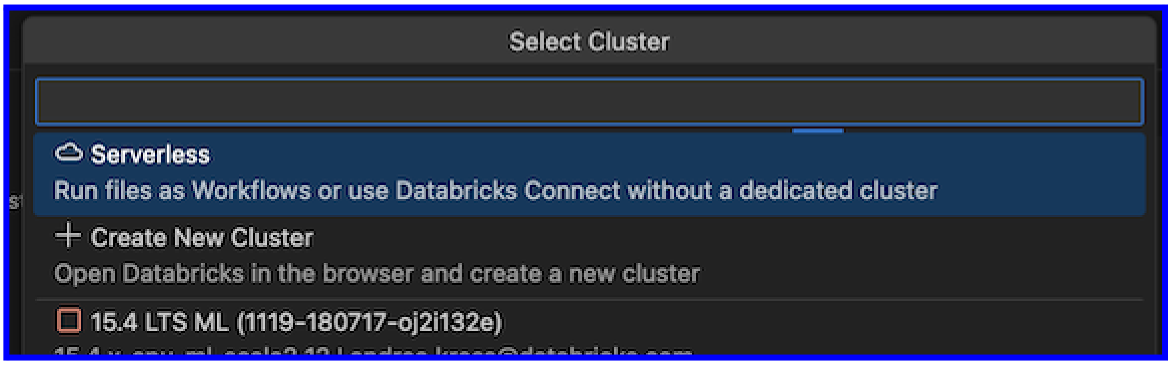
1. In the Configuration view, next to Cluster, click Select a cluster or the gear (Configure cluster) icon.



Target dev

Auth Type Profile 'DEFAULT'

2. In the Command Palette, select Serverless.



Serverless

Run files as Workflows or use Databricks Connect without a dedicated cluster

Create New Cluster

Open Databricks in the browser and create a new cluster

15.4 LTS ML (1119-180717-oj2i132e)

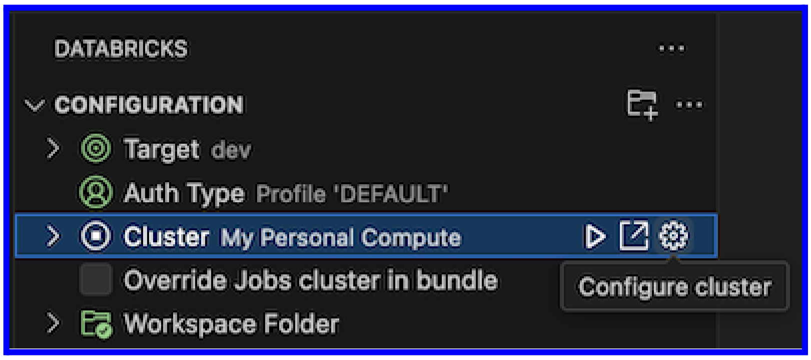
3.3 Create a new cluster

If you do not have an existing Azure Databricks cluster, or you want to create a new one:

1. In the Configuration view, next to Cluster, click the gear (Configure cluster) icon.
2. In the Command Palette, click Create New Cluster.
3. When prompted to open the external website (your Azure Databricks workspace), click Open.
4. If prompted, sign in to your Azure Databricks workspace.

Follow the instructions to create a cluster.

1. After the cluster is created and is running, go back to Visual Studio Code.
2. In the Configuration view, next to Cluster, click the gear (Configure cluster) icon.



Target dev

Auth Type Profile 'DEFAULT'

Cluster My Personal Compute

□ Override Jobs cluster in bundle

Workspace Folder

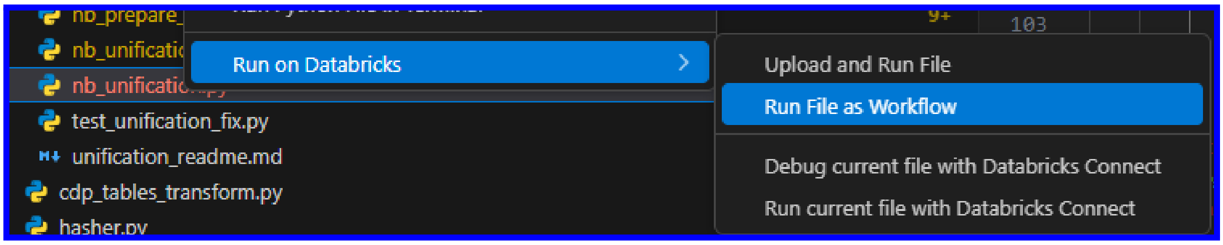
4. Run Python on a cluster and as a job using the Databricks extension

Run the code as a job

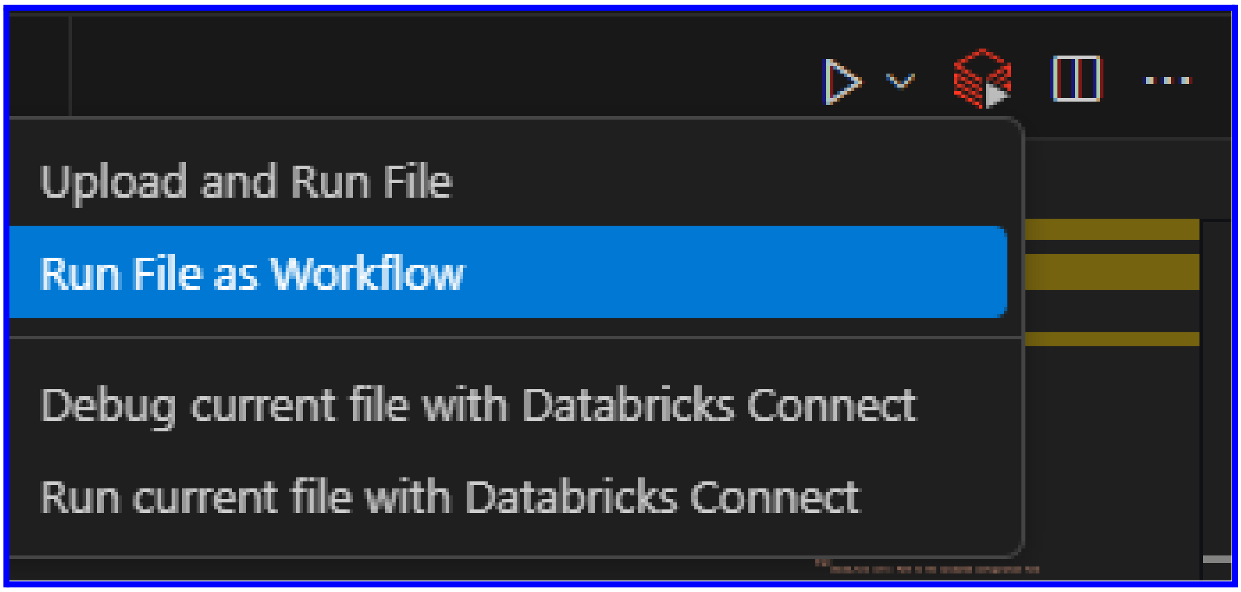
To run demo.py as a job, click the Run on Databricks icon next to the list of editor tabs, and then click Run File as Workflow. The output appears in a separate editor tab next to the demo.py file editor.

https://learn.microsoft.com/en-us/azure/databricks/dev-tools/vscode-ext/tutorial?source=recommendations

1 - By clicking with the right button on the Python file and choosing **"Run on Databricks"**

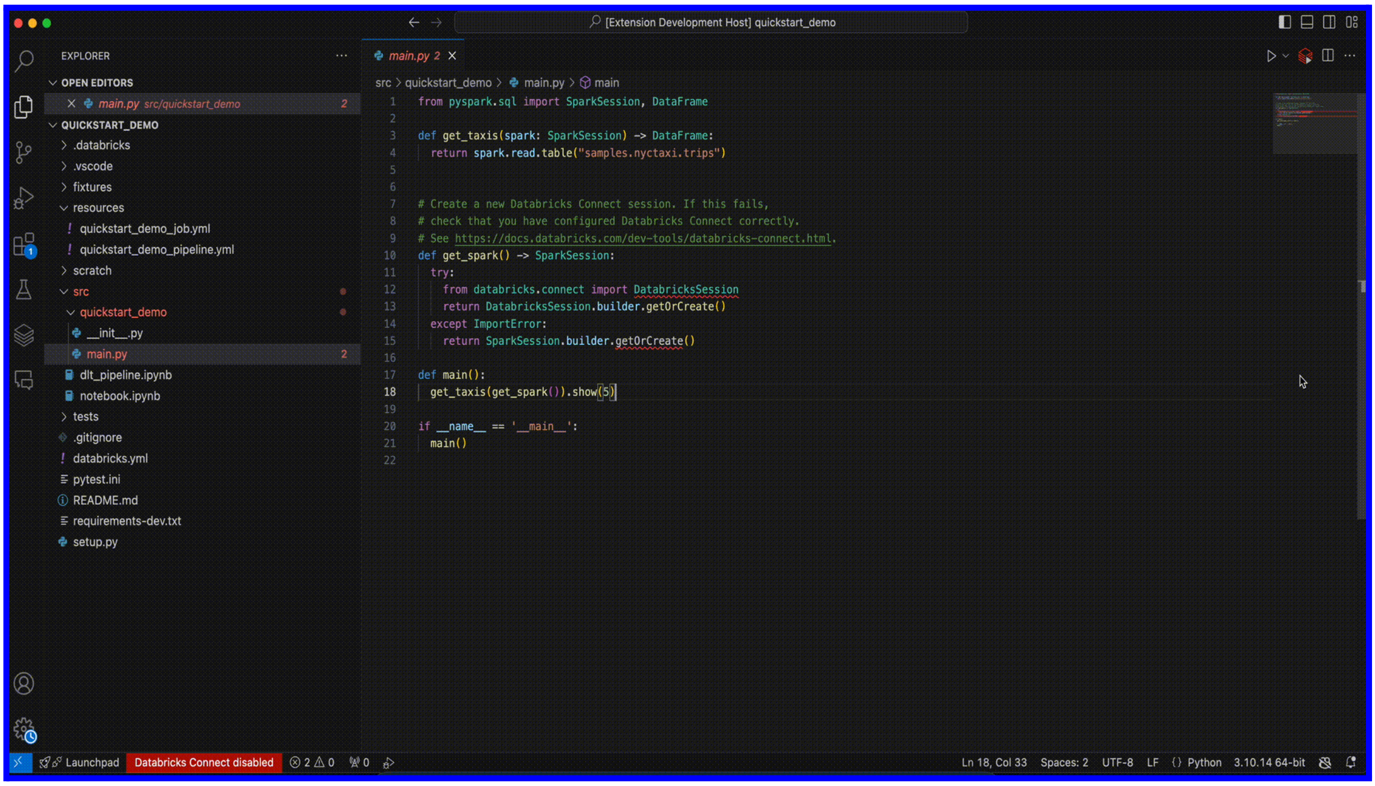


2 - Clicking on the left on the red icon



Upload and Run File
Run File as Workflow
Debug current file with Databricks Connect
Run current file with Databricks Connect

The code will be executed and the output displayed in the VsCode interface.



```python
from pyspark.sql import SparkSession, DataFrame

def get_taxis(spark: SparkSession) -> DataFrame:
  return spark.read.table("samples.nyctaxi.trips")


# Create a new Databricks Connect session. If this fails,
# check that you have configured Databricks Connect correctly.
# See https://docs.databricks.com/dev-tools/databricks-connect.html.
def get_spark() -> SparkSession:
  try:
    from databricks.connect import DatabricksSession
    return DatabricksSession.builder.getOrCreate()
  except ImportError:
    return SparkSession.builder.getOrCreate()

def main():
  get_taxis(get_spark()).show(5)

if __name__ == '__main__':
  main()
```

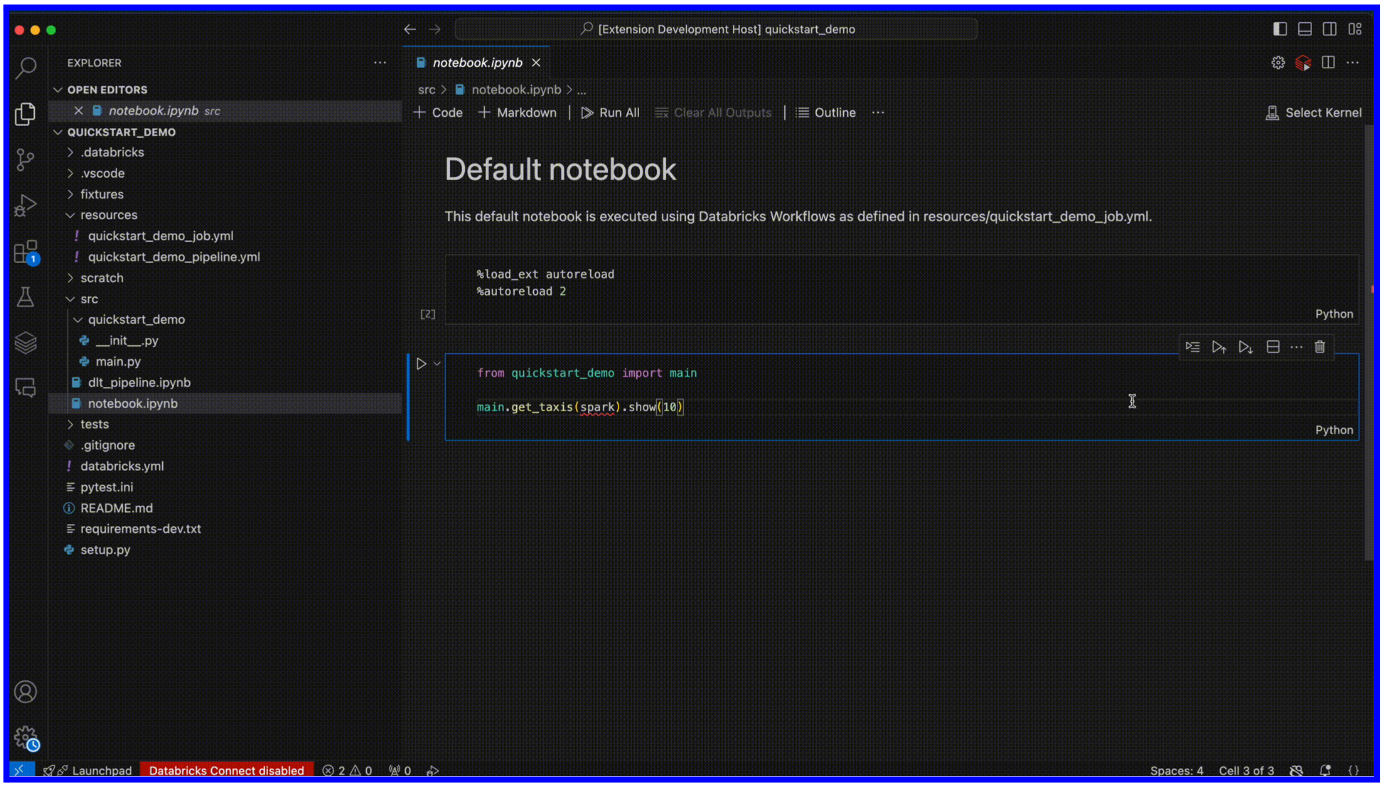
Running Notebooks as a Workflow

1. Create a python file or a python based notebook

a. You can create a python based notebook by exporting a notebook from the Databricks web application or use a notebook that is already tracked in git, such as https://github.com/databricks/notebook-best-practices

2. Click the "Databricks Run" icon in the tab bar and select "Run File as Workflow on Databricks"

This will run the file using the Jobs API on the configured cluster and render the result in a WebView.



Default notebook

This default notebook is executed using Databricks Workflows as defined in resources/quickstart_demo_job.yml.

```
%load_ext autoreload
%autoreload 2
```

```
from quickstart_demo import main

main.get_taxis(spark).show(10)
```

References:

https://docs.databricks.com/aws/pt/dev-tools/vscode-ext

https://learn.microsoft.com/en-us/azure/databricks/dev-tools/vscode-ext/configure