

The Analytics Workbench

The Analytics Workbench allows users to launch compute instances with Jupyter notebooks, using Python or R, in a personal workspace environment. Workbench add-ons include 1) Data Wrangler, a no-code data transformation, analysis, and visualization option and 2) SAS Viya Analytics Pro, a cloud-hosted SAS environment that allows scalable computing, data storage, and usage tracking to enable data access, transform, analysis, visualization, and mapping capabilities.

Note: There is a limited number of licenses available for SAS and Data Wrangler. The licenses are distributed on a first come first serve basis.

JupyterLab

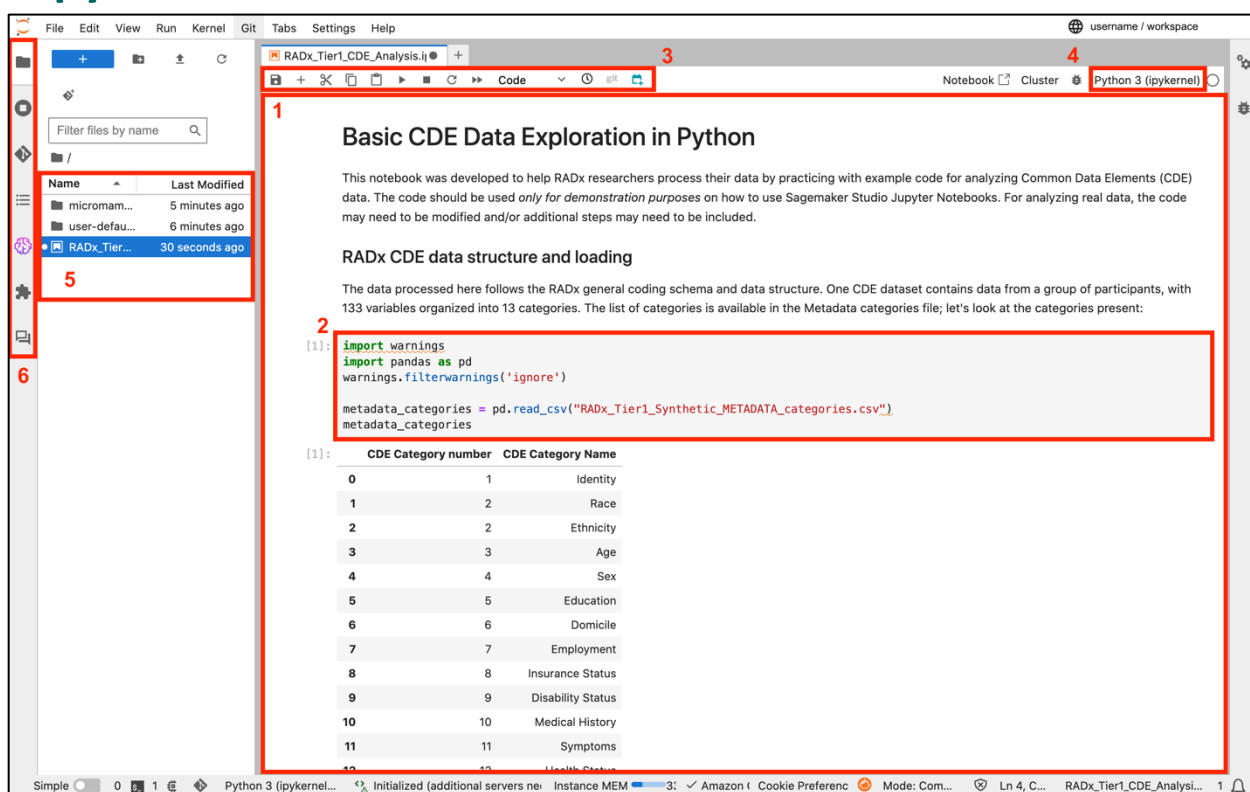


Figure 1: Labeled Jupyter Notebook Interface

The main elements of the JupyterLab editor are:

- Notebook:** A document containing analysis code, outputs, and any additional markdown or text.
- Cell:** A single section of a notebook where to enter code, markdown, or text.
- Toolbar:** Perform the most common notebook actions, including:
 - Save
 - Insert cell below
 - Cut selected cell
 - Copy selected cell
 - Paste from clipboard

- Run selected cell
 - Interrupt the kernel
 - Restart the kernel
 - Restart the kernel and run all cells
 - Change cell type (i.e., Code, Markdown, Raw)
 - Launch terminal
4. **Environment:** Displays the current notebook kernel type.
 5. **File Browser:** Displays lists of folders, notebooks, and other files.
 - The Personal Studio environment is a private, personal Amazon EFS directory.
 6. **Left sidebar:** Contains tabs to access the following functionalities.
 - **File Browser:** Displays lists of folders, notebooks, and other files.
 - **Running Terminals and Kernels:** View current kernels and terminals running in JupyterLab. Optionally shut down all or select resources (i.e., notebooks, terminals, kernels, apps, and instances).
 - **Git:** Connects to a Git repository for Git tool and operation access.
 - **Table of Contents:** Automatically generated for each notebook, Markdown file, or Python file open to navigate the document's structure with clickable entries.
 - **Extension Manager:** Enables and manages third-party JupyterLab extensions.
 - **Jupyter AI:** A JupyterLab tool to explore generative AI models and integrate them into notebooks.

Create and Launch a JupyterLab Space

The default workspace environment is a ml.t3.medium (2 vCPU, 4 GiB memory) instance type.

To create a new JupyterLab space:

1. When the Workbench is launched, select “JupyterLab” from the “Overview” section, or select “JupyterLab” from the “Applications” in the left panel (Figure 2).

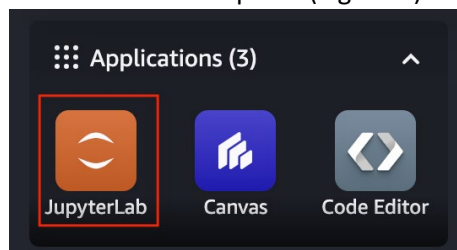


Figure 2: Workbench applications highlighting JupyterLab

2. Select “+ Create JupyterLab space” in the upper right corner of the JupyterLab page.
 - In the “Create JupyterLab space” dialog, specify a name for the space in the “Name” field. To finish, click “Create space.”
 - **Note:** Because the platform is shared, workspaces must have a unique name. If the workspace name already exists, the following error will appear at the bottom of the page (Figure 3):



Figure 3: Error message for workspace name exists

To launch a JupyterLab space:

1. From the Workbench Home page, select “JupyterLab” from the Overview section, or select “JupyterLab” from “Applications” in the left panel (Figure 2).
2. Select “Run” in the Action column of the JupyterLab space to start the workspace (Figure 4). This may take up to a minute to start.

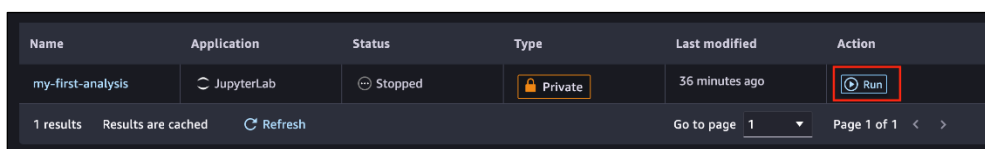


Figure 4: Start Running JupyterLab Space

3. Once the status changes to “Running”, select the “Open” icon in the Action column to launch JupyterLab in a new tab (Figure 5).

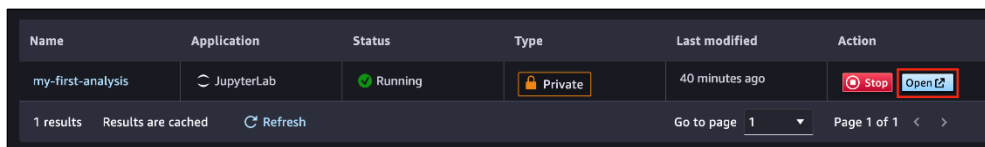


Figure 5: Open JupyterLab space

To create a new notebook:

1. From the landing page, select “File,” “New,” and “Notebook” (Figure 6).
 - o In the “Select Kernel” dialog, select a kernel on the dropdown menu. To finish, click “Select”, which launches the notebook.

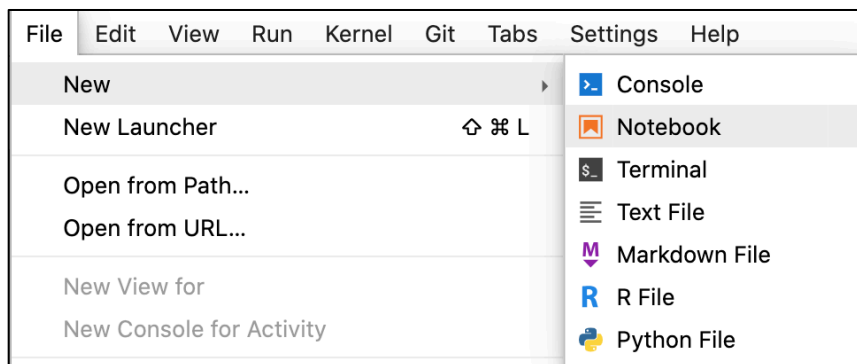


Figure 6: Launch Notebook from File Menu

2. From the Launcher page, click a preferred kernel in the Notebook section (Figure 7).

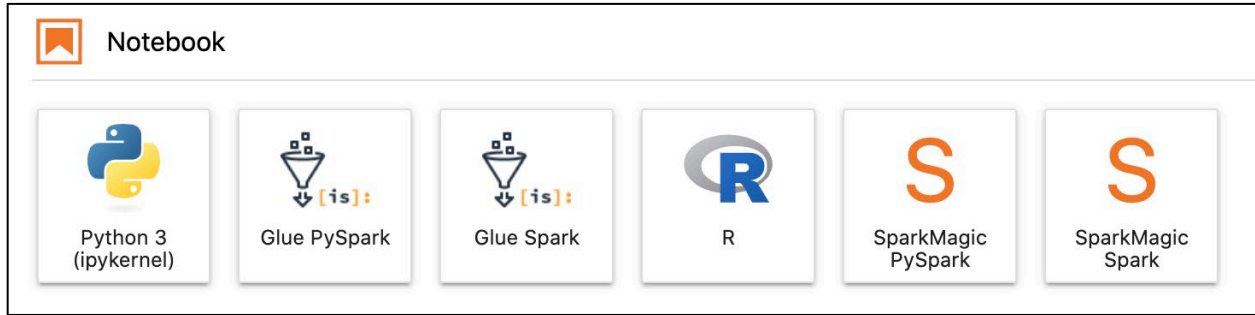





Figure 7: Launch Notebook Using Launcher

Upload and Download Files

To upload files from a local machine into a JupyterLab space:

1. In the left sidebar, choose the “File Browser”  icon.
2. In the File Browser, choose the “Upload Files”  icon.
3. Select the files to upload and choose “Open.”
4. Once the file appears in the home folder, double-click the file to open it in a new tab.

To download a file locally:

1. In the left sidebar, choose the File Browser  icon.
2. Right click the file and select “Download.”


To download an entire folder locally:

1. From the menu, choose “File,” “New,” and “Terminal,” which will launch a Terminal in a new JupyterLab tab.
2. Type the following command replacing `folder_name` and `/path/to/folder`:

```
zip -r -X folder_name.zip /path/to/folder
```
3. Once the folder is zipped and it appears in the File Browser, right click the .zip file and select “Download.”

Clone a Git Repository

Git repositories can be cloned into the JupyterLab home folder using the following steps:

1. Select the Git  icon in the left sidebar.
2. Choose “Clone a Repository.”
3. In the Clone Git Repository window, enter the Git URL (for example: <https://github.com/aws/amazon-sagemaker-examples.git>)
4. Under “Project directory to clone into,” enter the path to the local directory where the cloned directory should exist, otherwise Studio will clone the repository into the home directory.
5. Choose “Clone,” which will automatically open a new terminal window and clone the repository. This may take up to a minute depending on the repository size.
6. If the repository requires credentials, a prompt will appear to enter a username and personal GitHub account access token.

7. When complete, the File Browser will open, displaying the cloned repository.
8. Choose the Git icon to view the Git user interface, which tracks the repository.
9. To track a different repository, open the repository in the file browser and click the Git icon.

Create a persistent conda environment

Environments can be customized by installing and removing extensions and packages as needed. Any installed extensions and packages installed on the environment will persist. To create persistent conda environments in the JupyterLab application, use the following steps:

1. Open a JupyterLab space.
2. From the landing page, select “File,” “New,” and “Terminal”.
3. Within the terminal, create a new conda environment, replacing `myenv` with the desired environment name:

```
conda create -n myenv
```
4. Activate the environment:

```
conda activate myenv
```
5. Install any necessary packages for the environment, for example:

```
conda install numpy pandas
```
6. Install the ipykernel to create a kernel option. This step can be skipped if it has already been installed:

```
conda install ipykernel
```
7. Add the new conda environment to the Jupyter kernel, changing the `--display-name` option as preferred:

```
python -m ipykernel install --user --name myenv --display-name "MyEnvironment"
```
8. Verify installation of the kernel:

```
jupyter kernelspec list
```
9. When a notebook is launched, the new kernel should appear. If the kernel is not listed, close the tab and reopen the JupyterLab space.

Access Public Data

To access curated public and synthetic datasets on the RADx Data Hub’s Data Access page, follow the [Public Data Tutorial](#).

Datasets from the [AWS Registry of Open Data](#), an AWS-hosted repository of more than 400 publicly available datasets, can be copied into a JupyterLab environment using the following steps:

1. Identify a dataset of interest and find the associated Amazon Resource Name (ARN).
 - a. For example: [COVID-19 Data Lake](#)
 - b. ARN: `arn:aws:s3:::covid19-lake`
 - c. The bucket name is `covid19-lake`
2. From the JupyterLab landing page, select “File,” “New,” then “Terminal.”
3. Enter the following command:

```
aws s3 sync s3://covid-lake .
```
4. Replace `covid19-lake` with a selected dataset bucket name.

Change Environment

Notebooks launch with the minimum instance type available by default. The minimum instance type is appropriate for most tasks, however, a larger instance can be requested by submitting a [Support Request](#). Follow the instructions in the [User Support Requests Tutorial](#) and select “Workbench Support” when choosing a Request Type. Please provide as much detail as possible in the request for the support team to determine the best suitable environment. For more detailed information about available instance types and their performance capabilities, see [Available Studio Instance Types](#).

File Sync

If an added Workbench file does not appear in the File Browser of JupyterLab, the workspace should be resynced. Close the JupyterLab tab, and refresh the My Approved Data page. Then, follow the steps to relaunch the JupyterLab page. If the files still do not appear, the workspace may need to be manually synced with the following steps:

1. From the File menu, click “File,” “New,” and “Terminal.”
2. Enter the following into the Terminal: `./s3sync.sh`

If the files in a workspace are still missing, please submit a [Support Request](#).

Data Wrangler

By default, Data Wrangler uses the m5.4xlarge (16 vCPU, 64 GiB memory) instance type. To request a larger compute instance, please see the Change environment section. If a Data Wrangler instance has been provisioned, a Data Wrangler flow can be created using the following steps:

1. From the Workbench Home page, select “Canvas” from “Applications” in the left panel.
2. Click “Run Canvas” to start the instance. This may take up to 8 minutes.
3. Once the instance status has changed to “Running”, click the “Open Canvas” button to launch Canvas in a new tab.
4. Select the “Data Wrangler” application in the sidebar of Canvas applications and click “Create a data flow” which will open a dialog to rename the data flow for the analysis.
5. Click import data (Figure 8) and select the data file type (i.e., Tabular or Image). Select one of the data source options from the dropdown, or upload files directly. See [Import](#) to learn more about AWS data import options.
6. Data Wrangler can now be used to add transforms, analyze, and visualize your data. To learn more, see [Transform Data](#) and [Analyze and Visualize](#).
7. To export a data flow, click **Export** from the data flow page. To learn more about exporting data transformations to other platforms, see [Export](#).

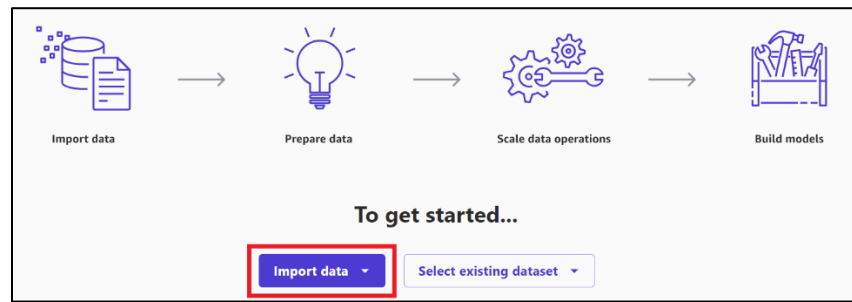


Figure 8: Import data to Data Wrangler flow

To stop running Canvas, click “Stop Canvas” on the Canvas homepage (Figure 9). Click the checkbox and the “Stop Canvas” button that appears to confirm shutdown. Be sure to save all work and data flows prior to shutdown.

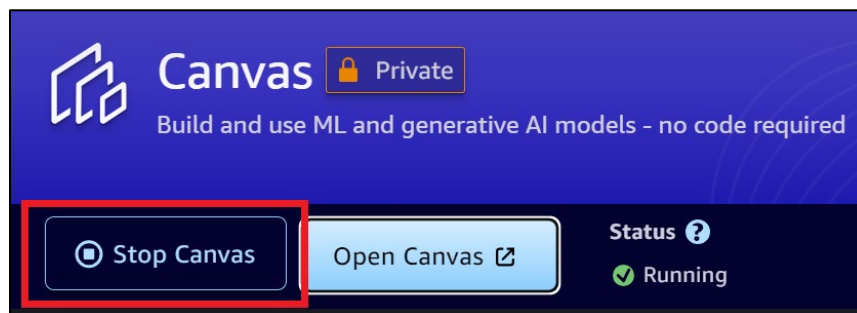


Figure 9: Stop Canvas application

SAS Viya

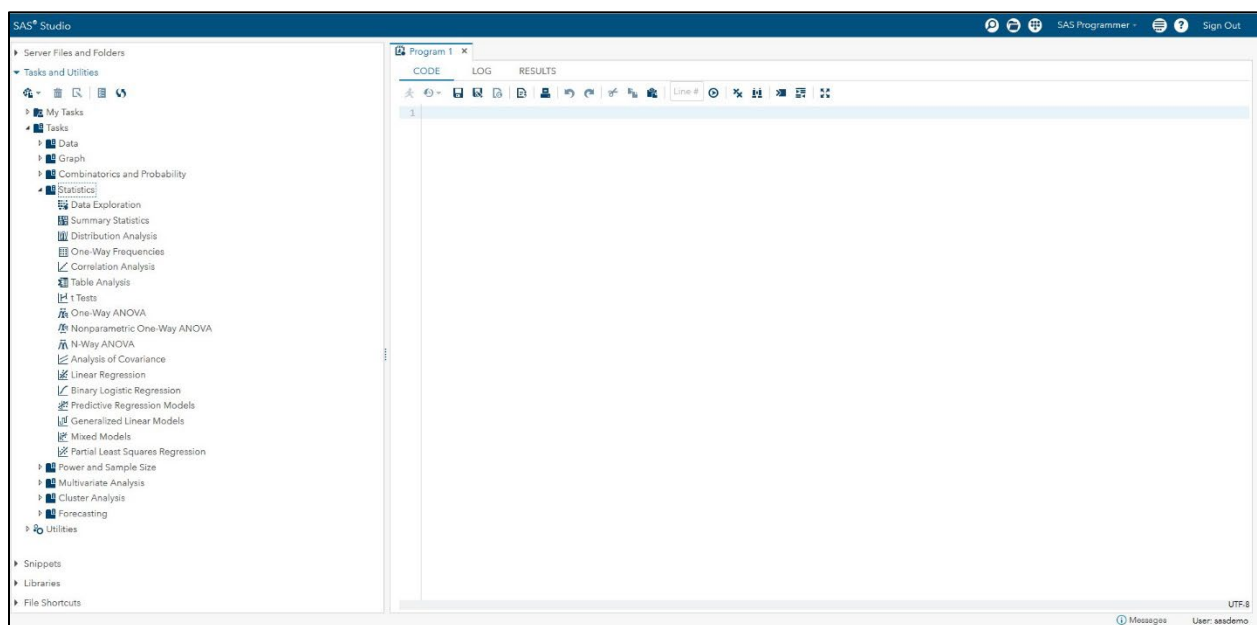



Figure 10: SAS Interface

From the My Approved Data or Public Data page, a “Launch SAS” button will appear if a license has been granted. To launch the SAS platform, click “Launch SAS.”


To start a new program:

1. From the main menu, click “New”, and “SAS Program”. A new blank program will open in the work area.


To save a SAS program:

2. Click the “Save”  icon on the work area toolbar.
3. Select the location where files will be saved.
4. Enter the name of the program.
5. Click “Save”.

To open a SAS program:

1. Click the  Open icon on the main menu bar.
2. Find the location of the program in the left panel.
3. Select the SAS program to open in the right panel.
4. Click “Open” and the program will appear in the work area.

To run just a portion of a SAS program, highlight the portion to run. To run the entire SAS program, no code needs to be highlighted:

1. Click the  Run icon on the work area toolbar.
2. Open the “Log” tab to confirm the program ran correctly.