IBM **TechXchange**

Hands-on lab guide

Citing your sources: Explaining generative AI output

Eric Martens
emartens@us.ibm.com
Learning Content Development, Data and AI

Loucas Loumakos
loucas.loumakos@ibm.com
Learning Content Development, Data and AI

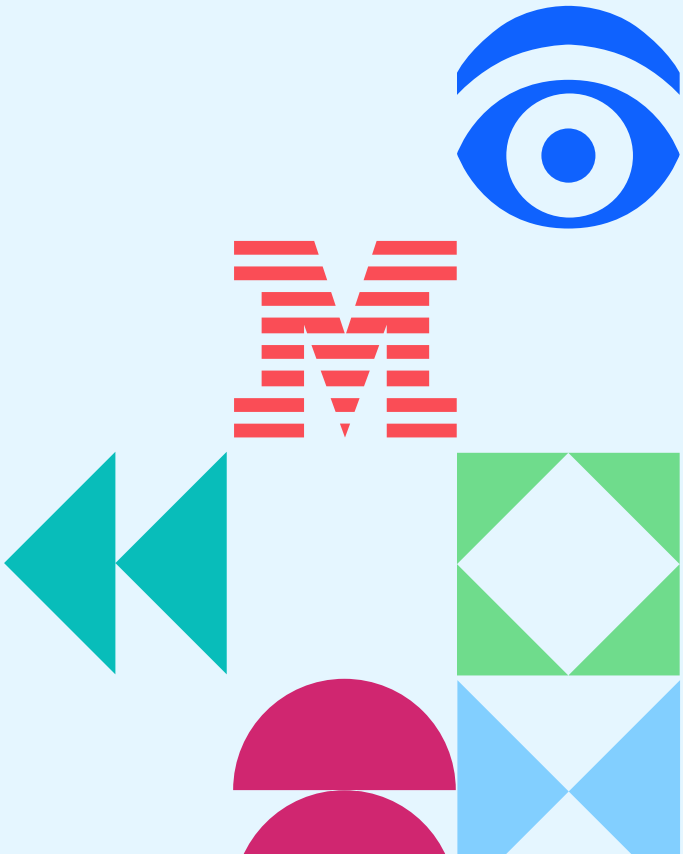


Table of Contents

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled,
isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

Notices and disclaimers (Continued)

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at

Learn more →

1 Introduction

The release of ChatGPT in November of 2022 sparked an explosion of interest in generative AI and large language models. The power and flexibility of the new models sparked a surge of creativity as organizations sought to apply new techniques to existing business problems. One of the most popular use cases for generative AI is retrieval-augmented generation (RAG), in which the model accesses additional authoritative information outside its training data set to answer questions and provide information. For example, an organization could provide its health care policy as the external dataset, and then allow the model to field employee questions about policy specifics. RAG offers the ability to customize and refine a general-purpose model for a specific use case without the time and expense of training a new model.

Additionally, the watsonx.governance platform allows organizations to generate explanations of the information provided by the RAG model, which can be critical for verifying the accuracy of the answers the model provides. In the health care policy example, the provided explanations can point to specific portions of the policy used to generate the response, allowing the user to trust that the model output is grounded in factual information, and is not a hallucination.

1.1 About this hands-on lab

In this lab, you will explore this capability by creating a project, associating it with the proper services, and running a Jupyter notebook that will allow you to see this service in action. Finally, you will explore how the Python SDK used in the lab can be easily integrated into an application that allows for the surfacing of the explanation to users.

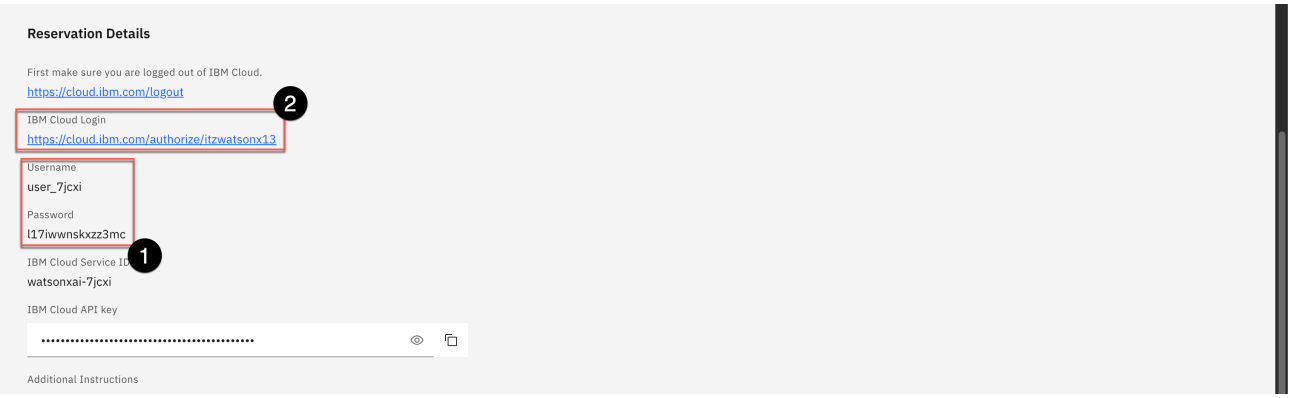
2 Setting up the environment

In this section, you will sign in to the watsonx environment, where you can build, test, deploy, and collaborate on data science and AI projects. Watsonx offers full access management, integration with dozens of databases and data connections, and a full suite of cutting-edge data science and AI tools from prompt labs to no-code rapid prototyping to industry-standard Jupyter notebook runtimes. For this lab, you will create a watsonx project, which organizes data, code, and other assets, and integrates with watsonx.ai and watsonx.governance services.

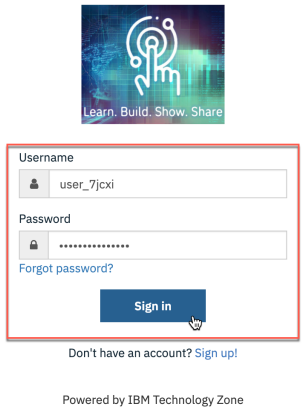
2.1 Sign into IBM Cloud

1. To begin, you will need to sign in to watsonx using the provided credentials. From your lab reservation screen, scroll down and locate the **Username** and **Password**. Copy and paste these values into a text file.

2. Click on the **IBM Cloud Login** link. The login window opens.



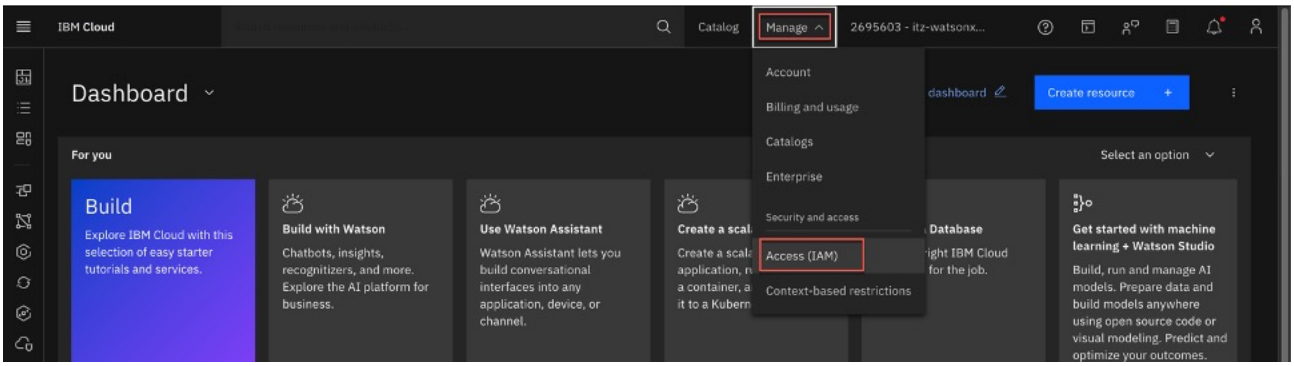
3. Enter the username and password into the appropriate fields and click the **Sign in** button. After you are signed in, you will be taken to the **IBM Cloud** dashboard.



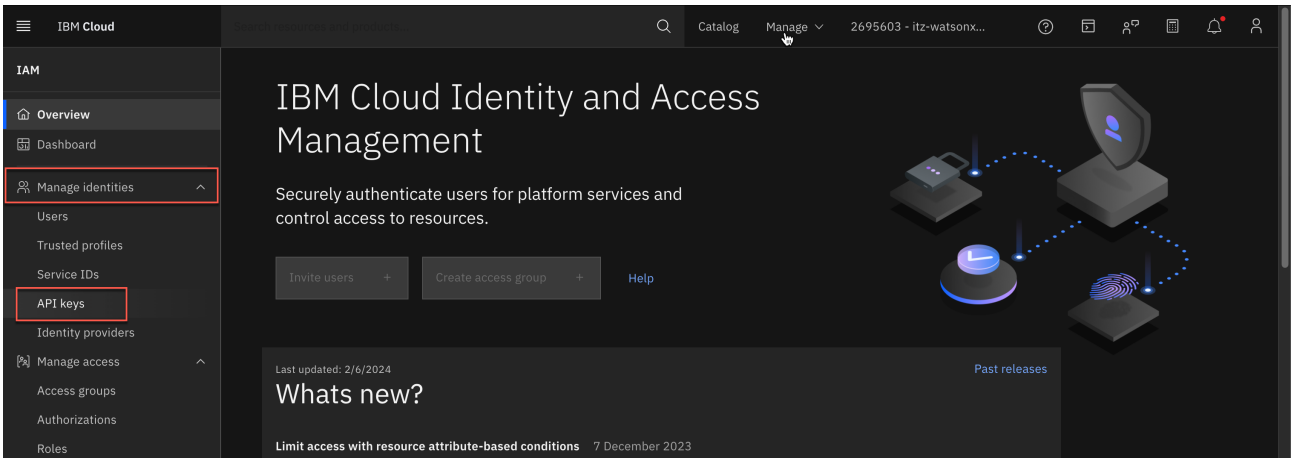
2.2 Generate an API key

To connect to services in your reserved account, you will need an API key, which can be generated via the user interface.

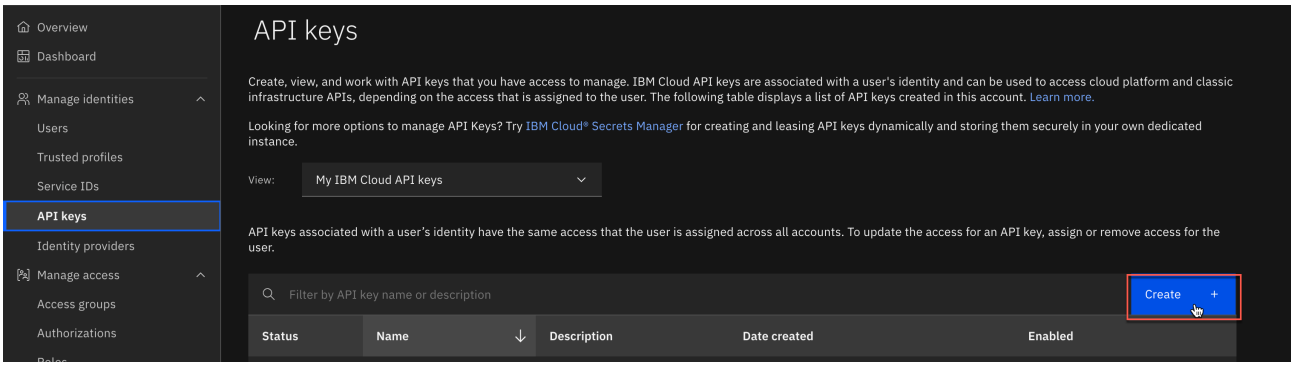
1. Click on the **Manage** button to open the menu, then click on the **Access (IAM)** menu item.



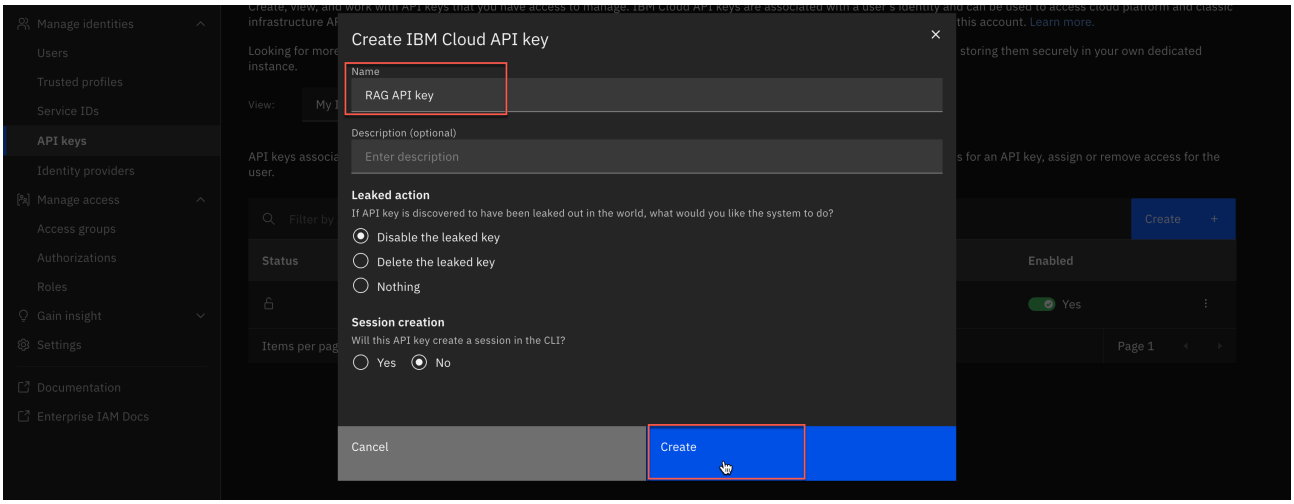
2. If necessary, click on the **Manage identities** menu item to expand it, then click on the **API keys** menu item.



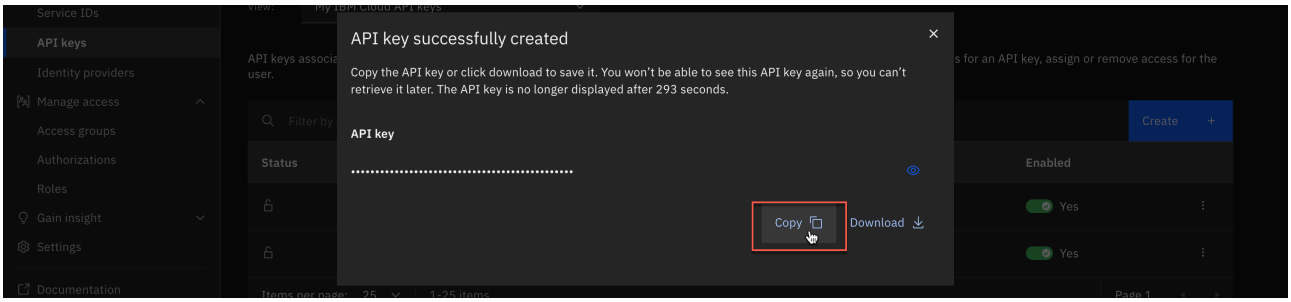
3. Click on the **Create** button. The **Create IBM Cloud API key** window opens.



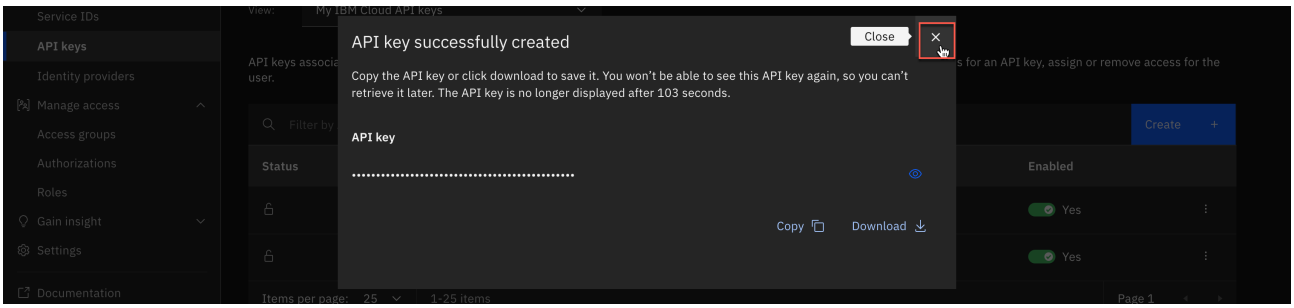
4. Give your API key a name such as **RAG API key**, then click the **Create** button.



5. Once the key has been created, click the **Copy** button to copy it to your clipboard, then paste the value into the text file that contains your login credentials. When you edit the Jupyter notebook later in the lab, this value will be used as the **API_KEY**.



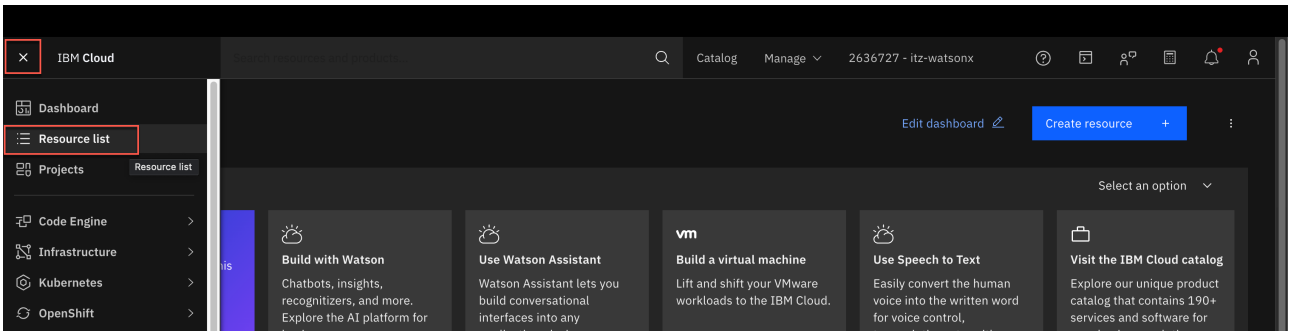
6. Click on the **X** button to close the API key window.



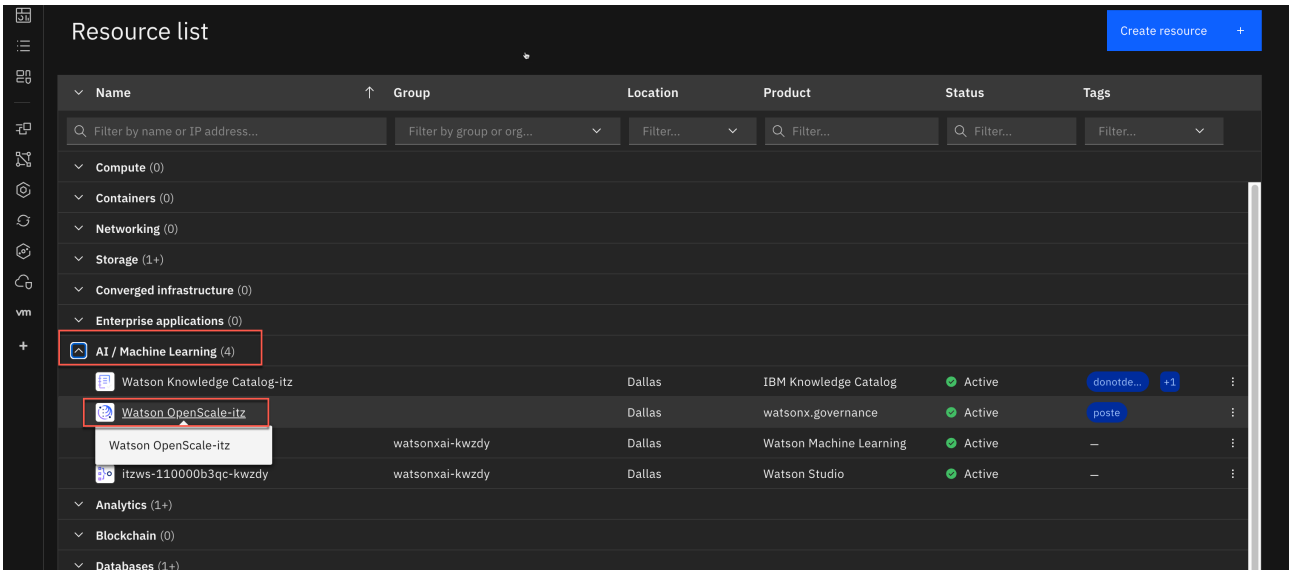
You now have the credentials necessary to proceed with the lab. In the next step, you will sign into the watsonx platform.

2.3 Sign into watsonx

1. Click on the hamburger menu in the upper left to open the menu, then click on the **Resource list** menu item. The **Resource list** page opens.

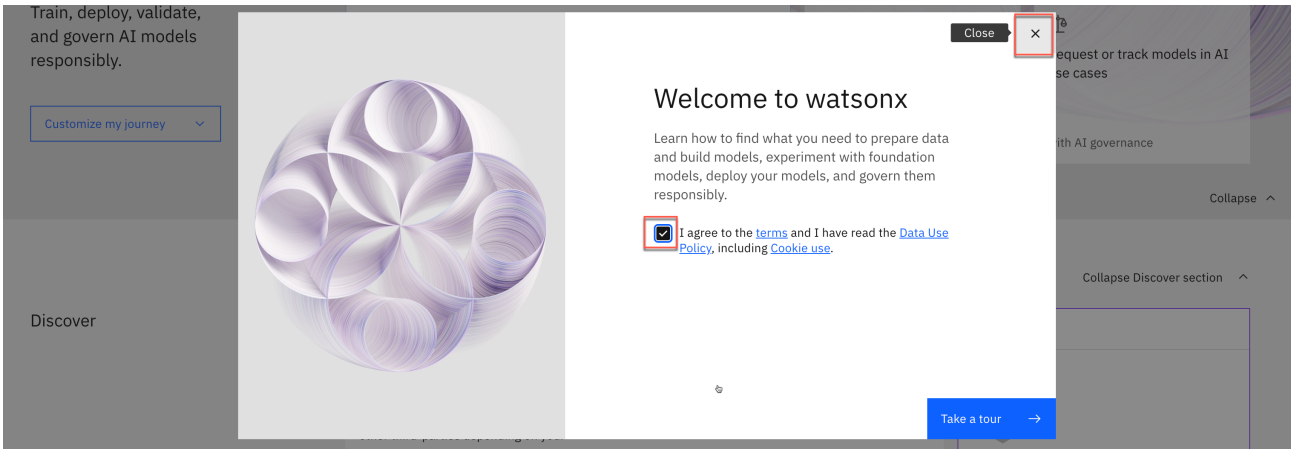


2. Locate the **AI / Machine Learning** item from the list and click on it to expand it. Locate the **Watson OpenScale** item in the expanded section and click on it. Note that the name may differ slightly from the one in the screenshot below.



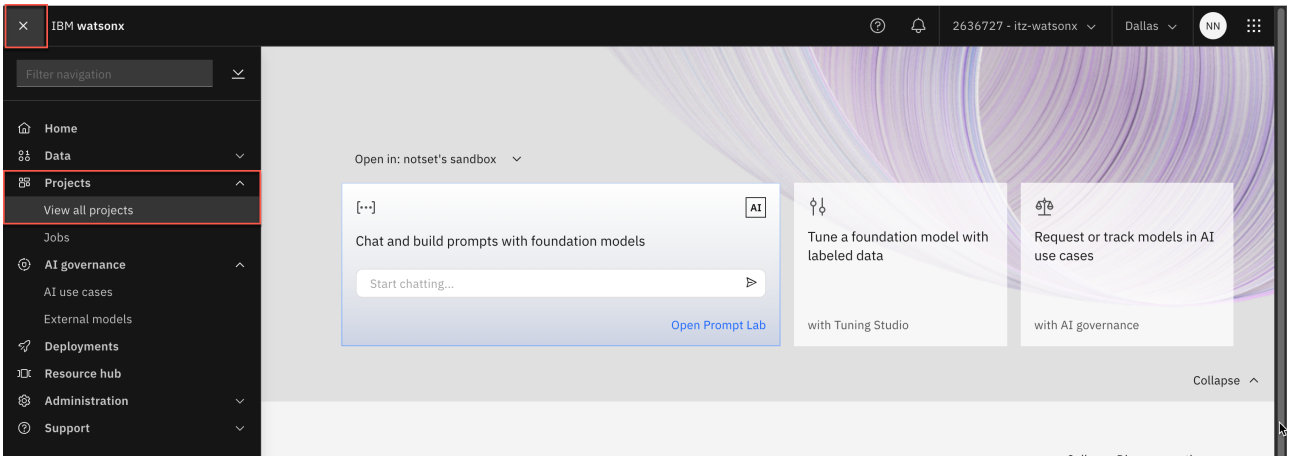
3. Click on the **Launch watsonx.governance** button. If you are asked to provide any additional information, wait a moment, then click the **Continue** button when it becomes enabled. Note that you may need to wait for up to two minutes for the watsonx.governance services to be prepared.

4. Read the terms and policies if you wish, then check the box to agree, and click the **X** button to close the popup window without taking the tour.



2.4 Create a watsonx project

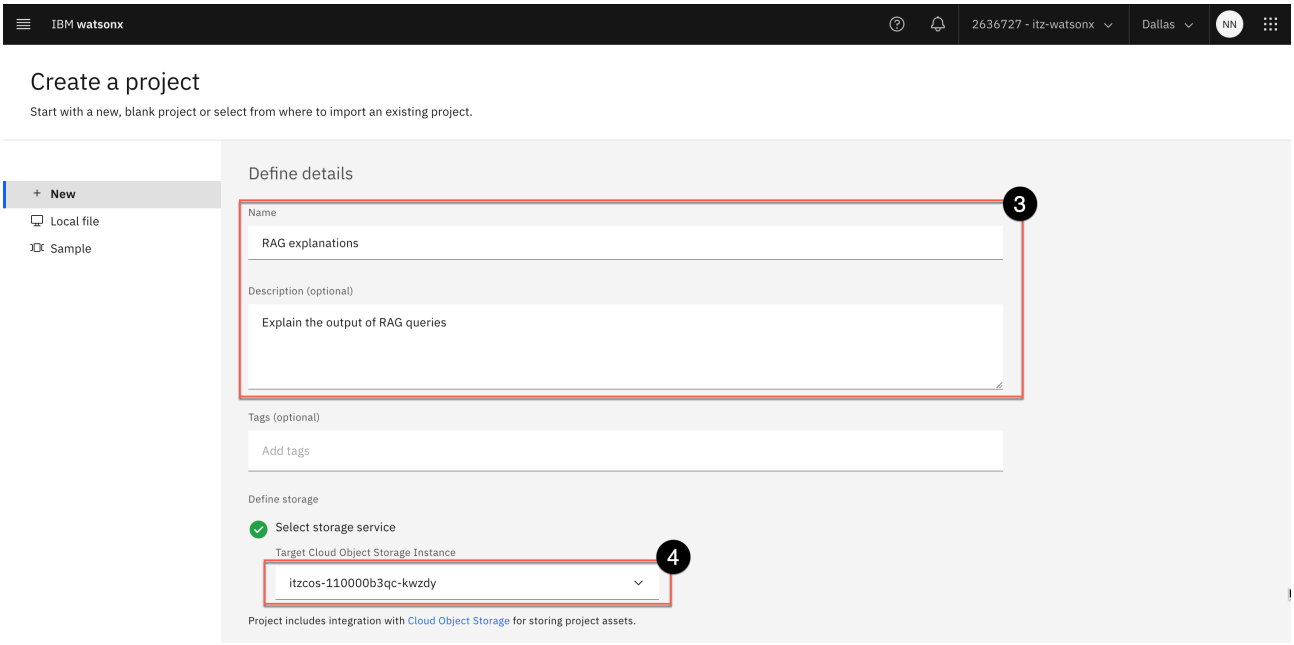
1. Click on the **hamburger menu** in the upper left to expand it, then click on the **Projects** item to expand it. Finally, click on the **View all projects** menu item.



2. Click the **New project** button to create a new project. The **Create a project** screen opens.

3. Fill out the project details, including a **Name** and optional **Description**.

4. Use the **Select storage service** dropdown to select the cloud object storage instance for your environment. If you have multiple options, select the one that most resembles the one in the screenshot below.



5. Click the **Create** button to create your project.

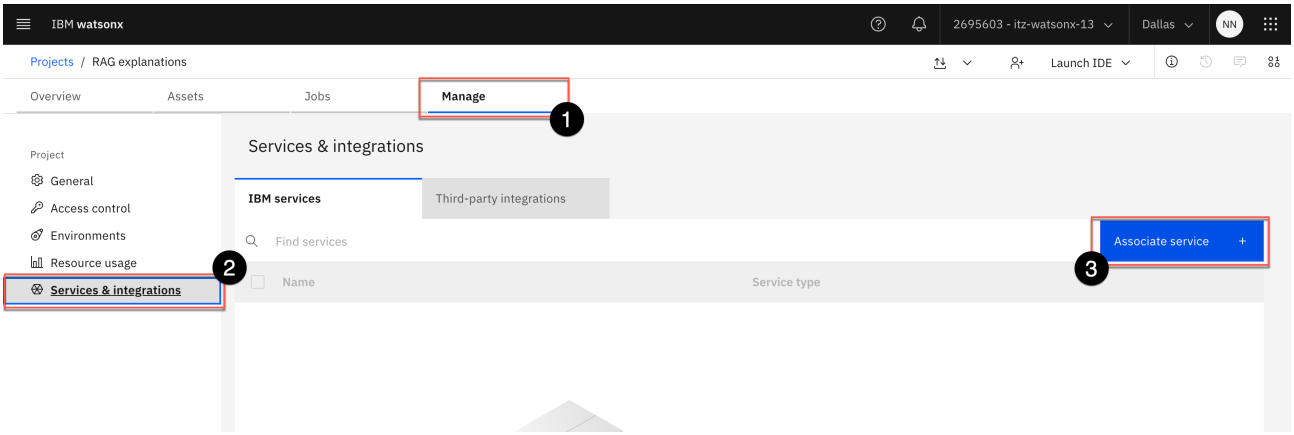
2.5 Associate a machine learning service

For project assets such as Jupyter notebooks to access the IBM Foundation models to perform RAG queries, you must associate a machine learning service with the project. Your lab environment comes with a provisioned machine learning service.

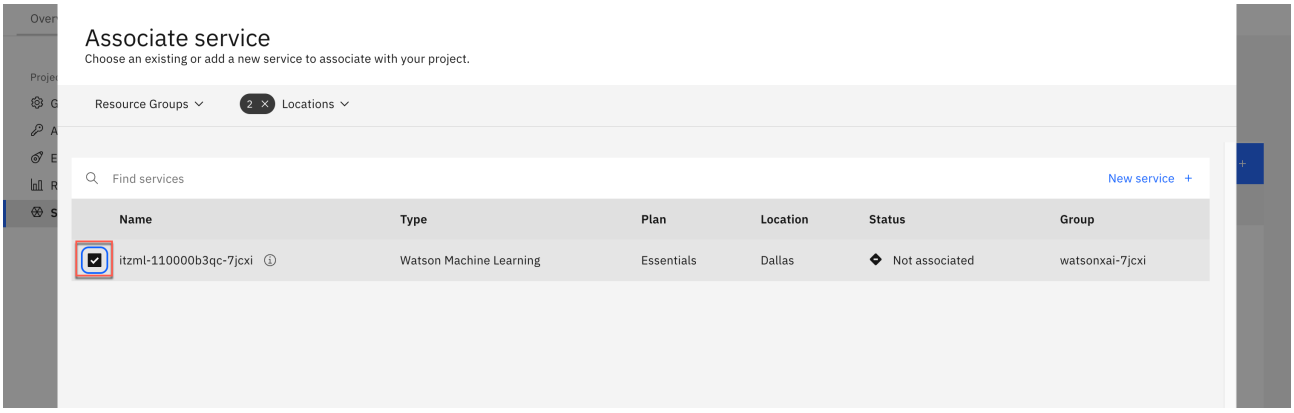
1. From your project screen, click on the **Manage** tab.

2. From the list on the left, click on the **Services & integrations** menu item.

3. Click the blue **Associate service** button. The **Associate service** window opens.



4. Locate and check the box to the left of the Watson Machine Learning service in the list.

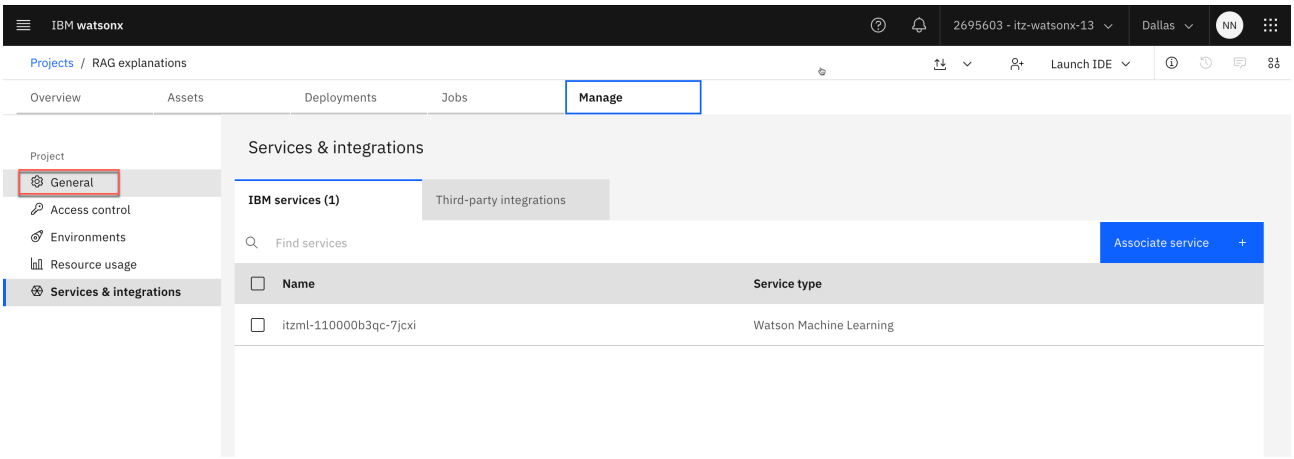


5. Click the **Associate** button to associate the service.

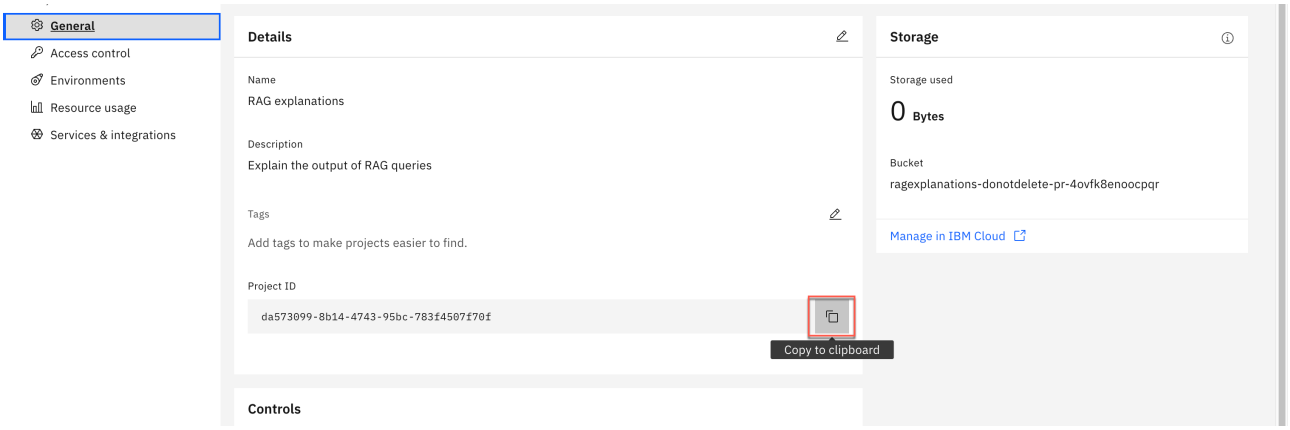
2.6 Get the project ID

Getting the ID of the project will allow you to query watsonx.ai Foundation models using the service you just associated.

1. Beneath the **Project** header on the left, click on the **General** menu item.



2. In the **Details** tile, locate the **Project ID** item and click the **Copy to clipboard** icon to copy the value. Paste it into the text file you have been using to store credentials. In the Jupyter notebook, this value will be the **PROJECT_ID**.



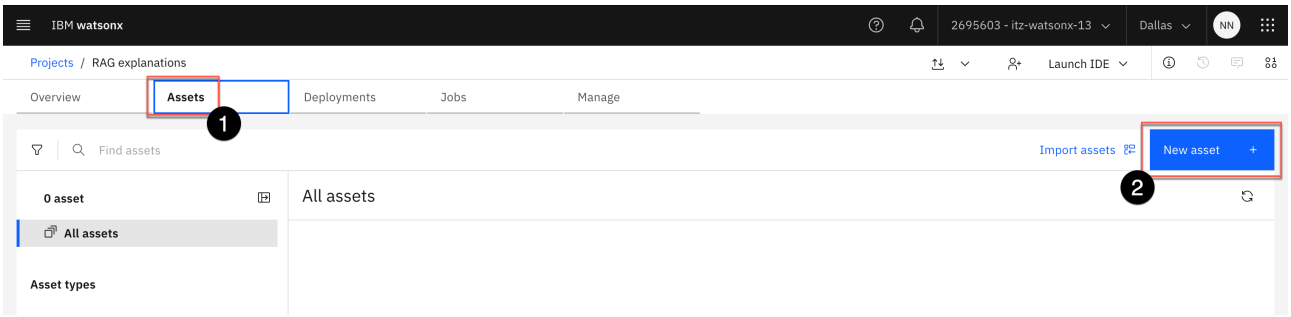
Your environment is now fully configured, and you have all of the credentials necessary to access Foundation models. In the next step, you will create a Jupyter notebook to query the models.

3 Executing the Jupyter notebook

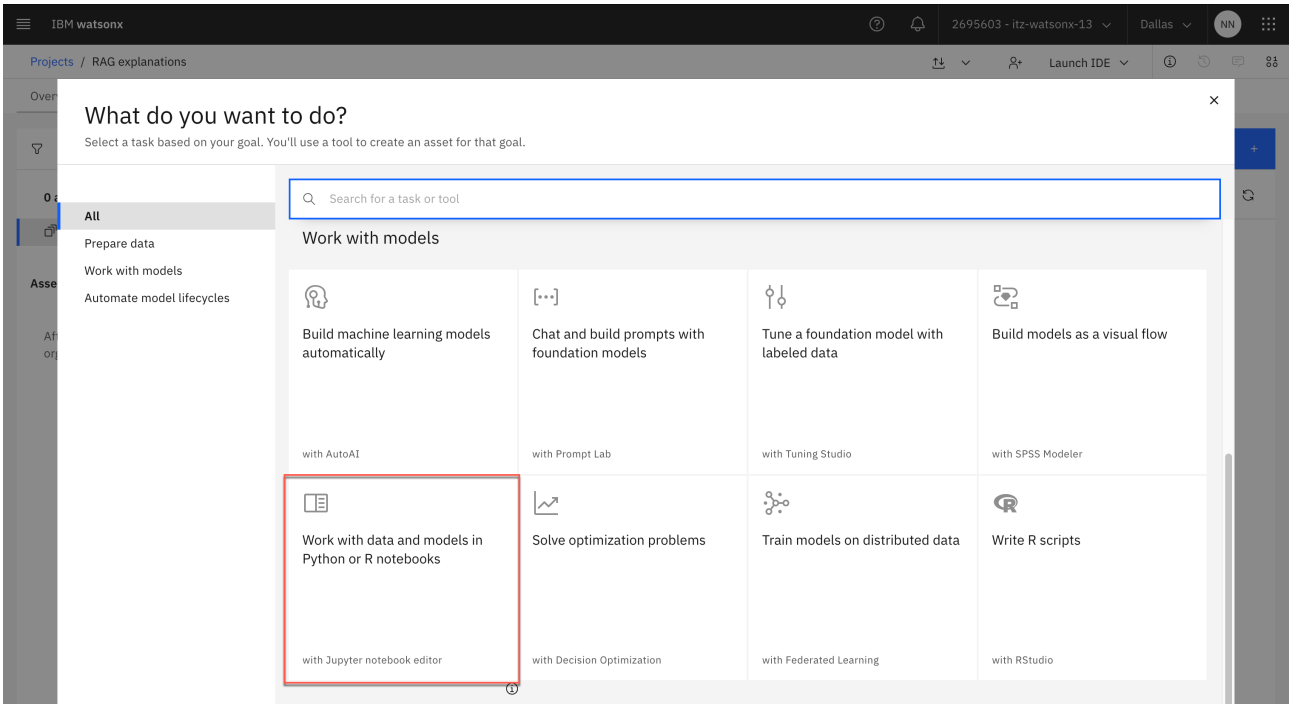
In this section of the lab, you will load a Jupyter notebook from a GitHub repository, input your credentials, and run it.

3.1 Create the notebook

1. Click on the **Assets** tab of your project.
2. Click on the **New asset** button. The **What do you want to do?** window opens.

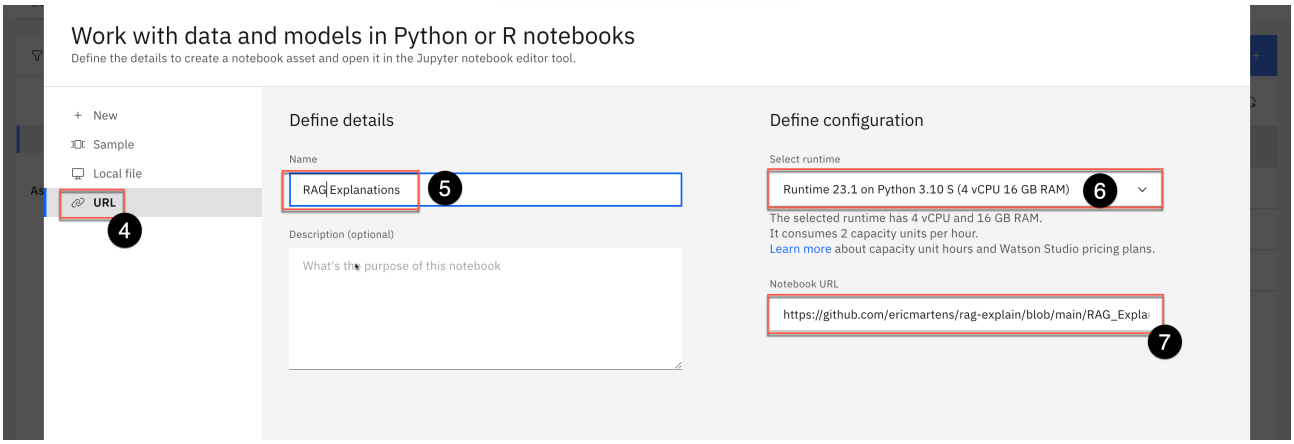


3. Scroll down to the **Work with models** section and click on the **Work with data and models in Python or R notebooks** tile.



4. Click on the **URL** menu option on the right.
5. Give your notebook a name such as **RAG Explanations**.
6. Click on the **Select runtime** dropdown and select **Runtime 23.1 on Python 3.10 S...** from the list.
7. Copy and paste the following value into the **Notebook URL** field:

```
https://github.com/ericmartens/rag-explain/blob/main/RAG_Explanations_TXC_2024.ipynb
```



8. Click the **Create** button. The notebook will created in your project, which can take up to two minutes to complete. Note that if the notebook creation step is taking too long, refreshing the page can often fix the problem.

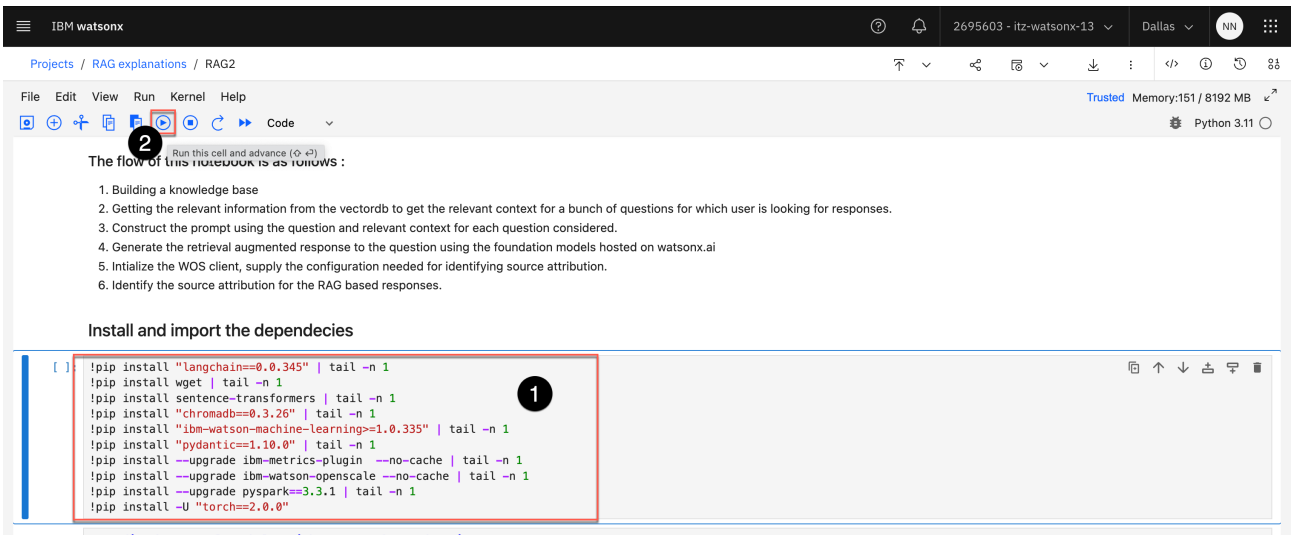
3.2 Install the necessary libraries in the notebook environment

Once the notebook has been created and appears on your screen, you must install additional libraries into the environment.

1. Scroll down, locate and click on the first code cell beneath the **Install and import dependencies** header. A blue highlight box will appear around the active cell.

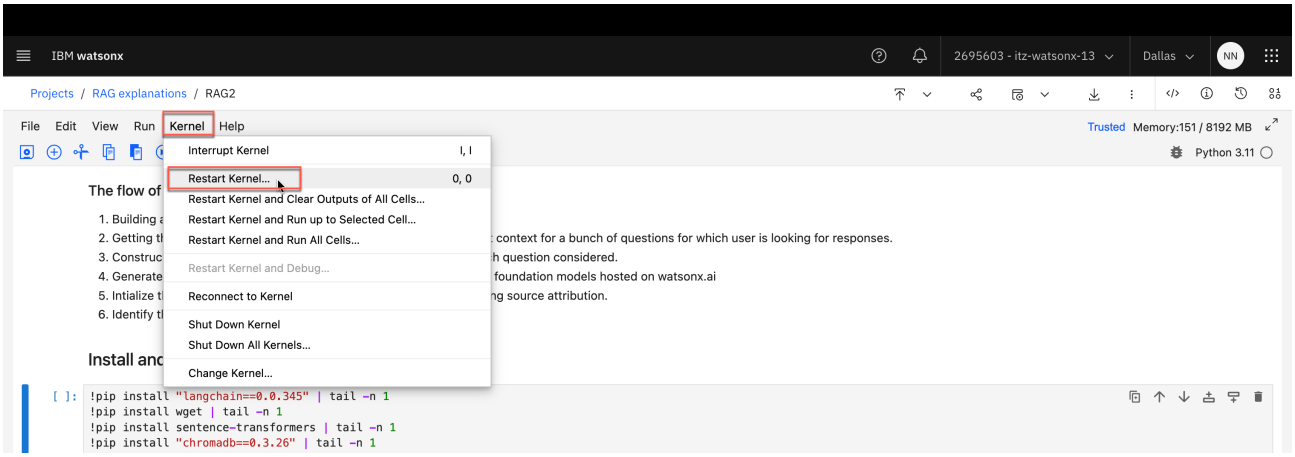
2. Run the cell by clicking the **Run** button from the notebook menu, or by using the **shift + return** keyboard shortcut. The notebook will use the pip utility to install a variety of libraries into the environment for the notebook to use. Note that this cell may generate some warning and error messages; these can be ignored.

   As the cell is running, an asterisk (*) will appear in the box to the left of the code in the cell. When the install process is finished, the asterisk will be replaced by a number, indicating that the cell has completed.



3. When the cell has completed running, you will need to restart the kernel to ensure that the environment is updated. Click on the **Kernel** menu item to open it, and then click on the **Restart kernel** menu option. A

warning notification window will ask you to confirm your choice. Click the **Restart** button to confirm. The environment's kernel will restart.



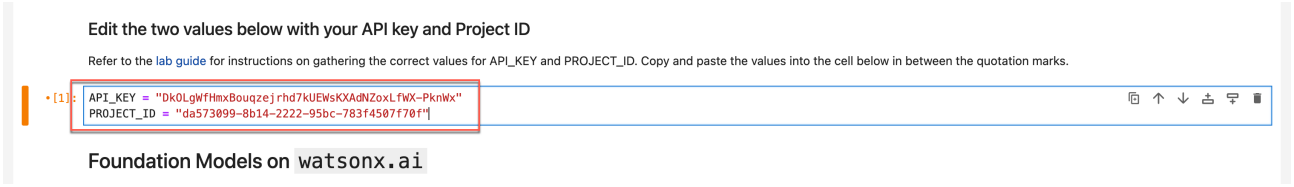
4. As the kernel restarts, a **Kernel restarting** status message will display in the upper right of the notebook screen. When the restart finishes, this status will disappear, and you may proceed with the rest of the lab.

3.3 Update the notebook with your credentials

When the kernel finishes restarting, you will need to add the API_KEY and PROJECT_ID credentials you gathered in previous steps.

1. Locate and click on the first code cell beneath the **Edit the two values below...** cell to make it active.

2. Copy and paste the **API_KEY** and **PROJECT_ID** values that you obtained in previous steps into the cell between the quotation marks on each line. When you are finished, your cell should resemble the one in the screenshot below:



Run the notebook

At this point, you may run through the notebook one cell at a time, beginning with the credentials cell you just edited. Notebook cells can be run by clicking on the **Run** button from the menu, or by using the **shift + return** keyboard shortcut. Note that some warning messages may appear; these are normal and can be ignored.

When you reach the **Custom information** section of the notebook, you may return to this guide for information on adding your own data to the project for use in the notebook.

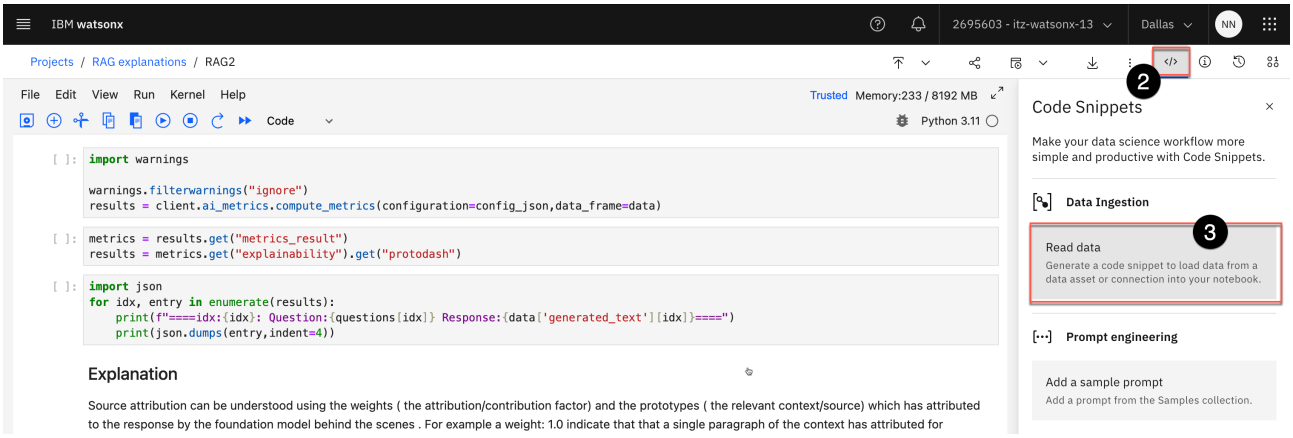
3.4 Adding and querying your own text

It is easy to add your own custom text for queries and explanations. Prior to completing this step, you will need to save the text to a file on your machine so that it can be imported into the notebook.

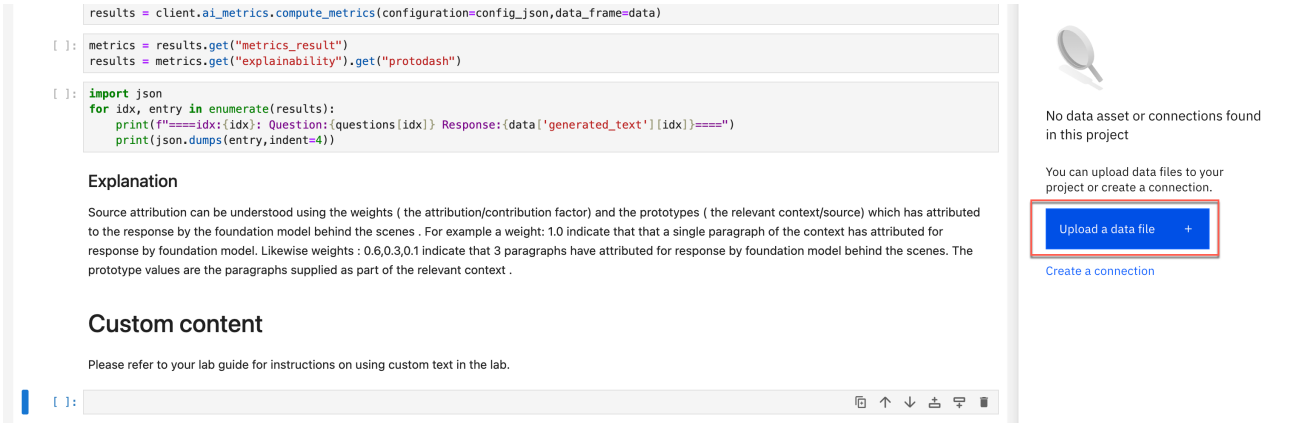
1. Click on the empty code cell beneath the **Custom content** title cell. Your cursor should appear in the empty code cell, and a blue highlight box will appear around the cell.

2. Click on the **Code snippets** button on the top right of the screen to open the **Code snippets** panel.

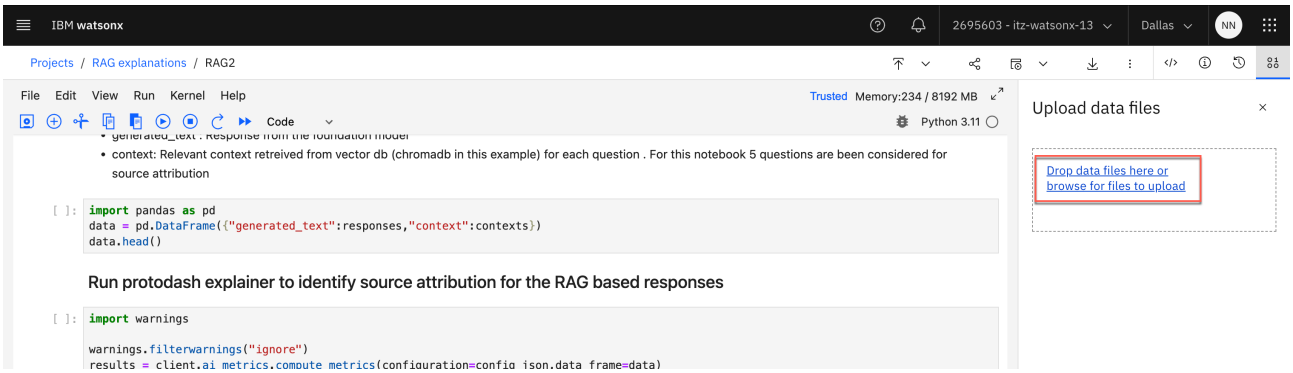
3. Click on the **Read data** tile.



4. Click on the **Upload a data file** button.

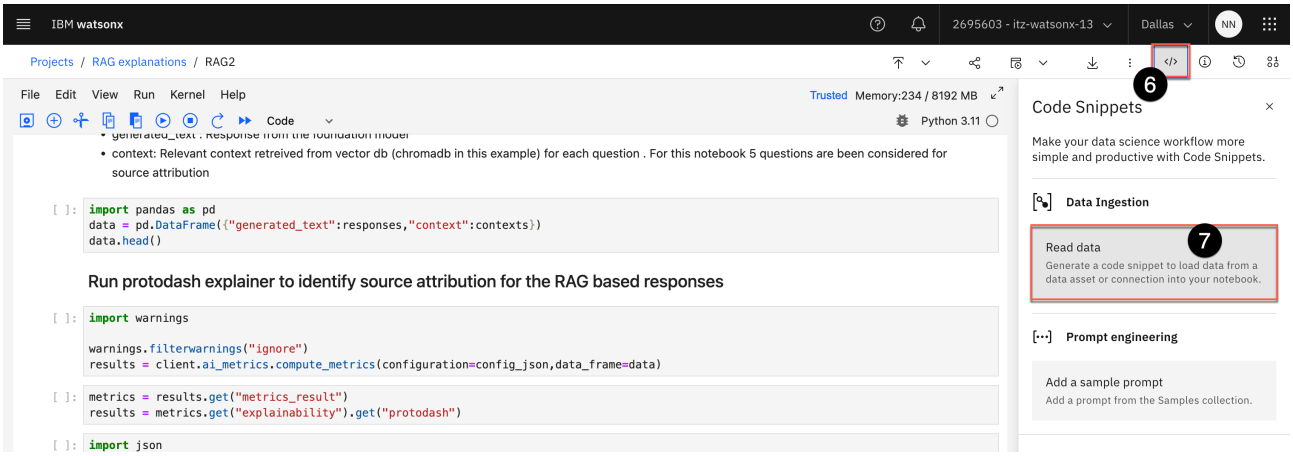


5. Click on the **Drop data files here...** link to browse to a file on your machine, and select the custom text you wish to work with.

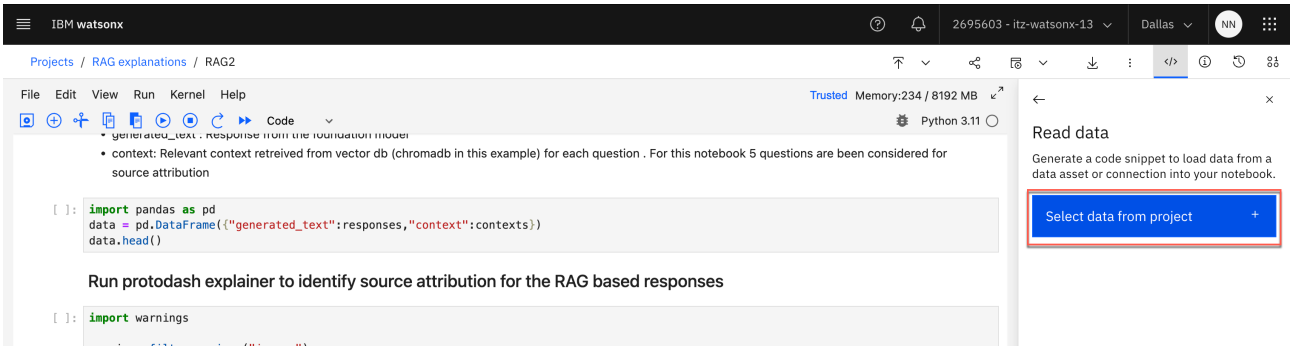


6. Once the file has finished uploading, it has been added as an asset in your project, and can be used by the notebook. Click on the **Code snippets** button once again.

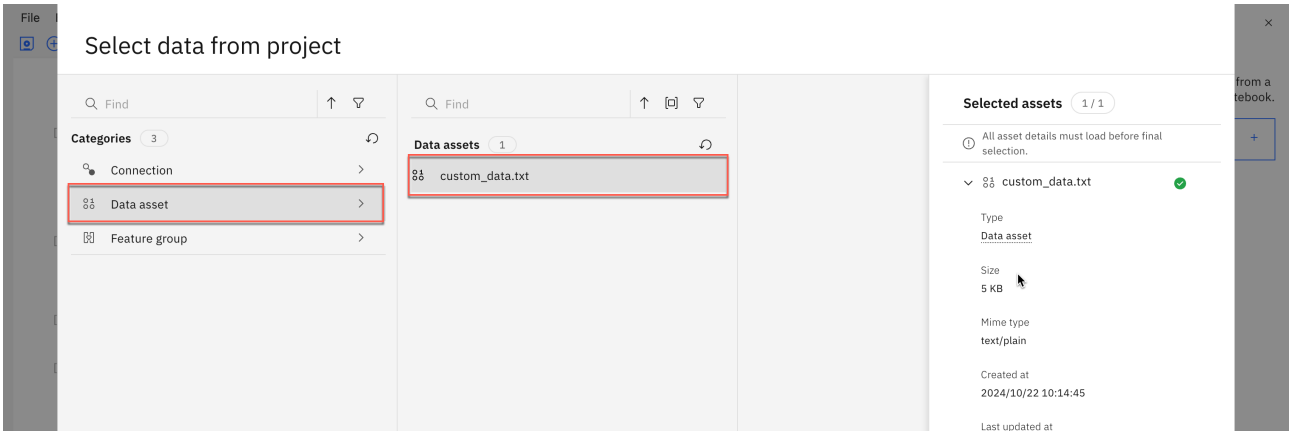
7. Click on the **Read data** tile.



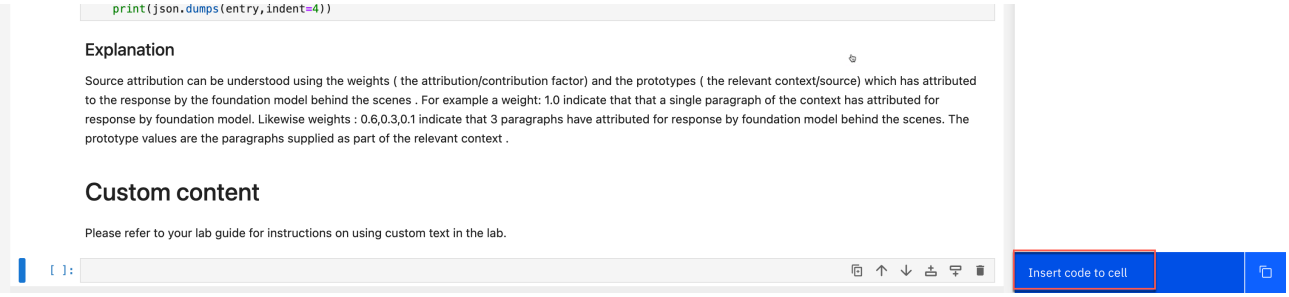
8. Click on the **Select data from project** button. The **Select data from project** window opens.



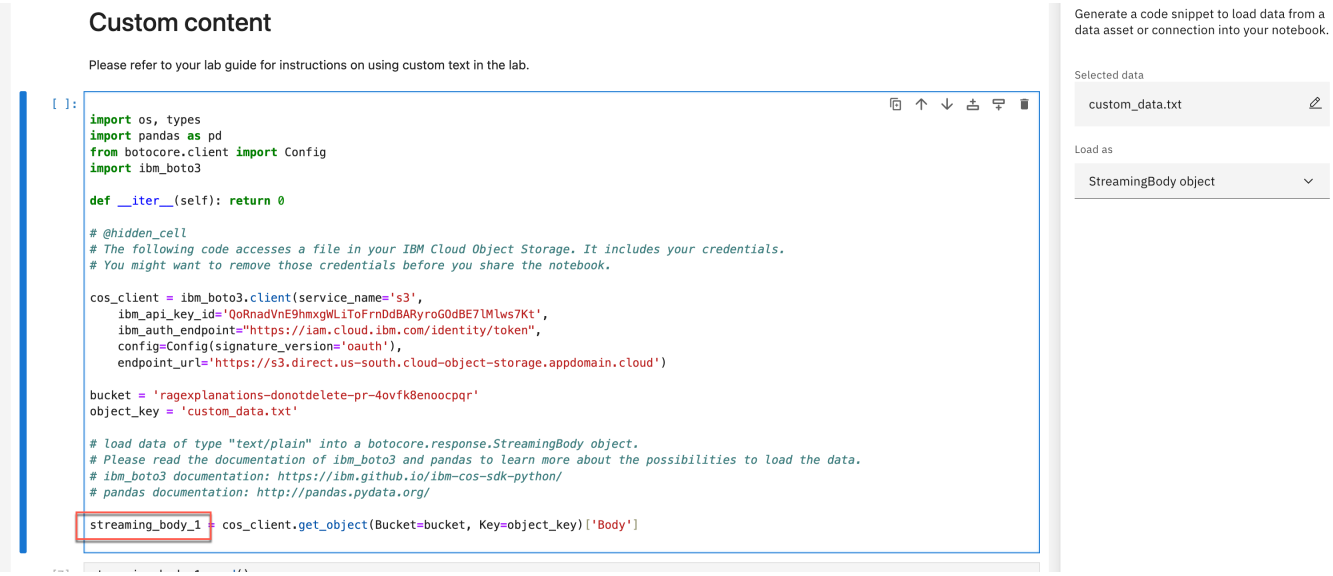
9. Click on the **Data asset** category, then click on the file you imported in a previous step and click on the **Select** button.



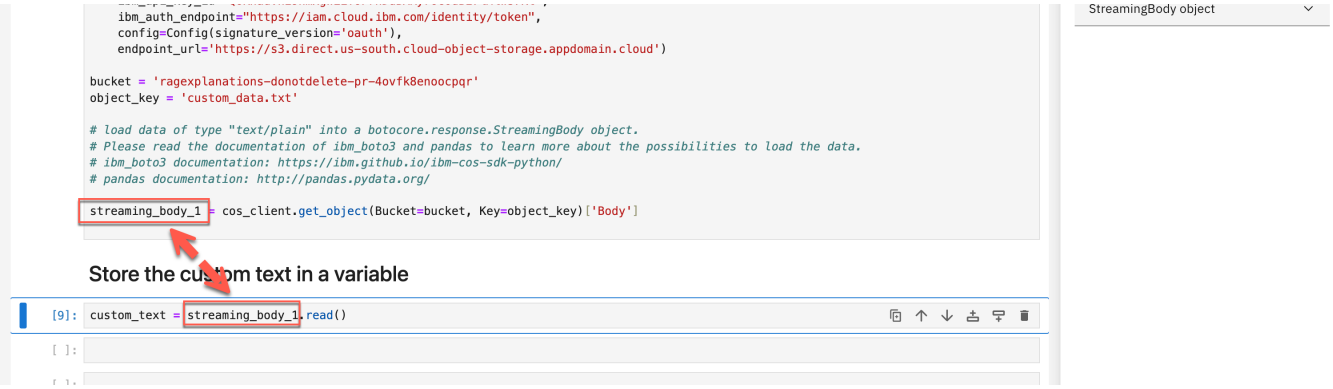
10. Click on the **Insert code to cell** button.



11. The notebook will create a code snippet to import the contents of the text file as a StreamingBody object into to the notebook. Run the cell to execute it, making note of the variable name used to hold the contents of the file.



12. Note that each time you use the code snippet to insert a new set of text, the variable name will increment (*streaming_body_1, streaming_body_2…*). You will need to ensure that the variable used in the next code cell is updated to match the variable used to store the imported data.



13. You may now proceed with the rest of the notebook, returning to the custom text loading cell to load new files if you wish.

4 Congratulations

You have explored the RAG explanation capabilities of watsonx.ai and watsonx.governance. Thank you for attending this session, and enjoy the remainer of your conference.