

How to setup Qiskit in IBM Watson Studio

Contents

**3 Set up Watson
Studio Free Trial**

**4 Create a Watson
Studio project**

**12 Create a blank
notebook**

**15 Install Qiskit
Runtime**

**19 Upload an existing
notebook**

**26 Connect your IBM
Quantum account**

**28 Upgrade to a paid
account**

Set up Watson Studio Free Trial

Create a free trial in your region

Go to [Watson Studio landing page](#) to create a free trial.

The free trial has 10 hours of compute / month available.

You can upgrade to a paid service at \$1.02 / hour of usage.

On this page

1. Select your region
2. Review and accept the terms, Data Use Policy, and Cookie use policy

IBM Watson Studio

Try IBM Watson Studio for free

Get started with a FREE trial of IBM Watson Studio on Cloud Pak for Data. See for yourself how IBM can transform your business into a data driven enterprise.

Select a region

Select the region closest to you or where you plan on hosting your data and services. If you have an existing data store, select that region.

Dallas (us-south) ▾

Services available in Dallas (us-south):

Featured services

- Watson Machine Learning
Deploy ML Models
- Watson Studio
Build and manage ML models

Additional services

- DataStage
Create data pipelines
- IBM Match 360 with Watson
Build customer profiles
- IBM Knowledge Catalog
Catalog and govern data
- watsonx.governance
Trust AI models
- Watson Query
Query data anywhere

I agree to the [terms](#) and I have read the [Data Use Policy](#), including [Cookie use](#).

Create account or log in

“ This is exactly what the data fabric promises, active metadata driving what we

Create an IBM Cloud account

After selecting your region, you can create a new IBM Cloud account, or login to an existing one.

We would recommend using your organization email address, if possible.

On this page

1. Enter your email
2. Verify your email with a code
3. Complete demographic questions
4. Review account notices

IBM Cloud

Catalog Cost estimator Docs

IBM Cloud

Create an IBM Cloud account

Already have an IBM Cloud account? [Log in](#)

Account information

Email

Hide My Email...
Create a unique, random address that forwards to your inbox.

Next ↓

Verify email

Personal information

Account notice

Continue →

Build for free on IBM Watson Studio

Join us in the cloud and start building today.

Collaboration for your entire data science team
Invite team members from across your organization to contribute data and work with machine and deep learning models in collaborative projects.

Code or no-code deep learning model building
Use drag-and-drop tooling from SPSS Modeler and Neural Network Modeler for visual model building and refinement.

Manage environments easily in the IBM Cloud
Test and deploy models using customizable compute environments, including Anaconda, Spark and GPU environments.

Build your machine learning model on us
Get started with a perpetual, cap-based Lite plan - with no time restrictions.

Finish signup process

If you're setting up your own account, use the default information here.

Else, ask your organization's IBM Cloud administrator to join your organization's account.

On this page

1. Review your selected account and resource group
2. Add your company name
3. Add your phone number

The screenshot shows the 'Provide your information to continue' page in IBM Watson Studio. The page has a dark background with white text. At the top left, it says 'IBM Watson Studio'. The main heading is 'Provide your information to continue'. Below this, there are several form fields: 'User account' with a dropdown menu showing 'Sanket Panda's Account', 'Resource group' with a dropdown menu showing 'Default', 'Company name' with an empty text input field, and 'Phone number' with a dropdown menu showing '+1' and an empty text input field. At the bottom left, there is a 'Continue' button. To the right of the form fields, there is a paragraph of text: 'IBM may use my contact data to keep me informed of products, services, and offerings: You can withdraw your marketing consent at any time by submitting an [opt-out request](#). Also, you may unsubscribe from receiving marketing emails by clicking the unsubscribe link in each email. More information on our processing can be found in the [IBM Privacy Statement](#).' Below this, there is another paragraph: 'By submitting this form, I acknowledge that I have read and understand the IBM Privacy Statement and I accept the product [Terms and Conditions](#) of this registration form.'

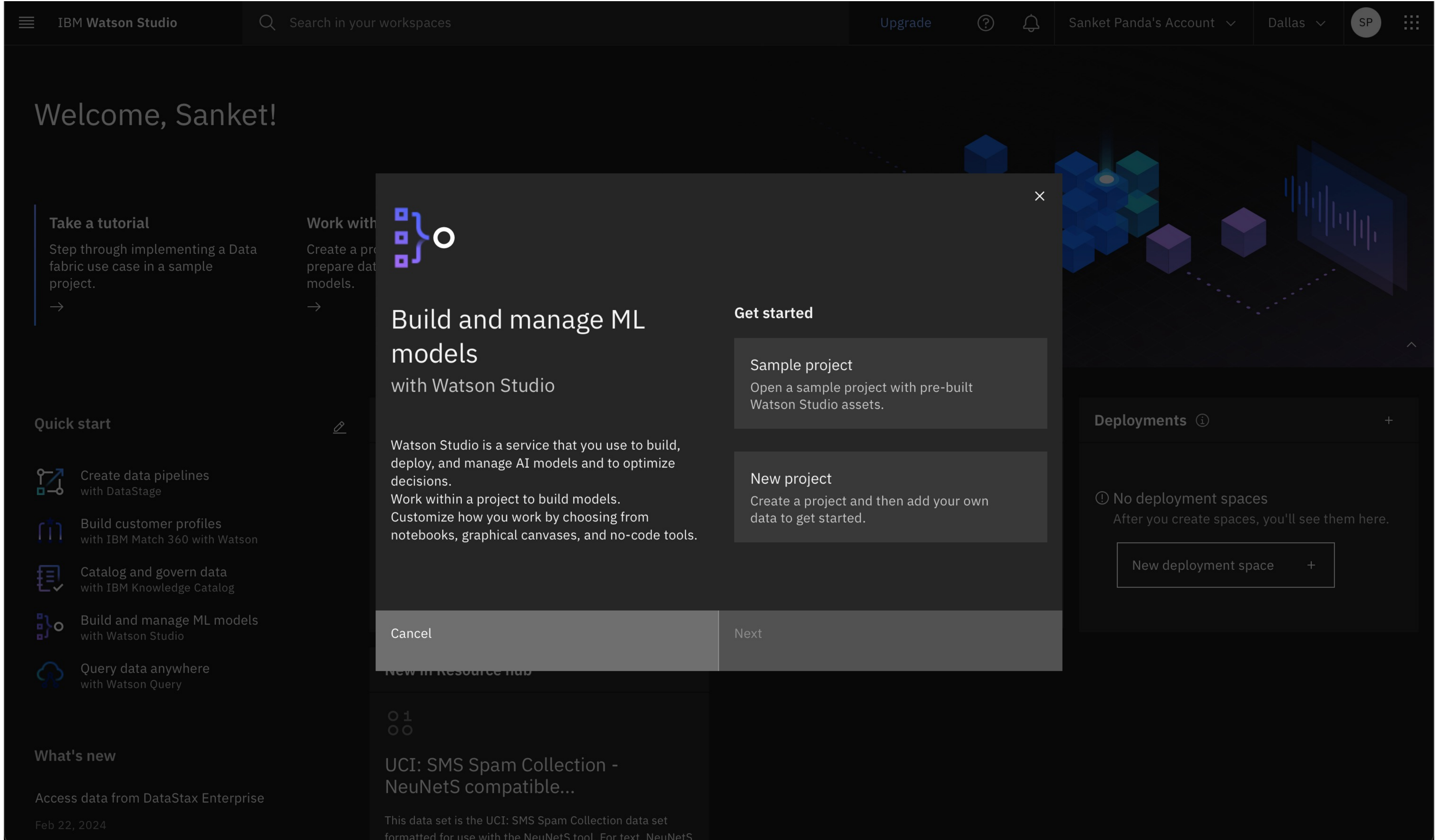
Create a Watson Studio project

Create a new project

Now that you're logged in, you can create a new Watson Studio project with Qiskit

On this page

- 1. Click "New Project"
- 2. Continue



Add project details

We'll create a blank project for this example. We'll add metadata to help identify what you're storing in this project.

This could be a generic “playground” name, or specific projects for a class.

On this page

1. Give your project a name and description
2. Continue

IBM Watson Studio

Search in your workspaces

Upgrade

Sanket Panda's Account

Dallas

SP

Create a project

Start with a new, blank project or select from where to import an existing project.

- + New
- Local file
- Resource hub

Define details

Name

Qiskit Demo

Description (optional)

Installing Qiskit and Qiskit Runtime in IBM Watson Studio

Storage

CloudObjectStorage

Project includes integration with [Cloud Object Storage](#) for storing project assets.

Advanced settings

Cancel

Create

Navigating your project

Watson Studio has a few core concepts to learn.

Assets can refer to notebooks, or data that you connect or upload.

Jobs refer to scheduling an automated run of a notebook.

Manage refers to access controls, budgeting, and integrations.

On this page

1. Click Assets

The screenshot shows the IBM Watson Studio interface for a project named "Qiskit Demo". The top navigation bar includes the IBM Watson Studio logo, a search bar, an "Upgrade" button, a help icon, a notification bell, the user's account name "Sanket Panda's Account", the location "Dallas", and a profile icon. Below the navigation bar, the breadcrumb "Projects / Qiskit Demo" is visible, along with icons for refresh, share, launch IDE, and other actions. The main content area is divided into four tabs: "Overview" (selected), "Assets", "Jobs", and "Manage". Under the "Overview" tab, there are three main sections: "Assets" (showing a bar chart icon and a "View all" link), "Resource usage" (displaying "0 CUH" for the current month), and "Project history" (showing a log entry: "You created project Qiskit Demo Today at 11:16 PM"). A "Readme" section is also present, with a placeholder text "Type project notes, reminders, or instructions".

Creating a blank notebook

Assets refer to notebooks and uploaded data.

For this demo, we'll create a blank notebook and install Qiskit Runtime.

On this page

1. Click "New asset"

IBM Watson Studio Search in your workspaces Upgrade ? Sanket Panda's Account Dallas SP

Projects / Qiskit Demo

Overview **Assets** Jobs Manage

Find assets Import assets New asset +

0 asset

All assets

Asset types

After you create assets, they are organized by asset type.

Start adding assets

To get started with assets, click **New asset** to create one in a tool, or **Import assets** to add existing ones.

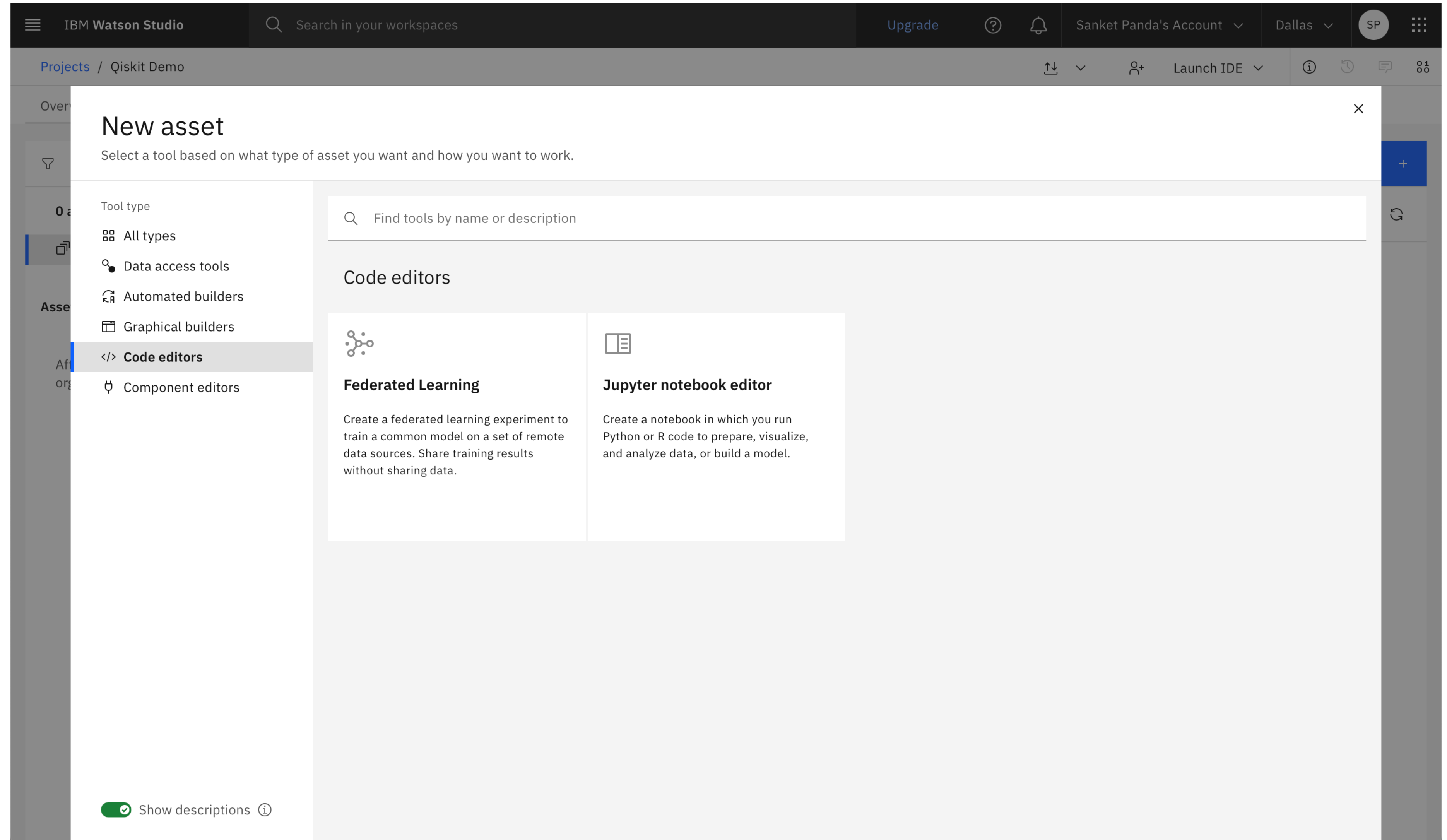
Create a blank notebook

Creating a blank notebook, part two

We'll create a new Jupyter notebook within the asset creator.

On this page

1. Click “Code editors”
2. Click “Jupyter notebook editor”



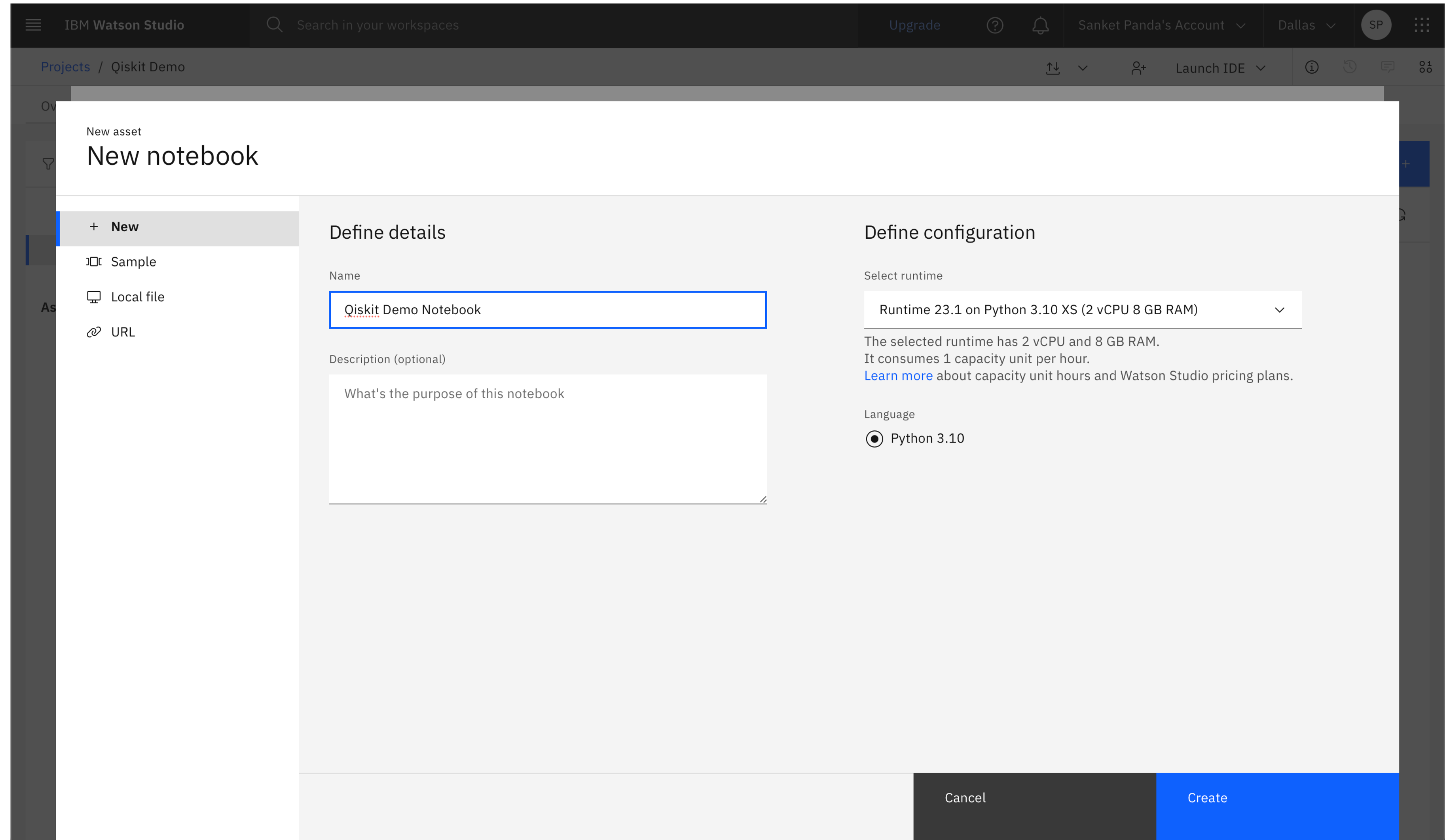
Creating a blank notebook, part three

We'll give our notebook a name and description.

Additionally, we'll setup a default Python environment. Generally, select the latest Runtime and latest Python.

On this page

1. Add a notebook name and description
2. Select a notebook runtime
3. Select a language



Install Qiskit Runtime

Install required packages

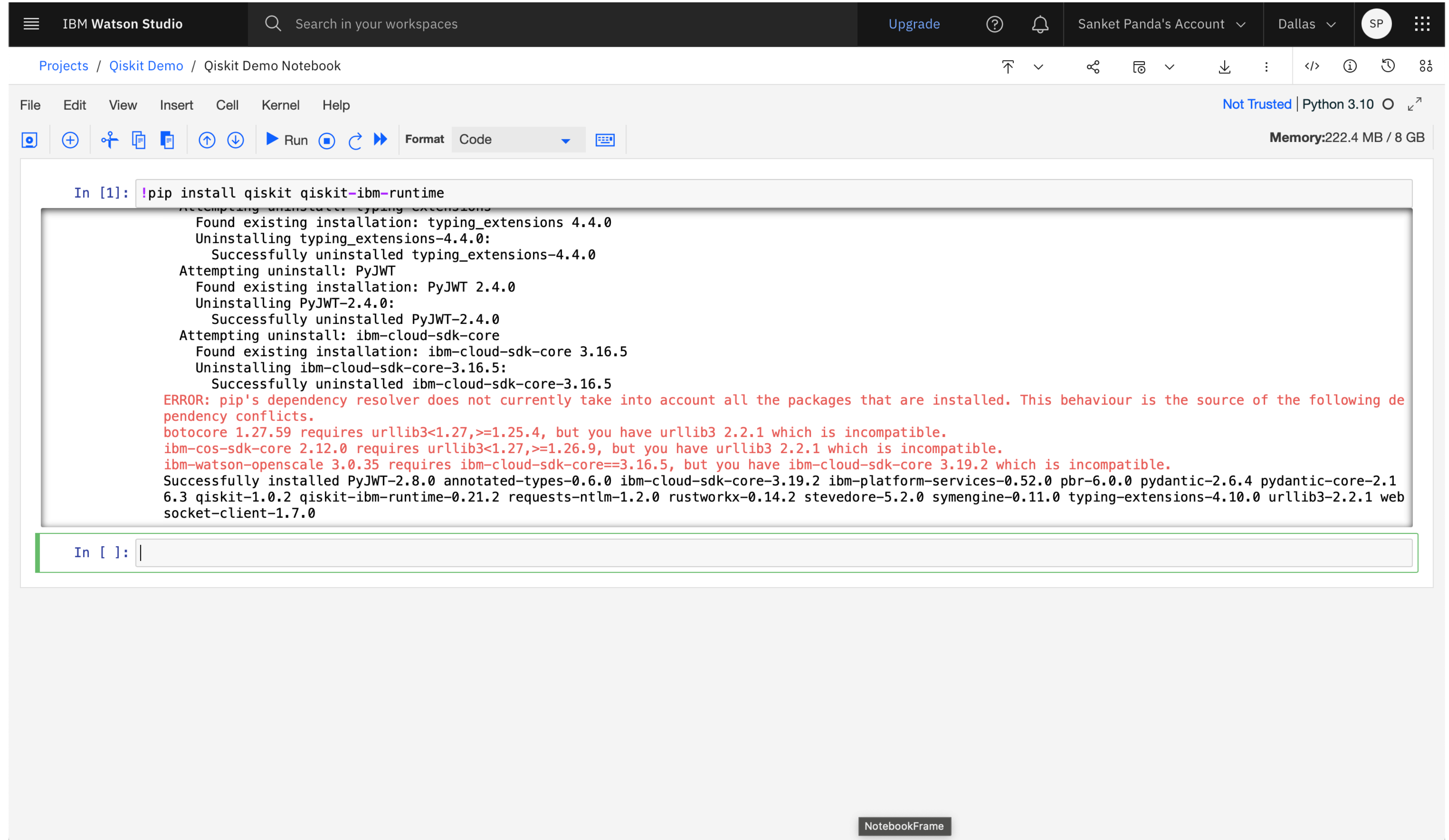
We'll install the basic packages needed for Qiskit Runtime. This includes the following command:

```
!pip install qiskit
qiskit-ibm-runtime
pylatexenc
```

```
!yes | pip uninstall
simplejson
```

On this page

1. Add the above command into an empty code cell
2. Click Run or CMD + Enter



The screenshot shows the IBM Watson Studio interface. The top navigation bar includes 'IBM Watson Studio', a search bar, an 'Upgrade' button, and user information for 'Sanket Panda's Account' in 'Dallas'. The notebook title is 'Qiskit Demo Notebook'. The toolbar contains various icons for file operations and a 'Run' button. The code cell contains the command: `!pip install qiskit qiskit-ibm-runtime`. The output shows the installation process, including the uninstallation of existing packages like `typing_extensions`, `PyJWT`, and `ibm-cloud-sdk-core`. A red error message states: `ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.` The error lists conflicts for `botocore`, `ibm-cos-sdk-core`, and `ibm-watson-openscale`. The final output lists the successfully installed packages: `PyJWT-2.8.0`, `annotated-types-0.6.0`, `ibm-cloud-sdk-core-3.19.2`, `ibm-platform-services-0.52.0`, `pbr-6.0.0`, `pydantic-2.6.4`, `pydantic-core-2.16.3`, `qiskit-1.0.2`, `qiskit-ibm-runtime-0.21.2`, `requests-ntlm-1.2.0`, `rustworkx-0.14.2`, `stevedore-5.2.0`, `symengine-0.11.0`, `typing-extensions-4.10.0`, `urllib3-2.2.1`, and `websocket-client-1.7.0`. Below the code cell is an empty input field for the next command.

Verify Qiskit installation

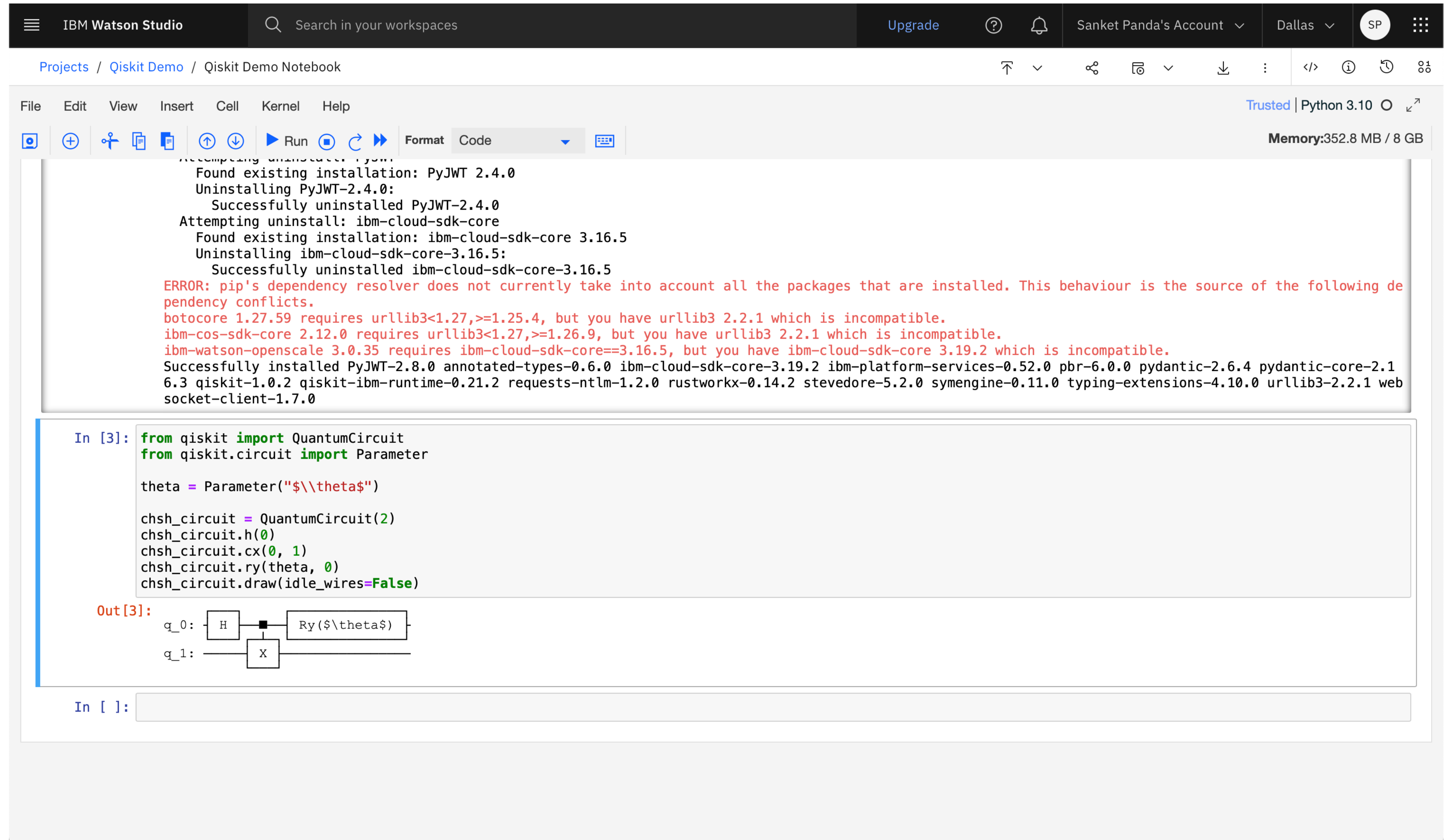
We'll test if Qiskit was successfully installed with the following code:

```
from qiskit import QuantumCircuit
from qiskit.circuit import Parameter
```

```
theta = Parameter("$\\theta$")
chsh_circuit = QuantumCircuit(2)
chsh_circuit.h(0)
chsh_circuit.cx(0, 1)
chsh_circuit.ry(theta, 0)
chsh_circuit.draw(idle_wires=False)
```

On this page

1. Add the above command into an empty code cell
2. Click Run or CMD + Enter



The screenshot shows the IBM Watson Studio interface. The top navigation bar includes 'IBM Watson Studio', a search bar, and user information. The main area displays a Jupyter Notebook with a code cell and its output. The code cell contains the same Python code as shown in the previous blocks. The output shows the installation of PyJWT and other packages, followed by a dependency conflict error. Below the error, a quantum circuit diagram is displayed, showing two qubits (q_0 and q_1) with H, X, and Ry gates.

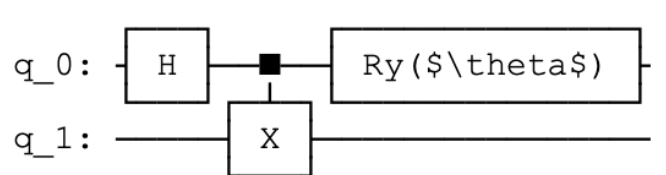
```
recompiling uninstalled pyjwt
Found existing installation: PyJWT 2.4.0
Uninstalling PyJWT-2.4.0:
  Successfully uninstalled PyJWT-2.4.0
Attempting uninstall: ibm-cloud-sdk-core
Found existing installation: ibm-cloud-sdk-core 3.16.5
Uninstalling ibm-cloud-sdk-core-3.16.5:
  Successfully uninstalled ibm-cloud-sdk-core-3.16.5
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
botocore 1.27.59 requires urllib3<1.27,>=1.25.4, but you have urllib3 2.2.1 which is incompatible.
ibm-cos-sdk-core 2.12.0 requires urllib3<1.27,>=1.26.9, but you have urllib3 2.2.1 which is incompatible.
ibm-watson-openscale 3.0.35 requires ibm-cloud-sdk-core==3.16.5, but you have ibm-cloud-sdk-core 3.19.2 which is incompatible.
Successfully installed PyJWT-2.8.0 annotated-types-0.6.0 ibm-cloud-sdk-core-3.19.2 ibm-platform-services-0.52.0 pbr-6.0.0 pydantic-2.6.4 pydantic-core-2.16.3 qiskit-1.0.2 qiskit-ibm-runtime-0.21.2 requests-ntlm-1.2.0 rustworkx-0.14.2 stevedore-5.2.0 symengine-0.11.0 typing-extensions-4.10.0 urllib3-2.2.1 websocket-client-1.7.0
```

```
In [3]: from qiskit import QuantumCircuit
from qiskit.circuit import Parameter

theta = Parameter("$\\theta$")

chsh_circuit = QuantumCircuit(2)
chsh_circuit.h(0)
chsh_circuit.cx(0, 1)
chsh_circuit.ry(theta, 0)
chsh_circuit.draw(idle_wires=False)
```

Out[3]:



```
In [ ]:
```

Verify Qiskit installation

We'll test if Qiskit was successfully installed with the following code:

```
from qiskit import QuantumCircuit
from qiskit.circuit import Parameter
```

```
theta = Parameter("$\\theta$")
chsh_circuit = QuantumCircuit(2)
chsh_circuit.h(0)
chsh_circuit.cx(0, 1)
chsh_circuit.ry(theta, 0)
chsh_circuit.draw(idle_wires=False)
```

On this page

1. Add the above command into an empty code cell
2. Click Run or CMD + Enter

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and user information. Below that, the notebook title 'Qiskit Demo Notebook' is visible. The main area contains a code cell with the following Python code:

```
In [3]: from qiskit import QuantumCircuit
from qiskit.circuit import Parameter

theta = Parameter("$\\theta$")

chsh_circuit = QuantumCircuit(2)
chsh_circuit.h(0)
chsh_circuit.cx(0, 1)
chsh_circuit.ry(theta, 0)
chsh_circuit.draw(idle_wires=False)
```

The output of the code cell is a quantum circuit diagram for two qubits, q_0 and q_1. Qubit q_0 starts with an H gate, followed by a CNOT gate with q_1 as the control and q_0 as the target. Qubit q_1 starts with an X gate, followed by a Ry gate with parameter θ and q_1 as the target. The circuit is drawn with idle wires for q_0 and q_1.

```
Out[3]:
```

Below the code cell, there's an empty input field for the next code cell.

Upload an existing notebook

Uploading a tutorial from IBM Quantum Learning

We can download a notebook from [IBM Quantum Learning's tutorial catalog](#) and upload it to Watson Studio.

On this page

1. Click “Download notebook”

The screenshot shows the IBM Quantum Learning website interface. At the top, there is a navigation bar with 'IBM Quantum Learning' on the left and 'Home', 'Catalog', 'Composer', and 'Lab' in the center. On the right side of the navigation bar, there are icons for search, a grid, and a 'Sign in' link. Below the navigation bar, the page is divided into a left sidebar and a main content area. The sidebar contains a breadcrumb 'Tutorials / Variational quantum eigensolver' and a list of sections: 'Background' (highlighted), 'Setup', 'Step 1: Map classical inputs to a quantum problem', 'Step 2: Optimize problem for quantum execution.', 'ISA Circuit', 'ISA Observable', 'Step 3: Execute using Qiskit Primitives.', and 'Step 4: Post-process, return result in classical format.'. The main content area features the title 'Variational quantum eigensolver' with a clock icon and '28 mins' below it. To the right of the title are two buttons: 'Download notebook' with a download icon and 'Open in Quantum Lab' with an external link icon. Below these buttons, there are sections for 'Category' (Workflow example) and 'Topics' (Scheduling). The main content area also includes a 'Background' section with a paragraph of text and a list of three components required for executing a VQE algorithm: 1. Hamiltonian and ansatz (problem specification), 2. Qiskit Runtime estimator, and 3. Classical optimizer. Below this list is another paragraph of text. At the bottom of the main content area, there is a 'Setup' section with the text 'Here we import the tools needed for a VQE experiment.'

Return to Assets to upload a new notebook

Once you're back in your Project's Assets, we can create a "New asset"

On this page

1. Click "New asset"

The screenshot shows the IBM Watson Studio interface for a project named "Qiskit Demo". The "Assets" tab is selected, showing a list of assets. The table below lists the assets:

Name	Last modified
Qiskit Demo Notebook Notebook	2 minutes ago Modified by you

On the right side of the interface, there is a "Data in this project" panel. It displays a file named "variational-quantum-eigensolver.ipynb" with a green checkmark and the text "Uploaded!". A "Dismiss" button is located next to the file name.

Return to Assets to upload a new notebook

Once you're back in your Project's Assets, we can create a "New asset"

On this page

1. Click "New asset"

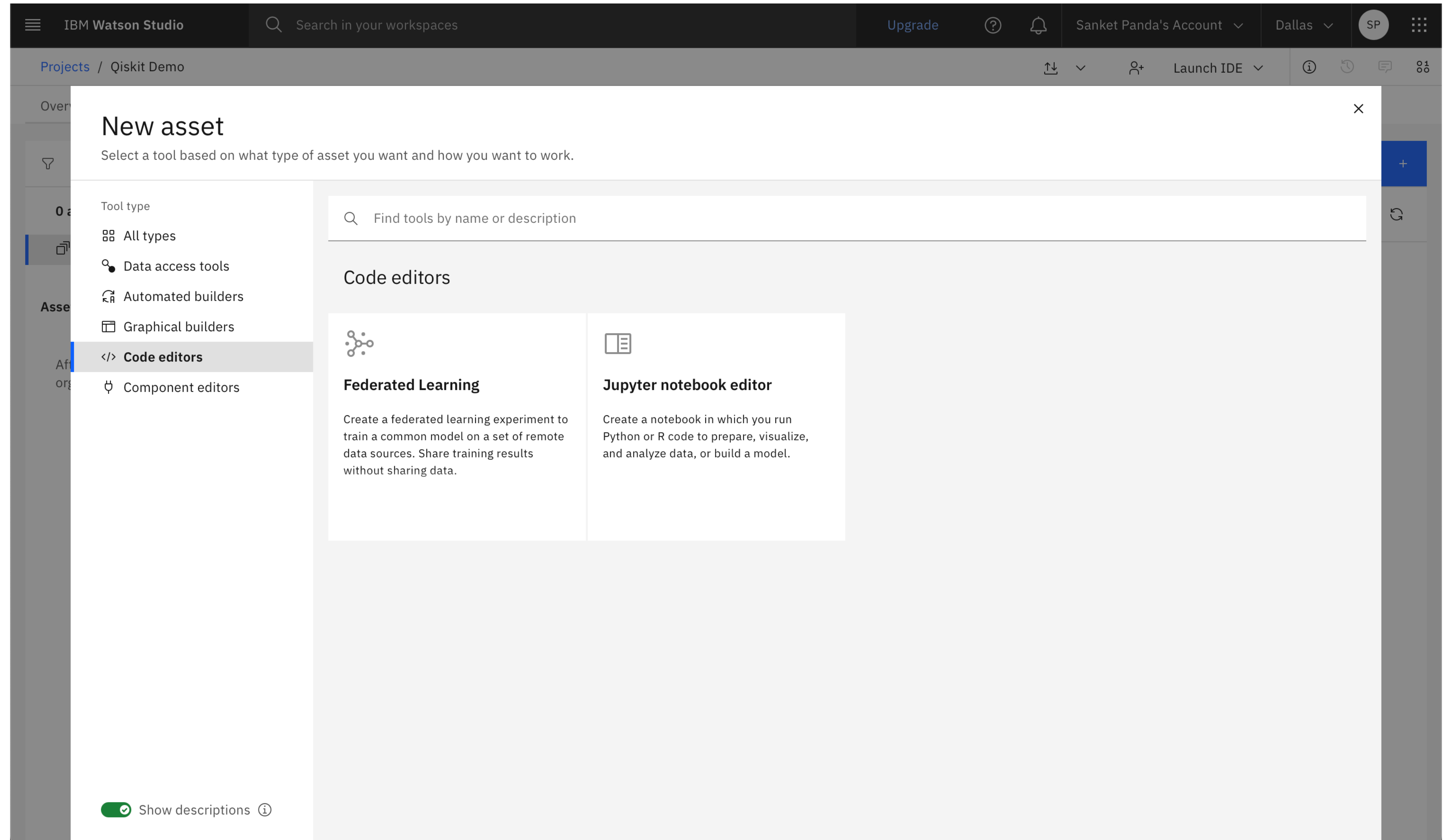
The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and user information. Below that, the breadcrumb 'Projects / Qiskit Demo' is visible. The main content area is divided into tabs: 'Overview', 'Assets' (selected), 'Jobs', and 'Manage'. In the 'Assets' tab, there's a search bar 'Find assets' and two buttons: 'Import assets' and 'New asset' (highlighted in blue). Below the search bar, it says '1 assets'. There's a sidebar on the left with 'Asset types' and 'Notebooks' (with a '1' badge). The main table shows 'All assets' with columns 'Name' and 'Last modified'. One asset is listed: 'Qiskit Demo Notebook' (Notebook), modified '2 minutes ago' by 'you'. On the right, a 'Data in this project' panel shows a dashed box for file upload and a notification for 'variational-quantum-eigensolver.ipynb' with a green checkmark and 'Uploaded!' status, and a 'Dismiss' button.

Creating a new notebook

We'll create a new Jupyter notebook within the asset creator.

On this page

1. Click "Code editors"
2. Click "Jupyter notebook editor"



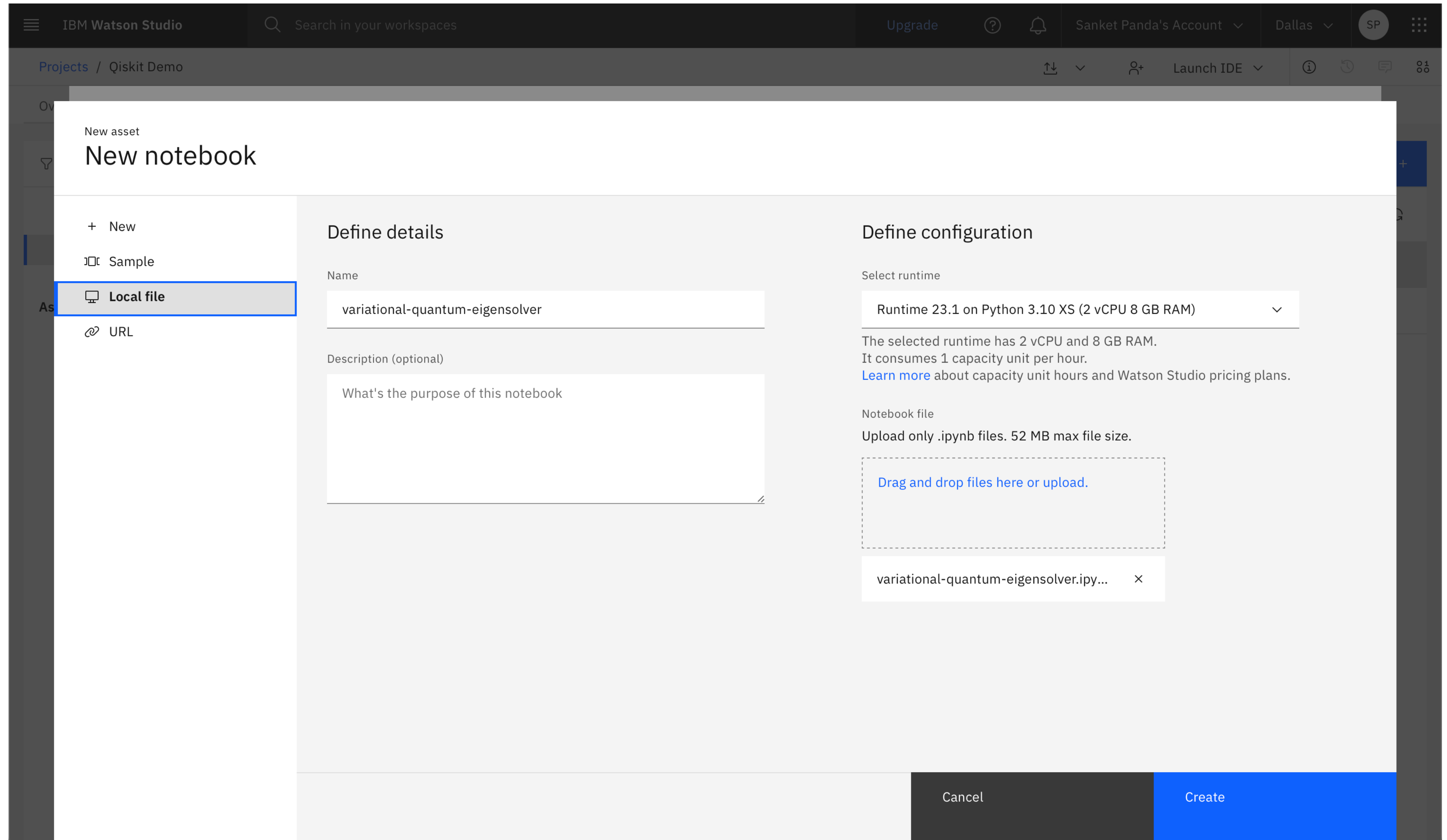
Uploading a local notebook file

Rather than creating a new notebook, we'll upload a local file instead.

This is largely the same process, although we'll additionally upload the `ipynb` file in the bottom right

On this page

1. Add your notebook name and description
2. Select your runtime
3. Upload your notebook file
4. Continue



Verify the uploaded notebook works

We'll click on the notebook in the assets, and verify this works.

If you run into any packages issue, you can easily run the following in a cell:

```
!pip install _____
```

Where the blank line is any package that the notebook requires

On this page

1. Run cells with CMD + Enter

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and user information 'Sanket Panda's Account' and 'Dallas'. Below that, the breadcrumb 'Projects / Qiskit Demo / variational-quantum-eigensolver' is visible. The main area contains a Jupyter notebook with the following content:

Background

[Variational quantum algorithms](#) are promising candidate hybrid-algorithms for observing the utility of quantum computation on noisy near-term devices. Variational algorithms are characterized by the use of a classical optimization algorithm to iteratively update a parameterized trial solution, or "ansatz". Chief among these methods is the Variational Quantum Eigensolver (VQE) that aims to solve for the ground state of a given Hamiltonian represented as a linear combination of Pauli terms, with an ansatz circuit where the number of parameters to optimize over is polynomial in the number of qubits. Given that size of the full solution vector is exponential in the number of qubits, successful minimization using VQE requires, in general, additional problem specific information to define the structure of the ansatz circuit.

Executing a VQE algorithm requires the following 3 components:

1. Hamiltonian and ansatz (problem specification)
2. Qiskit Runtime estimator
3. Classical optimizer

Although the Hamiltonian and ansatz require domain specific knowledge to construct, these details are immaterial to the Runtime, and we can execute a wide class of VQE problems in the same manner.

Setup

Here we import the tools needed for a VQE experiment.

```
In [1]: # General imports
import numpy as np
import warnings

warnings.filterwarnings("ignore")

# Pre-defined ansatz circuit and operator class for Hamiltonian
from qiskit.circuit.library import EfficientSU2
from qiskit.quantum_info import SparsePauliOp

# SciPy minimizer routine
from scipy.optimize import minimize

# Plotting functions
import matplotlib.pyplot as plt

In [2]: # runtime imports
```

Connect your IBM Quantum account

Running quantum jobs from your Watson Studio environment

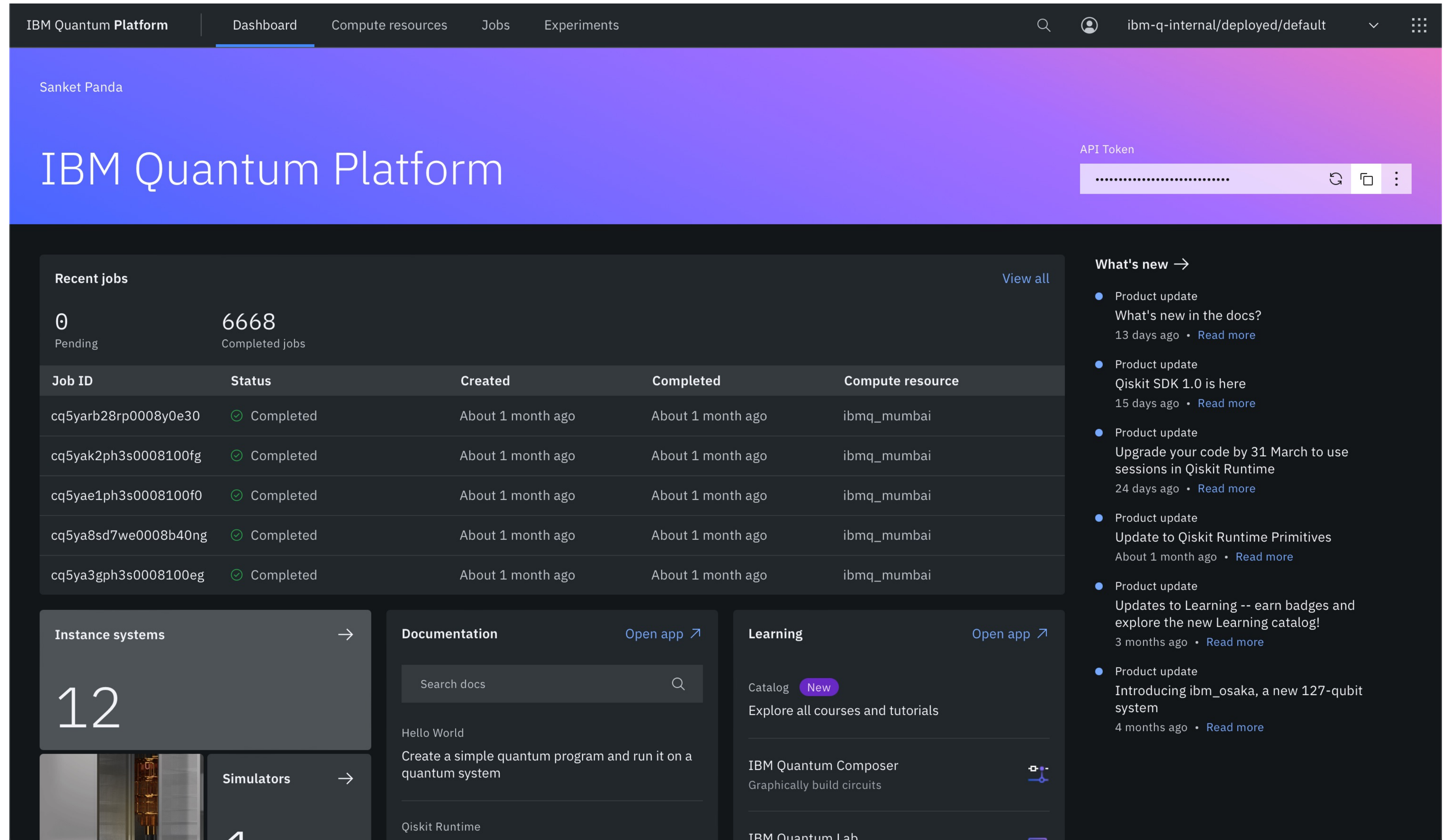
Copy your API Token from the [IBM Quantum Platform](#).

We'll be inputting this into notebooks where Qiskit Runtime is needed:

```
QiskitRuntimeService(  
    channel="ibm_quantum",  
    token="_____"  
)
```

On this page

1. Copy the API token from the top of the IBM Quantum Platform dashboard
2. Paste into cells that have Qiskit Runtime Service authentication



Upgrade to a paid IBM Cloud account

Upgrading to a paid Watson Studio

We'll need to add a credit card to the IBM Cloud account before upgrading to paid Watson Studio.

This option can be found in the Upgrade button on the top

On this page

1. Click "Upgrade"
2. Click "Add credit card"

Upgrade service plans

Upgrade services to paid plans.

Step 1: Add your credit card ⓘ

Add your credit card to buy services and to preserve your data. Optionally, contact sales for a free proof-of-concept or to sign up for a subscription.

[Add credit card](#) [Contact sales](#)

Step 2: Select the services you want to upgrade ⓘ

Service	Name	Use case	Current plan
Watson Studio	WatsonStudio	AI Governance, Data Science and MLOps	Lite
Watson Machine Learning	WatsonMachineLearning	AI Governance, Data Science and MLOps	Lite

[Manage all service instances](#)

