

## Demo- Develop Machine Learning Model To Identify Iris Flower Type using IBM cloud

Step1) Login to IBM cloud portal [cloud.ibm.com](https://cloud.ibm.com) with your credentials.

The screenshot shows the IBM Cloud dashboard. On the left, there's a sidebar with icons for Build, Projects, Classic Infrastructure, Security and Compliance, App Management, and App Development. The main area has a "For you" section with cards for "Build", "Track emissions with Carbon Calculator", "Use Watson Assistant", "Build with Watson", and "Get Started with Watson Studio". Below this are sections for "IBM Cloud status", "Recent support cases", and "Planned maintenance". At the bottom, there's a search bar and a taskbar with various application icons.

Step2) From top left ,Click on Navigation Menu → Resource list

The screenshot shows the "Resource list - IBM Cloud" page. The left sidebar is identical to the dashboard, showing the "Resource list" option selected. The main content area features a table with columns for Group, Location, Product, Status, and Tags. There are filter buttons for each column. The table currently shows 0 resources. At the bottom, there's a search bar and a taskbar.

### Step3) Make sure you deleted all resources under Storage and AI/ Machine Learning

The screenshot shows the IBM Cloud Resource list interface. On the left, there is a navigation sidebar with categories like Compute, Containers, Networking, Storage, Converged infrastructure, Enterprise applications, AI / Machine Learning, Analytics, Blockchain, and Databases. The AI / Machine Learning category is currently selected. The main area displays a table with columns: Name, Group, Location, Product, Status, and Tags. There are several filter and search input fields at the top of the table. A blue 'Create resource' button is located in the top right corner.

### Step4) Search for Watson Studio service

The screenshot shows the IBM Cloud Resource list interface with a search query 'Watson studio' entered in the search bar. The search results show '0 resource results found'. Below the search bar, there is a 'Catalog Results' section with a link to 'View all catalog results'. The main area displays a table with columns: Location, Product, Status, and Tags. There are several filter and search input fields at the top of the table. A blue 'Create resource' button is located in the top right corner.

## Step5) Create Watson Studio service with Free pricing

The screenshot shows the IBM Cloud Catalog interface. On the left, there's a sidebar with service details: Type (Service), Provider (IBM), Last updated (06/11/2024), Category (AI / Machine Learning), Compliance (HIPAA Enabled, IAM-enabled), and Location (Frankfurt). The main area displays the "Watson Studio" service card. It includes a summary table:

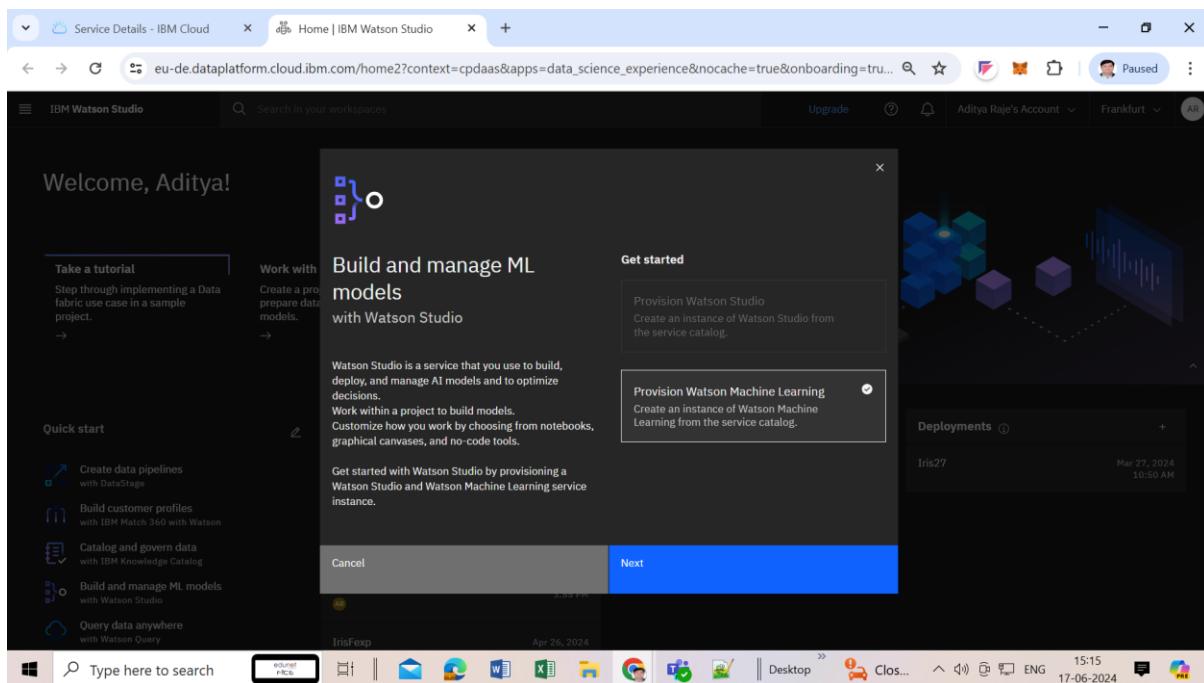
Watson Studio	Free
Location: Frankfurt	
Plan: Lite	
Service name: Watson Studio-i9	
Resource group: Default	

Below the summary, there's a note: "Develop sophisticated machine learning models using Notebooks and code-free tools to infuse AI throughout your business." There are two tabs: "Create" (selected) and "About". A dropdown menu shows "Select a location" with "Frankfurt (eu-de)" selected. A table titled "Select a pricing plan" shows the "Lite" plan with "1 authorized user" and "10 capacity unit-hours monthly limit". The "Pricing" column indicates it's "Free". To the right, there's a checkbox for accepting license agreements and a large blue "Create" button.

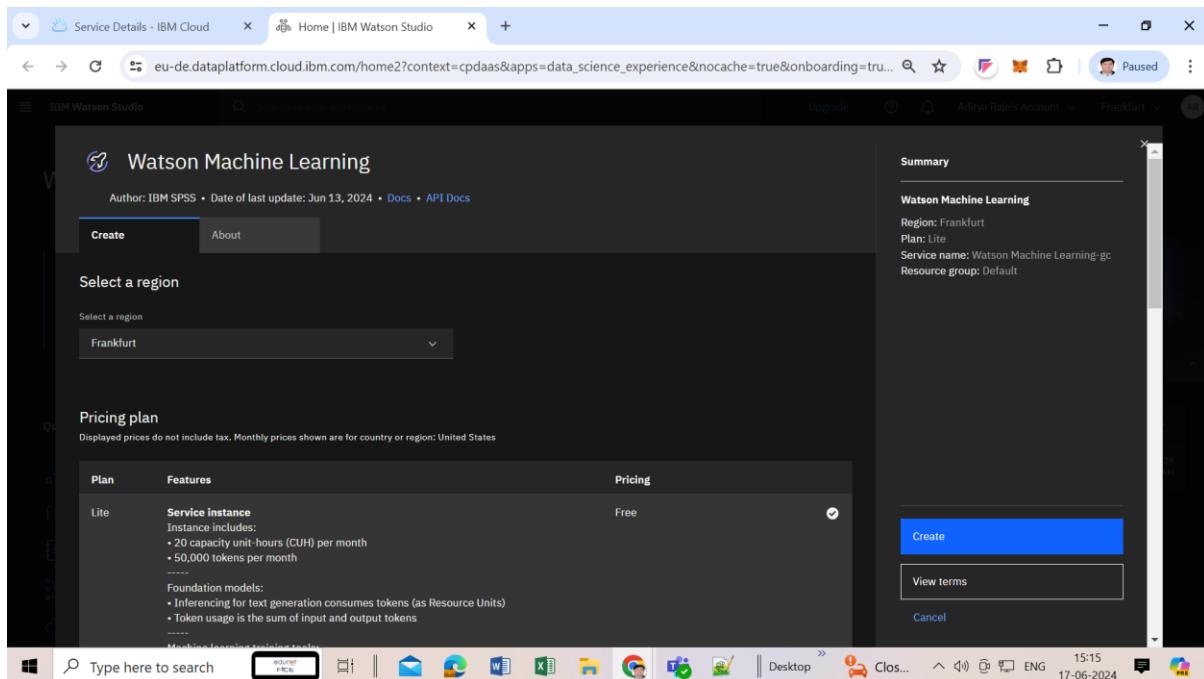
## Step6) Click on Launch In

The screenshot shows the "Service Details - IBM Cloud" page for the "Watson Studio-i9" service. The top navigation bar includes "Catalog", "Manage", and "Aditya Raje's Account". The main content area shows the "Watson Studio in Cloud Pak for Data and watsonx" section. It features a diagram illustrating the architecture: "IBM Watson Studio in Cloud Pak for Data and watsonx" sits atop "IBM Cloud Pak for Data, watsonx Unifying platforms", which sits on top of "IBM Cloud Base cloud infrastructure". Below the diagram, a note states: "IBM Watson Studio is part of IBM Cloud Pak for Data and watsonx, and serves as the AI capability of the data fabric architecture." At the bottom, there are "Helpful links" for "Documentation", "Learning path", and "Videos", along with a "Launch in" button.

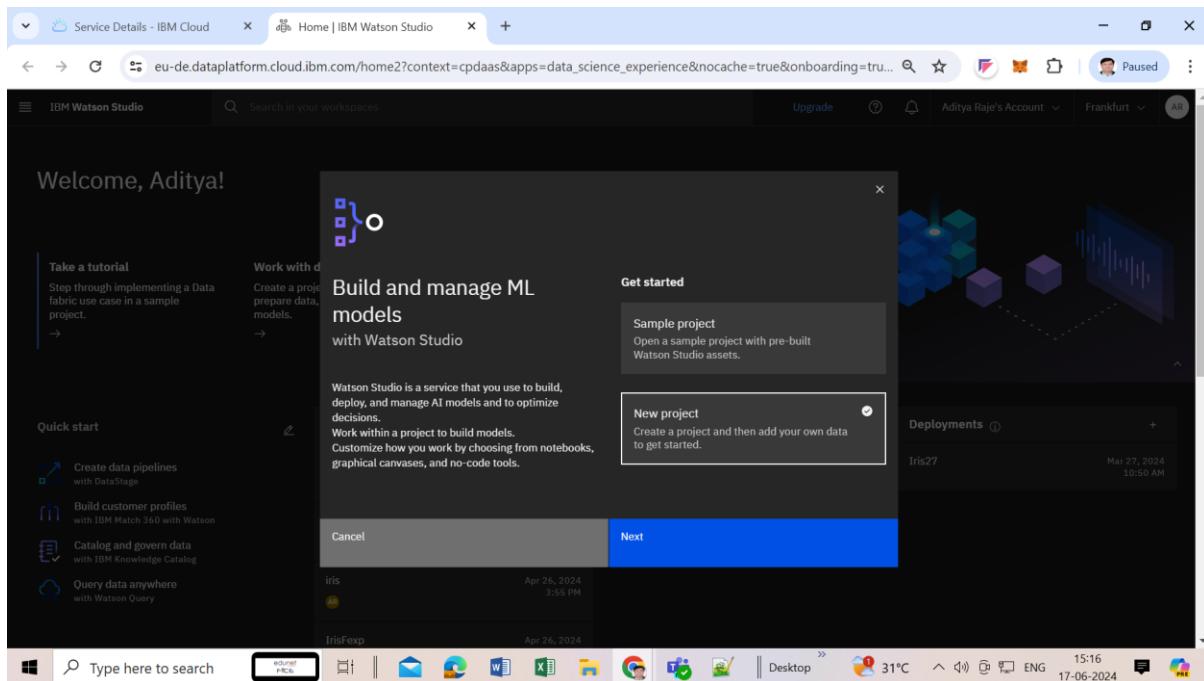
## Step6) Select Provision Watson Machine Learning, Click on Next



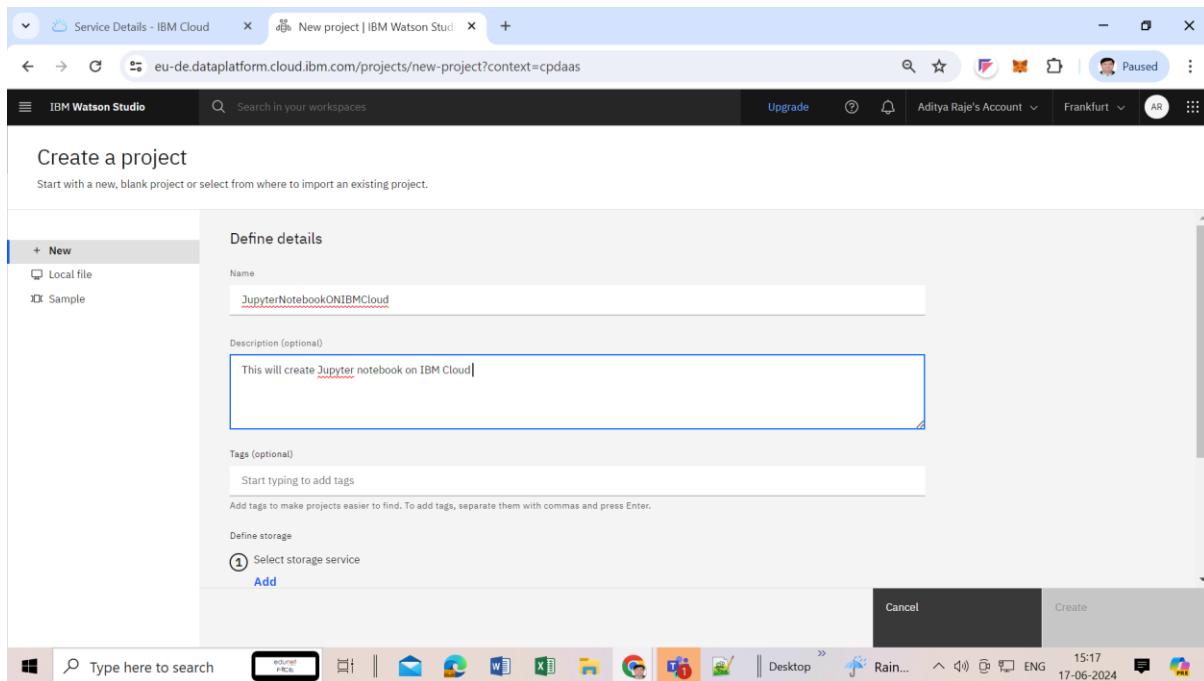
## Step7) Create Watson Machine Learning service



**Step8) Click on New Project and Next**



**Step9) In Create Project window – Provide details about Name , Description**



**Step10) Click on Add to create new storage service to upload files. It will open in new window.**

## Step11) Select Free Pricing with Lite plan then click Create

The screenshot shows the 'Pricing plan' section of the IBM Watson Studio interface. It displays three plan options: One Rate, Lite, and Standard. The 'Lite' plan is selected, indicated by a checked checkbox next to the 'Free' price column. The 'Summary' panel on the right provides details about the selected 'Cloud Object Storage' service, including its region (Global), plan (Lite), service name (Cloud Object Storage-rs), and resource group (Default). A large blue 'Create' button is prominently displayed at the bottom right of the dialog.

## Step12) Click on 2 Refresh

Your storage service name is appeared and Click on Create button

The screenshot shows the 'Create a project' screen in IBM Watson Studio. The 'Storage' field is populated with 'Cloud Object Storage-rs'. At the bottom right, there are two buttons: 'Cancel' and a large blue 'Create' button. The status bar at the bottom indicates it's 31°C, 15:21, and the date is 17-06-2024.

The screenshot shows the IBM Watson Studio interface for a project named "JupyterNotebookONIBMCLOUD". The "Overview" tab is selected. At the top, there are four cards: "Add users as collaborators", "Add data to work with", "Work with data and models in Python or R notebooks", and "Build machine learning models automatically". Below these are sections for "Assets" (with a "View all" link) and "Resource usage" (showing 0 CUH). On the right, there is a "Project history" panel with a message from the user: "You created project JupyterNotebookONIBMCLOUD Today at 3:22 PM". The bottom navigation bar includes icons for search, file, mail, calendar, etc., and the status bar shows "15:23 17-06-2024".

### Step13) Click on Manage tab- Services and Integrations

The screenshot shows the "Manage" tab selected in the IBM Watson Studio interface. The left sidebar has a "Services & integrations" section highlighted. The main area displays a "Services & integrations" section with tabs for "IBM services" and "Third-party integrations". A "Find services" search bar is present. Below it, a "Name" input field and a "Service type" dropdown are shown. A large "No services" message with the sub-instruction "Click Associate service or ask a project Admin to associate one" is displayed. The bottom navigation bar and status bar are visible.

### Step14) Click on Associate service button. Check checkbox – Associate

## Step15) Select Checkbox and Click on Associate

The screenshot shows two views of the IBM Watson Studio interface. The top view is a modal dialog titled 'Associate service' with the sub-instruction 'Choose an existing or add a new service to associate with your project.' It lists a single service entry:

Name	Type	Plan	Location	Status	Group
Watson Machine Learning-gc	Watson Machine Learning	Lite	Frankfurt	Not associated	Default

The 'Associate' button at the bottom right is highlighted in blue. The bottom view shows the main 'Manage' tab of the Watson Studio interface. On the left sidebar, 'Services & integrations' is selected. In the main pane, under 'IBM services (1)', it shows the same service entry:

Name	Service type
Watson Machine Learning-gc	Watson Machine Learning

The 'Associate service' button at the top right of this section is also highlighted in blue.

## Step16) Go to Environments – Template – New Template

The screenshot shows the IBM Watson Studio interface. The left sidebar has a 'Project' section with 'General', 'Access control', 'Environments' (which is selected), 'Resource usage', and 'Services & integrations'. The main area is titled 'Environments' and contains a sub-section 'Tool runtimes' with a button 'Templates (21)'. Below this is a search bar 'Find environment template'. A table lists five templates:

Template name	Tool	Language	Last modified
Default SPSS Modeler S 2 vCPU and 8 GB RAM	Modeler		3 yr ago Created by IBM
Default SPSS Modeler M 4 vCPU and 16 GB RAM	Modeler		3 yr ago Created by IBM
Default SPSS Modeler L 6 vCPU and 24 GB RAM	Modeler		3 yr ago Created by IBM
Default DataStage PX S 1 Conductor: 2 vCPU and 8 GB RAM	DataStage		4 yr ago Created by IBM
Default DataStage PX L 1 Conductor: 8 vCPU and 32 GB RAM	DataStage		4 yr ago Created by IBM

At the bottom right of the table is a blue button 'New template +'. The taskbar at the bottom includes icons for various applications like Microsoft Word, Excel, and Google Chrome.

## Step17) Provide Name, description; Change Hardware configuration to 4VCPU 32GB RAM - Create

The screenshot shows the 'New environment' dialog box. On the left, under 'Define environment details', the 'Name' field is filled with 'python310'. Below it, the 'Description (optional)' field contains the text 'This is 4VCPU with 16 GB RAM and runtime 23.1 on Python 3.10'. On the right, under 'Define configuration', the 'Type' section has 'Default' selected. The 'Hardware configuration' dropdown is set to '4 vCPU and 16 GB RAM'. The 'Software version' dropdown is set to 'Runtime 23.1 on Python 3.10'. At the bottom right of the dialog are 'Cancel' and 'Create' buttons. The taskbar at the bottom includes icons for various applications like Microsoft Word, Excel, and Google Chrome.

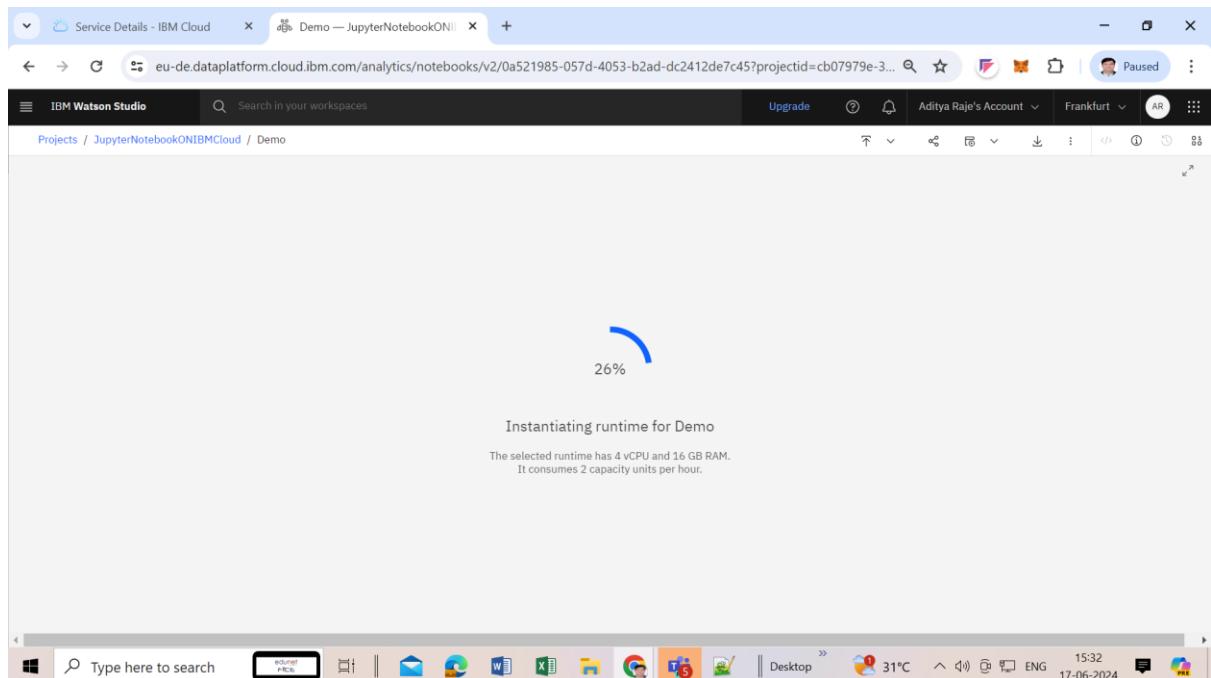
Step18) New Template is listed in list. From newly created Template name, click on **three dots** and select option **New Notebook**

The screenshot shows the IBM Watson Studio interface. On the left, there's a sidebar with 'Project' and 'Environments' selected. The main area is titled 'Environments' and shows a list of templates. A context menu is open over the first template, 'python310', which includes options like 'New notebook', 'Promote to space', 'Edit', and 'Delete'. The status bar at the bottom right shows the date as 17-06-2024 and time as 15:30.

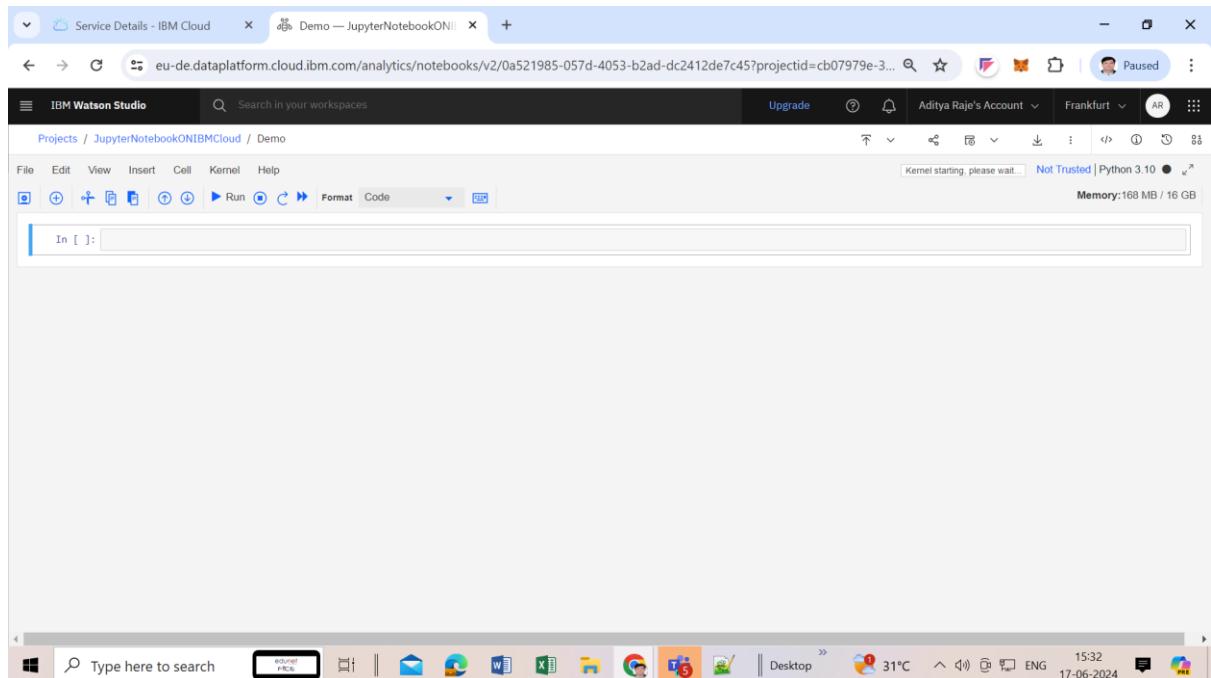
Step19) Provide name, description for New notebook and Click on **Create**.

The screenshot shows the 'New asset' dialog for creating a new notebook. In the 'Define details' section, the 'Name' field is set to 'Demo' and the 'Description (optional)' field contains the text 'This is demo notebook'. In the 'Define configuration' section, the runtime is selected as 'python310 (4 vCPU 16 GB RAM)'. The 'Language' is set to 'Python 3.10'. At the bottom right of the dialog, there are 'Cancel' and 'Create' buttons, with 'Create' being highlighted in blue.

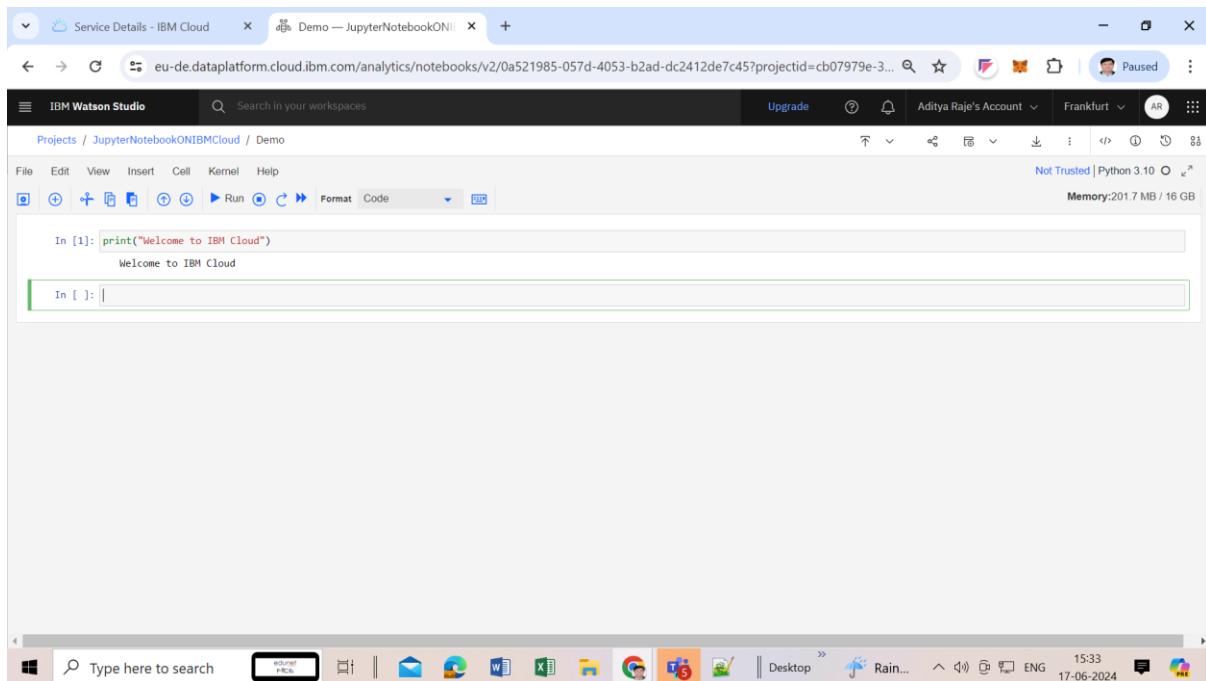
Step20) Wait for 100% to complete and load the Naitebook.



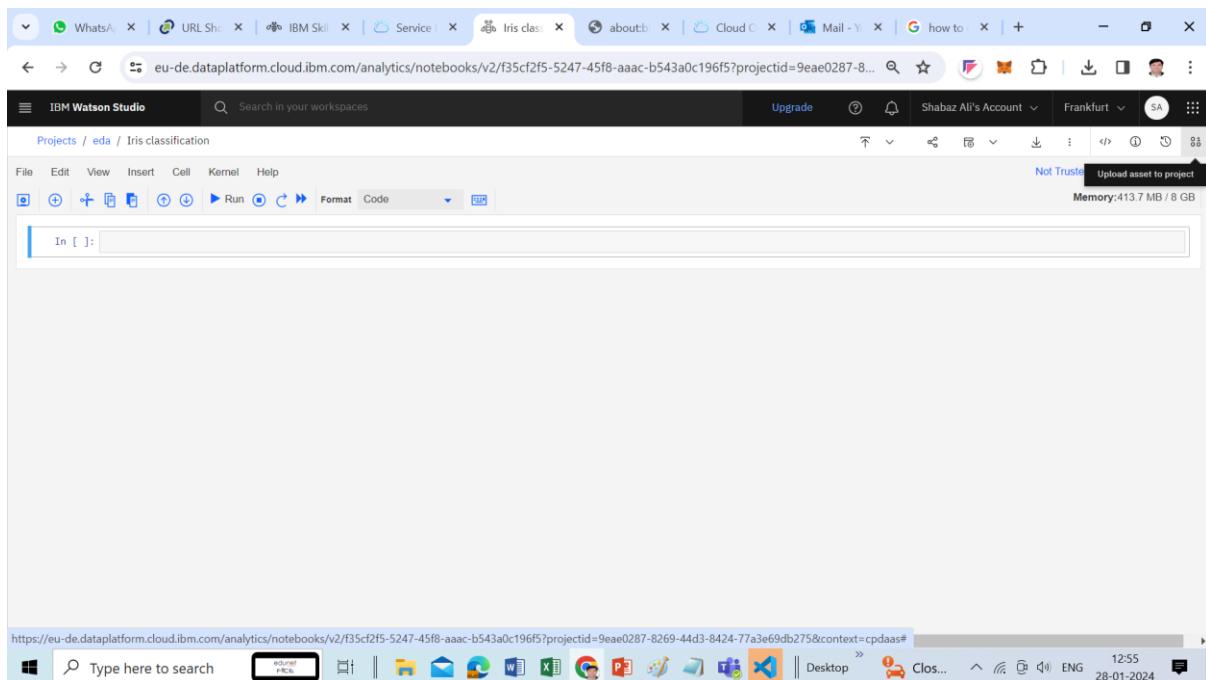
Step21) Now Notebook is ready



## Step22) Run python code



## Step23) From top right corner window, Click on icon Upload asset to project



Step24) In Data in this project, click on **Drop data files here or brose for files to upload**

The screenshot shows the IBM Watson Studio interface. On the right side, there is a panel titled "Data in this project". Within this panel, there is a dashed rectangular area containing the text "Drop data files here or browse for files to upload". This is the step where the user is instructed to upload the Iris.csv file.

Step25) Upload Iris.csv file.

Step26) Import Iris.csv file into code cell. Click on **Code Snippets** icon.

The screenshot shows the IBM Watson Studio interface with the "Code Snippets" icon highlighted in the toolbar. Below the toolbar, two code cells are visible: In [1] contains "import pandas as pd" and In [2] contains "import numpy as np". This is the step where the user imports the required libraries.

**Step27) From Data Ingestion section; Click on Read data.**

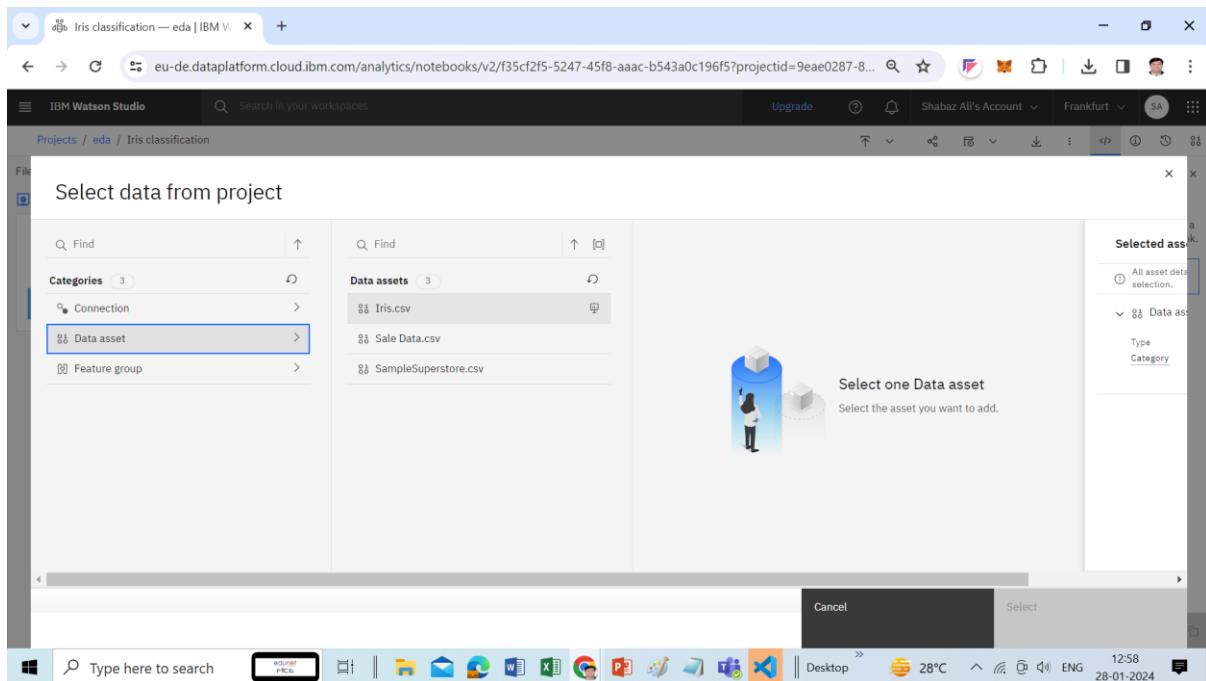
The screenshot shows the IBM Watson Studio interface. On the right side, there is a sidebar titled "Code Snippets" which contains a "Data Ingestion" section. Within this section, there is a "Read data" button. The main workspace shows some Python code in a notebook cell:

```
In [1]: import pandas as pd  
In [2]: import numpy as np
```

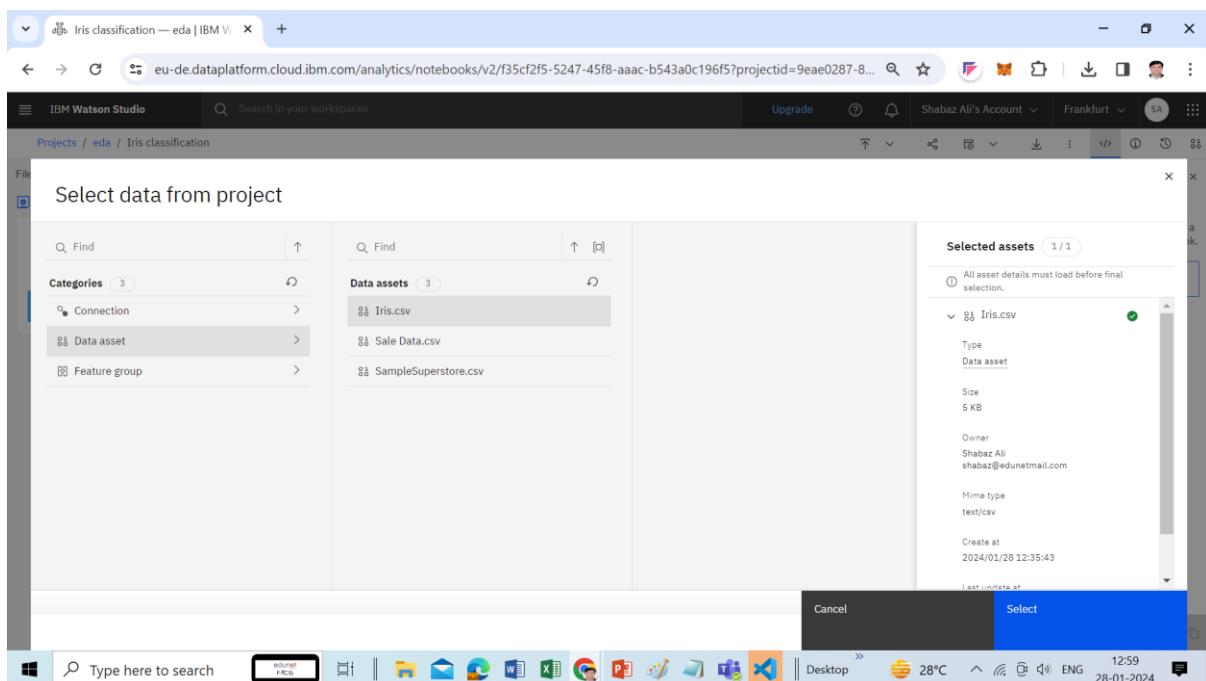
**Step28) Select data from project; Click Data asset**

The screenshot shows the "Select data from project" dialog box. On the left, there is a sidebar with a "Find" input field and a "Categories" section containing "Connection", "Data asset" (which is selected), and "Feature group". The main area is titled "Select an asset" with the sub-instruction "Select the asset you want to add." At the bottom right of the dialog are "Cancel" and "Select" buttons. The status bar at the bottom indicates it's 28°C and the date is 28-01-2024.

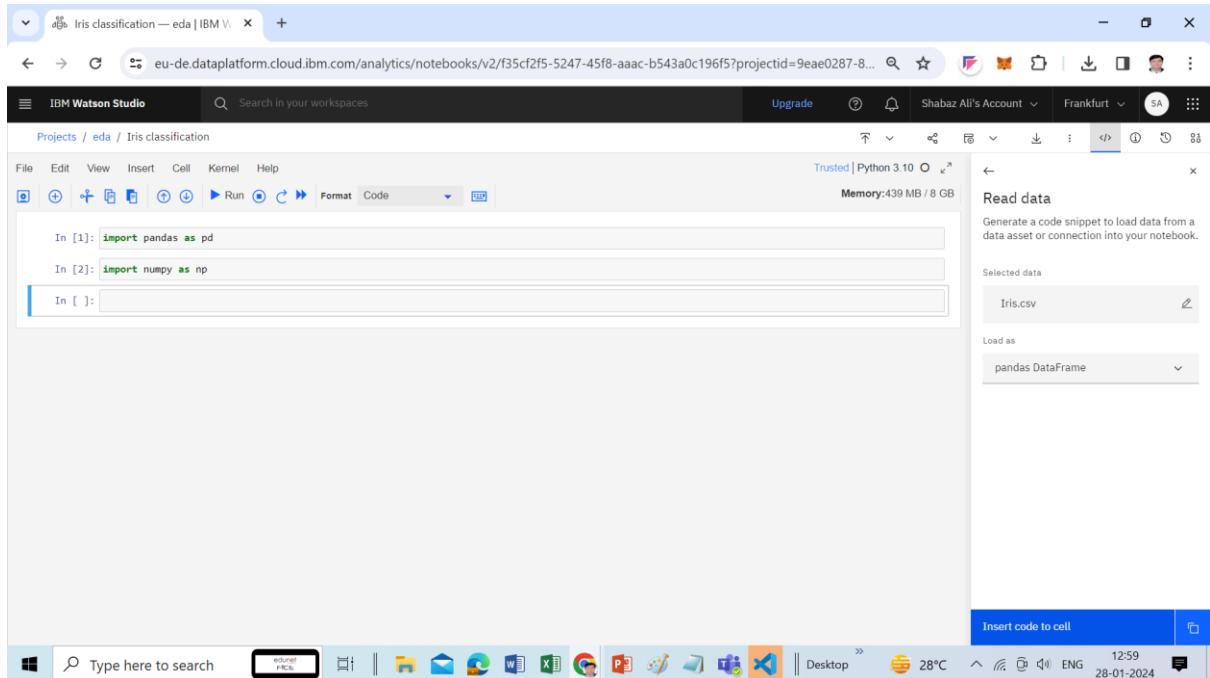
## Step29) Select your Iris.csv file



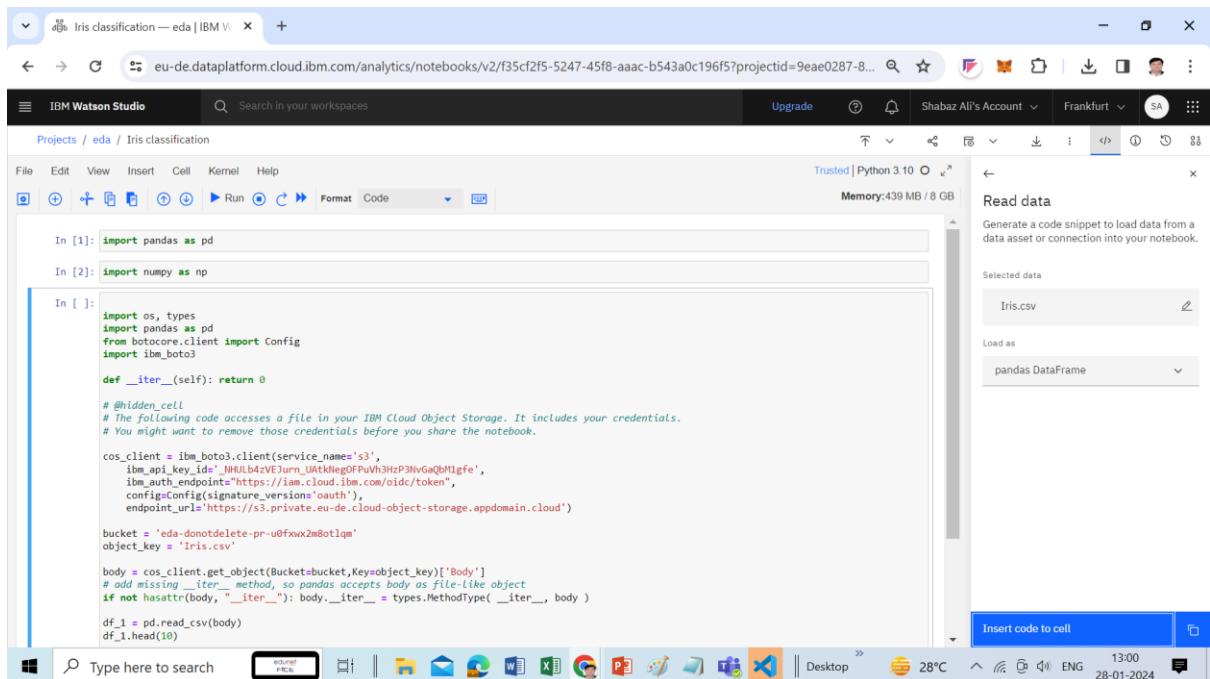
## Step30) Click on Select



### Step31) Copy code or Insert code to cell



### Step32) Code for importing Iris.csv file will get displayed in code cell.



**Step33) Click on Run icon**

The screenshot shows an IBM Watson Studio interface with a notebook titled "Iris classification". The code cell contains Python code to read the Iris dataset from an S3 bucket:

```
object_key = 'iris/iris.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df_1 = pd.read_csv(body)
df_1.head(10)
```

The output cell (Out[3]) displays the first 10 rows of the Iris dataset as a pandas DataFrame:

	ID	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

The notebook interface includes tabs for File, Edit, View, Insert, Cell, Kernel, Help, and a Trusted Python 3.10 environment. The right sidebar shows the "Read data" section with "Iris.csv" selected as the data asset and "pandas DataFrame" as the load format.

Step34) Use following code to develop ML Model for Iris flower type identification

```
dataset = df_1
```

```
#find out last five record  
dataset.tail()
```

```
#Print number of rows and columns of dataset  
dataset.shape
```

```
#Print columns name  
dataset.columns
```

```
#find out lable names  
dataset["Class Label"].values
```

```
#find out unique label names  
dataset["Class Label"].unique()
```

```
#Read data  
import numpy as np  
array = dataset.values  
array
```

```
#Divide the dataset into input feature (Independent features ) and output  
# (dependeant features)  
X = array[:,1:5]  
Y = array[:, 5]
```

```
# split dataset for training and testing  
from sklearn.model_selection import train_test_split  
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,  
random_state=42)
```

```
# Use DecisionTreeClassifier Algorithm to train model  
from sklearn import model_selection  
from sklearn.tree import DecisionTreeClassifier  
dt = DecisionTreeClassifier(max_depth=6, random_state=42)
```

```
dt
```

```
# Perform Model training  
dt.fit(X_train, Y_train)
```

```
# Perform testing  
predictions = dt.predict(X_test)
```

```
#Model Performance Evaluation  
from sklearn.metrics import accuracy_score  
Accuracy = accuracy_score(Y_test, predictions)  
print("\nAccuracy: ", Accuracy)
```

```
#Print confusion Matrix  
from sklearn.metrics import confusion_matrix  
  
Confusion_Matrix = confusion_matrix(Y_test, predictions)  
print("\nConfusion Matrix\n", Confusion_Matrix)
```

```
#Print classification report
from sklearn.metrics import classification_report
Classification_Report = classification_report(Y_test, predictions)
print("\nClassification Report\n", Classification_Report)
```

```
# change the training size and check the model performance
result=dt.predict([[[]]])
print(result)
```

```
# Predict the output form trianed Model
result=dt.predict([[5.1,3.5,1.4,0.2]])
print(result)
```

Congratulations! You successfully developed the ML Model using IBM Cloud.