

<b>PROJECT NAME</b>	Project - Digital Naturalist – AI Enabled tool for Biodiversity Researchers
<b>TEAM ID</b>	PNT2022TMID00699

## IBM CLOUD DEPLOYMENT:

### Resource List:

Resource list

Name	Group	Location	Product	Status	Tags
Cloud Object Storage-20	Default	Global	Cloud Object Storage	Active	—
Watson Machine Learning-pg	Default	Dallas	Watson Machine Learning	Active	—
Watson Studio-pv	Default	Dallas	Watson Studio	Active	cpdaas

### Creating New Project in Cloud:

New project

**Define details**

Name: Digital Naturalist

Description: Project description

**Storage**

Cloud Object Storage-dp

**Choose project options**

☐ Restrict who can be a collaborator ⓘ

☐ Mark as sensitive ⓘ

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

Cancel Create

## API Key Generation:

The screenshot shows the IBM Cloud IAM overview page. The left sidebar contains navigation links for IAM, Manage identities, Users, Trusted profiles, Service IDs, API keys, Identity providers, Manage access, Access groups, Authorizations, Roles, Gain insight, Inactive identities, Inactive policies, and Settings. The main content area features a large purple circular graphic, a 'Policies and rules' section with a '5 / 4020' indicator and a tip to minimize policies, and three informational cards: 'My user details' (showing user name, status, account name, email, account ID, and IAM ID), 'My IBM Cloud API keys' (showing 'Recently updated' and a 'Create one now' link), and 'Do more with IAM' (with links to FAQs and a next option).

The screenshot shows the IBM Cloud IAM API keys page. The left sidebar is identical to the previous image. The main content area is titled 'API keys' and includes a description of API keys and a 'Create' button. A modal window titled 'Create IBM Cloud API key' is open, displaying a form with the following fields:

- Name: api-key-for-wml
- Description: For accessing WML Services

The modal has 'Cancel' and 'Create' buttons at the bottom. The background page shows a 'No API keys' message and a 'Create' button.

The screenshot shows the IBM Cloud IAM 'API keys' page. A modal window titled 'API key details' is open, displaying the following information:

API key details	
<b>Name</b>	api-key-for-wml
<b>Description</b>	For accessing WML Services
<b>ID</b>	ApiKey-3ed523e5-589e-43b5-a493-c676d305f708
<b>Status</b>	Unlocked
<b>Email</b>	211419104303@smartinternz.com
<b>Created by</b>	VIJAYA LAKSHMI K
<b>Date created</b>	2022-11-15 18:42 GMT
<b>Last authentication</b>	--- GMT
<b>Auth count</b>	0

Below the modal, a note states: 'To view more details about the last time this API key was used, go to the [Activity Tracker](#) page.'

## Creating Deployment Space:

The screenshot shows the 'New deployment space' form in IBM Watson Studio. The form is divided into two main sections: 'Define details' and 'Select services'.

**Define details**

- Name:** project-deployments
- Description (Optional):** Deployment space description
- Deployment space tags (optional):** Add a tag

**Select services**

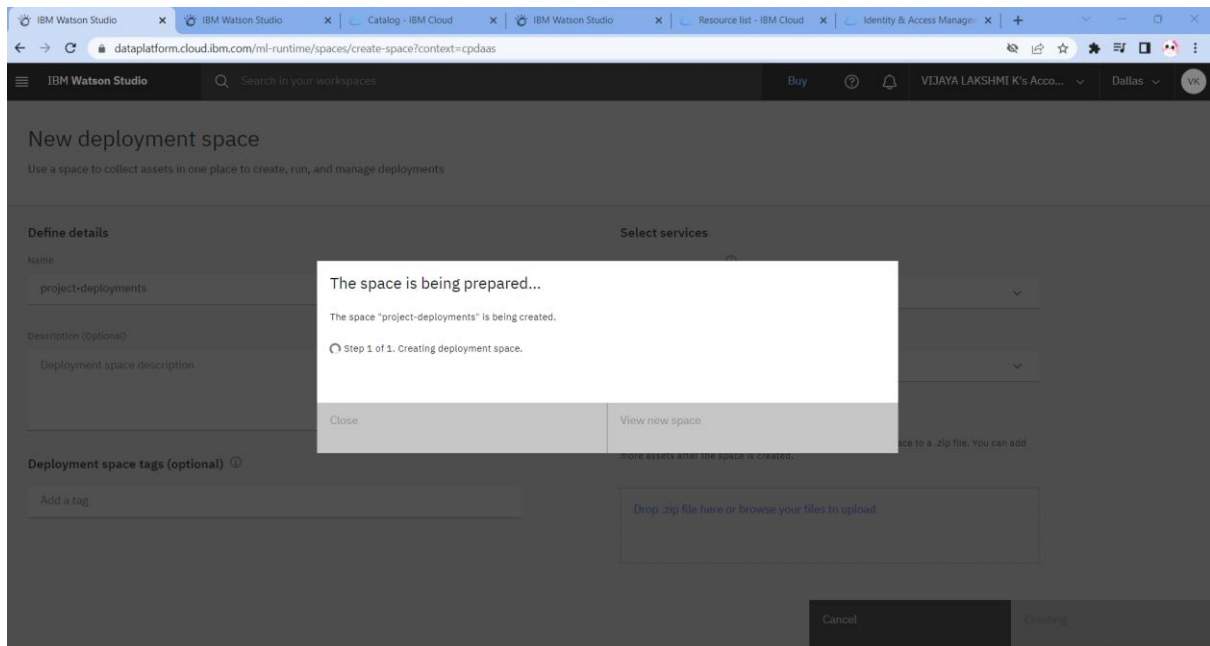
- Select storage service:** Cloud Object Storage-dp
- Select machine learning service (optional):** Watson Machine Learning-vo

**Upload space assets (optional)**

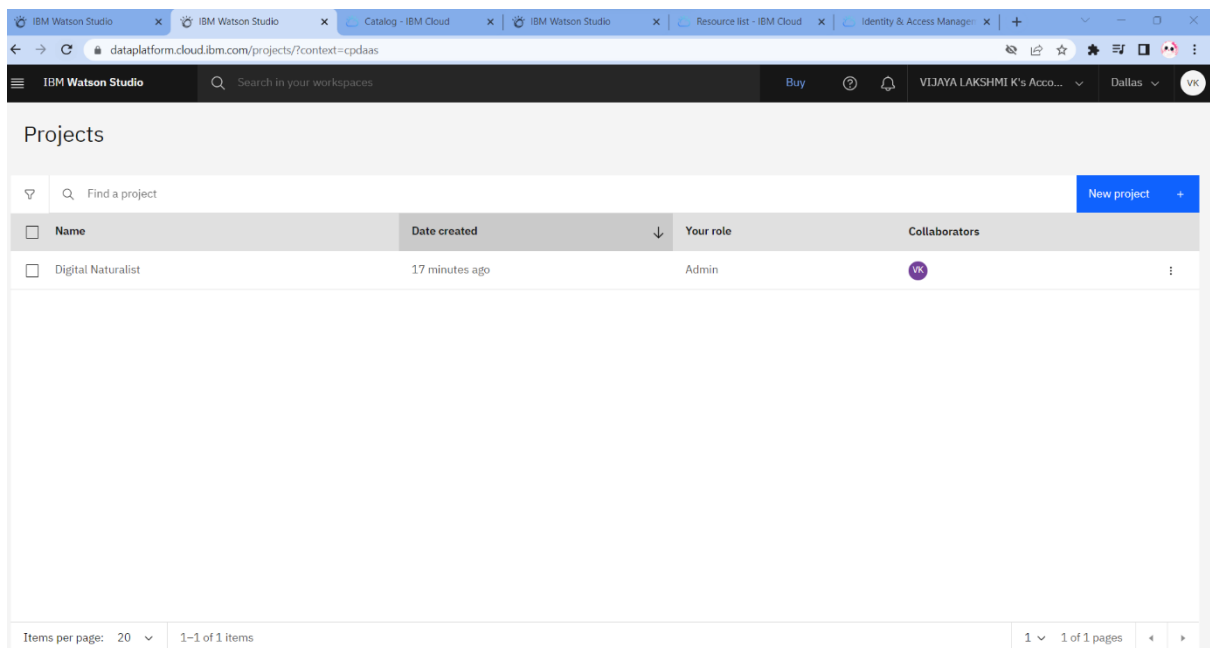
Populate your space with assets exported from a project or space to a .zip file. You can add more assets after the space is created.

Drop .zip file here or browse your files to upload

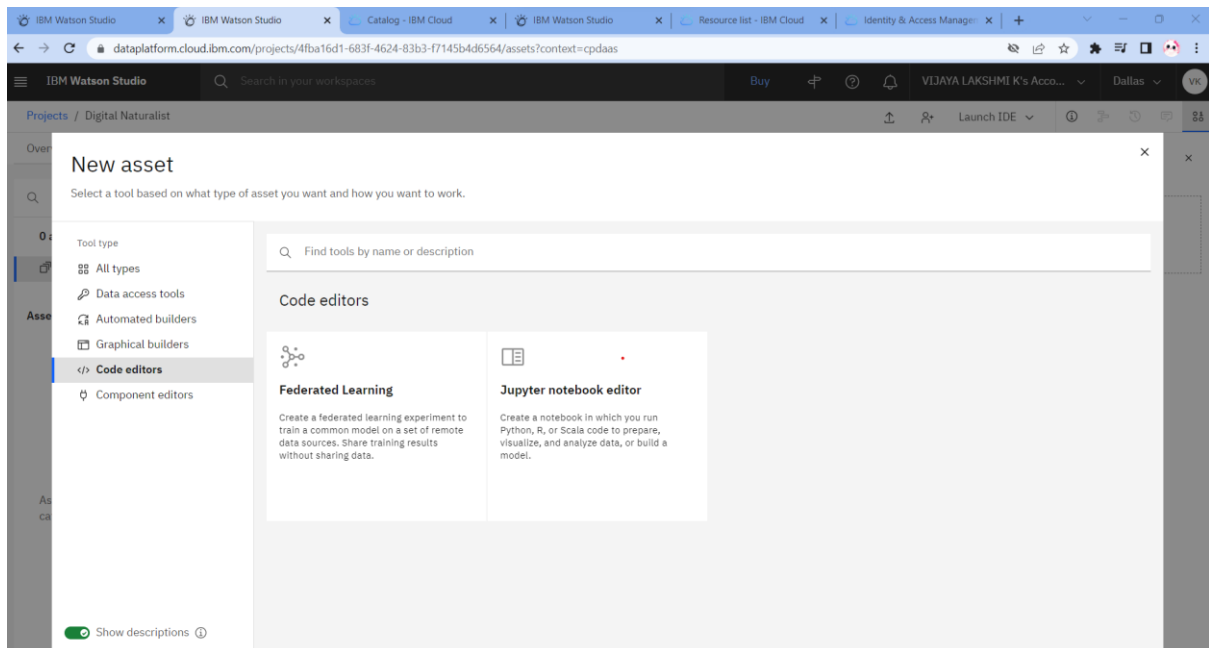
At the bottom right, there are 'Cancel' and 'Create' buttons.



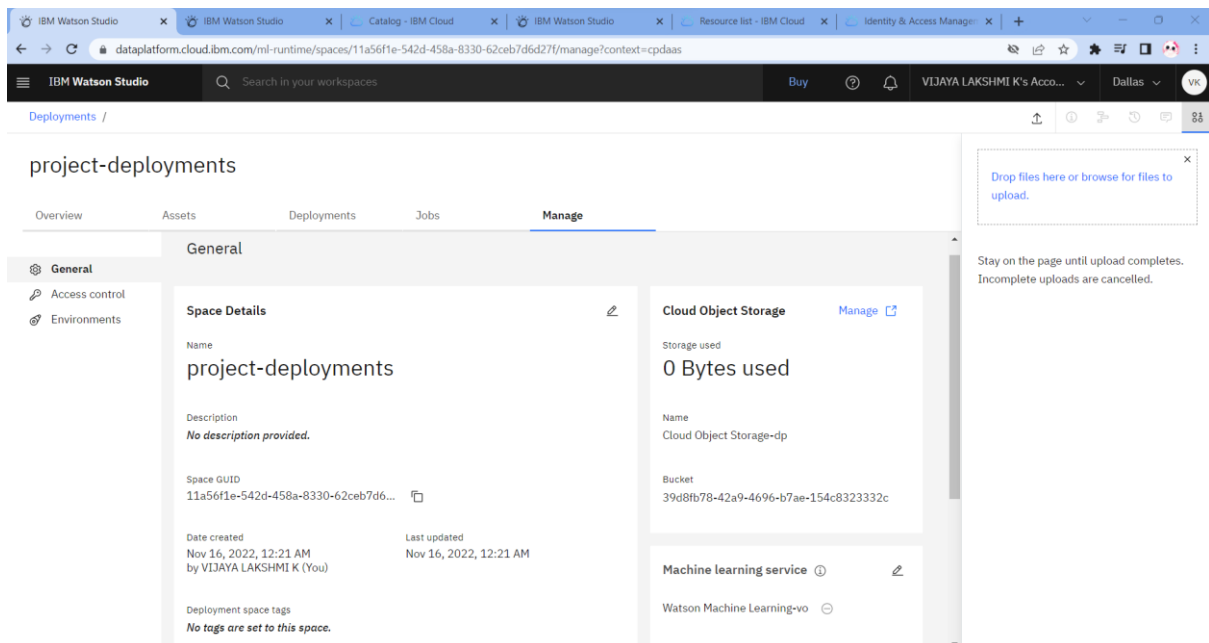
Project Created:

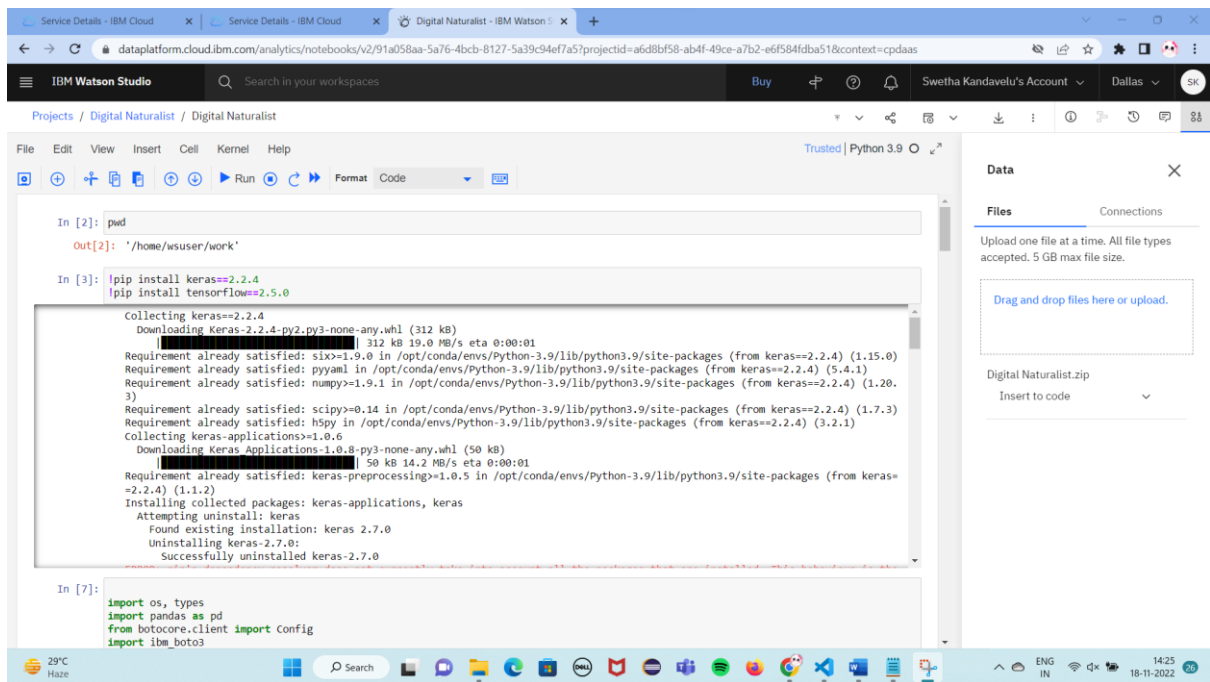
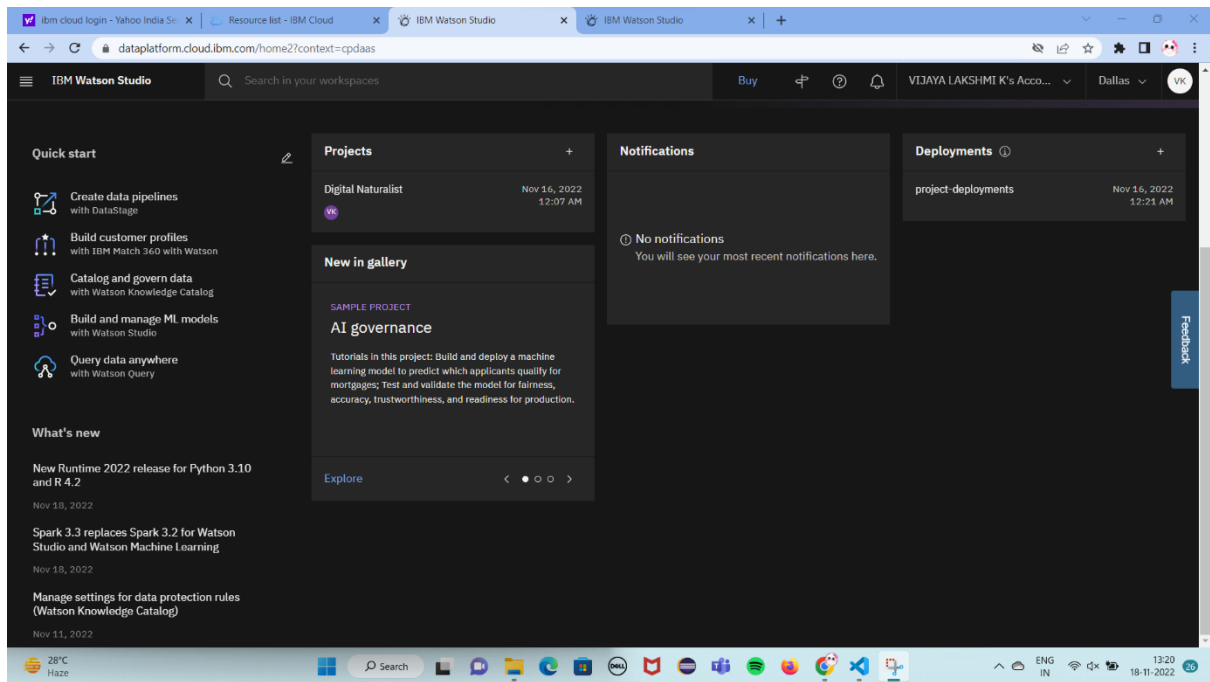


## Creating Jupyter Notebook:



## Deployments:





Service Details - IBM Cloud x Service Details - IBM Cloud x Digital Naturalist - IBM Watson x +

← → ↻ dataplatform.cloud.ibm.com/analytic/notebooks/v2/91a058aa-5a76-4bcb-8127-5a39c94ef7a5?projectId=a6d8bf58-ab4f-49ce-a7b2-e6f584fdba51&context=cpdaas

IBM Watson Studio Search in your workspaces Buy Swetha Kandavelu's Account Dallas SK

Projects / Digital Naturalist / Digital Naturalist

File Edit View Insert Cell Kernel Help Trusted Python 3.9

```
import pandas as pd
from boto3.client import Config
import boto3

def __iter__(self): return 0

# @hidden cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = boto3.client(service_name='s3',
    iam_api_key_id='cey4x0th6urn6kom9u3-azutw6chdQSHUJ2C_SFqvYb',
    iam_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'digitalnaturalist-donotdelete-pr-urjtmkmaqj9az'
object_key = 'Digital Naturalist.zip'

streaming_body_6 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a boto3.response.StreamingBody object.
# Please read the documentation of boto3 and pandas to learn more about the possibilities to load the data.
# boto3 documentation: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#python
# pandas documentation: http://pandas.pydata.org/

In [8]: #unzipping your data file
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_6.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

29°C Haze

Search

ENG IN 14:25 18-11-2022

Data

Files Connections

Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

Digital Naturalist.zip

Insert to code

Service Details - IBM Cloud x Service Details - IBM Cloud x Digital Naturalist - IBM Watson x +

← → ↻ dataplatform.cloud.ibm.com/analytic/notebooks/v2/91a058aa-5a76-4bcb-8127-5a39c94ef7a5?projectId=a6d8bf58-ab4f-49ce-a7b2-e6f584fdba51&context=cpdaas

IBM Watson Studio Search in your workspaces Buy Swetha Kandavelu's Account Dallas SK

Projects / Digital Naturalist / Digital Naturalist

File Edit View Insert Cell Kernel Help Trusted Python 3.9

```
In [10]: pwd
Out[10]: '/home/wsuser/work'

In [11]: import os
filenames=os.listdir('/home/wsuser/work/Digital Naturalist/')

In [12]: from tensorflow.keras.preprocessing.image import ImageDataGenerator

In [14]: # setting parameters for data augmentation for training data
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)

In [15]: # data augmentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)

In [16]: #performing data augmentation to train data
x_train=train_datagen.flow_from_directory(directory=r'/home/wsuser/work/Digital Naturalist/train',target_size=(64,64),batch_size=32,class_indices=
Found 5890 images belonging to 3 classes.

In [17]: x_test=test_datagen.flow_from_directory(directory=r'/home/wsuser/work/Digital Naturalist/test',target_size=(64,64),batch_size=32,class_
Found 5890 images belonging to 3 classes.

In [18]: #lets see the classes the different types of arrythmia is stored in
x_train.class_indices
Out[18]: {'Bird': 0, 'Flower': 1, 'Mammal': 2}
```

29°C Haze

Search

ENG IN 14:26 18-11-2022

Data

Files Connections

Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

Digital Naturalist.zip

Insert to code

## Deploying to Cloud:

The screenshot shows the IBM Watson Studio interface with a notebook titled "Digital Naturalist Deployment". The notebook content includes the following code and output:

```
In [1]: #Installing machine learning service
!pip install watson-machine-learning-client

Collecting watson-machine-learning-client
  Downloading watson-machine-learning-client-1.0.391-py3-none-any.whl (538 kB)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore->1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)
Requirement already satisfied: charset-normalizer==2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.19.5)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

```
In [17]: #Replace the credentials that you got from Watson Machine Learning Service
from ibm_watson_machine_learning import APIClient
uml_credentials={
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "VbAtsHdn0MnU006l0i4zVpJe-qQj2H2VnmMv_Ql1hPP"
}
```

The screenshot shows the IBM Watson Studio interface with the same notebook. The code and output for the second cell are as follows:

```
In [17]: #Replace the credentials that you got from Watson Machine Learning Service
from ibm_watson_machine_learning import APIClient
uml_credentials={
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "VbAtsHdn0MnU006l0i4zVpJe-qQj2H2VnmMv_Ql1hPP"
}
client=APIClient(uml_credentials)
```

```
In [18]: client=APIClient(uml_credentials)
```

```
In [19]: client
```

```
Out[19]: <ibm_watson_machine_learning.client.APIClient at 0x7f21c7a47670>
```

```
In [20]: def guid_from_space_name(client, space_name):
space=client.spaces.get_details()
#print(space)
return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])
```

```
In [22]: space_uid=guid_from_space_name(client, 'project-deployments')
print("space UID="+space_uid)
space UID=11a56f1e-542d-458a-8330-62ceb7d6d27f
```

```
In [23]: client.set_default_space(space_uid)
```

```
Out[23]: 'SUCCESS'
```

```
In [24]: client.software_specifications.list()
```

NAME	ASSET_ID	TYPE
default_py3.6	0062b8c9-8b7d-44a0-a9b9-46c416adcbd9	base
kernel-spark3.2-scala2.12	020d69ce-7ac1-5e68-ac1a-31189867356a	base
pytorch-onnx 1.3-py3.7-edt	069ea134-3346-5748-b513-49120e15d288	base



IBM Watson Studio

Projects / Digital Naturalist / Digital Naturalist Deployment

```
python3-tensorflow-rt22.1-py3.9 40e21237-1b2e-2222-021a-0c0b94291421 base
default_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base
autoai-ts-rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base
autoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base
pml-3.0.4.3 493bc695-16f1-5b05-bae8-91a8af80e9c7 base
spark-mllib_2.4-r_3.6 49403dff-92a9-4c87-a3d7-a42d0021c095 base
xgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base
pytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base
autoai-ts_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cb42cde base
spark-mllib_2.4-scala_2.11 55a70f99-7320-4be5-9fb9-9ed5a443af5 base
spark-mllib_3.0 5c180ca2-4977-5c2e-9d39-ff644ea8ff69 base
autoai-obm_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base
spss-modeler_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base
cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base
runtime-22.2-py3.10-xc 5e8cd8ff-db4a-5a6a-b8aa-2d4af9864dab base
autoai-kb_3.1-py3.7 632d4b22-10aa-5100-88f0-f52dfb6444d7 base
.....
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

```
In [25]: software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow-rt22.1-py3.9")
software_spec_uid

Out[25]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
In [ ]: model_details = client.repository.store_model(model='my_model_new.tar.gz', meta_props={
client.repository.ModelMetaNames.NAME: "CNN",
client.repository.ModelMetaNames.TYPE: "tensorflow_2.7",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
})

In [ ]: model_id = client.repository.get_model_uid(model_details)

In [ ]: model_id

In [ ]: client.repository.download(model_id, 'model.tar.gz')

In [ ]:
```

26°C Mostly clear

Search

ENG IN

22:52 18-11-2022