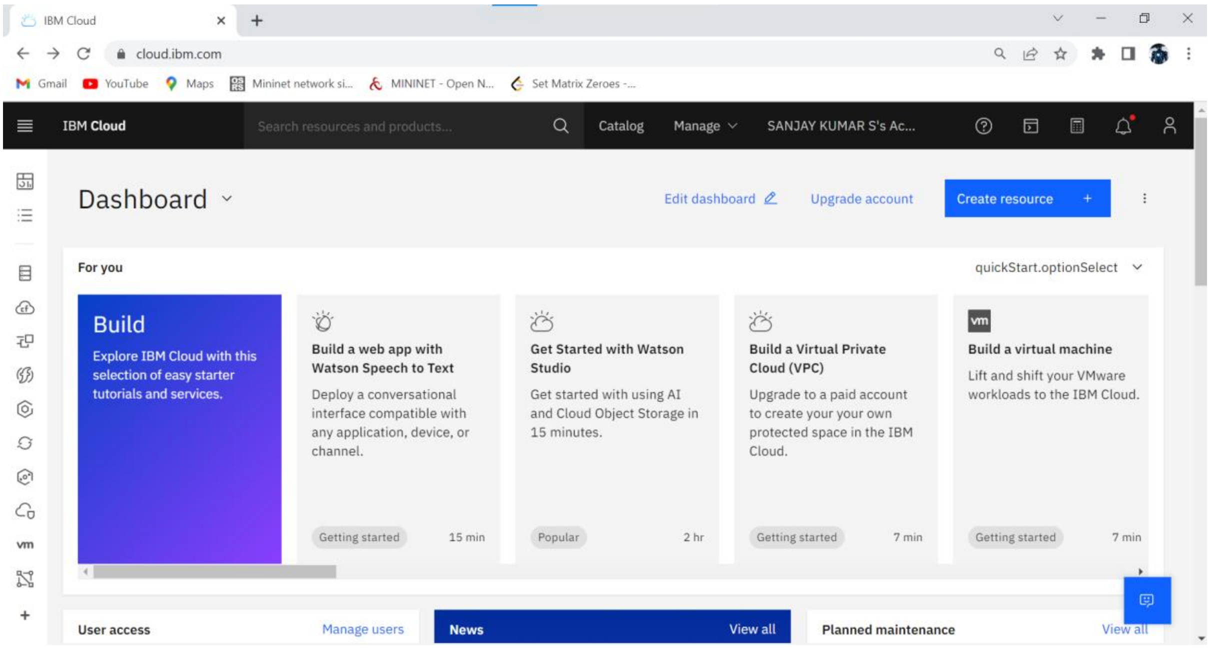
# Train the model on IBM

| | |
|---|---|
| Team ID | PNT2022TMID35762 |
| Project Name | DEMANDEST – AI POWERED FOOD DEMAND FORECASTER |

## IBM Cloud Dashboard:



Dashboard

For you

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Build a web app with Watson Speech to Text

Deploy a conversational interface compatible with any application, device, or channel.

Get Started with Watson Studio

Get started with using AI and Cloud Object Storage in 15 minutes.

Build a Virtual Private Cloud (VPC)

Upgrade to a paid account to create your your own protected space in the IBM Cloud.

Build a virtual machine

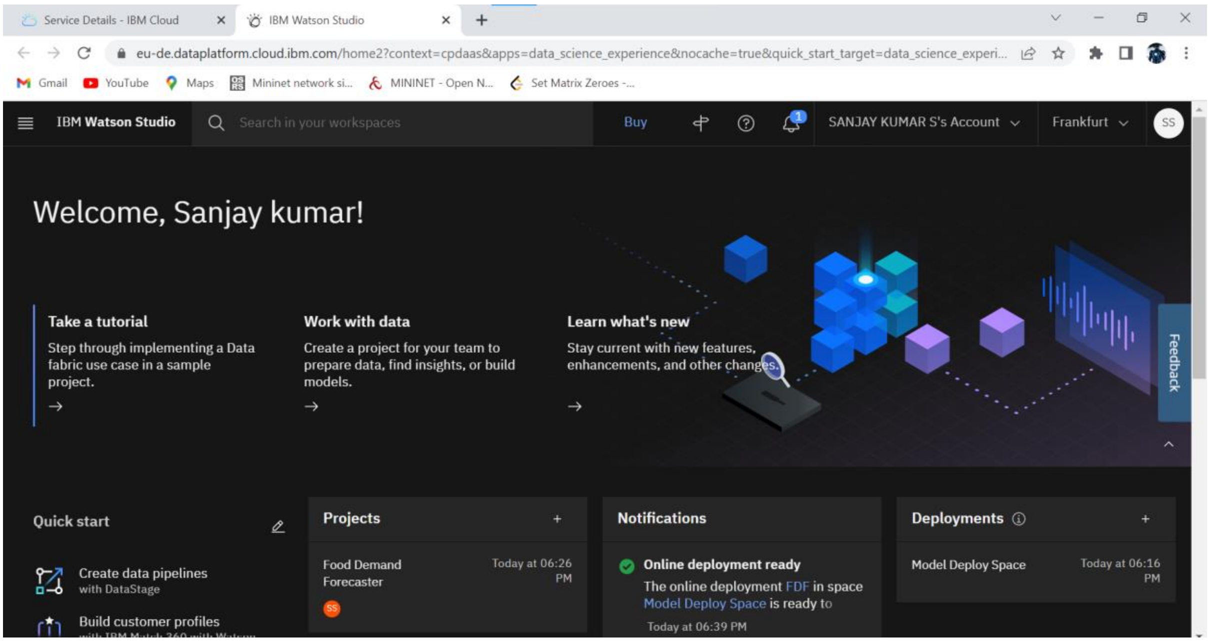
Lift and shift your VMware workloads to the IBM Cloud.

User access

News

Planned maintenance

## IBM Watson Studio:


Service Details - IBM Cloud
IBM Watson Studio
eu-de.dataplatform.cloud.ibm.com/home2?context=cpdaas&apps=data_science_experience&nocache=true&quick_start_target=data_science_experi...
Gmail
YouTube
Maps
Mininet network si...
MININET - Open N...
Set Matrix Zeroes -...
IBM Watson Studio
Search in your workspaces
Buy
SANJAY KUMAR S's Account
Frankfurt
SS
Feedback
SS

Welcome, Sanjay kumar!

Take a tutorial

Step through implementing a Data fabric use case in a sample project.

Work with data

Create a project for your team to prepare data, find insights, or build models.

Learn what's new

Stay current with new features, enhancements, and other changes.

Quick start

Create data pipelines with DataStage

Build customer profiles with IBM Match 360 with Watson

Projects

Food Demand Forecaster — Today at 06:26 PM

Notifications

Online deployment ready

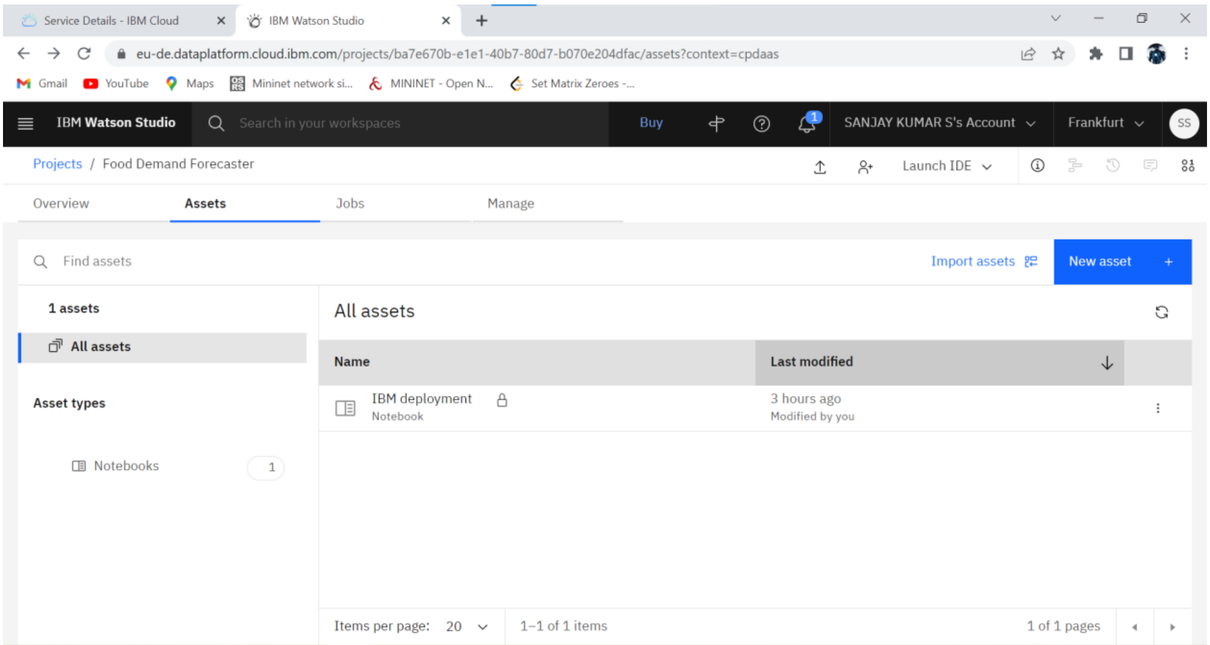
The online deployment FDF in space Model Deploy Space is ready to

Today at 06:39 PM

Deployments

Model Deploy Space — Today at 06:16 PM

# Deployed Model:



eu-de.dataplatform.cloud.ibm.com/projects/ba7e670b-e1e1-40b7-80d7-b070e204dfac/assets?context=cpdaas

Projects / Food Demand Forecaster

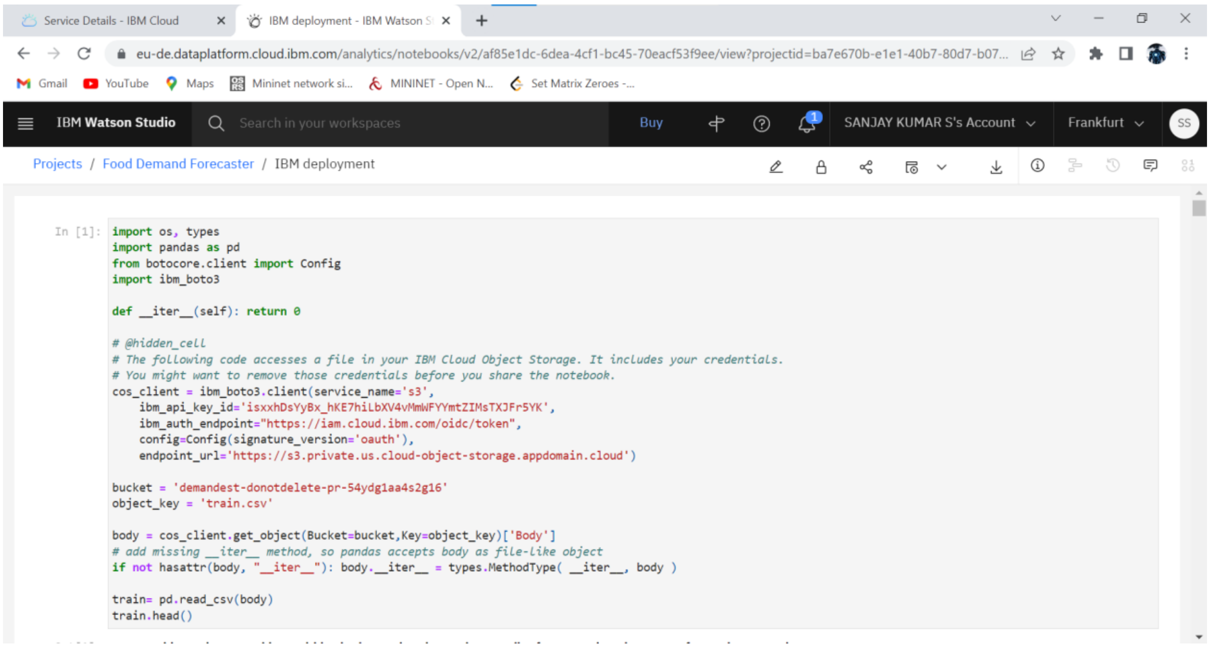
1 assets

All assets

| Name | Last modified |
|---|---|
| IBM deployment<br>Notebook | 3 hours ago<br>Modified by you |

Items per page: 20 1–1 of 1 items 1 of 1 pages

# Deployed code:

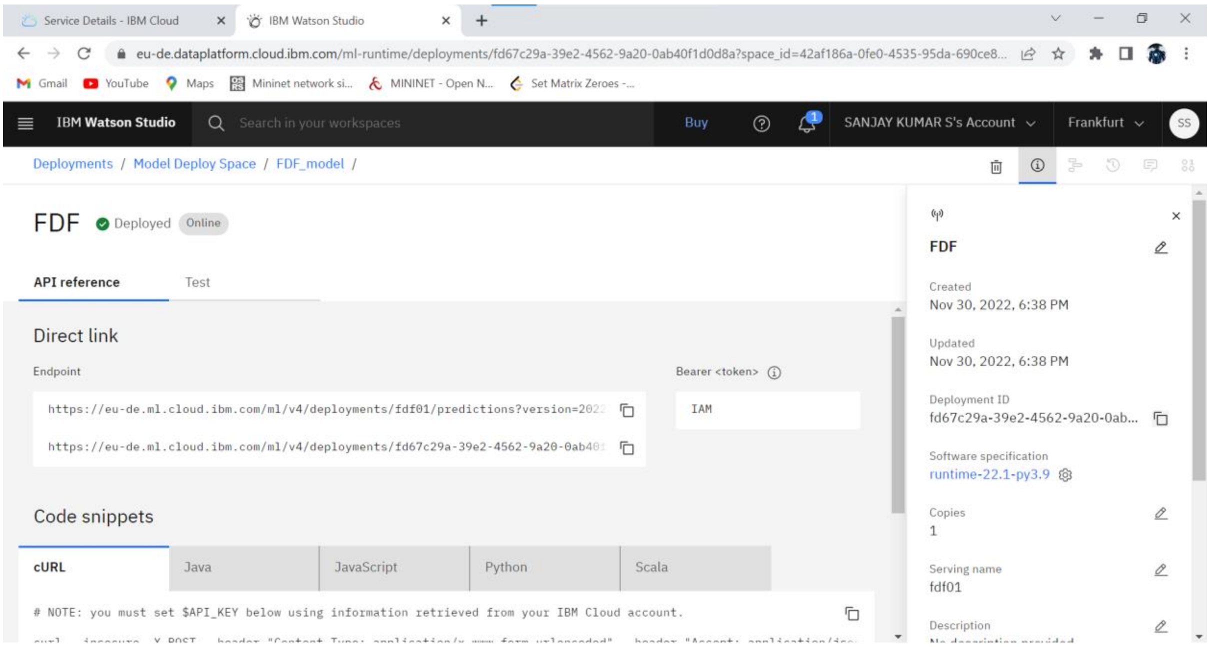


eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/af85e1dc-6dea-4cf1-bc45-70eacf53f9ee/view?projectid=ba7e670b-e1e1-40b7-80d7-b07...

Projects / Food Demand Forecaster / IBM deployment

```
In [1]: import os, types
        import pandas as pd
        from botocore.client import Config
        import ibm_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 = ibm_boto3.client(service_name='s3',
            ibm_api_key_id='isxxhDsYyBx_hKE7hiLbXV4vMmWFYYmtZIMsTXJFr5YK',
            ibm_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 = 'demandest-donotdelete-pr-54ydg1aa4s2g16'
        object_key = 'train.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 )

        train= pd.read_csv(body)
        train.head()
```

## Deployment ID & Endpoint:


Service Details - IBM Cloud
IBM Watson Studio
eu-de.dataplatform.cloud.ibm.com/ml-runtime/deployments/fd67c29a-39e2-4562-9a20-0ab40f1d0d8a?space_id=42af186a-0fe0-4535-95da-690ce8...
Gmail
YouTube
Maps
Mininet network si...
MININET - Open N...
Set Matrix Zeroes -...
IBM Watson Studio
Search in your workspaces
Buy
1
SANJAY KUMAR S's Account
Frankfurt
SS
Deployments / Model Deploy Space / FDF_model /
API reference
Test
cURL
Java
JavaScript
Python
Scala

FDF Deployed Online

Direct link

Endpoint

```
https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/fdf01/predictions?version=2022
https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/fd67c29a-39e2-4562-9a20-0ab40
```

Bearer <token>

IAM

Code snippets

```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account.
```

FDF

Created
Nov 30, 2022, 6:38 PM

Updated
Nov 30, 2022, 6:38 PM

Deployment ID
fd67c29a-39e2-4562-9a20-0ab...

Software specification
runtime-22.1-py3.9

Copies
1

Serving name
fdf01

Description

## app_ibm.py:

```
import pandas as pd

import numpy as np

import pickle

import os

import requests


API_KEY = "v3n6gq55C00788iHyYsU0cm1GG__LDIaxvhoZT6hB79m"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":

 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token_response.json()["access_token"]


header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}


from flask import Flask,request, render_template

app=Flask(__name__,template_folder="templates")

@app.route('/')
```

```
def index():
    return render_template('index.html')
@app.route('/index')
def about():
    return render_template('index.html')
@app.route('/index1')
def page():
    return render_template('predict.html',prediction_text=0)
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
    input_features = [float(x) for x in request.form.values()]
    print(input_features)
    features_value = [[np.array(input_features)]]
    print(features_value)

    payload_scoring = {"input_data":[{"field": [['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine','city_code', 'region_code', 'category']],"values": [input_features]}]}
    response_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/fd67c29a-39e2-4562-9a20-0ab40f1d0d8a/predictions?version=2022-11-30', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    prediction =response_scoring.json()
    print("dir(predict) :" ,dir(prediction))
    print(prediction)
    #print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
    output = prediction['predictions'][0]['values'][0][0]
    return render_template("predict.html", prediction_text=int(output))

if __name__ == '__main__':
      app.run(debug=False)
```