



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Хмарні технології обробки даних

Лабораторна робота №2

Використання AWS Simple Storage Service (S3)

Мета: ознайомитись з використанням AWS Simple Storage Service (S3)

Перевірив:

Виконав:

студент III курсу

групи ФБ-01

Сахній Н.Р.

Київ 2023

ФБ-01 Сахній Назар

Завдання:

- Створити бакет S3
- Налаштувати доступ до нього з інстансу, створеного у попередній лабораторній роботі
- Ознайомитись зі способами взаємодії з ним

Хід виконання роботи

1. Для початку програматично засобами мови програмування Python отримаємо дані щодо курсу гривні у JSON-форматі на інстанс:

```
nazar@snz24 ~[~nazar/KPI/CloudTech]
$ ssh -i kali.pem ec2-user@3.72.67.43
load pubkey "kali.pem": invalid format

#
_#_####_
~\#####\
~~~\#####|
~~~~\###|\
~~~~~\#/
~~~~~V-+--> https://aws.amazon.com/linux/amazon-linux-2023
~~~~~\
~~~~~/m/

Last login: Sun Apr 9 13:37:02 2023 from 178.158.203.65
[ec2-user@ip-172-31-31-220 ~]$ mkdir CloudTech; cd CloudTech; python3
Python 3.9.16 (main, Feb 23 2023, 00:00:00)
[GCC 11.3.1 20221121 (Red Hat 11.3.1-4)] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import requests, json
>>> for val in ["usd", "eur"]:
...     url = f"https://bank.gov.ua/NBU_Exchange/exchange_site?start=20210101&end=20211231&valcode={val}&sort=exchangedate&order=desc&json"
...     response = requests.get(url)
...     if response.status_code == 200:
...         data = response.json()
...         with open(f"{val}_exchange_rates.json", "w") as fp:
...             json.dump(data, fp)
...             print(f"JSON {val}-data written to ./(val)_exchange_rates.json")
...
JSON usd-data written to ./usd_exchange_rates.json
JSON eur-data written to ./eur_exchange_rates.json
>>> quit()
[ec2-user@ip-172-31-31-220 CloudTech]$ ls -la
total 180
drwxr-xr-x. 2 ec2-user ec2-user 68 Apr 9 13:40 .
drwx----- 12 ec2-user ec2-user 16384 Apr 9 13:40 ..
-rw-r--r--- 1 ec2-user ec2-user 75090 Apr 9 13:40 eur_exchange_rates.json
-rw-r--r--- 1 ec2-user ec2-user 86030 Apr 9 13:40 usd_exchange_rates.json
```

2. Напишемо скрипт, що створить відповідний csv-файл з даними, конвертуючи отриманий json-файл з пункту 1:

```
[ec2-user@ip-172-31-31-220 CloudTech]$ nano script.py

ec2-user@ip-172-31-31-220:~/CloudTech

Файл  Дія  Редагувати  Вигляд  Допомога

GNU nano 5.8                                script.py
import json, csv
import boto3

for val in ["usd", "eur"]:
    json_file = f"{val}_exchange_rates.json"
    # Open the JSON file and load the data
    with open(json_file, "r") as json_filepath:
        data = json.load(json_filepath)

    csv_file = f"{val}_exchange_rates.csv"
    # Create the CSV file and write the data
    with open(csv_file, "w", newline="") as csv_filepath:
        writer = csv.writer(csv_filepath)

        # Write the headers
        headers = list(data[0].keys())
        writer.writerow(headers)
```

```
# Write the data
for row in data:
    writer.writerow(list(row.values()))

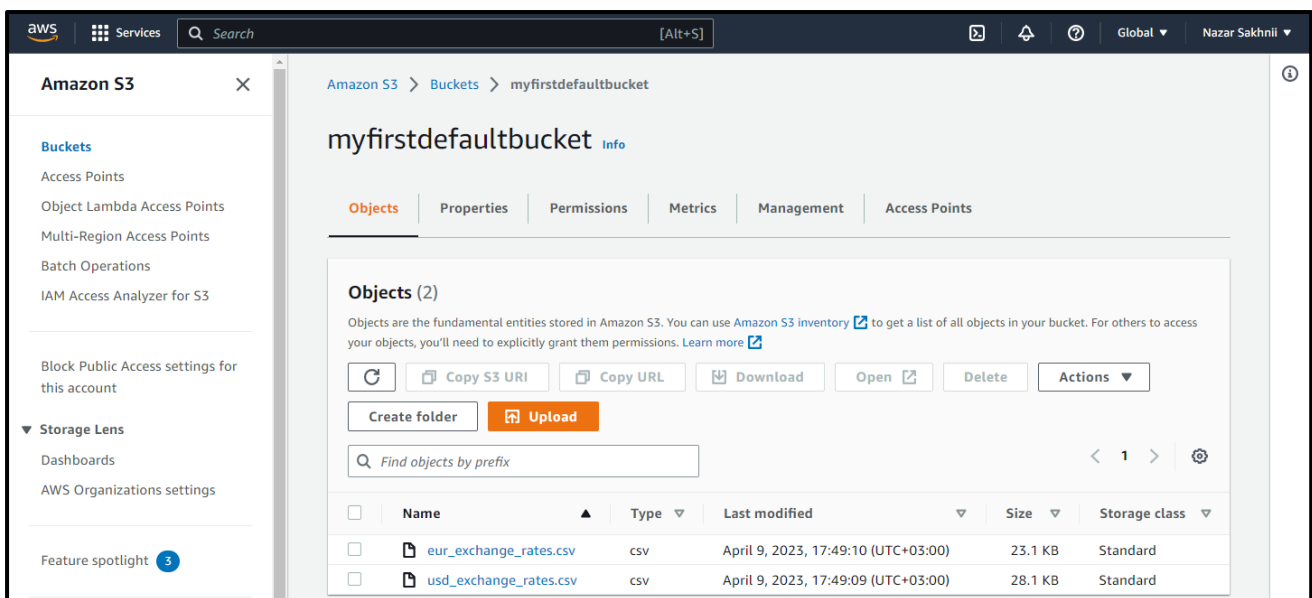
# Open a connection with Amazon S3
s3 = boto3.resource("s3")

# Upload the CSV file to S3
s3.Bucket("myfirstdefaultbucket").upload_file(csv_file, csv_file)

[ec2-user@ip-172-31-31-220 CloudTech]$ chmod +x ./script.py
[ec2-user@ip-172-31-31-220 CloudTech]$ python3 ./script.py
[ec2-user@ip-172-31-31-220 CloudTech]$ ls -la
total 240
drwxr-xr-x. 2 ec2-user ec2-user 145 Apr 9 14:49 .
drwx----- 12 ec2-user ec2-user 16384 Apr 9 13:40 ..
-rw-r--r--. 1 ec2-user ec2-user 23698 Apr 9 14:53 eur_exchange_rates.csv
-rw-r--r--. 1 ec2-user ec2-user 75090 Apr 9 13:40 eur_exchange_rates.json
-rwxr-xr-x. 1 ec2-user ec2-user 797 Apr 9 14:51 script.py
-rw-r--r--. 1 ec2-user ec2-user 28798 Apr 9 14:53 usd_exchange_rates.csv
-rw-r--r--. 1 ec2-user ec2-user 86030 Apr 9 13:40 usd_exchange_rates.json
```

3. Створені csv-файли мають бути програматично вивантажені на S3.

```
[ec2-user@ip-172-31-31-220 CloudTech]$ aws s3 ls s3://myfirstdefaultbucket
2023-04-09 14:53:16      23698 eur_exchange_rates.csv
2023-04-09 14:53:16      28798 usd_exchange_rates.csv
```



4. Розробимо скрипт для читання файлів з бакету та візуалізації курсу валют засобами Python.

```
jupyter jupyter Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [1]: import boto3
import pandas as pd
import matplotlib.pyplot as plt
from io import BytesIO

In [2]: # Connect to S3 bucket using boto3
s3 = boto3.client("s3")

# Download the USD exchange rates file
s3.download_file("myfirstdefaultbucket", "usd_exchange_rates.csv", "usd_exchange_rates.csv")

# Download the EUR exchange rates file
s3.download_file("myfirstdefaultbucket", "eur_exchange_rates.csv", "eur_exchange_rates.csv")
```

```
In [3]: # Read the USD exchange rate data from the CSV file
usd_exchange_rate_data = pd.read_csv(f"./usd_exchange_rates.csv", parse_dates=["exchangedate"], dayfirst=True)
usd_exchange_rate_data["exchangedate"] = pd.to_datetime(usd_exchange_rate_data["exchangedate"], format="%d.%m.%Y")
print(usd_exchange_rate_data)
```

	exchangedate	r030	cc	txt	enname	rate	units
0	2021-12-31	840	USD	Долар США	US Dollar	27.2782	1 \
1	2021-12-30	840	USD	Долар США	US Dollar	27.2077	1
2	2021-12-29	840	USD	Долар США	US Dollar	27.2235	1
3	2021-12-28	840	USD	Долар США	US Dollar	27.2306	1
4	2021-12-27	840	USD	Долар США	US Dollar	27.2747	1
...
360	2021-01-05	840	USD	Долар США	US Dollar	28.4310	1
361	2021-01-04	840	USD	Долар США	US Dollar	28.2746	1
362	2021-01-03	840	USD	Долар США	US Dollar	28.2746	1
363	2021-01-02	840	USD	Долар США	US Dollar	28.2746	1
364	2021-01-01	840	USD	Долар США	US Dollar	28.2746	1

```
In [4]: # Read the EUR exchange rate data from the CSV file
eur_exchange_rate_data = pd.read_csv(f"./eur_exchange_rates.csv", parse_dates=["exchangedate"], dayfirst=True)
eur_exchange_rate_data["exchangedate"] = pd.to_datetime(eur_exchange_rate_data["exchangedate"], format="%d.%m.%Y")
print(eur_exchange_rate_data)
```

	exchangedate	r030	cc	txt	enname	rate	units	rate_per_unit
0	2021-12-31	978	EUR	Євро	Euro	30.9226	1	30.9226 \
1	2021-12-30	978	EUR	Євро	Euro	30.7760	1	30.7760
2	2021-12-29	978	EUR	Євро	Euro	30.8401	1	30.8401
3	2021-12-28	978	EUR	Євро	Euro	30.8196	1	30.8196
4	2021-12-27	978	EUR	Євро	Euro	30.8450	1	30.8450
...
360	2021-01-05	978	EUR	Євро	Euro	34.9389	1	34.9389
361	2021-01-04	978	EUR	Євро	Euro	34.7396	1	34.7396
362	2021-01-03	978	EUR	Євро	Euro	34.7396	1	34.7396
363	2021-01-02	978	EUR	Євро	Euro	34.7396	1	34.7396
364	2021-01-01	978	EUR	Євро	Euro	34.7396	1	34.7396

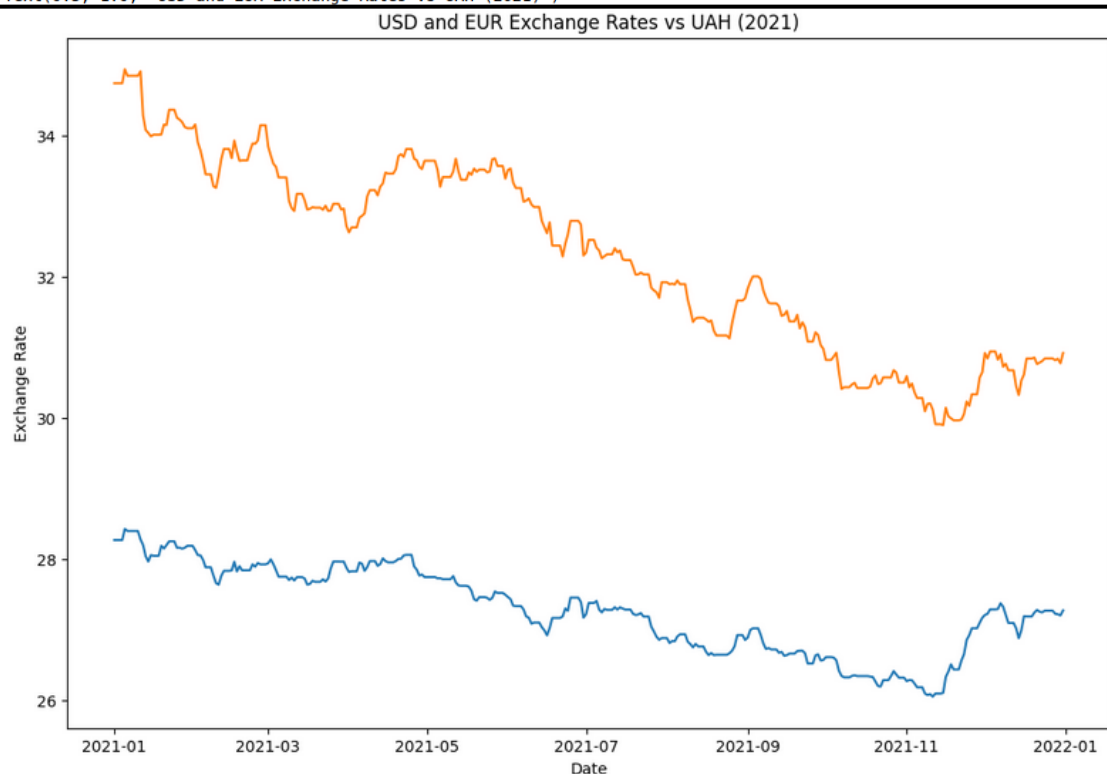
5. Побудуємо графік із курсом гривні щодо іноземних валют (Долар США та Євро) для 2021 року:

```
In [5]: # Plot the exchange rate data
fig, ax = plt.subplots(figsize=(12, 8))

plt.plot(usd_exchange_rate_data["exchangedate"], usd_exchange_rate_data["rate"], label="USD")
plt.plot(eur_exchange_rate_data["exchangedate"], eur_exchange_rate_data["rate"], label="EUR")

ax.set_xlabel("Date")
ax.set_ylabel("Exchange Rate")
ax.set_title("USD and EUR Exchange Rates vs UAH (2021)")
```

Out[5]: Text(0.5, 1.0, 'USD and EUR Exchange Rates vs UAH (2021)')



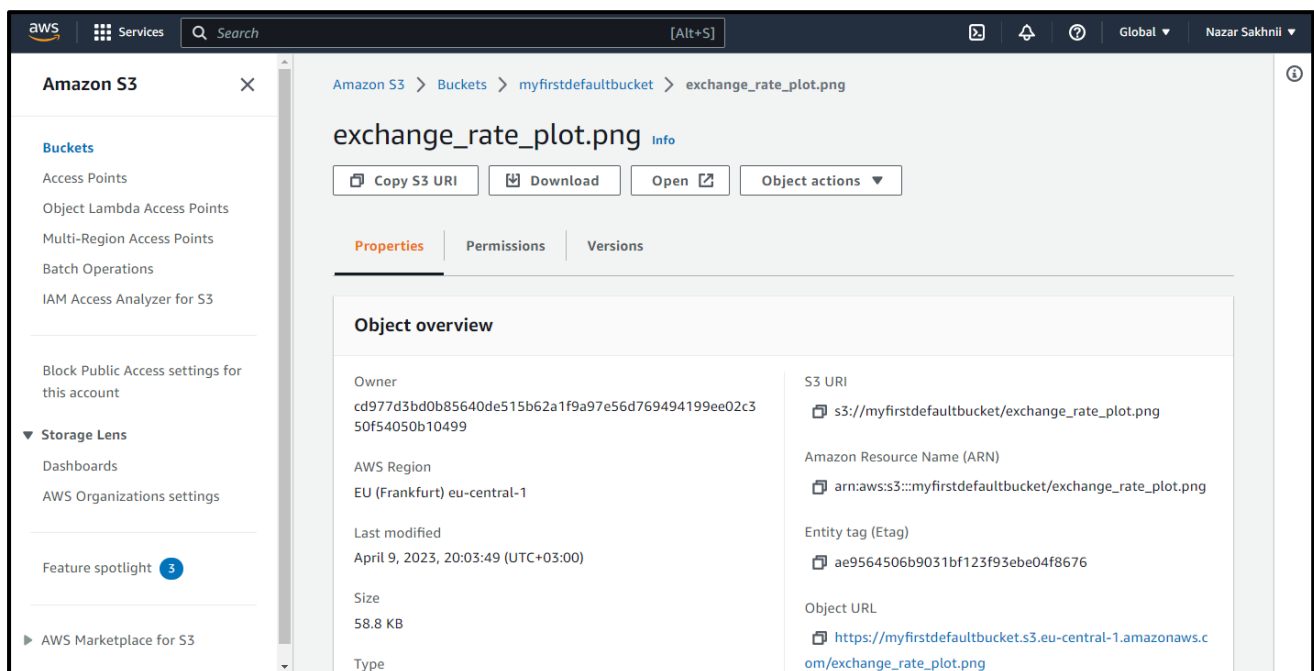
6. Збережемо побудований графік у бакет:

```
In [6]: # Save the plot image to a BytesIO buffer
img_buffer = BytesIO()
fig.savefig(img_buffer, format="png")
img_buffer.seek(0)

Out[6]: 0

In [7]: # Upload the plot image to S3 bucket
s3.put_object(Bucket="myfirstdefaultbucket", Key="exchange_rate_plot.png", Body=img_buffer.getvalue())

Out[7]: {'ResponseMetadata': {'RequestId': 'NABN24F72CR5T1F2',
'HostId': 'MBwktCzg7/KLS/uHBiSLXA5W93RAJWHpghuTtMSiLRQLPhviiyJDLoKh7T5xz+Lvupag+PbpQIPc=',
'HTTPStatusCode': 200,
'HTTPHeaders': {'x-amz-id-2': 'MBwktCzg7/KLS/uHBiSLXA5W93RAJWHpghuTtMSiLRQLPhviiyJDLoKh7T5xz+Lvupag+PbpQIPc=',
'x-amz-request-id': 'NABN24F72CR5T1F2',
'date': 'Sun, 09 Apr 2023 17:03:49 GMT',
'x-amz-server-side-encryption': 'AES256',
'etag': '"ae9564506b9031bf123f93ebe04f8676"',
'server': 'AmazonS3',
'content-length': '0'},
'RetryAttempts': 0},
'ETag': '"ae9564506b9031bf123f93ebe04f8676"',
'ServerSideEncryption': 'AES256'}
```



Висновки:

У ході виконання лабораторної роботи виникали невеликі питання, які було досить швидко вирішені після того, як я уважніше перечитав надані методичні матеріали та покрокову інструкцію до виконання лабораторної роботи. Якщо конкретно, то трохи не зрозумів спочатку, що доступ до бакету необхідно було налаштувати саме із уже створеного інстансу, а не з основної ОС.

Під час візуалізації даних, довелося трохи проглянути в інтернеті відповідні матеріали про методи взаємодії з великою кількістю даних та побудовою графіків. Та й основне для даної лабораторної роботи – це було вміння застосовувати основні команди із створення й налаштування AWS інстансів та S3 бакетів.