



Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Фізико-технічний інститут

## Комп'ютерний практикум №2 Використання AWS Simple Storage Service (S3)

предмет «Хмарні технології обробки даних»

**Роботу виконав:**

Студент 3 курсу ФТІ, групи ФІ-91  
Цибульник Антон Владиславович

**Приймали:**

Шелестов Андрій Юрійович  
Колотій Андрій Всеволодович

# Мета

Ознайомитись з використанням AWS Simple Storage Service (S3).

## Завдання

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

## Хід виконання роботи

### 0. Підготовчий етап

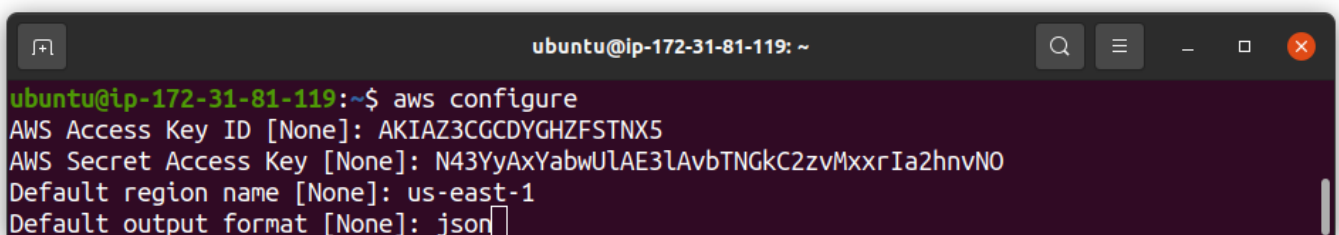
#### Створення бакету

Створення бакету S3 можливе різними способами. У цій роботі я користувався AWS Command Line Interface (AWS CLI). Спершу слід встановити власне засоби AWS CLI на інстанс, доступ до якого можна отримати, зробивши аналогічні кроки, як це зазначалося у 3 пункті Лабораторної роботи №1. Тепер встановимо AWS CLI, ввівши команду

```
sudo apt install awscli
```

Наступним кроком вкажемо усі необхідні для подальшої роботи параметри через команду `aws configure`. Для цього спочатку створимо окремого користувача на інстансі. Зробити це можна за допомогою розділу Identity and Access Management (IAM). Знайшовши цей розділ самостійно або перейшовши за [посиланням](#), послідовно обираємо вкладки «IAM resources» → «Users» → «Add users».

Крок за кроком проходимо п'ять етапів створення користувача, як це зображено на Рис. 2 та Рис. 3. На кінець отримуємо шукані параметри Access key ID та Secret access key новоствореного юзера. Скориставшись [інструкцією](#), нарешті виконуємо команду `aws configure` через консоль (Рис. 1).



```
ubuntu@ip-172-31-81-119: ~  
ubuntu@ip-172-31-81-119:~$ aws configure  
AWS Access Key ID [None]: AKIAZ3CGCDYGHZFSTNX5  
AWS Secret Access Key [None]: N43YyAxYabwU1AE31AvbTNGkC2zvMxxrIa2hvn0  
Default region name [None]: us-east-1  
Default output format [None]: json
```

Рис. 1: Виконання команди `aws configure`

## Add user



### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[+ Add another user](#)

### Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Select AWS credential type\*
- ☒ **Access key - Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
  - ☐ **Password - AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Рис. 2: Перший етап створення користувача

## Add user



### Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

#### User details

User name	tom
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

#### Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	<a href="#">AdministratorAccess</a>

#### Tags

...

Рис. 3: Передостанній етап створення користувача

Фінальним кроком цього підрозділу створюємо новий бакет (Рис. 4) консольною командою

```
aws s3 mb s3://bucket01my
```

Слід зауважити, що ім'я бакету має бути унікальним. Знайшовши розділ «S3» самостійно або перейшовши за [посиланням](#), переконуємося, що бакет створений і є у переліку (Рис. 5).

```
ubuntu@ip-172-31-81-119: ~  
ubuntu@ip-172-31-81-119:~$ aws s3 mb s3://bucket01my  
make_bucket: bucket01my  
ubuntu@ip-172-31-81-119:~$
```

Рис. 4: Створення бакету

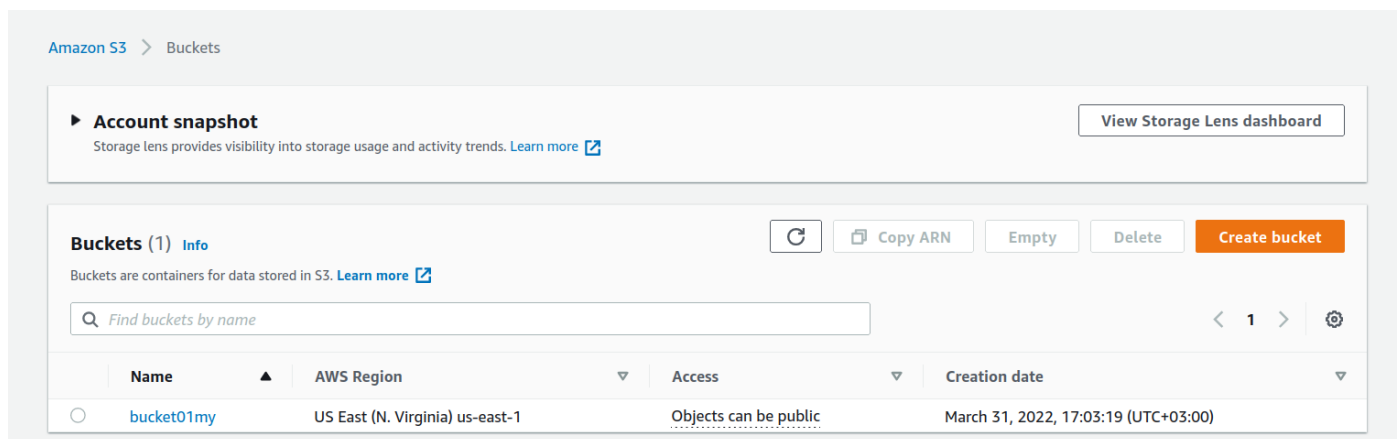


Рис. 5: Перелік бакетів

## Початок роботи з jupyter notebook

Для налаштування Python API потрібно встановити пакетний менеджер pip. Після оновлення дерева пакетів виконуємо команди

```
sudo apt update  
sudo apt install python3-pip
```

Процес завантаження зображено на Рис. 6. Надалі завантажуюмо на інстанс власне jupyter notebook:

```
sudo pip3 install jupyter notebook
```

Якщо виникли проблеми із цілісним завантаженням jupyter notebook, варто спробувати запустити іншу команду встановлення:

```
sudo pip3 install --upgrade --force-reinstall --no-cache-dir jupyter
```

Постає завдання: запустити щойно встановлений jupyter notebook на інстансі та працювати з ним з локального робочого місця. Виконуємо команди (як на Рис. 8):

```
jupyter notebook --no-browser --port 8889          на інстансі  
ssh -i lab.pem -N -f -L localhost:8888:localhost:8889  локально
```

```
ubuntu@ip-172-31-81-119: ~  
ubuntu@ip-172-31-81-119:~$ pip  
Command 'pip' not found, but can be installed with:  
sudo apt install python3-pip  
ubuntu@ip-172-31-81-119:~$ sudo apt install python3-pip
```

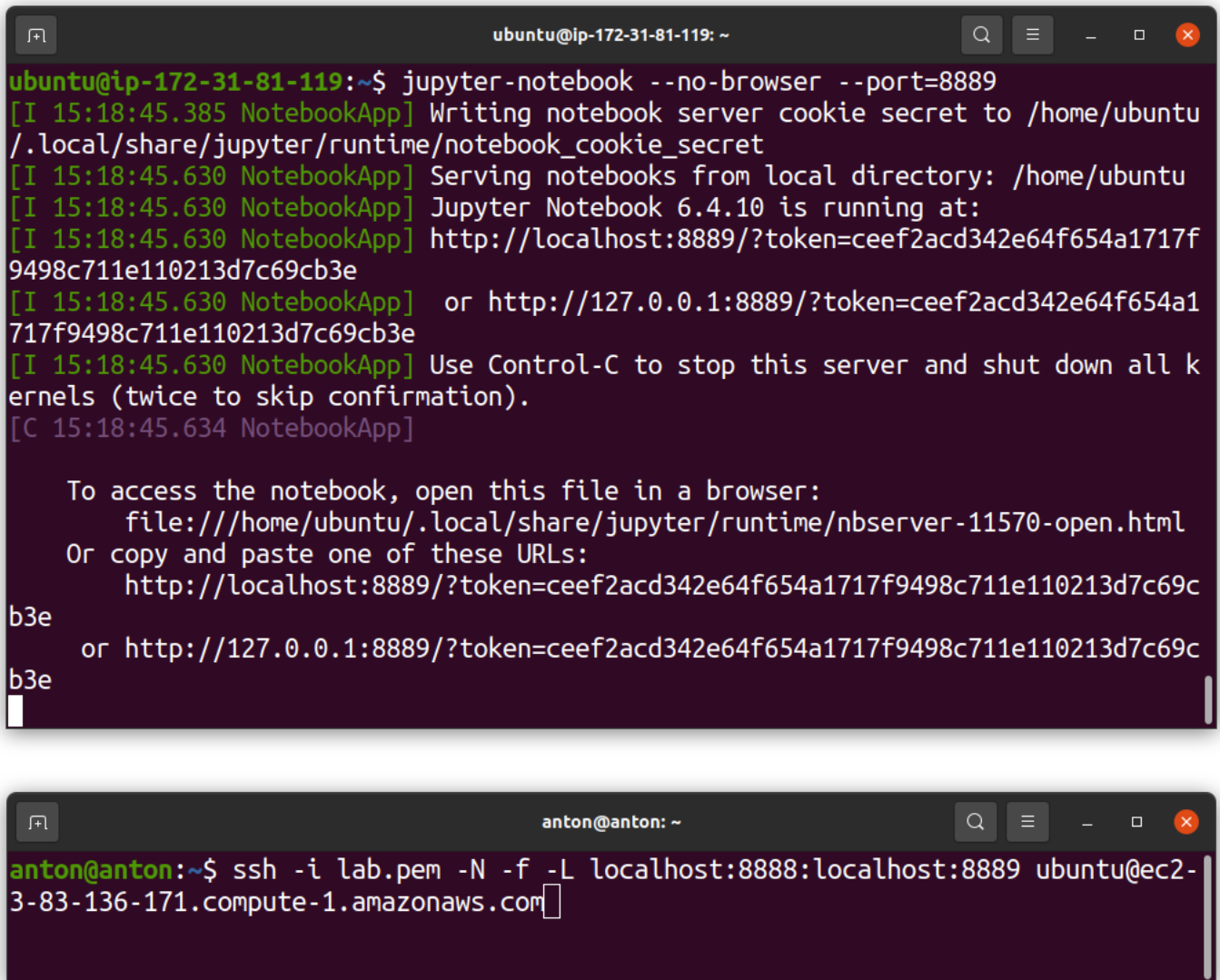
```
ubuntu@ip-172-31-81-119: ~  
Selecting previously unselected package libctf0:amd64.  
Preparing to unpack .../08-libctf0_2.34-6ubuntu1.3_amd64.deb ...  
Unpacking libctf0:amd64 (2.34-6ubuntu1.3) ...  
Selecting previously unselected package binutils-x86-64-linux-gnu.  
Preparing to unpack .../09-binutils-x86-64-linux-gnu_2.34-6ubuntu1.3_amd64.deb ...  
Unpacking binutils-x86-64-linux-gnu (2.34-6ubuntu1.3) ...  
Selecting previously unselected package binutils.  
Preparing to unpack .../10-binutils_2.34-6ubuntu1.3_amd64.deb ...  
Unpacking binutils (2.34-6ubuntu1.3) ...  
Selecting previously unselected package libc-dev-bin.  
Preparing to unpack .../11-libc-dev-bin_2.31-0ubuntu9.7_amd64.deb ...  
Unpacking libc-dev-bin (2.31-0ubuntu9.7) ...  
Selecting previously unselected package linux-libc-dev:amd64.  
Preparing to unpack .../12-linux-libc-dev_5.4.0-107.121_amd64.deb ...  
Unpacking linux-libc-dev:amd64 (5.4.0-107.121) ...  
Selecting previously unselected package libcrypt-dev:amd64.  
Preparing to unpack .../13-libcrypt-dev_1%3a4.4.10-10ubuntu4_amd64.deb ...  
Unpacking libcrypt-dev:amd64 (1:4.4.10-10ubuntu4) ...  
Selecting previously unselected package libc6-dev:amd64.  
Preparing to unpack .../14-libc6-dev_2.31-0ubuntu9.7_amd64.deb ...  
Unpacking libc6-dev:amd64 (2.31-0ubuntu9.7) ...  
Preparing to unpack .../15-cpp-9_9.4.0-1ubuntu1~20.04.1_amd64.deb ...  
Unpacking cpp-9 (9.4.0-1ubuntu1~20.04.1) over (9.3.0-17ubuntu1~20.04) ...  
Progress: [ 18%] [#####.....]
```

Рис. 6: Встановлення `pip`

Завантаживши `pip` (Рис. 6) й встановивши віддалений доступ до `jupyter notebook` (Рис. 8), переходимо за посиланням <http://localhost:8888/> й вказуємо `token`. Відтак відкриваємо дерево папок й створюємо новий файл `Lab2.ipynb` (Рис. 7).



Рис. 7: Сховище `jupyter notebook`



The image contains two terminal window screenshots. The top window, titled 'ubuntu@ip-172-31-81-119: ~', shows the command 'jupyter-notebook --no-browser --port=8889' being executed. It displays several status messages from the 'NotebookApp' including writing the cookie secret, serving notebooks from the local directory, and providing the URL 'http://localhost:8889/?token=ceef2acd342e64f654a1717f9498c711e110213d7c69cb3e'. It also provides an alternative URL 'http://127.0.0.1:8889/?token=ceef2acd342e64f654a1717f9498c711e110213d7c69cb3e' and instructions to use Control-C to stop the server. The bottom window, titled 'anton@anton: ~', shows the command 'ssh -i lab.pem -N -f -L localhost:8888:localhost:8889 ubuntu@ec2-3-83-136-171.compute-1.amazonaws.com' being entered.

```
ubuntu@ip-172-31-81-119:~$ jupyter-notebook --no-browser --port=8889
[I 15:18:45.385 NotebookApp] Writing notebook server cookie secret to /home/ubuntu/.local/share/jupyter/runtime/notebook_cookie_secret
[I 15:18:45.630 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 15:18:45.630 NotebookApp] Jupyter Notebook 6.4.10 is running at:
[I 15:18:45.630 NotebookApp] http://localhost:8889/?token=ceef2acd342e64f654a1717f9498c711e110213d7c69cb3e
or http://127.0.0.1:8889/?token=ceef2acd342e64f654a1717f9498c711e110213d7c69cb3e
[I 15:18:45.630 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 15:18:45.634 NotebookApp]

To access the notebook, open this file in a browser:
    file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-11570-open.html
Or copy and paste one of these URLs:
    http://localhost:8889/?token=ceef2acd342e64f654a1717f9498c711e110213d7c69cb3e
or http://127.0.0.1:8889/?token=ceef2acd342e64f654a1717f9498c711e110213d7c69cb3e

anton@anton:~$ ssh -i lab.pem -N -f -L localhost:8888:localhost:8889 ubuntu@ec2-3-83-136-171.compute-1.amazonaws.com
```

Рис. 8: Віддалений доступ до jupyter notebook

## 1. Отримання даних щодо курсу гривні

Завданням етапу є програматичне отримання даних у JSON-форматі на інстанс, а саме: інформації щодо курсу гривні за 2021 рік. Перейшовши за [посиланням](#), опиняємося на сторінці Національного банку України. Послідовно обираємо

- «API для розробників»
- «1. Офіційний курс гривні до іноземних валют ... »
- «Інструкція до сервісу отримання курсів гривні до іноземних валют ... »

Клікаємо на посилання, яке має вид <https://bank.gov.ua/NBUStatService/v1/statdirectory/exchangenew?json>, маючи змогу переглянути таким чином дані про курс гривні. Додаючи до посилання фрагмент `&date=20210110` можна отримати інформацію про курс на конкретну дату (наприклад, на 10.01.2021). Наступним кроком буде програматичний етап: зчитування знайдених даних засобами Python через хмарний jupyter notebook.

Підключимо такі дві бібліотеки: `requests` для роботи із запитами та `json` для перетворення й збереження відповідного формату файлів. За допомогою методу `request.get()` отримаємо дані з сайту, поступово крок за кроком зчитуючи й записуючи інформацію про 10 число кожного місяця (Лістинг 1, рядки 4-16).

Лістинг 1: Отримання даних

```
1 import requests as request
2 import json
3
4 response = request.get(
5     "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchangenew?json" +
6     "&date=20210110")
7 json_data = response.json()
8
9 for i in range(2,13):
10     if (i < 10):
11         response = request.get(
12             "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchangenew?json" +
13             "&date=20210" + str(i) + "10")
14         json_data = json_data + response.json()
15     else:
16         response = request.get(
17             "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchangenew?json" +
18             "&date=2021" + str(i) + "10")
19         json_data = json_data + response.json()
```

Наступним кроком перетворимо отриману інформацію з `json` рядка у `json` файл. Тобто, фактично, завантажимо всі дані на інстанст у вигляді файлу, який назовемо `request.json` (Лістинг 2, рядки 18-21).

Для перевірки візулізуємо таблицю даних за допомогою бібліотеки `pandas`, завантаження якої зображено на Рис. 9. Отримали таблицю із спільною інформацією про усі курси валют для кожного із дванадцяти місяців. Результати самої візуалізації можна побачити на Рис. 10, а саме: перші й останні шість рядків.

Лістинг 2: Завантаження даних

```
18 jsonString = json.dumps(json_data, ensure_ascii=False, indent = 4)
19 jsonFile = open("request.json", "w")
20 jsonFile.write(jsonString)
21 jsonFile.close()
22
23 import pandas as pd
24
25 df = pd.read_json("request.json")
26 print(df.head(6), "\n")
27 print(df.tail(6))
```



```
In [2]: pip install pandas

Collecting pandas
  Downloading pandas-1.4.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
    |████████████████████████████████████████| 11.7 MB 15.5 MB/s eta 0:00:01
    |████████████████████████████████████████| 6.2 MB 15.5 MB/s eta 0:00:01
5.5 MB/s eta 0:00:01
    |████████████████████████████████████████| 8.2 MB 15.5 MB/s eta 0:00:01
Collecting pytz>=2020.1
  Downloading pytz-2022.1-py2.py3-none-any.whl (503 kB)
    |████████████████████████████████████████| 503 kB 33.3 MB/s eta 0:00:01
Collecting numpy>=1.18.5; platform_machine != "aarch64" and platform_machine != "arm64" and python_version < "3.10"
  Downloading numpy-1.22.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.8 MB)
    |████████████████████████████████████████| 16.8 MB 27.1 MB/s eta 0:00:01
Requirement already satisfied: python-dateutil>=2.8.1 in ./local/lib/python3.8/site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in ./local/lib/python3.8/site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.22.3 pandas-1.4.1 pytz-2022.1
Note: you may need to restart the kernel to use updated packages.
```

Рис. 9: Завантаження бібліотеки pandas

	r030	txt	rate	cc	exchangedate		r030	txt	rate	cc	exchangedate
0	36	Австралійський долар	21.8659	AUD	10.01.2021	717	981	Ларі	8.7225	GEL	10.12.2021
1	124	Канадський долар	22.2610	CAD	10.01.2021	718	986	Бразильський реал	4.8554	BRL	10.12.2021
2	156	Юань Женьмінбї	4.3944	CNY	10.01.2021	719	959	Золото	48163.3500	XAU	10.12.2021
3	191	Куна	4.6098	HRK	10.01.2021	720	961	Срібло	595.3300	XAG	10.12.2021
4	203	Чеська крона	1.3286	CZK	10.01.2021	721	962	Платина	25553.2700	XPT	10.12.2021
5	208	Данська крона	4.6847	DKK	10.01.2021	722	964	Паладій	48795.8900	XPD	10.12.2021

Рис. 10: Перші й останні рядки таблиці даних

## 2. Конвертація json-файлу у csv-файл

Для поставленого завдання виконаємо ряд команд (Лістинг 3), які дозволять власне конвертувати файл й зберегти його на інстансі. Перевірку збереження можна провести, відкривши сховище jupyter notebook (Рис 7).

Лістинг 3: Конвертація файлу

```
29 df = pd.read_json (r"request.json")
30 df.to_csv (r"request.csv", index = None)
```

## 3. Вивантаження інформації на S3

Вивантажимо отриманий файл `request.csv` з інстансу на бакет. Зробимо це за допомогою команди `cp`, яка є одним з інструментів **AWS CLI**. Насамперед завершимо сеанс підключення до хмарного ноутбуку, натиснувши в командному рядку, через який відбувається доступ до інстансу, комбінацію клавіш **Ctrl+C** й ввівши **yes** у діалоговому рядку. Тож після підготовчих кроків виконуємо команду

```
aws s3 cp request.csv s3://bucket01my
```

Перевіримо успішне виконання команди, обравши назву свого бакету (як на Рис. 5) серед списку усіх бакетів. Результат перевірки зображено на Рис. 11.



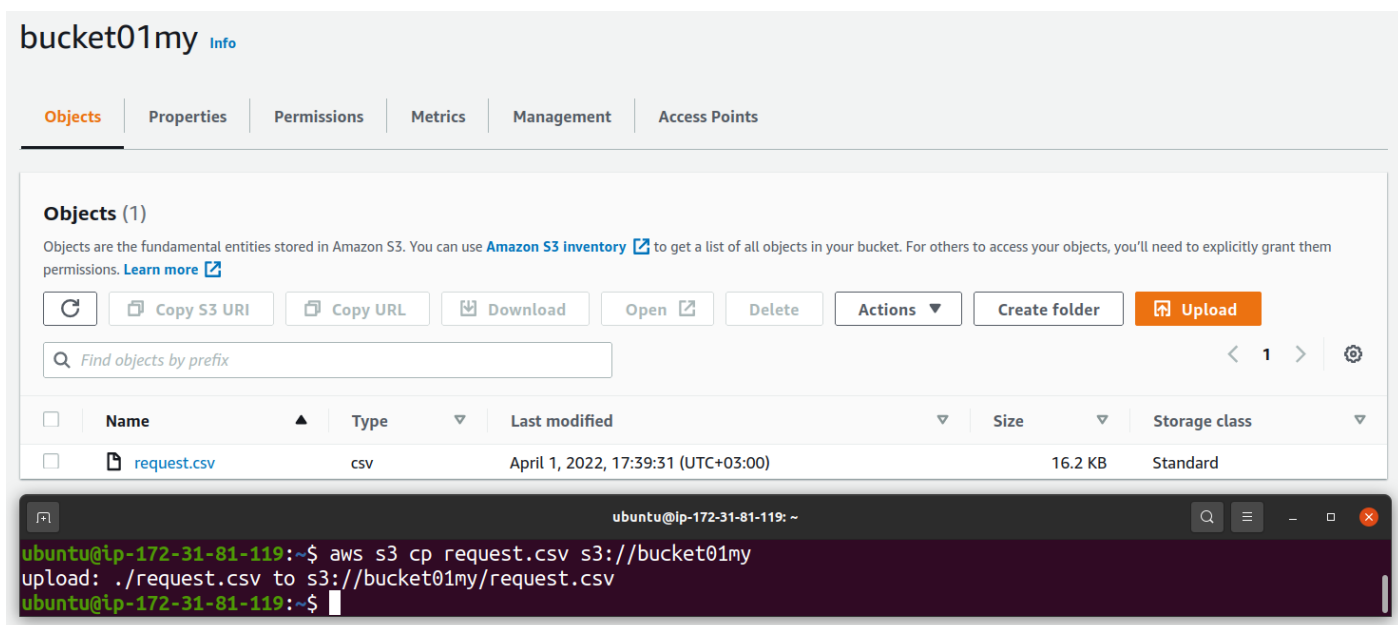


Рис. 11: Вивантаження інформації на S3

## 4. Скрипт для читання файлів з бакету

Наразі на бакеті маємо завантажений файл `request.csv`, який зберігає дані про курс гривні за дванадцять місяців. Маємо намір зчитувати безпосередньо з бакету дані й працювати з ними на jupyter notebook.

В нагоді стануть засоби мови Python. Тому перш за все знову підключаємося до хмарного ноутбуку (як це було зазначено на Рис. 8). Наступним кроком завантажимо бібліотеку `boto3` (аналогічним чином, як було завантажено бібліотеку `pandas` на Рис. 9). Програмний етап можна побачити на Лістингу 4.

Останнім кроком перевіримо успішне виконання поставленого завдання, візуалізуювши зчитану з файлу таблицю даних (Лістинг 4, рядок 39). Результат має бути аналогічним до таблиці на Рис. 10.

Лістинг 4: Зчитування даних з бакету

```
32 import boto3
33
34 s3 = boto3.client("s3")
35
36 obj = s3.get_object(Bucket="bucket01my", Key="request.csv")
37
38 df_from_s3 = pd.read_csv(obj["Body"])
39 print(df_from_s3)
```

## 5. Графіки курсу гривні

Для малювання графіків застосуємо створену якраз для подібних цілей бібліотеку `matplotlib`. Попередньо завантаживши її, напишемо скрипт (Лістинг 5).

## Лістинг 5: Графіки курсу гривні

```
41 import matplotlib.pyplot as plt
42
43 plt.figure(figsize=(14, 4.5))
44
45 def plotting(currency, color):
46     x = []
47     y = []
48     month = 1
49
50     for i in df_from_s3.index:
51         cc = currency
52         if (month < 10): date = "10.0" + str(month) + ".2021"
53         else: date = "10." + str(month) + ".2021"
54
55         if df_from_s3.loc[i,"cc"] == cc and df_from_s3.loc[i,"exchangedate"] == date:
56             x.append(month)
57             y.append(df_from_s3.loc[i,"rate"])
58             month = month + 1
59
60     plt.xlabel("Monthes")
61     plt.ylabel("Values of " + currency)
62     plt.grid()
63     plt.title(currency + " to UAH")
64     plt.plot(x,y,color=color)
65
66 plt.subplot(1,2,1)
67 plotting("USD", "green")
68 plt.subplot(1,2,2)
69 plotting("EUR", "blue")
70
71 plt.savefig("exchange_rate.png")
72 plt.show()
```

Наведемо ключове пояснення до написаної вище програми: пробігаючи циклом `for` по масиву індексів (тобто покроково по кожному рядку таблиці даних), висмикуємо в окремий масив інформацію про курс гривні з тих рядків, які мають потрібну нам валюту долара «USD» чи євро «EUR» й вказану дату (10 число кожного місяця). Надалі саме з новостворених масивів поточною малюємо графічні залежності.

Технічно реалізацію виконано за допомогою таких ключових елементів:

<code>df.index</code>	масив індексів усіх рядків
<code>df.loc[i]</code>	атрибут для пробігу по <i>i</i> -му рядку
<code>plt.figure()</code>	встановлення розмірів зображення
<code>plt.subplot()</code>	поєднання двох графіків на одному малюнку
<code>plt.grid()</code>	розмірна сітка

Результати етапу зафіксовано на Рис. 12.

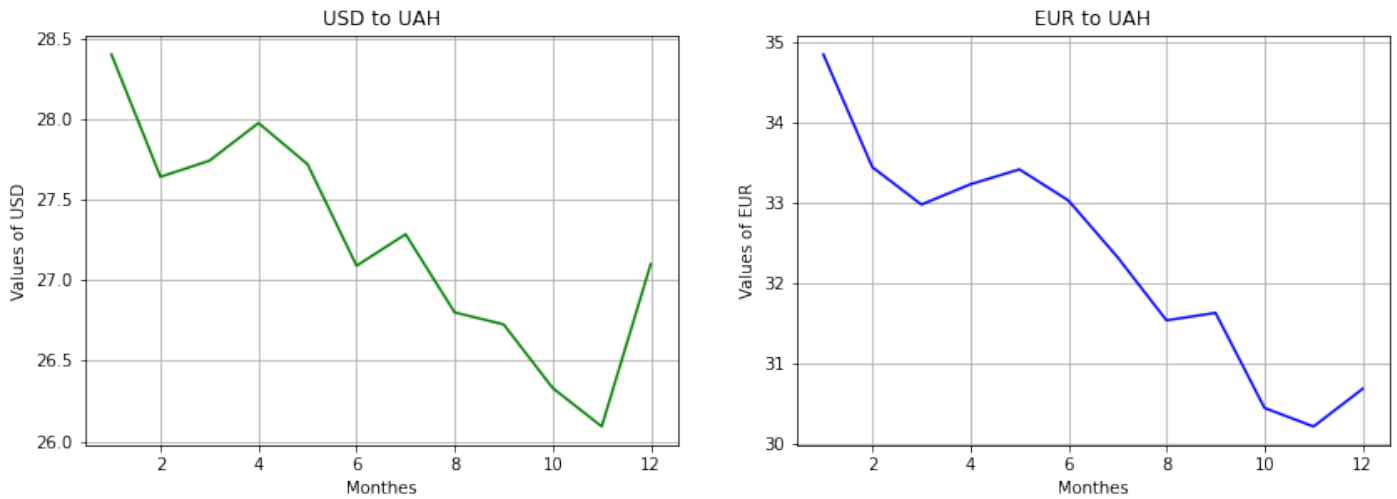


Рис. 12: Результати візуалізації курсу гривні

## 6. Збереження графіків на бакет

Перш за все завантажимо отриманий у попередньому пункті графік на інстанс. Зробимо це за допомогою команди, наведеної на Лістингу 5, рядок 71. Відкривши сховище jupyter notebook, маємо переконатися в успішності виконання.

Наступним етапом повторимо усі кроки пункту 3 (сторінка 7), щоб отриманий графік вивантажити з інстансу на бакет. Щойно зазначені кроки послідовно зображено на малюнку нижче (Рис. 13). Остаточний результат виконання шостого пункту зазначено на Рис. 14.

```

ubuntu@ip-172-31-81-119: ~
^C[I 14:02:22.199 NotebookApp] interrupted
Serving notebooks from local directory: /home/ubuntu
1 active kernel
Jupyter Notebook 6.4.10 is running at:
http://localhost:8889/?token=06238f0f509df3fee8e54a76349e3747367170ce84c147f1
or http://127.0.0.1:8889/?token=06238f0f509df3fee8e54a76349e3747367170ce84c147f
1
Shutdown this notebook server (y/[n])? y
[C 14:02:23.489 NotebookApp] Shutdown confirmed
[I 14:02:23.489 NotebookApp] Shutting down 1 kernel
[I 14:02:23.490 NotebookApp] Kernel shutdown: 0b754db5-6c00-4713-954b-8e153c54f1
4f
[I 14:02:23.913 NotebookApp] Shutting down 0 terminals
ubuntu@ip-172-31-81-119:~$ aws s3 cp exchange_rate.png s3://bucket01my
upload: ./exchange_rate.png to s3://bucket01my/exchange_rate.png
ubuntu@ip-172-31-81-119:~$

```

Рис. 13: Вивантаження малюнка на бакет

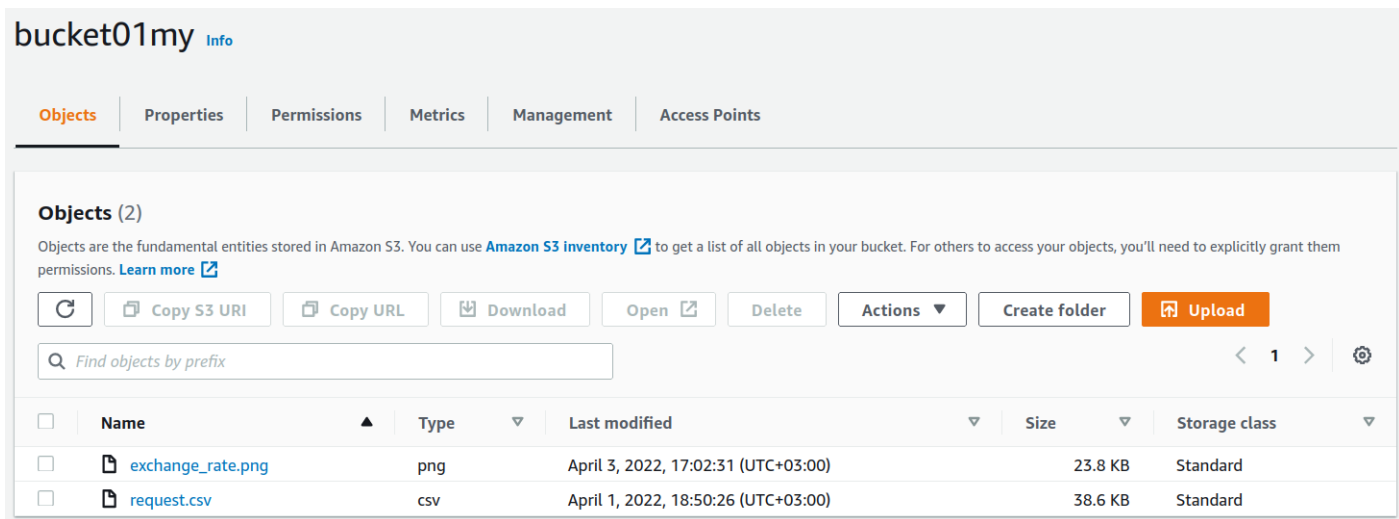


Рис. 14: Результат вивантаження малюнка на бакет

## 7. Перелік проблем протягом виконання роботи

Пункт 0: (а) При створенні бакету інструментами AWS CLI лише з п'ятої-десятої спроби вдалося підібрати унікальне ім'я новоствореного бакету. Крім того, наскільки я зрозумів, деякі імена були неприйнятними через надто велику довжину символів;

(б) Як вже зазначалося при описі підготовчого кроку, у мене виникла проблема із цілісним завантаженням jupyter notebook. На жаль, після застосування звичайної команди завантаження, наведеної в методичних вказівках до лабораторної роботи, не вдалося віднайти усі пакети. Як я довідався з відповідей на сайті <https://stackoverflow.com>: «It seems to me as though the installation has messed up somehow.» Проблему вирішило застосування команди

```
sudo pip3 install -upgrade -force-reinstall -no-cache-dir jupyter
```

(в) При спробі встановити віддалений зв'язок з хмарним ноутбуком з локального сховища (тобто при створенні SSH-тунелю), рекомендації в методичці для операційної системи Linux вказували додати компонент виду `ubuntu@54.71.56.14` до основної команди. На жаль, результуюча команда просто не виконувалася. Натомість довелося скористатися вказівками до операційної системи Windows, де рекомендовалося додати рядок виду `ubuntu@Public_IPv4_adress`.

Пункт 1: Під час початку роботи із jupyter notebook постійно стикався з проблемою неправильного кодування json рядків (стосується в основному Лістингу 1). Компілятор видавав повідомлення: «JSONDecodeError». Час від часу, перезапускаючи jupyter notebook, цю проблему вдавалося оминати й усі рядки компілювалися справно.

Пункт 3: Замість запропонованих в методичці інструментів бібліотеки `boto3`, вирішив скористатися простішою на вигляд командою `cp` інструментів AWS CLI. Із цим була пов'язана одна проблема: при повторній спробі вивантажити на бакет певний файл, довелося додатково оновити усі інструменти командою

```
pip install --upgrade awscli
```

Пункт 5: Основна складність, яка виникла з кодом на Лістингу 5, полягала у спробах віднайти спосіб доступу до рядків таблиці. Інакше прорідити дані й відсіяти непотрібні здавалося неможливим. Проблема вирішив знайдений метод `df.loc[i]`. Решта труднощів з кодом полягала у необхідності додаткового встановлення бібліотек `pandas`, `boto3` та `matplotlib`.