

Міністерство освіти та науки України

Національний університет водного господарства та  
природокористування

Кафедра автоматизації, електротехнічних та комп'ютерно-  
інтегрованих технологій

**04-03-439М**

## **МЕТОДИЧНІ ВКАЗІВКИ**

до виконання лабораторних робіт з дисципліни **«Хмарні технології та Big Data»** для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» денної та заочної форм навчання

Рекомендовано науково-методичною  
радою з якості ННІ ЕАВГ  
Протокол № 6 від 28.01.2025 р.

Рівне – 2025

Методичні вказівки до виконання лабораторних робіт з дисципліни «**Хмарні технології та Big Data**» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» денної та заочної форм навчання. [Електронне видання] / Присяжнюк О. В. – Рівне : НУВГП, 2025. – 46 с.

Укладач: Присяжнюк О. В., к.т.н., доцент кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Відповідальний за випуск: Древецький В. В., д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Керівник освітньої програми «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»: Христюк А. О., к.т.н., доцент кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

© О. В. Присяжнюк, 2025  
© НУВГП, 2025

## Зміст

Лабораторна робота №1. Використання AWS Management Console.....	4
Лабораторна робота №2. Використання AWS Simple Storage Service (S3).....	20
Лабораторна робота №3. Програмаіичний доступ до AWS Simple Storage Service (S3).....	25
Лабораторна робота №4. Використання AWS DynamoDB.....	34

# Лабораторна робота №1

## Використання AWS Management Console

**Мета роботи:** отримати базові навички по використанню AWS Management Console та створити власний мікро-сервер для подальшого використання.

### 1.1. План виконання

- Реєстрація в AWS
- Створення власного віртуального мікро-сервера
- Отримання віддаленого доступу через SSH
- Вивчення елементів моніторингу серверу та налаштування
- Документування зробленої роботи у вигляді деталізованого звіту з коментарями

### 1.2. Порядок виконання роботи

Для початку потрібно зареєструватись у AWS за посиланням <https://portal.aws.amazon.com/billing/signup#/start>

portal.aws.amazon.com/billing/signup#/start/email

aws

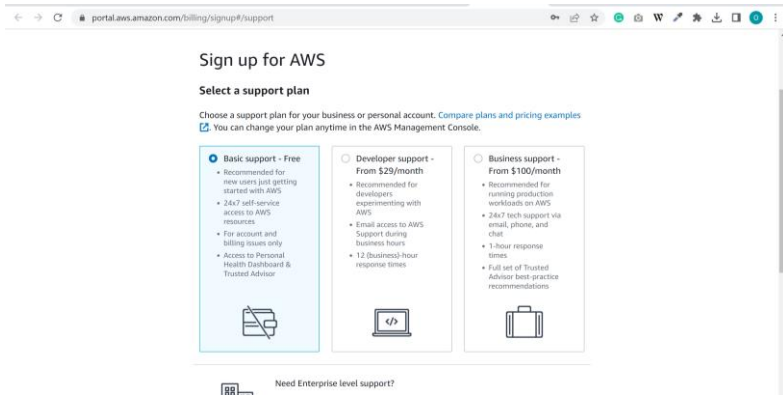
Sign up for AWS

Root user email address  
Used for account recovery and some administrative functions.

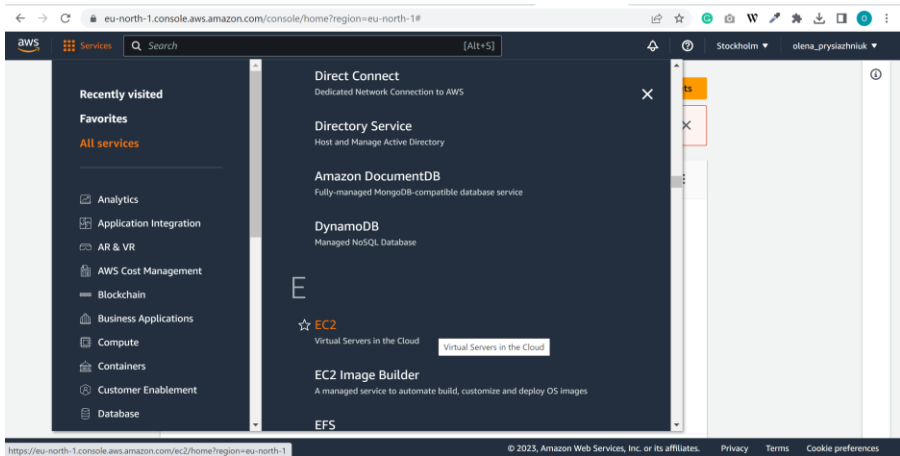
AWS account name  
Choose a name for your account. You can change this name in your account settings after you sign up.

OR

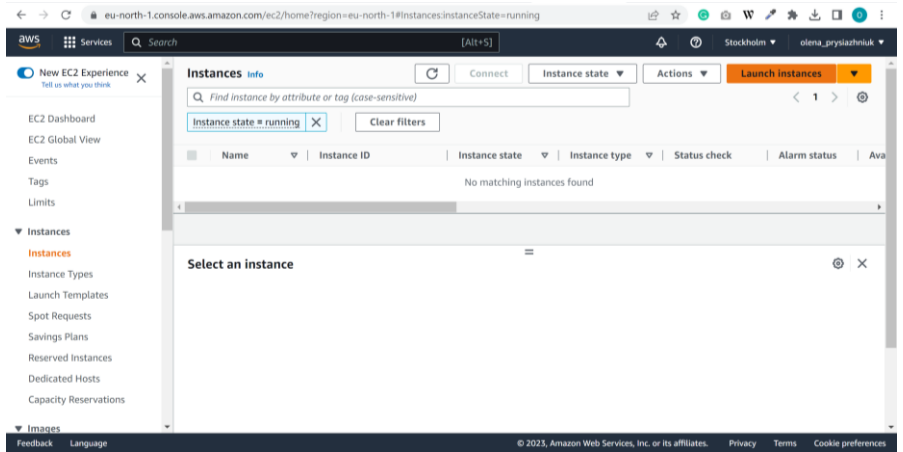
Для реєстрації потрібно вказати кредитку картку, з якої Amazon спише та поверне 1\$. При виборі тарифного плану вкажіть Basic.



Після реєстрації має бути доступною Amazon Management Console з цілим спектром сервісів. В даній роботі нас цікавить EC2 (Elastic Computing Service)



З сервісу EC2 маємо можливість запустити новий віртуальний сервер (інстанс) – помаранчева кнопка Launch Instance.



При створенні вказуйте опцію Free tier Only щоб уникнути плати за ресурси. Після завершення створення інстансу Вам буде запропоновано створити пару ключів RSA та зберегти їх.

### Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

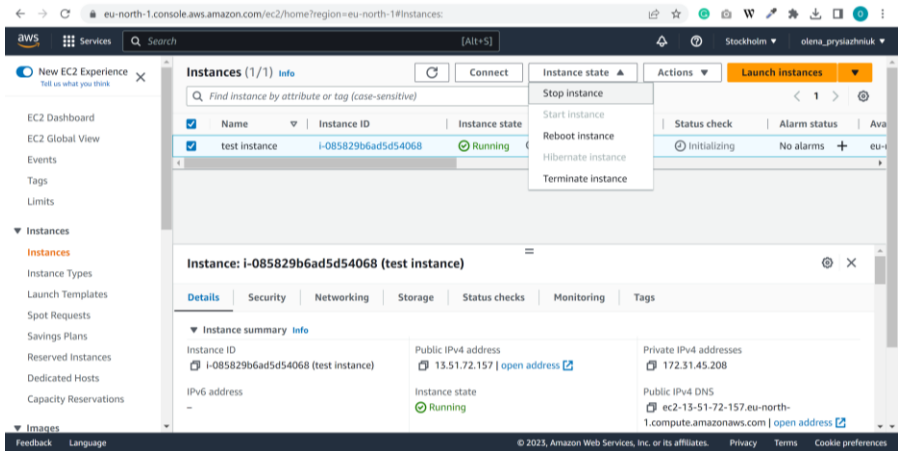
Create a new key pair ▾

**Key pair name**

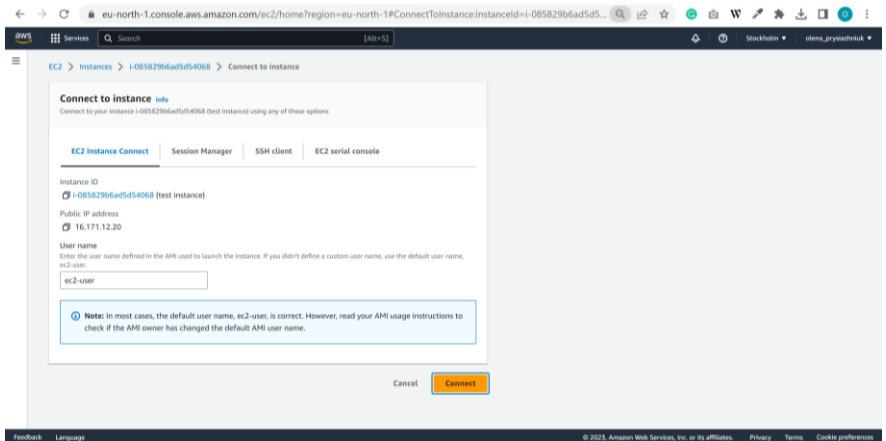
  

**You have to download the private key file (\*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.**

Після успішного створення інстансу він з'явиться у списку активних.



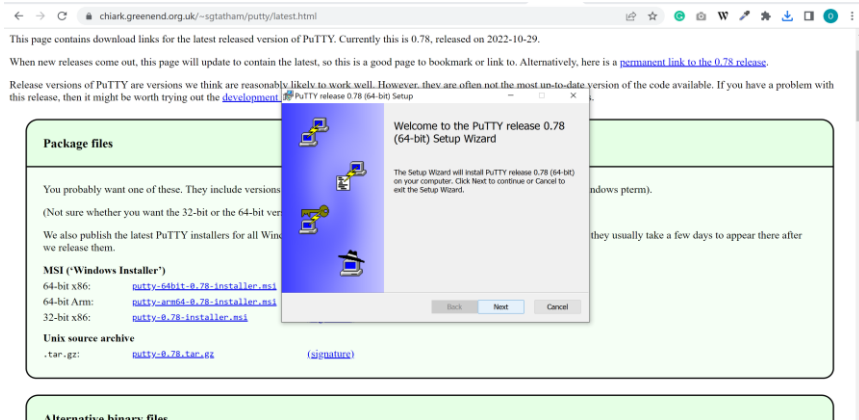
Доступ до інстансу можна отримати прямо з браузера, через ssm manager та SSH.



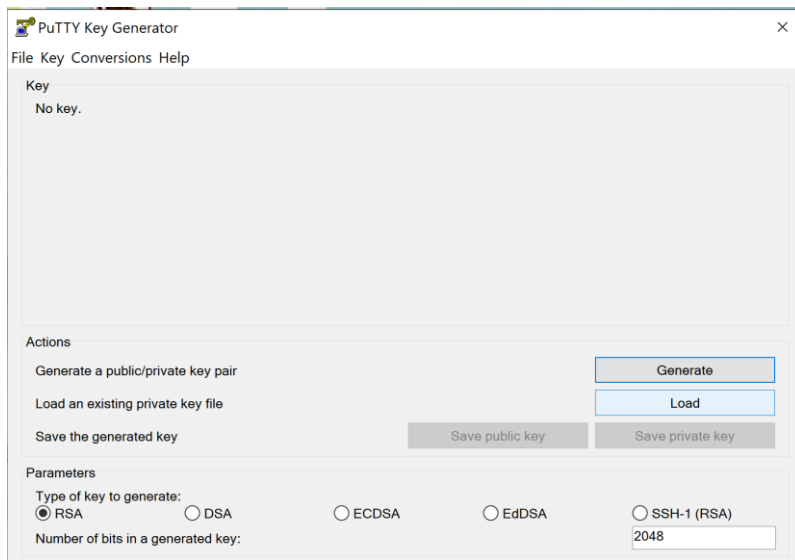
Після натискання кнопки Connect в новій вкладці відкриється вікно з'єднання.





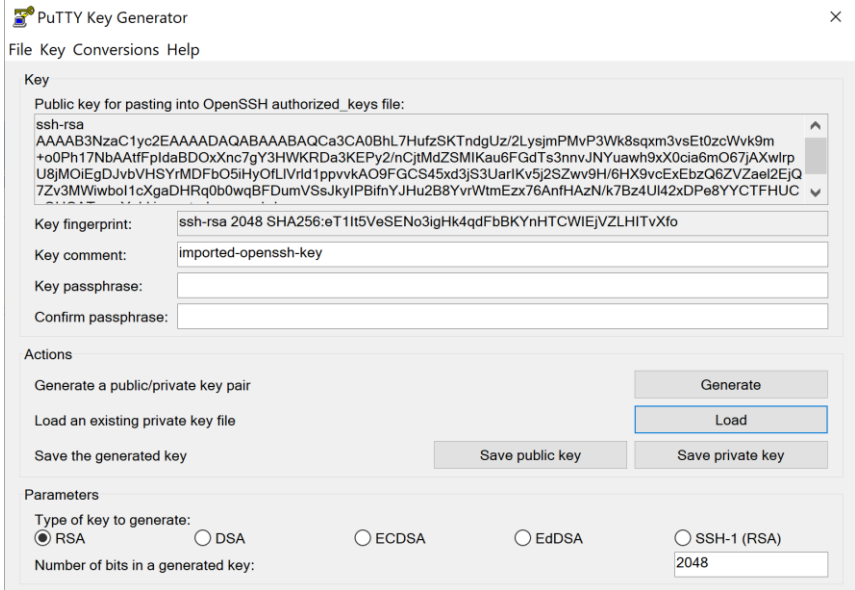
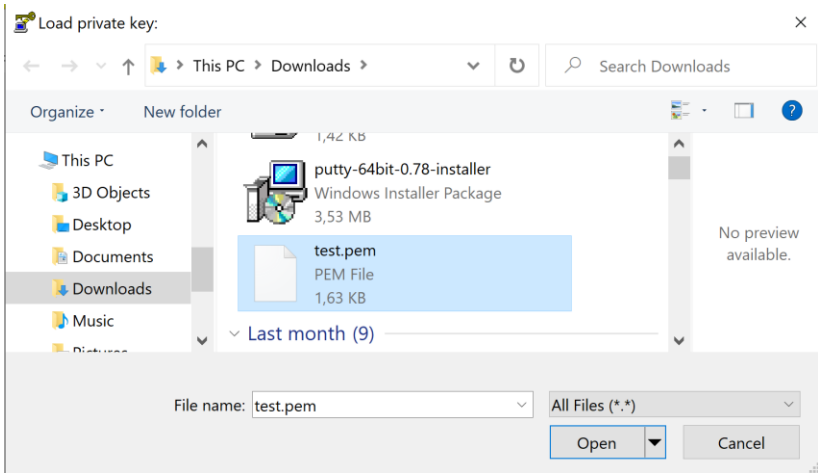


Для початку перетворіть .pem файл на .ppk (ключ в форматі PuTTY). Відкрийте програму PuTTYgen та натисніть кнопку Load.



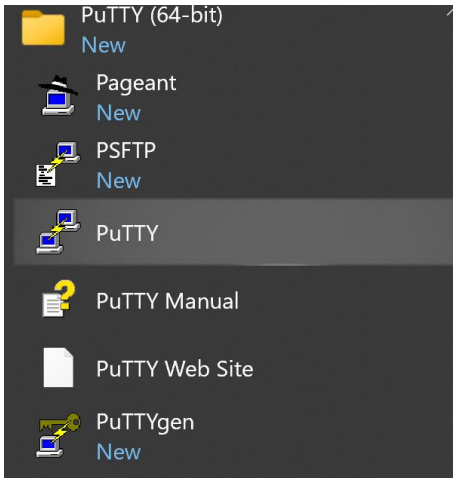
У вікні, що відкрилось внизу справа оберіть All Files (усі файли). Оберіть .pem файл, який був завантажений на ваш

компютер при створенні пари ключів при налаштуванні інстансу. Ви отримаєте повідомлення про успішне імпортування, натисніть OK. Натисніть Save private key (зберегти приватний ключ).

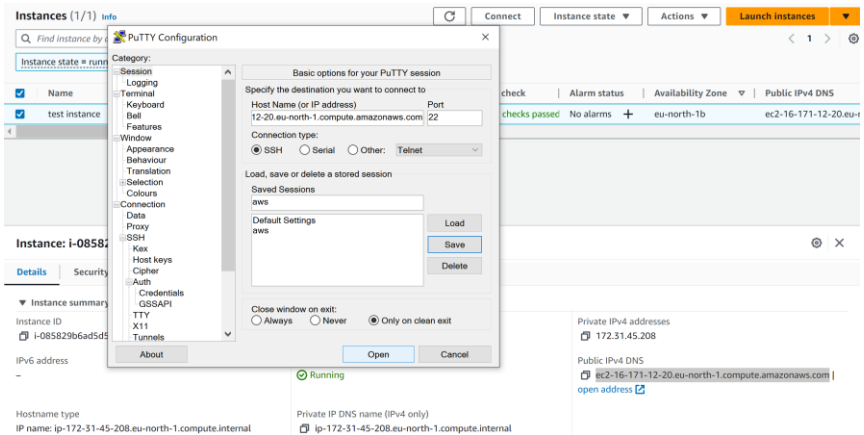


Ви отримали запитання, натисніть «Так». Дайте будь-яку назву файлу та збережіть у зручне вам місце.

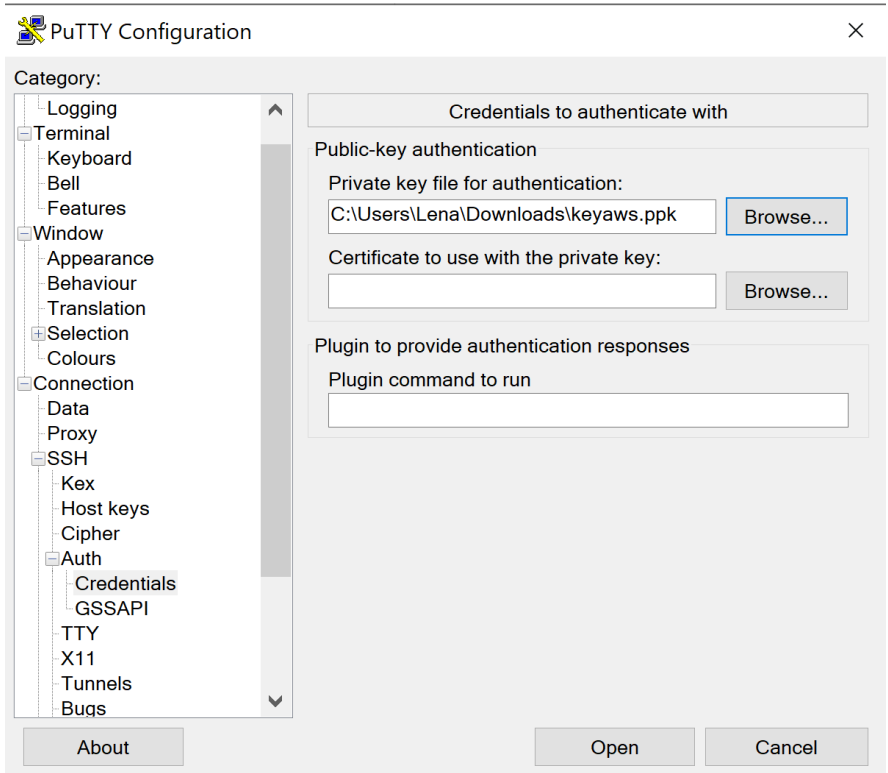
Для підключення до серверу відкрийте PuTTY.

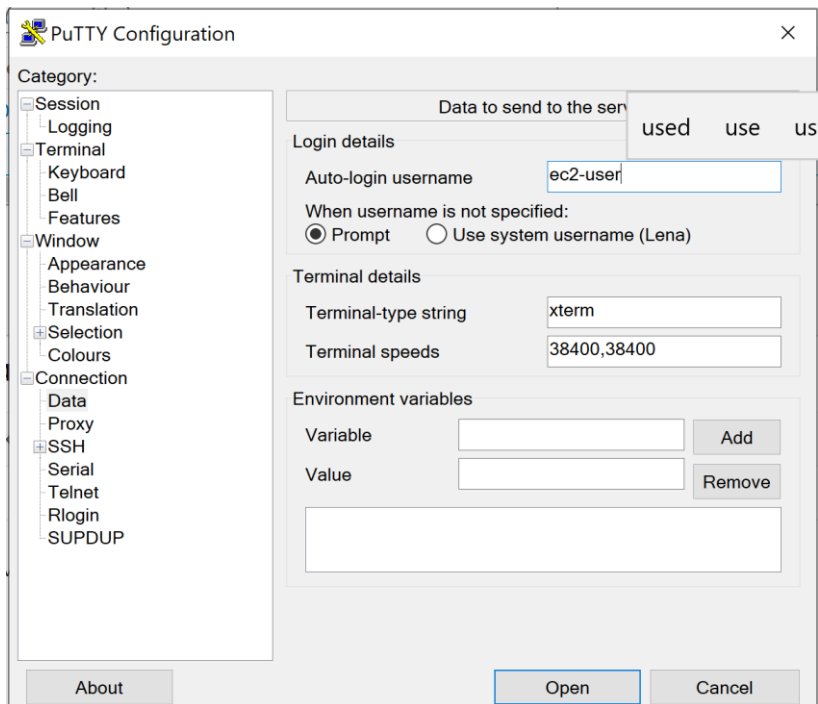


В рядку Hostname вкажіть Public IP адресу в форматі `ec2-user@54.234.225.175`, або Public DNS в форматі [ec2-user@ec2-54-161-94-141.compute-1.amazonaws.com](https://ec2-user@ec2-54-161-94-141.compute-1.amazonaws.com)



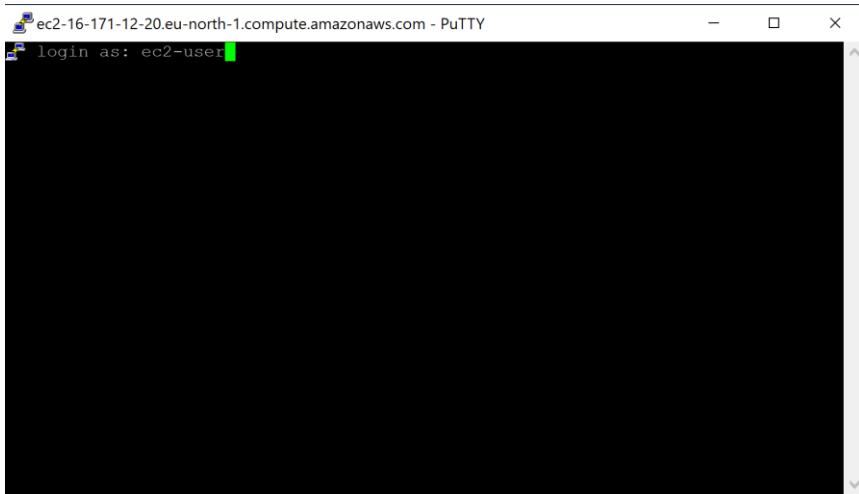
Щоб завантажити ключ у бічному меню PuTTY, оберіть пункт SSH >> Auth >> Credentials та у рядку Private file for authentication натисніть Browse. Оберіть раніше збережений ключ .ppk.



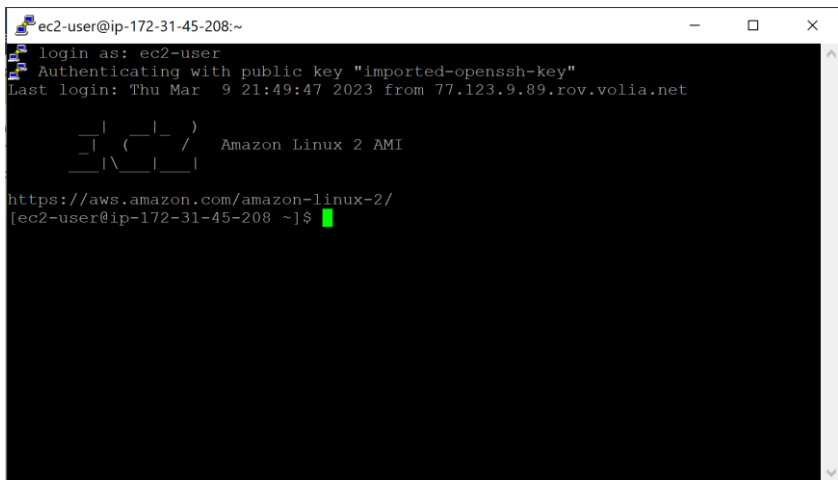


Задля зручності рекомендую зберегти сесію. Це дозволить не вказувати дані користувача. Для цього в бічному меню оберіть пункт Session, в рядку Saved Session вкажіть довільну назву та натисніть кнопку Save.

Якщо отримувате помилки при вході, перевірте, чи обраний файл ключа (ssh >> auth), чи правильно вказане ім'я користувача (за замовчуванням ec2-user), чи правильно вказана публічна IP-адреса.



```
ec2-16-171-12-20.eu-north-1.compute.amazonaws.com - PuTTY
login as: ec2-user
```



```
ec2-user@ip-172-31-45-208:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Mar  9 21:49:47 2023 from 77.123.9.89.rov.volia.net

  _ | ( _ | _ )
  _ | ( _ | _ /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-45-208 ~]$
```

Створимо текстовий файл на сервері за допомогою редактора vim. Для цього необхідно ввести команду

**vim some\_file\_name.txt**

Якщо файлу з такою назвою немає в поточній директорії, він буде створений.

```
ec2-user@ip-172-31-45-208:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Thu Mar  9 21:07:57 2023 from ec2-13-48-4-202.eu-north-1.compute.amazonaws.com  
  
  _ |  ( _ | _ )  
 _ |  \  _ | _ |  
                                     Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
ec2-user@ip-172-31-45-208 ~]$ vim newfile.txt  
ec2-user@ip-172-31-45-208 ~]$
```

У файл записуємо будь-яку інформацію, для цього потрібно натиснути Insert і ввести текст, після завершення введення натиснути Esc і ввести :wq, після чого натиснути Enter.

```
ec2-user@ip-172-31-45-208:~  
Hello!  
  
-- INSERT --  
1,7 All
```

В кореневій директорії на сервері з'явиться створений текстовий файл.

```
ec2-user@ip-172-31-45-208:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Thu Mar 9 21:07:57 2023 from ec2-13-48-4-202.eu-north-1.compute.amaz  
onaws.com  
  
  _ | _ | _ )  
  _ | ( _ | /  
  _ | \ _ | _ |  
                                     Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-45-208 ~]$ vim newfile.txt  
[ec2-user@ip-172-31-45-208 ~]$ vim newfile.txt  
[ec2-user@ip-172-31-45-208 ~]$ ls -la  
total 24  
drwx----- 3 ec2-user ec2-user 130 Mar  9 21:28 .  
drwxr-xr-x  3 root     root     22 Mar  9 20:28 ..  
-rw----- 1 ec2-user ec2-user  7 Mar  9 21:06 .bash_history  
-rw-r--r-- 1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout  
-rw-r--r-- 1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile  
-rw-r--r-- 1 ec2-user ec2-user 231 Jul 15 2020 .bashrc  
-rw-rw-r-- 1 ec2-user ec2-user  7 Mar  9 21:28 newfile.txt  
drwx----- 2 ec2-user ec2-user 29 Mar  9 20:28 .ssh  
-rw----- 1 ec2-user ec2-user 887 Mar  9 21:28 .viminfo  
[ec2-user@ip-172-31-45-208 ~]$
```

Для того щоб скопіювати файл з робочого комп'ютера необхідно скористатись командним рядком.

Команда для копіювання наступна

```
pscp -i yourkey.ppk yourfilename ec2-  
user@public_DNS:/home/ec2-user/
```

Тут `yourkey.ppk` це файл зі згенерованим приватним ключем. Якщо даний файл знаходиться не в поточній директорії, то потрібно вказувати повний шлях до файла. `yourfilename` - це шлях до файла, який копіюється з локального комп'ютера

Наприклад, якщо поточна директорія в якій виконується команда `C:\Users\Lena\Downloads` і файл з приватним ключем знаходиться в даній директорії, а файл копіюється `C:\work\python_labs\lab1_1.py`, то команда матиме вигляд

```
Command Prompt  
C:\Users\Lena\Downloads>pscp -i keyaws.ppk C:\work\python_labs\lab1_1.py ec2-user@ec2-16-171-12-20.eu-north-1.compute.amazonaws.com:/home/ec2-user  
lab1_1.py | 1 kB | 1.4 kB/s | ETA: 00:00:00 | 100%  
C:\Users\Lena\Downloads>
```

В даному випадку на сервер був скопійований python файл.



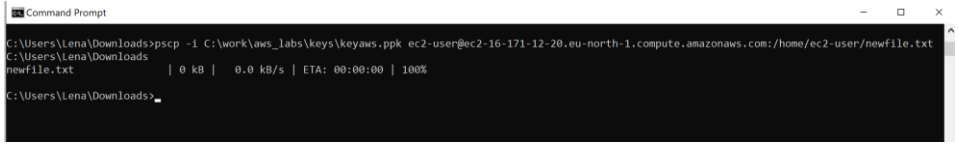
```
ec2-user@ip-172-31-45-208:~  
drwx----- 3 ec2-user ec2-user 153 Mar  9 22:10 .  
drwxr-xr-x  3 root      root      22 Mar  9 20:28 ..  
-rw----- 1 ec2-user ec2-user  32 Mar  9 21:42 .bash_history  
-rw-r--r--  1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout  
-rw-r--r--  1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile  
-rw-r--r--  1 ec2-user ec2-user 231 Jul 15 2020 .bashrc  
-rw-rw-r--  1 ec2-user ec2-user  7 Mar  9 21:28 newfile.txt  
-rw----- 1 ec2-user ec2-user  0 Mar  9 22:10 .python_history  
drwx----- 2 ec2-user ec2-user 29 Mar  9 20:28 .ssh  
-rw----- 1 ec2-user ec2-user 887 Mar  9 21:28 .viminfo  
[ec2-user@ip-172-31-45-208 ~]$ ls -la  
total 28  
drwx----- 3 ec2-user ec2-user 170 Mar  9 22:21 .  
drwxr-xr-x  3 root      root      22 Mar  9 20:28 ..  
-rw----- 1 ec2-user ec2-user  32 Mar  9 21:42 .bash_history  
-rw-r--r--  1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout  
-rw-r--r--  1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile  
-rw-r--r--  1 ec2-user ec2-user 231 Jul 15 2020 .bashrc  
-rw-rw-r--  1 ec2-user ec2-user 1434 Mar  9 22:21 lab1_1.py  
-rw-rw-r--  1 ec2-user ec2-user  7 Mar  9 21:28 newfile.txt  
-rw----- 1 ec2-user ec2-user  0 Mar  9 22:10 .python_history  
drwx----- 2 ec2-user ec2-user 29 Mar  9 20:28 .ssh  
-rw----- 1 ec2-user ec2-user 887 Mar  9 21:28 .viminfo  
[ec2-user@ip-172-31-45-208 ~]$
```

Оскільки на сервері встановлено інтерпретатор python, даний файл можна запустити на виконання

```
ec2-user@ip-172-31-45-208:~  
-rw-r--r--  1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile  
-rw-r--r--  1 ec2-user ec2-user 231 Jul 15 2020 .bashrc  
-rw-rw-r--  1 ec2-user ec2-user  7 Mar  9 21:28 newfile.txt  
-rw----- 1 ec2-user ec2-user  0 Mar  9 22:10 .python_history  
drwx----- 2 ec2-user ec2-user 29 Mar  9 20:28 .ssh  
-rw----- 1 ec2-user ec2-user 887 Mar  9 21:28 .viminfo  
[ec2-user@ip-172-31-45-208 ~]$ ls -la  
total 28  
drwx----- 3 ec2-user ec2-user 170 Mar  9 22:21 .  
drwxr-xr-x  3 root      root      22 Mar  9 20:28 ..  
-rw----- 1 ec2-user ec2-user  32 Mar  9 21:42 .bash_history  
-rw-r--r--  1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout  
-rw-r--r--  1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile  
-rw-r--r--  1 ec2-user ec2-user 231 Jul 15 2020 .bashrc  
-rw-rw-r--  1 ec2-user ec2-user 1434 Mar  9 22:21 lab1_1.py  
-rw-rw-r--  1 ec2-user ec2-user  7 Mar  9 21:28 newfile.txt  
-rw----- 1 ec2-user ec2-user  0 Mar  9 22:10 .python_history  
drwx----- 2 ec2-user ec2-user 29 Mar  9 20:28 .ssh  
-rw----- 1 ec2-user ec2-user 887 Mar  9 21:28 .viminfo  
[ec2-user@ip-172-31-45-208 ~]$ python3 lab1_1.py  
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.  
[ 'Lorem', 'Ipsum', 'is', 'simply', 'dummy', 'text', 'of', 'the', 'printing', 'and  
'typesetting', 'industry', 'Lorem', 'Ipsum', 'has', 'been', 'the', 'industry'  
's', 'standard', 'dummy', 'text', 'ever', 'since', 'the', '1500s']  
[ 'Lorem', 'Ipsum', 'is', 'simply', 'dummy', 'text', 'of', 'the', 'printing', 'and  
'typesetting', 'industry', 'Lorem', 'Ipsum', 'has', 'been', 'the', 'industry'  
's', 'standard', 'dummy', 'text', 'ever', 'since', 'the', '1500s']  
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.  
[ec2-user@ip-172-31-45-208 ~]$
```

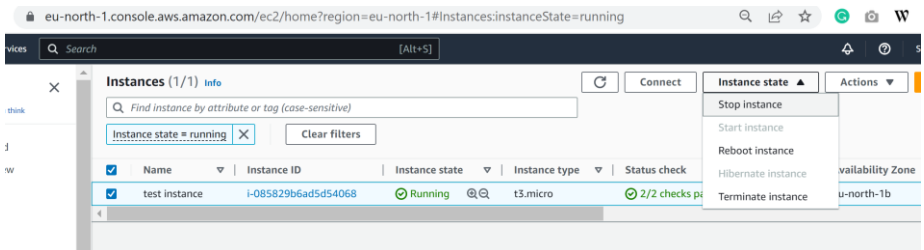
Щоб скопіювати файл з інстансу на локальний комп'ютер використовуємо команду в командному рядку

**pscp -i yourkey.ppk ec2-user@publicDNS:/home/ec2-user/yourfilename C:[local\_destination\_path]\[folder]**



```
Command Prompt
C:\Users\Lena\Downloads>pscp -i C:\work\aws_labs\keys\keyaws.ppk ec2-user@ec2-16-171-12-20.eu-north-1.compute.amazonaws.com:/home/ec2-user/newfile.txt
C:\Users\Lena\Downloads
newfile.txt | 0 kB | 0.0 kB/s | ETA: 00:00:00 | 100%
C:\Users\Lena\Downloads>
```

По завершенні роботи потрібно вимкнути інстанс (Stop instance) – квота безкоштовного використання не є безмежною. Це можна зробити з меню Instance State для списку інстансів.



Протягом перших 12 місяців Amazon надає безкоштовно 750 годин для екземпляру t2.micro та ОС (AMI) з позначкою Free tier eligible. Ці години можуть бути використані одним екземпляром, що працює протягом повного місяця (31 день \* 24 години = 744 години), або кількома екземплярами Amazon EC2, що використовуються протягом місяця. Будь-які години, що перевищують безкоштовний рівень, стягуватимуться за цінами на вимогу. Тому після виконання всіх лабораторних робіт варто видалити екземпляр. Для цього натисніть на галочку поруч з ним та у верхньому куті оберіть меню Instance state, оберіть «Terminate Instance». Ви отримаєте повідомлення про зупинення екземпляру.

### 1.3. Завдання

1. Зареєструватись в AWS
2. Створити мікро-інстанс
3. Отримати доступ до нього
4. Навчитись моніторити використання ресурсів
5. Навчитися завантажувати файли на інстанс (створити пустий файл \*.txt та завантажити його на інстанс через термінал)
6. Відкрити файл на інстансі за допомогою редактора Vim (додати текст у файл, першим рядком обов'язково повинно бути прізвище та ім'я, номер групи)
7. Завантажити із інстанса змінений текстовий файл
8. Завантажити на інстанс будь-який робочий python-скрипт та запустити на виконання.
8. Результати усіх кроків оформити у вигляді детального звіту зі скріншотами
9. Навести перелік проблем, вирішення яких було складним в ході виконання роботи в розділі висновків.

### 1.4. Додаткові джерела інформації

1. <https://aws.amazon.com/>
2. [https://www.youtube.com/watch?v=YB\\_qanudIzA](https://www.youtube.com/watch?v=YB_qanudIzA)
3. <https://www.youtube.com/watch?v=8bIW7qlldLg>
4. <https://www.youtube.com/watch?v=bi7ow5NGC-U>
5. <https://websetnet.net/uk/how-to-use-vim-editor/>
6. <https://asf.alaska.edu/how-to/data-recipes/moving-files-into-and-out-of-an-aws-ec2-instance-windows/>

# Лабораторна робота №2

## Використання AWS Simple Storage Service (S3)

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

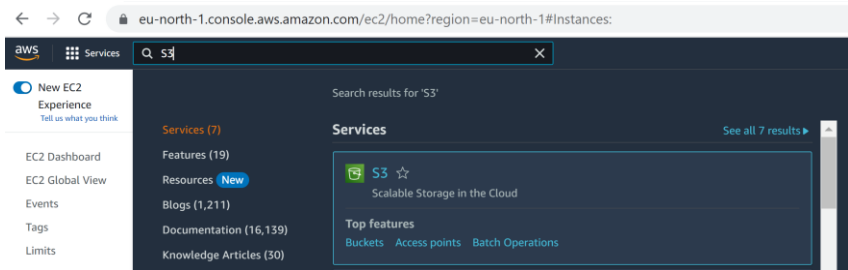
### 2.1. План виконання

- Створити бакет S3
- Ознайомитись зі способами взаємодії з бакетом S3
- Документування зробленої роботи у вигляді деталізованого звіту з коментарями

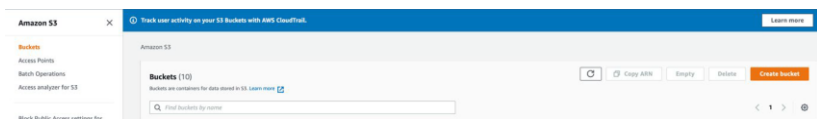
### 2.2. Порядок виконання роботи

В межах Free Tier для використання доступно 5 Gb сховища S3 (20 000 запитів на читання та 2000 запитів на запис).

Створення бакета S3 можливе різними способами: через AWS Management Console, через AWS CLI тощо.



Так, через сервіс S3 бакет може бути створений через процедуру Create Bucket.



*Створення бакету з веб-інтерфейсу*

Потрібно вказати ім'я бакету та регіон (той самий, в якому було раніше створено інстанс). При створенні варто заборонити публічний доступ до створеного бакета.

Amazon S3 > Create bucket

## Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

### General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - *optional*  
Only the bucket settings in the following configuration are copied.

### Block Public Access settings for bucket

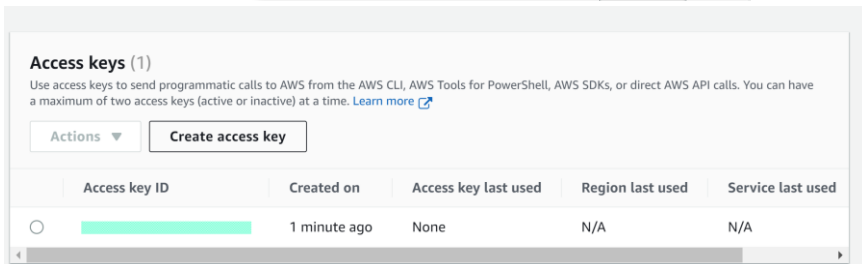
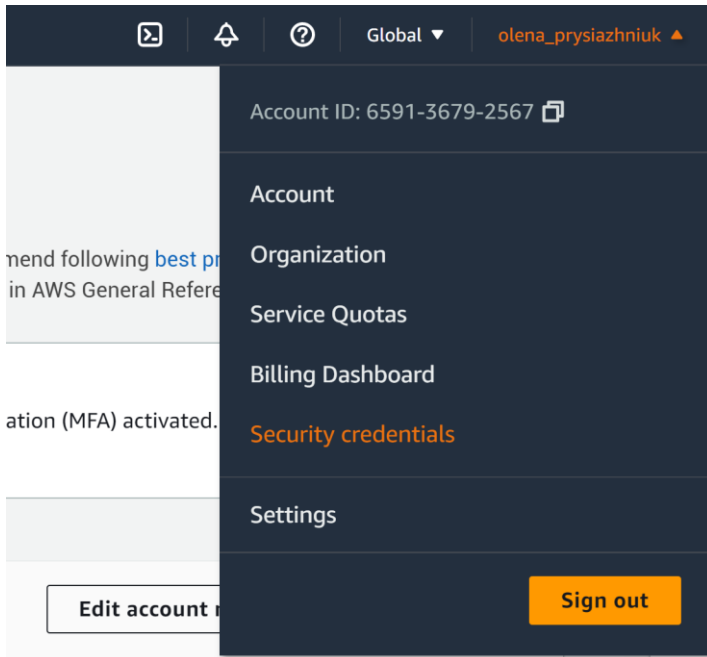
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

### *Параметри бакету*

Для програматичної роботи з ресурсами AWS (через API) вам знадобиться раніше створений ключ користувача (Access key ID, Secret access key).

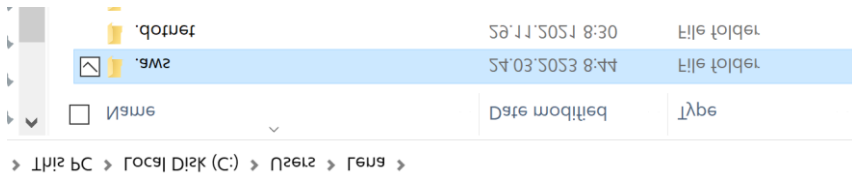
Їх можна отримати з профіля користувача (пункт Security Credentials), пункт Access keys (access key ID and secret access key), кнопка Create new Access key (запишіть ключі у файл).



### *Створення ключів для програматичного доступу*

Бажано створювати окремого користувача (не root). Це можна створити шляхом використання сервісу керування користувачами IAM (розділ Users, кнопка Add user). Важливо при цьому не забути надати користувачу програматичний доступ.





Детальніше про використання:

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-using.html>

Наприклад перевірити список бакетів можна командою

```
C:\Users\Lena>aws s3 ls
2023-03-24 10:41:57 testbucket1209fd
```

### Для опрацювання самостійно

- Ознайомитись з основами використання AWS CLI для роботи з S3 (робота з файлами, папками, копіювання даних з інстансу на бакет та навпаки) як через AWS CLI (в консольному режимі), так і у веб-інтерфейсі AWS Management Console. Для довідки - <https://docs.aws.amazon.com/cli/latest/reference/s3/>

### 2.4. Додаткові джерела інформації

1. Andreas Wittig, Michael Wittig “Amazon Web Services in Action”
2. <https://www.youtube.com/watch?v=i4YFFWcyeFM>



## Лабораторна робота №3

### Програматичний доступ до AWS Simple Storage Service (S3)

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

#### 3.1. План виконання

- Вивчити елементи програматичного доступу до даних на S3 з Python
- Налаштувати доступ до бакета S3 з віртуального мікро-сервера, створеного у попередній лабораторній роботі
- Документування зробленої роботи у вигляді деталізованого звіту з коментарями

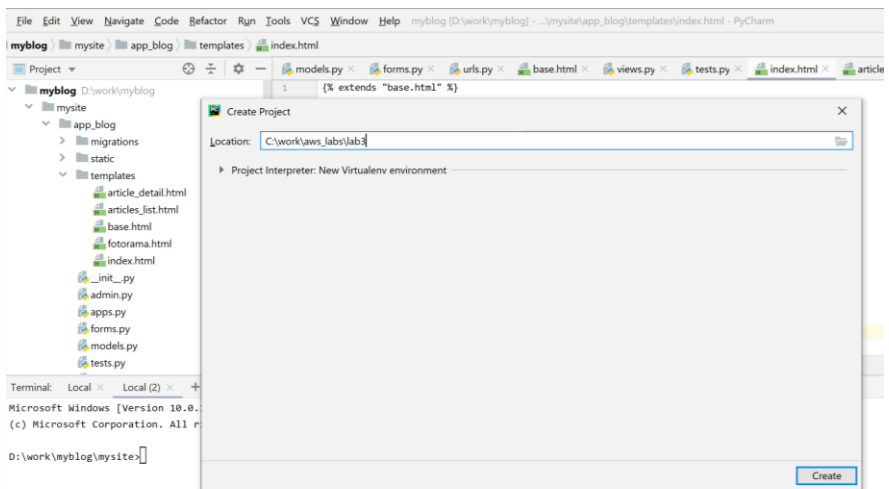
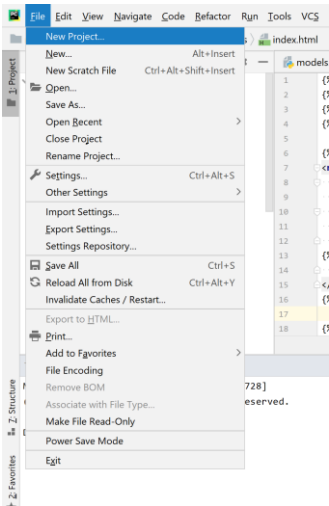
#### 3.2. Порядок виконання роботи

AWS SDK для Python (Boto3) використовується для створення, налаштування та керування службами AWS, такими як Amazon Elastic Compute Cloud (Amazon EC2) і Amazon Simple Storage Service (Amazon S3). SDK надає об'єктно-орієнтований API, а також низькорівневий доступ до сервісів AWS.

Детальніше про роботу з boto3:

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-examples.html>

Для виконання лабораторної роботи зручно використовувати наприклад PyCharm. Якщо встановлений раніше, наприклад при виконанні лабораторних робіт з програмування. Створимо новий проект та python файл в ньому, а також скопіюйте в дану директорію будь який файл який пізніше буде перенесений в бакет S3:



В терміналі внизу вікна PyCharm відкрійте нову вкладку. Рядок повинен починатись із (venv) тобто ви працюватимете у активованому віртуальному оточенні.

Якщо працюєте з іншим IDLE, потрібно самостійно створити віртуальне оточення, активувати його і запускати на виконання файл лабораторної роботи з–під активованого віртуального оточення.

Для налаштування Python API потрібно встановити пакетний менеджер `pip` на ваш комп'ютер та власне сам Python SDK для AWS *pip install boto3*.

Для встановлення `pip` в командному рядку виконайте команди

```
curl -O https://bootstrap.pypa.io/get-pip.py
```

```
python3 get-pip.py --user
```

Віртуальне середовище - це можливість мати кілька Python проектів і працювати над ними паралельно. При цьому один іншому не буде перешкоджати. Це окремі папки із своєю копією Python та усіх інсталюваних додаткових бібліотек.

Virtualenv – це інструмент, який дозволяє налаштувати безліч “копій” вашого заінсталюваного Python, і працювати над кількома Python проектами одночасно не заважаючи один одному. Цей підхід дозволяє уникати конфліктів між заінсталюваними пакетами та їхніми версіями. Іншими словами можете уявити, що кожне віртуальне середовище це окрема папочка зі своїм набором налаштувань, які не заважають іншому такому ж віртуальному середовищу. Це як поставити на Windows програму лише для одного користувача, а інші користувачі мають свої власні набори програм.

Маючи `Pip` можемо з легкістю однією командою встановити `virtualenv` пакет. В PowerShell або `cmd` в контексті будь-якої папки запускаєте наступну команду (з правами адміністратора):

```
pip install virtualenv
```

Також тепер ви можете ввести команду `virtualenv` і вона виведе вам інструкцію по використанню.

У віртуальному середовищі маємо наступні важливі для нас папки:

- `bin`: бінарники (`python`, `pip`, `activate`);
- `lib`: пітонівські пакети, сюди будуть також інсталювані

пакети (підпапка `python3.X/site-packages` – в даному випадку версія Python, але може відрізнятись), сюди ми не раз будемо заглядати під час розробки.

Перейшовши в обрану директорію, створюємо там нове віртуальне середовище для проекту:

```
python -m venv venv
```

Для подальшої роботи всередині віртуального оточення його потрібно активувати командою

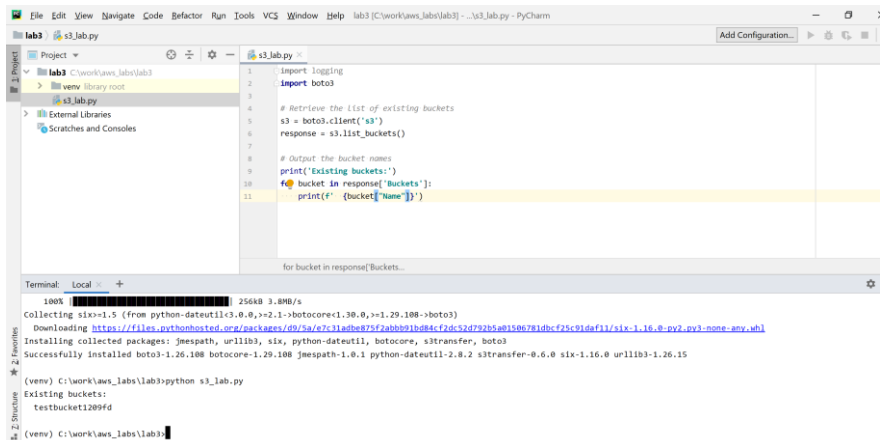
```
venv\Scripts\activate
```

І після цього інсталийте `boto3`

```
pip install boto3
```

Далі можна переходити до написання програми що взаємодіє з Amazon S3.

Наприклад для отримання інформації про всі бакети скрипт буде мати вигляд:



Для копіювання файлу що попередньо був доданий в директорію проекту виконайте скрипт:

AWS SDK для Python надає пару методів для завантаження файлу в сегмент S3.

Метод `upload_file` приймає ім'я файлу, ім'я бакета та ім'я об'єкта. Цей метод обробляє великі файли, розділяючи їх на менші частини та завантажуючи кожен частину паралельно.

```
import logging
import boto3
from botocore.exceptions import ClientError
import os

def upload_file(file_name, bucket, object_name=None):
    """Upload a file to an S3 bucket

    :param file_name: File to upload
    :param bucket: Bucket to upload to
    :param object_name: S3 object name. If not specified then
file_name is used
    :return: True if file was uploaded, else False
    """

    # If S3 object_name was not specified, use file_name
if object_name is None:
        object_name = os.path.basename(file_name)

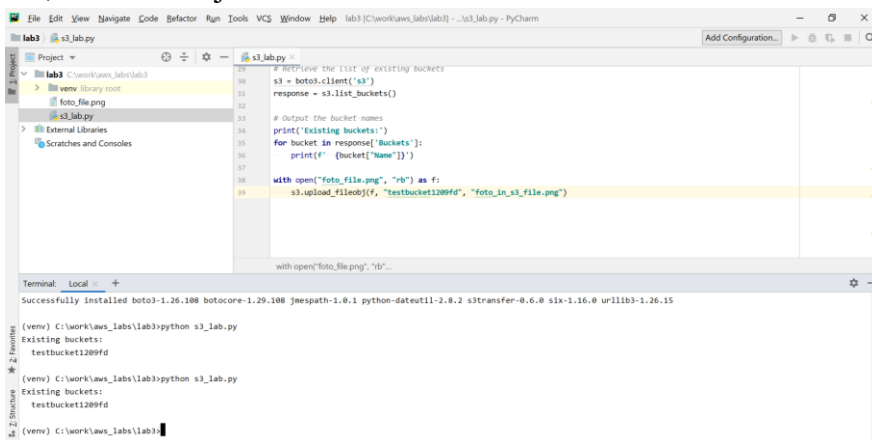
    # Upload the file
    s3_client = boto3.client('s3')
    try:
        response = s3_client.upload_file(file_name, bucket,
object_name)
    except ClientError as e:
        logging.error(e)
        return False
```

## return True

Метод `upload_fileobj` приймає читабельний файлоподібний об'єкт. Файловий об'єкт має бути відкритий у двійковому, а не текстовому режимі.

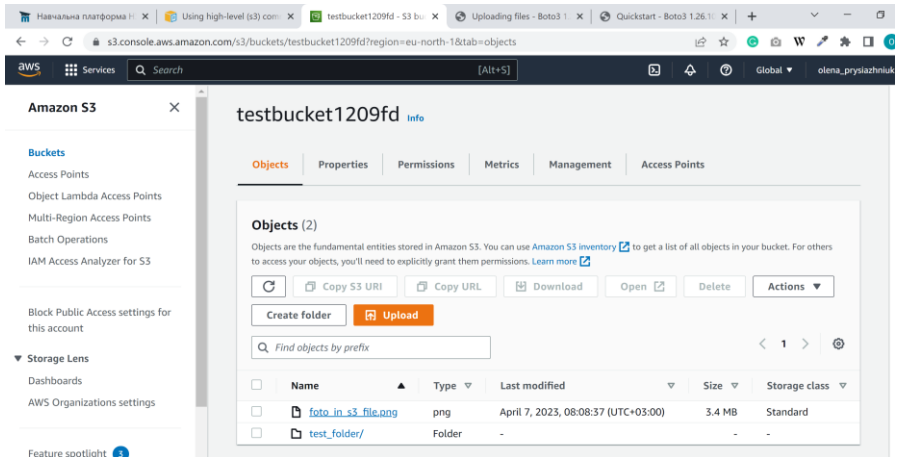
```
s3 = boto3.client('s3')  
with open("FILE_NAME", "rb") as f:  
    s3.upload_fileobj(f, "BUCKET_NAME",  
"OBJECT_NAME")
```

Методи `upload_file` і `upload_fileobj` надаються класами `S3 Client`, `Bucket` і `Object`.



```
lab3 [C:\work\aws_lab\lab3] - vs3_lab.py - PyCharm  
Project  
lab3 C:\work\aws_lab\lab3  
  view library root  
  foto_file.png  
  s3_lab.py  
  External Libraries  
  Scratches and Consoles  
s3_lab.py  
28 # Retrieve the list of existing buckets  
29 s3 = boto3.client('s3')  
30 response = s3.list_buckets()  
31  
32 # Output the bucket names  
33 print('Existing buckets:')  
34 for bucket in response['Buckets']:  
35     print(f' {bucket["Name"]}')  
36  
37 with open("foto_file.png", "rb") as f:  
38     s3.upload_fileobj(f, "testbucket1209fd", "foto_in_s3_file.png")  
39  
with open("foto_file.png", "rb"...  
Terminal: Local ... +  
Successfully installed boto3-1.26.100 botoecore-1.29.100 jmespath-1.0.1 python-dateutil-2.8.2 s3transfer-0.6.0 six-1.16.0 urllib3-1.26.15  
(venv) C:\work\aws_lab\lab3>python s3_lab.py  
Existing buckets:  
testbucket1209fd  
(venv) C:\work\aws_lab\lab3>python s3_lab.py  
Existing buckets:  
testbucket1209fd  
(venv) C:\work\aws_lab\lab3>
```

Перейдіть на сайт Amazon AWS у створений бакет S3 і переконайтесь що файл був завантажений:



Для завантаження з бакету на локальний комп'ютер:

Методи, надані AWS SDK для Python для завантаження файлів, подібні до тих, що надаються для завантаження файлів.

Метод `download_file` приймає імена сегмента та об'єкта для завантаження, а також ім'я файлу для збереження файлу.

```
import boto3
```

```
s3 = boto3.client('s3')
```

```
s3.download_file('BUCKET_NAME', 'OBJECT_NAME',  
'FILE_NAME')
```

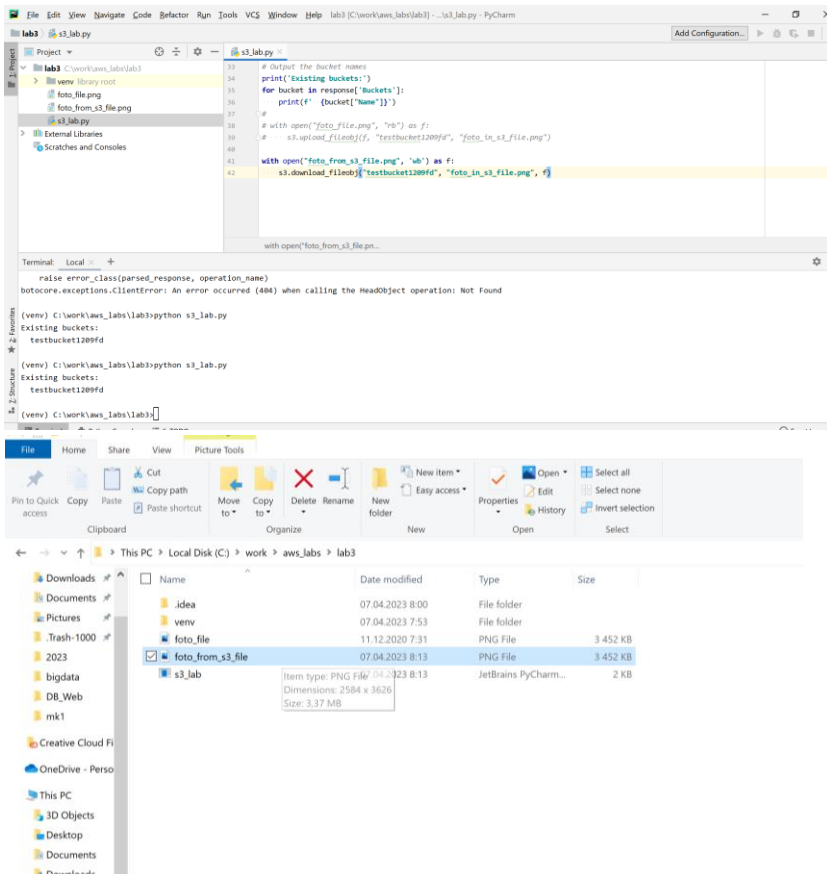
Метод `download_fileobj` приймає записуваний файлоподібний об'єкт. Файловий об'єкт має бути відкритий у двійковому, а не текстовому режимі.

```
s3 = boto3.client('s3')
```

```
with open('FILE_NAME', 'wb') as f:
```

## s3.download\_fileobj('BUCKET\_NAME', 'OBJECT\_NAME', f)

Подібно до своїх двоюрідних братів завантаження, методи завантаження надаються класами S3 Client, Bucket і Object, і кожен клас забезпечує однакову функціональність.



### 3.3. Завдання

1. Виконати кроки з порядку виконання лабораторної роботи.
2. Написати скрипт, що створить текстовий файл з рандомно згенерованих даних.



3. Створений файл має програматично вивантажуватись на S3.
4. Завантажити створений скрипт на інстанс EC2 та запустити на інстансі.
5. Результати усіх кроків оформити у вигляді детального протоколу зі скріншотами та командами в консолі які використовувалися
6. Навести перелік проблем, вирішення яких було складним в ході виконання роботи в розділі висновків до протоколу

### **3.4. Додаткові джерела інформації**

1. Andreas Wittig, Michael Wittig “Amazon Web Services in Action”
2. <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>

# Лабораторна робота №4

## Використання AWS DynamoDB

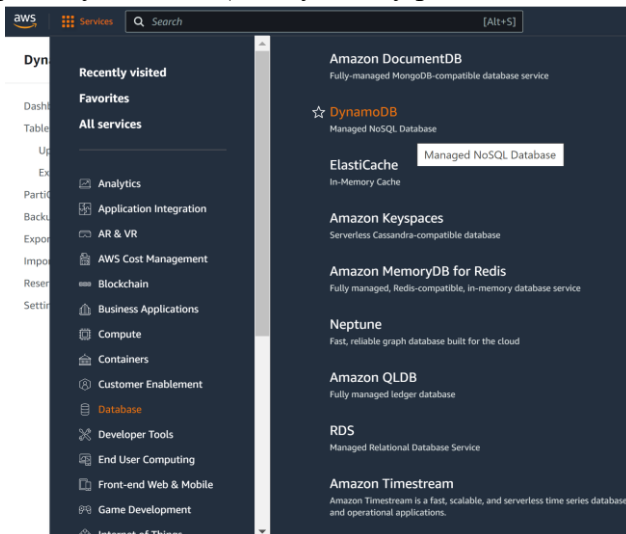
**Мета роботи:** ознайомитись з використанням AWS DynamoDB (serverless database)

### 4.1. План виконання

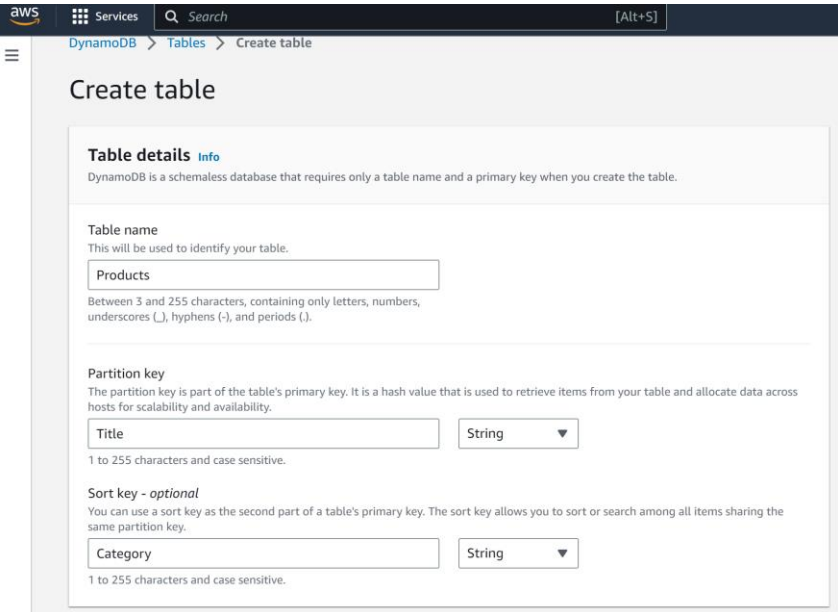
- Спроекувати структуру даних (таблицю).
- Вивчити способи роботи з даними засобами DynamoDB.
- Документування зробленої роботи у вигляді деталізованого звіту з коментарями

### 4.2. Порядок виконання роботи

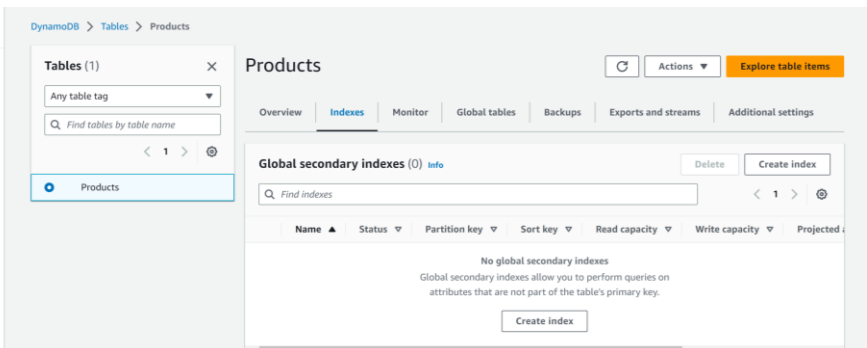
Створення таблиці DynamoDb здійснюється традиційно як через AWS CLI, так і засобами AWS Management Console. Для створення таблиці засобами AWS Management Console оберіть сервіс DynamoDB (доступний у розділі Databases)



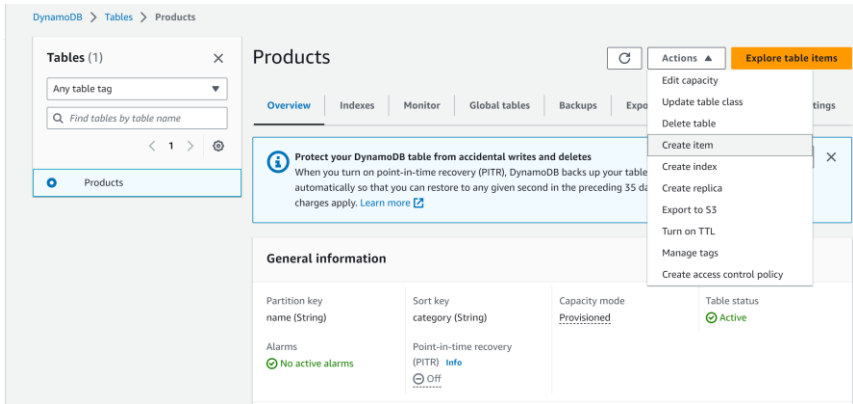
та натисніть кнопку «Create Table». Створимо таблицю Products, додавши до неї в якості ключа текстове поле Name та поле Category, за яким можна буде здійснювати пошук.



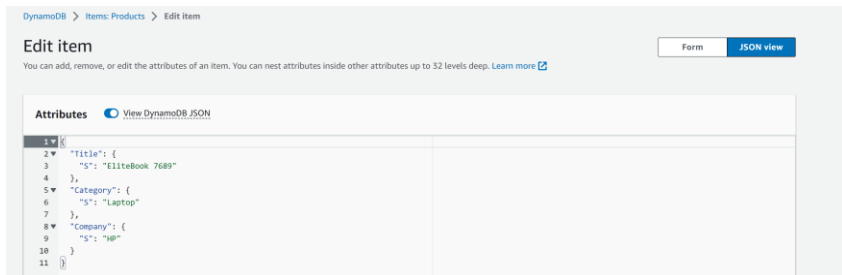
Створення таблиці DynamoDB Створимо кілька записів для цієї таблиці, кожен запис (Item) містить кілька атрибутів (це можна зробити в розділі Items).



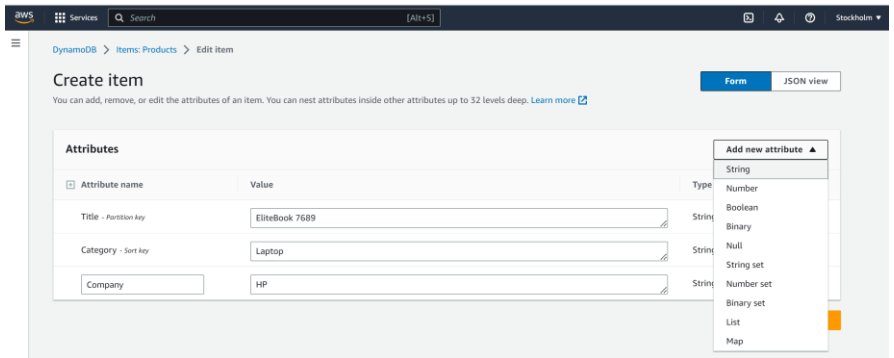
## Внесемо основні частини запису через Create Item



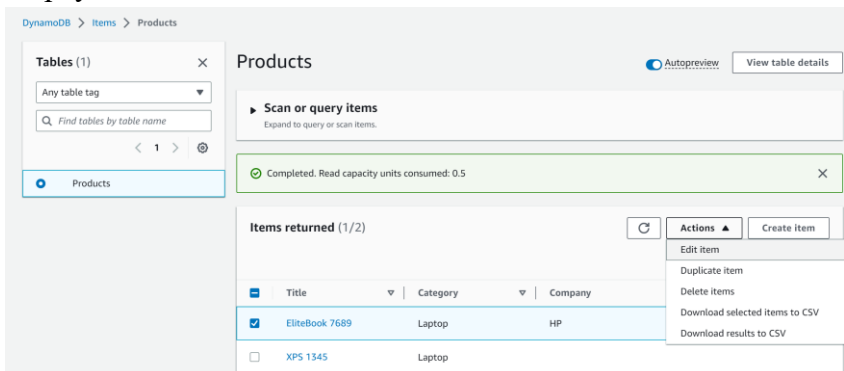
Кількість атрибутів може бути розширена досить гнучко – вона може бути іншою в сенсі додаткових полів – фактично є повна аналогія з представленням даних в JSON.



Доступні декілька режимів редагування запису в базі даних та створення додаткових атрибутів, а саме Form view та JSON view



Об'єкти-записи можна змінювати, дублювати, видаляти, експортувати тощо.



Працюючи з фільтрами можна здійснювати пошук даних.

## Доступ з використанням AWS CLI

Значну частину операцій з dynamodb можна зробити і із використанням AWS CLI

(<https://docs.aws.amazon.com/cli/latest/reference/dynamodb/index.html#cliaws-dynamodb>) та

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithItems.html>)

```
aws dynamodb create-table --table-name Products --
attribute-definitions AttributeName=Title,AttributeType=S
```

```
AttributeName=Category,AttributeType=S --key-schema
AttributeName=Title,KeyType=HASH
AttributeName=Category,KeyType=RANGE --provisionedthroughput
ReadCapacityUnits=10,WriteCapacityUnits=5
```

DynamoDB надає чотири операції для основних функцій створення, читання, оновлення та видалення (CRUD). Усі ці операції є атомарними.

PutItem — створити елемент.

GetItem — читання елемента.

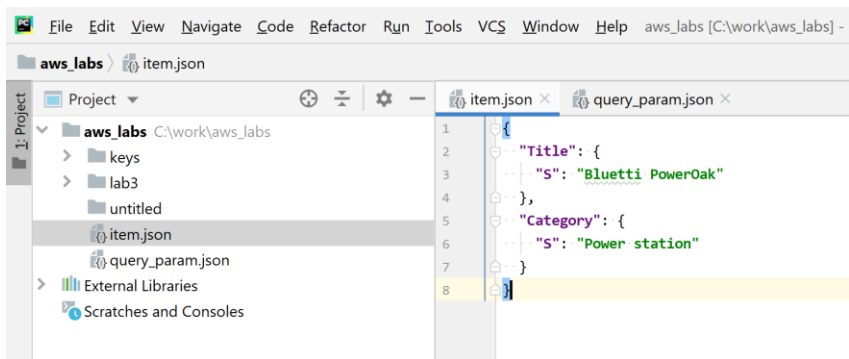
UpdateItem — оновити елемент.

DeleteItem — видалити елемент.

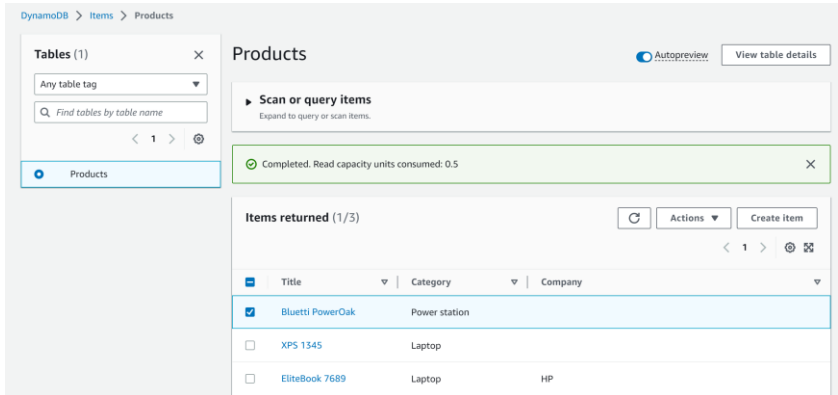
Для кожної з цих операцій потрібно вказати первинний ключ елемента, з яким ви хочете працювати. Наприклад, щоб прочитати елемент за допомогою GetItem, ви повинні вказати ключ розділу та ключ сортування (якщо є) для цього елемента.

Так для додавання запису можна використати метод put-item, передавши йому відповідний JSON

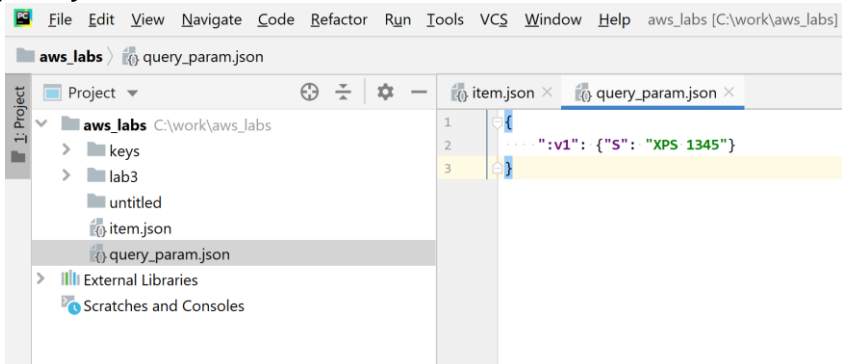
```
aws dynamodb put-item --table-name Products --item
file://C:\work\aws_labs\item.json --return-consumed-capacity
TOTAL
```



```
C:\Users\Lena>aws dynamodb put-item --table-name Products --item file://C:\work\aws_labs\item.json --return-consumed-capacity TOTAL
{
  "ConsumedCapacity": {
    "TableName": "Products",
    "CapacityUnits": 1.0
  }
}
```



aws dynamodb query --table-name Products --key-condition-expression "Title = :v1" --expression-attribute-values file://C:\work\aws\_labs\query\_param.json --return-consumed-capacity TOTAL



```
Command Prompt
"ConsumedCapacity": {
  "TableName": "Products",
  "CapacityUnits": 1.0
}
}
C:\Users\Lena>aws dynamodb query --table-name Products --key-condition-expression "Title = :v1" --expression-attribute-values file://C:\work\aws_labs\query_param.json --return-consumed-capacity TOTAL
{
  "Items": [
    {
      "Title": {
        "S": "XPS 1345"
      },
      "Category": {
        "S": "Laptop"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 1,
  "ConsumedCapacity": {
    "TableName": "Products",
    "CapacityUnits": 0.5
  }
}
C:\Users\Lena>
```

## Доступ з використанням AWS SDK for Python (Boto3)

DynamoDB.ServiceResource і DynamoDB.Table дозволяє створювати таблиці, записувати елементи в таблиці, змінювати існуючі елементи, отримувати елементи та запитувати/фільтрувати елементи в таблиці. Документація доступна за посиланням <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/dynamodb.html>

Наприклад для отримання доступу до таблиці використовується вже знайомий метод `boto3.resource('dynamodb')` з вказанням в якості ресурсу `'dynamodb'`.



```

File Edit View Navigate Code Refactor Run Tools VCS Window Help lab4[C:\work\aws_lab4\lab4] - ...aws_db.py - PyCharm
lab4 aws_db.py
Project
lab4 C:\work\aws_lab4\lab4
View Library root
aws_db.py
products.py
External Libraries
Scratches and Consoles
aws_db.py
1 import boto3
2
3 # Get the service resource.
4 dynamodb = boto3.resource('dynamodb')
5
6
7 # Instantiate a table resource object without actually
8 # creating a DynamoDB table. Note that the attributes of this table
9 # are lazy-loaded: a request is not made nor are the attribute
10 # values populated until the attributes
11 # on the table resource are accessed or its load() method is called.
12 table = dynamodb.Table('Products')
13
14 # Print out some data about the table.
15 # This will cause a request to be made to DynamoDB and its attribute
16 # values will be set based on the response.
17 print(table.creation_date_time)
18
Terminal: Local
Successfully installed boto3-1.26.117 boto-core-1.29.111 jmespath-1.0.1 python-dateutil-2.8.2 s3transfer-0.6.0 six-1.16.0 urllib3-1.26.15
WARNING: You are using pip version 19.3.1, however version 21.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(venv) C:\work\aws_lab4\lab4>python aws_db.py
2023-04-21 08:17:33.809000+03:00
(venv) C:\work\aws_lab4\lab4>

```

## Приклад програматичного додання запису в таблицю

```

lab4 aws_db.py
Project
lab4 C:\work\aws_lab4\lab4
View Library root
aws_db.py
products.py
External Libraries
Scratches and Consoles
aws_db.py
8 # Instantiate a table resource object without actually
9 # creating a DynamoDB table. Note that the attributes of this table
10 # are lazy-loaded: a request is not made nor are the attribute
11 # values populated until the attributes
12 # on the table resource are accessed or its load() method is called.
13 table = dynamodb.Table('Products')
14
15 # Print out some data about the table.
16 # This will cause a request to be made to DynamoDB and its attribute
17 # values will be set based on the response.
18 print(table.creation_date_time)
19
Terminal: Local
boto-core.exceptions.ClientError: An error occurred (ValidationException) when calling the PutItem operation: One or more parameter values were invalid: Missing the key category in the item
(venv) C:\work\aws_lab4\lab4>python aws_db.py
2023-04-21 08:17:33.809000+03:00
(venv) C:\work\aws_lab4\lab4>

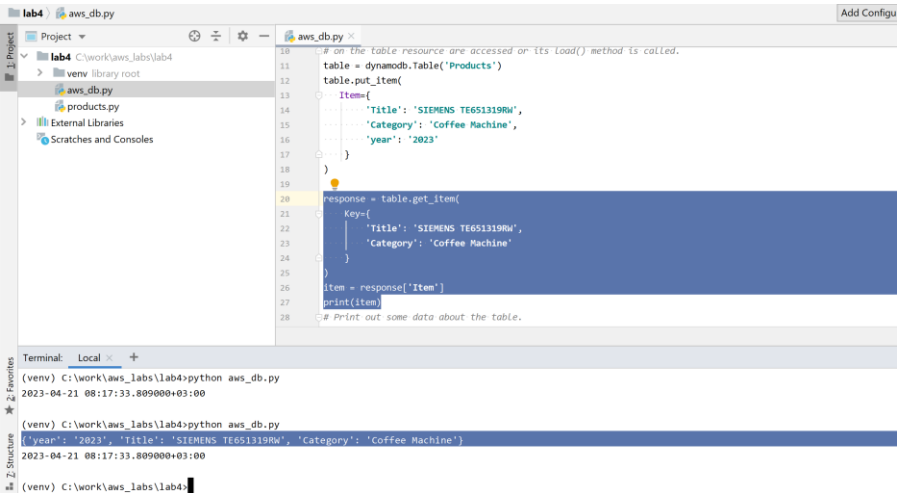
```

Items returned (1/4) Refresh Actions Create item

< 1 > Settings Close

	Title	Category	Company	year
<input checked="" type="checkbox"/>	SIEMENS TEG51319RW	Coffee Machine		2023
<input type="checkbox"/>	Bluetti PowerOak	Power station		
<input type="checkbox"/>	XPS 1345	Laptop		
<input type="checkbox"/>	EliteBook 7689	Laptop	HP	

Також можна отримати об'єкт за допомогою `DynamoDB.Table.get_item()`:



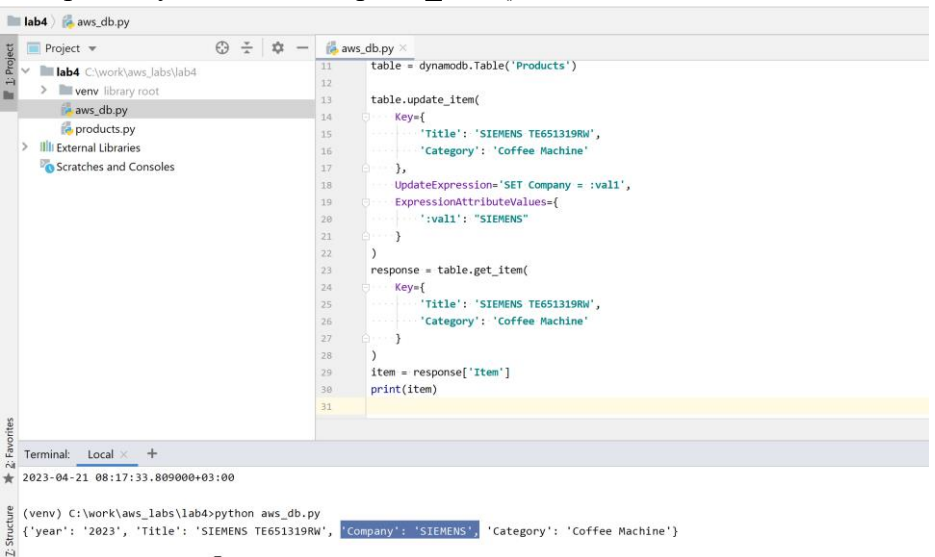
```
lab4 aws_db.py Add Configu
Project
lab4 C:\work\aws_labs\lab4
  venv library root
  aws_db.py
  products.py
  External Libraries
  Scratches and Consoles
aws_db.py
10 # on the table resource are accessed on its load() method is called.
11 table = dynamodb.Table('Products')
12 table.put_item(
13     Item={
14         'Title': 'SIEMENS TE651319RW',
15         'Category': 'Coffee Machine',
16         'year': '2023'
17     }
18 )
19
20 response = table.get_item(
21     Key={
22         'Title': 'SIEMENS TE651319RW',
23         'Category': 'Coffee Machine'
24     }
25 )
26 item = response['Item']
27 print(item)
28 # Print out some data about the table.

Terminal Local +
(venv) C:\work\aws_labs\lab4>python aws_db.py
2023-04-21 08:17:33.809000+03:00

(venv) C:\work\aws_labs\lab4>python aws_db.py
{'year': '2023', 'Title': 'SIEMENS TE651319RW', 'Category': 'Coffee Machine'}
2023-04-21 08:17:33.809000+03:00

(venv) C:\work\aws_labs\lab4>
```

Для того щоб оновити атрибути елемента в таблиці використовується метод `update_item()`:



```
lab4 aws_db.py
Project
lab4 C:\work\aws_labs\lab4
  venv library root
  aws_db.py
  products.py
  External Libraries
  Scratches and Consoles
aws_db.py
11 table = dynamodb.Table('Products')
12
13 table.update_item(
14     Key={
15         'Title': 'SIEMENS TE651319RW',
16         'Category': 'Coffee Machine'
17     },
18     UpdateExpression='SET Company = :vall',
19     ExpressionAttributeValues={
20         ':vall': 'SIEMENS'
21     }
22 )
23 response = table.get_item(
24     Key={
25         'Title': 'SIEMENS TE651319RW',
26         'Category': 'Coffee Machine'
27     }
28 )
29 item = response['Item']
30 print(item)
31

Terminal Local +
2023-04-21 08:17:33.809000+03:00

(venv) C:\work\aws_labs\lab4>python aws_db.py
{'year': '2023', 'Title': 'SIEMENS TE651319RW', 'Company': 'SIEMENS', 'Category': 'Coffee Machine'}
```

### 4.3. Завдання

1. Виконати кроки з порядку виконання лабораторної роботи використовуючи базу даних з індивідуального варіанту.

2. Заповнити базу даних трьома способами (AWS CLI, AWS Management Console, Python boto3) додавши мінімум 20 записів.

3. Виконати програматичне (boto3) та з використанням AWS CLI видалення декількох записів.

4. Виконати програматичну (boto3) та з використанням AWS CLI вибірку записів.

5. Результати усіх кроків оформити у вигляді детального протоколу зі скріншотами та командами в консолі які використовувалися

Варіант 1

База даних: Бібліотека

Мінімальний список характеристик: - Інвентарний номер, автор книги, назва, рік видання, ціна, чи є новим виданням, коротка анотація.

Варіант 2

База даних: Оптова база

Мінімальний список характеристик: - Код товару, назва товару, кількість на складі, вартість одиниці товару, примітки - опис товару.

Варіант 3

База даних: Договірна діяльність організації

Мінімальний список характеристик: - Шифр договору, найменування організації, терміни виконання, сума договору, примітка про вид договору.

Варіант 4

База даних: Персональні ЕОМ

Мінімальний список характеристик: - Фірма-виробник, тип процесора, тактова частота, об'єм ОЗУ, об'єм жорсткого диска, дата випуску

#### Варіант 5

База даних: Каталог відеокліпів

Мінімальний список характеристик: - Код відеострічки, дата запису, тривалість, тема, вартість.

#### Варіант 6

База даних: Замовлення

Мінімальний список характеристик: - Прізвище, ім'я, по батькові клієнта, номер рахунку, адреса, телефон.

#### Варіант 7

База даних: Телефон

Мінімальний список характеристик: - Номер телефону, прізвище, ім'я, по батькові абонента, вулиця, будинок, корпус, квартира.

#### Варіант 8

База даних: Косметичний магазин

Мінімальний список характеристик: - Назва продукції, ціна, вид продукції, кількість, фірма-виробник..

#### Варіант 9

База даних: Лісове господарство

Мінімальний список характеристик: - Найменування зеленого масиву, площа, основна порода, чи є заповідником, дата останньої перевірки.

#### Варіант 10

База даних: Викладач

Мінімальний список характеристик: - Прізвище, ім'я та по батькові викладача, посада (асистент, ст. викладач, доцент, професор), кафедра, початок роботи, домашня адреса, телефон.

Варіант 11

База даних: Знайомі

Мінімальний список характеристик: - Прізвище, ім'я та по батькові, телефон, стать, дата народження, хобі.

Варіант 12

База даних: Відділ кадрів

Мінімальний список характеристик: - Прізвище, ім'я, по батькові, домашня адреса, телефон, дата народження, посада, дата зарахування, зарплата, стаж роботи, освіта.

Варіант 13

База даних: Автомобіль

Мінімальний список характеристик: - Номерний знак автомобіля, марка автомобіля, його технічний стан, місцезнаходження автомобіля.

Варіант 14

База даних: Аеропорт

Мінімальний список характеристик: - Номер рейса, пункт призначення, дата рейса, тип літака, час вильоту, час у дорозі, чи є маршрут міжнародним, тип і ціна квитка.

Варіант 15

База даних: Домоуправління

Мінімальний список характеристик: - Номер під'їзду, номер квартири, загальна площа, корисна площа, кількість кімнат.

#### **4.4. Додаткові джерела інформації**

1. Andreas Wittig, Michael Wittig “Amazon Web Services in Action”

2. <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.html>