**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра інформатики та програмної інженерії**

**ЗВІТ**

**з лабораторної роботи №2**

**з навчальної дисципліни «Інфраструктура інформаційних систем»**

**Виконав:**

Студент 3 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Лошак В.І.

**Перевірив:**

Жаріков Е.В.

**Київ 2024**

Table of Contents

[**Комплекс лабораторних робіт «Побудова інфраструктури з використанням хмарних послуг AWS, що реалізують обчислення, сховища та мережі»** 4](#_Toc165976193)

[Access the AWS Management Console 4](#_Toc165976194)

[1. Module 1: Launching an EC2 instance and attaching an EBS volume 5](#_Toc165976195)

[1.1. Task 1. Start creating the instance and assign a name 5](#_Toc165976196)

[1.2. Task 2. Application and OS Images 6](#_Toc165976197)

[1.3. Task 3. Choose an instance type 7](#_Toc165976198)

[1.4. Task 4. Choose a key pair 7](#_Toc165976199)

[1.5. Task 5. Network settings 8](#_Toc165976200)

[1.6. Task 6. Configure storage 9](#_Toc165976201)

[1.7. Task 7. Advanced details 9](#_Toc165976202)

[1.8. Task 8. Review the instance and launch 11](#_Toc165976203)

[1.9. Task 9. Access your EC2 instance 12](#_Toc165976204)

[1.10. Task 10. Update the security group 13](#_Toc165976205)

[1.11. Task 11. Create an inbound rule 14](#_Toc165976206)

[1.12. Task 12. Test the rule 15](#_Toc165976207)

[1.13. Task 13. Attach an EBS volume to your EC2 instance 15](#_Toc165976208)

[2. Module 2: Creating an S3 bucket 18](#_Toc165976209)

[2.1. Task 1. Create an S3 bucket 18](#_Toc165976210)

[2.2. Task 2. Add a bucket policy to make the content publicly available 20](#_Toc165976211)

[2.3. Task 3. Upload an HTML document 22](#_Toc165976212)

[2.4. Task 4. Test your website 24](#_Toc165976213)

[3. Module 4: Building the Amazon Virtual Private Cloud 26](#_Toc165976214)

[3.1. Create a VPC 26](#_Toc165976215)

[3.2. Create Your Public Subnets 27](#_Toc165976216)

[3.3. Create an Internet Gateway 31](#_Toc165976217)

[3.4. Create a Route Table, Add Routes, And Associate Public Subnets 32](#_Toc165976218)

[3.5. Create a Security Group for your Web Server 35](#_Toc165976219)

[3.6. Launch a Web Server in your Public Subnet 36](#_Toc165976220)

[3.7. Create Private Subnets for your MySQL Server 40](#_Toc165976221)

[3.8. Create a Security Group for your Database Server 41](#_Toc165976222)

[3.9. Create a Database Subnet Group 43](#_Toc165976223)

[3.10. Create an Amazon RDS Database 44](#_Toc165976224)

[3.11. Connect Your Address Book Application to Your Database 47](#_Toc165976225)

[4. Module 5: Introduction to AWS Lambda 48](#_Toc165976226)

[4.1. Create the Amazon S3 Buckets 49](#_Toc165976227)

[4.2. Create an AWS Lambda Function 51](#_Toc165976228)

[4.3. Test Your Function 55](#_Toc165976229)

[4.4. Monitoring and Logging 57](#_Toc165976230)

# **Комплекс лабораторних робіт «Побудова інфраструктури з використанням хмарних послуг AWS, що реалізують обчислення, сховища та мережі»**

Access the AWS Management Console

A screenshot of a computer

Description automatically generated

Після натискання на кнопку Start вдалося отримати доступ до консолі AWS для введення подальших команд.

A screenshot of a computer

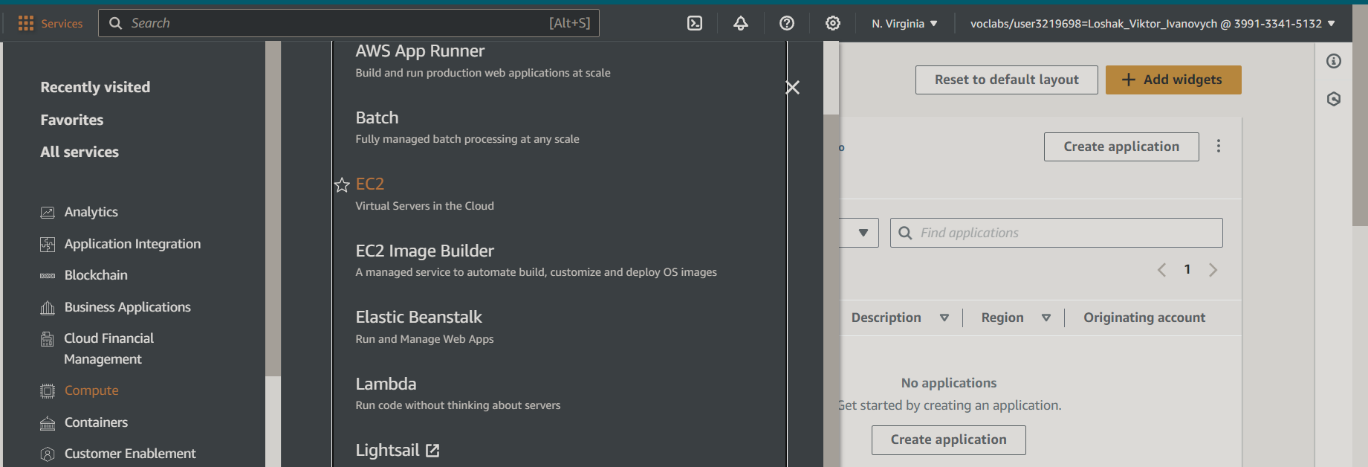
Description automatically generated

Інтерфейс консолі AWS після натискання на клавішу AWS має наведений вище вигляд.

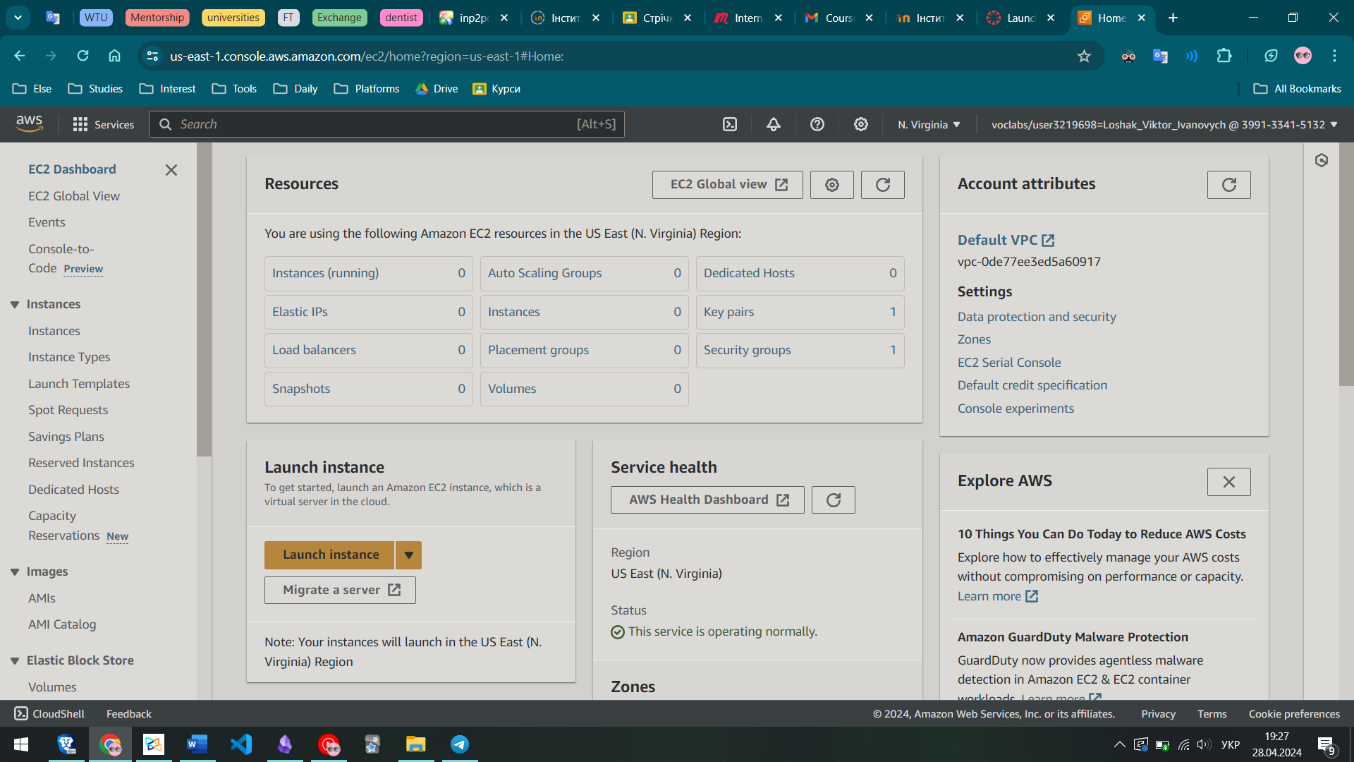
1. Module 1: Launching an EC2 instance and attaching an EBS volume

**Завдання:** У цій лабораторній роботі ви створюєте екземпляр Amazon Elastic Compute Cloud (Amazon EC2), який розміщує простий веб-сайт, а потім приєднуєте до нього том Amazon Elastic Block Store (Amazon EBS). «Ваше ім’я» необхідно замінити на ім’я та прізвище студента.

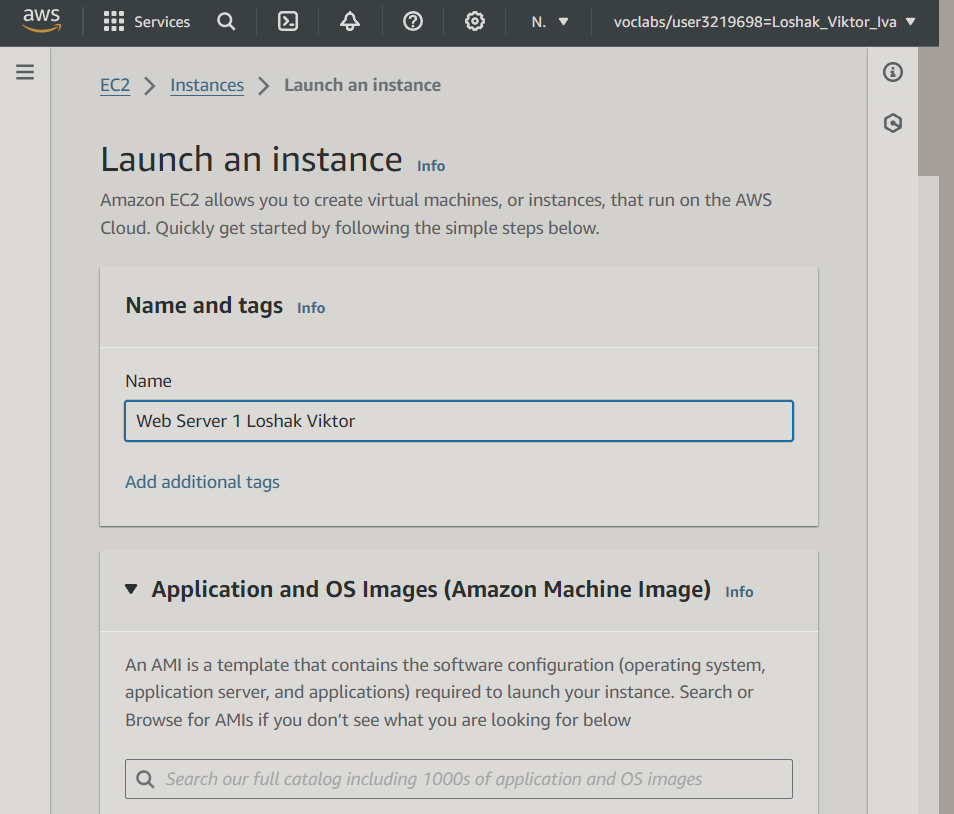
* 1. Task 1. Start creating the instance and assign a name



Я відкрив головне меню AWS і вибрав EC2 серед усіх доступних сервісів. EC2 використовується для запуску віртуальних серверів у хмарі.

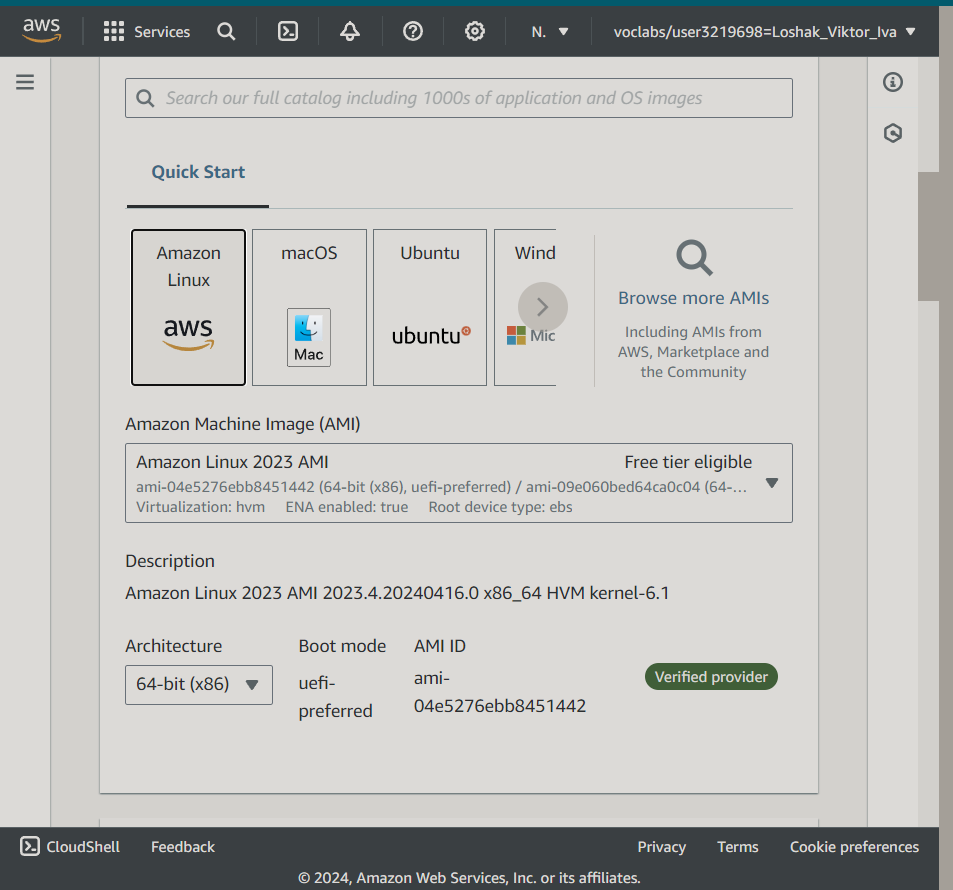


На інформаційній панелі EC2, де можна побачити огляд поточних ресурсів, таких як запущені інстанси, доступні IP-адреси, балансувальники навантаження тощо. Я натиснув на кнопку "Launch instance"



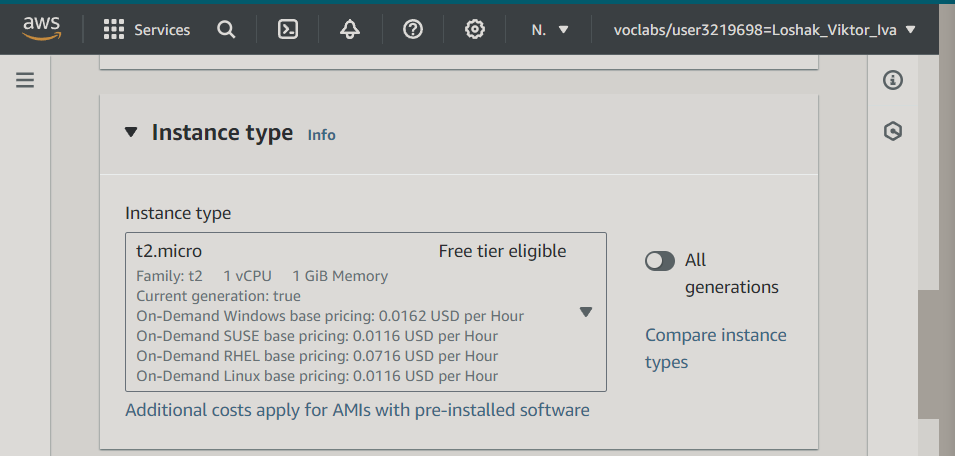
На цьому кроці я почав процес запуску нового EC2 інстансу. Я ввів назву "Web Server 1 Loshak Viktor" для майбутнього сервера, щоб легко його ідентифікувати серед інших інстансів. Я маю можливість додати додаткові теги для організації та управління моїм інстансом.

* 1. Task 2. Application and OS Images



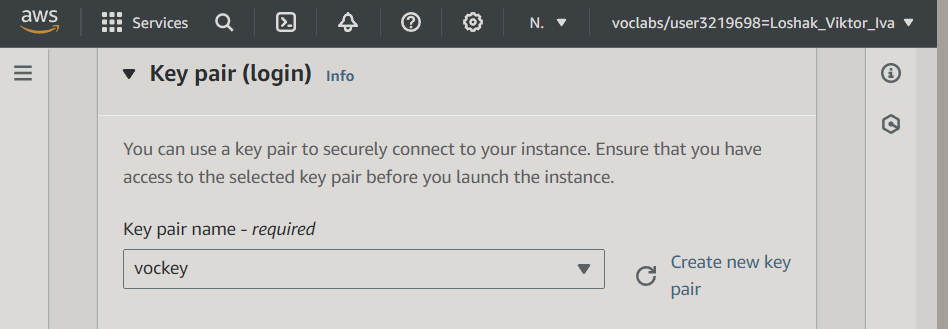
Я зупинив свій вибір на Amazon Linux 2023 AMI(Amazon machine image), оскільки воно підтримується та оптимізоване для хмарних сервісів AWS і входить до безкоштовного тарифного плану.

* 1. Task 3. Choose an instance type



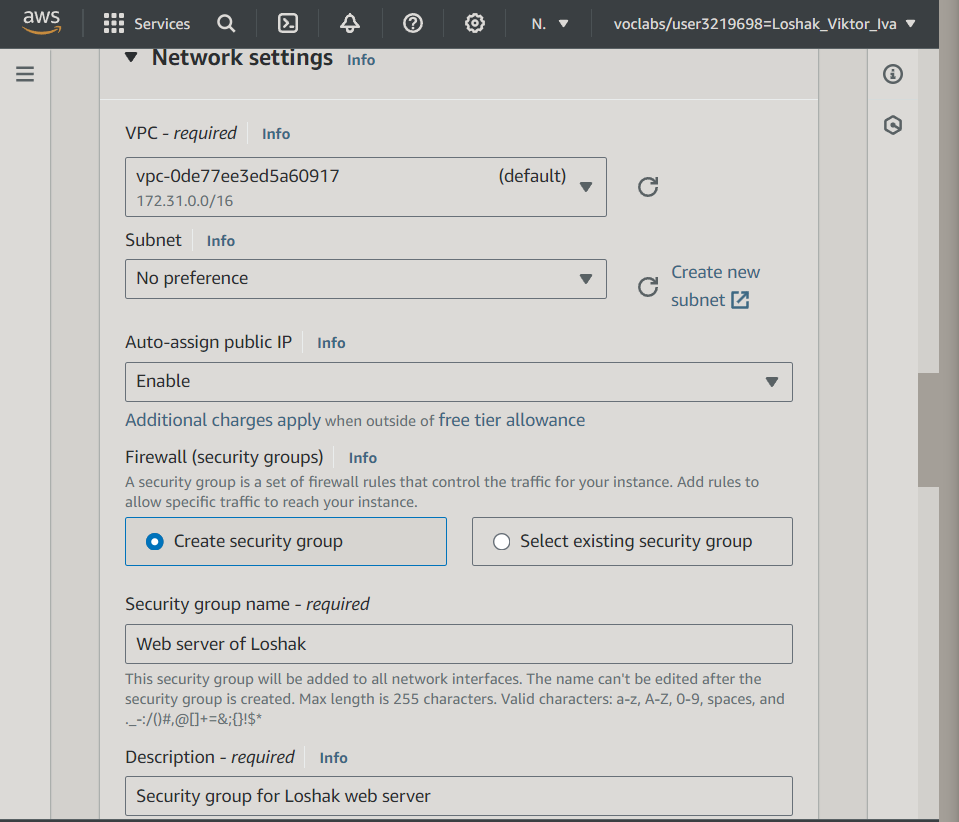
Я обрав "t2.micro" через те, що він входить до безкоштовного тарифного плану. Цей тип інстансу забезпечує 1 vCPU та 1 GB пам'яті, що більш ніж достатньо для основних завдань, які я планую виконати.

* 1. Task 4. Choose a key pair

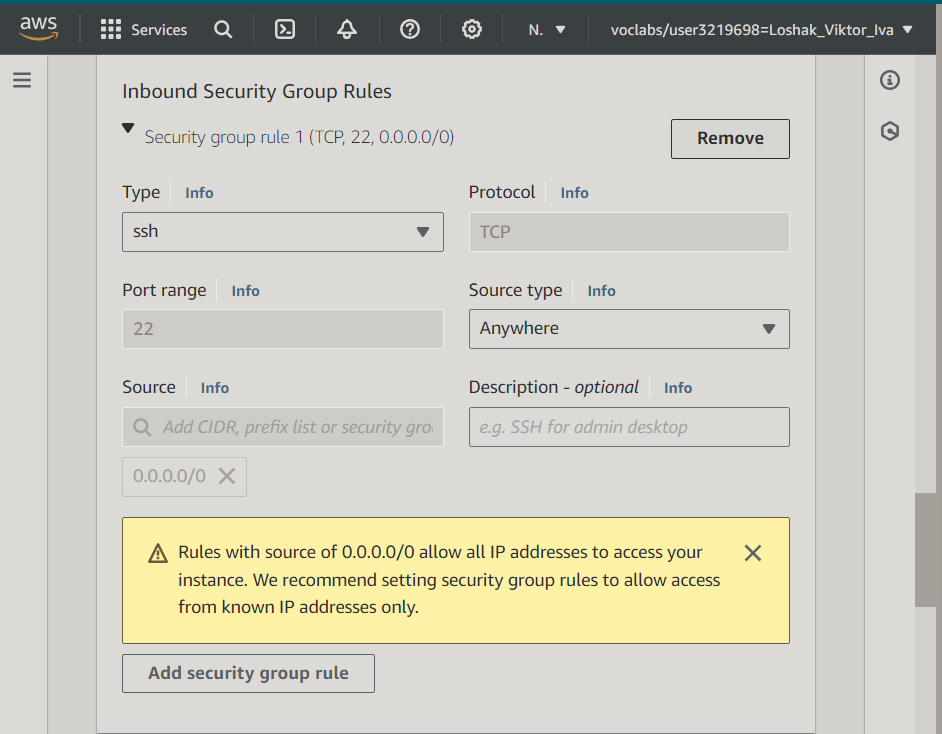


Обравши назву ключової пари "vockey", я підготував надійний спосіб доступу до інстансу. Використання іменованої ключової пари — стандартна процедура безпеки, котра забезпечує безпечне SSH-з'єднання.

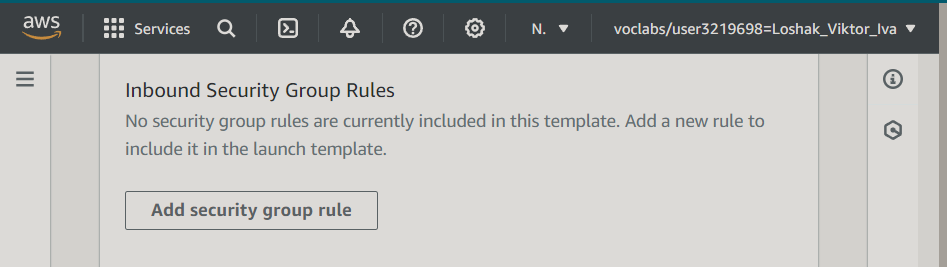
* 1. Task 5. Network settings



Настроювання мережевих параметрів було ключовим етапом. Я визначив назву групи безпеки як "Web server of Loshak" і надав опис, щоб згодом легше ідентифікувати призначення створеної групи

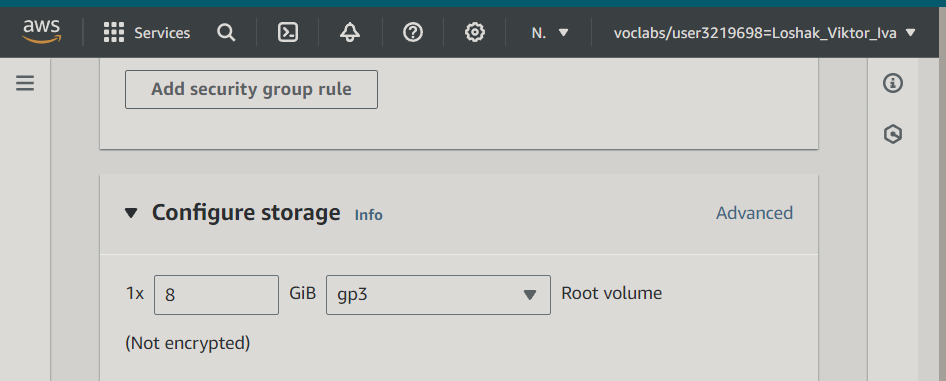


Бачимо що доступ до інстансу через SSH (порт 22) дозволений з будь-якої IP-адреси, що забезпечить мені доступ до сервера з будь-якого місця



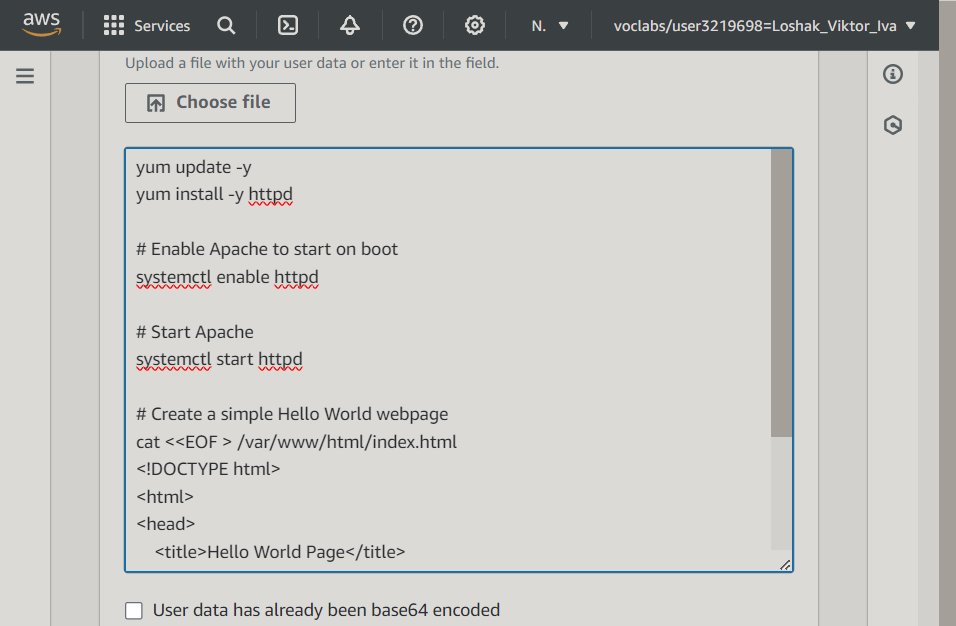
Останній крок перед запуском інстансу — видалення існуючого правила. Я переконався, що жодні зайві правила не включені, що мінімізує ризики небажаного доступу.

* 1. Task 6. Configure storage



Я конфігурував сховище для інстансу, встановивши 8 ГБ на твердотільному диску gp3 як основний том. Такий обсяг достатній для запуску веб-сервера та базових додатків

* 1. Task 7. Advanced details



У розділі даних користувача я ввів скрипт для автоматичного оновлення системи та встановлення веб-сервера Apache при першому запуску інстансу. Команди включають активування та запуск httpd сервісу, а також створення простої веб-сторінки з написом "Hello World".

Script contents:

#!/bin/bash

# Update the server

yum update -y

# Install Apache web server

yum install -y httpd

# Enable Apache to start on boot

systemctl enable httpd

# Start Apache

systemctl start httpd

# Create a simple Hello World webpage

cat <<EOF > /var/www/html/index.html

<!DOCTYPE html>

<html>

<head>

    <title>Hello World Page Loshak</title>

</head>

<body>

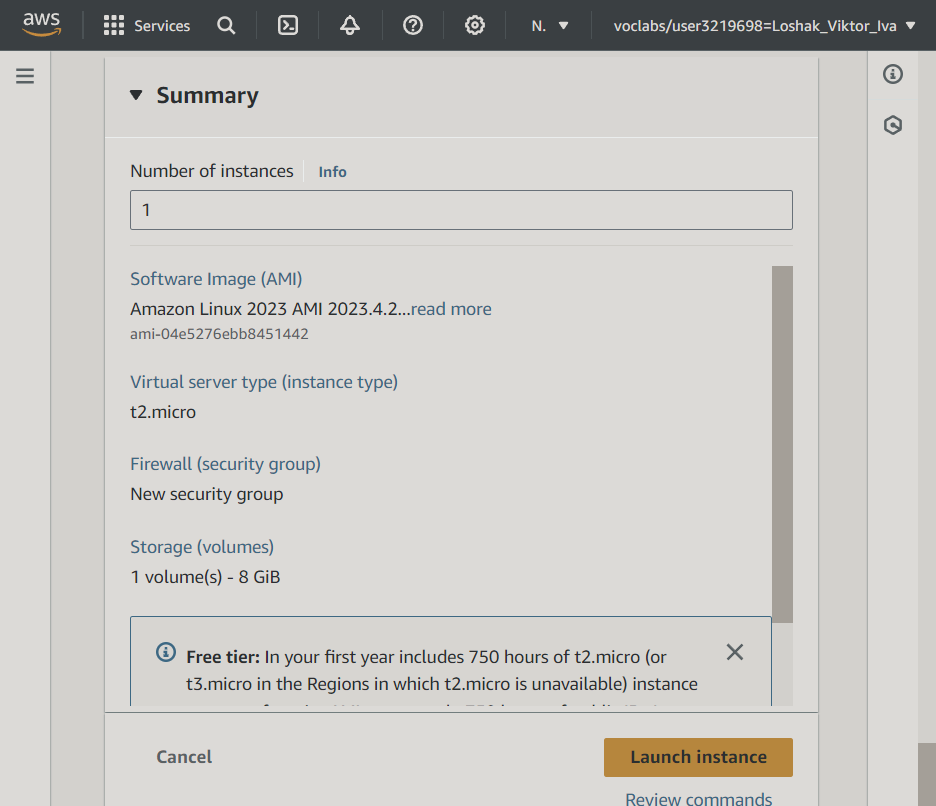
    <h1>Hello World from Loshak!</h1>

</body>

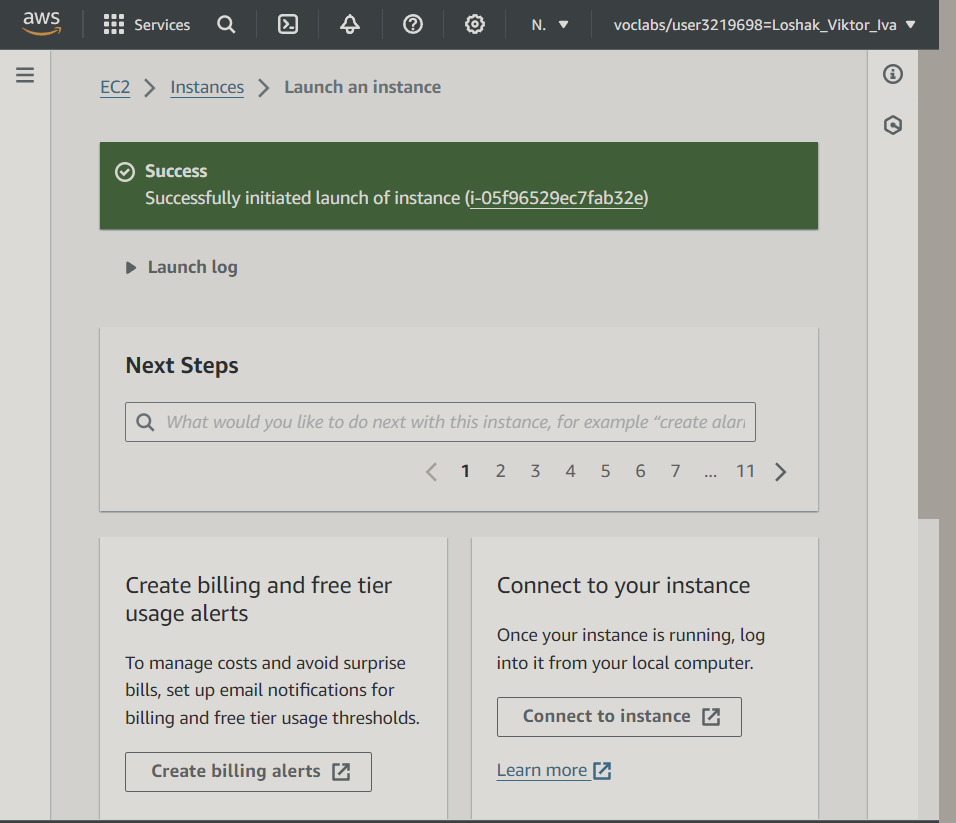
</html>

EOF

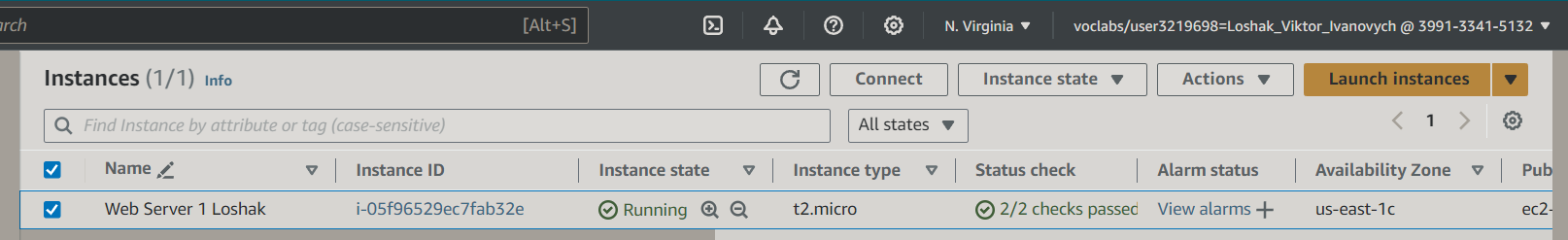
* 1. Task 8. Review the instance and launch



Завершуючи налаштування, я перевірив усі параметри інстансу в розділі підсумку. Всі виглядає коректно: обрано один інстанс типу t2.micro із встановленим Amazon Linux AMI, новоствореною групою безпеки та 8 ГБ сховища.

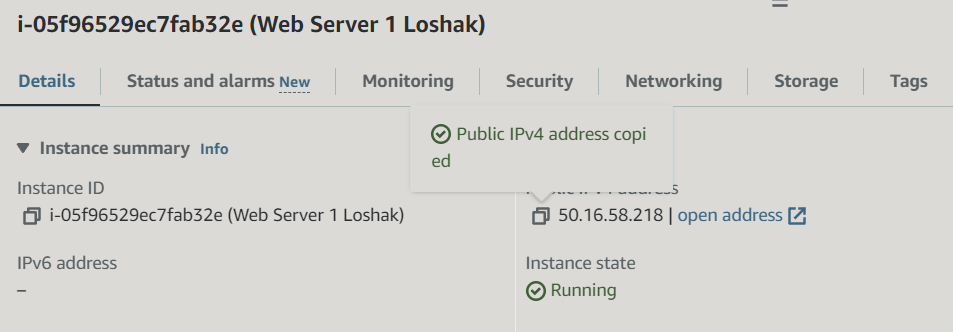
‘

Після натискання "Launch instance" я отримав підтвердження про успішний запуск інстансу. AWS також запропонував наступні кроки для управління інстансом, включно з підключенням до нього та налаштуванням оповіщень про витрати.

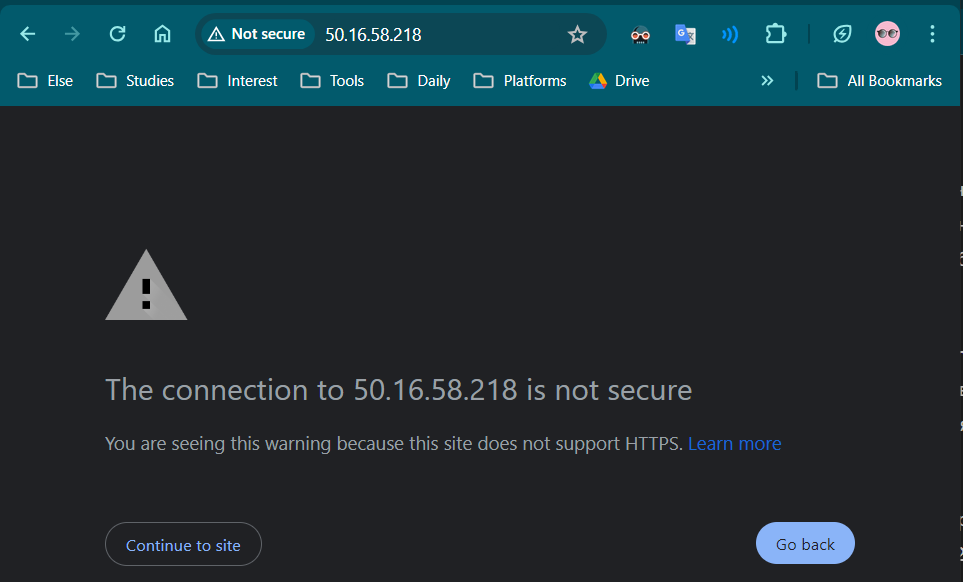


На останньому знімку екрану видно, що інстанс "Web Server 1 Loshak" успішно запущений і працює, як вказано в EC2 dashboard.

* 1. Task 9. Access your EC2 instance

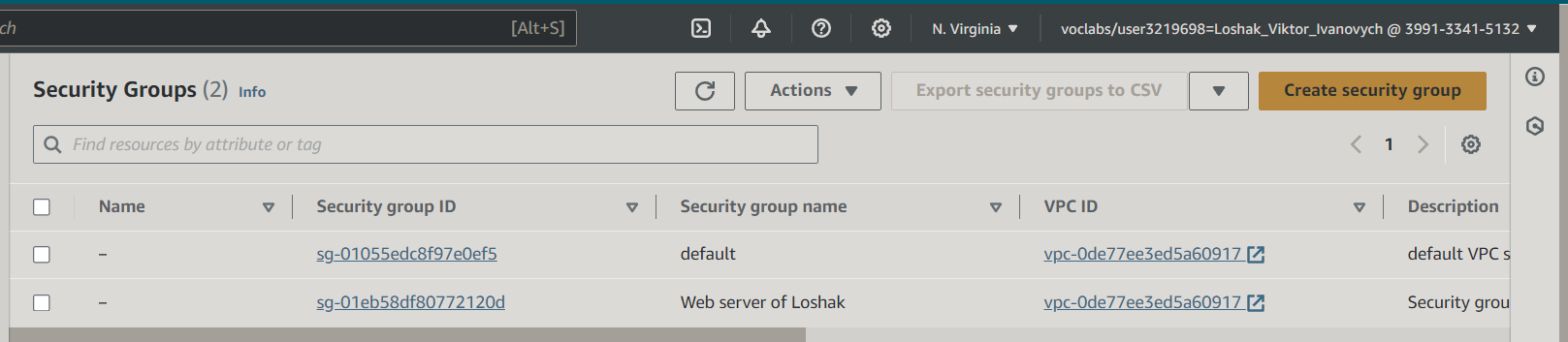


Скопіювавши публічну IPv4 адресу інстансу, я готовий підключитися до веб-сервера.

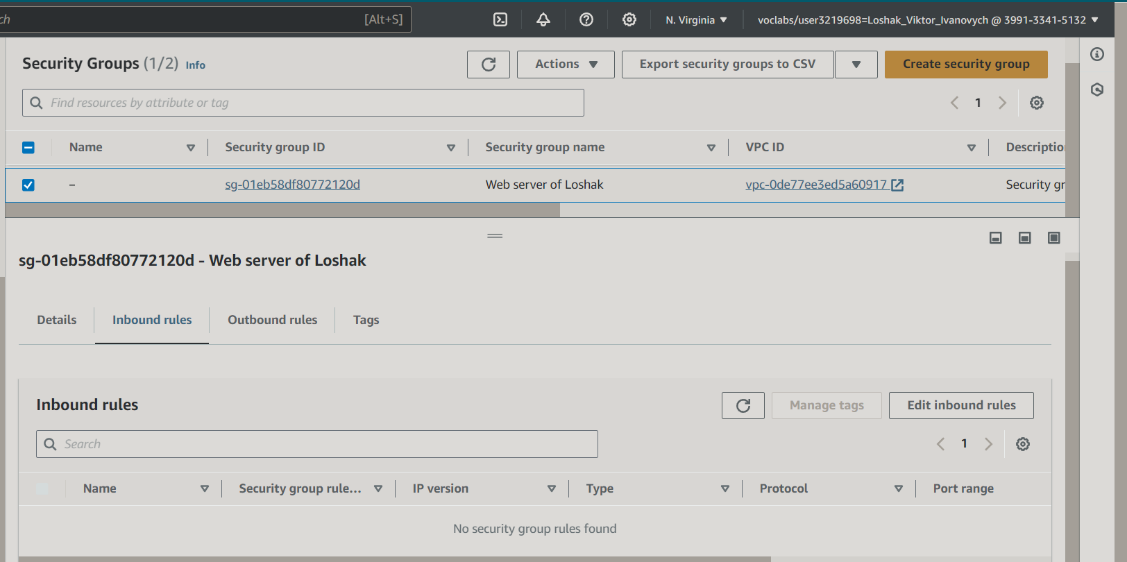


При спробі доступу до веб-сервера через браузер, я стикнувся з попередженням про безпеку. Після прийняття умов з’єднання сервер видав помилку про вичерпання часу очікування. Оновимо security group щоб отримати доступ до сторінки.

* 1. Task 10. Update the security group

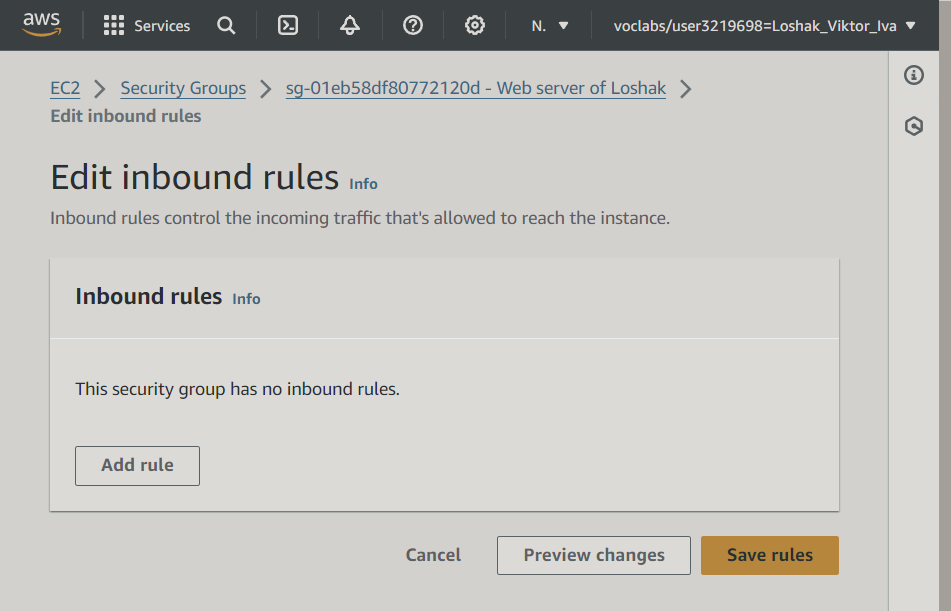


Переглядаю список груп безпеки, де видно створену мною групу "Web server of Loshak". Вона відіграє ключову роль у захисті мого інстансу, визначаючи, який трафік дозволений до та з мого сервера.

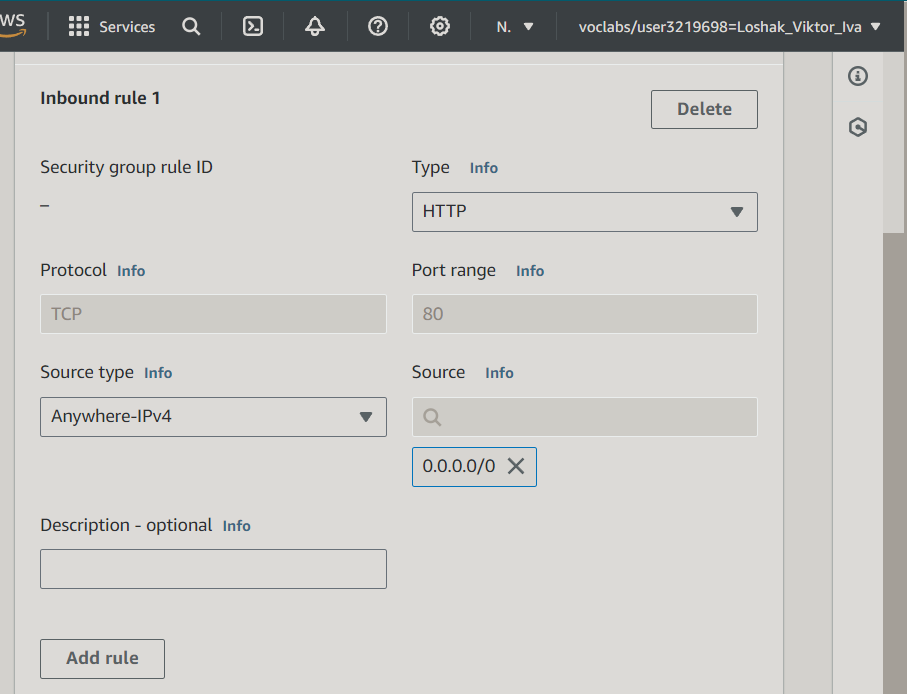


У деталях групи безпеки я переконуюсь, що інбаунд правила відсутні.

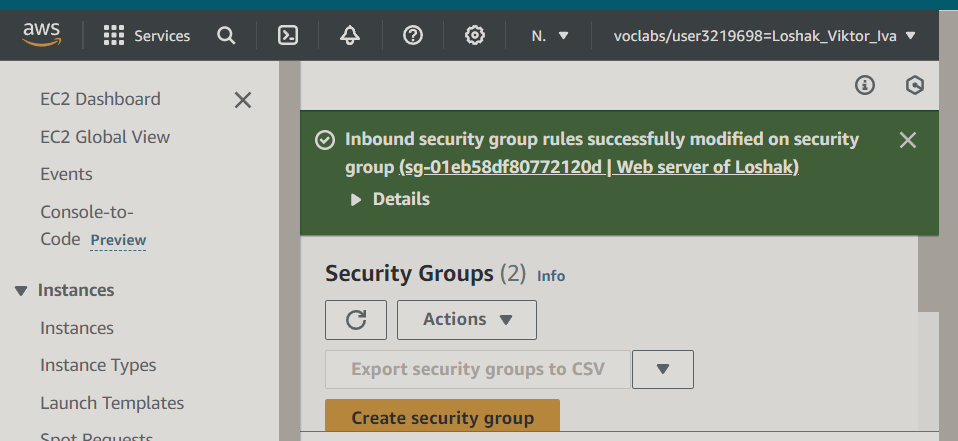
* 1. Task 11. Create an inbound rule



Я зайшов у налаштування групи безпеки, аби додати вхідні правила. Наразі жодні правила не були встановлені.

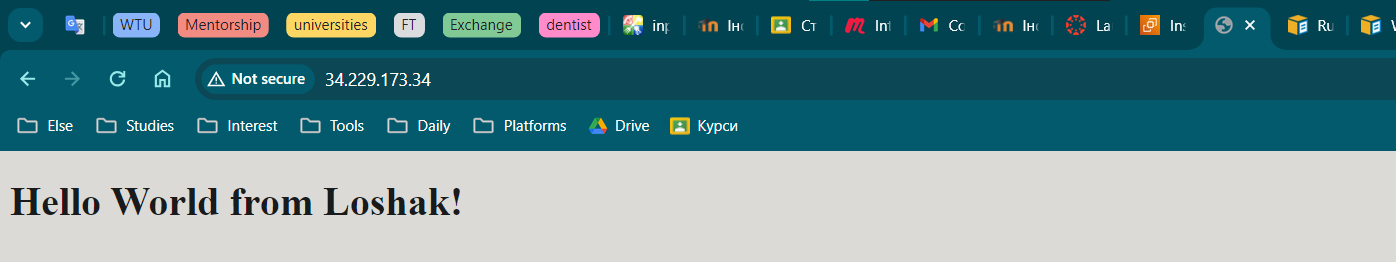


Додав правило для HTTP трафіку, дозволяючи з'єднання на порт 80 з будь-якої IP-адреси. Це необхідний крок для доступу до веб-сервера.



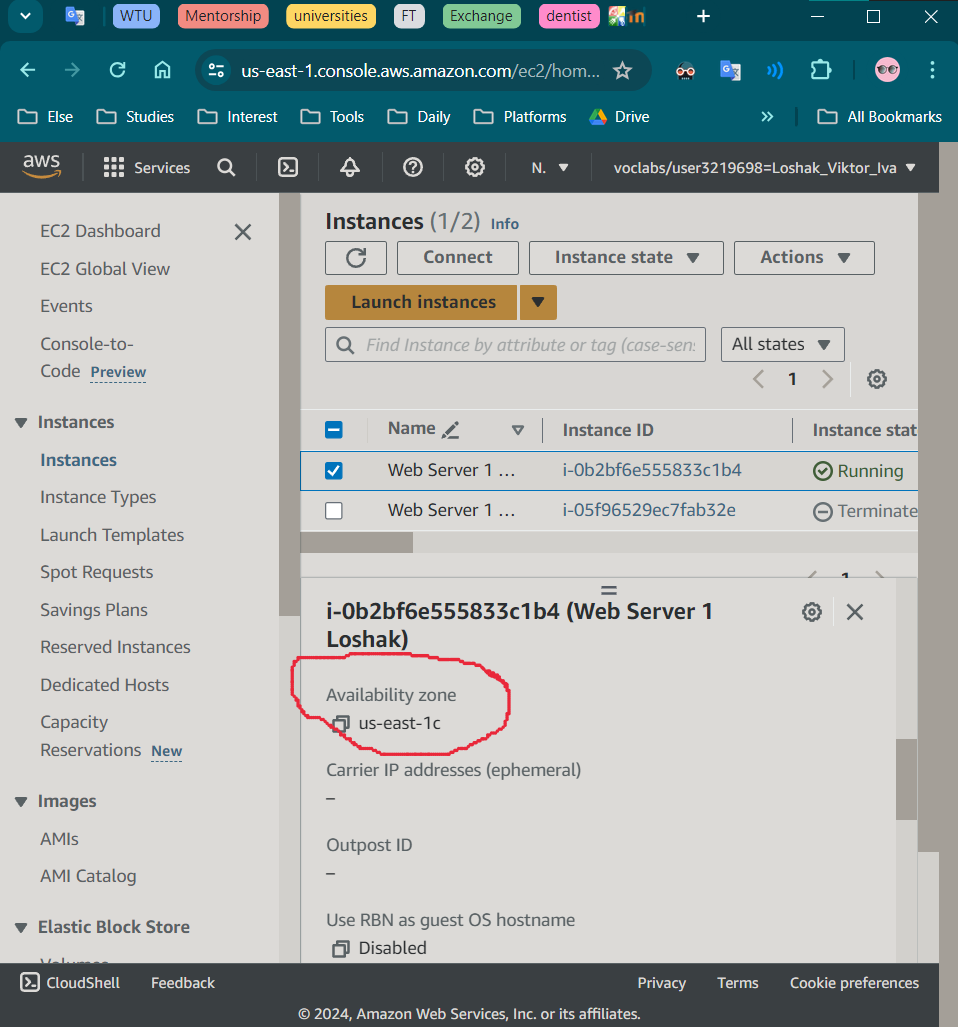
Підтвердив зміни у правилах групи безпеки, що дозволять вхідний HTTP трафік до мого інстансу.

* 1. Task 12. Test the rule

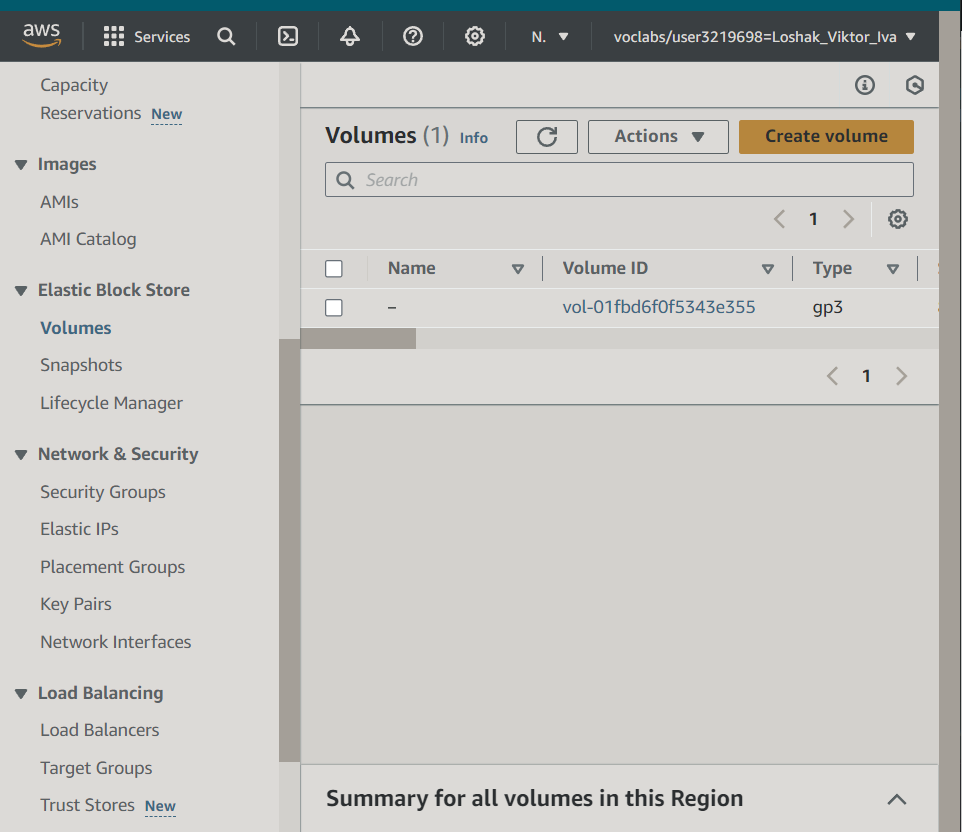


Відкривши публічну IP-адресу мого інстансу у веб-браузері, я бачу сторінку з написом "Hello World from Loshak!". Це підтверджує, що веб-сервер Apache працює, і мій скрипт ініціалізації успішно виконався.

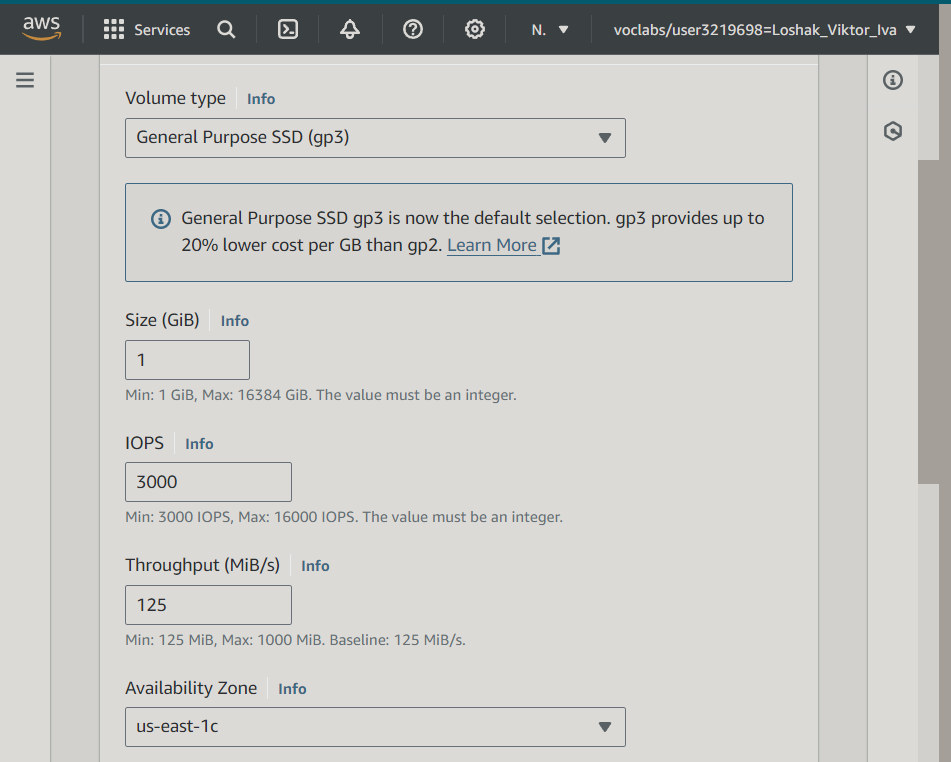
* 1. Task 13. Attach an EBS volume to your EC2 instance



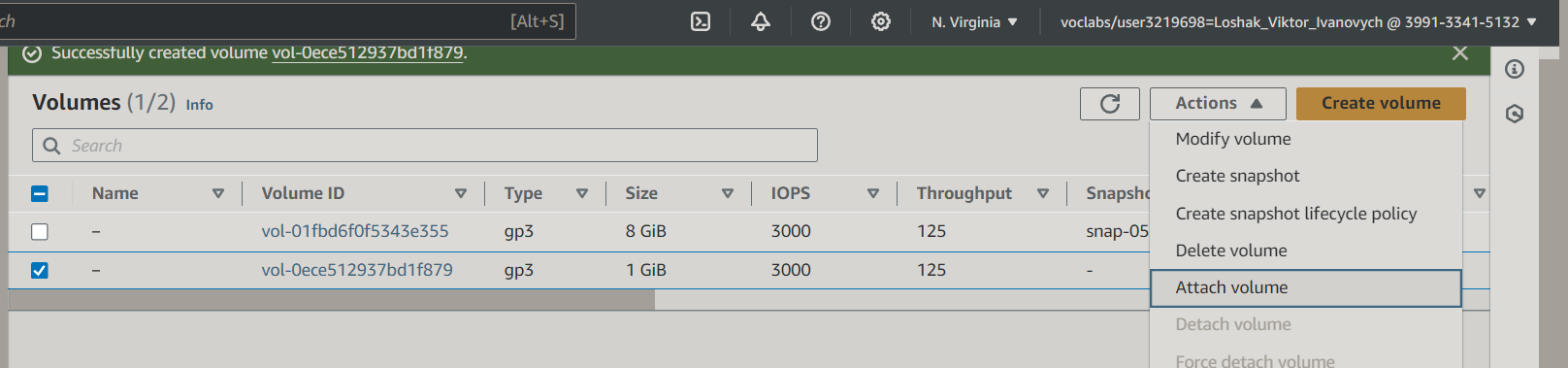
Переглядаю інформацію про інстанс "Web Server 1 Loshak", знаходжу доступну інформацію про зону доступності сервера.



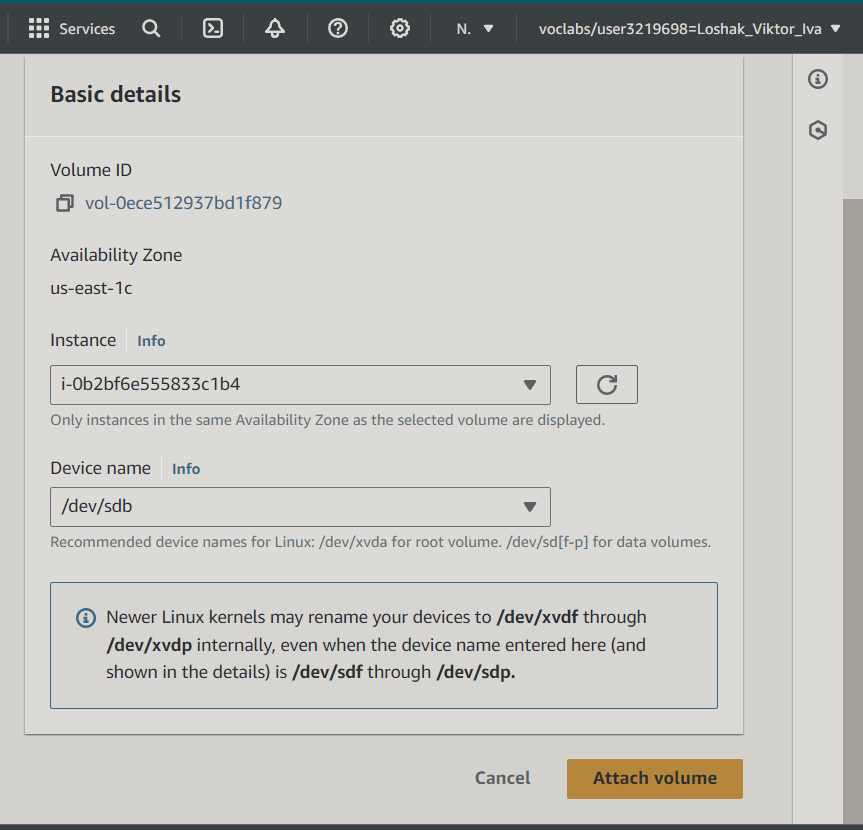
Тут я перевіряю список томів Elastic Block Store, де відображений один активний том gp3. Це підтверджує, що сховище для мого інстансу готове до використання.



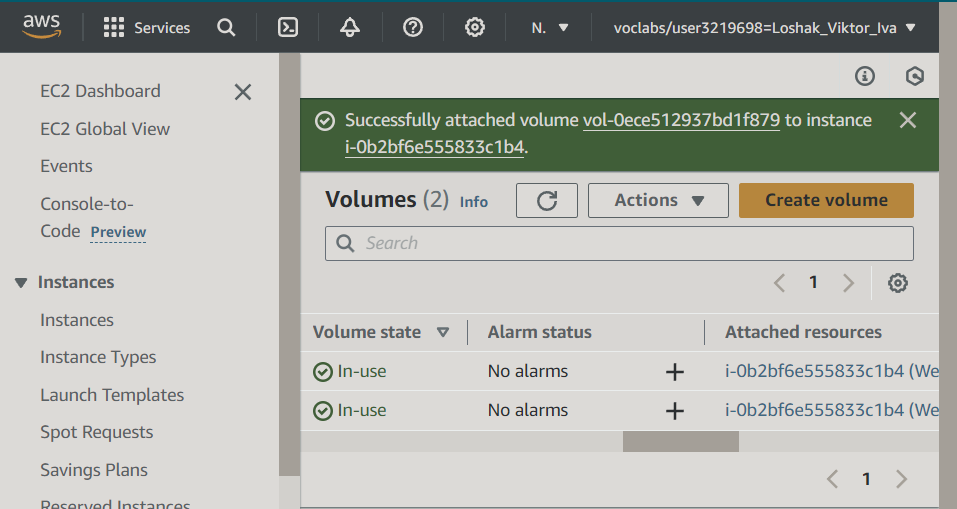
На цьому скріншоті я створюю новий том, вибираючи тип gp3 для оптимальної продуктивності та вартості. Я також вказую ту саму зону доступності що і в сервера



Том успішно створений та готовий до приєднання. З'явився у списку томів, підтверджуючи, що я маю два доступних томи.



Останнім кроком є приєднання новоствореного тома до інстансу. Вибравши потрібний інстанс я завершую процес.

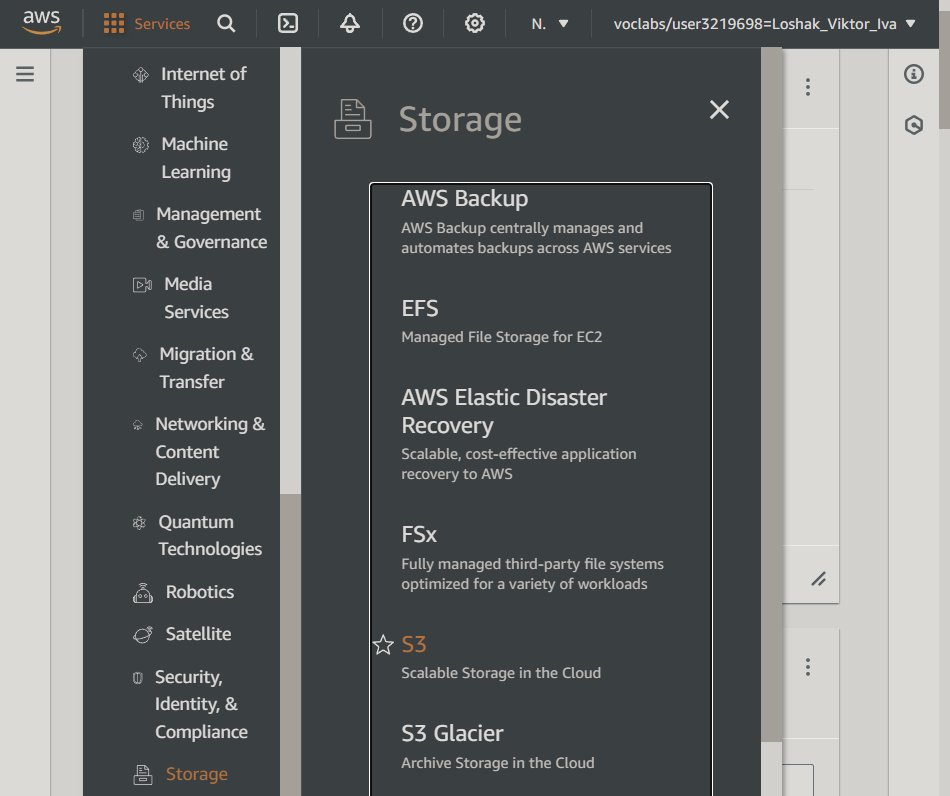


Том успішно приєднано до інстансу, як показано у консолі EC2. Тепер у мене є додатковий об'єм сховища для роботи з даними.

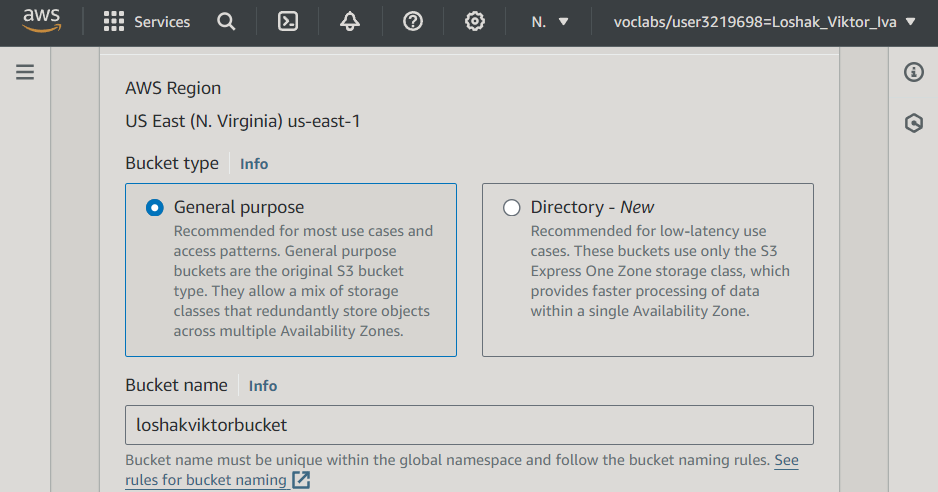
1. Module 2: Creating an S3 bucket

**Завдання:** Виконайте ці кроки, щоб створити сегмент Amazon Simple Storage Service (Amazon S3) для розміщення статичного веб-сайту. Статичний веб-сайт є фіксованим і відображає однаковий вміст для кожного користувача. Натомість, динамічний веб-сайт використовує розширене програмування для забезпечення взаємодії з користувачем і відображення різного вмісту залежно від вибору користувача.

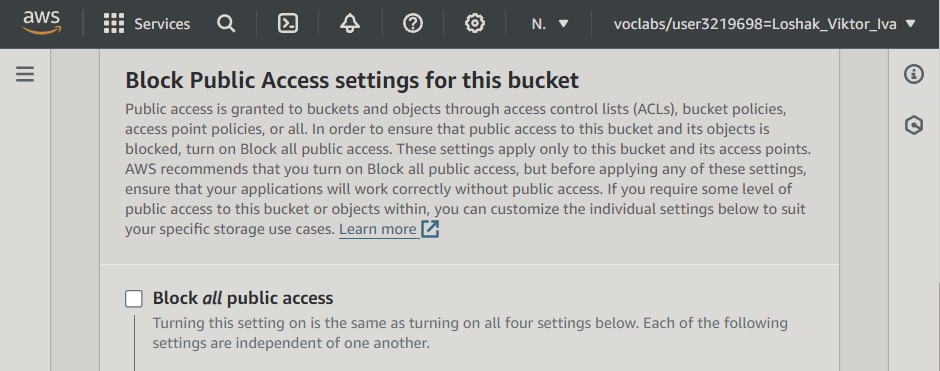
* 1. Task 1. Create an S3 bucket



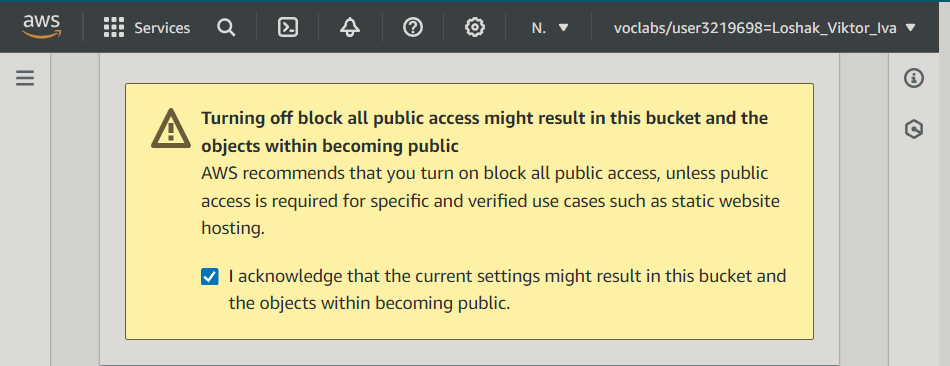
Я відкрив меню сервісів AWS і вибрав S3 для створення нового сховища, важливого для зберігання та управління об'єктами даних.



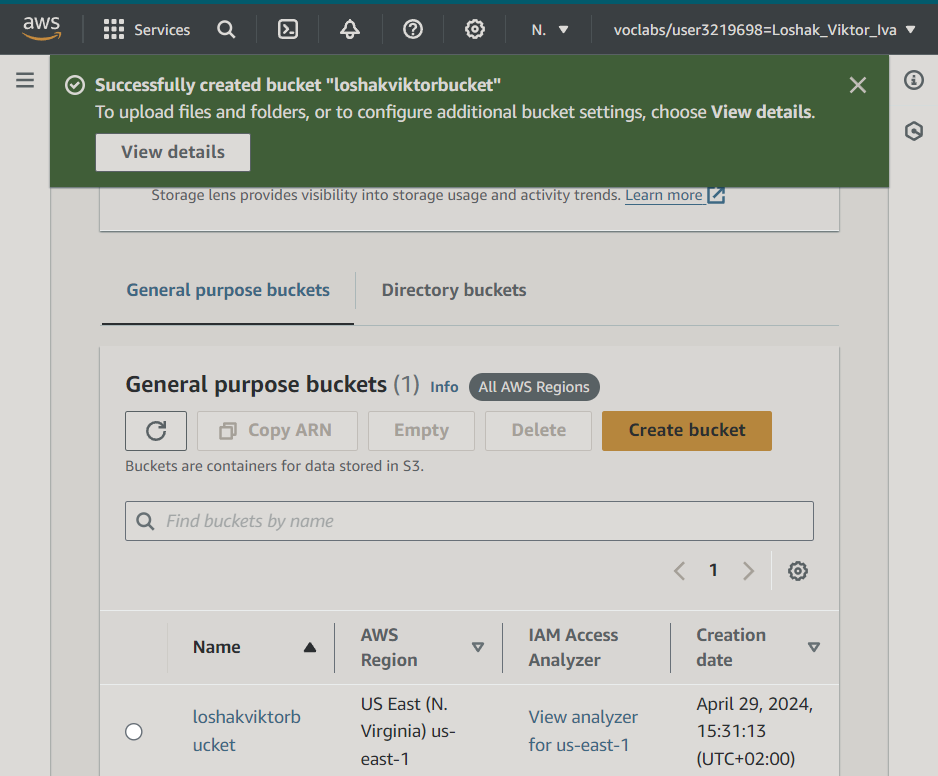
Налаштувавши параметри для нового S3 бакета, я задав йому унікальну назву, яка є обов'язковою для ідентифікації у глобальному просторі AWS.



Я заблокував усі публічні доступи до бакета, щоб забезпечити безпеку зберіганих даних, відповідно до рекомендацій AWS.

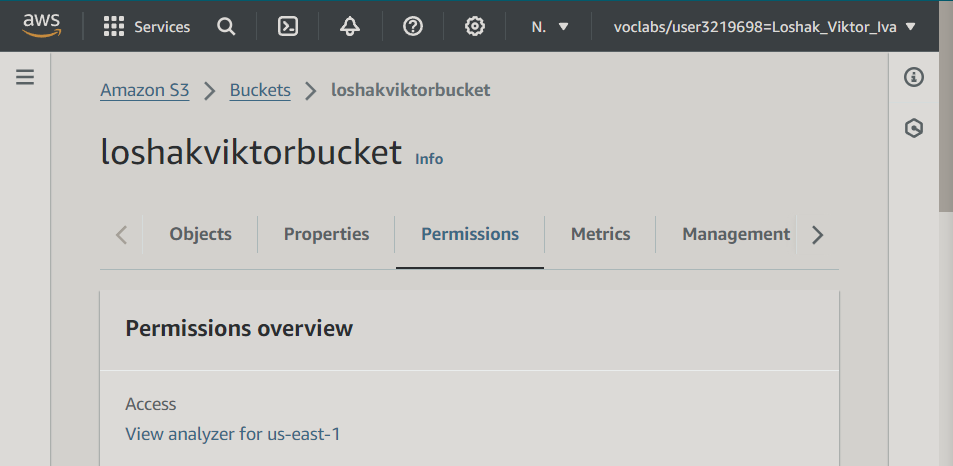


Підтвердив, що розумію наслідки відключення блокування публічного доступу, що може бути необхідним для певних сценаріїв використання.

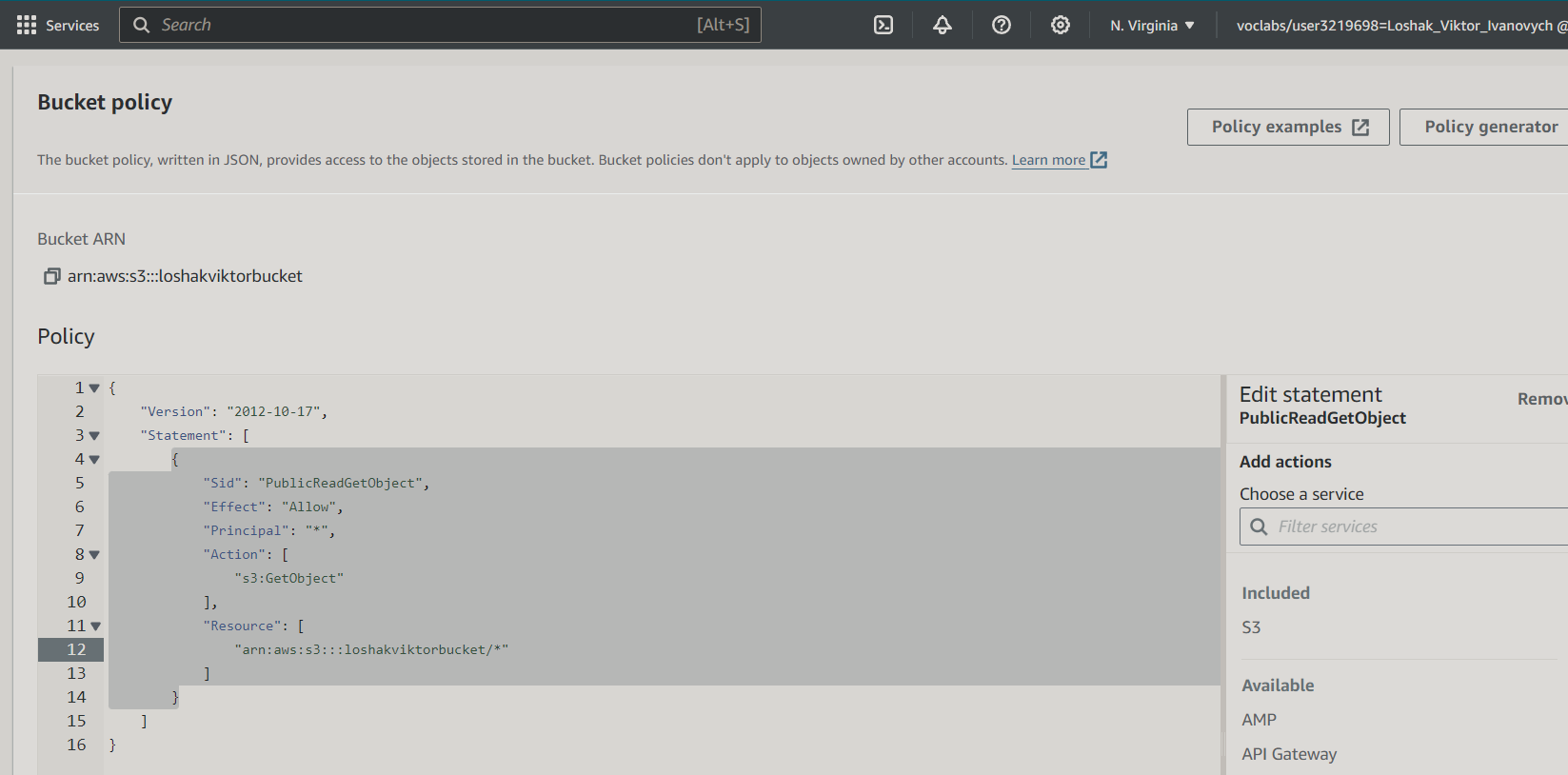


Бакет успішно створено, і тепер я можу завантажувати файли та керувати додатковими налаштуваннями бакета.

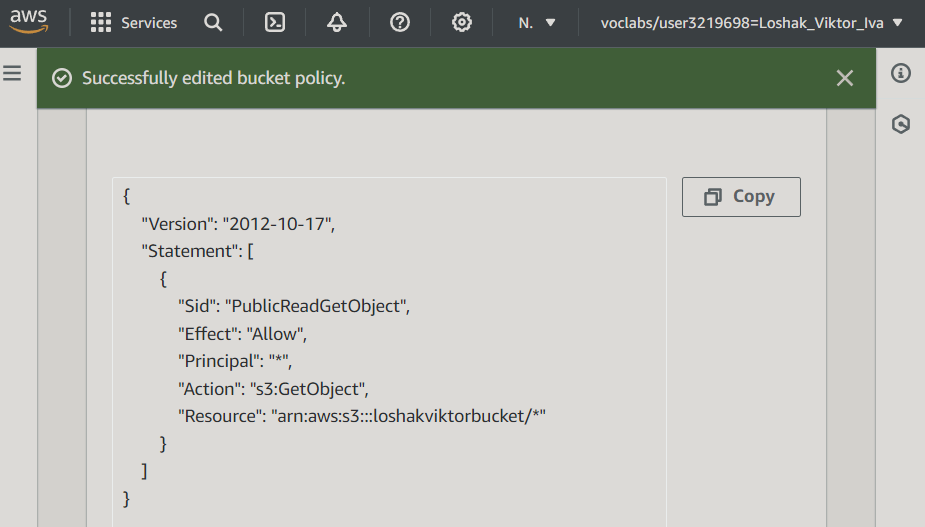
* 1. Task 2. Add a bucket policy to make the content publicly available



Відкрив вкладку дозволів для новоствореного S3 бакета, переконавшись, що доступність налаштована відповідно до вимог безпеки.



Я створив політику для S3 бакета, яка дозволяє публічний читальний доступ до об'єктів, зазначених у політиці. Це налаштування важливе для роздачі статичного контенту.



Після редагування, я зберіг нову політику бакета. Політика відтепер дозволить користувачам читати файли у бакеті, що необхідно для широкої публікації даних.

Json що відіображає налаштування політики бакета:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Personal Information</title>

</head>

<body>

    <h1>Welcome to My Page</h1>

    <p><strong>Name:</strong> Loshak Viktor</p>

    <p><strong>University:</strong> Igor Sikorsky Kyiv Polytechnic Institute</p>

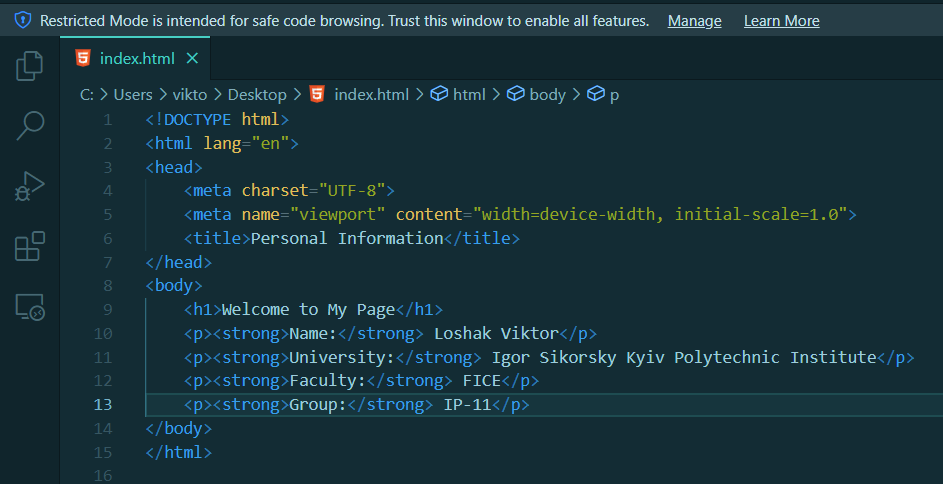
    <p><strong>Faculty:</strong> FICE</p>

    <p><strong>Group:</strong> IP-11</p>

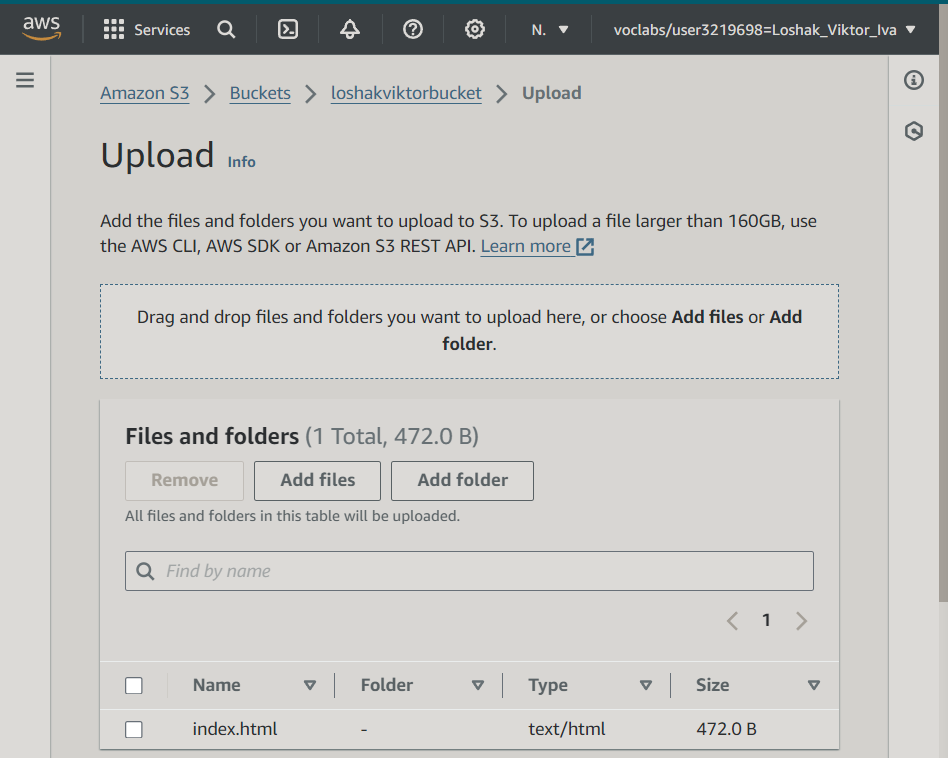
</body>

</html>

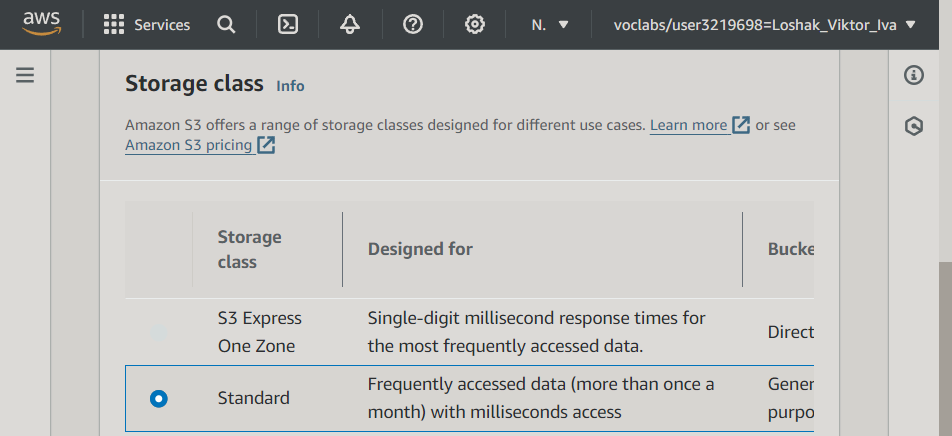
* 1. Task 3. Upload an HTML document



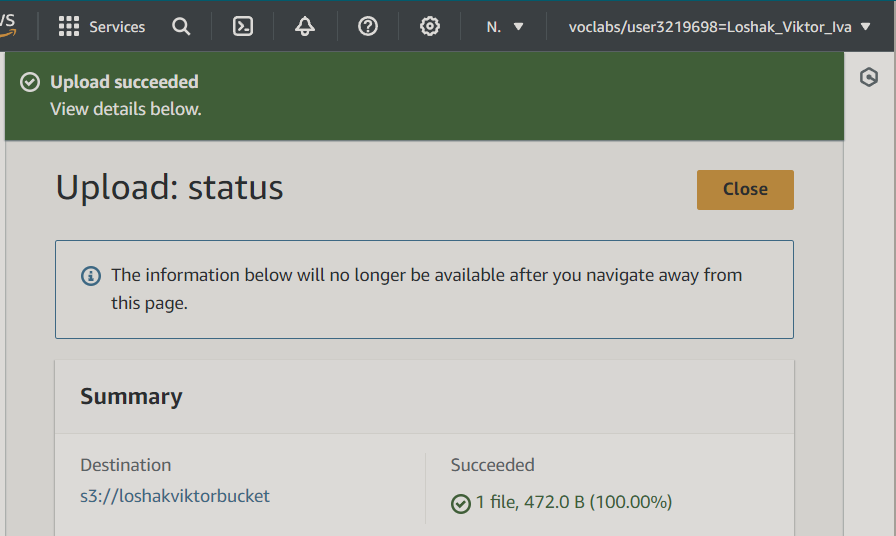
Створив основну веб-сторінку в HTML для публікації. На ній зазначено моє ім'я, університет та факультет. Файл називається `index.html`.



Завантажив `index.html` до бакету S3, яке я створив у попередніх кроках. Веб-сторінка повинна відображатися як головна сторінка мого веб-сайту.

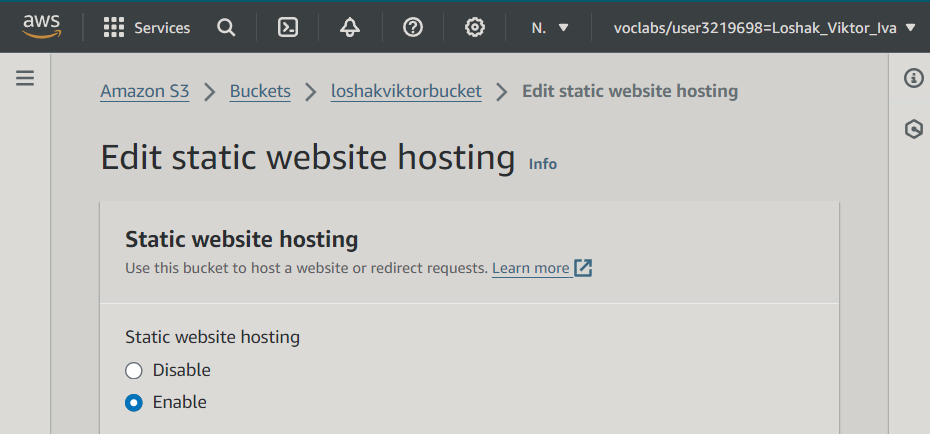


Обрав тип сховища `Standard` для файлу `index.html` в S3, оскільки цей клас оптимальний для файлів, до яких часто звертаються.

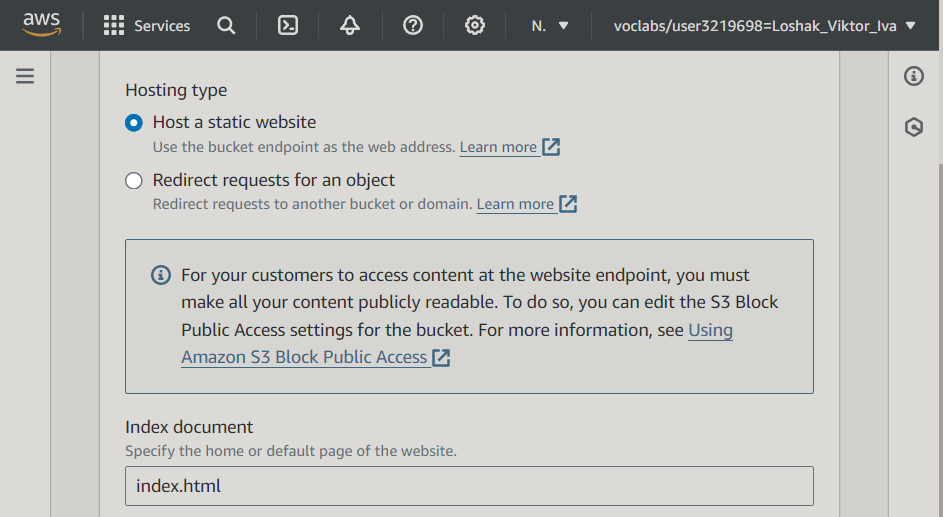


Успішно завантажив файл `index.html` до мого відра S3, і тепер він доступний для використання через Інтернет.

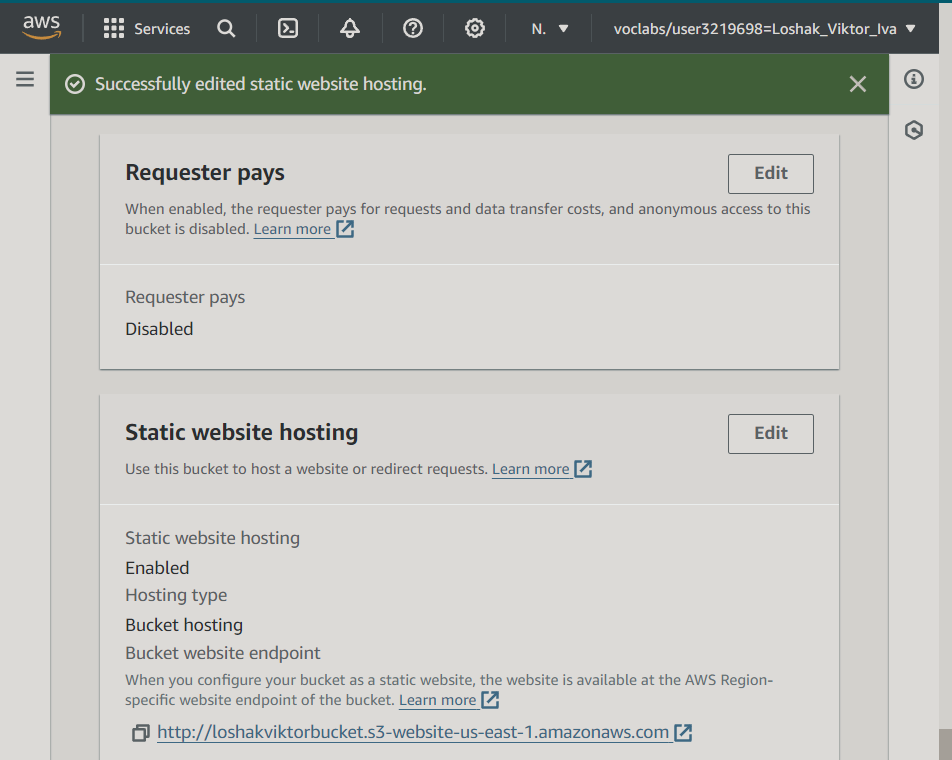
* 1. Task 4. Test your website



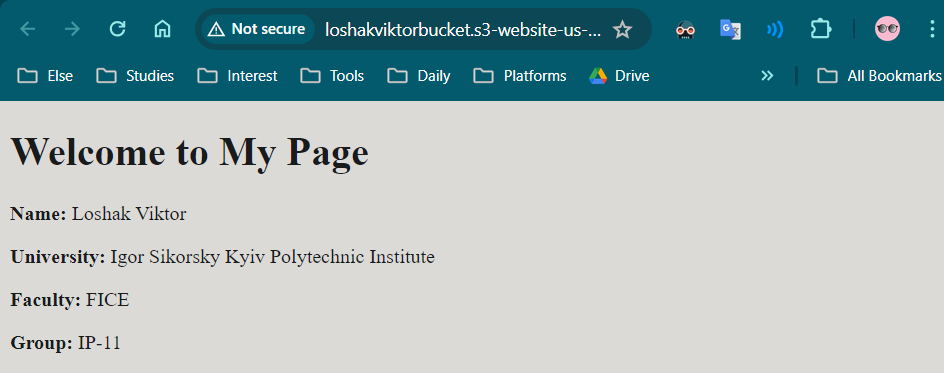
Я відкрив налаштування хостингу статичного веб-сайту в своєму S3 бакеті на AWS. На цьому етапі я обираю "Enable", щоб активувати функцію хостингу статичного веб-сайту.



Тут я встановлюю параметри хостингу: вибираю опцію "Host a static website" і вказую назву індексного документа для мого веб-сайту, який буде "index.html". Це необхідно, щоб S3 бакет знав, яку сторінку відображати як домашню.



Після збереження налаштувань, AWS повідомляє мене, що налаштування статичного веб-хостингу були успішно оновлені. Тепер у мене є посилання на кінцеву точку мого бакета, де я можу переглянути свій веб-сайт.

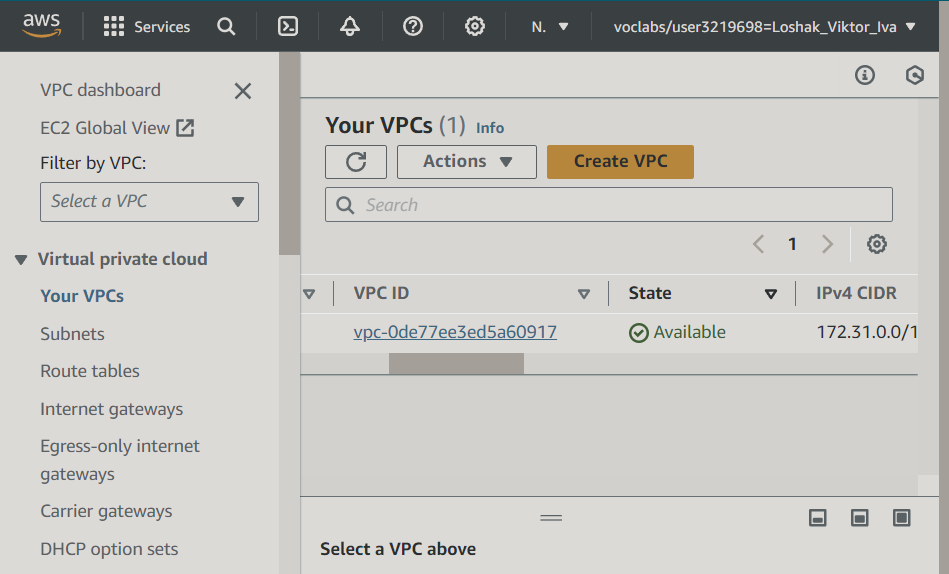


Це головна сторінка мого статичного веб-сайту, який розміщено на S3. Я ввів своє ім'я, університет, факультет і групу. Створення цього веб-сайту є частиною лабораторної роботи, яку я виконую у рамках курсу AWS.

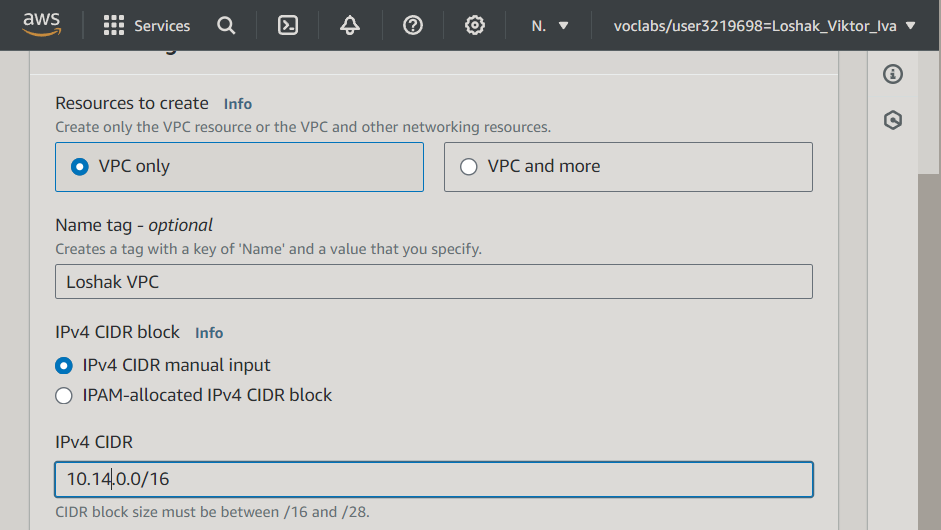
1. Module 4: Building the Amazon Virtual Private Cloud

**Завдання:** У цій лабораторній роботі ви створите базову віртуальну приватну хмару (VPC) без використання Майстер VPC. VPC, який ви створюєте, включатиме веб-сервер і базу даних Amazon RDS. Створивши обидва, ви підключите свою програму адресної книги, що працює на вашому веб-сервері, до свого екземпляра Amazon RDS для MySQL. Після успішного налаштування програми адресної книги за допомогою примірника RDS ви зможете додавати та видаляти контакти з адресної книги.

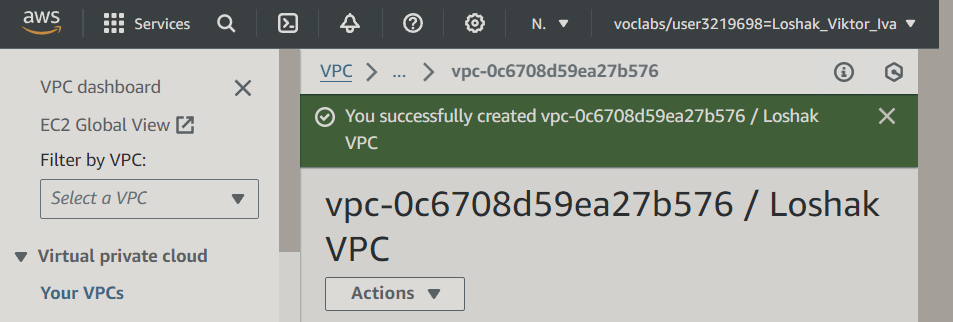
* 1. Create a VPC



На цьому екрані я переглядаю панель керування VPC на Amazon AWS. Я можу бачити, що вже маю один VPC доступний, але зараз моя задача – створити новий віртуальний приватний хмарний ресурс.

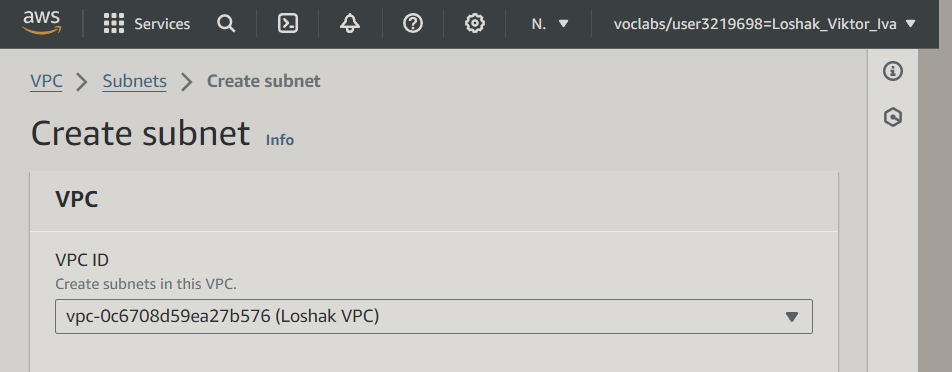


Тут я налаштовую параметри для нового VPC. Я вибрав "VPC only", щоб створити лише ресурс VPC, а також дав йому назву "Loshak VPC". Також я вказав власний блок CIDR для IPv4: 10.14.0.0/16. Це дозволяє мені визначити простір IP адрес, який буде використовуватися всередині VPC.



На цьому екрані відображається повідомлення про успішне створення нового VPC з назвою "Loshak VPC". Тепер я маю новий VPC, готовий для налаштування субмереж та інших налаштувань безпеки.

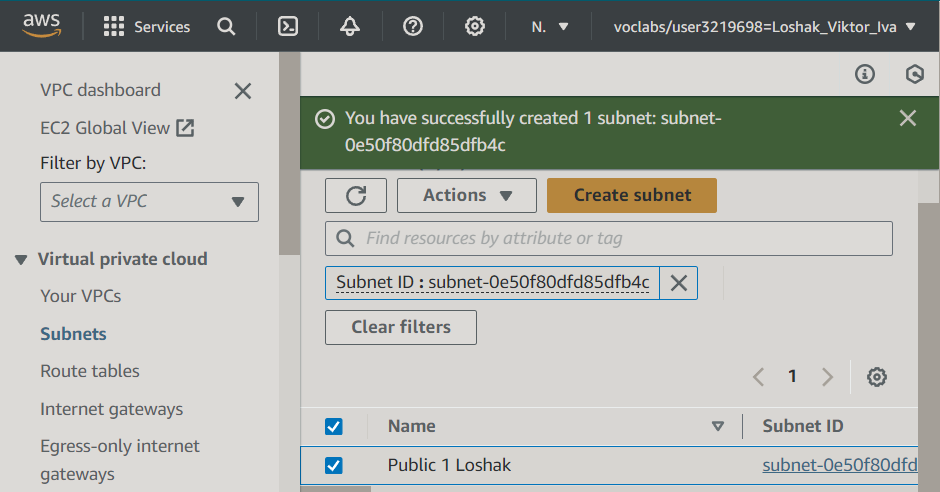
* 1. Create Your Public Subnets



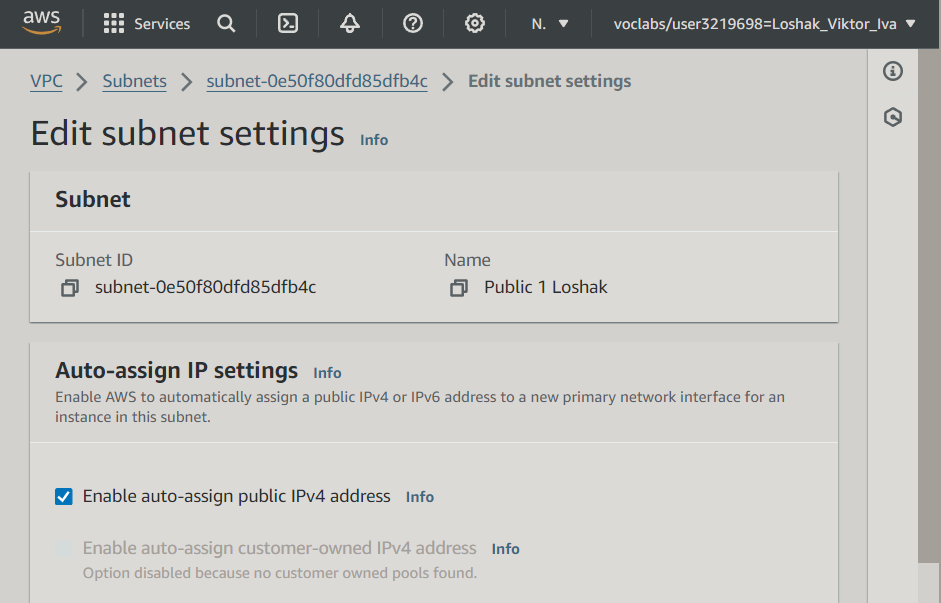
Тут я розпочав процес створення нової субмережі у вже створеному VPC. Я обрав "Loshak VPC" із списку, щоб визначити, де саме має бути розміщена ця субмережа.



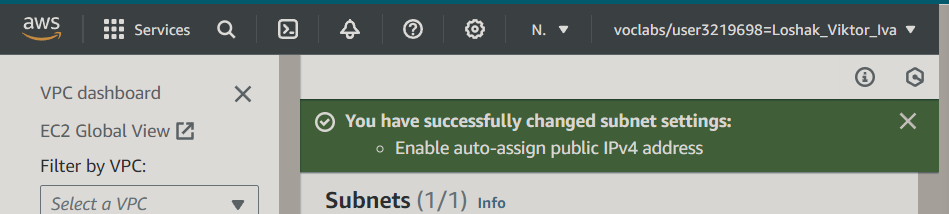
На цьому екрані я встановлюю конкретні параметри субмережі. Я назвав її "Public 1 Loshak" і вибрав зону доступності "us-east-1a", що відноситься до регіону "US East (N. Virginia)". Також я вказав CIDR блок для субмережі: 10.14.1.0/24, що є частиною більшого CIDR блоку VPC.



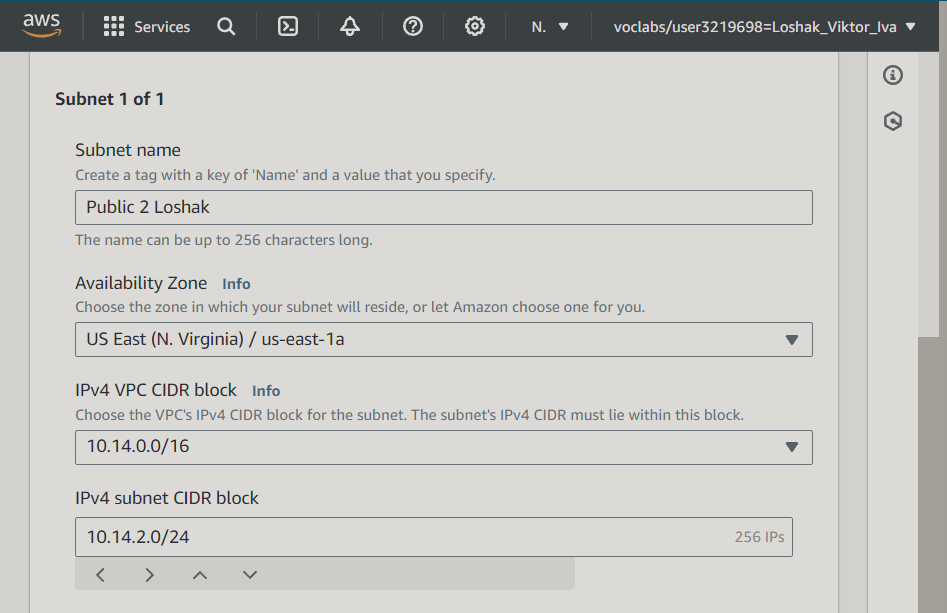
Тут відображається повідомлення про успішне створення субмережі "Public 1 Loshak" з ID subnet-0e50f80dfd85dfb4c. Це означає, що тепер у моєму VPC є субмережа для публічного використання.



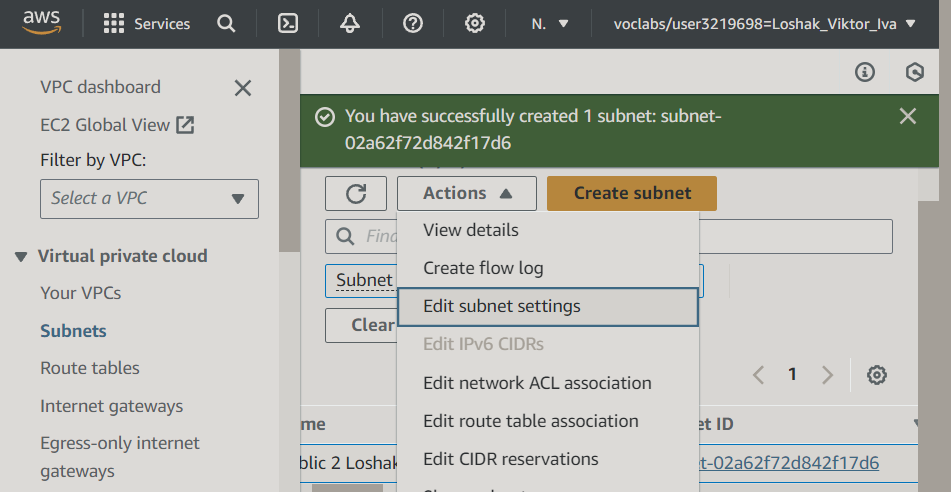
На цьому етапі я редагую налаштування субмережі, зокрема включаю опцію "Enable auto-assign public IPv4 address". Це дозволить AWS автоматично призначати публічні IPv4 адреси новим instances у цій субмережі.



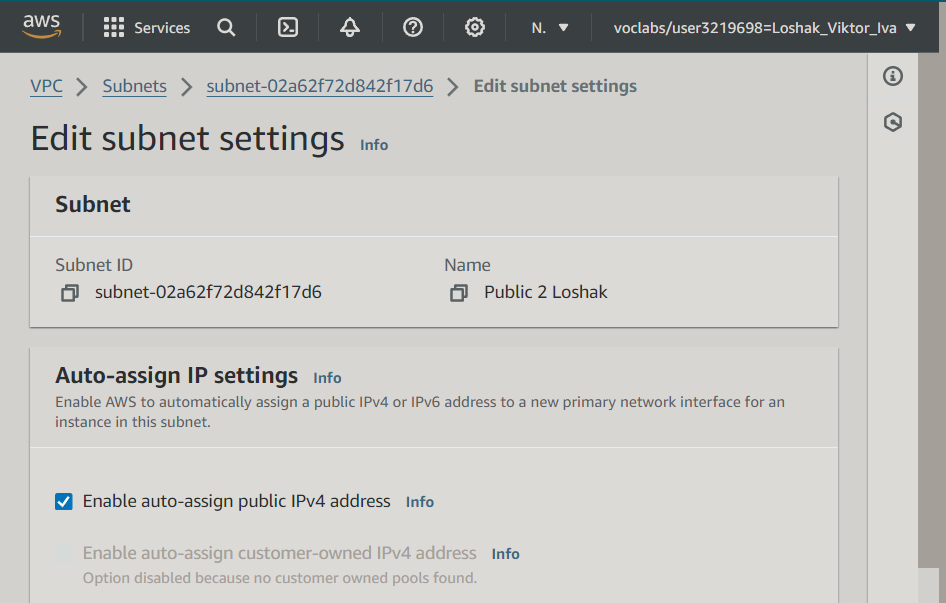
І останній крок – підтвердження зміни налаштувань субмережі. AWS повідомляє мене, що налаштування "Enable auto-assign public IPv4 address" було успішно включено для субмережі "Public 1 Loshak".



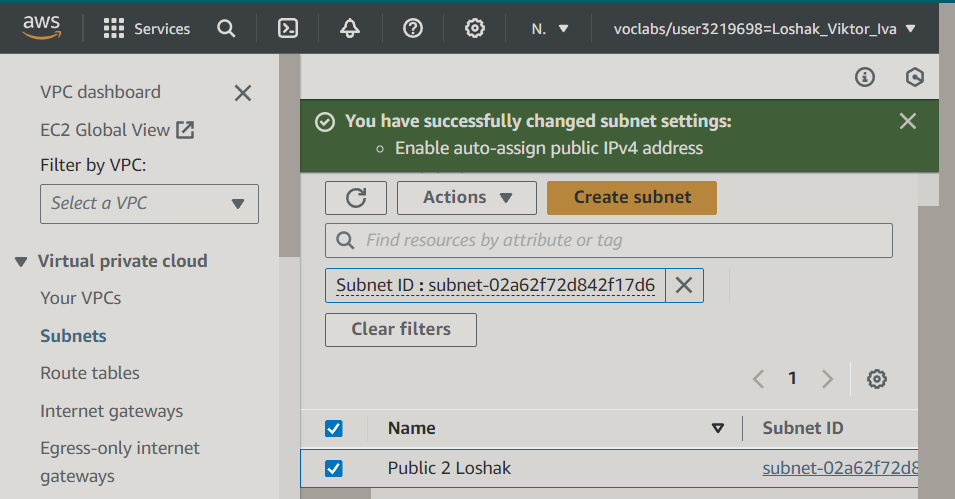
Я створюю другу субмережу у моєму VPC, що назвав "Public 2 Loshak". Я вибираю ту ж зону доступності, що й для першої субмережі – "us-east-1a", але з іншим CIDR блоком – 10.14.2.0/24, щоб розділити адресний простір між субмережами.



Тут відображено підтвердження успішного створення субмережі "Public 2 Loshak" з конкретним Subnet ID. Тепер у мене є дві субмережі для розміщення ресурсів.

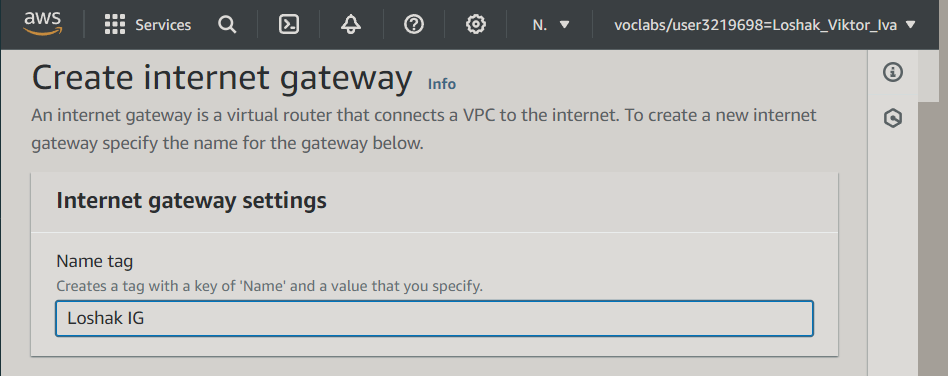


На цьому екрані я налаштовую параметри для нової субмережі "Public 2 Loshak", зокрема активую опцію "Enable auto-assign public IPv4 address", щоб автоматично надавати публічні IP-адреси новим instances.



Ось підтвердження, що налаштування субмережі "Public 2 Loshak" було успішно змінено, і функція автоматичного призначення публічних IPv4 адрес була включена.

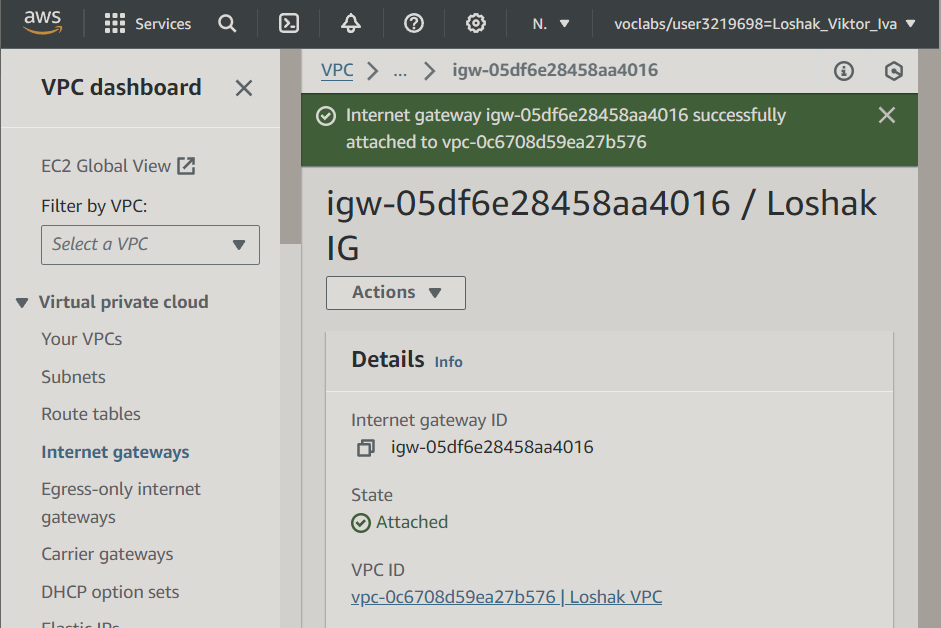
* 1. Create an Internet Gateway



Я створюю інтернет-шлюз для мого VPC, щоб забезпечити йому з'єднання з інтернетом. Я назвав цей шлюз "Loshak IG", що відображається в налаштуваннях інтернет-шлюзу.

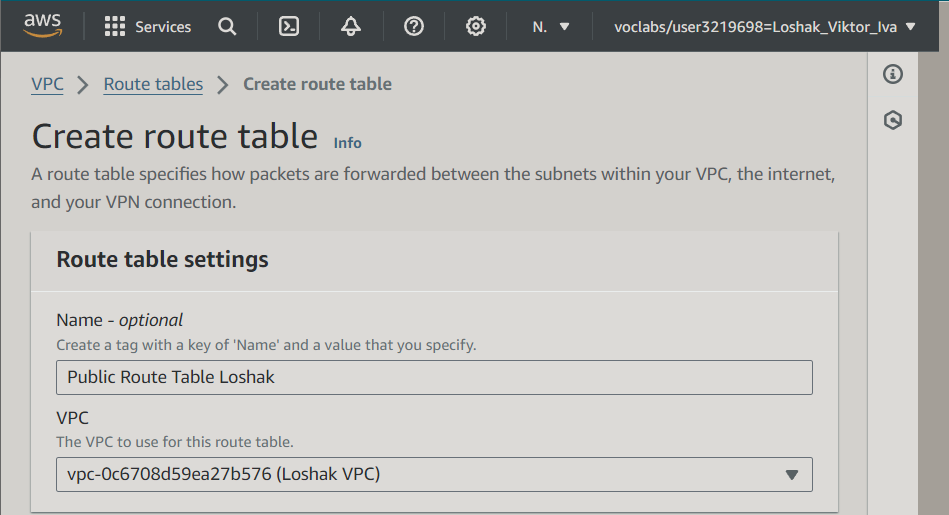


На цьому екрані я приєдную новостворений інтернет-шлюз до свого VPC, зазначеного як "vpc-0c6708d59ea27b576 / Loshak VPC". Цей крок дозволяє моєму VPC взаємодіяти з зовнішнім світом через інтернет.

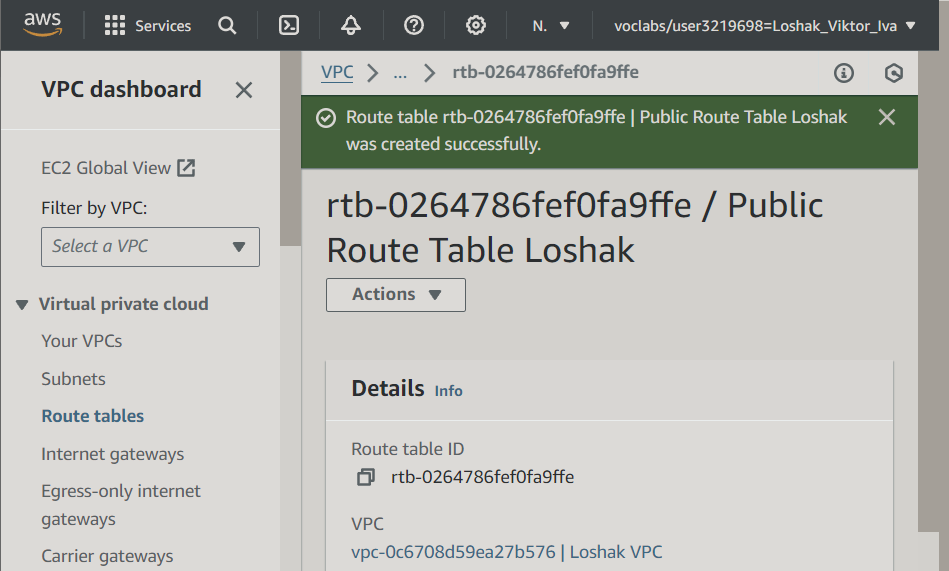


Тут відображено повідомлення про успішне приєднання інтернет-шлюзу до мого VPC. Це означає, що налаштування з'єднання з інтернетом для мого VPC завершено і воно готове до використання.

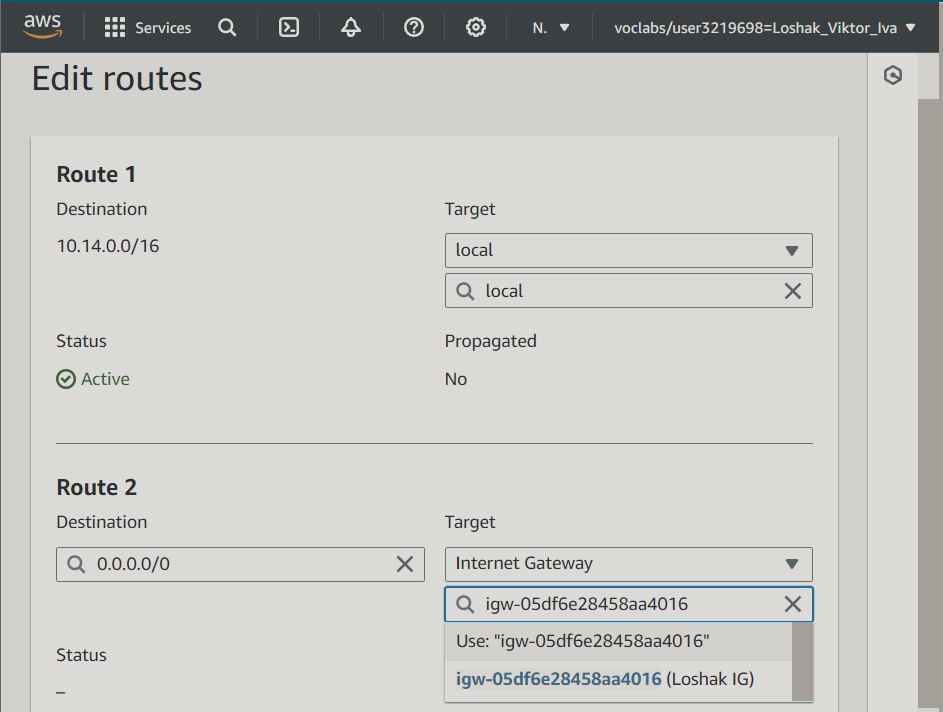
* 1. Create a Route Table, Add Routes, And Associate Public Subnets



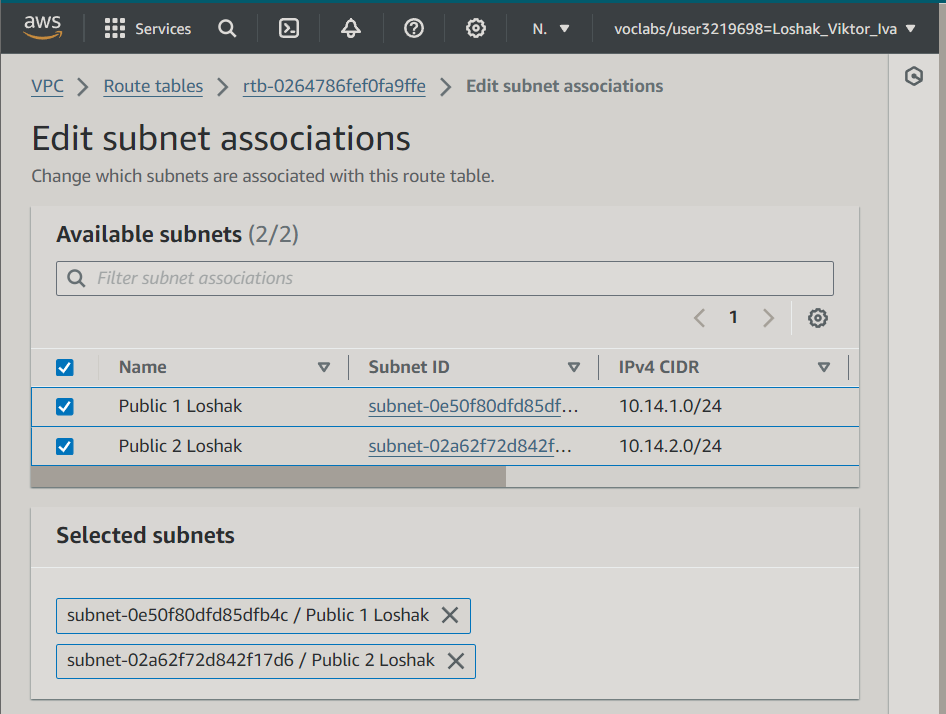
Я створюю таблицю маршрутизації для мого VPC, яку назвав "Public Route Table Loshak". Таблиця маршрутизації визначає, як пакети пересилаються між субмережами всередині VPC, до інтернету, а також до VPN з'єднань.



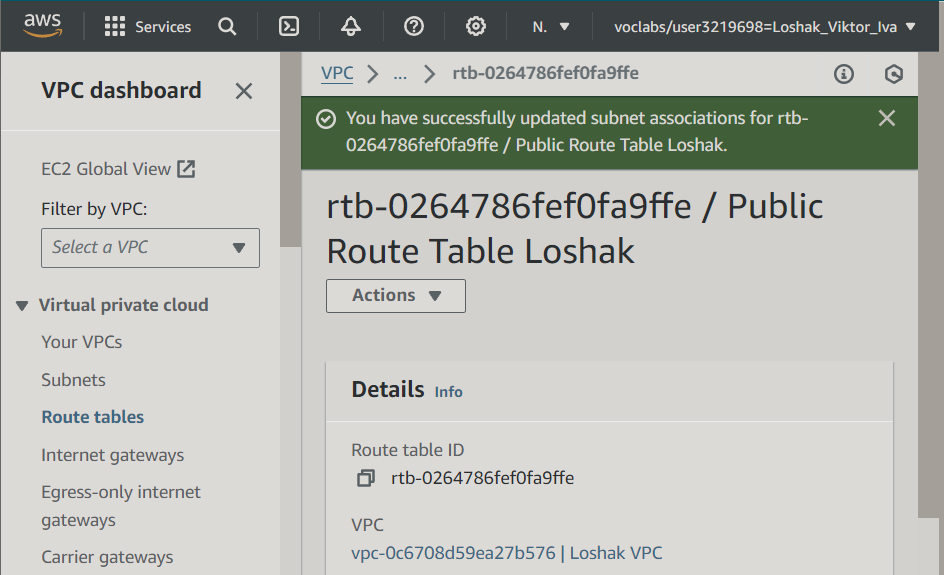
Цей скріншот підтверджує створення нової таблиці маршрутизації "Public Route Table Loshak" з ID rtb-0264786fef0fa9ffe. Таблиця готова до конфігурації маршрутів.



Тут я редагую маршрути для таблиці "Public Route Table Loshak". Я додаю маршрут для всіх IP (0.0.0.0/0) і націлюю його на інтернет-шлюз "igw-05df6e28458aa4016 (Loshak IG)", що забезпечує вихід до інтернету для мережі.

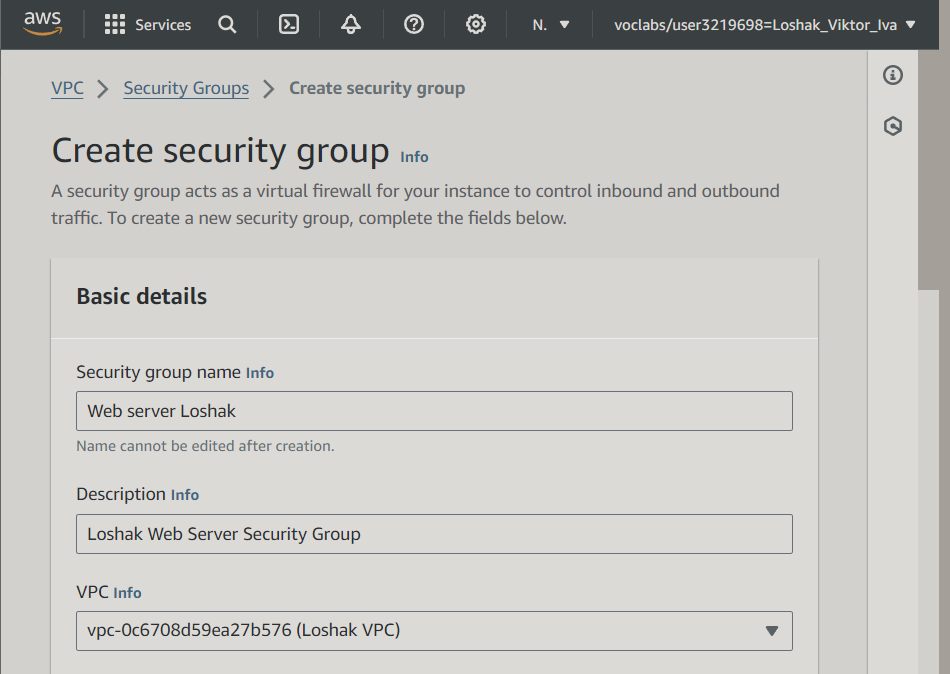


На цьому екрані я асоціюю обидві мої публічні субмережі "Public 1 Loshak" і "Public 2 Loshak" з новоствореною таблицею маршрутизації. Це забезпечує, що трафік з цих субмереж буде коректно маршрутизуватися.

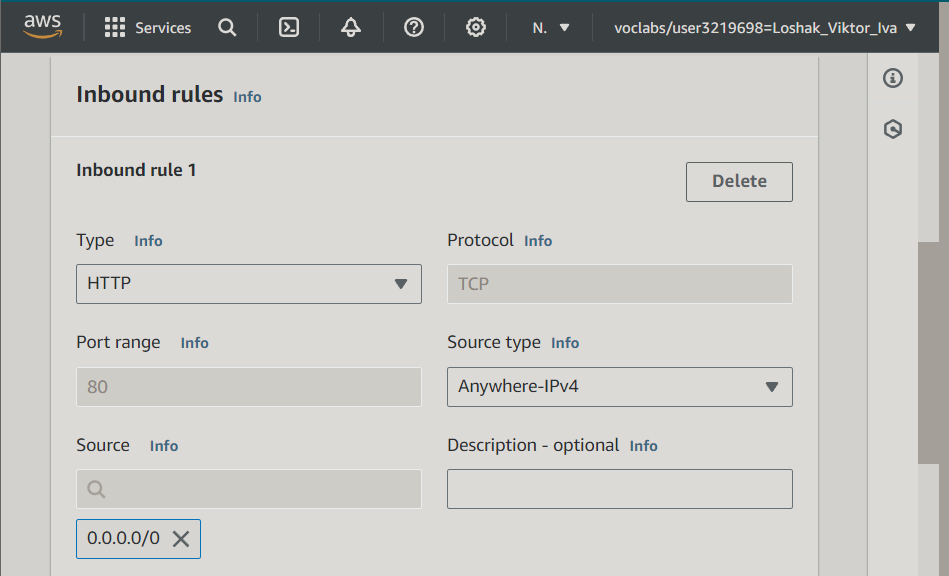


Тут відображається повідомлення про успішне оновлення асоціацій субмереж для таблиці маршрутизації "Public Route Table Loshak". Це завершує процес налаштування маршрутизації для мого VPC.

* 1. Create a Security Group for your Web Server



Я створюю групу безпеки для свого веб-сервера в AWS, називаючи її "Web server Loshak". Група безпеки діє як віртуальний брандмауер для керування вхідним і вихідним трафіком of instances, що входять до неї.

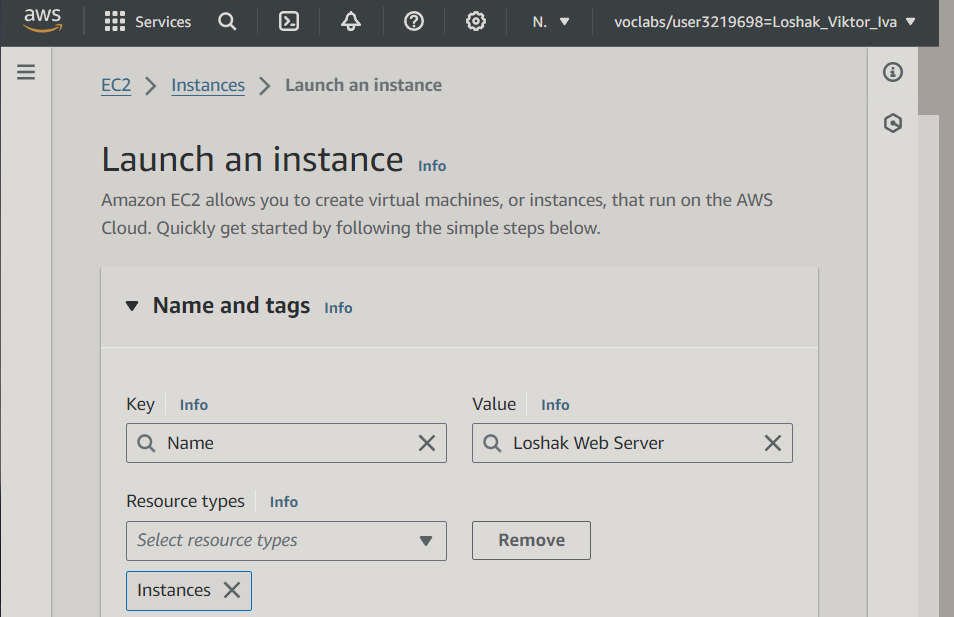


Тут я налаштовую правила вхідного трафіку для створеної групи безпеки. Я додаю правило для HTTP трафіку (протокол TCP на порту 80), дозволяючи доступ до веб-сервера з будь-якої точки Інтернету (0.0.0.0/0).



Цей скріншот підтверджує успішне створення групи безпеки "Web server Loshak". Тепер мій instance веб-сервера захищений правилами, які контролюють доступ до нього ззовні.

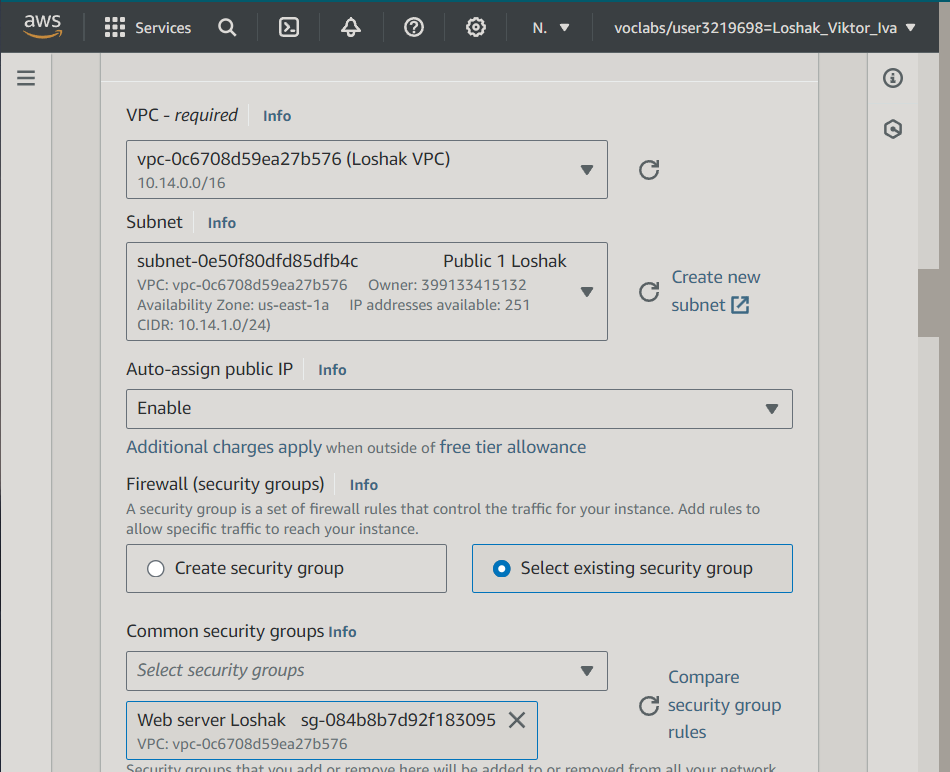
* 1. Launch a Web Server in your Public Subnet



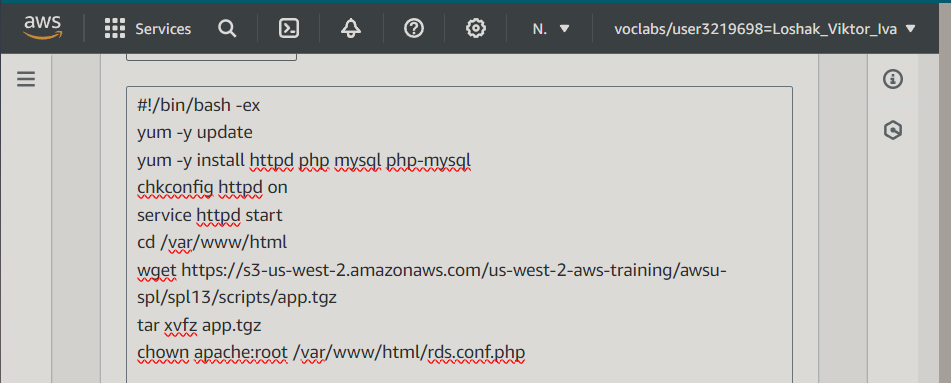
Я запускаю новий віртуальний сервер на AWS EC2. Назва цього сервера — "Loshak Web Server". Цей крок необхідний для розгортання веб-сервера, який буде обробляти веб-запити.



Тут я вибираю образ операційної системи для instance — Amazon Linux 2023, який є популярним вибором для веб-серверів завдяки своїй стабільності та підтримці з боку Amazon.



На цьому екрані я вибираю VPC та субмережу для мого сервера, зокрема "Public 1 Loshak", що забезпечує йому публічний доступ до інтернету. Також я вказую використати раніше створену групу безпеки "Web server Loshak".



Цей скріншот показує частину скрипта, який я використовую для автоматичного налаштування веб-сервера на instance. Скрипт включає команди для оновлення системи, встановлення веб-сервера Apache та необхідних компонентів PHP та MySQL.

**Текст скрипта в повному вигляді:**

#!/bin/bash -ex

yum -y update

yum -y install httpd php mysql php-mysql

chkconfig httpd on

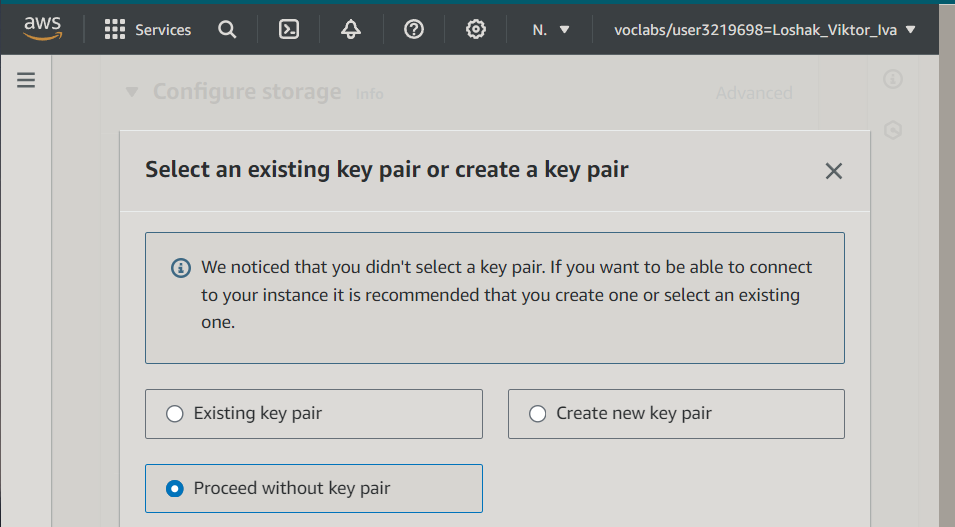
service httpd start

cd /var/www/html

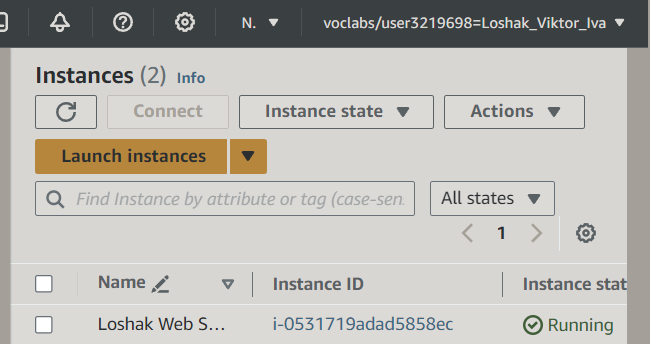
wget https://s3-us-west-2.amazonaws.com/us-west-2-aws-training/awsu-spl/spl-13/scripts/app.tgz

tar xvfz app.tgz

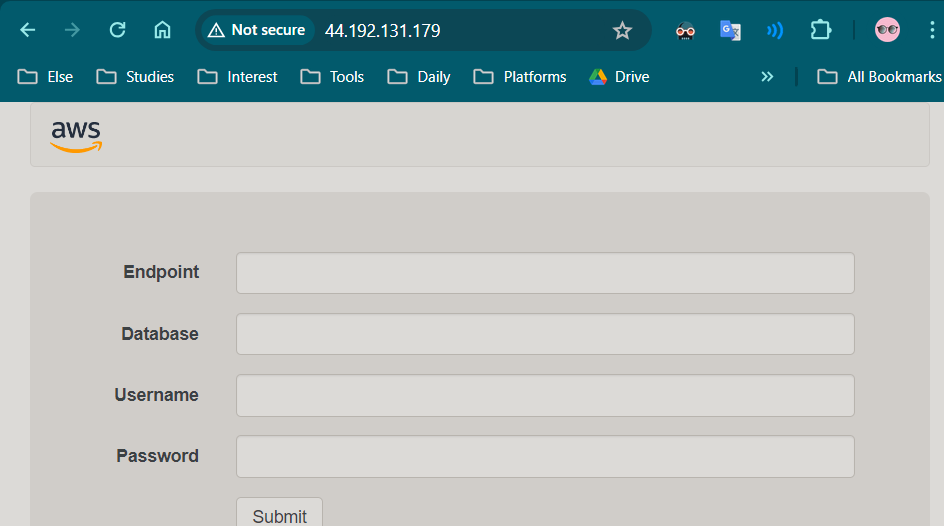
chown apache:root /var/www/html/rds.conf.php



Тут я отримав попередження про необхідність вибору ключа для доступу до instance. Я вирішив продовжити без ключа, що означає, що до сервера не можна буде підключитися через SSH.

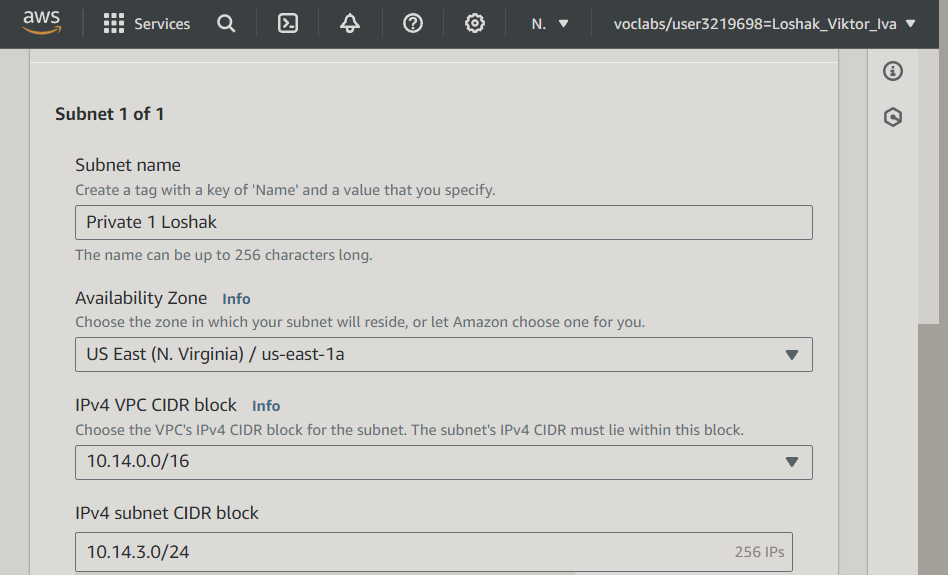


Цей скріншот підтверджує, що instance "Loshak Web Server" була успішно запущена і вже працює.

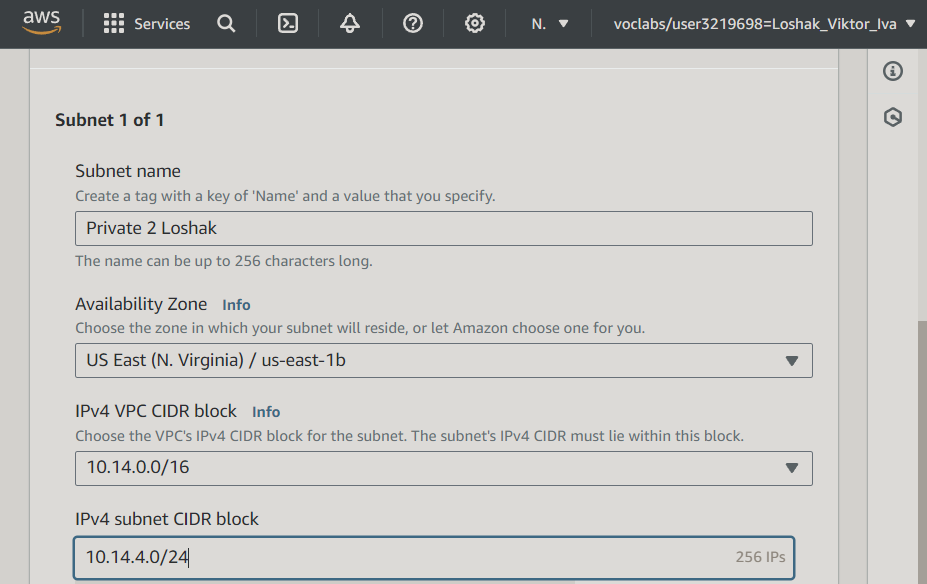


Інтерфейс розгорнений на веб сервері запитує дані для з'єднання з базою даних, такі як endpoint, назва бази даних, ім'я користувача та пароль. Ця форма дозволить користувачам встановлювати з'єднання з базою даних через веб-сторінкy.

* 1. Create Private Subnets for your MySQL Server

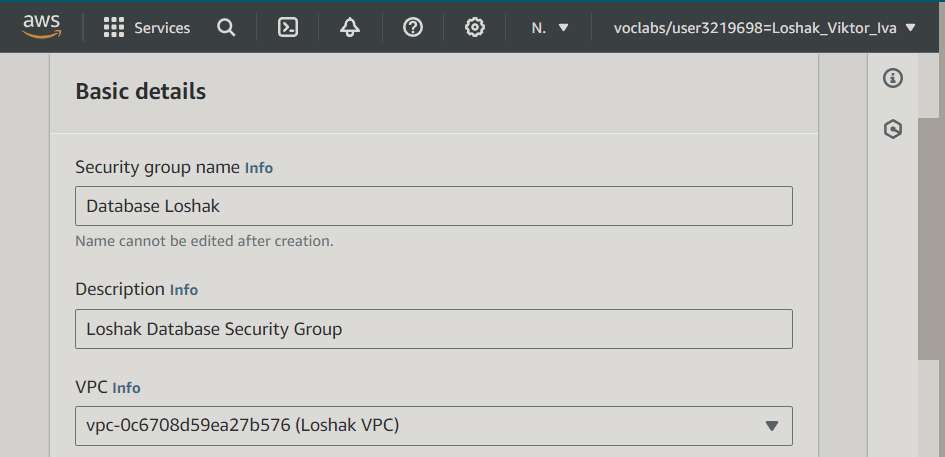


Я створюю нову приватну субмережу в моєму VPC на AWS, яку я назвав "Private 1 Loshak". Це субмережа в зоні доступності us-east-1a. Я обрав блок IP-адрес 10.14.3.0/24 з загального блоку VPC 10.14.0.0/16. Ця субмережа призначена для внутрішніх потреб моєї інфраструктури, які не потребують прямого доступу з інтернету.

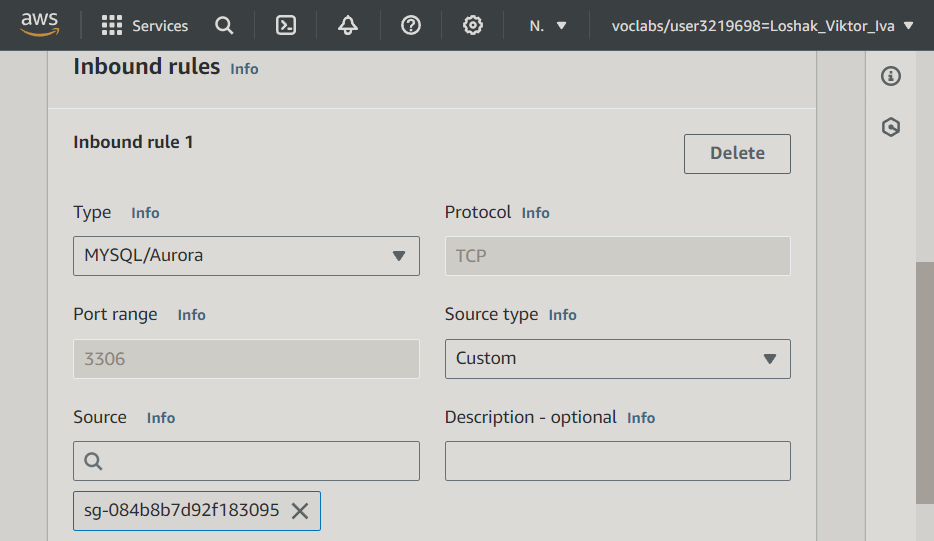


Тут я створюю ще одну приватну субмережу, "Private 2 Loshak", в іншій зоні доступності us-east-1b для забезпечення високої доступності. Ця субмережа має блок IP-адрес 10.14.4.0/24. Ці налаштування дозволяють мені організувати балансування навантаження та забезпечити надійність моїх сервісів всередині AWS.

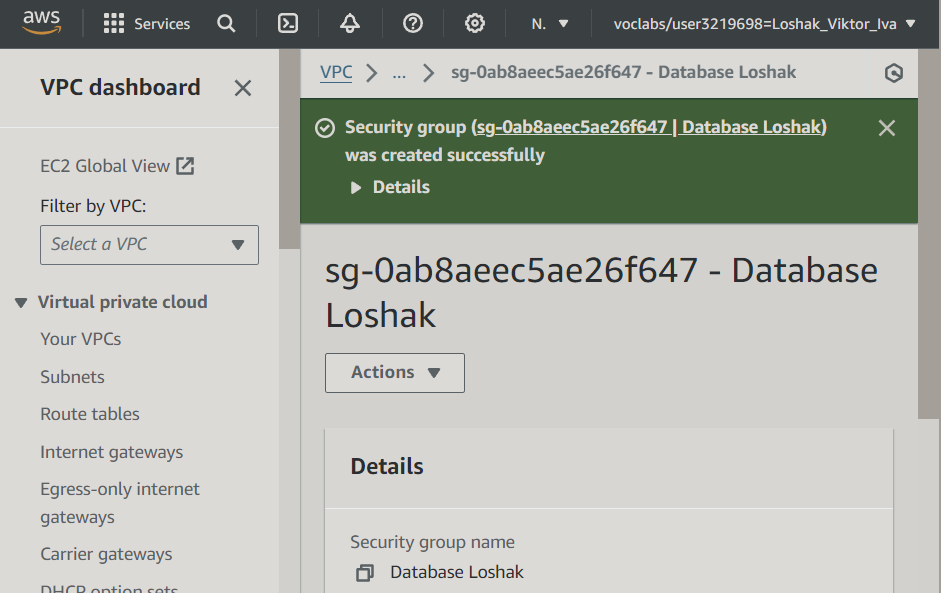
* 1. Create a Security Group for your Database Server



Я створюю нову групу безпеки для бази даних в моєму VPC на AWS, яку називаю "Database Loshak". Ця група безпеки буде керувати доступом до бази даних, визначаючи, які запити можуть до неї надходити.



На цьому екрані я налаштовую правило для групи безпеки, яке дозволяє трафік MySQL на порту 3306. Я обмежую доступ лише з моєї веб-серверної групи безпеки, що забезпечує високий рівень захисту та ізоляцію бази даних від неавторизованого доступу.

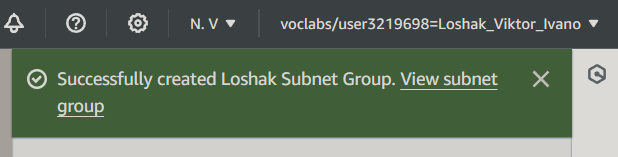


Цей скріншот підтверджує успішне створення групи безпеки "Database Loshak" з ID sg-0ab8aeec5ae26f647. Тепер база даних захищена визначеними правилами, що значно знижує ризик несанкціонованих доступів.

* 1. Create a Database Subnet Group

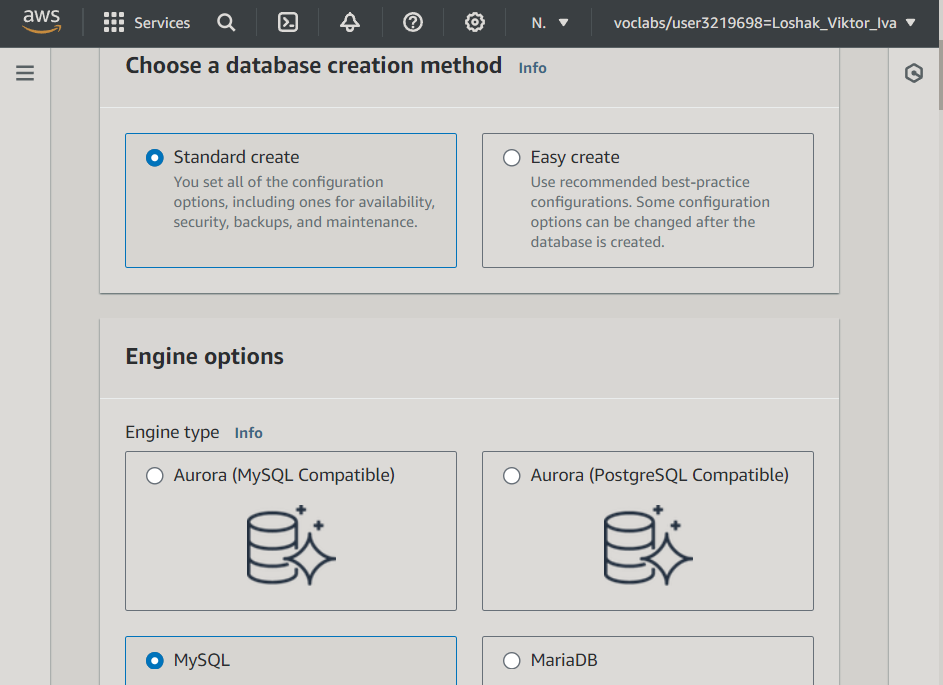


Я створюю групу субмереж під назвою "Loshak Subnet Group" у моєму VPC (Loshak VPC). Ця група включає субмережі в двох зонах доступності, us-east-1a та us-east-1b, що забезпечує вищу доступність та стійкість сервісів баз даних, які будуть розміщені у цих субмережах.

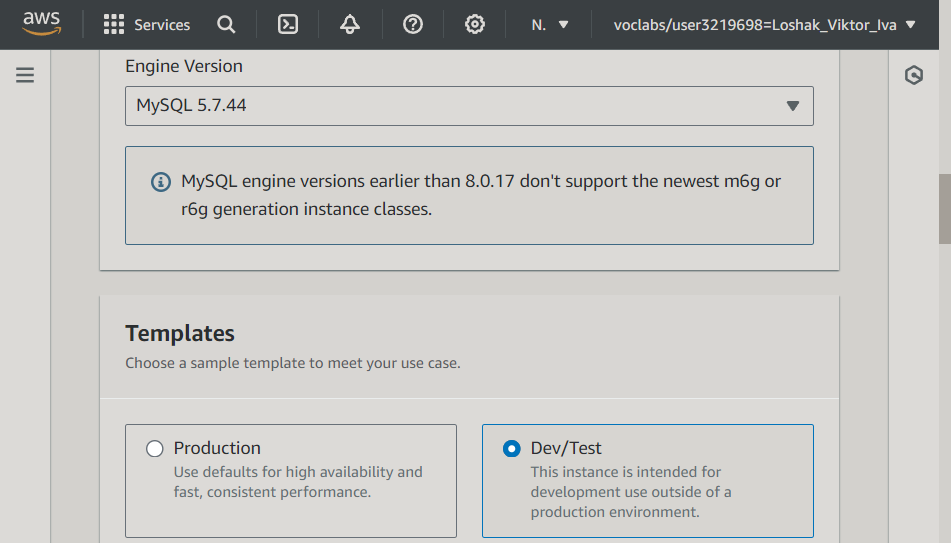


Цей скріншот підтверджує успішне створення "Loshak Subnet Group". Тепер я можу використовувати цю групу для розгортання баз даних в обраному VPC, забезпечуючи, що бази даних доступні з різних локацій для забезпечення відмовостійкості.

* 1. Create an Amazon RDS Database



На цьому скріншоті я обираю метод створення бази даних, вибираючи стандартну конфігурацію, яка дозволяє мені налаштувати всі параметри, включаючи налаштування доступності, безпеки, резервного копіювання та обслуговування.



Я вибрав MySQL як двигун бази даних і вказав версію MySQL 5.7.44 для моєї інстанції. Я обрав шаблон "Dev/Test" для інстанції, що підходить для розробки та тестування поза виробничим середовищем.

A screenshot of a computer

Description automatically generated

На цьому етапі я ввів ідентифікатор інстанції бази даних як "Loshak DB".

A screenshot of a computer

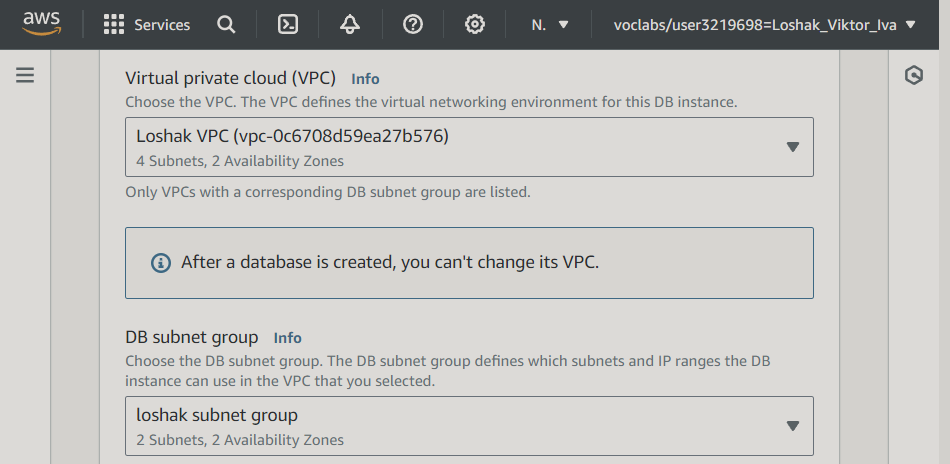
Description automatically generated

Я обрав тип інстанції db.t3.small з 2 віртуальними ЦПУ та 2 ГБ оперативної пам'яті, що підходить для моїх тестових цілей.

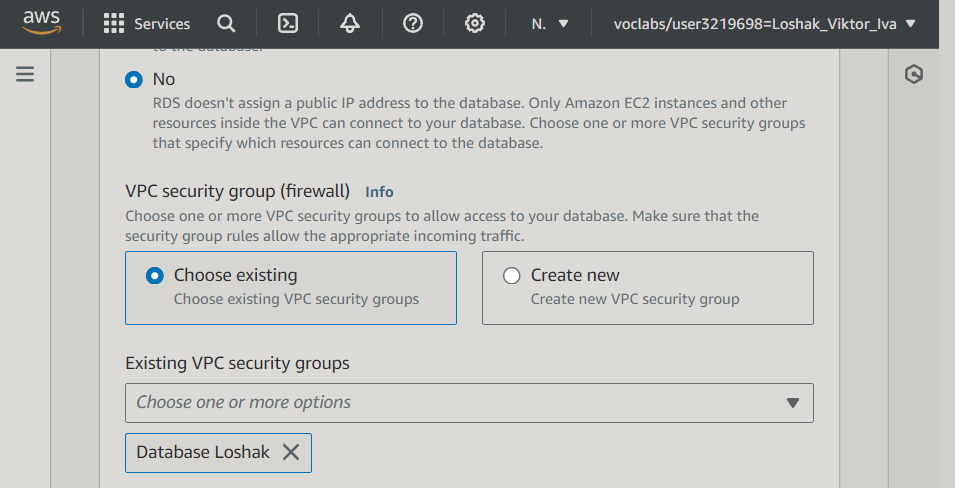
A screenshot of a computer

Description automatically generated

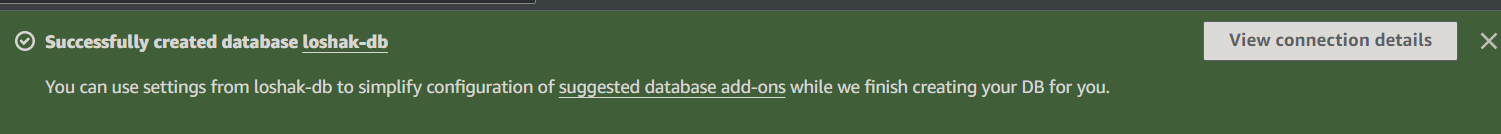
Я вибрав тип зберігання General Purpose SSD (gp2) з виділеним розміром в 100 ГБ.



Тут я вказав віртуальну власну хмару (VPC) для інстанції бази даних та вибрав раніше створену групу підмереж "loshak subnet group".

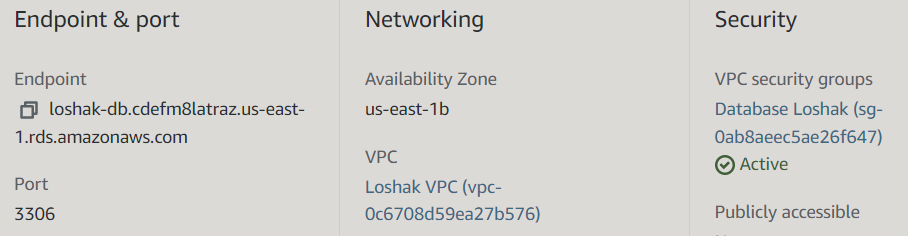


Я обрав існуючу групу безпеки "Database Loshak" для нової бази даних, щоб забезпечити налаштування файрволу.

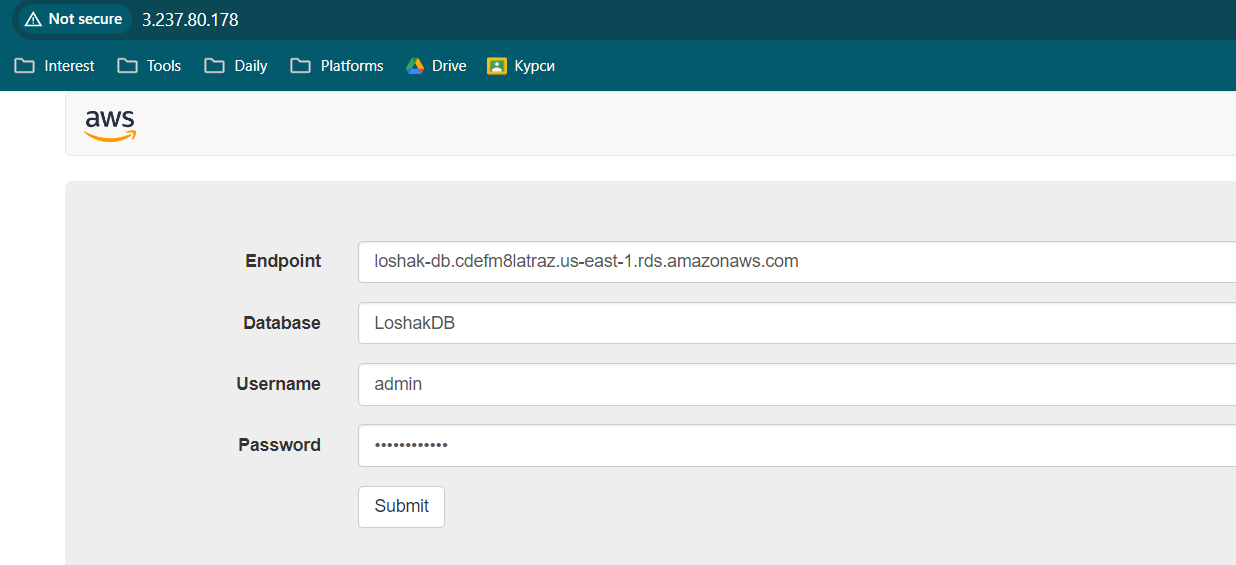


Завершення створення бази даних, де система показує повідомлення про успішне створення інстанції "loshak-db".

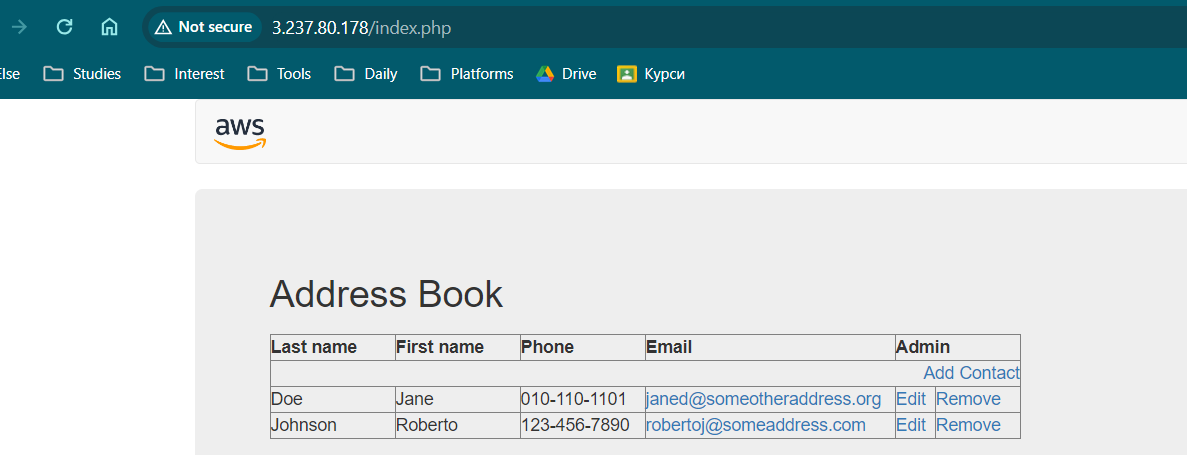
* 1. Connect Your Address Book Application to Your Database



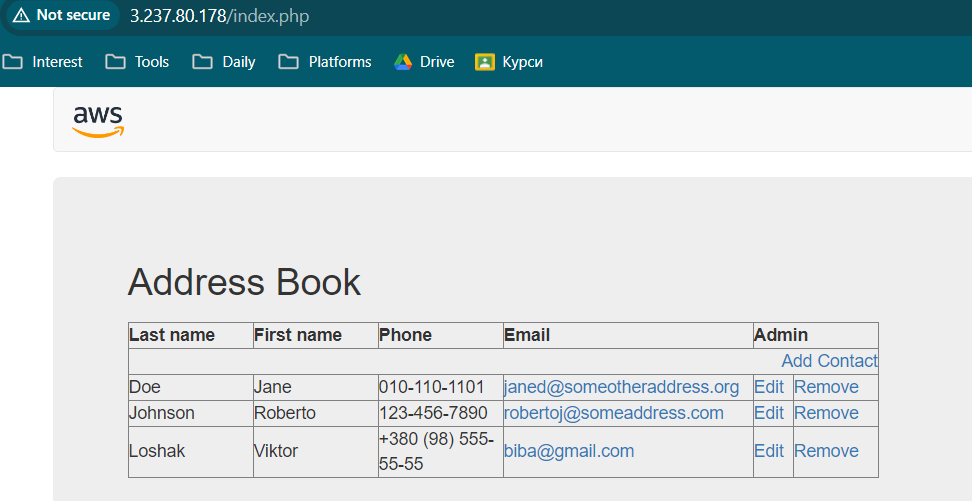
На цьому скріншоті я переглядаю інформацію про базу даних, яку я створив на AWS. Тут видно, що ендпоінт (адреса для підключення) бази даних називається "loshak-db.cdefm8latraz.us-east-1.rds.amazonaws.com" і вона доступна через порт 3306.



Тут я ввів дані для доступу до бази даних через веб-інтерфейс. У поля введення я вписав ендпоінт бази даних, її назву "LoshakDB", ім'я користувача "admin", та пароль. Це необхідно для того, щоб підключитися до бази даних і продовжити роботу з нею.



На цьому екрані зображено веб-інтерфейс "Address Book", де можна бачити список зареєстрованих контактів. Це показує, що я успішно підключився до бази даних і можу переглядати інформацію. Ви можете бачити базові дані про кожен контакт, такі як прізвище, ім'я, телефон, і електронну адресу.

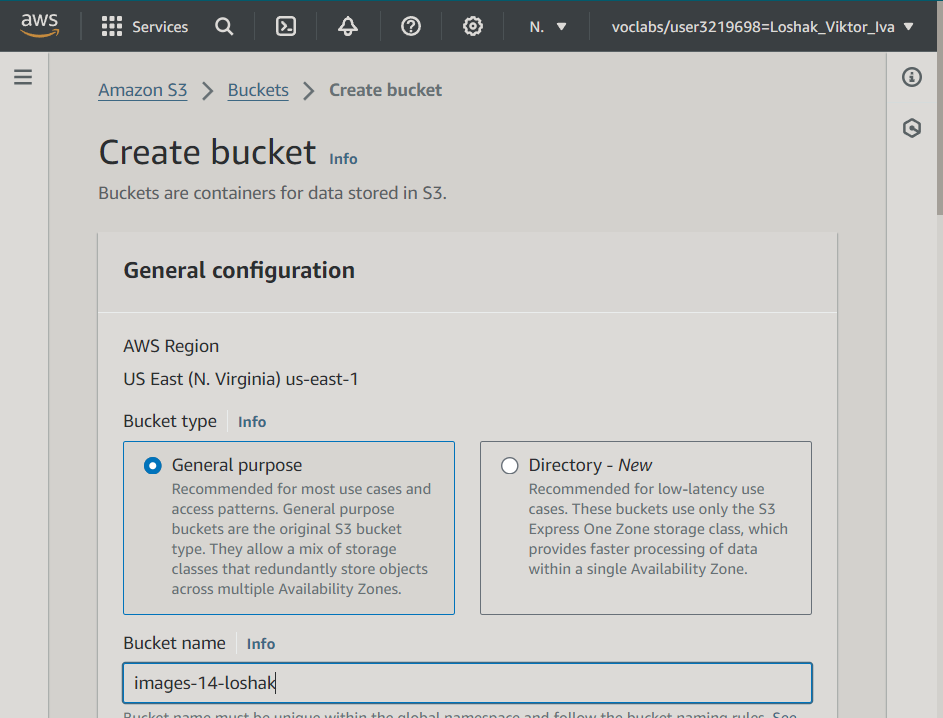


Тут я додав новий контакт "Viktor Loshak" з телефоном "+380 (98) 555-55" та електронною адресою "biba @gmail.com" у веб-інтерфейс "Address Book". Цей крок показує, як можна використовувати інтерфейс для управління даними в базі даних розташованій в AWS через веб.

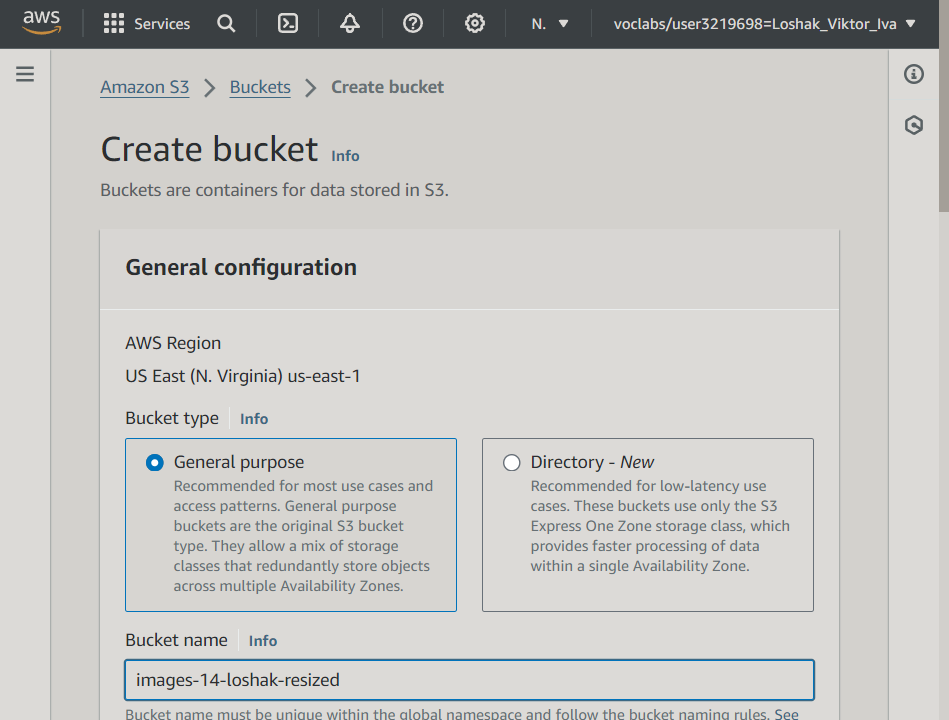
1. Module 5: Introduction to AWS Lambda

Лабораторія надає базове пояснення AWS Lambda. Вона продемонструє кроки, необхідні для початку створення функції Lambda в event-driven середовищі.

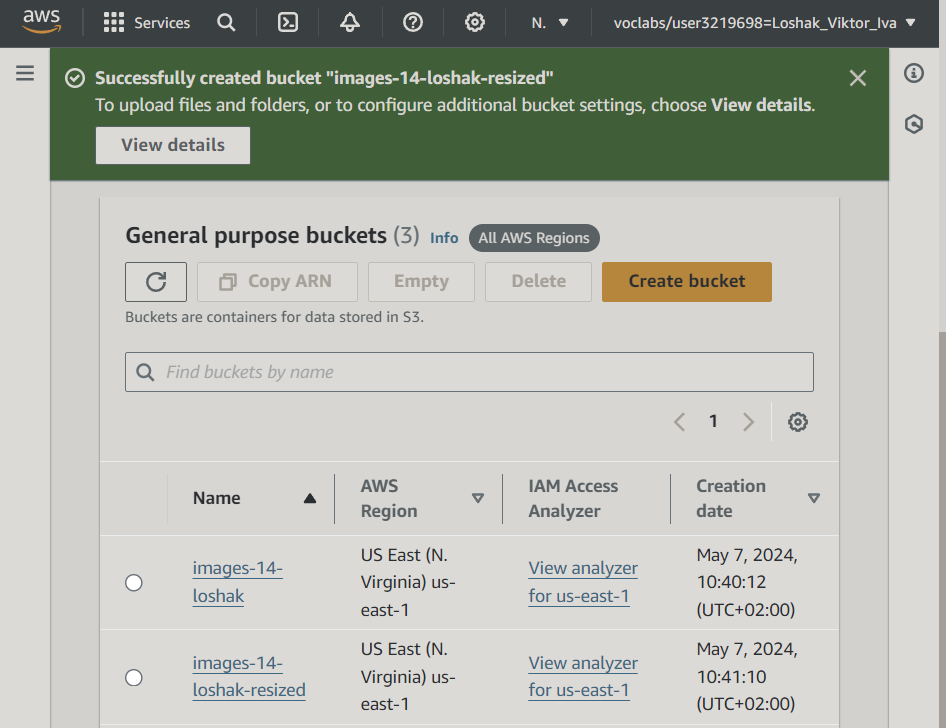
* 1. Create the Amazon S3 Buckets



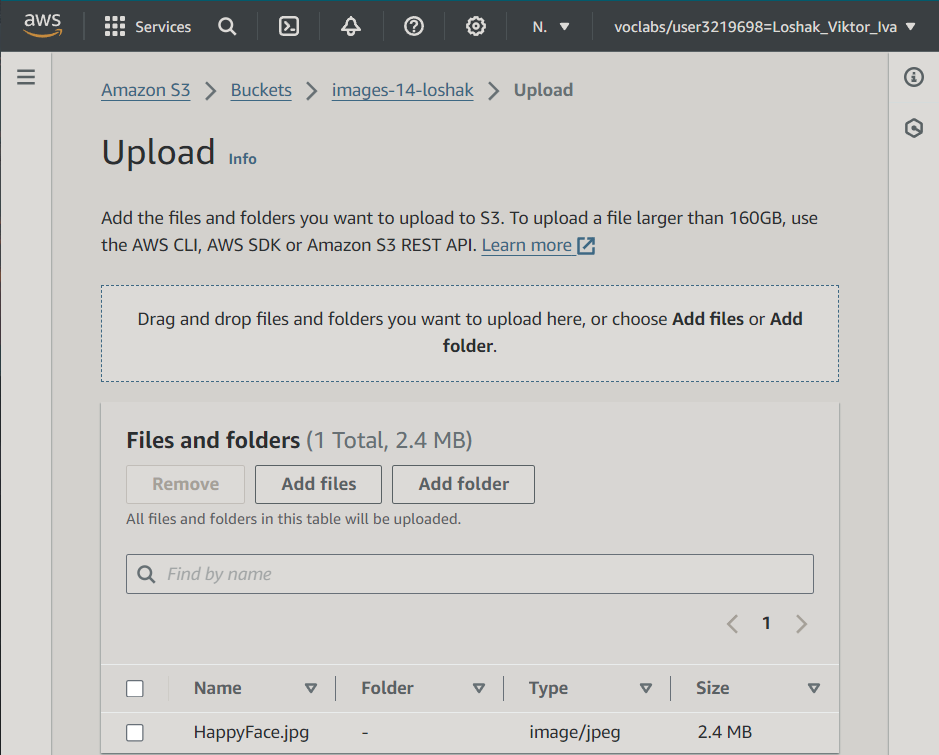
Я створюю новий bucket у Amazon S3 для зберігання даних. На екрані вказано регіон "US East (N. Virginia) us-east-1" і обрано тип bucket "General purpose". Також я вводжу назву bucket "images-14-loshak". Цей крок необхідний для організації зберігання різноманітних файлів у хмарі.



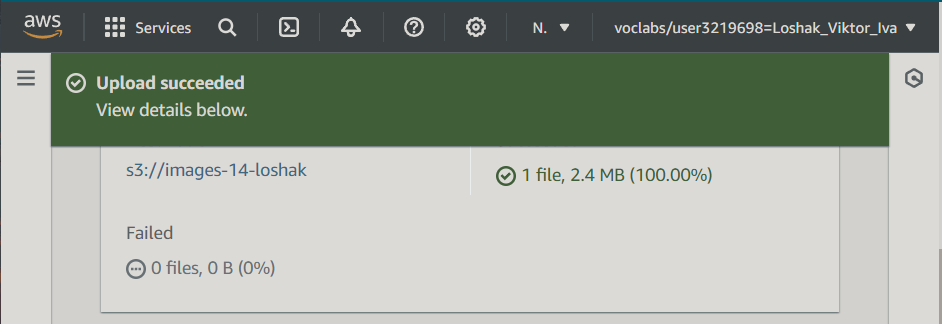
Тут я створюю другий bucket "images-14-loshak-resized" який буде використовуватися для зберігання зменшених зображень.



Цей скріншот показує успішне створення bucket "images-14-loshak" та "images-14-loshak-resized" в Amazon S3. Такий результат засвідчує, що я правильно налаштував параметри і bucket готовий до використання.

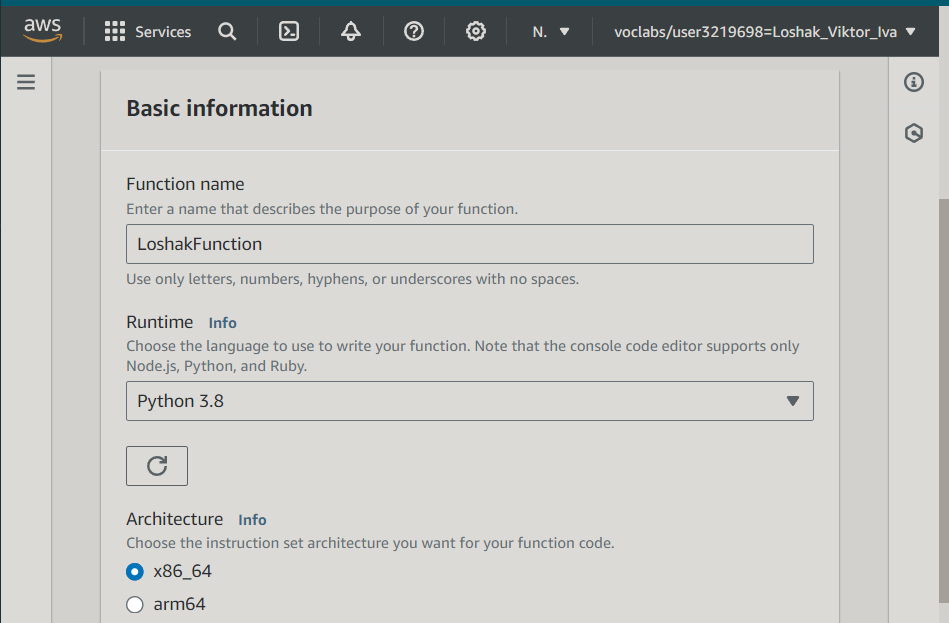


На цьому екрані я виконую завантаження файлу "HappyFace.jpg" розміром 2.4 MB у вже створений bucket. Видно інтерфейс, який дозволяє додати файли та папки для завантаження.

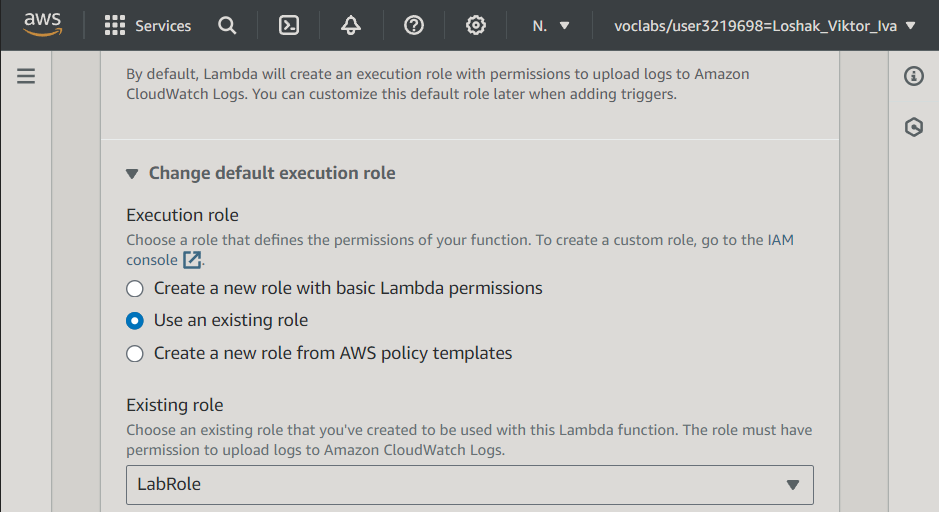


Тут підтверджується успішне завантаження файлу "HappyFace.jpg" у bucket "images-14-loshak". Це показує, що файл був коректно переданий і збережений у хмарі AWS.

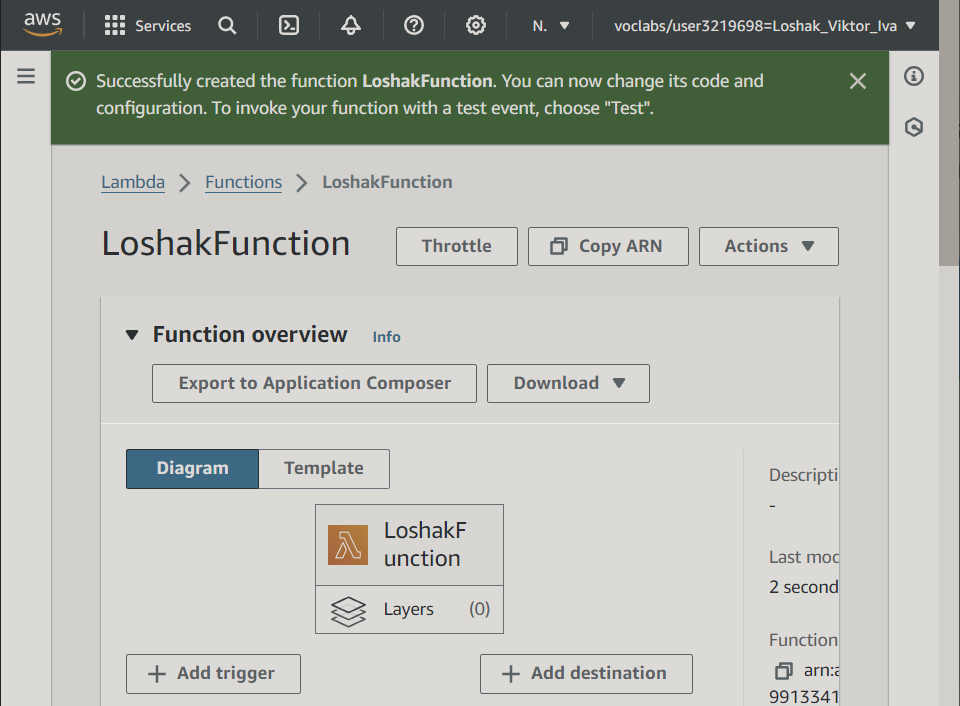
* 1. Create an AWS Lambda Function



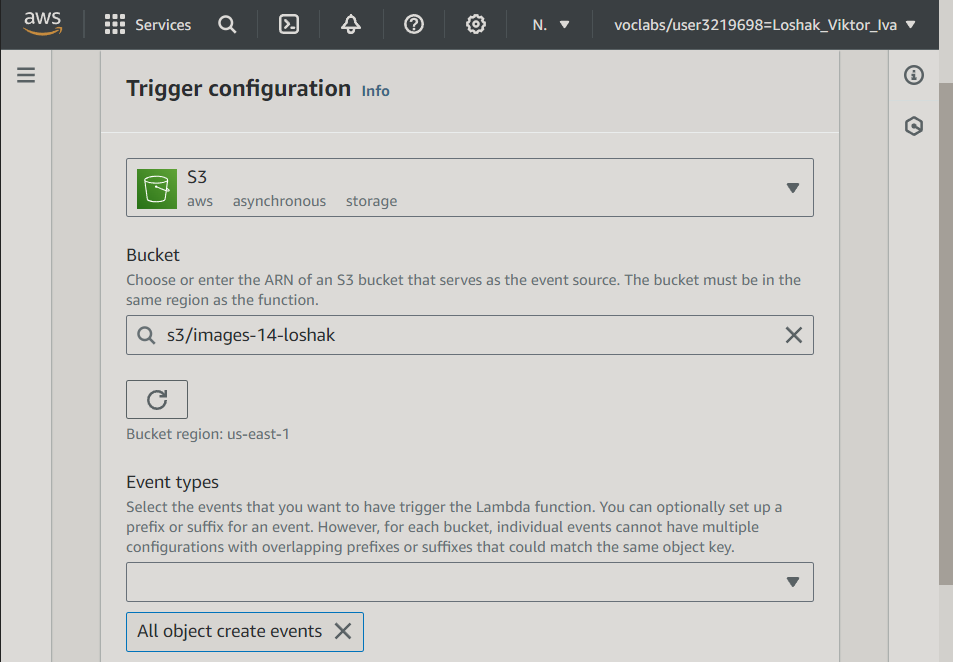
Я створюю нову функцію в AWS Lambda. Назва функції "LoshakFunction", обрана мова програмування – Python 3.8, архітектура – x86\_64. Цей етап важливий для налаштування середовища, в якому буде виконуватися моя функція.



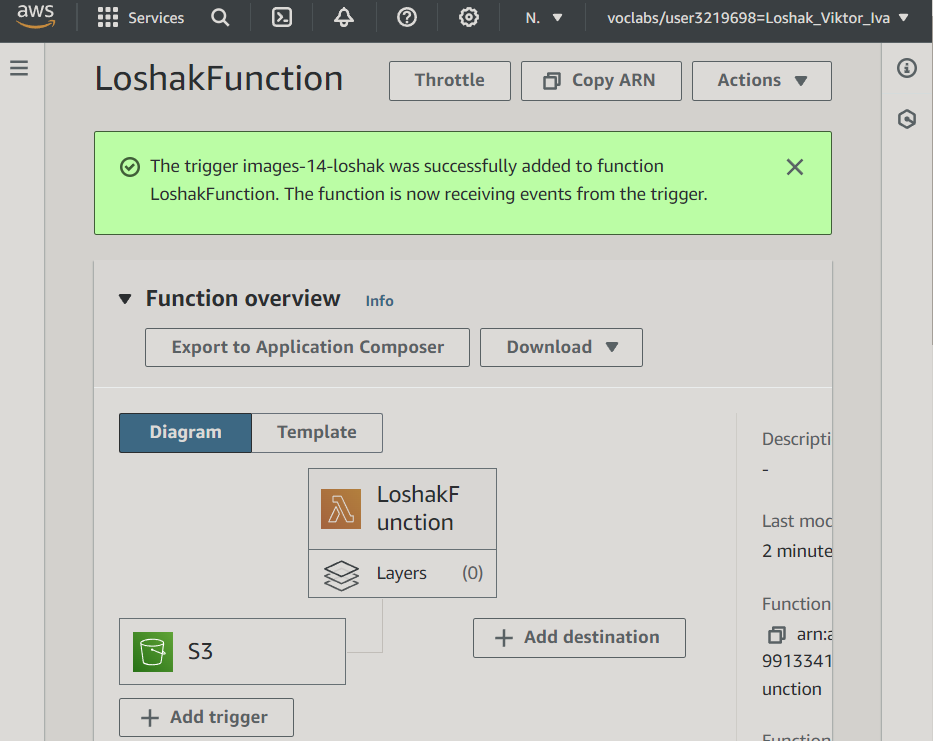
Тут я вибираю існуючу роль виконання "LabRole" для моєї Lambda функції. Ця роль визначає дозволи, які будуть надані функції під час її виконання.



На цьому скріншоті показано успішне створення функції "LoshakFunction". Зараз я можу змінювати її код і конфігурацію та викликати функцію з тестовими подіями.



Я налаштовую тригер для Lambda функції, використовуючи як джерело подій S3 bucket "images-14-loshak". Тригер реагуватиме на всі події створення об'єктів у цьому bucket.



Тригер "images-14-loshak" успішно додано до функції "LoshakFunction". Тепер функція буде автоматично викликатися при додаванні нових файлів у вказаний bucket.

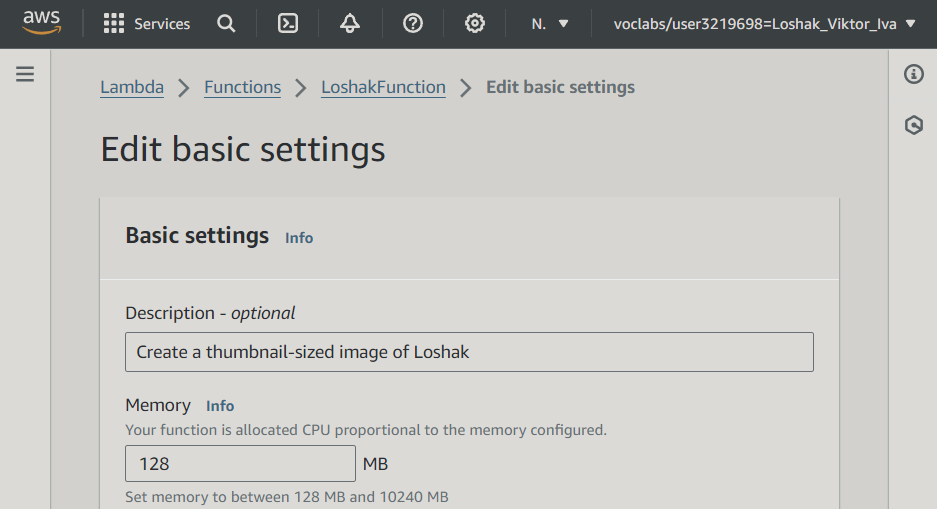
A screenshot of a computer

Description automatically generated

На цьому етапі я завантажую код функції у вигляді .zip файлу "CreateThumbnail.zip". Це дозволить оновити код функції для обробки зображень.

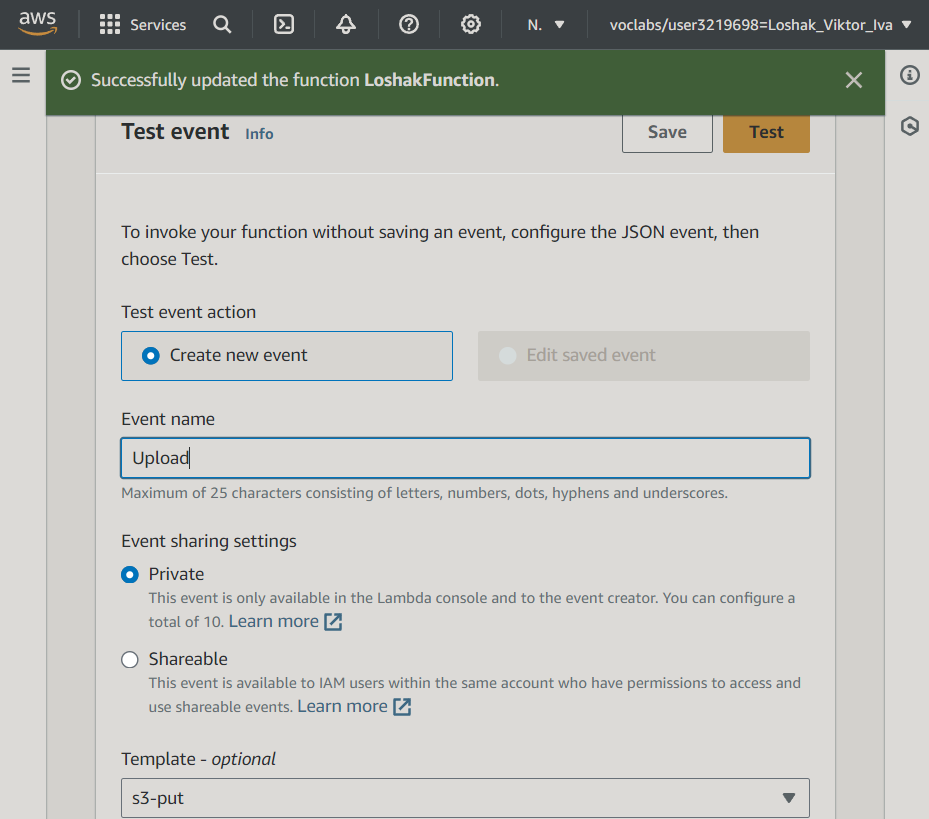


Завантаження файлу "CreateThumbnail.zip" успішно завершено. Функція "LoshakFunction" тепер містить актуальний код для виконання.

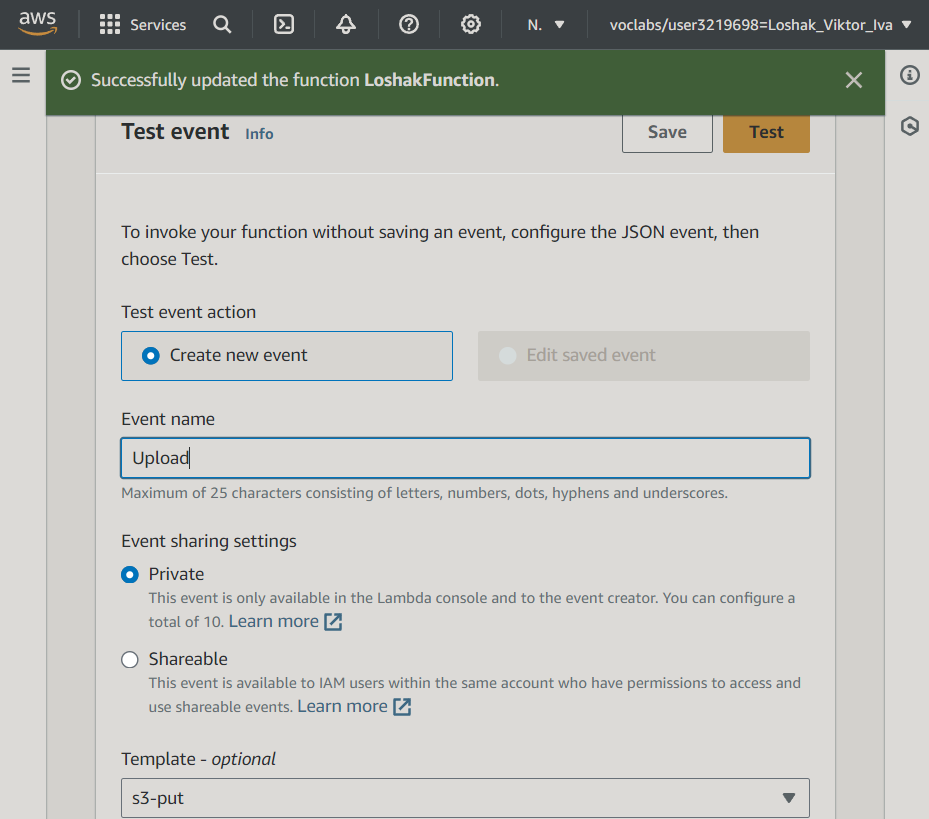


Тут я редагую базові налаштування функції, зокрема опис та обсяг використовуваної пам'яті, який я встановив на 128 MB. Це необхідно для коректної роботи функції при обробці даних.

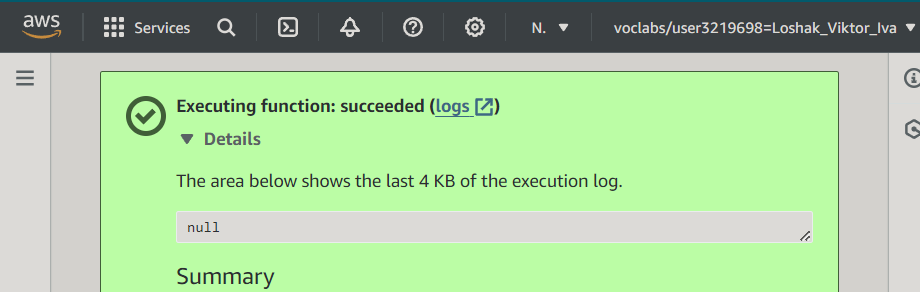
* 1. Test Your Function



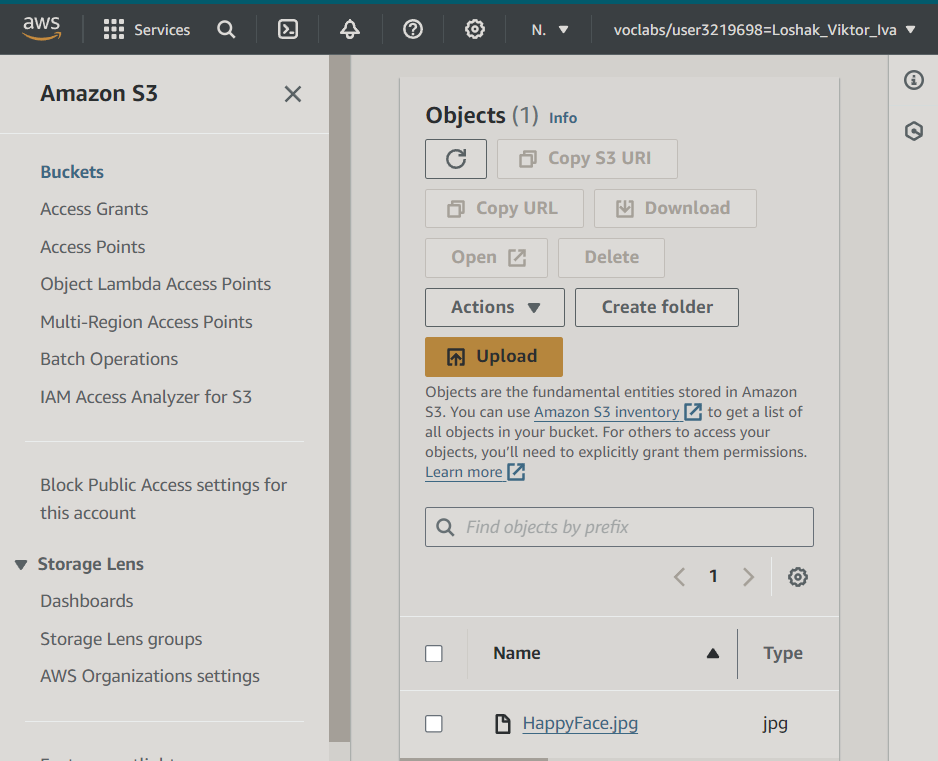
На цьому екрані я створюю новий тестовий подію для моєї Lambda функції "LoshakFunction". Я називаю подію "Upload" та вибираю шаблон події для S3. Це дозволить мені імітувати завантаження файлу у S3, щоб перевірити реакцію функції.



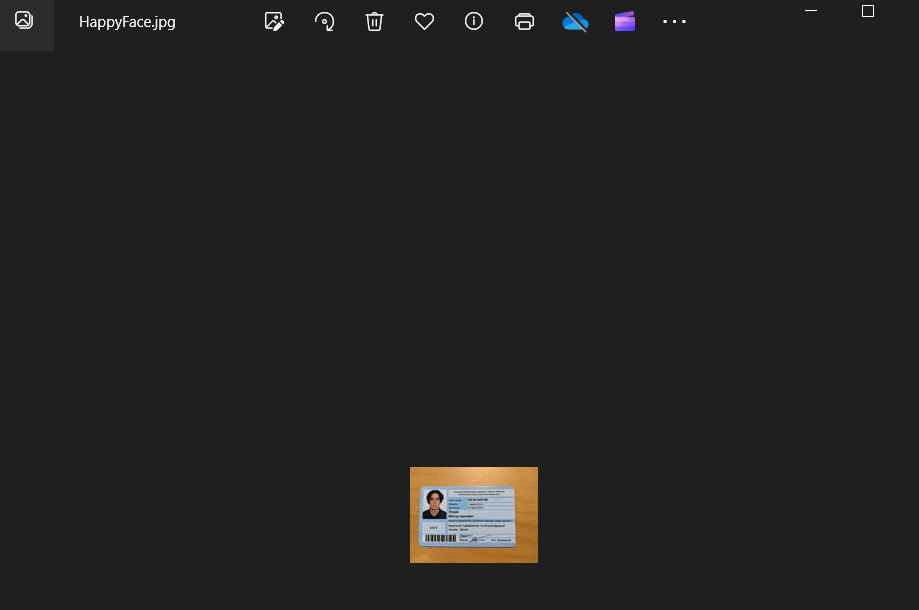
Тут я налаштовую параметри події JSON, яка імітує завантаження файлу "HappyFace.jpg" у S3 bucket. Цей скріншот показує деталі події, включно з ім’ям bucket і ключем об'єкта.



Цей скріншот підтверджує успішне виконання функції "LoshakFunction" після імітації тестової події. Хоча в журналі виконання зазначено "null", це може означати, що функція виконана без видимих помилок або повернення даних.

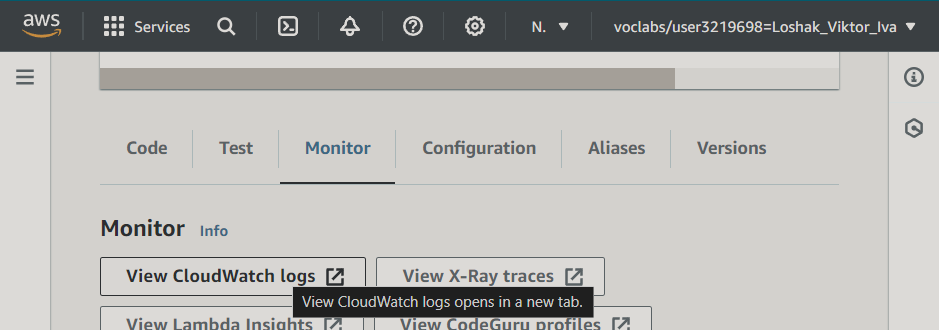


Тут відображено вміст S3 bucket "images-14-loshak", де можна бачити файл "HappyFace.jpg". Це показує, що файл успішно збережений у S3.

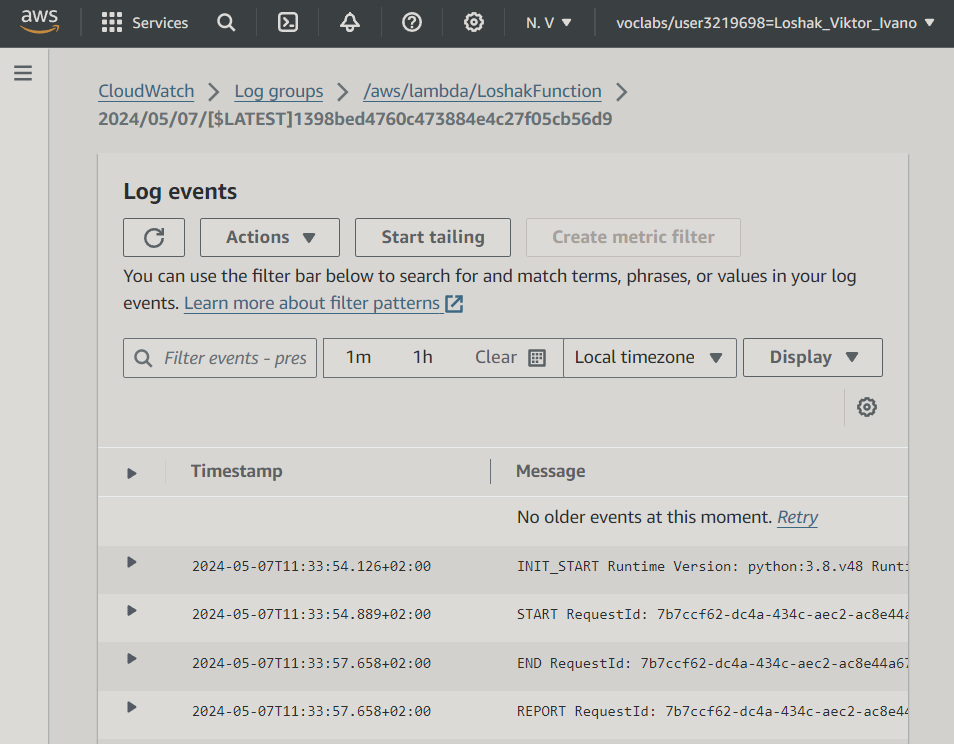


На останньому скріншоті зображено вміст зміненого файлу "HappyFace.jpg", який тепер є зменшеною копією оригіналу. Це підтверджує, що Lambda функція працювала правильно, обробляючи та модифікуючи зображення відповідно до встановлених параметрів.

* 1. Monitoring and Logging



На цьому скріншоті я переглядаю вкладку "Monitor" у консолі AWS Lambda для моєї функції "LoshakFunction". Тут я маю можливість перейти до перегляду логів в CloudWatch, що допоможе мені аналізувати виконання та оптимізувати роботу функції.



Цей скріншот демонструє логи CloudWatch для функції "LoshakFunction". В логах видно часові мітки старту та завершення виконання функції, а також звіт про використання ресурсів. Це дає мені змогу відстежувати події, пов'язані з функцією, і забезпечує інформацію для подальшого аналізу та налаштування.