Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *інфокомунікацій*
(повна назва)
Кафедра *інформаційно-мережної інженерії*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розгортання програм Python 3 на AWS EC2 за допомогою сервера Apache
(тема)

Виконав: студент *2* курсу, групи *ІМІм-19-2*
Корчак М.В.
(прізвище, ініціали)
Спеціальність
172 "Телекомунікації та радіотехніка"
(код і повна назва спеціальності)
Тип програми
Освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма
Інформаційно-мережна інженерія
(повна назва освітньої програми)

Керівник *доцент Кривенко С.А.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ *Безрук В.М.*
(підпис) (прізвище, ініціали)

2021 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____

Керівник_____

Харківський національний університет радіоелектроніки

Факультет _____ _інфокомунікацій_____

Кафедра_____інформаційно-мережної інженерії_____
Рівень вищої освіти _____другий (магістерський)_____
Спеціальність_____172 Телекомунікації та радіотехніка_____
(код і повна назва)
Тип програми _____освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)
Освітня програма_____Інформаційно- мережна інженерія_____

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____*Корчаку Миколі Віталійовичу*_____
(прізвище, ім'я, по батькові)

1. Тема роботи ___*Розгортання програм Python 3 на AWS EC2 за допомогою сервера Apache*_____
_____
затверджена наказом по університету від _12_ _березня 2021р. № _350СТ_
2. Термін подання студентом роботи до екзаменаційної комісії *12 травня 2021 р.*

3. Вихідні дані до роботи *Мова програмування Python 3.7. Платформа хмарних обчислень Amazon Web Services (AWS). Amazon Elastic Compute Cloud (EC2) — сервіс IaaS що надає в користування віртуальні сервери, які контролюються API, основані на гіпервізорі Xen.. Apache HTTP-сервер— відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Web Server Gateway Interface (WSGI) — стандарт взаємодії між Python-програмою, яка виконується на стороні сервера, і самим веб-сервером, наприклад, Apache* .

4. Перелік питань, що потрібно опрацювати в роботі _____
вибір технологій для використання Python у веб-розробці;
реалізація програми на мові Python;
модель розгортання інфраструктури;
результати дослідження.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) *Слайди у форматі Power Point (вступ, переваги мови програмування Python, вибір хмарного провайдеру, результат роботи*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Приміт ка
1	*Ознайомлення із завданням. Уточнення*	*12.03.21*	
2	*Підбір літератури за темою роботи.*	*14.03-30.03.21*	
3	*Виконання розділу 1*	*31.03-02.04.21*	
4	*Виконання розділу 2*	*03.04-05.04.21*	
5	*Виконання розділу 3*	*06.04-08.04.21*	
6	*Виконання розділу 4*	*09.04-30.04.21*	
7	*Оформлення презентаційного матеріалу,*	*01.05-12.05.21*	
	підготовка до захисту у ЕК		

Дата видачі завдання 12 березня 2021р.

Студент _____
(підпис)

Керівник роботи _____ *доц. . Кривенко С.А.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка 50с., 13 рис., 7 джерел.

Об’єкт роботи – хмарний сервіс Amazon Web Services та мова програмування Python.

Мета роботи – написання web-додатку за допомоги мови програмування Python, робота та розробка алгоритму розгортання на Amazon Web Services.

В роботі виконано аналіз сервісів, які пропонують хмарні обчислення. Результатом роботи став алгоритм розгортання web-додатку на Amazon Web Services. При розробці використано фреймворк Flask, при розгортанні використано web-сервер Apache.

Результати роботи можуть бути використані в навчальному процесі.

ХМАРНІ ТЕХНОЛОГІЇ, МОВА ПРОГРАМУВАННЯ PYTHON, ФРЕЙМВОРК FLASK, AMAZONE WEB SERVICES, VIRTUAL PRIVATE CLOUD, APACHE .

ABSTRACT

Explanatory note 50p., 13 fig., 7 sources.

The object of work is the Amazon Web Services cloud service and the Python programming language.

The purpose of the work is to write a web-application using the Python programming language, work and creation of a deployment algorithm on Amazon Web Services.

The analysis of services that offer cloud computing is performed in the work. The result is an algorithm for deploying a web application on Amazon Web Services. The Flask framework was used in the development, and the Apache web server was used in the deployment.

The results of the work can be used in the learning process.

CLOUD TECHNOLOGIES, PYTHON PROGRAMMING LANGUAGE, FLASK FRAMEWORK, AMAZONE WEB SERVICES, VIRTUAL PRIVATE CLOUD, APACHE.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AWS – Amazon Web Services - комерційне публічне хмара, яка підтримується і розвивається компанією Amazon з 2006 року.

WEB - система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету.

Elastic IP- еластичний IP-адрес наданий Amazon Web Services.

VPC – Virtual Private Cloud або віртуальна приватна хмара.

EC2 – Elastic Compute Cloud

ВСТУП

Хмарні обчислення – це майбутнє сфери надання комунікаційних послуг. Про це свідчить постійне зростання їх присутності на ринку. Все більше корпоративних мереж та нових розгортаються завдяки хмарним сервісам. Це обумовлено багатьма факторами, деякі із них це вартість обслуговування та розгортання, доступність даних із будь-якого пристрою, властивості масштабуватись без значних фінансових витрат, збереження даних. Для розгортання web-додатку було обрано Amazon Web Services. Додаток написаний завдяки мові програмування Python та міні фреймворку Flask. Розроблений алгоритм дозволяє розгорнути написаний додаток з допомогою серверу Apache у середовищі Amazon Web Services.

1 ВИБІР ТЕХНОЛОГІЙ ДЛЯ ВИКОРИСТАННЯ PYTHON У ВЕБ-РОЗРОБЦІ

Для початку слід зрозуміти що таке WEB-розробка. Це поняття описує створення, та підтримку web сайтів. В основному розробка поділяється на frontend та backend. Frontend – це та частина з якою взаємодіє користувач (включає: html, css, JavaScript). Backend - містить всю логіку роботи сайту або веб-додатку, так звану бізнес-логіку. Та взаємодіє із базою даних.

1.1 Мова програмування Python

Python було розроблено наприкінці 1980х років Гвідо ван Россумом з National Research Institute for Mathematics and Computer Science, який знаходиться в Нідерландах, в якості послідовника мови ABC.

Python – це інтерпретована об’єктно орієнтована мова високого рівня із динамічною типізацією. Він є одним із найпривабливіший мов програмування коли йдеться мова про швидкість написання програмного забезпечення. Окрім цього Python може використовуватись у якості мови для написання сценаріїв та слугувати зв’язуючим елементом для поєднання різних існуючих компонентів. Через простий синтаксис спрощується супровід коду на всіх етапах розробки, тестування, підтримання. Не варто забувати що це також знижує вартість розробки додатків ті їх подальше обслуговування. Завдяки підтримці модулів та пакетів з’являється, так звана, модульність програми, яка дозволяє використовувати код повторно, що значною мірою підвищую швидкість розробки. Процес налагоджування програм Python простий через те що помилка або невірний фрагмент коду не викликає помилку сегментації, ці помилки виникають коли зафіксовано спробу звернення до недоступних до запису ділянок пам’яті, або при заборонених операціях із нею. Для

налагоджування доступна функція виконання step-by-step, виконуючи по одному рядку коду за раз.

Також Python входить до списку популярних мов програмування за останні роки.

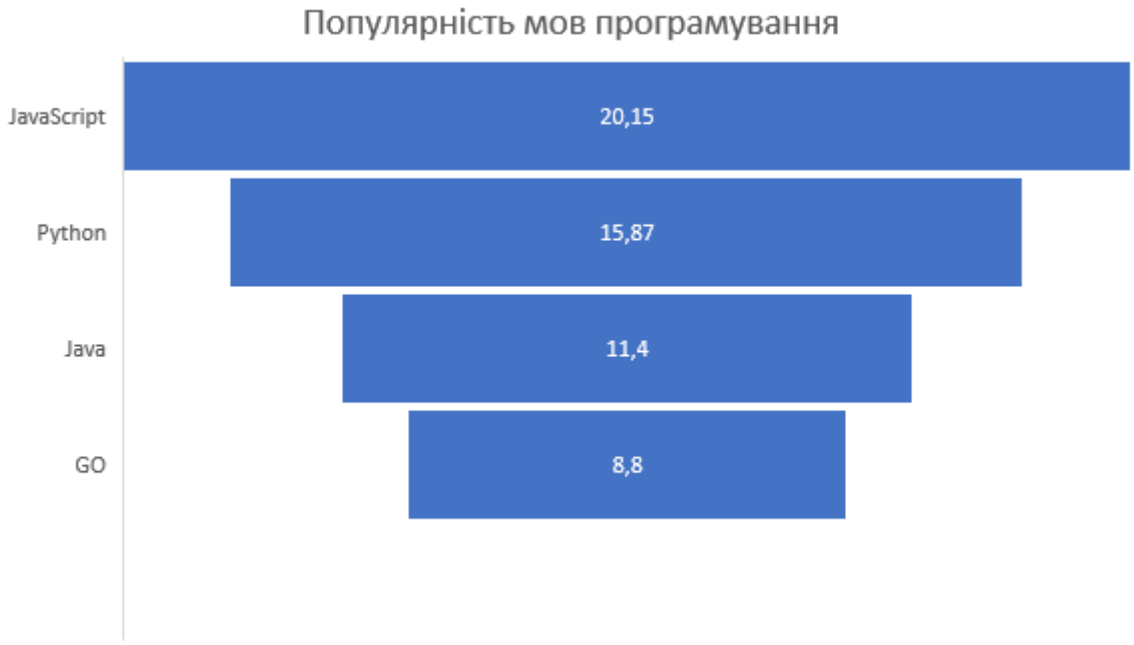


Рисунок. 1.1 – Популярність мов програмування за даними сервісу GitHub

Про це також свідчить вибір компаній таких як Instagram та Spotify саме Python для їх додатків.

Наступна причина для використання мови Python це можливість безкоштовно її використовувати та розповсюджувати. Відомий як продукт із відкритим кодом, це надає роботу із великим набором бібліотек, тому що вся інформація доступна в відкритому доступі в мережі Інтернет. Це означає що копіювання та розповсюдження Python продукту необмежене. Це робить його доволі корисним у світі цифрових систем, оскільки він забезпечує гнучкість на ринку та дозволяє взаємодіяти із багатьма секторами промисловості.

Переваги використання Python для веб-розробки:

- Легкий в вивченні. Простота у синтаксисі дозволяє ефективніше працювати із складними системами та забезпечує більшу продуктивність у взаємодії між розробниками, які зайняті роботою над одним проектом.
- Висока популярність серед веб-розробників. Це означає що якщо ви зіштовхуєтесь із помилкою, при налагоджуванні додатку, або маєте проблему при реалізації нової функції. Є велика вірогідність того що рішення вже знайдене та доступне в мережі Інтернет.
- Гарна читабельність. Через те шо мова Python схожа із англійською це робить код більш зрозумілим при написанні та подальшому використанні.
- Великий вибір бібліотек. Існує велика кількість доступних бібліотек доступних до завантаження, що в свою чергу розширює та прискорює розробку.
- Корисні фреймворки. Вони схожі на набори інструментів із включення пакетів і бібліотек, та модулів із стандартним кодом, що дозволяє підвищити швидкість розробки[1].

1.2 Огляд фрейморку Flask

Flask – це веб-фреймворк. Це означає що він забезпечує користувача набором інструментів, технологій та бібліотек які можуть бути використані у побудуванні різноманітних веб-додатків. Flask являє собою один із, так званих, мікро-фреймворків. Мікро означає що він є фреймворком майже незалежним від зовнішніх бібліотек. Це тягне за собою як позитивні сторони так і негативні. Позитивні полягають у легкості фреймворку, та невеликій залежності від оновлень. Протилежна сторона це накопичення роботи, яку доведеться виконувати власноруч, коли додаток буде нарощувати функціонал, це потребує додавання плагінів та збільшення списку залежностей. У випадку Flask він постачається із залежностями Jinja2, Werkzeg.

Werkzeg- бібліотека веб-додатків WSGI, яка включає в себе:

- Інтерактивний відладчик
- Повноцінний об’єкт запиту з об’єктами для взаємодії із заголовками, даними формами, файлами, файли cookie.
- Утиліти HTTP
- Потоковий сервер WSGI використовується у випадку локальної розробки додатку [2]

Werkzeg – це велика бібліотека веб-додатків. спочатку було розроблено як простий набір інструментів для додатків WSGI, в подальшому трансформувався у той продукт який ми знаємо зараз – велику бібліотеку.

Jinja2 – повноцінний шаблонізатор для Python із функціями:

- Ізольованим режимом виконання що дозволяє відстежувати кожний аспект виконання шаблону
- Наслідуванням шаблонів, дозволяє використовувати схожі або однакові макети для різних шаблонів
- Можливістю налаштування вихідних даних
- Для допомоги дизайнерам Jinja2 йде із набором «помічників» які допомагають вирішувати загальні задачі, такі як розділення послідовностей на декілька стовпців та багато іншого.[3]

1.3 Що таке веб-сервер

Для початку що таке веб-сервер – це визначення відноситься до двох аспектів фізичного обладнання та програмного забезпечення які працюють, як правило, одночасно.

Визначення з боку апаратної частини. Веб-сервер – це комп’ютер на якому знаходиться програмне забезпечення та інші необхідні компоненти для роботи серверу, такі як: HTML, CSS, JavaScript, різного роду файли. Сервер з’єднан з інтернет мережою та обмінюється файлами із користувачами

Визначення з боку програмної частини. Веб-сервер є поєднанням різних протоколів та методів доступу до файлів, які розміщенні на сервері. Використовується для відображення сторінок у вашому браузері, приклад роботи наведений на (Рисунок 1.2).

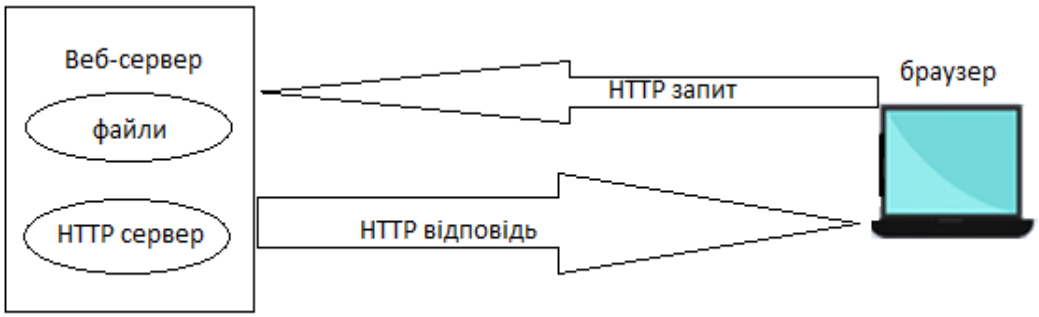


Рисунок. 1.32 – Схема взаємодії веб-серверу та користувача

1.4 Вибір Веб-серверу

Вибір був серед двох найбільш розповсюджених веб-серверів, а саме: Apache та NGINX.

Apache – найбільш розповсюджений веб-сервер. Розроблен та розповсюджується за ліцензією відкритого вихідного коду. Apache використовується майже на половині усіх веб-сайтів, а саме 46%. Цей веб-сервер можна встановити та використовувати безкоштовно. Однією із переваг Apache є його спроможність обробляти великі об’єми трафіку із мінімальним налаштуванням. Також слід зазначити його власність до легкої масштабованості. Завдяки модульній структурі цей веб-сервер можна налаштувати так як вам потрібно, та робити те що потрібно. Ще одна із переваг – можливість видалити непотрібні модулі задля покращення ефективності [4].
Особливості веб-серверу Apache:

- Сумісність із IPv6

- Геолокація за IP-адресою
- Підтримка HTTP/2
- Динамічні модулі які можна встановити
- FTP з’єднання
- Балансування навантаження
- Автоіндексація
- Відстеження Сеансу
- Перезапис URL

NGINX – це програмне забезпечення із відкритим кодом для веб-обслуговування, зворотного проксі-серверу кешування, балансування навантаження. Цей сервер основувався як веб-сервер розрахований максимальну продуктивність та стабільність. Окрім можливостей HTTP серверу він може функціонувати як проксі-сервер для електронної пошти, ті зворотній проксі-сервер та балансування навантаження для серверів HTTP, TCP, UPD.

NGINX – один із найкращих веб-серверів та рішення для доставки додатків, що використовуються веб-сайтами з високим трафіком, такими як Dropbox, Netflix та Zynga. Понад 400 мільйонів веб-сайтів у всьому світі засновані та функціонують із NGINX для швидкого. Деякі особливості серверу NGINX:

- Як програмне забезпечення " все-в-одному", що балансує навантаження, шлюз API, веб-сервер та зворотний проксі-сервер, розроблене для архітектур, що працюють у хмарі, NGINX допомагає вам пришвидшити роботу з IT-інфраструктури та модернізації додатків.
- NGINX - це багатофункціональний інструмент. За допомогою NGINX є можливість використовувати той самий інструмент, що і ваш балансувач навантаження, зворотний проксі-сервер, кеш вмісту та веб-сервер, мінімізуючи обсяг інструментарію та конфігурації, необхідний для

організації. NGINX пропонує документацію та широкий спектр електронних книг, відео, для швидкого налаштування та запуску серверу.

- NGINX продовжує розвиватися. Протягом останнього десятиліття NGINX був на передовій розвитку сучасного Інтернету і допоміг прокласти шлях на шляху від HTTP / 2 до підтримки мікросервісів. Оскільки розробка та доставка веб-додатків продовжує розвиватися, NGINX продовжує додавати функції, що забезпечують доставку додатків - від підтримки конфігурації з використанням JavaScript, до підтримки динамічних модулів . За основами неофіційної статистики сервер NGINX обирають ресурси які мають велику кількість звернень, для яких вкрай важливо швидкодія та здатність обробляти велику кількість звернень [5].

1.5 Вибір хмарного провайдеру

Для початку навіщо взагалі переходити до хмарних провайдерів.

Безпека. Постійні інновації Amazon у сфері кібербезпеки, постійний моніторинг безпеки, як одна із обов’язкових складових для успішного функціонування хмарних сервісів, значно ефективніша за внутрішню безпеку, в більшості випадків.

Масштабованість, хмара надає можливість розширювати будь-який із аспектів інфраструктури за запитом або навіть в автоматичному режимі. Може надаватись такі розширення як збільшення пропускної здатності, не треба і говорити що збільшення пропускної здатності за допомогою хмарного провайдеру набагато простіша ніж збільшувати її на приватному сервері. Також можливо збільшити кількість пам’яті та обчислювальної потужності.

Доступність. Сервіси доступні в хмарі доступні для користувачів в будь якій точці світу, доступні також багато функцій для моніторингу помилок та загальної статистики.

Економія. В більшості випадків провайдер надає можливість платити лише за ті ресурси що використовуються. Також економію відчутно коли планується збільшення проектів, для цього лише потрібно збільшити потужність, а не закуповувати обладнання.

Можливість аварійного відновлення.

1.5.1 Веб-сервіси Amazon

Платформа Amazon охоплює майже всі функції які входять до хмарних обчислень такі як інструменти розробника, інструменти управління. Доступ до майже будь-якого аспекту інфраструктури такого як доступ до обчислювальної потужності, сховищу даних. Поєднання усіх цих інструментів надає потужну основу для подальшої розробки та супроводження кінцевого продукту.

Цінова політика. Корисним бонусом до широкого функціоналу стає грамотна цінова політика. Вона знаходиться на конкурентоспроможному рівні та включає безкоштовні рівні, для стартапів та приватних осіб. Це гарна можливість безкоштовно, до певної межі, оцінити переваги використання хмарної архітектури.

Переваги Amazon Web Services це довга історія надання хмарних послуг. Постійні покращення різноманітних аспектів. Велика кількість документації, що надає потужну допомогу при різного роду труднощів. Також не слід забувати про спеціалістів з обслуговування клієнтів, які можуть надати різного роду інформацію від видів послуг та цінової політики до технічних питань. Велика кількість зон досяжності, які тримаються в тайні, щоб забезпечити більшу надійність системи.

1.5.2 Microsoft Azure

Подібно до AWS, Azure надає повний спектр рішень для розробників. Є можливість управляти розгортанням та управлінням віртуальних машин як заманеться. Також слід відмітити що Azure та AWS підтримують великі

паралельні обчислення, на відміну від наступного кандидата від компанії Google.

Цінова політика має залежність від типу продукту який потрібен користувачу. Ціна за час оренди серверу може бути від 0.099 до 0,149 доларів. Слід зазначити через конкуренцію на ринку хмарних обчислень ціни знаходяться приблизно в однаковому рівні.

Переваги окрім повного спектру функцій платформа Azure є однією із найшвидших рішень. Якщо є необхідності саме у швидкому розгортанні, масштабованості та роботі то Azure є лідером.

1.5.3 Google Cloud Computing

У цієї платформи також є великий набір сервісів для розробників. Слід сказати про продукт AppEngine який дозволяє розробникам створювати додатки без втручання до серверної частини. Є можливість виконувати обчислення, зберігати дані у GCP. Хоча платформа від Google має в розпорядженні меншу кількість сервісів, про те для розробки є все необхідне [6].

1.6 Веб-сервіс EC2

Обчислювальний хмара Amazon Elastic Compute Cloud (Amazon EC2) – це веб-сервіс із надання безпечних обчислювальних ресурсів в «хмарі». Він допомагає розробникам, спрощуючи проведення обчислень в хмарі. Простий веб-інтерфейс сервісу Amazon EC2 дозволяє отримати доступ до обчислювальних ресурсів і налаштувати їх з мінімальними зусиллями, у розділі 3 буде розглянуто приклад розгортання цього сервісу. Він надає користувачам повний контроль над обчислювальними ресурсами, а також перевірену обчислювальну середу Amazon для роботи. Amazon EC2 пропонує обчислювальну платформу з найбільш широкими функціональними

можливостями, яка дозволяє вибрати процесор, сховище, мережу, операційну систему і модель сплати, іншими словами надає повний контроль над платформою.

1.7 Virtual Private Cloud

Virtual Private Cloud (VPC, Віртуальна приватна хмара) – дозволяє виділити ізольований логічний розділ, для запуску віртуальної мережі. Користувач має повний контроль над своєю віртуальною середою та має можливість задавати власний пул IP-адрес, проводити налаштування таблиці маршрутизації, створювати підмережі та підключати інтернет шлюзи. Окрім цього є можливість налаштувати Virtual Private Network для з’єднання із існуючим у користувача центром обробки даних задля розширення можливостей останнього. Один із варіантів розгортання VPC це варіант для розміщення веб-серверів, розміщеного у загальнодоступній частині. А внутрішні системи, наприклад бази даних, сервери додатків розмістити у приватній частині, без доступу до інтернету. Окрім користувач має можливість налаштовувати взаємодію між елементами розгорнутими в кожній із частин. Для кожного об’єкту розгорнутого в підмережі присвоюється сірий IP адрес, а для кожної підмережі відповідно білий для безпосереднього доступу до Інтернету. Для аккаунтів які створенні після 2013 року Amazon пропоную автоматично створену VPC за замовчуванням [7].

Загальний вигляд VPC зображено на(рисунок 1.3).

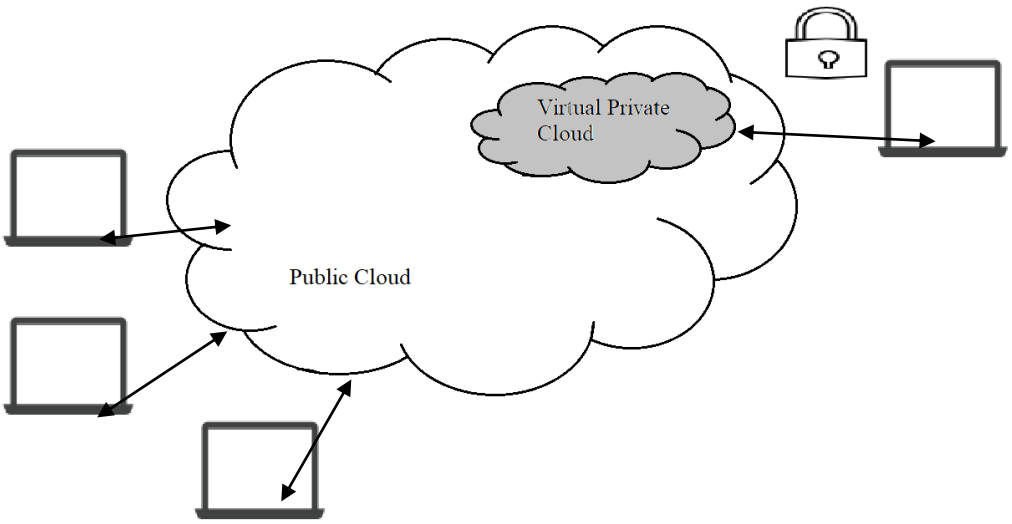


Рисунок. 1.73 – Структура хмарного середовища із VPC

VPC поєднує масштабованість та зручність публічних хмар із ізоляцією приватних.

Ключові концепції які використовує AWS VPS:

- Virtual Private Cloud – віртуальна мережа виділена для користувача AWS
- Підмережа – діапазон IP-адрес в віртуальній приватній хмарі
- Інтернет-шлюз – шлюз який підключається для забезпечення з'єднання VPC з Інтернетом
- Блок Classless Inter-Domain Routing (CIDR) – один із способів виділення IP-адрес та їх маршрутизації
- Таблиця маршрутизації містить правила, які мають назву маршрут. Вони використовуються для визначення куди спрямовано трафік із VPC

Доступ до мережі Інтернет. Ви контролюєте, як екземпляри, що запускаються в VPC, отримують доступ до ресурсів за межами VPC. VPC за замовчуванням включає в себе інтернет-шлюз, і кожна підмережа за замовчуванням є загальнодоступною. Кожен екземпляр, що запускається в підмережі за замовчуванням, має приватний IPv4-адрес і загальнодоступний

IPv4-адрес. Ці екземпляри можуть зв'язуватися з Інтернетом через Інтернет-шлюз. Інтернет-шлюз дозволяє вашим екземплярам підключатися до Інтернету через мережі Amazon EC2.

1.8 Підсумок

На основі проведеного аналізу ринку хмарних послуг вибір був серед головними представниками хмарних сервісів, а саме Amazon Web Services, Microsoft Azure, Google Cloud Platform. Вибір засновувався на критеріях значного обсягу документації, достатнього набору сервісів для розгортання веб-серверу, зрозумілої ціновою політикою, яка може задовільнити починаючого розробника. Головними конкурентами виявились Amazon Web Services та Microsoft Azure. Вибір пав саме на AWS через значно більший обсяг робочої документації та доступність сервісів, високий рівень безпеки.

Наступний до вибору це веб-сервер, обидва варіанти є прийнятними, але було обрано Apache за його розповсюдженість, проблеми які можуть виникнути повинні бути описані в офіційній документації або в мережі інтернет.

2 РЕАЛІЗАЦІЯ ПРОГРАМИ НА МОВІ PYTHON

Реалізація веб-додатку доступна за посиланням https://github.com/KolyaK/diplomPython.git. Структура проекту має вигляд:

…..diplomPython/

…….templates/

…….HomePage.html

…….result.html

…….main.py

…….calculate.py

Логіка роботи програми схематично можна описати як на рисунку(2.1).

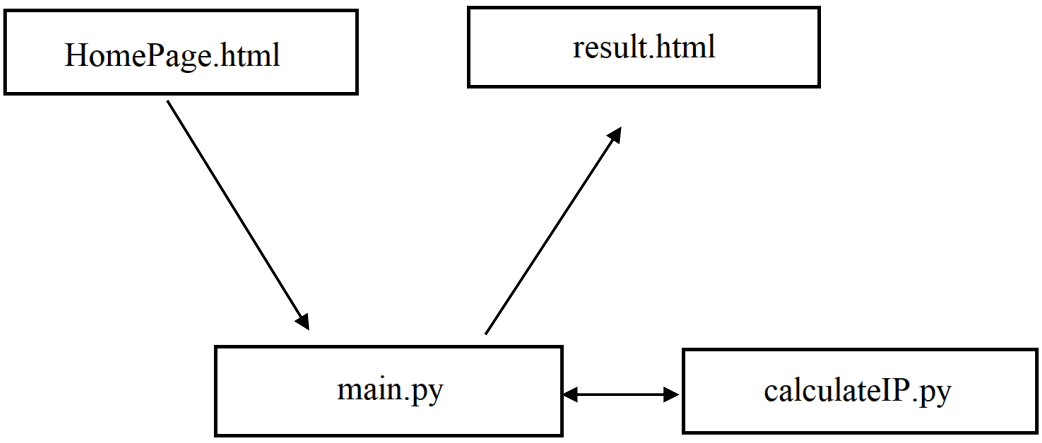


Рисунок. 2.1– Схема роботи програми

Зараз докладніше про те що коїться. При відвідуванні сторінки HomePage.html для взаємодії пропонується 2 форми для вхідних даних, таких як на (рисунок 2.2).

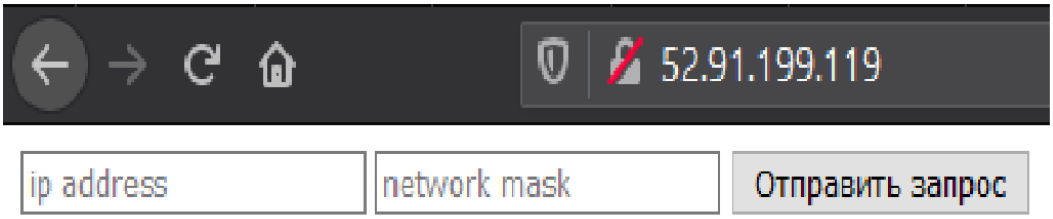


Рисунок. 2.2 - Вигляд домашньої сторінки

Після введення даних в ці форми, із html ці дані передаються до main.py, який в свою чергу передає їх до calculateIP.py файлу. В я кому виконуються усі розрахунки та перетворення із IP адресом. Далі приведено приклад коду який керую цими діями.

<form method="POST">

Метод який передає дані до головного файлу програми.

<input name="text" placeholder="ip address">

<input name="mask" placeholder="network mask">

<input type="submit">

</form>

Наступний лістинг із головного файлу main.py/

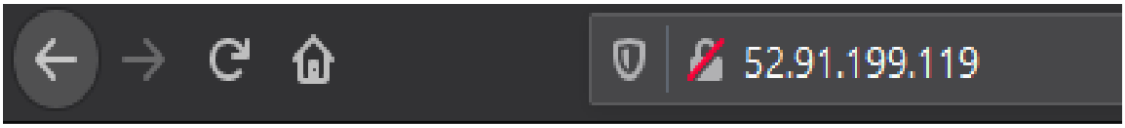
```
from flask import Flask, request, render_template
def my_form():
    return render_template('HomePage.html')
@app.route('/', methods=['GET', 'POST'])
def my_form_post():
    ip_address = request.form['text']
    mask = request.form['mask']
    ip = CalculateIp(ip_address)
    ip.set_mask(mask)
    ip.ip_to_bin()
```

Ця частина приймає дані із HomePage та передає їх до calculateIP.

return render_template("result.html", ipAddress = ip_address,

```
binaryForm = ip.get_bin_ip(),
mask = mask,
binary_mask = ip.bin_mask(),
decimal_wildcard = ip.get_dec_wildcard(),
network = ip.get_network(),
hosts = ip.host(),
broadcast = ip.decimal_broadcast(),
classIP = ip.class_of_ip(mask),)
```

Ця частина передаю вже повернену інформацію із calculateIP до файлу result.html, який вже візуалізує їх у вигляді таблиці, лістинг цього файлу буде наведено у Додатку А. Результат всіх перетворень має вигляд html сторінки із таблицею, яка включає усі перетворення, здійснені у цій програмі та має вигляд (рисунок 2.3)



IP address	124.80.7.45
Class of IP	Class B
Binary form ip	01111100.01010000.00000111.00101101
Mask	15
Decimal wildcard	0.1.255.255
Network	124.80.0.0
Broadcast	124.81.255.255
Hosts	131070

Рисунок. 2.3 - Результат роботи програми

Для відображення результатів використовується шаблонізатор задля підвищення читання головної частини коду. більш швидкого та безпечного редагування виводу, завдяки роботі в окремому html файлі. Невеликий приклад коду приведене нижче

```
<table border="1">
        <tr>
          <td >IP address</td>
          <td>{{ipAddress}}</td>
        </tr>
        <tr>
            <td> Class of IP </td>
            <td>{{classIP}}</td>
        </tr>
        <tr>
</tr>
   </table>
```

У фігурні скобки заключено дані які передаються із головного файлу додатку main.py

3 РОЗГОРТАННЯ ІНФРАСТРУКТУРИ

3.1 Розгортання EC2

Процес розгортання EC2 після налаштування VPC є доволі простим, але є моменти які, на мою думку, треба виділити. Для запуску треба прослідувати наступним крокам:

- Перейти до консолі https://console.aws.amazon.com/ec2
- Натиснути кнопку Launch instances
- Наступний крок це вибіл операційної системи, вибір слід робити опираючись на свій досвід роботи із різними дистрибутивами. Я обрав для роботи дистрибутив який пропонує Amazon на основі CentOS.
- Далі слід обрати тип архітектури ec2, про це докладніше можна дізнатися за посиланням https://aws.amazon.com/ec2/instance-types/.
- Наступний етап – конфігурація є важливою частиною в полі «Network» обрати VPC яку було створено раніше. В полі «Subnet» обрати одну із підмереж, які також було розгорнуто раніше. На останок, в цьому етапі, в полі «auto-assign Pubic IP» обрати Enable. Це дозволить тримати публічну IP адресу.
- Наступне на чому слід загострити увагу є налаштування безпеки. Якщо це перший подібний екземпляр подібного типу, то слід обрати Create a new security group. Та за допомогою кнопки Add Rule добавити нові правила, а саме:
  - Type:SSH, Port Range:22, Source:Custom 0.0.0.0/0 це дозволяє доступ з будь-якої адреси(є можливість обрати лише для IP-адреси з якої здійснено підключення до AWS консолі)

- Type:HTTP, PortRange:80, Source:Custom 0.0.0.0/0, ::/0 (на відміну від попереднього кроку це дозволяє будь-який вхідний та вихідний трафік)
- Type:HTTPS, PortRange:443, Source:Custom 0.0.0.0/0, ::/0

– На останок після запуску буде запропоновано вибрати існуючу або створити нову пару ключів. Які використовуються для доступу до EC2 через протокол SSH.

У разі успішного виконання моє бути приблизно такий результат на консолі EC2(рисунок 3.1)

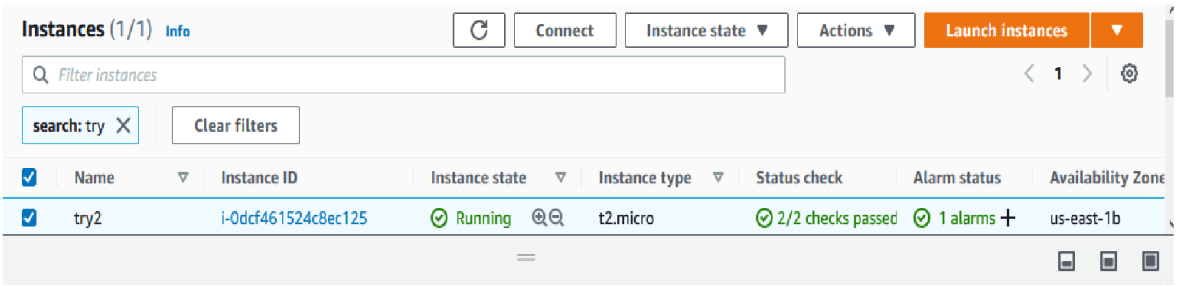


Рисунок. 3.1 Приклад консолі EC2

3.2 Встановлення mod_wsgi

Mod_wsgi – це модуль для серверу Apache який надає сумісний із WSGI інтерфейс для розміщення веб-додатків Python.

Вимоги до Apache. Можна використовувати версії 2.0, 2.2, 2.4. Слід зазначити для систем Linux, у випадку якщо Apache було встановлено із репозиторія, потрібно також встановити відповідний пакет «dev» Apache.

Вимоги до Python. Версія необхідна старша за Python 2.6 або Python 3.3 відповідно. Як у випадку із попередньою вимогою для систем Linux , якщо Python встановлено із репозиторія слід встановити «dev» Python. Бажано щоб

Python біло встановлено із загальним доступом, інакше використання пам’яті буде більшим.

Розпакування вихідного коду. Архів вихідного коду можна отримати із посилання https://github.com/GrahamDumpleton/mod_wsgi/releases. Для розпакування слід виконати команду

tar xvfz mod_wsgi-X.Y.tar.gz

Замість X та У слід встановити завантажену версію.

Налаштування вихідного коду. Щоб налаштувати пакет, готовий до побудови, запустіть скрипт “configure” з каталогу вихідного коду:

./configure

Сценарій налаштування спробує ідентифікувати встановлену програму Apache, використовуючи пошук у різних стандартних місцях для інструментів побудови Apache, що входять до вашого дистрибутиву під назвою «apxs2» або «apxs». Якщо його не знайдено в жодному із цих стандартних місць, буде здійснено пошук вашої PATH.

Яку установку Python використовувати, буде визначено шляхом пошуку виконуваного файлу “python” у вашому PATH.

Якщо ці програми знаходяться не в стандартному розташуванні, їх неможливо знайти у вашому PATH, або ви хочете використовувати альтернативні версії з тими, що знайдені, опції --with-apxs та --with-python можна використовувати разом із сценарієм “configure”:

```
./configure --with-apxs=/usr/local/apache/bin/apxs \
  --with-python=/usr/local/bin/python
```

У деяких дистрибутивах Linux, таких як SUSE та CentOS, потрібно буде використовувати --with-apxs опцію та вказати або “/ usr / sbin / apxs2-worker”, або “/ usr / sbin / apxs2-prefork”. Це необхідно, оскільки розподіли Linux дозволяють встановлювати пакети “dev” для обох варіантів Apache MPM одночасно, тоді як інші дистрибутиви Linux цього не роблять.

Якщо в системі встановлено кілька версій Python, і ви не використовуєте ту, яка є стандартною, можливо це може привести до того що PATH, успадкований програмою Apache під час запуску знайде альтернативну версію. Крім того, директиву WSGIPythonHome слід використовувати для вказівки точного розташування інсталяції Python. Якщо цього не зробити, версія Python, що працює в Apache, може спробувати використовувати модулі Python із неправильної версії Python

Завантаження модуля в Apache. Після того як модулі Apache були встановлені в його каталог його необхідно налаштувати для фактичного завантаження модуля. Те як це робиться та в який із файлів конфігурації це розміщується, залежить від версії Apache та версії дистрибутиву Linux. Тому іноді є необхідність читати документацію операційної системи. В простішому випадку потрібно в основний конфігураційний httpd.conf файл добавити строку:

LoadModule wsgi_module modules/mod_wsgi.so.

Після добавлення модуля до конфігураційного файлу необхідно перезавантажити Apache, щоб упевнитись що все працює як належне. Якщо використовується стандартна версія Apache від Apache Software Foundation то перезавантаження виконується наступною командою:

apachectl restart.

Якщо при цьому виникли якісь проблеми або якщо ви оновлюєте версію mod_wsgi, є рекомендація замість перезавантаження зупинити і запустити Apache.

apachectl stop
apachectl start

Слід зазначити що в залежності від дистрибутиву Linux, де Apache заздалегідь встановлено, в результаті остання командо може не працювати або працювати із помилками. В цих випадках слід використовувати дозволенний спосіб перезапуску системних служб. В моєму випадку використовувався дистрибутив на основі RedHat та команди мали наступний вигляд:

systemctl stop httpd

systemctl start httpd

Майже завжди команди для перезавантаження Apache необхідно виконуваи від імені root або через sudo. Якщо все пройшло без помилок. В файлі журналу помилок Apache буде наступний запис:

Apache/2.4.8 (Unix) mod_wsgi/4.4.21 Python/2.7 configured

3.3 Налаштування mod_wsgi

Перший спосіб це використання WSGIScriptAlias для зазначення який додаток WSGI буде використане ті поєднує його із зазначеним URL

WSGIScriptAlias /myapp /usr/local/wsgi/scripts/myapp.wsgi

Остання частина повинна вказувати на місце знаходження файлу із кодом WSGI. Слеш в кінець не треба ставити, якщо це відноситься до реального файлу, а не директорії. Для перевірки працездатності mod_wsgi використаємо приклад наведений в документації до mad_wsgi:

```
def application(environ, start_response):
    status = '200 OK'
    output = b'Hello World!'
    response_headers = [('Content-type', 'text/plain'),
                        ('Content-Length', str(len(output)))]
    start_response(status, response_headers)
    return [output]
```

Слід зазначити що використовувати абсолютний шлях до WSGI скріпту а не тільки за назвою модуля Python. Одна із причин цього те шо Apache контролює доступ до WSGI додатку. У випадку якщо файли знаходяться за межами відомих Apache, необхідно повідомити що ці файли можуть бути використані.

```
<Directory /usr/local/wsgi/scripts>
    Order allow,deny
    Allow from all
```

```
</Directory>
```

Ці команди використовуються для версії молодшу за 2.4.

```
<Directory /usr/local/wsgi/scripts>
   Require all granted
</Directory>
```

А ці команди для версії з 2.4 та більш нову.

Для WSGIScriptAlias, що би змонтувати додаток у корені сайту, достатньо використати «/» замість точки монтування.

```
WSGIScriptAlias / /usr/local/wsgi/scripts/myapp.wsgi
```

Якщо є необхідність змонтувати декілька додатків, є можливість вказати директиви декілька разів. При цьому пріоритет віддається тому який написано першим. Слід зазначити що при цьому скріпти які використовують додаткові URL-адреси слід ставити в початок.

```
WSGIScriptAlias /wiki /usr/local/wsgi/scripts/mywiki.wsgi
WSGIScriptAlias /blog /usr/local/wsgi/scripts/myblog.wsgi
WSGIScriptAlias / /usr/local/wsgi/scripts/myapp.wsgi
```

Ще один спосіб це порівняння директиви WSGIScriptAlias із каталогом, який містить декілька або навіть один додаток WSGI.

```
WSGIScriptAlias /wsgi/ /usr/local/wsgi/scripts/
```

В цьому випадку частина після URL-адреси використовується для ідентифікації додатку WSGI до виконання в зазначеному каталозі. При цьому и точка монтування і шлях до каталогу мають містити «/» на кінці.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

4.1 Налаштування VPC

Вимоги. Підмережі VPC мають знаходитись в різних зонах досяжності в регіоні де буде знаходитись робочий простір. Зона досяжності – це окремі місяця, які сконструйовані таким чином, щоб буди захищеними від збоїв в інших зонах досяжності. Це дає захист програм від збоїв в одній зоні досяжності завдяки резервній зоні досяжності. Для розгортання та налаштування VPC слід дотримуватись наступних кроків:

Виділення Elastic IP-адреси. Elastic IP – є статичною IPv4-адресою призначеною для динамічних хмарних обчислень. Вона поєднується із аккаунтом AWS і залишається у використанні поки власник не відмовиться від неї. Призначення Elastic IP поєднати екземпляр користувача із Інтернетом. Для отримання цієї адреси потрібно виконати:

- Відкрити консоль Amazone VPC яку можна знайти за адресою https://console.aws.amazon.com/ec2
- На панелі навігації слід обрати Elastic IPs
- Наступний крок натиснути кнопку Allocate Elastic IP address
- На сторінці Allocate Elastic IP address для публічного пулу адрес iPv4 виберіть пул IPv4-адрес Amazon , загальнодоступну IPv4-адресу, яку ви додаєте до свого облікового запису AWS , або пул адрес IPv4 , що належить клієнту , а потім виберіть Allocate
- Радіти отриманню Elastic IP

В кінці буде отримано результат подібний до (рисунок 4.1)

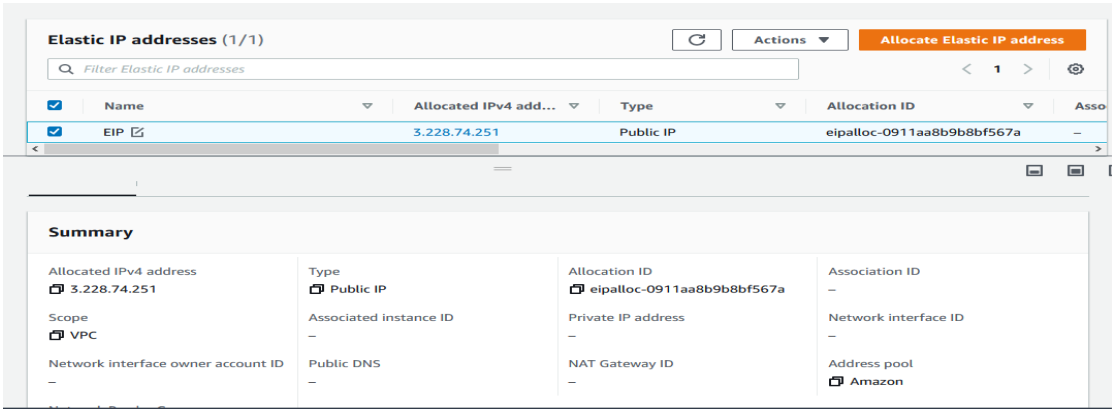


Рисунок. 4.1 - Приклад отримання Elastic IP

Наступний крок це створення VPC, для цього слід дотримуватись наступних кроків:

- Відкрити Amazon VPC консоль за адресою https://console.aws.amazon.com/vpc/
- У полі навігації вібрати Your VPCs
- На відкритій сторінці натиснути кнопку Create VPC, у правому верхньому куті
- У полі Name потрібно ввести назву створюваної VPC, рекомендую використовувати назву яка описує призначення цієї хмари, в моєму випадку це «web-serv»
- У наступному полі IPv4 CIDR block Amazon рекомендую використовувати блок CIDR із приватного діапазону. Наприклад 10.0.0.0/16, пропоную так і зробити
- Для блоку IPv4 CIDR залишити за замовчуванням «No IPv6 CIDR block»
- Також залишити стандартним наступне значення Tenancy
- Натиснути кнопку Create VPC
- Якщо все пройшло вдало має вийти результат подібний до (рисунок 4.2)

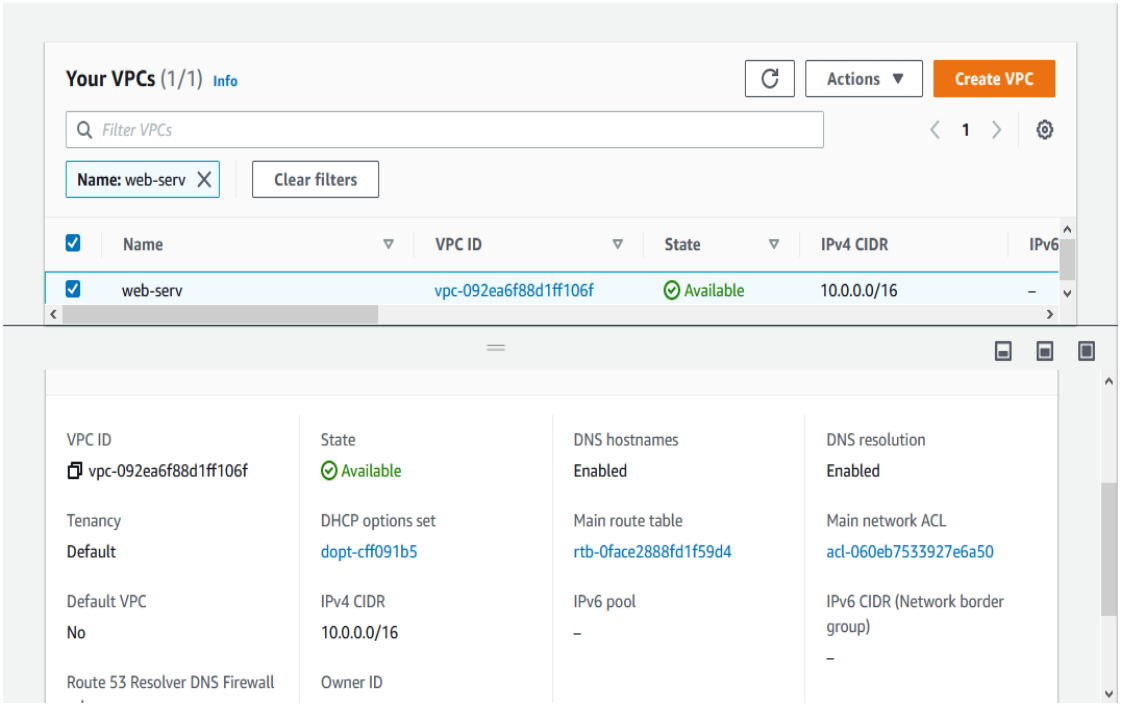


Рисунок. 4.12 – Приклад створеної VPC

Наступний крок – створення двох підмереж одна приватна друга публічна, які розміщені у різних зонах досяжності.

- Відкрити Amazon VPC консоль за адресою https://console.aws.amazon.com/vpc/
- Для створення першої підмережі натиснути кнопку Create subnet
- В меню VPC ID обрати раніше створену VPC
- В налаштуваннях саме підмережі в першому полі потрібно ввести назву майбутньої підмережі, в моєму випадку «public sub web serv»
- Далі потрібно обрати зону досяжності, як було написано вище, підмережі мають розміщуватись в різних зонах, отже вибрану в цьому пункті зону неможливо обрати при створенні приватної підмережі
- Ввести в поле IPv4 CIDR block значення «10.0.1.0/24»
- Натиснути кнопку Create subnet

– Для створення другої підмережі слід повторити попередні пункти окрім вибору зони досяжності (вона має бути іншою) та в полі IPv4 CIDR block ввести значення «10.0.2.0/24»

Вигляд вкладки Subnets має бути подібною до (рисунок 4.3)

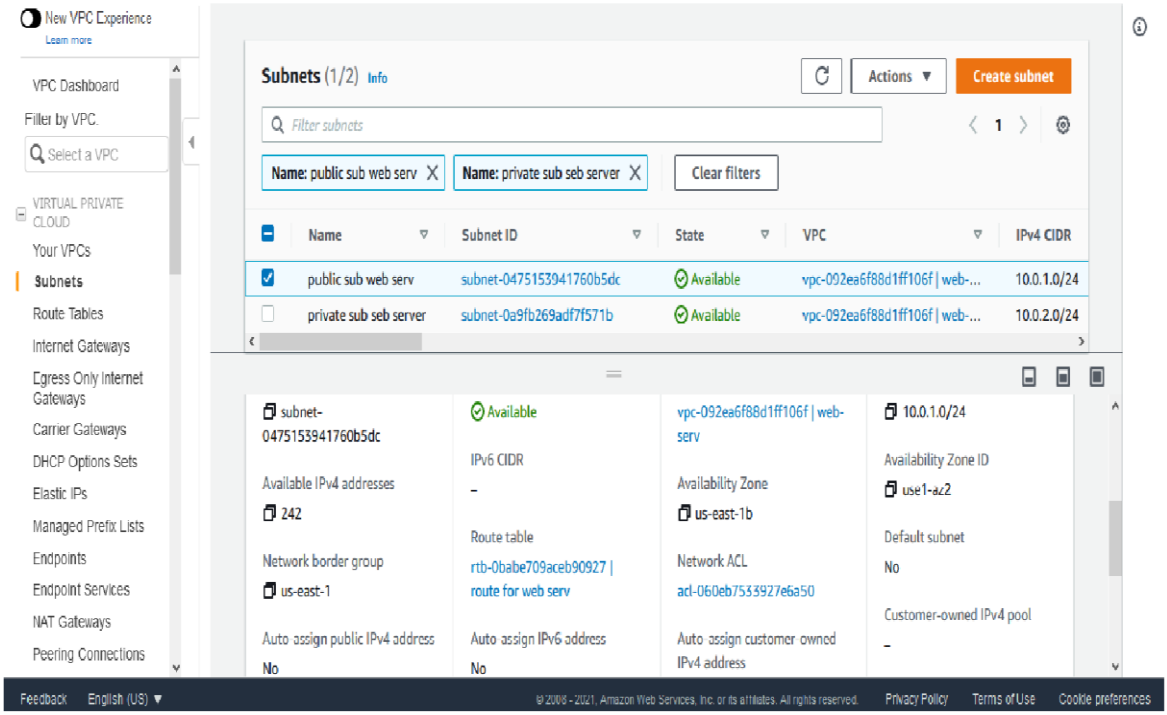


Рисунок. 4.1.1 – Приклад створених підмереж

Наступний крок створення інтернет-шлюзу. Інтернет шлюз слугую двом речам: надавати ціль в таблиці маршрутизації VPC для трафіку із Інтернету та виконувати перетворення мережевих адрес для екземплярів із загальнодоступними адресами. Для створення необхідно лише перейти на вкладку Internet Gateway та натиснути кнопку Create Internet Gateway, після чого лише вказати ім’я майбутнього шлюзу, наприклад «Internet gateway for web serv».

Після створення відкриється вкладка на якій є можливість приєднати створений шлюз до VPC для цього:

- Натиснути на кнопку Actions в правому верхньому куті, та обрати Attach to VPC
- У меню Available VPCs обрати VPC до якої слід приєднати цей щлюз.

Для з’єднання всього створеного раніше потрібно створити Route table необхідно:

- Перейти у вкладку Route Tables
- Натиснути Create route table після цього назначити ім’я для таблиці маршрутизації та прив’язати її до певної VPC
- Після повернення до вкладки Route tables слід обрати створену таблицю
- Нижче списку таблиць маршрутизації з’явиться вкладка Routes,при обранні якою, стане доступна кнопка Edit routes, яку потрібно натиснути для редагування маршрутів.
- Для доступу VPC до мережі інтернет слід прописати правило Destination:0.0.0.0/0 (це дозволяє трафік із будь-яких джерел) Target: раніше створений інтернет-шлюз
- Во вкладці Subnet associations слід добавити обидві створені раніше підмережі.

Якщо результат успішний він має бути подібним до (рисунок 4.4-4.5)

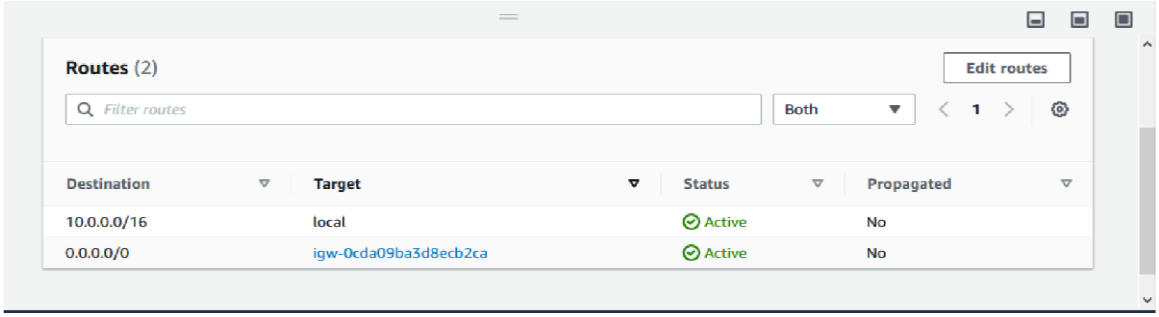


Рисунок. 4.1.2 – приклад вкладки Routes

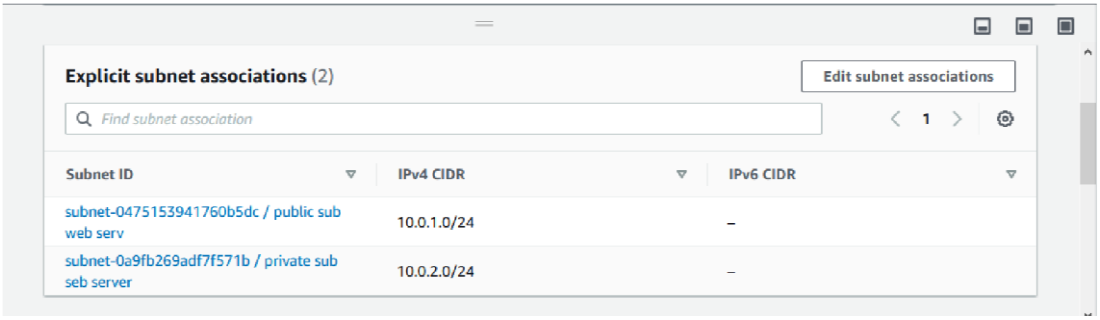


Рисунок. 4.1.3 - Приклад вкладки Subnet associations

4.2 Налаштування Apache та mod_wsgi

Після підключення до EC2 через протокол SSH слід виконати команди:

sudo yum install httpd

Успішне виконання видає текст типу:

Installed:

httpd.x86_64 0:2.4.46-1.amzn2

Dependency Installed:

apr.x86_64 0:1.6.3-5.amzn2.0.2

apr-util.x86_64 0:1.6.1-5.amzn2.0.2

apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2

generic-logos-httpd.noarch 0:18.0.0-4.amzn2

httpd-filesystem.noarch 0:2.4.46-1.amzn2

httpd-tools.x86_64 0:2.4.46-1.amzn2

mailcap.noarch 0:2.1.41-2.amzn2

mod_http2.x86_64 0:1.15.14-2.amzn2

Complete!

Наступним, згідно із документацією, слід встановити http-devel за допомогою команди:

sudo yum install httpd-devel

Після виконання повинне буди схоже повідомлення про успішне виконання.

Про успішне встановлення серверу має свідчити вивід тестової сторінки в браузері, я на (рисунок 4.6)

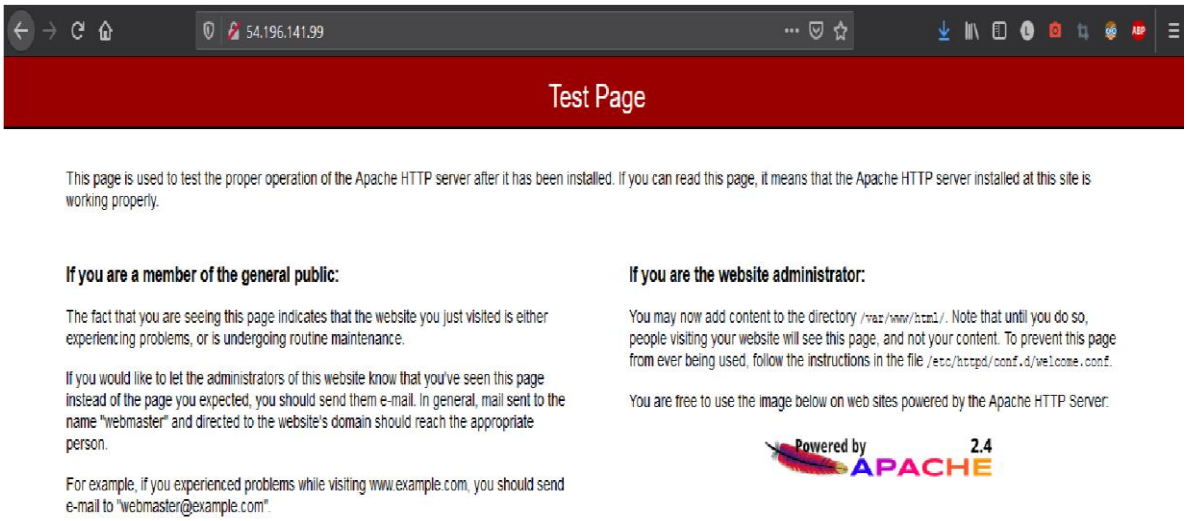


Рисунок. 4.2.6 - Тестова сторінка Apache

Наступний крок встановлення модуля mod_wsgi, це питання вже описано у розділі 3.2, а в цьому випадку було використано команду.

sudo yum install mod_wsgi

Наступним кроком має бути створення робочої директорії та налаштування Apache на роботу із ними. В даному випадку весь проект міститься на сервісі github, для взаємодії з ним необхідно додатково встановити пакет git за допомогою команди «sudo yum install git» та наступною «git clone https://github.com/KolyaK/diplomPython.git», також за цим посилання можна ознайомитись із кодом та структурою додатку. Наступний етап це створення wsgi файлу із наступним змістом:

```
import sys
sys.path.insert(0, 'usr/bin/python3')
from main import app as application
```

Цей фрагмент вказує на знаходження Python та імпортує файл main як додаток для виконання. Та вказує на місце знаходження інтерпретатору Python .Останній крок це прописати в конфігурації Apache наступні рядки :

```
WSGIDaemonProcess main python-home=/var/www/fl/venvs/diplom
 WSGIPythonPath usr/lib64/python3.7
    WSGIApplicationGroup %{GLOBAL}
    WSGIScriptAlias / /var/www/diplom/fappw.wsgi
    <Directory /var/www/diplom>
        Require all granted
    </Directory>
```

Докладніше про цей фрагмент:

WSGIDaemonProcess вказує які процеси демону мають бути створені та до яких запущених процесів делеговано. Звернімо увагу на те що процеси демону які повині бути створені за допомогою WSGIDaemonProcess , мають бути делеговані за допомогою WSGIcriptAlias. Python-home=/var/www/fl/venvs/diplom вказує на місцезнаходження віртуальної середи яка використовується цим демоном. Віртуальна середа створюється за допомогою команди python -m venv.

Треба зберігати обережність віртуальна середа має бути створена тою версією Python яка використовувалась при компіляції mod_wsgi. Неможливо примусити працювати mod_wsgi із версією Python, іншою від використаною при компілюванні. Тобто неможливо одну й ту саму версію запустити на Python другої та третьої версії одночасно.

WSGIPythonPath використовується для позначення директорій в яких може бути здійснено пошук додаткових модулів Python.

WSGIScriptAlias використовується для того щоб вказати на кінцевий каталог, який містить файл сценарію wsgi. В цьому випадку файл fappw.wsgi буде визвано при звернені до домашнього адресу, наприклад hhtp://www.example.com. Питанням безпеки є місцерозташування WSGI

сценаріїв, слід уникати розміщення їх у DocmentRoot, це директорія яка формує головне дерево документів доступних із мережі інтернет. Якщо фал буде досяжний із мережі цим можуть скористатися зловмисники для своїх цілей. WSGIScriptAlias спрощую відображення URL адрес та одночасно вказує на місце знаходження файлів сценарію. Для налаштування доступні наступні параметри.

Process-group=name вказує в якій групі процесів буде виконуватися сценарій WSGI. Якщо встановлено значення «%{CLOBAL}» як ім’я групи процесів, це означає шо ця група процесів буде виконуватися в контексті стандартних процесів Apache. Такі сценарії будуть виконуватись із найменшою затримкою, але вони будуть використовувати однакове процесорний простір що й інші модулю.

Остання частина помічає каталог як один із каталогів , які належать Apache та дозволяє запроси до нього. Цей приклад актуальний для версії починаючи із Apache 2.4.

Для версії молодшу за Apache 2.4. треба використовувати наступний синтаксис.

```
<Directory /var/www/diplom>
    Order allow,deny
        Allow from all
    </Directory>
```

На цьому налаштування закінчені.

ВИСНОВОК

В ході атестаційної роботи було виконано аналіз ринку хмарних провайдерів, та для подальшого аналізу обрану лідерів. Після порівняльного аналізу було обрано Amazon Web Services у якості провайдеру. Розгортання проведено на сервісі EC2.Для забезпечення вимог безпеки було задіяно Virtual Private Cloud, це забезпечило безпеку та гнучкість у налаштуванні всієї мережевої інфраструктури. Для розробки додатку було використано мову програмування Python за швидкість написання додатків та доволі легкий початок використання цієї мови. У якості фреймворку використано мікро-фреймворк Flask за його мінімальну кількість налаштувань перед початком розробки. У свою чергу він забезпечив легкий для зрозуміння додаток, у випадку його подальшого супроводження. Увесь цей стек працює завдяки веб-серверу Apache 2.4 та додатковим модулем mod_wsgi.

Головною метою цієї роботи було розробка методології розгортання прикладних програм на AWS. Шлях викладений є цілком робочим та актуальним. Налаштування проведені в ході виконання забезпечують необхідний мінімум для розгортання будь-якого додатку написаного із допомогою Python та Flask. Із кодом програми можна ознайомитись за посиланням https://github.com/KolyaK/diplomPython.git

ПЕРЕЛІК ПОСИЛАНЬ

1. Tiago M. Why use Python for Web Development? [Електронний ресурс] / Madeira Tiago. – 2020. – Режим доступу до ресурсу: https://www.imaginarycloud.com/blog/why-use-python-for-web-development/.
2. Jinja [Електронний ресурс] – Режим доступу до ресурсу: https://www.palletsprojects.com/p/jinja/.
3. Werkzeug [Електронний ресурс] – Режим доступу до ресурсу: https://www.palletsprojects.com/p/werkzeug/.
4. Hernandez J. What is Apache? In-Depth Overview of Apache Web Server [Електронний ресурс] / Jovan Hernandez – Режим доступу до ресурсу: https://www.sumologic.com/blog/apache-web-server-introduction/.
5. What is NGINX? [Електронний ресурс] – Режим доступу до ресурсу: https://www.nginx.com/resources/glossary/nginx/.
6. Comparing AWS vs Azure vs Google Cloud Platforms For Enterprise App Development [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://medium.com/@distillerytech/comparing-aws-vs-azure-vs-google-cloud-platforms-for-enterprise-app-development-28ccf827381e.
7. What Is a Virtual Private Cloud (VPC)? [Електронний ресурс] – Режим доступу до ресурсу: https://www.cloudflare.com/learning/cloud/what-is-a-virtual-private-cloud/.