

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Аналіз та порівняння розробки інфраструктури на базі Amazon Web Services і
використання власних серверів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Денис ОМЕЛЬЧЕНКО
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД- 42

Денис ОМЕЛЬЧЕНКО

Ім'я, ПРІЗВИЩЕ

Керівник: Юлія КАГРАМАНОВА

науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Рецензент: _____

науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІПЗАС

_____ Каміла СТОРЧАК

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Омельченко Денис Вадимович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Аналіз та порівняння розробки інфраструктури на базі Amazon Web Services і використання власних серверів

керівник кваліфікаційної роботи _____ **Юлія КАГРАМАНОВА**

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36

2. Строк подання кваліфікаційної роботи «31» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

1. Науково-технічна література з теми бакалаврської роботи.

2. Принципи функціонування хмарних обчислень і серверної інфраструктури.

3. Основні принципи та технології використання Amazon Web Services (AWS).

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз використання Amazon Web Services

2. Аналіз використання власних серверів

3. Порівняння AWS та власних серверів

5. Ілюстративний матеріал: *презентація*

6. Дата видачі завдання: «27» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	15.03.24 – 01.04.24	
2	Аналіз інфраструктури на базі AWS та власних серверів	02.04.24 – 15.04.24	
3	Визначення критеріїв порівняння та розробка методології дослідження	16.04.24 – 21.04.24	
4	Розробка інфраструктури на базі AWS	22.04.24 – 27.04.24	
5	Аналіз та порівняння результатів	28.04.24 – 13.05.24	
6	Оформлення роботи: вступ, висновки, реферат	14.05.24 – 19.05.24	
7	Розробка демонстраційних матеріалів	20.05.24 – 25.05.24	

Здобувач(ка) вищої освіти

(підпис)

Денис ОМЕЛЬЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Юлія КАГРАМАНОВА

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавр: 53 стор., 27 рис. , 20 джерела.

Мета роботи – дослідити та порівняти розробку інфраструктури на базі Amazon Web Services (AWS) та використання власних серверів, з урахуванням економічних, технічних та експлуатаційних аспектів.

Об'єкт дослідження – інформаційні технології та інфраструктура для розробки і розгортання програмного забезпечення.

Предмет дослідження – методи та підходи до розробки IT-інфраструктури з використанням хмарних сервісів AWS та власних серверів..

Короткий зміст роботи: У роботі досліджено можливості та особливості розробки IT-інфраструктури за допомогою хмарних сервісів AWS, а також проведено порівняння з традиційним підходом використання власних серверів.

Також проведено аналіз економічних та технічних аспектів обох підходів, включаючи витрати, масштабованість, надійність, безпеку та гнучкість. У практичній частині роботи розроблено та налаштовано IT-інфраструктуру на базі AWS, включаючи використання сервісів Amazon EC2, Amazon S3, Amazon RDS. Проведено тестування продуктивності та надійності створеної інфраструктури.

КЛЮЧОВІ СЛОВА: ХМАРНІ ТЕХНОЛОГІЇ, AMAZON WEB SERVICES, ВЛАСНІ СЕРВЕРИ, ІНФРАСТРУКТУРА, ЕКОНОМІЧНИЙ АНАЛІЗ, ТЕХНІЧНИЙ АНАЛІЗ, EC2, S3, VPC, RDS.

ABSTRACT

Text part of the bachelor level qualification work: 53 pages, 27 pictures, 20 sources.

The purpose of the study - is to investigate and compare the development of infrastructure based on Amazon Web Services (AWS) and the use of own servers, taking into account economic, technical and operational aspects.

The object of research is information technology and infrastructure for software development and deployment.

The subject of the study is the advantages and disadvantages of using AWS cloud services and own servers to create IT infrastructure.

Summary of the work: The paper investigates the possibilities and features of developing IT infrastructure using AWS cloud services, as well as compares them with the traditional approach of using own servers.

The paper also analyses the economic and technical aspects of both approaches, including costs, scalability, reliability, security and flexibility. In the practical part of the work, an AWS-based IT infrastructure was developed and configured, including the use of Amazon EC2, Amazon S3, and Amazon RDS services. The performance and reliability of the created infrastructure have been tested.

KEYWORDS: *CLOUD TECHNOLOGIES, AMAZON WEB SERVICES, OWN SERVERS, INFRASTRUCTURE, ECONOMIC ANALYSIS, TECHNICAL ANALYSIS, EC2, S3, VPC, RDS.*

ЗМІСТ

ВСТУП.....	10
1 ІТ-ІНФРАСТРУКТУРА: ВАЖЛИВІ СКЛАДОВІ	12
1.1 Історія еволюції ІТ-інфраструктури.....	13
1.2 Взаємодія апаратного та програмного забезпечення в ІТ-інфраструктурі	14
1.3 Типи ІТ-інфраструктури.....	16
2 ОГЛЯД ТЕХНІЧНИХ ЗАСОБІВ.....	18
2.1 Огляд AWS	18
2.2 EC2	20
2.3 Віртуальна приватна хмара (VPC)	28
2.4 Amazon Simple Storage (S3)	30
2.5 Amazon RDS.....	34
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПОБУДОВИ ІТ ІНФРАСТРУКТУРИ У ХМАРІ	39
ВИСНОВКИ.....	61
ПЕРЕЛІК ПОСИЛАНЬ.....	63
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	65

ВСТУП

Актуальність теми: У сучасному інформаційному просторі питання розробки та підтримки інфраструктури ІТ-систем набуває все більшої актуальності. Зростання обсягів даних, складність програмного забезпечення та динамічне середовище ринку роблять традиційні підходи до розробки інфраструктури неефективними.

Мета дослідження: Метою даного дипломного проекту є аналіз та порівняння двох основних підходів до розробки інфраструктури ІТ-систем: використання Amazon Web Services (AWS) та використання власних серверів.

Завдання дослідження:

- Дослідити основні концепції та принципи роботи AWS.
- Проаналізувати переваги та недоліки використання AWS для розробки інфраструктури ІТ-систем.
- Дослідити особливості розробки інфраструктури на базі власних серверів.
- Провести порівняльний аналіз AWS та власних серверів за низкою критеріїв, таких як вартість, масштабованість, гнучкість, безпека та надійність.
- На основі проведеного аналізу зробити висновки щодо доцільності використання AWS або власних серверів для розробки інфраструктури ІТ-систем у конкретних випадках.

Об'єкт дослідження: Об'єктом дослідження є AWS та власні сервери як два альтернативні підходи до розробки інфраструктури ІТ-систем.

Предмет дослідження: Предметом дослідження є порівняльний аналіз AWS та власних серверів за низкою критеріїв, що впливають на ефективність розробки та експлуатації інфраструктури ІТ-систем.

Методи дослідження: Для досягнення мети та виконання завдань дослідження будуть використовуватися такі методи:

- Аналіз наукової літератури та публікацій з даної теми

- Вивчення офіційної документації AWS та інших платформ хмарних обчислень.
- Проведення експериментів та тестування AWS та власних серверів.
- Порівняльний аналіз отриманих результатів.

Очікувані результати:

В результаті дослідження очікується отримати:

- Глибоке розуміння принципів роботи AWS та інших платформ хмарних обчислень.
- Детальний аналіз переваг та недоліків використання AWS для розробки інфраструктури ІТ-систем.
- Систематизовану інформацію про особливості розробки інфраструктури на базі власних серверів.
- Об'єктивний порівняльний аналіз AWS та власних серверів за низкою критеріїв.
- Практичні рекомендації щодо доцільності використання AWS або власних серверів для розробки інфраструктури ІТ-систем у конкретних випадках.

Практична значимість:

Результати дослідження можуть бути використані для:

- Обґрунтування вибору платформи для розробки інфраструктури ІТ-систем.
- Оптимізації витрат на розробку та експлуатацію інфраструктури ІТ-систем.
- Підвищення ефективності та надійності ІТ-систем.

Публікації:

1. V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ» 18.04.2024 - ПЕРСПЕКТИВА ХМАРНИХ ТЕХНОЛОГІЙ У МАЙБУТНЬОМУ
2. V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ» 18.04.2024- ВПЛИВ ХМАРНИХ ТЕХНОЛОГІЙ НА РОЗВИТОК ШТУЧНОГО ІНТЕЛЕКТУ

1 ІТ-ІНФРАСТРУКТУРА: ВАЖЛИВІ СКЛАДОВІ ТА ЇХ ЗНАЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ

ІТ-інфраструктура (інформаційно-технологічна інфраструктура) - це комплекс апаратних, програмних та інформаційних ресурсів, які необхідні для підтримки роботи інформаційних систем та забезпечення їх функціонування. Вона є основою для будь-якої ІТ-системи, включаючи веб-сайти, корпоративні мережі, програмні додатки та бази даних.

Технології впливають майже на всі аспекти сучасного бізнесу - від роботи окремого працівника до операцій і товарів та послуг. При належній організації мережі технології можна оптимізувати для покращення комунікації, підвищення ефективності та продуктивності.

Якщо ІТ-інфраструктура є гнучкою, надійною та безпечною, вона може допомогти підприємству досягти своїх цілей та забезпечити конкурентну перевагу на ринку. З іншого боку, якщо ІТ-інфраструктура не впроваджена належним чином, бізнес може зіткнутися з проблемами зв'язку, продуктивності та безпеки, такими як системні збої та порушення. Загалом, наявність належним чином впровадженої інфраструктури може бути фактором, що впливає на те, чи буде бізнес прибутковим чи ні.

Завдяки ІТ-інфраструктурі компанія може:

- Забезпечити позитивний клієнтський досвід, надаючи безперервний доступ до свого веб-сайту та інтернет-магазину.
- Швидко розробляти та виводити рішення на ринок.
- Збирати дані в режимі реального часу для прийняття швидких рішень.
- Підвищити продуктивність працівників.

1.1 Історія еволюції IT-інфраструктури

Технології розвиваються швидко. Компанії повинні еволюціонувати в тому, як вони проектують, розвивають та експлуатують технологічну інфраструктуру. Фактом є те, що без гнучкої та адаптивної IT-інфраструктури важко керувати сучасною компанією. Коли технології, що входять до її складу, правильно налаштовані і добре об'єднані в мережу, вони можуть підвищити ефективність, значно покращуючи багато процесів, але якщо це не так, вони стають бар'єром для розширення бізнесу і прогресу.

IT-інфраструктура в сучасних організаціях - це результат 50-річної еволюції та розвитку систем обробки, зберігання та передачі даних. Перш ніж знати, як вона має виглядати, важливо зрозуміти, як вона еволюціонувала протягом цих років.

Виділяють сім важливих періодів, які, сформували еволюцію обладнання, програмного забезпечення, мереж та даних, що разом складають IT-інфраструктуру.

1959: Період мейнфреймів та міні-комп'ютерів

Поява машин IBM 1401 та 7090. Початок масового комерційного використання мейнфреймів.

1981: Період персональних комп'ютерів

Ера персональних комп'ютерів почалася з появою IBM PC в північноамериканських компаніях, а також Wintel PC, стандартного настільного комп'ютера з ОС Windows.

1983: Період клієнт/сервер

У компаній з'являється можливість розподіляти обчислювальну роботу на менших і дешевших машинах, ніж централізовані мейнфрейми (міні-комп'ютери). На початку цього періоду мережева операційна система Novell NetWare була ключовою для мереж клієнт/сервер.

1992: Період корпоративних обчислень

На початку 1990-х років на рівні підприємств виникла потреба в мережових і програмних стандартах, здатних інтегрувати несумісні мережі і додатки в одну інфраструктуру.

2000: Період хмарних і мобільних обчислень

З'явилася обчислювальна модель, здатна забезпечити доступ до спільних ресурсів і сховища даних через мережу. Це означає доступ до «хмар» з будь-якої точки світу і з будь-якого пристрою, підключеного до мережі. Особливо виділяються Amazon, Hewlett-Packard, IBM і Dell.

2010: Період зростання та експансії

IT-інфраструктура стає технологічним драйвером для важливих секторів економіки. Її потужність постійно зростає, особливо коли компанії усвідомлюють і розуміють, що їхнім найважливішим ресурсом є інформація, і її захист стає критично важливим.

1.2 Взаємодія апаратного та програмного забезпечення в IT-інфраструктурі: роль операційної системи та мережових компонентів

Компоненти IT-інфраструктури складаються з взаємозалежних елементів, і дві основні групи компонентів - це апаратне та програмне забезпечення. Апаратне забезпечення використовує програмне забезпечення - наприклад, операційну систему - для роботи. Так само операційна система керує системними ресурсами та апаратним забезпеченням. Операційні системи також з'єднують програмні додатки та фізичні ресурси за допомогою мережових компонентів.



Рис 1.1 Основні компоненти ІТ-інфраструктури

Апаратне забезпечення

Апаратні компоненти можуть включати в себе:

- Апаратні компоненти можуть включати в себе
- Настільні комп'ютери
- Сервери
- Центри обробки даних
- Хаби
- Маршрутизатори
- Комутатори
- Приміщення

Програмне забезпечення

Програмні компоненти можуть включати в себе:

- Системи управління контентом (CMS)
- Управління взаємовідносинами з клієнтами (CRM)
- Планування ресурсів підприємства (ERP)
- Операційні системи

– Веб-сервери

Приміщення

Приміщення або фізичні об'єкти забезпечують простір для мережевого обладнання, серверів та центрів обробки даних. Вони також включають мережеві кабелі в офісних будівлях для з'єднання компонентів ІТ-інфраструктури між собою.

Мережа

Мережі складаються з комутаторів, маршрутизаторів, концентраторів і серверів. Комутатори з'єднують мережеві пристрої, такі як маршрутизатори, сервери та інші в локальних мережах (LAN). Маршрутизатори дозволяють пристроям у різних локальних мережах обмінюватися даними та переміщувати пакети між мережами. Концентратори з'єднують кілька мережевих пристроїв в єдине ціле.

Сервер

Основним апаратним компонентом, необхідним для ІТ-інфраструктури підприємства, є сервер. Сервери - це, по суті, комп'ютери, які дозволяють декільком користувачам отримувати доступ до ресурсів і спільно використовувати їх.

Серверна кімната/центр обробки даних

Організації розміщують кілька серверів у приміщеннях, які називаються серверними або центрами обробки даних. Центри обробки даних є ядром більшості мереж.

1.3 Типи ІТ-інфраструктури

Всього виділяють два основних типи інфраструктури - традиційна(власна) та хмарна. Також окремо виділяють комбінацію власної та хмарної ІТ-інфраструктур, а саме гібридну.

Традиційна інфраструктура складається з апаратних і програмних компонентів: комп'ютерів, серверів, дата-центрів, маршрутизаторів, комутаторів і різного роду програмного забезпечення. Зазвичай вона розташована безпосередньо

в офісі компанії. За конфігурацію та адміністрування відповідає власний або залучений системний адміністратор. Як правило, створення такої інфраструктури вимагає більше ресурсів, фізичного простору та витрат порівняно з іншими типами інфраструктури.

Хмарна ІТ-інфраструктура означає, що кінцеві користувачі можуть отримати доступ до інфраструктури через Інтернет і використовувати обчислювальні ресурси без встановлення локальних компонентів. Технологія віртуалізації з'єднує фізичні сервери незалежно від їхнього географічного розташування. Це дозволяє користувачам отримувати доступ до даних з будь-якого місця. Завдяки хмарній інфраструктурі (IaaS) клієнти можуть миттєво підключати потрібні їм послуги і відключати ті, які більше не потрібні. Капітальні витрати на придбання та обслуговування обладнання бере на себе провайдер. Таким чином, ви платите лише за потужність, необхідну для роботи вашої ІТ-системи. Ресурси хмарних провайдерів розміщені в захищених дата-центрах з високим рівнем безпеки. ІТ-спеціалісти постійно перевіряють системи на вразливості та забезпечують надійний захист інформації. Тому хмара є хорошим вибором для компаній, які не можуть дозволити собі наймати професійний ІТ-персонал та встановлювати дороге обладнання та програмне забезпечення.

Гібридна ІТ-інфраструктура представляє собою комбінацію традиційної локальної і хмарної ІТ-інфраструктур. Вона дозволяє організаціям поєднати переваги обох підходів та оптимізувати роботу з даними та інфраструктурою. У гібридній інфраструктурі окремі компоненти (сервери, додатки, бази даних) можуть бути розміщені як в межах організації, так і на зовнішніх хмарних платформах. Це дозволяє зберігати критичні дані та додатки власному контролю, тоді як менш критичні частини можуть бути розміщені у хмарних сервісах, забезпечуючи масштабованість та гнучкість. Гібридна інфраструктура дозволяє організаціям максимально використовувати наявні ресурси та збільшувати ефективність використання обладнання та інфраструктури. Крім того, вона забезпечує більшу надійність та безпеку, оскільки критичні дані можуть бути збережені локально, а менш критичні дані можуть бути збережені у хмарі.

2 ОГЛЯД ТЕХНІЧНИХ ЗАСОБІВ

2.1 Огляд AWS

Amazon Web Services, Inc. (AWS) - це дочірня компанія Amazon, яка надає платформи хмарних обчислень на вимогу та API приватним особам, компаніям та урядам на платній основі. Клієнти часто використовують їх у поєднанні з автомасштабуванням (процес, який дозволяє клієнту використовувати більше обчислень під час високого навантаження на додаток, а потім зменшувати масштаби, щоб зменшити витрати, коли трафік зменшується). Ці веб-сервіси хмарних обчислень надають різні послуги, пов'язані з мережею, обчисленнями, сховищем, проміжним програмним забезпеченням, Інтернетом речей та іншими обчислювальними потужностями, а також програмними інструментами через ферми серверів AWS. Це звільняє клієнтів від управління, масштабування та виправлення апаратного забезпечення та операційних систем. Однією з основних послуг є Amazon Elastic Compute Cloud (EC2), яка дозволяє користувачам мати у своєму розпорядженні віртуальний кластер комп'ютерів з надзвичайно високою доступністю, з яким можна взаємодіяти через Інтернет за допомогою REST API, CLI або консолі AWS. Віртуальні комп'ютери AWS емулюють більшість атрибутів реального комп'ютера, включаючи апаратні центральні процесори (CPU) та графічні процесори (GPU) для обробки; локальну/оперативну пам'ять; жорсткий диск (HDD)/SSD; вибір операційних систем; мережу; та попередньо завантажене прикладне програмне забезпечення, таке як веб-сервери, бази даних та системи управління відносинами з клієнтами (CRM).

Послуги AWS надаються клієнтам через мережу серверних ферм AWS, розташованих по всьому світу. Плата базується на поєднанні використання (відомого як «модель оплати за фактом»), апаратного забезпечення, операційної системи, програмного забезпечення або мережевих функцій, обраних абонентом, з

урахуванням вимог до доступності, резервування, безпеки та сервісних опцій. Абоненти можуть платити за один віртуальний комп'ютер AWS, виділений фізичний комп'ютер або кластери будь-якого з них. Amazon забезпечує окремі частини безпеки для абонентів (наприклад, фізичну безпеку центрів обробки даних), тоді як інші аспекти безпеки є відповідальністю абонента (наприклад, управління обліковими записами, сканування вразливостей, виправлення). AWS працює з багатьох глобальних географічних регіонів, включаючи сім у Північній Америці.

Amazon пропонує AWS підписникам як спосіб отримати великомасштабні обчислювальні потужності швидше і дешевше, ніж будувати власну фізичну серверну ферму. Всі послуги виставляються на основі використання, але кожна служба вимірює використання по-різному. Станом на 1 квартал 2024 року AWS має 33% частку ринку хмарної інфраструктури, тоді як наступні два конкуренти, Microsoft Azure та Google Cloud, мають 21% та 10% відповідно.

Станом на березень 2024 року AWS працює в 33 географічних «регіонах»: вісім у Північній Америці, один у Південній Америці, вісім у Європі, три на Близькому Сході, один в Африці та дванадцять в Азіатсько-Тихоокеанському регіоні.

Більшість регіонів AWS увімкнено за замовчуванням для акаунтів AWS. Регіони, додані після 20 березня 2019 року, вважаються регіонами, що вмикаються, і вимагають від користувача явного увімкнення для того, щоб регіон можна було використовувати в акаунті. Для увімкнених регіонів ресурси керування ідентифікацією та доступом (IAM), такі як користувачі та ролі, поширюються лише на увімкнені регіони.

Кожен регіон повністю знаходиться в межах однієї країни, і всі його дані та послуги залишаються в межах визначеного регіону. Кожен регіон має кілька «зон доступності», які складаються з одного або декількох окремих центрів обробки даних, кожен з яких має резервне живлення, мережу та можливість підключення, розміщених в окремих приміщеннях. Зони доступності автоматично не забезпечують додаткову масштабованість або надлишковість в межах регіону,

оскільки вони навмисно ізольовані одна від одної, щоб запобігти поширенню збоїв між зонами. Деякі сервіси можуть працювати в різних зонах доступності (наприклад, S3, DynamoDB), тоді як інші можуть бути налаштовані на реплікацію між зонами, щоб розподілити попит і уникнути простоїв через збої.

Станом на грудень 2014 року Amazon Web Services експлуатувала приблизно 1,4 мільйона серверів в 11 регіонах і 28 зонах доступності. Глобальна мережа місць розташування AWS Edge складається з понад 300 точок присутності по всьому світу, включаючи місця розташування в Північній Америці, Європі, Азії, Австралії, Африці та Південній Америці.

Станом на березень 2024 року AWS оголосила про запланований запуск шести додаткових регіонів у Малайзії, Мексиці, Новій Зеландії, Таїланді, Саудівській Аравії та Європейському Союзі. У середині березня 2023 року Amazon Web Services підписала угоду про співпрацю з урядом Нової Зеландії щодо будівництва великих центрів обробки даних у Новій Зеландії.

2.2 Огляд EC2

Amazon Elastic Compute Cloud (EC2) - це популярний веб-сервіс від Amazon Web Services (AWS), який дозволяє створювати та керувати віртуальними машинами, відомими як інстанси, в інфраструктурі AWS. EC2 надає масштабовані обчислювальні потужності в хмарі, які можна змінювати та масштабувати.

Інновація EC2:

- Amazon автоматизував процес забезпечення, усунувши необхідність втручання людини.
- Це змінило перспективу ІТ-індустрії, підвищивши очікування щодо доступності ресурсів і кинувши виклик традиційним підходам.
- Швидка підготовка ресурсів та відсутність витрат часу на це привабили користувачів EC2 - від приватних осіб до великих організацій.

Унікальні характеристики EC2:

- EC2 використовує віртуалізацію для створення віртуальних машин («інстансів») з операційними системами.
- На відміну від традиційної віртуалізації, EC2 зберігає лише необхідні дані, коли екземпляри не працюють, що зменшує витрати.
- Він підтримує два типи образів машин Amazon (AMI): S3 та EBS, кожен з яких має специфічні особливості зберігання.

Архітектура та керування додатками EC2:

- Успіх роботи з EC2 залежить від розуміння його унікальних характеристик і розробки додатків відповідно до них.
- Додатки повинні використовувати можливості EC2 для оптимізації продуктивності та економічної ефективності.

Переваги EC2

Це дозволяє користувачам швидко створювати, налаштовувати та запускати віртуальні сервери, відомі як екземпляри EC2, для запуску додатків і сервісів.

- Ви можете обирати з широкого спектру типів екземплярів залежно від ваших конкретних вимог, таких як обчислювальна потужність, пам'ять і сховище.
- Екземпляри EC2 можна легко масштабувати за потребою, сплачуючи лише за ті ресурси, які вам потрібні.
- Ви маєте повний контроль над екземпляром, включаючи конфігурацію операційної системи та програмного забезпечення.
- Доступ до екземплярів EC2 можна отримати віддалено, підключаючись до них та керуючи ними через Інтернет.

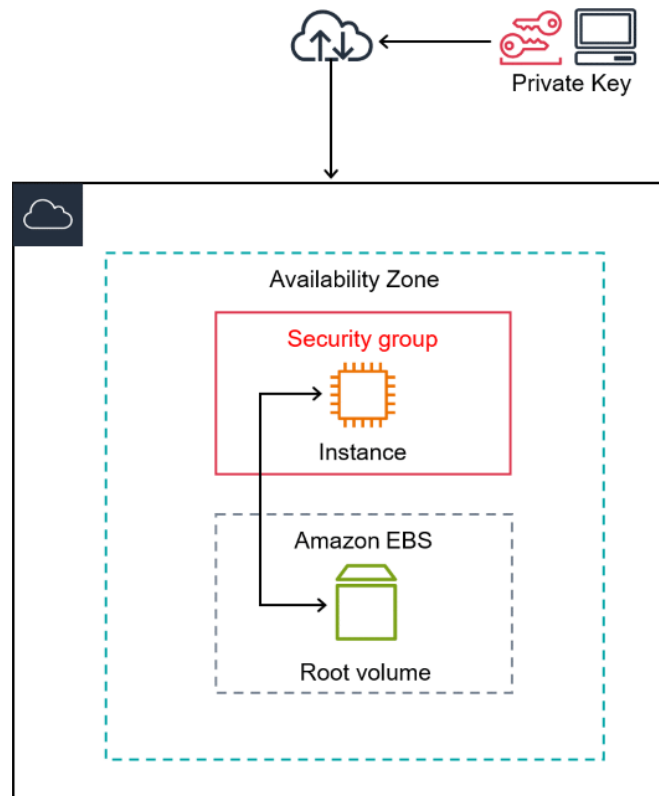


Рис 2.1 Схема основних компонентів EC2

Amazon machine images (AMIs)

Коли справа доходить до встановлення операційної системи (ОС) на інстанс EC2, першим кроком є наявність завантажувального образу. AMI (Amazon Machine Images) - це попередньо налаштовані шаблони з операційною системою, програмами та програмним забезпеченням, необхідними для запуску інстансу EC2.

- AMI використовуються як основа для запуску інстансів EC2.
- Цей спрощений процес усуває необхідність ручного встановлення операційної системи та спрощує налаштування ваших інстансів EC2.
- AWS пропонує широкий вибір готових AMI, включаючи різні операційні системи та конфігурації програмного забезпечення.
- Ви також можете створити свій власний AMI, почавши з існуючого AMI, змінивши його за необхідності, а потім зберігши його як новий образ.
- До AMI можна надавати спільний доступ іншим акаунтам AWS, що дозволить вам співпрацювати з іншими користувачами або поширювати ваші власні конфігурації.

- Під час запуску інстанса EC2 ви можете вибрати потрібний AMI і запустити кілька інстансів з одного AMI, якщо це необхідно.

Розміри та варіанти конфігурації EC2

- Операційні системи: EC2 підтримує широкий спектр операційних систем, включаючи такі популярні, як Amazon Linux, Ubuntu, Windows Server та інші, що забезпечує гнучкість у виборі потрібної ОС.
- Типи інстансів: EC2 пропонує широкий вибір типів інстансів з різними обчислювальними можливостями, пам'яттю, сховищем та мережевими можливостями, щоб відповідати конкретним вимогам робочих навантажень.
- Сімейства інстансів: Інстанси згруповані в сімейства на основі їх передбачуваного використання, наприклад, загального призначення, оптимізовані для пам'яті, оптимізовані для обчислень, оптимізовані для зберігання і т.д.
- Розміри інстансів: Кожне сімейство інстансів включає кілька варіантів розмірів з різним розподілом ресурсів, що дозволяє вибрати правильний баланс процесора, пам'яті та сховища для вашого додатку.
- Моделі ціноутворення: EC2 надає можливість використовувати екземпляри на вимогу (оплата за фактом використання), зарезервовані інстанси (авансовий платіж за довгострокове використання) та спотові інстанси (ціноутворення на основі заявок) для оптимізації економічної ефективності.
- Варіанти зберігання: Інстанси EC2 пропонують різні варіанти зберігання, включаючи сховище інстансів (ефемерне сховище), томи Elastic Block Store (EBS) та Amazon S3 для зберігання об'єктів.
- Мережа та безпека: Інстанси EC2 можна розгортати в середовищі віртуальної приватної хмари (VPC), що дозволяє налаштовувати мережеві конфігурації, розміщувати підмережі, створювати групи безпеки та контролювати доступ.
- Автоматичне масштабування: EC2 Auto Scaling дозволяє динамічно масштабувати інстанси на основі попиту додатків, забезпечуючи оптимальну

продуктивність і економічну ефективність шляхом автоматичного регулювання кількості інстансів.

- Балансування навантаження: Еластичне балансування навантаження (ELB) розподіляє вхідний трафік між декількома інстансами, підвищуючи доступність, масштабованість і відмовостійкість ваших додатків.
- Метадані інстансів: EC2 надає метадані інстансів, доступні зсередини інстансу, що дозволяє програмам отримувати цінну інформацію про конфігурацію, мережу та атрибути інстансу.

Еластичне сховище блоків (EBS)

Elastic Block Store (EBS) - це сервіс блочного зберігання даних на базі хмари AWS. EBS зберігає величезні обсяги даних у блоках, які працюють як жорсткі диски (так звані томи). Можна використовувати його для зберігання будь-яких типів даних, включаючи файлові системи, транзакційні дані, NoSQL та реляційні бази даних, резервні копії, контейнери та додатки.

EBS пропонується через платформу AWS і вимагає наявності облікового запису AWS. Платформа дозволяє оптимізувати продуктивність і ціни. Наприклад, ви можете змінити розмір і тип тома, а також видалити надлишкові томи. На платформі можна налаштувати параметри резервного копіювання та відновлення даних.

Деякі параметри резервного копіювання:

- Реплікація EBS - EBS автоматично реплікує томи в межах зони доступності (AZ). Якщо щось трапиться з томом, то буде копія для відновлення.
- Резервне копіювання AWS - це платна послуга, яку пропонує AWS за додаткову плату.
- Знімки EBS - миттєві версії ваших даних, які слугують додатковими резервними копіями.

Том EBS - це пристрій, який працює як жорсткий диск. Можна використовувати томи EBS для створення файлових систем, зберігання баз даних або запуску програм.

Найпоширеніші випадки використання томів EBS:

- Часті оновлення - зберігання даних, які потребують частих оновлень. Наприклад: програми баз даних і системні диски екземплярів.
- Високопродуктивні програми, які потребують безперервного сканування диска.
- Інстанси EC2 - після приєднання тому EBS до екземпляра EC2, том EBS виконує функцію фізичного жорсткого диска.

EBS спочатку було створено для використання з екземплярами EC2 і часто автоматично налаштовується як вибір за замовчуванням для екземплярів EC2 нижчого рівня. Однак томи EBS працюють незалежно від EC2, і можна використовувати EBS як основне сховище. EBS забезпечує високу доступність і довговічність та ідеально підходить для інтенсивних додатків.

Типи томів EBS

Продуктивність і ціна вашого сховища EBS визначатимуться типом обраних вами томів.

Amazon EBS пропонує чотири типи томів, які виконують різні функції.

Томи на основі SSD-дисків

- SSD загального призначення (gp2) - том EBS за замовчуванням, сконфігурований для забезпечення максимально можливої продуктивності за найнижчу ціну. Рекомендується для інтерактивних додатків з низькою затримкою, а також для операцій розробки та тестування.
- SSD з виділеним IOPS (io1) - налаштований для забезпечення високої продуктивності для критично важливих додатків. Ідеально підходить для

баз даних NoSQL, реляційних навантажень з інтенсивним введенням/виведенням і робочих навантажень додатків.

Томи на основі жорстких дисків (HDD)

- Жорсткий диск з оптимізованою пропускною здатністю (st1) - це недороге магнітне сховище. Рекомендується для великих послідовних робочих навантажень, які визначають продуктивність у пропускній здатності.
- « Cold HDD» (sc1) - для регулювання ємності використовується стрибкоподібна модель, що забезпечує найдешевше магнітне сховище. Ідеально підходить для холодних великих послідовних робочих навантажень.

Групи безпеки

- Групи безпеки мають вирішальне значення для мережевої безпеки в AWS.
- Вони регулюють вхідний і вихідний трафік для інстансів EC2.
- Групи безпеки засновані на правилах і не містять жодних фактичних даних.
- Правила в групах безпеки можуть посилатися на IP-адреси або інші групи безпеки.

Security Groups Diagram

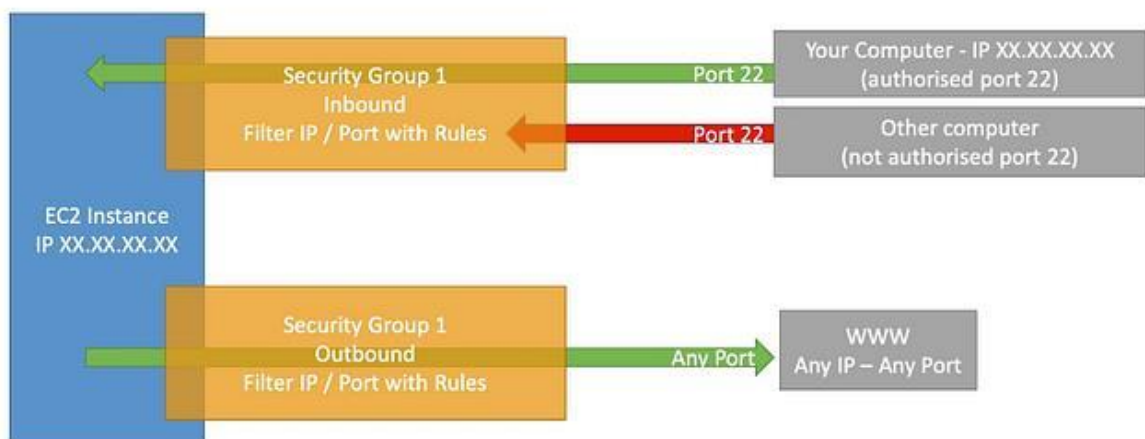


Рис 2.2 Схема принципу роботи групи безпеки

1. Гнучкість приєднання: Групи безпеки можуть бути приєднані до декількох інстансів для узгодженої політики безпеки.
2. Залежність від регіону та VPC: Групи безпеки обмежуються певними регіонами та комбінаціями VPC.
3. Зовнішній вплив: Групи безпеки працюють за межами інстансів EC2, блокуючи трафік, який інстанс не бачить.
4. Окремий захист SSH: Рекомендується підтримувати окрему групу безпеки для доступу по SSH.
5. Підказки щодо усунення несправностей: Тайм-аут вказує на проблеми з групою безпеки, в той час як «відмова у з'єднанні» вказує на помилки програми або проблеми із запуском.
6. Поведінка трафіку за замовчуванням: За замовчуванням вхідний трафік заблоковано, а вихідний - дозволено.

Дані користувача

- Користувацькі дані - це інформація, надана користувачем, яка передається інстансам EC2 під час їх запуску.
- Вони можуть бути у вигляді скриптів, конфігураційних файлів або простого текстового введення.
- Користувацькі дані дозволяють налаштовувати та конфігурувати інстанси відповідно до конкретних вимог.
- Під час запуску інстансу надані користувацькі дані виконуються автоматично.
- Сценарії User Data можна використовувати для встановлення програмних пакетів, конфігурації служб, налаштування користувацьких параметрів або виконання будь-яких інших необхідних завдань для підготовки інстансу.
- Дані користувача зазвичай використовуються для конфігурації системи, розгортання додатків та автоматизації налаштування інстансів.

2.3 Віртуальна приватна хмара (VPC)

Amazon Web Services (AWS) пропонує віртуальну приватну хмару (VPC) як спосіб створити віртуальну мережу в хмарі. У VPC ви можете створювати публічні та приватні підмережі для контролю мережевого трафіку та забезпечення безпеки ваших ресурсів.

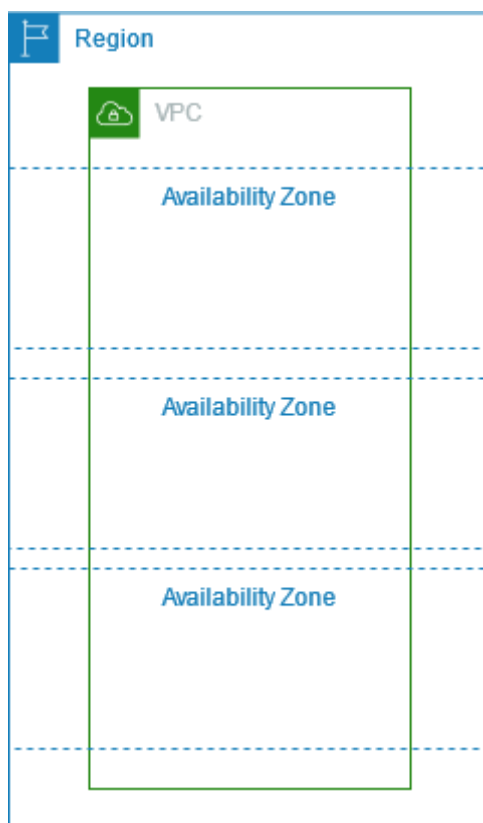


Рис 2.3 Схеми VPC в AWS

Підмережа у VPC: Підмережа - це діапазон IP-адрес у мережі VPC. IP-адреси в підмережі можна використовувати для запуску ресурсів, таких як інстанси EC2, інстанси RDS тощо. У межах VPC можна створити кілька підмереж, пов'язаних з різними зонами доступності (AZ) в регіоні. Це забезпечує високу доступність і відмовостійкість ваших ресурсів.

Публічна підмережа: Загальнодоступна підмережа - це підмережа у VPC, яка має прямий шлях до Інтернету через Інтернет-шлюз (IGW). Ресурси, запущені в загальнодоступній підмережі, мають загальнодоступну IP-адресу і можуть безпосередньо зв'язуватися з Інтернетом. Загальнодоступні підмережі зазвичай

використовуються для ресурсів, які потребують доступу з інтернету, наприклад, веб-серверів або балансувальників навантаження.

Приватна підмережа: Приватна підмережа - це підмережа у VPC, яка не має прямого шляху до інтернету. Ресурси, запущені в приватній підмережі, не мають публічної IP-адреси і не можуть безпосередньо зв'язуватися з Інтернетом. Однак вони можуть взаємодіяти з ресурсами в інших підмережах у тому ж VPC або з іншими мережами, підключеними до вашого VPC за допомогою віртуальної приватної мережі (VPN) або AWS Direct Connect. Приватні підмережі зазвичай використовуються для ресурсів, які не потребують доступу з інтернету, наприклад, баз даних або серверів додатків.

Переваги використання публічних та приватних підмереж: Використання публічних та приватних підмереж у VPC надає декілька переваг, а саме:

1. **Безпека:** Використовуючи публічні та приватні підмережі, ви можете створити багаторівневу модель безпеки для захисту ваших ресурсів. Загальнодоступні підмережі відкриті для Інтернету і більш вразливі до атак, в той час як приватні підмережі ізольовані і менш схильні до зовнішніх загроз.
2. **Оптимізація витрат:** Використання публічних і приватних підмереж може допомогти оптимізувати витрати. Наприклад, ви можете запустити ресурси, які потребують доступу з Інтернету, у публічній підмережі, а ресурси, які не потребують доступу з Інтернету, - у приватній підмережі. Таким чином, ви зможете уникнути зайвих витрат на передачу даних і зменшити загальні витрати.
3. **Масштабованість:** Публічні та приватні підмережі дозволяють більш ефективно масштабувати ресурси у VPC. Ви можете запускати нові ресурси в публічних або приватних підмережах і підключати їх до інших ресурсів в межах VPC або зовнішніх мереж за допомогою VPN або Direct Connect.

2.4 Amazon Simple Storage (S3)

Це хмарний сервіс зберігання об'єктів від AWS, який забезпечує високомасштабоване, безпечне та довговічне зберігання об'єктів і файлів. Це сервіс, за допомогою якого ви можете зберігати та отримувати файли, документи, зображення, відео та інші типи даних. Це онлайн-жорсткий диск, на який ви можете завантажувати та вивантажувати файли з будь-якого місця та в будь-який час, використовуючи підключення до Інтернету. Amazon S3 - це економічно ефективно рішення для зберігання та резервного копіювання даних, яке автоматично масштабується зі збільшенням обсягу даних. AWS рекламує його як рішення для зберігання даних з «нескінченим масштабуванням». S3 відомий своєю масштабованістю, довговічністю та безпекою. Ви можете зберігати практично необмежену кількість даних, і він розроблений для забезпечення довговічності об'єктів на рівні 99.999999999% . Це означає, що якщо ви зберігаєте 10 000 000 об'єктів у S3, ви можете розраховувати на втрату одного об'єкта в середньому раз на 10 000 років. Така висока довговічність робить ваші дані надійно захищеними та доступними, коли вони вам знадобляться. Amazon S3 широко використовується для різноманітних завдань, таких як резервне копіювання та аварійне відновлення, аналітика великих даних, доставка контенту, озера даних, медіа-хостинг і статичний хостинг веб-сайтів. Він простий у використанні та легко інтегрується з іншими сервісами AWS, що робить його потужним інструментом для бізнесу будь-якого розміру.

В Amazon S3 дані зберігаються в контейнерах, які називаються «кошиками», і ви не можете зберігати будь-які дані в S3, не створивши попередньо «кошик». Кожне відро має глобально унікальне ім'я (унікальне для всіх регіонів AWS і акаунтів в одному розділі), і ви можете створити кілька відер, щоб організувати свої дані різними способами. При створенні кошика необхідно вказати дві деталі: ім'я кошика і регіон, в якому ви хочете створити кошик, тому що кошики визначаються на регіональному рівні.

Існують домовленості (правила) щодо іменування S3-відер. Правила іменування такі:

- Назви повинні мати довжину від 3 до 63 символів.
- Назви можуть складатися лише з малих літер, цифр, крапок і дефісів. Отже, назви відер не можуть складатися з великих літер і символів підкреслення.
- Вони повинні починатися і закінчуватися на літеру або цифру.
- Назви контейнерів не можуть бути IP-адресами.
- Вони не повинні починатися з префікса `xp` - або закінчуватися суфіксом - `s3alias`
- І, як ми вже бачили, ім'я кошика має бути глобально унікальним.

S3 - це хмарний сервіс зберігання об'єктів. Це означає, що в S3 дані зберігаються у вигляді об'єктів. Об'єкт - це основна одиниця зберігання даних, і він являє собою файл, який може бути будь-яким типом даних: текстовий файл, зображення, відео, резервна копія бази даних або будь-який інший тип файлу. Кожен об'єкт у S3-відрі унікально ідентифікується за допомогою ключа, який являє собою рядок, що може мати довжину до 1024 символів. Ключ слугує ім'ям об'єкта, а також забезпечує спосіб упорядкування та пошуку об'єктів у кошику. Окремі об'єкти, що зберігаються в S3, можуть мати розмір до 5 ТБ. Важливо зазначити, що в S3 об'єкти зберігаються в пласкій структурі, тобто немає власне папок або каталогів, як у традиційній файлової системі. Однак S3 дозволяє створювати логічну ієрархію об'єктів за допомогою префіксів у ключах об'єктів. Наприклад, якщо ви завантажите об'єкт з ключем «папка/підпапка/об'єкт.txt», S3 створить префікс «папка/підпапка/», який поводитиметься як папка або каталог. Ви можете використовувати консоль управління AWS, AWS CLI або SDK для створення і керування цими префіксами, як каталогами.

S3 Security

Безпека будь-якого ІТ-ресурсу або послуги дуже важлива, і до неї потрібно ставитися серйозно в будь-який момент часу. Amazon S3 має різні способи забезпечення безпеки. Ви можете забезпечити безпеку на основі користувачів, використовуючи політики IAM, або на основі ресурсів, використовуючи: bucket-політики, об'єктні списки контролю доступу (ACL) та списки контролю доступу до об'єктів (bucket Access Control Lists).

Політики S3-бакетів

Політики S3 - це документи у форматі JSON, які визначають дозволи для S3-відер та їхнього вмісту (тобто об'єктів у них). Вони дозволяють визначити набір правил, які дозволяють або забороняють доступ до S3-бакетів та їх вмісту для певних користувачів, груп або навіть для загального доступу. Їх можна використовувати для керування різними діями, які можна виконати над S3-відром, наприклад, читання, запис або видалення. Ці політики можуть застосовуватися на рівні кошика або на рівні об'єктів всередині кошика, що дає змогу детально контролювати доступ до даних. Нижче наведено приклад типової політики для S3-відерця:

```
{
  "Version": "2024-04-24",
  "Statement": [
    {
      "Sid": "AllowGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::test/*"
    }
  ]
}
```

Наведена вище політика дозволяє будь-кому отримувати об'єкти (файли) з S3-відра «test». Поле Principal має значення «*», щоб вказати, що будь-який обліковий запис або користувач AWS може виконати дію s3:GetObject. У полі Resource вказується ARN (Amazon Resource Name - ім'я ресурсу AWS) для кошика та його об'єктів, а у полі Sid - ім'я для ідентифікації політики.

S3 Версії

Це функція Amazon S3, яка дозволяє зберігати кілька версій об'єкта в одному кошику. Версії - це параметр, який вмикається на рівні папки. Коли керування версіями ввімкнене для кошика, будь-який об'єкт, завантажений до цього кошика, матиме унікальний ідентифікатор версії, пов'язаний з ним. Можна завантажити нову версію об'єкта, просто завантаживши файл з тим самим ключем (назвою), що й у наявного об'єкта. Кожна версія об'єкта матиме унікальний ідентифікатор версії, і ви зможете отримати доступ до всіх версій об'єкта та керувати ними за допомогою S3 API або консолі управління AWS. Якщо версійність увімкнено після завантаження об'єкта, ця версія об'єкта (тобто об'єкт, завантажений до увімкнення версійності) матиме ідентифікатор версії «null». Найкраще ввімкнути керування версіями для ваших S3-відер, оскільки це допомагає запобігти ненавмисному видаленню, надаючи вам можливість відновити видалені об'єкти, а також дозволяє відкотитися до попередньої версії об'єкта. У будь-який момент часу S3-зок може перебувати в одному з трьох станів: неверсіонований (за замовчуванням), з увімкненою версією або з призупиненою версією.

Реплікація S3

Це функція S3, яка дозволяє автоматично реплікувати об'єкти з одного S3-відра до іншого в тому ж або іншому регіоні AWS. Реплікація допомагає підвищити довговічність і доступність даних, а також дозволяє відповідати нормативним вимогам до реплікації даних і аварійного відновлення. Щоб виконати реплікацію S3, як на вихідному, так і на цільовому S3-бакетах має бути ввімкнена підтримка версій, оскільки реплікація покладається на наявність ідентифікаторів версій, щоб визначити, які об'єкти змінилися і потребують реплікації. S3 підтримує реплікацію в одному регіоні (SRR) і міжрегіональну реплікацію (CRR).

Міжрегіональна реплікація використовується для автоматичної та асинхронної реплікації об'єктів між різними регіонами AWS. За допомогою міжрегіональної реплікації ви можете створити репліку об'єктів S3 в іншому регіоні, ніж вихідний, для резервування даних, аварійного відновлення та зменшення затримок доступу до даних.

Реплікація в один регіон дозволяє автоматично й асинхронно реплікувати об'єкти між кошиками, які знаходяться в одному регіоні AWS. SRR допомагає підвищити довговічність і доступність даних, створюючи точну копію об'єктів в іншому кошику в тому ж регіоні. Це може бути корисно для різних випадків використання, таких як створення резервних копій для аварійного відновлення, розподіл контенту в декількох місцях для кращого часу доступу або дотримання нормативних вимог, які вимагають надмірності даних.

2.5 Amazon RDS

Amazon RDS - це реляційна база даних, яка використовує SQL як мову запитів Query. Amazon RDS дозволяє керувати Postgres, MySQL, MariaDB, Oracle, Microsoft SQL Server та Aurora (власною базою даних AWS) у хмарі.

Звичайно базу даних можна розвернути і на інстансі EC2, але дуже переваг має RDS відносно EC2, а саме:

1. Керований сервіс: Amazon RDS - це повністю керований сервіс, що означає, що AWS займається різними аспектами адміністрування баз даних, такими як автоматизоване забезпечення, виправлення ОС, резервне копіювання та оновлення програмного забезпечення. Це звільняє вас від багатьох рутинних завдань з обслуговування, дозволяючи зосередитися на розробці вашого додатку та бізнес-логіці.
2. Автоматичне резервне копіювання та відновлення в потрібний момент: RDS автоматично виконує регулярне резервне копіювання і дозволяє відновлювати базу даних до певного моменту часу в межах періоду

зберігання. Це допомагає захистити від втрати даних і спрощує відновлення після збоїв.

3. Висока доступність за допомогою Multi-AZ: RDS пропонує розгортання Multi-AZ, при якому база даних автоматично реплікується в декількох зонах доступності. Якщо основна база даних стає недоступною, RDS автоматично перемикається на резервний екземпляр в іншій зоні доступності, забезпечуючи кращу доступність і надлишковість даних.
4. Репліки для читання: RDS дозволяє створювати репліки бази даних для читання, щоб розвантажити трафік читання з основного екземпляра. Це підвищує продуктивність читання і знижує навантаження на основний екземпляр.
5. Масштабованість: RDS підтримує як вертикальне, так і горизонтальне масштабування. Вертикальне масштабування передбачає збільшення розміру екземпляра, в той час як горизонтальне масштабування передбачає додавання реплік для читання. AWS впорається зі складнощами масштабування базової інфраструктури за вас.
6. Моніторинг та метрики: RDS надає комплексні інформаційні панелі моніторингу та звіти про продуктивність, які допоможуть вам відстежувати продуктивність бази даних і виявляти потенційні проблеми на випередження.
7. Безпека та відповідність вимогам: Amazon RDS надає різні функції безпеки, такі як шифрування в стані спокою та під час передачі, ізоляція мережі та контроль доступу на основі IAM. Він також забезпечує відповідність галузевим стандартам, що полегшує дотримання нормативних вимог.
8. Вікна резервного копіювання та обслуговування: RDS дозволяє визначати вікна для резервного копіювання та технічного обслуговування, гарантуючи, що рутинні завдання не порушать працездатність вашого додатку.
9. Спрощене налаштування та конфігурація: RDS спрощує початкове налаштування та конфігурацію бази даних, заощаджуючи час і зусилля порівняно з ручним налаштуванням та конфігурацією бази даних на екземплярах EC2.

RDS (Amazon Relational Database Service) пропонує функцію автоматичного масштабування сховища, яка дозволяє динамічно збільшувати обсяг сховища у вашому інстансі бази даних RDS. Вона автоматизує процес масштабування сховища бази даних, усуваючи необхідність ручного втручання і гарантуючи, що ваш додаток зможе впоратися з підвищеними вимогами до зберігання даних.

Ось ключові аспекти автоматичного масштабування сховища RDS:

Автоматичне масштабування: RDS постійно відстежує використання сховища вашої бази даних. Коли він виявляє, що у вас закінчується вільне місце в сховищі бази даних, він автоматично масштабує ємність сховища, щоб вмістити зростаючий обсяг даних.

Максимальний поріг зберігання: Щоб контролювати витрати і запобігти несподіваному масштабуванню, ви можете встановити максимальний поріг зберігання. Він визначає верхню межу або максимальну межу для вашого сховища баз даних. Як тільки сховище досягне цього порогу, RDS не буде масштабуватися понад цю межу.

Критерії автоматичної модифікації: Автоматичне масштабування сховища RDS враховує певні критерії перед автоматичною модифікацією сховища:

- Вільне сховище становить менше 10% від виділеного сховища.
- Стан низького обсягу сховища триває щонайменше 5 хвилин, запобігаючи непотрібному масштабуванню через тимчасові сплески.
- З моменту останньої модифікації сховища пройшло щонайменше 6 годин, що забезпечує стабільний період перед тим, як розглядати наступну дію масштабування.

Корисно для непередбачуваних робочих навантажень: Автоматичне масштабування сховища особливо корисне для додатків з непередбачуваними робочими навантаженнями або моделями зростання даних. Воно дозволяє вашій базі даних легко пристосовуватися до мінливих потреб у сховищі без ручного втручання.

Підтримка декількох механізмів керування базами даних: Автоматичне масштабування сховища доступне для всіх механізмів баз даних RDS, включаючи MariaDB, MySQL, PostgreSQL, SQL Server і Oracle. Це означає, що ви можете використовувати цю функцію незалежно від того, яку СУБД ви використовуєте в RDS.

Увімкнувши функцію автоматичного масштабування сховища в RDS, ви можете гарантувати, що ваша база даних зможе динамічно адаптуватися до мінливих вимог до сховища, що допоможе вам підтримувати оптимальну продуктивність і доступність для вашої програми без необхідності постійного моніторингу та ручних налаштувань.

Репліки читання RDS для масштабування читання

База даних RDS забезпечує до 15 реплік читання в межах або між зонами доступності AWS. І ці репліки зчитування з'єднуються з головною базою даних, що в кінцевому підсумку узгоджуються.

Припустимо додаток має великий трафік, і клієнту потрібно додати функцію звітування, яка не має функції оновлення або видалення, а лише зчитує дані. Якщо використовувати без використання опції читання реплік, це означає, що нам доведеться використовувати головну базу даних для створення цієї функції також. Що впливає на загальну продуктивність програми. Там ми можемо використовувати репліку читання, щоб отримати прочитані дані для функції звітування.

При використанні Amazon RDS Read Replicas в межах одного регіону AWS ви не несете додаткових витрат на мережеву передачу даних. Це пов'язано з тим, що реплікація даних між екземплярами RDS бази даних та їх репліками для читання в межах одного регіону виконується всередині мережевої інфраструктури AWS, і вона не тарифікується як передача даних між різними зонами доступності (AZ) або регіонами.

Однак, якщо ви налаштуєте репліки читання RDS у різних регіонах AWS, реплікація даних включатиме мережеву передачу між регіонами, і ви можете понести додаткові витрати на передачу даних. AWS розглядає передачу даних між

регіонами як міжрегіональну передачу даних, яка може бути дорожчою, ніж передача даних у межах одного регіону.

RDS Multi-AZ забезпечує високу доступність і автоматичне обхід відмов для вашої бази даних RDS, використовуючи синхронну реплікацію для підтримки резервної копії в іншій зоні доступності. Це підвищує доступність додатків, автоматично обробляє обхід відмов і має вирішальне значення для аварійного відновлення. Використовуйте репліки RDS для читання в різних регіонах для додаткового резервування в сценаріях аварійного відновлення.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПОБУДОВИ ІТ ІНФРАСТРУКТУРИ У ХМАРІ

Для реалізації було використано провайдер хмарних послуг Amazon Web Services. Для початку в нашій інфраструктурі створимо окрему VPC, де буде наша мережева інфраструктура. Нижче наведено приклад створення VPC:

The screenshot shows the AWS Management Console interface for creating a new VPC. The breadcrumb navigation indicates the path: VPC > Your VPCs > Create VPC. The main heading is 'Create VPC' with an 'Info' link. Below the heading is a descriptive sentence: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.'

The 'VPC settings' section contains the following configuration options:

- Resources to create:** Two radio buttons are present. 'VPC only' is selected, and 'VPC and more' is unselected.
- Name tag - optional:** A text input field contains the value 'test_vpc'. The description states: 'Creates a tag with a key of 'Name' and a value that you specify.'
- IPv4 CIDR block:** Two radio buttons are present. 'IPv4 CIDR manual input' is selected, and 'IPAM-allocated IPv4 CIDR block' is unselected. Below this, a text input field contains '10.0.0.0/16'. A note below the field states: 'CIDR block size must be between /16 and /28.'
- IPv6 CIDR block:** Four radio buttons are present. 'No IPv6 CIDR block' is selected, and the other three options ('IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me') are unselected.
- Tenancy:** A dropdown menu is set to 'Default'.

Рис 3.1 Створення нашої VPC з назвою test_vpc

За замовчуванням створені VPC не мають доступу до мережі Інтернет. Тому було створено інтернет-шлюз (Internet Gateway) та потім підключено до VPC.

aws Services Search [Alt+S]

✓ You successfully created vpc-0520deaebeb49400c / test_vpc

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

ig_test

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Name ig_test Remove

Add new tag
You can add 49 more tags.

Cancel Create internet gateway

Рис.3.2 Створення Internet Gateway

aws Services Search [Alt+S]

VPC > Internet gateways > Attach to VPC (igw-0d382607b5c8b1120)

Attach to VPC (igw-0d382607b5c8b1120) [Info](#)

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

vpc-0520deaebeb49400c

► AWS Command Line Interface command

Cancel Attach internet gateway

Рис 3.3 Підключення Internet Gateway до VPC

Після цього треба створити наші підмережі в VPC. Дані підмережі будуть створені в 2 зонах доступності (Availability Zones) за для більш меншої затримки, стійкості до відмов та підвищеної доступності. В кожній зоні повинно бути по 3

підмережі: публічна, приватна і для бази даних. Оскільки наша VPC має набір IP-адресів 10.0.0.0/16, то всі наші підмережі будуть мати IP-адреса вигляду 10.0.zx./24, де z – номер зони доступності, а x – тип підмережі. (Приклад: 10.0.11.0/24 – набір публічної підмережі в зоні А, 10.0.23.0/24 – набір підмережі бази даних в зоні Б)

The screenshot shows the 'Subnet settings' page in the AWS console. The page title is 'Subnet settings' with a subtitle 'Specify the CIDR blocks and Availability Zone for the subnet.' Below this, it indicates 'Subnet 1 of 1'. The 'Subnet name' field is set to 'public_test_A'. The 'Availability Zone' is set to 'Europe (Stockholm) / eu-north-1a'. The 'IPv4 VPC CIDR block' is set to '10.0.0.0/16'. The 'IPv4 subnet CIDR block' is set to '10.0.11.0/24', with a note indicating '256 IPs'. There is a 'Tags - optional' section with one tag: 'Name' with value 'public_test_A'. At the bottom right, there are 'Cancel' and 'Create subnet' buttons.

Рис 3.4 Створення підмережі в VPC

<input type="checkbox"/>	public_test_A	subnet-0ca76ec517d33f798	✔ Available	vpc-0520deaebeb49400c test...	10.0.11.0/24
<input type="checkbox"/>	public_test_B	subnet-0e9b5c3767e6fec2e	✔ Available	vpc-0520deaebeb49400c test...	10.0.21.0/24
<input type="checkbox"/>	private_test_A	subnet-046f2adc05af3ca4	✔ Available	vpc-0520deaebeb49400c test...	10.0.12.0/24
<input type="checkbox"/>	private_test_B	subnet-0ead7612252a6b0c0	✔ Available	vpc-0520deaebeb49400c test...	10.0.22.0/24
<input type="checkbox"/>	db_test_A	subnet-0cb0229b7017955bb	✔ Available	vpc-0520deaebeb49400c test...	10.0.13.0/24
<input type="checkbox"/>	db_test_B	subnet-05f4194c1dc70fd7e	✔ Available	vpc-0520deaebeb49400c test...	10.0.23.0/24

Рис 3.5 Список створених підмереж

Далі треба налаштувати таблиці маршрутів(Route Tables) для підмереж. Для публічних мереж, треба додати шлях 0.0.0.0/0 для обміну трафіком через інтернет шлюз. В даному випадку наша мережа може спокійно обмінюватися трафіком в обидві сторони.

The screenshot shows the AWS Management Console interface for editing routes in a VPC. The breadcrumb navigation is: VPC > Route tables > rtb-0e952bd33e9ebb4ae > Edit routes. The main heading is 'Edit routes'. Below it, there is a table with columns for Destination, Target, and Status. The first row shows Destination '10.0.0.0/16', Target 'local', and Status 'Active'. The second row shows Destination '0.0.0.0/0', Target 'Internet Gateway', and Status '-'. There is an 'Add route' button at the bottom left.

Destination	Target	Status
10.0.0.0/16	local	✔ Active
0.0.0.0/0	Internet Gateway	-

Рис 3.6 Таблиця маршрутів публічної підмережі

Для наших приватних підмереж створимо NAT Gateway. Шлюз NAT робить щось подібне до Інтернет-шлюзу, але він працює лише в один бік: інстанси приватної підмережі можуть підключатися до сервісів за межами VPC, але зовнішні сервіси не можуть ініціювати з'єднання з цими екземплярами. Оскільки 2 різні приватні підмережі то для кожної зони доступності має бути свій власний NAT шлюз.

aws Services Search [Alt+S]

✓ Elastic IP address 16.171.76.68 (eipalloc-080c677a99208fde5) allocated.

VPC > NAT gateways > Create NAT gateway

Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.

Public
 Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.

▶ **Additional settings** [Info](#)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key: Value - optional:

You can add 49 more tags.

NAT gateways (2) [Info](#)

Find resources by attribute or tag

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...	Primary network...	VPC
<input type="radio"/> NAT_test_A	nat-0784da2be6f475572	Public	<input type="radio"/> Pending	-	-	10.0.11.9	eni-00c74d832f46a...	vpc-0520deae
<input type="radio"/> NAT_test_B	nat-04bf7ccb044f4854c	Public	<input type="radio"/> Pending	-	-	10.0.21.245	eni-081d522ff67d51...	vpc-0520deae

VPC > Route tables > rtb-08457b934f25099df > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0/16	local	<input checked="" type="radio"/> Active	No
<input type="text" value="0.0.0.0"/>	NAT Gateway	<input checked="" type="radio"/> Active	No

Рис 3.7 Створення NAT шлюзів і підключення їх до приватних підмереж.

На цьому наша побудова VPC закінчена і в результаті ми отримали, розгорнуті 6 підмереж та 4 таблиці маршрутів: 2 публічні, які можуть і приймати і відправляти трафік в інтернет, 2 приватні, які можуть тільки відправляти трафік в інтернет і 2 підмережі баз даних, які можуть відправляти трафік лише в самій VPC.

<input type="checkbox"/>	Public_route	rtb-0e952bd33e9ebb4ae	2 subnets	-	Yes	vpc-0520deaebeb49400c test...	058264357035
<input type="checkbox"/>	Private_route_A	rtb-0f4e5b4e136ff974f	subnet-04f6f2adc05af3c...	-	No	vpc-0520deaebeb49400c test...	058264357035
<input type="checkbox"/>	DB_route	rtb-0288e1c8f3a879fba	2 subnets	-	No	vpc-0520deaebeb49400c test...	058264357035
<input type="checkbox"/>	Private_route_B	rtb-08457b934f25099df	subnet-0ead7612252a6b...	-	No	vpc-0520deaebeb49400c test...	058264357035

Рис 3.8 –результат створення таблиць маршрутів

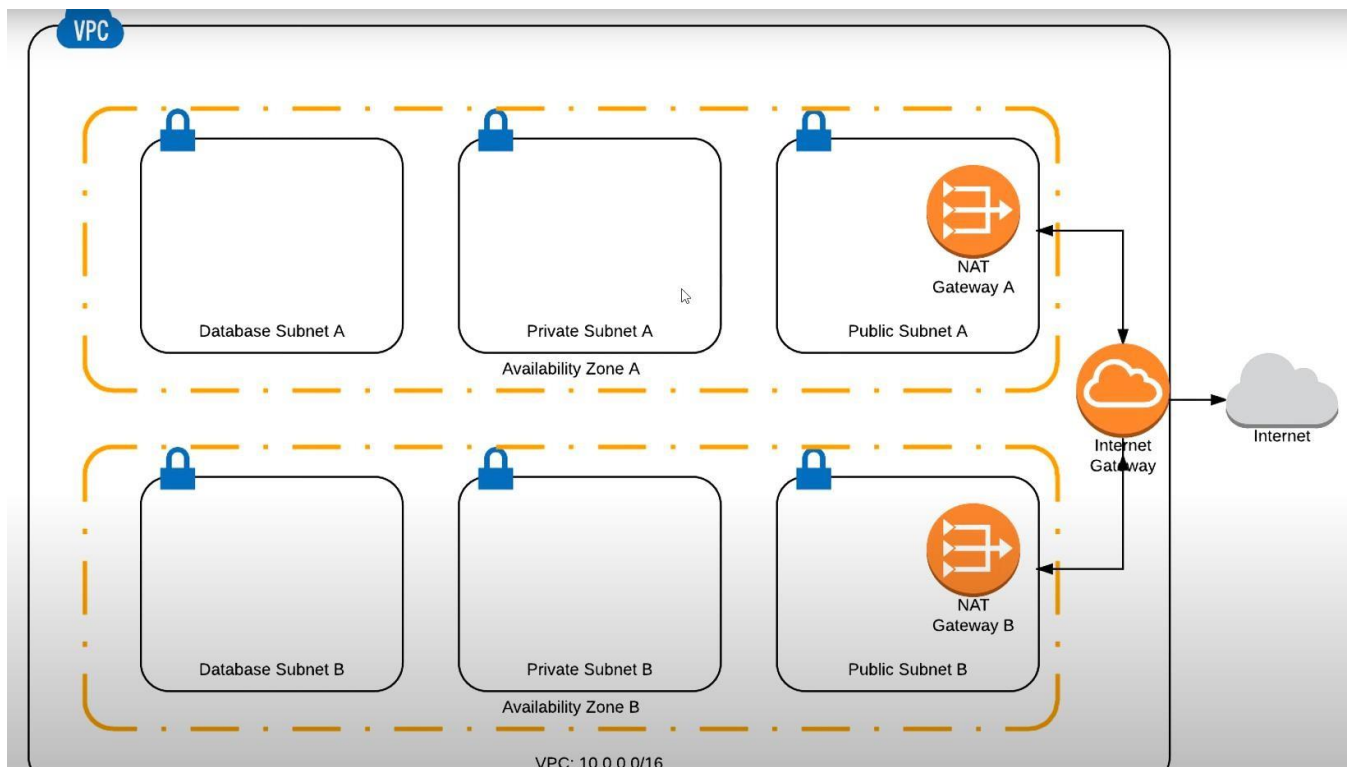


Рис 3.9 Схема розгорнутої VPC

Наступним етапом є підняття віртуальних машин в цих підмережах. Спочатку треба створити групу безпеки для цих машин. В них буде два правила, перше – дозвіл на мережевий протокол SSH, за для подальшого підключення на віртуальну машину, друге – ICMP, за для утиліти ping.

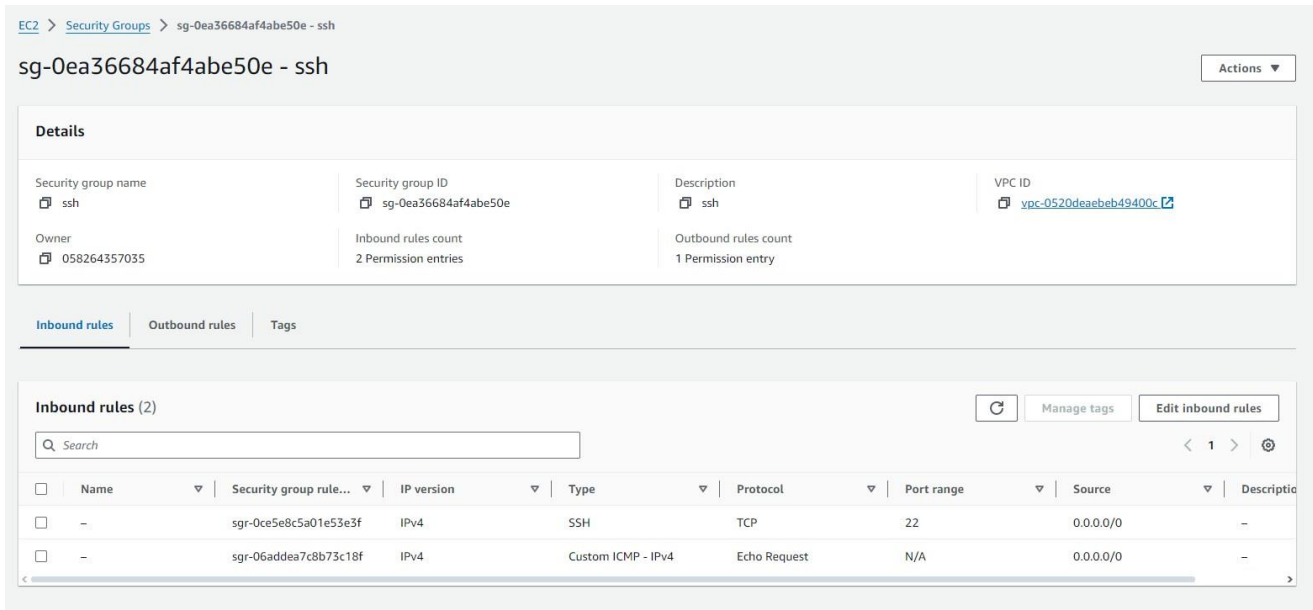


Рис 3.10 Група безпеки з доданими правилами

Далі вже розгортаємо віртуальні машини в створених підмережах, з створеною групою безпеки. За бажанням ви можете змінити операційну систему, додати власні зображення ОС (AMI), додати додаткові жорсткі диски до машин та інше.

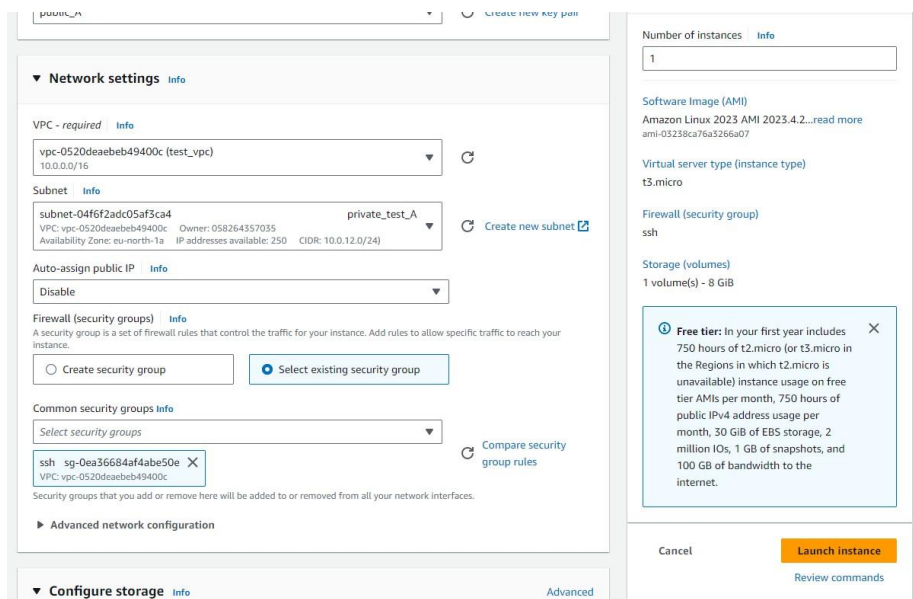


Рис 3.11 Створення віртуальної машини

Тепер за допомогою сервісу RDS, створимо потрібну нам базу даних. За власними потребами налаштуємо базу даних. Одним з плюсів даного сервісу є те, що можна не розгортати другу базу даних в іншій зоні доступності, а просто зробити вже репліку створеної бази даних.

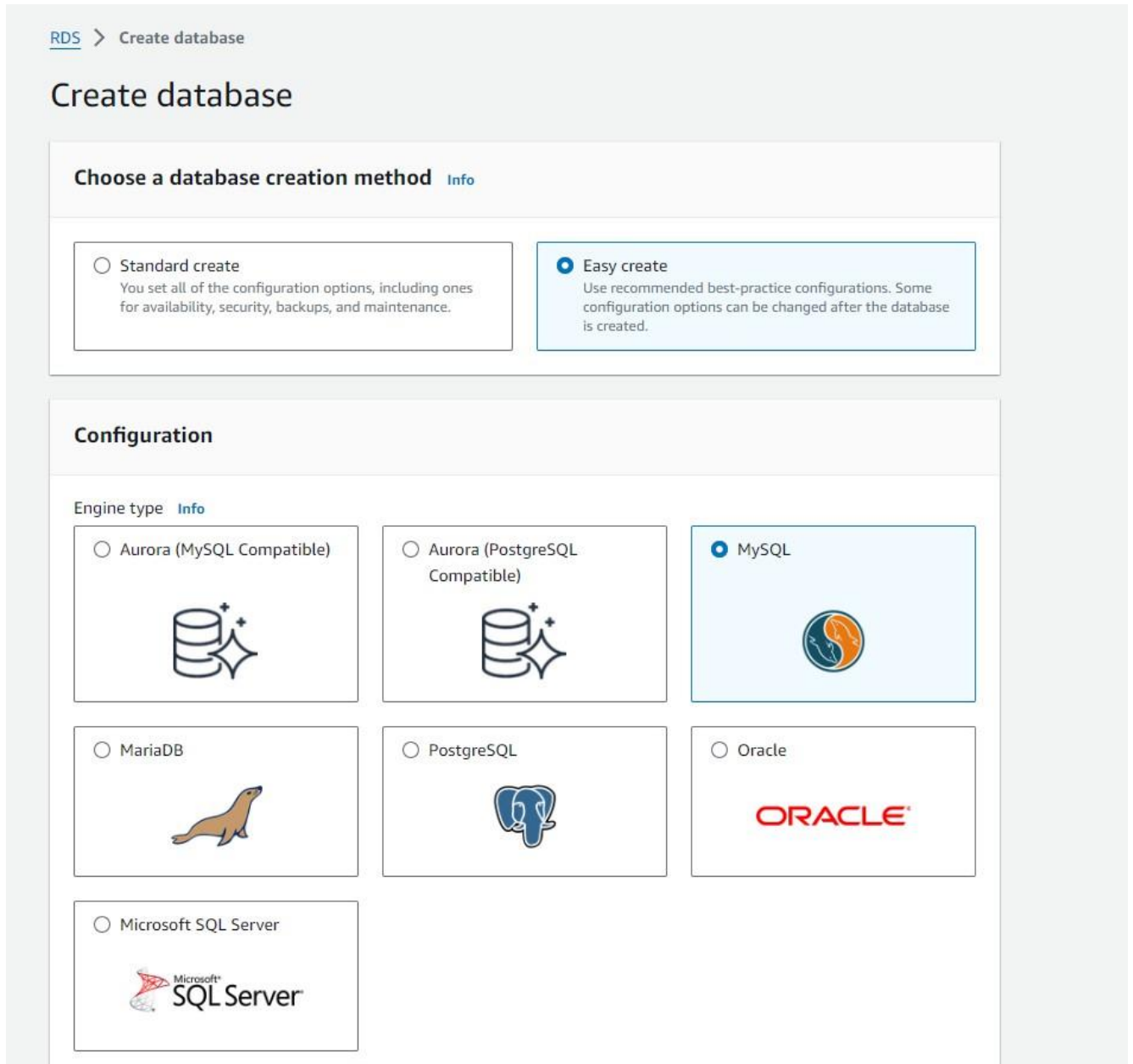


Рис 3.12 Створення бази даних

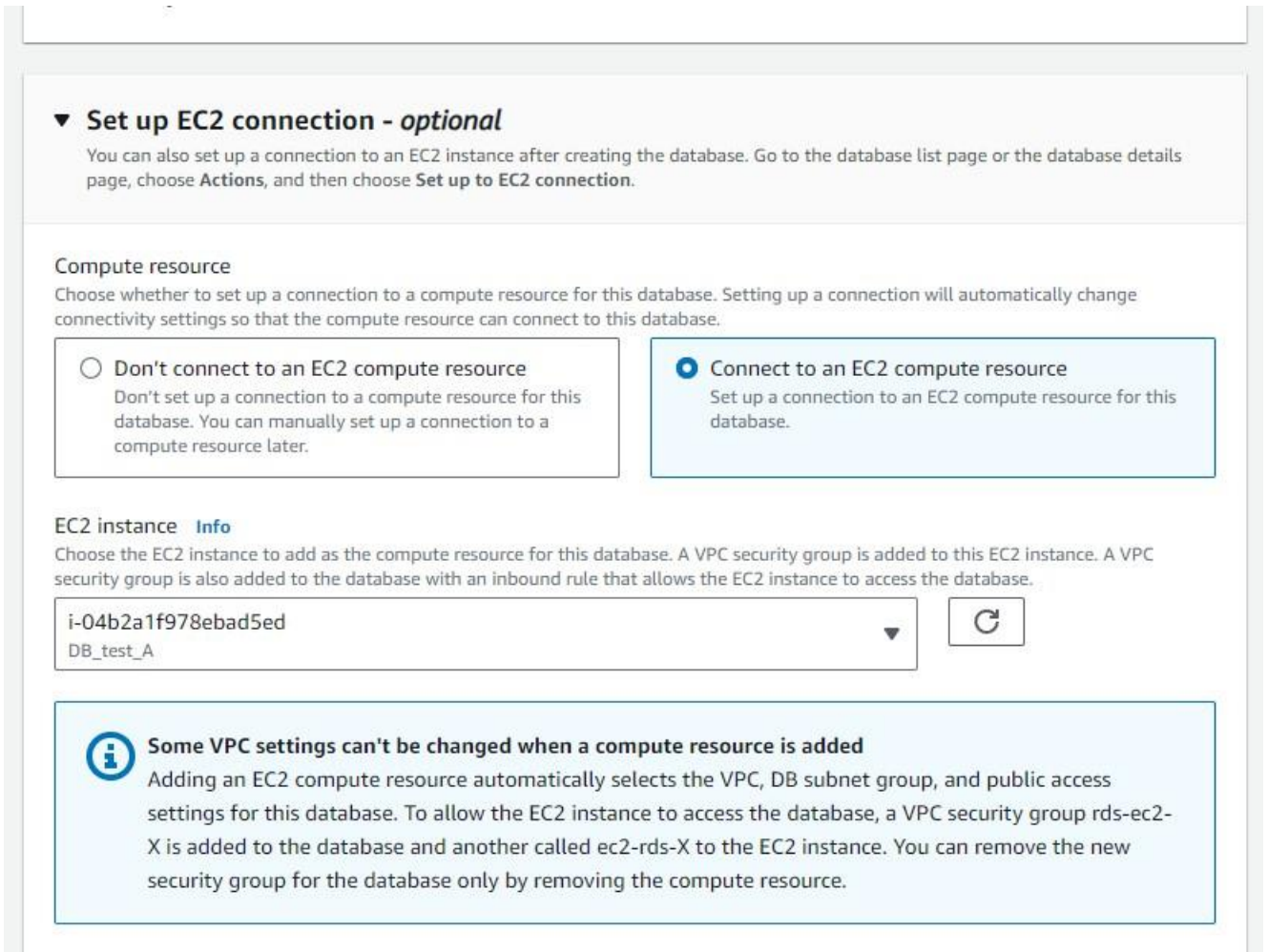


Рис 3.13 Підключення бази даних до інстансу

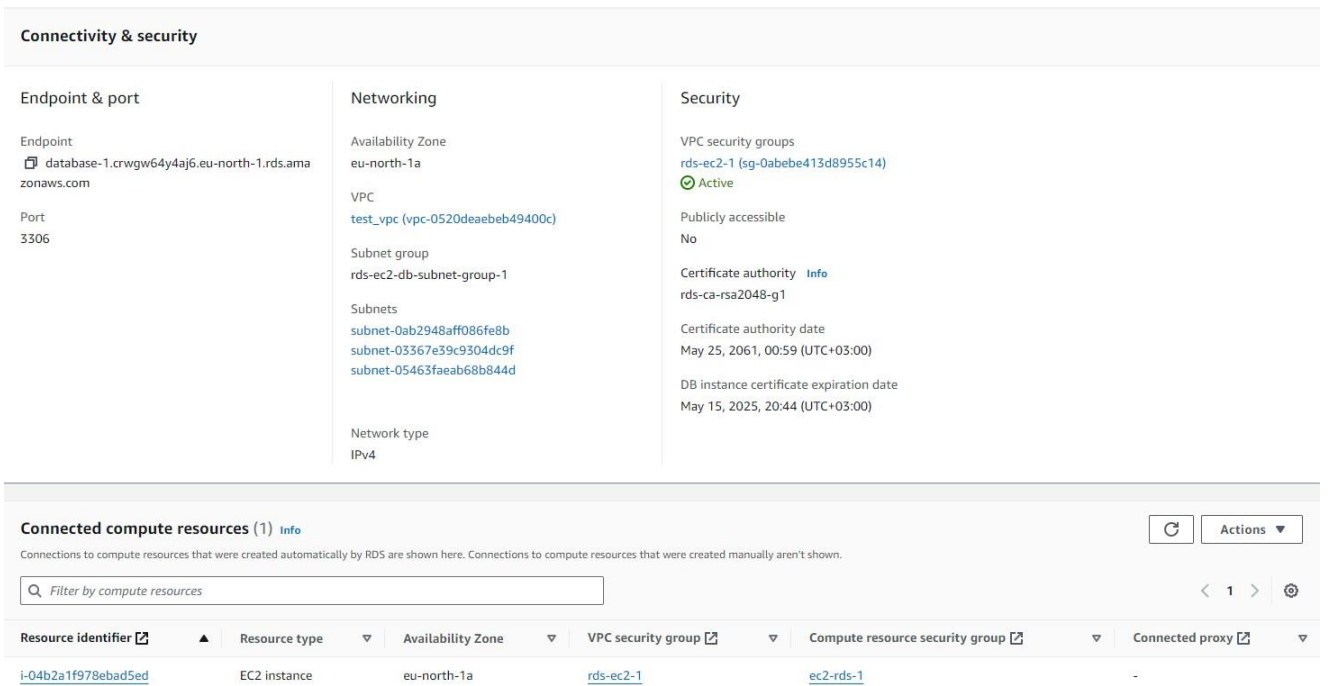


Рис 3.14 Піднята база даних

Наступним кроком є створення бакету S3. S3 бакет є глобальним, тому треба надати таке ім'я котре буде унікальним.

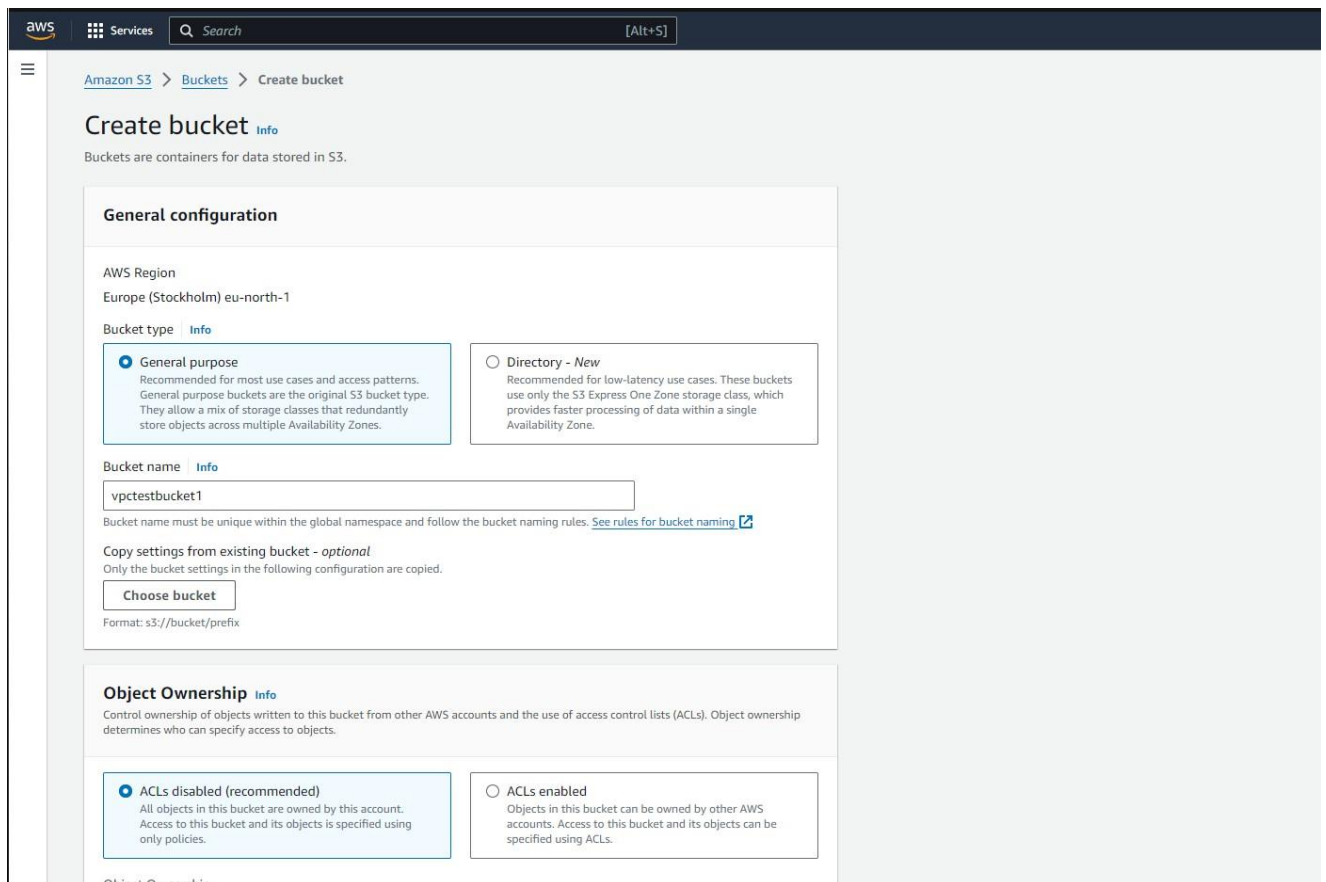


Рис 3.15 Створення S3 бакету

Тепер щоб поєднати S3 бакет і нашу інфраструктуру потрібно створити IAM роль для наших інстансів.

IAM в AWS розшифровується як управління ідентифікацією та доступом. Як випливає з назви, це служба AWS для управління ідентифікацією та доступом до ресурсів і сервісів. Керовані політики - Керовані політики - це прості JSON документи, які визначають, що можна і що не можна робити з Ресурсами цією Ідентифікацією. Ці політики можна використовувати стільки разів, скільки потрібно. Керовані політики AWS попередньо визначені AWS, а політики, керовані клієнтом, повинні бути визначені нами.

Дамо нашим інстансам можливість читати наші бакети за допомогою вже створеного JSON файлу, який має вигляд:

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::vpctestbucket1/*",
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": "10.0.0.0/24"
      }
    }
  }
]
}

```

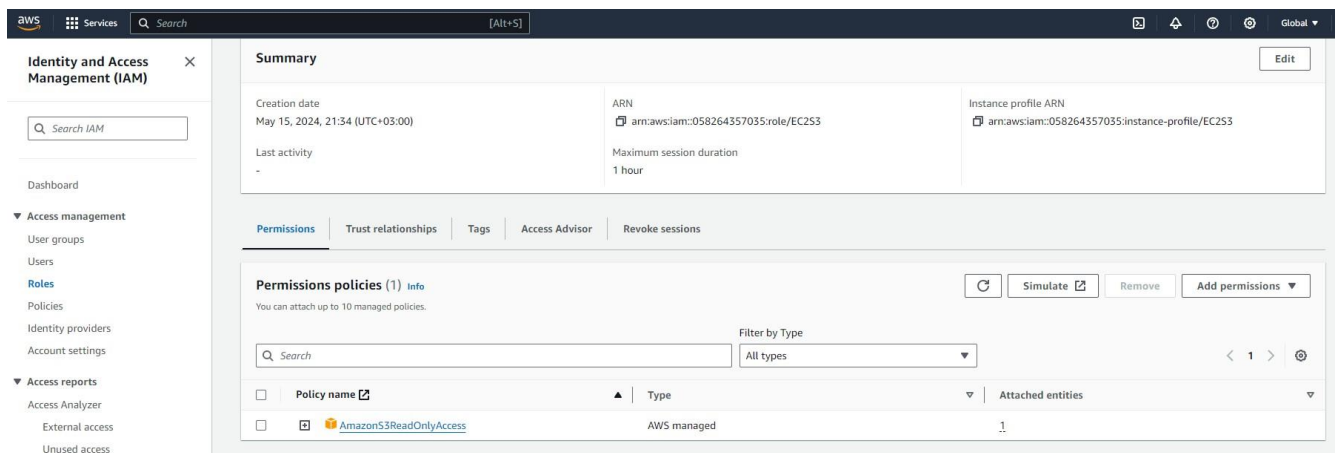


Рис 3.16 Створена IAM роль

Далі надамо нашим інстансам модифіковану IAM роль.

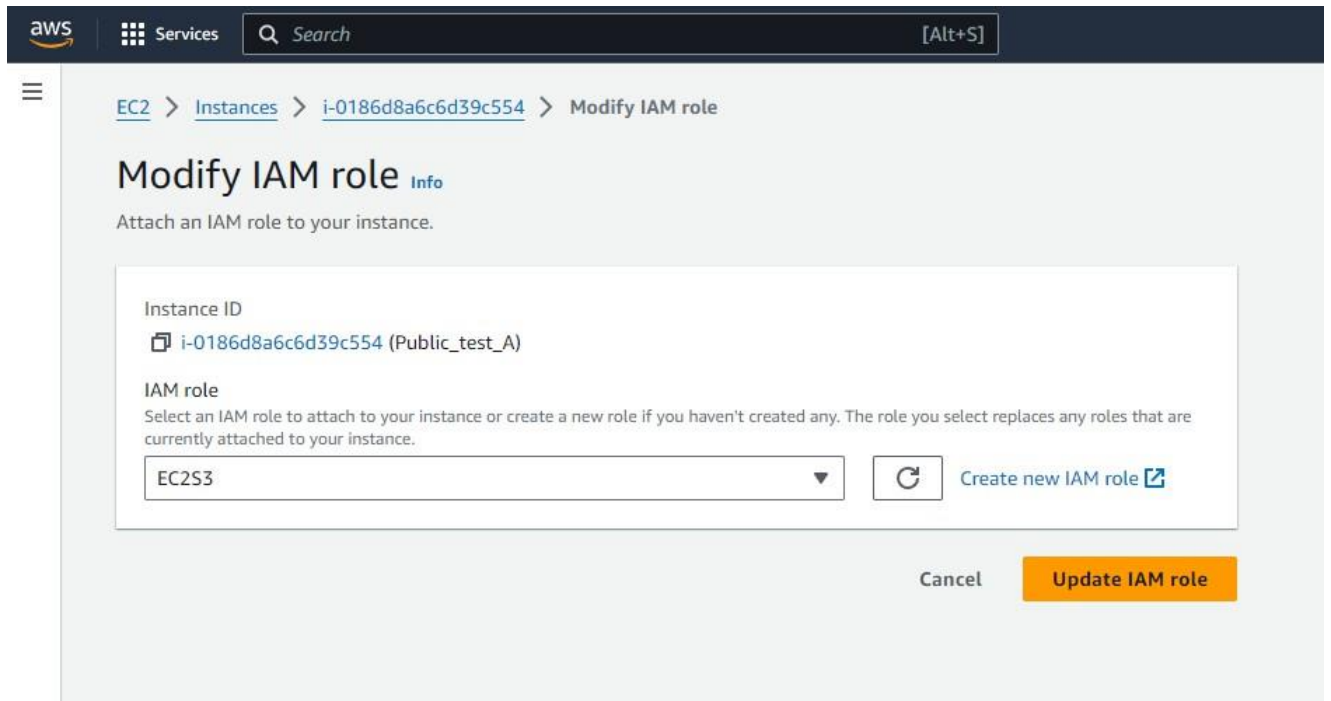


Рис 3.17 Надання нової ролі віртуальній машині

На цьому побудова інфраструктури закінчена. Наступний крок – це перевірка. Для перевірки з’єднання була використана програма PuTTY. PuTTY – безкоштовний продукт для підключення до серверу через мережеві протоколи SSH, Telnet і Rlogin. Для підключення нам знадобиться IP інстанса, порт(SSH – 22 порт), та ключ в форматі .ppk. Даний ключ створюється самим Амазоном при створенні віртуальної машини. Даний ключ може бути в 2 форматах: .ppk – для входу за допомогою Putty та .pem – для OpenSSH.

Спочатку перевіримо за допомогою команди `aws s3 ls` наш список бакетів, який доступний для нашої підмережі. Далі в бакет було розміщено певний файл, і за допомогою команди `aws s3 ls s3://vpctestbucket1`(назва бакету) було перевірено чи має підмережа доступ для читання бакету.

```
[ec2-user@ip-10-0-12-39 ~]$ aws s3 ls
2024-05-15 11:16:55 vpctestbucket1
[ec2-user@ip-10-0-12-39 ~]$ aws s3 ls s3://vpctestbucket1
2024-05-15 18:40:40      1679 key.pem
[ec2-user@ip-10-0-12-39 ~]$
```

Рис 3.18 Перевірка підключення до бакета

Наступні наші перевірки це правильність таблиць маршрутів. Щоб це перевірити за допомогою PuTTY треба зайти на кожну віртуальну машину і

прописати команду ping на google.com, щоб перевірити з'єднання з інтернетом та на внутрішній IP адрес в межах нашої VPC.

```

ec2-user@ip-10-0-11-86:~
Using username "ec2-user".
Authenticating with public key "public_A" from agent

#
~\  #####          Amazon Linux 2023
~~  \#####\
~~   \###|
~~   \#/  https://aws.amazon.com/linux/amazon-linux-2023
~~   V~'  '->
~~~~
~~~~
~/m/'

Last login: Wed May 15 16:32:35 2024 from 185.191.178.30
[ec2-user@ip-10-0-11-86 ~]$ ping google.com
PING google.com (142.250.74.142) 56(84) bytes of data:
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=1 ttl=54 time
=4.07 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=2 ttl=54 time
=4.09 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=3 ttl=54 time
=4.07 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=4 ttl=54 time
=4.09 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=5 ttl=54 time=4.08 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=6 ttl=54 time=4.11 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=7 ttl=54 time=4.09 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=8 ttl=54 time=4.04 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=9 ttl=54 time=4.13 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=10 ttl=54 time=4.06 ms
64 bytes from arnl1s11-in-f14.1e100.net (142.250.74.142): icmp_seq=11 ttl=54 time=4.08 ms

```

Рис 3.19 Виконання команди ping google.com з публічної підмережі

```

ttt min/avg/max/mdev = 4.037/4.076/4.126/0.020 ms
[ec2-user@ip-10-0-11-86 ~]$ ssh -i public_A.pem ec2-user@10.0.12.39

#
~\  #####          Amazon Linux 2023
~~  \#####\
~~   \###|
~~   \#/  https://aws.amazon.com/linux/amazon-linux-2023
~~   V~'  '->
~~~~
~~~~
~/m/'

Last login: Wed May 15 16:33:55 2024 from 10.0.11.86
[ec2-user@ip-10-0-12-39 ~]$ ping google.com
PING google.com (216.58.211.14) 56(84) bytes of data:
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=1 ttl=54 time=3.24 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=2 ttl=54 time=2.98 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=3 ttl=54 time=3.00 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=4 ttl=54 time=2.99 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=5 ttl=54 time=2.97 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=6 ttl=54 time=3.01 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=7 ttl=54 time=2.97 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=8 ttl=54 time=3.04 ms
64 bytes from muc03s13-in-f14.1e100.net (216.58.211.14): icmp_seq=9 ttl=54 time=2.97 ms
^C
--- google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 801lms
ttd min/avg/max/mdev = 2.966/3.018/3.243/0.082 ms
[ec2-user@ip-10-0-12-39 ~]$

```

Рис 3.20 Виконання команди ping google.com з приватної підмережі

```

[ec2-user@ip-10-0-12-39 ~]$ exit
Logout
Connection to 10.0.12.39 closed.
[ec2-user@ip-10-0-11-86 ~]$ ssh -i public_A.pem ec2-user@10.0.13.139
The authenticity of host '10.0.13.139 (10.0.13.139)' can't be established.
ED25519 key fingerprint is SHA256:zh8aXglDxkKti4v8UeEtwSsearlfx6i6lYzWN7Nykqk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.0.13.139' (ED25519) to the list of known hosts.

      #_
     _###_
    _###_#
   _###_#
  _###_#
 _###_#
V~'  '->
      #_
     _###_
    _###_#
   _###_#
  _###_#
 _###_#
V~'  '->

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-13-139 ~]$
[ec2-user@ip-10-0-13-139 ~]$
[ec2-user@ip-10-0-13-139 ~]$
[ec2-user@ip-10-0-13-139 ~]$ ping google.com
PING google.com (216.58.207.238) 56(84) bytes of data.

```

Рис 3.21 Виконання команди ping google.com з підмережі бази даних

```

[ec2-user@ip-10-0-11-86 ~]$ ping 10.0.12.39
PING 10.0.12.39 (10.0.12.39) 56(84) bytes of data.
 64 bytes from 10.0.12.39: icmp_seq=1 ttl=127 time=0.184 ms
 64 bytes from 10.0.12.39: icmp_seq=2 ttl=127 time=0.174 ms
 64 bytes from 10.0.12.39: icmp_seq=3 ttl=127 time=0.179 ms
 64 bytes from 10.0.12.39: icmp_seq=4 ttl=127 time=0.163 ms
 64 bytes from 10.0.12.39: icmp_seq=5 ttl=127 time=0.200 ms
^C
--- 10.0.12.39 ping statistics ---
 5 packets transmitted, 5 received, 0% packet loss, time 4124ms
 rtt min/avg/max/mdev = 0.163/0.180/0.200/0.012 ms
[ec2-user@ip-10-0-11-86 ~]$

```

Рис 3.22 Виконання команди ping з публічної підмережі на приватну

```

Last login: Wed May 15 17:02:15 2024 from 185.191.178.30
[ec2-user@ip-10-0-11-86 ~]$ ping 10.0.13.139
PING 10.0.13.139 (10.0.13.139) 56(84) bytes of data.
 64 bytes from 10.0.13.139: icmp_seq=1 ttl=127 time=0.187 ms
 64 bytes from 10.0.13.139: icmp_seq=2 ttl=127 time=0.136 ms
 64 bytes from 10.0.13.139: icmp_seq=3 ttl=127 time=0.133 ms
 64 bytes from 10.0.13.139: icmp_seq=4 ttl=127 time=0.132 ms
 64 bytes from 10.0.13.139: icmp_seq=5 ttl=127 time=0.136 ms
 64 bytes from 10.0.13.139: icmp_seq=6 ttl=127 time=0.139 ms
 64 bytes from 10.0.13.139: icmp_seq=7 ttl=127 time=0.152 ms
^C
--- 10.0.13.139 ping statistics ---
 7 packets transmitted, 7 received, 0% packet loss, time 6252ms
 rtt min/avg/max/mdev = 0.132/0.145/0.187/0.018 ms
[ec2-user@ip-10-0-11-86 ~]$

```

Рис 3.23 Виконання команди ping з публічної підмережі на базу даних

Після проведення перевірок базова ІТ-інфраструктура готова. Вже готову інфраструктуру можна далі апгрейдити за своїми бажаннями та вимогами, запускати додатки та інше.

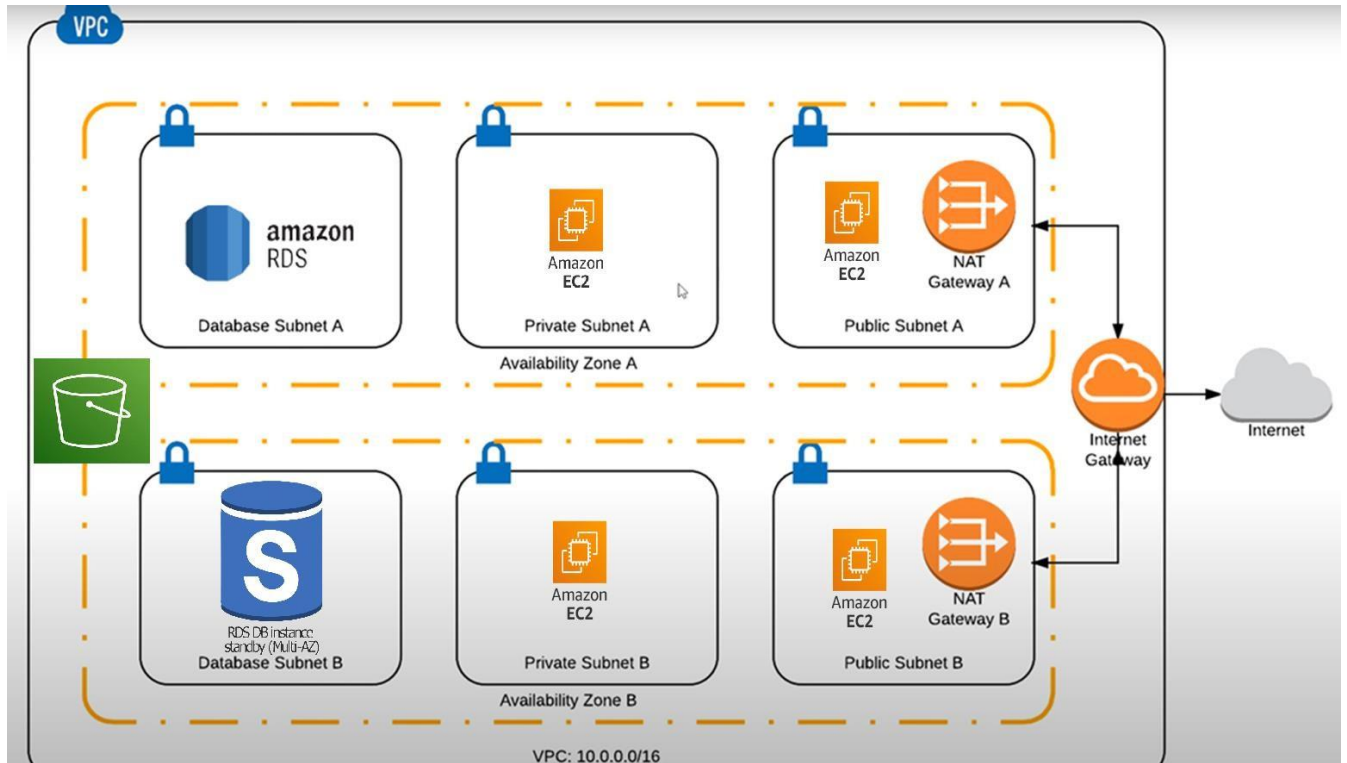


Рис 3.24 Кінцева схема ІТ-інфраструктури

Щодо використання досвіду використання власних серверів – то б дана інфраструктура на власних серверах має свої переваги і недоліки. А саме:

Переваги:

1. Повний контроль

Конфігурація та налаштування:

- Операційна система: Ви можете обрати будь-яку операційну систему (Linux, Windows Server, FreeBSD тощо) та версію, яка найкраще відповідає вашим вимогам.

- Параметри безпеки: Можливість налаштування міжмережевих екранів, VPN, системи виявлення вторгнень (IDS), що дозволяє підвищити рівень безпеки.
- Програмне забезпечення: Можна встановлювати будь-яке спеціалізоване ПЗ, налаштовувати серверні ролі (веб-сервер, бази даних, файловий сервер) відповідно до потреб.

Безпека:

- Інтеграція з існуючими системами захисту: Можливість налаштування відповідно до корпоративної політики безпеки.
- Контроль доступу: Встановлення багатофакторної автентифікації (MFA), налаштування прав доступу на основі ролей (RBAC).

Політика резервного копіювання:

- Регулярне резервування: Ви можете налаштувати власний графік створення резервних копій, вибирати частоту та тип резервування (повне, інкрементне, диференційоване).
- Місце зберігання: Можливість зберігати резервні копії на різних носіях або в різних локаціях для підвищення надійності.

2. Продуктивність

Висока продуктивність:

- Апаратні ресурси: Можна встановлювати процесори з високою тактовою частотою, велику кількість оперативної пам'яті та швидкі SSD-накопичувачі.
- Оптимізація під конкретні задачі: Налаштування параметрів системи, таких як розмір сторінки, кешування, управління ресурсами для забезпечення максимальної ефективності.

Низька латентність:

- Місцеве розташування: Сервери можна розмістити у дата-центрі, ближчому до користувачів, що зменшує затримки.

- Оптимізація мережевих налаштувань: Налаштування QoS (Quality of Service) для забезпечення пріоритету критичних даних.

3. Зниження витрат у довгостроковій перспективі

Відсутність щомісячних платежів:

- Одноразові витрати: Після початкових витрат на обладнання ви не платите щомісячних рахунків за оренду.
- Планування бюджету: Легше планувати бюджет без необхідності враховувати змінні витрати на хмарні сервіси.

Знижені витрати на масштабування:

- Розширення інфраструктури: Додавання нових серверів або оновлення існуючих компонентів можна здійснювати поетапно, залежно від потреб.
- Використання вживаного обладнання: Можливість використання вживаного або оновленого обладнання для зниження витрат.

4. Незалежність від провайдерів

Автономність:

- Управління інфраструктурою: Ви самостійно управляєте всіма аспектами серверної інфраструктури, включаючи апаратні та програмні оновлення.
- Політики та правила: Відсутність залежності від змін у політиці та умовах обслуговування зовнішніх постачальників.

Гарантована доступність:

- Власне обслуговування: Ви контролюєте всі технічні аспекти обслуговування серверів, що дозволяє забезпечити високу надійність.
- Планове обслуговування: Можливість планувати час технічного обслуговування відповідно до графіку роботи вашої компанії.

5. Гнучкість та адаптивність

Інтеграція з існуючою інфраструктурою:

- Сумісність: Легше забезпечити сумісність з існуючими системами та програмним забезпеченням вашої організації.
- Мінімізація перебоїв: Плавна інтеграція нових серверів або служб з мінімальними перебоями у роботі.

Адаптація під специфічні потреби:

- Швидка адаптація: Можливість швидко вносити зміни до конфігурації серверів для підтримки нових бізнес-процесів або програм.
- Гнучкість налаштувань: Налаштування серверів під конкретні робочі навантаження або проекти.

6. Конфіденційність даних

Контроль над даними:

- Місцеве зберігання: Зберігання даних у власних дата-центрах або офісах забезпечує фізичний контроль.
- Правила доступу: Встановлення власних правил доступу до даних, що зменшує ризик витоку інформації.

Безпека передачі даних:

- Шифрування: Можливість налаштування шифрування даних при передачі та зберіганні для підвищення безпеки.
- Моніторинг: Власні засоби моніторингу та аналізу безпеки для виявлення та запобігання можливих загроз.

7. Висока стабільність та надійність

Контроль над інфраструктурою:

- Якісне обладнання: Використання надійного та високоякісного апаратного забезпечення.

- Проактивне обслуговування: Регулярне профілактичне обслуговування для запобігання поломкам.

Планування та резервування:

- Планування обслуговування: Можливість планувати технічне обслуговування та оновлення відповідно до потреб бізнесу.
- Резервні системи: Налаштування резервних систем для забезпечення безперервності роботи у разі відмови основного обладнання.

Недоліки:

1. Високі початкові витрати

Інвестиції у обладнання:

- Закупівля серверів: Вартість серверного обладнання може бути значною, особливо для потужних серверів з високою продуктивністю.
- Супутні витрати: Необхідність придбання додаткового обладнання, такого як маршрутизатори, комутатори, системи зберігання даних, мережеві карти, кабелі тощо.

Витрати на інфраструктуру:

- Фізична локація: Необхідність обладнання серверної кімнати або дата-центру з відповідною системою вентиляції, охолодження, електроживлення.
- Безперебійне живлення (UPS): Витрати на системи безперебійного живлення для захисту від перебоїв у електропостачанні.
- Безпека приміщень: Встановлення систем фізичної безпеки, таких як камери спостереження, сигналізації, системи доступу.

2. Технічне обслуговування та підтримка

Потреба у фахівцях:

- Команда технічної підтримки: Наймання кваліфікованих ІТ-фахівців для управління та обслуговування серверів.

- Постійне навчання: Інвестування у навчання та підвищення кваліфікації співробітників для роботи з новими технологіями та обладнанням.

Витрати на обслуговування:

- Оновлення програмного забезпечення: Регулярне оновлення операційних систем, серверного програмного забезпечення, систем безпеки.
- Ремонт і заміна обладнання: Витрати на ремонт або заміну компонентів, що вийшли з ладу, такі як жорсткі диски, вентилятори, блоки живлення.

3. Масштабованість

Обмежена масштабованість:

- Фізичні обмеження: Обмежений простір для розміщення додаткового обладнання в існуючих приміщеннях.
- Тривалі процеси: Процес закупівлі, встановлення та налаштування нового обладнання може займати багато часу.

Гнучкість:

- Час на адаптацію: Додавання нових серверів або збільшення потужності існуючих може потребувати значного часу для налаштування та інтеграції.
- Витрати на модернізацію: Часто потрібні значні фінансові витрати на модернізацію інфраструктури для підтримки зростаючих вимог.

4. Вразливість до фізичних ризиків

Катастрофи:

- Стихійні лиха: Сервери можуть бути пошкоджені або знищені внаслідок природних катастроф, таких як пожежі, повені, землетруси.
- Техногенні ризики: Ризики, пов'язані з аваріями, такими як перебої в електропостачанні, витік води, пожежа через коротке замикання.

Крадіжки та вандалізм:

- Зломи: Ризик фізичних зломів і крадіжок обладнання або даних.

- Вандалізм: Можливість пошкодження обладнання внаслідок дій зловмисників.

5. Проблеми з безпекою

Кібербезпека:

- Атаки: Необхідність постійного захисту від кібератак, таких як DDoS-атаки, спроби проникнення, віруси та шкідливе програмне забезпечення.
- Моніторинг: Постійний моніторинг мережевого трафіку та діяльності для виявлення загроз безпеці.

Витрати на безпеку:

- Захист даних: Витрати на впровадження та підтримку систем шифрування, міжмережових екранів, антивірусного ПЗ та інших заходів безпеки.
- Резервування даних: Затрати на створення і підтримку резервних копій та планів відновлення після катастроф.

6. Високі витрати на електроенергію

Споживання електроенергії:

- Високі витрати: Сервери споживають значну кількість електроенергії, особливо при використанні потужних компонентів та систем охолодження.
- Охолодження: Додаткові витрати на охолодження серверного приміщення для підтримки оптимальної температури роботи обладнання.

7. Часові витрати

Управління інфраструктурою:

- Час на налаштування: Початкове налаштування та конфігурація серверів можуть вимагати значного часу і зусиль.
- Постійне обслуговування: Регулярні технічні роботи, моніторинг та обслуговування можуть відволікати від основної діяльності.

Відновлення після збоїв:

- Проблеми з відновленням: У разі серйозних збоїв або поломок може знадобитися значний час для відновлення нормальної роботи системи.
- Планування резервного копіювання: Витрати часу на налаштування і управління резервними копіями та планами відновлення

ВИСНОВКИ

У дипломній роботі проведено всебічний аналіз та порівняння розробки інфраструктури на базі Amazon Web Services (AWS) та використання власних серверів. Дослідження охоплює аспекти продуктивності, вартості, масштабованості, безпеки та адміністрування, що дозволяє зробити низку важливих висновків.

1. **Продуктивність та масштабованість:** Використання AWS надає значні переваги у масштабованості та продуктивності. AWS дозволяє динамічно налаштовувати ресурси в залежності від поточних потреб бізнесу, забезпечуючи високу доступність та швидке реагування на зміни у навантаженні. Власні сервери мають обмежені можливості для масштабування і потребують додаткових витрат часу та ресурсів на оновлення обладнання та розширення інфраструктури.
2. **Вартість:** AWS пропонує гнучкі моделі оплати, що дозволяють платити тільки за використані ресурси. Це особливо вигідно для стартапів та компаній з нерегулярними піками навантаження. Водночас, власні сервери потребують значних капітальних вкладень на початковому етапі та постійних витрат на підтримку та обслуговування. Проте для великих компаній з передбачуваним та стабільним навантаженням використання власних серверів може бути економічно вигіднішим у довгостроковій перспективі.
3. **Безпека:** AWS забезпечує високий рівень безпеки, пропонуючи різноманітні інструменти для захисту даних та управління доступом. Однак, використання хмарних сервісів вимагає додаткових зусиль для налаштування політик безпеки та відповідності регуляторним вимогам. Власні сервери надають повний контроль над фізичним доступом до обладнання та можуть бути налаштовані відповідно до специфічних потреб компанії, що може бути критичним для певних галузей з високими вимогами до конфіденційності даних.

4. Адміністрування та підтримка: AWS знімає з користувачів значну частину обов'язків з адміністрування та підтримки інфраструктури, надаючи автоматизовані інструменти для управління та моніторингу. Це дозволяє зменшити навантаження на ІТ-персонал і зосередитись на основних бізнес-процесах. Використання власних серверів вимагає наявності кваліфікованого персоналу для їх обслуговування та управління, що збільшує операційні витрати.

На основі проведеного аналізу можна зробити висновок, що вибір між використанням Amazon Web Services та власних серверів залежить від конкретних потреб та ресурсів компанії. AWS є оптимальним рішенням для компаній, які цінують гнучкість, масштабованість та мінімальні початкові витрати. Власні сервери можуть бути вигіднішими для компаній з стабільними навантаженнями та специфічними вимогами до безпеки та конфіденційності даних.

Таким чином, кожен підхід має свої переваги та недоліки, і рішення про вибір інфраструктури повинно базуватися на детальному аналізі бізнес-вимог та стратегічних цілей компанії.

ПЕРЕЛІК ПОСИЛАНЬ

1. AWS VPC design best practices - geeksforgeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/aws-vpc-design-best-practices/> (date of access: 23.04.2024).
2. Baron S. *Aws: the complete beginner's guide to mastering amazon web services*. Independently Published, 2020. 168 с.
3. Cloud computing services - amazon web services (AWS). *Amazon Web Services, Inc*. URL: https://aws.amazon.com/?nc1=h_ls (дата звернення: 03.04.2024).
4. Cloud J. *Aws: the ultimate amazon web services guide from beginners to advanced*. Independently Published, 2020. 260 с.
5. Difference between internet gateway and NAT gateway - geeksforgeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/difference-between-internet-gateway-and-nat-gateway/> (date of access: 26.04.2024).
6. Edu Eyrie India Private Limited. Amazon VPC: everything you need to know. *LinkedIn: Log In or Sign Up*. URL: <https://www.linkedin.com/pulse/amazon-vpc-everything-you-need-know-edueyrie-india-private-limited-qlocс/> (дата звернення: 16.04.2024).
7. Eng P. M. V. M. Evolution of IT infrastructure. *LinkedIn: Log In or Sign Up*. URL: <https://www.linkedin.com/pulse/evolution-infrastructure-paul-m-veillard/> (дата звернення: 04.04.2024).
8. IT infrastructure. *Deutschland | Let There Be Change | Accenture*. URL: <https://www.accenture.com/cz-en/insights/cloud/it-infrastructure-index> (дата звернення: 04.04.2024).
9. IT infrastructure evolution towards scalability, productivity and security - Interfaz. *Interfaz*. URL: <https://interfaz.io/2022/07/it-infrastructure-evolution-towards-scalability-productivity-and-security/> (дата звернення: 05.04.2024).
10. King T. H. *Aws: the ultimate guide from beginners to advanced for the amazon web services*. Independently Published, 2019. 197 с.

11. On-Premises architecture. *eLabNext*.
URL: <https://www.elabnext.com/hosting/elab-on-premises/on-premises-architecture/> (date of access: 29.04.2024).
12. Overview of amazon web services - overview of amazon web services.
URL: <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html> (дата звернення: 04.04.2024).
13. Ryan M., Lucifredi F. AWS system administration: best practices for sysadmins in the amazon cloud. O'Reilly Media, 2018. 384 с.
14. URL: <https://www.spiceworks.com/tech/it-strategy/articles/what-is-it-infrastructure/> (дата звернення: 04.04.2024).
15. What is amazon EC2? - amazon elastic compute cloud.
URL: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html> (дата звернення: 18.04.2024).
16. What is amazon relational database service (amazon RDS)? - amazon relational database service.
URL: <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html> (дата звернення: 23.04.2024).
17. What is amazon S3? - amazon simple storage service.
URL: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html> (дата звернення: 22.04.2024).
18. What is amazon VPC? - amazon virtual private cloud.
URL: <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html> (дата звернення: 19.04.2024).
19. What is an IT infrastructure? Types and components. *Cleveroad Inc.* - *Web and App development company*. URL: <https://www.cleveroad.com/blog/what-is-it-infrastructure/> (дата звернення: 05.04.2024).
20. What is IT infrastructure? | IBM. *IBM in Deutschland, Österreich und der Schweiz*.
URL: <https://www.ibm.com/topics/infrastructure> (дата звернення: 04.04.2024).

Державний університет інформаційно-комунікаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Аналіз та порівняння розробки інфраструктури на базі Amazon Web Services і використання власних серверів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав: Омельченко Д.В. ІСД-42
Науковий керівник роботи: Каграманова
Ю.К.

Київ - 2024

- **Мета роботи** – дослідити та порівняти розробку інфраструктури на базі Amazon Web Services (AWS) та використання власних серверів.
- **Об'єкт дослідження** – інформаційні технології та інфраструктура для розробки і розгортання програмного забезпечення.
- **Предмет дослідження** – методи та підходи до розробки IT-інфраструктури з використанням хмарних сервісів AWS та власних серверів.
- **Актуальність теми** - Вибір між AWS і власними серверами для розробки та хостингу ПЗ стає все більш важливим через економію коштів, масштабованість, безпеку та зручність використання.
- **Наукова новизна** - Дослідження порівнює AWS і власні сервери за ефективністю, безпекою, зручністю та відповідністю потребам, надаючи рекомендації щодо оптимального вибору інфраструктури.
- **Завдання дослідження:**
 1. Проаналізувати особливості розробки IT-інфраструктури з використанням Amazon Web Services.
 2. Порівняти ключові відмінності інфраструктур на базі AWS і власних серверів.

Elastic Compute Cloud

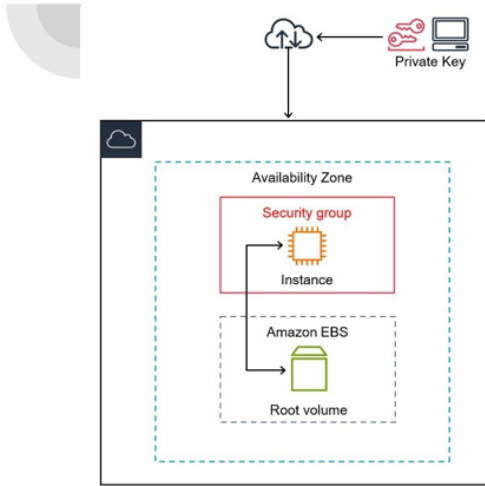
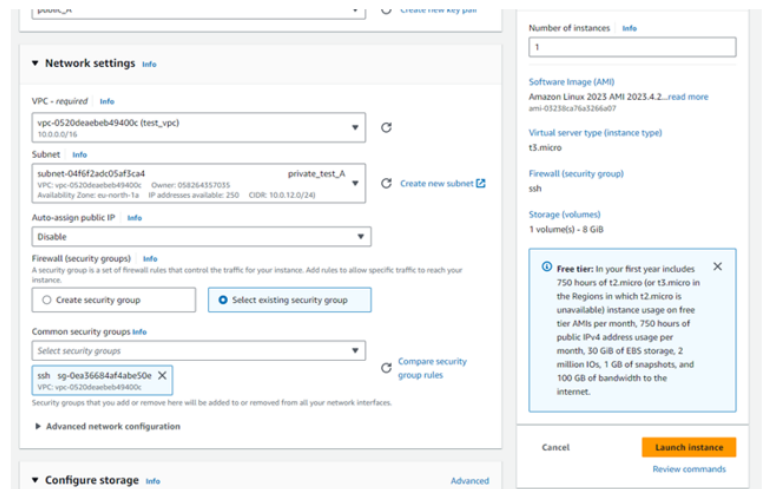
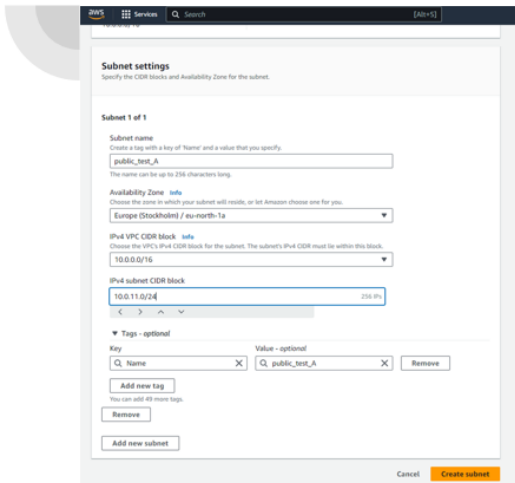


Схема побудови EC2



Вигляд веб-інтерфейсу створення інстансу

Virtual Private Cloud



Вигляд веб-інтерфейсу створення підмережі

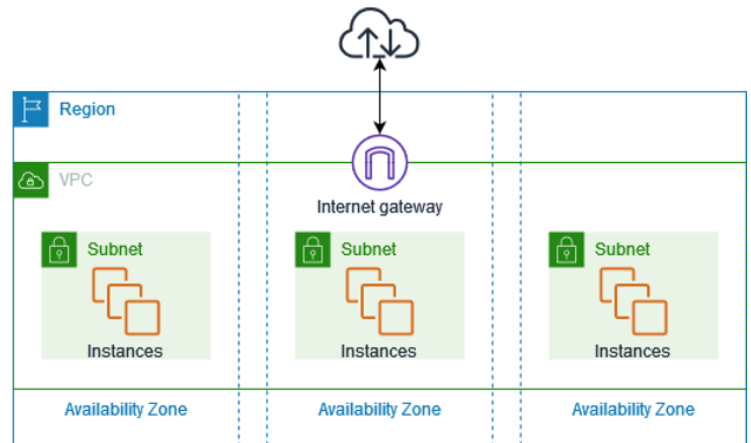
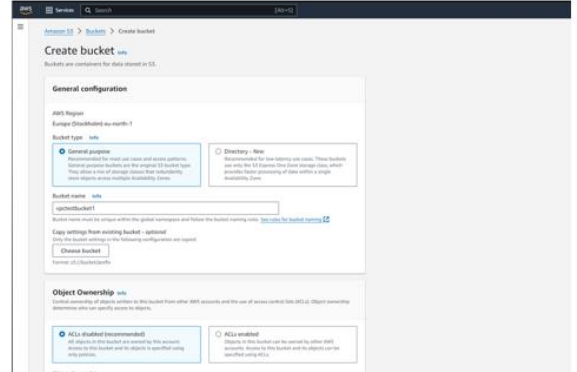


Схема побудови VPC

Simple Storage Service

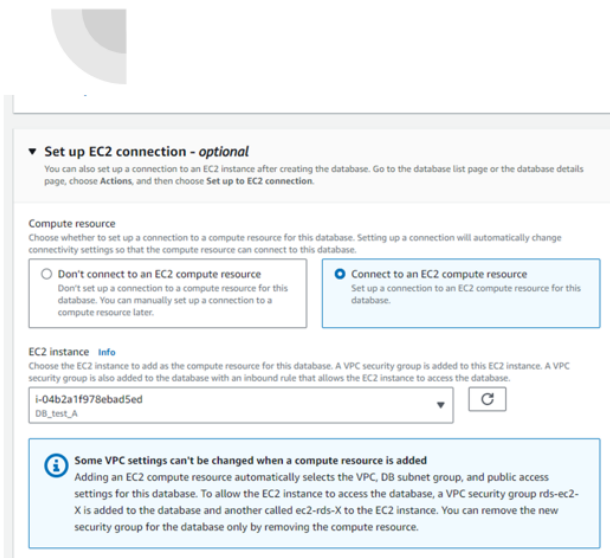


Схема принципу роботи S3



Вигляд веб-інтерфейсу створення бакету

Relational Database Service



Вигляд веб-інтерфейсу створення бази даних на основі інстансу

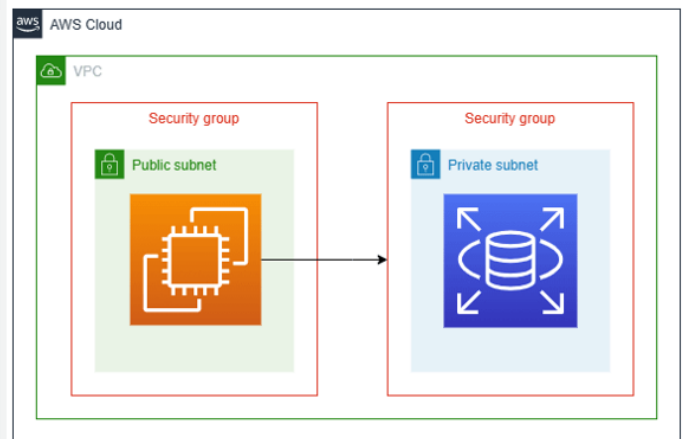


Схема використання бази даних в VPC

Результати створення та тестування

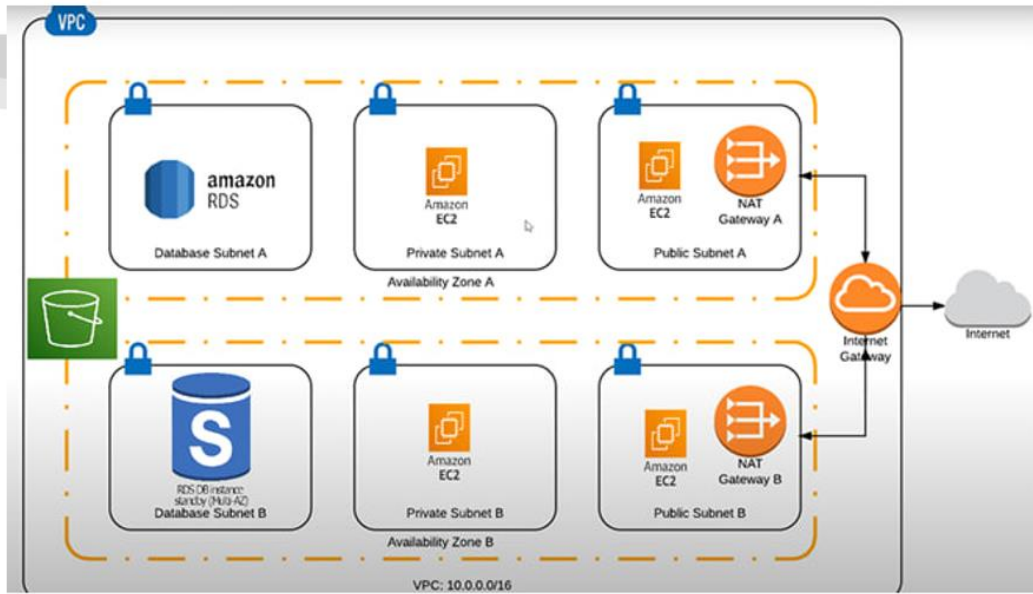


Схема готової хмарної інфраструктури

```

ec2-user@ip-10-0-11-86:
└─$ ssh -i public_A.pem ec2-user@10.0.12.39
Warning: Permanently added '10.0.12.39' (EC2::Linux) to the list of known hosts.
Last login: Wed May 15 16:32:35 2024 from 185.191.178.30
ec2-user@ip-10-0-12-39 ~$ ping google.com
PING google.com (142.250.74.142): 56(84) bytes of data:
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=1 ttl=54 time
=4.07 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=2 ttl=54 time
=4.09 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=3 ttl=54 time
=4.07 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=4 ttl=54 time
=4.05 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=5 ttl=54 time=4.08 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=6 ttl=54 time=4.11 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=7 ttl=54 time=4.09 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=8 ttl=54 time=4.04 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=9 ttl=54 time=4.13 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=10 ttl=54 time=4.06 ms
64 bytes from arn1111-in-f14.1e100.net (142.250.74.142): icmp_seq=11 ttl=54 time=4.08 ms

```

Перевірка публічної мережі на доступність в інтернет

```

ec2-user@ip-10-0-12-39 ~$ ssh -i public_A.pem ec2-user@10.0.13.139
Warning: Permanently added '10.0.13.139' (EC2::Linux) to the list of known hosts.
Last login: Wed May 15 16:33:55 2024 from 10.0.11.86
ec2-user@ip-10-0-12-39 ~$ ping google.com
PING google.com (216.58.207.238): 56(84) bytes of data:
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=1 ttl=54 time=3.24 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=2 ttl=54 time=2.98 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=3 ttl=54 time=3.00 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=4 ttl=54 time=2.99 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=5 ttl=54 time=2.97 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=6 ttl=54 time=3.01 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=7 ttl=54 time=2.97 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=8 ttl=54 time=3.04 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=9 ttl=54 time=2.97 ms
--- google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 801ms
rtt: min/avg/max/mdev = 2.966/3.012/3.243/0.082 ms

```

Перевірка мережі бази даних на доступність в інтернет

```

ec2-user@ip-10-0-11-86 ~$ ssh -i public_A.pem ec2-user@10.0.12.39
Warning: Permanently added '10.0.12.39' (EC2::Linux) to the list of known hosts.
Last login: Wed May 15 16:33:55 2024 from 10.0.11.86
ec2-user@ip-10-0-12-39 ~$ ping google.com
PING google.com (216.58.211.14): 56(84) bytes of data:
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=1 ttl=54 time=3.24 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=2 ttl=54 time=2.98 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=3 ttl=54 time=3.00 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=4 ttl=54 time=2.99 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=5 ttl=54 time=2.97 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=6 ttl=54 time=3.01 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=7 ttl=54 time=2.97 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=8 ttl=54 time=3.04 ms
64 bytes from msc03a13-in-f14.1e100.net (216.58.211.14): icmp_seq=9 ttl=54 time=2.97 ms
--- google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 801ms
rtt: min/avg/max/mdev = 2.966/3.012/3.243/0.082 ms

```

Перевірка приватної мережі на доступність в інтернет

Висновки

В рамках випускної кваліфікаційної роботи було розглянуто основні сервіси хмарної платформи AWS, їх призначення та функції. З їх допомогою була розроблена і розгорнута IT інфраструктура. Хоча дану інфраструктуру можна удосконалити, результат є задовільним, враховуючи складність використання безкоштовної версії AWS. На даному прикладі я побачив різницю в питаннях адміністрування, вартості, безпеки та масштабованості власної і хмарної інфраструктур.

Дана робота стала для мене цінним джерелом знань з області хмарних технологій. Завдяки ній я глибоко занурився у цю тему і переконався, що хмарна інфраструктура має свої переваги і значний потенціал.

Апробація

1. V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ»
18.04.2024 - ПЕРСПЕКТИВА ХМАРНИХ ТЕХНОЛОГІЙ У МАЙБУТНЬОМУ
2. V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ»
18.04.2024- ВПЛИВ ХМАРНИХ ТЕХНОЛОГІЙ НА РОЗВИТОК ШТУЧНОГО ІНТЕЛЕКТУ

Дякую за увагу!