**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА КОМП’ЮТЕРНОЙ ІНЖЕНЕРІЇ**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: **“Проектування мережевої архітектури для хмарних**

**обчислень з урахуванням вимог до їх продуктивності,**

**масштабованості та безпеки“**

на здобуття освітнього ступеня магістра

зі спеціальності \_\_\_123\_\_Комп’ютерна інженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(код, найменування спеціальності)*

освітньо-професійної програми Комп’ютерні системи та мережі

*(назва)*

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

\_\_\_\_\_\_\_\_\_\_ Владислав ДРИГА

*(підпис) Ім’я, ПРІЗВИЩЕ здобувача*

Виконав здобувач(ка) вищої освіти гр. КСДМ -61

Владислав ДРИГА

Ім’я, ПРІЗВИЩЕ

Керівник к.т.н., доцент Вячеслав Черевик

*науковий ступінь, ,* Ім’я, ПРІЗВИЩЕ

*вчене звання*

Рецензент*:* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*науковий ступінь,* Ім’я, ПРІЗВИЩЕ

*вчене звання*

**Київ 2024**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

Кафедра Комп’ютерної інженерії

Ступінь вищої освіти Магістр

Спеціальність Комп’ютерні системи та мережі

Освітньо-професійна програма Комп’ютерні системи та мережі

**ЗАТВЕРДЖУЮ**

Завідувач кафедрою КІ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Наталія ЛАЩЕВСЬКА

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Дризі Владиславу Олексійовичу

*(прізвище, ім’я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Проектування мережевої архітектури для хмарних обчислень з урахуванням вимог до їх продуктивності, масштабованості та безпеки

керівник кваліфікаційної роботи В’ячеслав ЧЕРЕВИК к.т.н., професор,

*(Ім’я, ПРІЗВИЩЕ науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, вимоги до середовища інтеграції, параметри мережевої архітектури.

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

Дослідження принципів проектування мережі в хмарних провайдерах

Оцінка вимог продуктивності, масштабу

Розробка рекомендацій для побудови надійної системи

5. Перелік графічного матеріалу: *презентація 19 слайдів*

6. Дата видачі завдання «7» жовтня 2023 р

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів  кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
| 1 | Аналіз наявної науково-технічної літератури | 19.10-05.11.23 |  |
| 2 | Вивчення матеріалів для проектування мережі | 05.11-12.11.23 |  |
| 3 | Дослідження хмарних мережевих сервісів | 13.11-19.11.23 |  |
| 4 | Аналіз вимог проектування мережі | 20.11-25.11.23 |  |
| 6 | Дослідження концепцій та рекомендацій дизайну | 04.12-10.12.23 |  |
| 7 | Оформлення роботи: вступ, висновки, реферат | 11.12-20.12.23 |  |
| 8 | Розробка демонстраційних матеріалів | 21.12-29.12.23 |  |

Здобувач вищої освіти \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Владислав ДРИГА

*(підпис) (Ім’я, ПРІЗВИЩЕ)*

Керівник

кваліфікаційної роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_ В’ячеслав ЧЕРЕВИК

*(підпис) (Ім’я, ПРІЗВИЩЕ)*

**РЕФЕРАТ**

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 72 стор., 8 табл., 11 рис., 21 джерел.

*Мета роботи* *–* розробка принципів побудови ефективної мережевої архітектури, що відповідає сучасним вимогам хмарних сервісів.

*Об’єкт дослідження –* проектування мережевої архітектури для хмарних обчислень.

*Предмет дослідження* *–* мережева архітектура для хмарних обчислень.

*Короткий зміст роботи*: Дипломна робота присвячена проектуванню мережевої архітектури для хмарних обчислень, зосереджуючись на продуктивності, масштабованості та безпеці. Вона охоплює такі аспекти, як принципи проектування мереж у хмарних провайдерів, оцінка потреб у продуктивності та масштабованості, а також рекомендації щодо створення надійної системи. Також обговорюються віртуалізація, балансування навантаження, мережеві протоколи, гнучкість та безпека в архітектурі хмарних мереж. Розроблено принципи для ефективної мережевої архітектури, що відповідає сучасним вимогам хмарних служб.

КЛЮЧОВІ СЛОВА: GOOGLE CLOUD, БЕЗПЕКА, МАШТАБУВАННЯ, ПРОДУКТИВНІСТЬ, БАЛАНСУВАННЯ НАВАНТАЖЕННЯ, БРАНДМАУЕР, МЕРЕЖА.

**ABSTRACT**

The textual part of the qualification work for obtaining a master's degree: 72 pages, 8 tables, 11 figure, 21 sources.

The goal of the work is to develop principles for constructing an effective network architecture that meets the modern requirements of cloud services.

The object of research is the design of network architecture for cloud computing.

The subject of research is network architecture for cloud computing.

Brief content of the work: The thesis is dedicated to designing network architecture for cloud computing, focusing on performance, scalability, and security. It covers aspects such as principles of network design in cloud providers, evaluation of performance and scalability requirements, and recommendations for building a reliable system. It also discusses virtualization, load balancing, network protocols, flexibility, and security in cloud network architecture. Principles for an effective network architecture that meets modern cloud service requirements were developed.

KEYWORDS: GOOGLE CLOUD, SECURITY, SCALABILITY, PERFORMANCE, LOAD BALANCING, FIREWALL, NETWORK.

**ЗМІСТ**

**ВСТУП** …………………………………………………………………………………9

1. **ПОСТАНОВКА ЗАДАЧІ** …………………………………………………….10
2. Характеристика Моделей Хмарних Обчислень ……………………………...10
3. Мережна архітектура в хмарних обчисленнях …………………………….…12
4. Продуктивність, масштабованість і безпека в хмарних мережах ……….......13
5. АНАЛІЗ ВИМОГ ЩОДО ПРОЕКТУВАННЯ МЕРЕЖІ У ХМАРІ …............18
6. Вимоги до продуктивності …………………………………………………….18
7. Вимоги до масштабованості ………………...………………………………...19
8. Вимоги безпеки ……………………………………………………………...…21
9. Інтеграція вимог ………………………………………………………………..22
10. **ДОСЛІДЖЕННЯ ПРИНЦИПІВ ДИЗАЙНУ ХМАРНОЇ МЕРЕЖІ** …….
11. 25
    1. Дизайн, орієнтований на продуктивність ………………… ………….……......25
       1. Оптимізація TCP для продуктивності мережі ……………………...… 30
       2. Варіанти балансування навантаження ………………………………... 34
    2. Дизайн, орієнтований на масштабованість …………………………………...45
       1. Підключення кількох мереж VPC ……………………………….……..48
       2. Використання мережевого пірингу VPC ………………………………50
       3. Використання зовнішньої маршрутизації ……………………………..51
       4. Використання Cloud VPN для підключення мереж VPC ……….…….51
       5. Використання Cloud Interconnect …………………………………........52
       6. Характеристика віртуальних пристроїв для керування трафіком ........53
    3. Дизайн, орієнтований на безпеку ………………………..................................54
    4. Інтеграція принципів проектування ……………………………………..……62
12. **АНАЛІЗ РЕЗУЛЬТАТІВ ПРОЕКТУВАННЯ ХМАРНОЇ МЕРЕЖІ** ...…68
13. Оцінка продуктивності ……………………………………………………...…68
14. Оцінка масштабованості ………………………………………………………69
15. Оцінка безпеки …………………………………………………………………71
16. Результати дослідження ………………………………………………….……72
17. Рекомендації ……………………………………………………………………74

**ВИСНОВОК** ………………………………………………………………….………78

**ПЕРЕЛІК ПОСИЛАНЬ** ………………………………………………….……….…80

**ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ** ………………………………………………82

**ВСТУП**

Ця дипломна робота присвячена актуальній проблемі проектування мережевої архітектури для хмарних обчислень, зосереджуючись на їх продуктивності, масштабованості та безпеці.

Мета роботи полягає у розробці принципів побудови ефективної мережевої архітектури, що відповідає сучасним вимогам хмарних сервісів. Використовуючи теоретичні та експериментальні методи, дослідження спрямоване на аналіз існуючих рішень та визначення оптимальних параметрів проектування.

Наукова новизна відображається у використанні сучасних підходів до проектування мережевих архітектур.

Сучасний швидкозмінний інформаційний ландшафт обумовлює необхідність удосконалення мережевих архітектур для хмарних обчислень. Зростання обсягу даних, вимог до продуктивності та безпеки, а також поява нових технологій вимагають ретельного аналізу та проектування хмарних мереж.

Мета дослідження – проаналізувати вимоги при проектуванні хмарних мереж та розробити рекомендії впровадження продуктивності, масштабованості та безпеки.

Об’єкт дослідження – проектування мережевої архітектури для хмарних обчислень.

Предмет дослідження – мережева архітектура для хмарних обчислень.

Методи дослідження - методи аналізу, систематизації, класифікації, а також методи статистичного та літературного аналізу.

Завдання дослідження:

* провести аналіз чинників, що впливають на продуктивність, масштабованість та безпеку;
* дослідити взаємодії масштабованості, продуктивності та безпеки;
* розробити рекомендації для проектування хмарної мережі.

**1 ПОСТАНОВКА ЗАДАЧІ**

**1.1 Характеристика Моделей Хмарних Обчислень**

Цей розділ досліджує Інфраструктуру як Сервіс (IaaS)[3], Платформу як Сервіс (PaaS), Програмне забезпечення як Сервіс (SaaS) та різні моделі розгортання (публічні, приватні, гібридні).

Інфраструктура як Сервіс (IaaS) - IaaS надає віртуалізовані обчислювальні ресурси через інтернет, дозволяючи користувачам орендувати віртуальні машини, сховища та мережеву інфраструктуру. Ключовими характеристиками є масштабованість, гнучкість та модель ціноутворення "плати, як використовуєш". Користувачі мають контроль над базовою інфраструктурою.

Платформа як Сервіс (PaaS) - PaaS пропонує платформу, що дозволяє користувачам розробляти, запускати та управляти застосунками, не маючи справи з складнощами інфраструктури. Спрощені процеси розробки, автоматичне масштабування та акцент на розгортанні застосунків. Базова інфраструктура абстрагована.

Програмне забезпечення як Сервіс (SaaS) - SaaS доставляє програмні застосунки через інтернет на підписку, усуваючи потребу в локальній установці. Доступність з будь-якого пристрою з підключенням до Інтернету, автоматичні оновлення та зменшення обов'язків з технічного обслуговування для користувачів.

У врахуванні Інфраструктури як Сервіс (IaaS) важливість полягає у високому рівні контролю, гнучкості вибору програмного забезпечення та конфігураційних параметрів, а також масштабованості інфраструктури. Однак його недоліки пов'язані з відповідальністю користувачів за управління інфраструктурою, що може бути і складною, і часоємкою справою.

Платформа як Сервіс (PaaS) надає спрощені процеси розробки, спрощене розгортання та автоматичне масштабування в залежності від попиту. З іншого боку, його недоліки включають обмежений контроль над базовою інфраструктурою та потенційні обмеження у виборі програмного забезпечення.

Для Програмного забезпечення як Сервіс (SaaS) переваги включають доступність з будь-якого пристрою, автоматичні оновлення та зменшення витрат на технічне обслуговування. Однак він має недоліки, такі як обмежені опції настроювання та залежність від постачальника SaaS для оновлень та функцій.

Серед Хмарних сервісів можна охарактеризувати такі моделі розгортання[1]:

Публічний хмарний сервіс - Послуги надаються через інтернет та доступні для загальної громадськості. Економічний, масштабований та легко керується. Ресурси спільно використовуються сторонніми постачальниками.

Приватний хмарний сервіс - Інфраструктура хмари використовується виключно однією організацією, надаючи більший контроль над ресурсами. Підвищена безпека, настроюваність та контроль над оточенням. Зазвичай дорожчий, ніж публічний хмарний сервіс.

Гібридний хмарний сервіс - Поєднує в собі моделі публічного та приватного хмарних сервісів, дозволяючи обміну даними та застосунками між ними. Пропонує гнучкість, дозволяючи організаціям балансувати витрати та контроль на основі конкретних потреб.

**1.2. Мережна архітектура в хмарних обчисленнях**

Вступаючи в глибини мережної архітектури в хмарних обчисленнях, ми відкриваємо світ, де технологічні інновації та інженерні рішення сплітаються разом, створюючи основу для сучасного цифрового світу. Ця архітектура не лише підтримує безперервну та ефективну роботу хмарних сервісів, але й відіграє ключову роль у забезпеченні безпеки, масштабованості та високої доступності послуг. Розглянемо критичні компоненти та стратегії, які формують сучасну мережну архітектуру, що лежить в основі хмарних обчислень, та оцінимо її вплив на загальну ефективність та гнучкість хмарних систем.

Віртуалізація є основним елементом, що абстрагує фізичні ресурси у віртуальні екземпляри. Це поліпшує використання ресурсів та динамічне їх розподіл. Найкращі практики включають ефективне управління гіпервізорами, оптимізацію розміщення віртуальних машин та мінімізацію конфліктів ресурсів.

Балансування навантаження критично для розподілу мережевого трафіку між серверами, балансування навантаження гарантує продуктивність та високу доступність. У літературі висловлюється підтримка динамічних алгоритмів балансування навантаження, врахування різних робочих навантажень та їх вплив на продуктивність системи.

Різноманітні мережеві протоколи лежать в основі мережевої архітектури хмар, полегшуючи спілкування між віртуалізованими екземплярами та службами. Найкращі практики включають в себе врахування безпеки при виборі протоколу, оптимізацію для комунікації з низькою затримкою та дотримання стандартів галузі[7].

Масштабованість передбачає роботу зі зростаючими навантаженнями та вимогами до ресурсів за допомогою горизонтального або вертикального масштабування. Найкращі практики акцентують на динамічному масштабуванні в залежності від попиту, ефективному розподілі ресурсів та врахуванні безстатевої архітектури.

Гнучкість, пов'язана з масштабованістю, передбачає автоматичне надання та відчуження ресурсів в залежності від попиту для оптимального використання ресурсів та ефективності вартості. Найкращі практики включають автоматизовані політики масштабування, моніторинг в реальному часі для змін навантаження та ефективне використання інструментів постачальників хмарних послуг.

Безшовна інтеграція віртуалізації, балансування навантаження та мережевих протоколів є критичною. Найкращі практики включають стандартизовані протоколи комунікації, всебічне розуміння взаємозв'язків та безпроблемну інтеграцію через API.

Безпека є найважливішою в мережевій архітектурі хмар. Найкращі практики включають протоколи шифрування, безпечні засоби управління доступом та регулярні аудити безпеки для захисту віртуалізованих середовищ, захисту даних під час передачі та оберігання від кіберзагроз.

**1.3 Продуктивність, масштабованість і безпека в хмарних мережах**

Продуктивність, масштабованість і безпека в хмарних мережах є ключовими факторами, які визначають ефективність та надійність хмарних обчислювальних сервісів.

Продуктивність у контексті хмарних систем відіграє ключову роль, оскільки вона безпосередньо впливає на ефективність та здатність системи задовольнити потреби користувачів. У цьому розділі розглядаються декілька важливих аспектів продуктивності в хмарних обчисленнях[10].

Серед аспектів продуктивності мережі можна розглянути:

* швидкість обробки даних;
* час відгуку мережі;
* пропускна здатність;
* оптимізація ресурсів;
* доступність ресурсів;
* оптимізація мережевих протоколів;
* моніторинг та аналітика.

Швидкість обробки даних - це один з найважливіших аспектів продуктивності хмарної системи. Вона визначає, наскільки швидко система може обробляти запити користувачів. Оптимізація алгоритмів та використання ефективних обчислювальних ресурсів забезпечують високу швидкість обробки.

Час відгуку мережі - це час, який проходить від моменту надходження запиту до моменту отримання відповіді. Він важливий для користувачів, оскільки визначає, наскільки швидко вони отримують відповіді на свої запити. Час відгуку може залежати від багатьох факторів, включаючи географічне розташування дата-центрів, оптимізацію мережі та пропускну здатність.

Пропускна здатність мережі визначає, скільки даних може бути передано через мережу за певний час. Вона важлива для передачі великих обсягів даних та для додатків, які вимагають високої пропускної здатності, наприклад, стрімінг відео або великі бази даних[4].

Ефективне використання обчислювальних ресурсів, таких як процесори, пам'ять та дисковий простір, забезпечує вищу продуктивність. Автоматизоване масштабування та балансування навантаження допомагають оптимізувати використання ресурсів.

Неперервний доступ до обчислювальних ресурсів та додатків є критичним для забезпечення високої продуктивності. Стратегії відновлення після збоїв та висока надійність системи забезпечують, що ресурси завжди доступні.

Використання сучасних, ефективних мережевих протоколів може знизити затримки та підвищити пропускну здатність. Протоколи, такі як HTTP/2, QUIC та інші оптимізовані рішення, можуть значно покращити загальну продуктивність мережі.

Постійний моніторинг та аналітика дозволяють виявляти та усувати мережеві збої та затримки, підтримуючи високу продуктивність. Збір метрик та використання аналітики для оптимізації ресурсів є ключовими для підтримання високої продуктивності.

Масштабованість у хмарних мережах є одним з найважливіших атрибутів, оскільки вона забезпечує гнучкість та ефективність хмарних систем у відповіді на змінювані потреби користувачів. У цьому розділі розглядаються ключові аспекти масштабованості у хмарних мережах.

Масштабованість у хмарних мережах означає, що система може швидко збільшувати або зменшувати свої ресурси для адаптації до коливань в запитах та навантаженні. Це може включати додавання або видалення віртуальних машин, збільшення обсягу сховища, або розширення пропускної здатності мережі.

Існує два основних типи масштабування: горизонтальне та вертикальне. Горизонтальне масштабування включає додавання більшої кількості вузлів (наприклад, серверів) до системи, тоді як вертикальне масштабування означає підвищення потужності існуючих вузлів (наприклад, збільшення RAM або CPU)[17].

Багато хмарних платформ надають можливість автоматичного масштабування, яке регулює ресурси відповідно до поточного навантаження без втручання людини. Це забезпечує, що ресурси доступні, коли вони потрібні, і зменшуються, коли попит знижується, оптимізуючи витрати та ефективність.

Масштабованість вимагає гнучкої архітектури, яка може підтримувати розширення без втрати продуктивності. Мікросервісна архітектура, розподілені системи та контейнеризація часто використовуються для забезпечення такої гнучкості.

Ефективне управління ресурсами є ключем до масштабованості, включаючи алокацію, моніторинг та оптимізацію ресурсів. Системи управління ресурсами дозволяють адміністраторам визначати політики масштабування та контролювати використання ресурсів.

Важливо, щоб при масштабуванні системи продуктивність залишалася стабільною та передбачуваною. Це означає, що система повинна забезпечувати однаковий рівень обслуговування незалежно від масштабу.

Масштабованість повинна бути економічно виправданою. Оптимізація витрат пов'язана з масштабуванням є важливою для багатьох організацій. Хмарні провайдери часто пропонують гнучкі платіжні моделі, які дозволяють організаціям платити лише за використані ресурси[21].

Безпека є одним з найважливіших аспектів хмарних мереж. Вона включає захист даних, інфраструктури та мережевих операцій від несанкціонованого доступу, витоку даних, атак та інших загроз.

Шифрування даних є одним з найефективніших способів захисту даних від несанкціонованого доступу. Воно дозволяє перетворити дані в нечитабельний формат, який може бути розшифрований лише за допомогою спеціального ключа.

Захист ідентичності та доступу є важливим для запобігання несанкціонованому доступу до хмарних систем. Він включає використання сильних паролів, багатофакторної аутентифікації та інших методів захисту.

Постійний моніторинг та виявлення загроз є важливим для виявлення та усунення потенційних загроз безпеці. Він включає використання систем виявлення вторгнень (IDS), систем виявлення аномалій (IDS) та інших інструментів безпеки.

Дотримання законодавчих та нормативних вимог є важливим для захисту даних та конфіденційності користувачів. Воно включає дотримання таких вимог, як GDPR, CCPA та інших нормативних актів[20].

**2 АНАЛІЗ ВИМОГ**

**2.1 Вимоги до продуктивності**

Продуктивність у мережі хмари впливає на кілька ключових чинників, кожен з яких відіграє важливу роль у забезпеченні ефективності та ефективності операцій. Цей розділ глибоко аналізує фактори, що безпосередньо впливають на продуктивність в мережі хмари.

Продуктивність мережі хмари значно визначається розсудливим розподілом обчислювальних ресурсів. Ефективне розподілення віртуальних машин, контейнерів та інших ресурсів забезпечує збалансоване навантаження, запобігає вузьким місцям в ресурсах та оптимізує загальну продуктивність системи.

Масштабованість є невід'ємною умовою для продуктивності в динамічному хмаровому середовищі. Системи повинні безшовно масштабувати ресурси вгору чи вниз в залежності від попиту, запобігаючи недо- чи перевикористанню і дозволяючи організаціям ефективно адаптуватися до змінних навантажень[11].

Автоматизація процесів надання ресурсів та їх управління сприяє продуктивності, мінімізуючи ручне втручання. Автоматизоване масштабування, балансування навантаження та оптимізація ресурсів покращують ефективність операцій та реагування на коливання робочого навантаження.

Продуктивність покращується за допомогою оптимізації рішень щодо зберігання даних. Ефективні механізми зберігання, включаючи дедуплікацію та стиснення даних, сприяють економії витрат та покращенню продуктивності при витягуванні даних.

Доступність даних є ключовою для продуктивності. Хмарові мережі повинні забезпечувати швидке та безпечне витягування даних, щоб користувачі та додатки могли швидко отримувати необхідну інформацію.

Продуктивність безпосередньо залежить від швидкості обробки даних. Високопродуктивні можливості обробки даних, включаючи паралельну обробку та розподілене обчислення, сприяють ефективному аналізу даних та продуктивності додатків[.

Розробка інтуїтивних інтерфейсів користувача є ключовою для продуктивності. Зручні інструментальні панелі та інтерфейси спрощують навігацію та взаємодію, скорочуючи час навчання та підвищуючи ефективність користувача.

Продуктивність підтримується потужними інструментами співпраці, що інтегруються в хмарову мережу. Функції співпраці в реальному часі, обмін документами та засоби зв'язку сприяють безперервному командному взаємодії та обміну інформацією[19].

Забезпечення доступності та відгуку хмарових послуг є критичним для продуктивності користувача. Швидкі часи реакції та стабільна доступність послуг сприяють позитивному досвіду користувача та ефективному виконанню завдань.

**2.2 Вимоги до масштабованості**

Масштабованість, основний аспект хмарних обчислень, вимагає глибокого розуміння викликів та необхідностей для проектування міцної мережевої архітектури. Ключовим є динамічне виділення ресурсів, що гарантує можливість динамічного виділення або звільнення ресурсів в залежності від попиту. Гнучкість в мережевій архітектурі відіграє важливу роль, дозволяючи масштабуванню обчислювальних ресурсів, таких як віртуальні машини та сховище, розширюватися чи зменшуватися відповідно до змінних робочих навантажень. Механізми автомасштабування, включаючи визначені політики масштабування та спрацьовувачі на основі метрик, таких як використання ЦП чи вхідний мережевий трафік, є необхідними для коригування ресурсів у реальному часі. Для масштабованості потрібна ефективна служба оркестрації, яка дозволяє координацію та управління розгортанням служб по розподілених системах[8].

Масштабування продуктивності важливе для збереження оптимальної продуктивності системи при її масштабуванні. Балансування навантаження, критичний аспект, включає механізми, що розподіляють вхідний трафік між кількома серверами для запобігання перевантаженню та забезпечення оптимального використання ресурсів. Часто використовуються парадигми розподіленого обчислення для обробки більших робочих навантажень без втрати продуктивності. Крім того, ефективні стратегії кешування сприяють продуктивності шляхом зменшення зайвої обробки та підвищення часів відгуку.

Врахування вартості є невід'ємною частиною вимог до масштабованості. Стратегії економічного масштабування є важливими, оптимізуючи виділення ресурсів та їх звільнення для уникнення непотрібних витрат. Постійний моніторинг використання ресурсів та оптимізація вартості є важливим. Впровадження інструментів та практик для аналізу використання ресурсів дозволяє приймати обгрунтовані рішення для управління витратами. Моделі виділення ресурсів за вимогою забезпечують, що ресурси виділяються лише тоді, коли це потрібно, запобігаючи непотрібним витратам під час періодів низького попиту.

**2.3 Вимоги безпеки**

Безпека є високою пріоритетом в мережевій архітектурі хмари, захищаючи дані, ідентифікацію користувачів і забезпечуючи відповідність законодавчим нормам. Ключові аспекти безпеки в мережевій архітектурі хмари включають безпеку даних, управління ідентифікацією та доступом, а також врахування вимог щодо відповідності та юридичні аспекти[13].

Щодо безпеки даних, суттєвими є надійні механізми шифрування для даних під час передачі та збереження, дрібні засоби контролю доступу, такі як контроль доступу на основі ролей (RBAC), а також безпечні засоби резервного копіювання та відновлення. Ці заходи забезпечують конфіденційність, цілісність даних та їх доступність у випадку інциденту з безпекою.

У сфері управління ідентифікацією та доступом, багаторівнева аутентифікація (MFA) додає додатковий рівень безпеки, федерація ідентичностей дозволяє здійснювати безпечну аутентифікацію в різних системах, а управління привілеями доступу (PAM) забезпечує обмежений та моніторований доступ для облікових записів з привілеями.

Відповідність індустрійним стандартам, таким як HIPAA, GDPR чи PCI DSS, є критичною. Дотримання юридичних норм, пов'язаних з місцеперебуванням даних, урахування того, де дані можуть бути збережені та оброблені, є важливим для уникнення юридичних ускладнень. Надійний план реагування на інциденти, включаючи механізми виявлення та реагування на інциденти з безпекою, а також дотримання юридичних вимог до звітування про інциденти, сприяють прозорості та обліковості.

**2.4 Інтеграція вимог**

Продуктивність, масштабованість і безпека в мережевій архітектурі хмари взаємопов'язані і вимагають збалансованого підходу, який враховує можливі компроміси та визначає синергії для оптимального загального дизайну.

Щодо компромісів досягнення високої продуктивності може включати спрощення доступу користувачів, але це повинно бути збалансовано з урахуванням аспектів безпеки для запобігання несанкціонованому доступу. Швидка масштабованість за допомогою автоматизованого забезпечення не повинна компрометувати безпеку, що вимагає масштабованих механізмів, розроблених для запобігання несанкціонованому доступу. Крім того, вартість впровадження заходів з безпеки, досягнення масштабованості чи підвищення продуктивності повинна бути ретельно розглянута, важачи довгострокові переваги порівняно із початковими витратами.

Синергії можна знайти в практиках, таких як Управління ідентифікацією та доступом (IAM), де надійні практики сприяють як безпеці, так і продуктивності. Засоби шифрування, розроблені для захисту даних, також сприяють масштабованості, а відповідність юридичним нормам не лише забезпечує юридичну відповідність, але й сприяє безпеці. Автоматизовані механізми масштабування, що підтримують еластичність, можуть сприяти як масштабованості, так і продуктивності при безпечному проектуванні. Ефективне розподіл ресурсів підтримує як масштабованість, так і продуктивність, користуючись оптимізованою доступністю ресурсів для користувачів[17].

**2.5 Випадки та сценарії використання**

У сценарії, де користувач потребує доступу до критичних ресурсів, збережених у хмарі, під час пікового періоду використання, мережева архітектура розроблена для забезпечення швидкого доступу до даних, зберігаючи високу продуктивність користувача. Це досягається за допомогою впровадження засобів контролю доступу для гарантії того, що користувачі можуть отримувати лише авторизовані ресурси, таким чином зберігаючи безпеку даних.

У іншому випадку, пов'язаному з користувацькою співпрацею в хмарі, члени команди, розташовані в різних місцях, співпрацюють над проектом. Тут мережева архітектура сприяє продуктивності через інтеграцію інтуітивних інструментів та інтерфейсів для співпраці. Одночасно впроваджуються безпечні механізми управління ідентифікацією та доступом для надання авторизованим членам команди доступу, забезпечуючи безпеку даних.

У сценаріях, де існує сезонний попит на послуги електронної комерції, торговельна платформа, яка переживає зростання трафіку під час сезонної розпродажу, використовує автоматизовані механізми масштабування. Ці механізми динамічно надають додаткові ресурси для обробки збільшеного трафіку, забезпечуючи ефективність вартості шляхом додавання ресурсів лише за потреби, тим самим оптимізуючи витрати.

Так само в сценарії безперервного розгортання та масштабування для зростаючої стартапу, мережева архітектура підтримує еластичність для обробки зростаючих навантажень. Механізми масштабування продуктивності, такі як балансування навантаження та розподілений обчислень, впроваджені для підтримання оптимальних рівнів продуктивності під час операцій масштабування[1].

У випадках, пов'язаних з інцидентами безпеки, такими як спроба несанкціонованого доступу до чутливих даних у хмарі, заходи безпеки, такі як багаторівнева аутентифікація та засоби контролю доступу, діють для запобігання несанкціонованому доступу. План реагування на інциденти швидко активується для розслідування та зменшення спроби несанкціонованого доступу.

У випадку порушення даних, де чутлива інформація клієнта компрометована, мережева архітектура надає пріоритет безпеці даних через шифрування та безпечні механізми резервного копіювання. Крім того, план реагування на інциденти включає процедури повідомлення про порушення даних відповідно до юридичних норм, забезпечуючи прозорість та відповідальність. Ці практичні сценарії та ситуації ілюструють, як вимоги до продуктивності, масштабованості та безпеки застосовуються в різних реальних ситуаціях в мережевій архітектурі хмари.

**3 ДОСЛІДЖЕННЯ ПРИНЦИПІВ ДИЗАЙНУ ХМАРНОЇ МЕРЕЖІ**

Цей розділ зосереджується на аналізі ключових принципів, які формують дизайн мережевої архітектури для хмарних обчислень. Розглянемо три основні орієнтації дизайну: продуктивність, масштабованість та безпека. Кожен аспект розглядається детально, починаючи з проектування, орієнтованого на продуктивність, який зосереджується на максимізації ефективності ресурсів та оптимізації мережевих операцій. Далі йде аналіз масштабованості, що включає стратегії динамічного розподілу ресурсів та гнучкого реагування на змінні вимоги. Нарешті, розглядається безпека, яка включає заходи захисту даних та управління доступом. Завершується розділ інтеграцією цих принципів для створення збалансованої та ефективної мережевої архітектури[7].

**3.1 Дизайн, орієнтований на продуктивність**

У цьому розділі були наведені принципи дизайну, спрямовані на максимізацію продуктивності в архітектурі хмарної мережі. Акцент робиться на створенні середовища, яке оптимізує використання ресурсів, використовує автоматизацію та оркестрацію для оптимізації операцій та використовує підхід до дизайну, орієнтованого на користувача. Для дослідження було обрано хмарний провайдер Google Cloud, так як він є одним із найбільшим постачальником хмарних послуг у світі.

Принципи ефективності ресурсів включають динамічне розподіл ресурсів, використання технологій віртуалізації та контейнеризації, імплементацію моніторингу та оптимізацію ресурсів. Динамічний розподіл ресурсів підтримує вимоги робочого навантаження за допомогою автоматичного масштабування, зменшуючи затори. Віртуалізація та контейнеризація покращують ізоляцію та гнучкість, а інструменти моніторингу ресурсів та передбачувальна аналітика оптимізують розподіл ресурсів прогнозуюче.

Розглянемо, як в Google Cloud досліджується продуктивність віртуальних машин та які компоненти продуктивності будуть розглянуті. Важливо визначити, що продуктивність віртуальних машин в Google Cloud це сукупність факторів, які впливають на їхню пропускну здатність та ефективність роботи[6].

Google Cloud враховує пропускну здатність для екземпляра віртуальної машини (VM), а не для мережевого інтерфейсу (vNIC) чи IP-адреси. Швидкість виходу визначається типом віртуальної машини, але важливо розуміти, що досягти максимально можливої швидкості виходу можна лише в конкретних умовах.

Додаткові віртуальні мережеві інтерфейси (vNIC) та IP-адреси на них не збільшують пропускну здатність для віртуальної машини. Наприклад, навіть якщо віртуальна машина має два мережеві адаптери, як у випадку n1-standard-8, загальна вихідна пропускна здатність залишається обмеженою на рівні 16 Гбіт/с, а не 16 Гбіт/с для кожного vNIC.

Google Cloud регулює вихідну пропускну здатність, встановлюючи максимальну швидкість виходу для кожної віртуальної машини в залежності від типу цієї машини та призначення пакетів. Це визначається на основі доступу до пункту призначення пакетів через маршрути у мережі VPC або поза нею. Вихідна пропускна здатність включає в себе дані, що передаються через всі мережеві інтерфейси віртуальної машини, а також дані, які передаються на постійні диски, приєднані до цієї віртуальної машини.

Максимальна вихідна пропускна здатність для віртуальної машини зазвичай становить 2 Гбіт/с на кожне віртуальне CPU (vCPU), хоча можливі деякі варіації і винятки залежно від конкретної серії віртуальних машин. У таблиці 1 наведено діапазон максимальних обмежень для вихідної пропускної здатності для трафіку, який маршрутизується в межах мережі VPC, специфічно для рівня стандартної мережі, і не враховуючи мережевої продуктивності рівня VM Tier\_1[16].

Таблиця 1 – Характеристика пропускної здатності серій машин

|  |  |  |
| --- | --- | --- |
| Серія Віртуальної  машини | Найнижчий максимальний вихідний ліміт на віртуальну машину | Найвищий максимальний вихідний ліміт на віртуальну машину |
| E2 | 1 Gbps | 16 Gbps |
| C3 | 23 Gbps | 100 Gbps |
| C3D | 20 Gbps | 100 Gbps |
| T2D | 10 Gbps | 32 Gbps |
| N2,C2, N2D, and C2D | 10 Gbps | 32 Gbps |
| H3 | N/A | 200 Gbps |
| N1 | 10 Gbps | 16 - 32 Gbps |

Максимальна вихідна пропускна здатність віртуальної машини не є статичним параметром і може змінюватися залежно від різних чинників. Ось деякі з цих факторів:

* тип гостьового драйвера ethernet: gvnic може забезпечити кращу продуктивність в порівнянні з мережевим інтерфейсом virtio;
* розмір пакета: розмір пакетів може впливати на пропускну здатність;
* накладні витрати на протокол: різні протоколи можуть мати свої накладні витрати, що впливають на швидкість передачі даних;
* кількість потоків: багато потоків можуть конкурувати за пропускну здатність;
* налаштування драйвера ethernet гостьової ос: параметри, такі як розвантаження контрольної суми та розвантаження сегментації tcp (tso), можуть впливати на продуктивність;
* перевантаження мережі: зайнятість мережі може обмежувати пропускну здатність;
* конфлікт із постійними дисками: у випадку конкуренції між записом на постійний диск та іншим мережевим трафіком, 60% пропускної здатності надається для запису на диск, і 40% залишається для іншого мережевого трафіку.

Для досягнення максимальної вихідної пропускної здатності для віртуальної машини потрібно виконати наступні дії:

* увімкнути продуктивність мережі tier\_1 для віртуальних машин з великими машинами загального призначення та оптимізованими для обчислень;
* використовувати найбільший блок максимальної передачі (mtu) мережі vpc, який підтримується вашою мережевою топологією, для зменшення накладних витрат на заголовки пакетів і збільшення пропускної здатності корисних даних.

Виходячи з точки зору віртуальної машини-відправника і доступних IP-адрес призначення, Google Cloud обмежує вихідний трафік з урахуванням кількох правил, включаючи максимальну вихідну пропускну здатність для кожної віртуальної машини та маршрутизацію міжрегіонального трафіку.

Вихідна пропускна здатність для кожної віртуальної машини: максимальна пропускна здатність для всіх з’єднань від віртуальної машини до пунктів призначення за межами мережі VPC – це менша максимальна вихідна пропускна здатність для однієї віртуальної машини та одна з таких характеристик[10]:

* + 25 Гбіт/с, якщо мережу Tier\_1 увімкнено;
  + 7 Гбіт/с, якщо мережу Tier\_1 не ввімкнено;
  + 1 Гбіт/с для віртуальних машин H3.

Наприклад, незважаючи на те, що примірник n2-standard-16 має максимальну вихідну пропускну здатність на віртуальну машину 32 Гбіт/с, вихідна пропускна здатність кожної віртуальної машини від примірника n2-standard-16 до зовнішніх пунктів призначення становить або 25 Гбіт/с, або 7 Гбіт/с, залежно від того, чи ввімкнено мережу Tier\_1.

Максимальна пропускна здатність для кожного унікального 5-кортежного підключення від віртуальної машини до пункту призначення за межами мережі VPC становить 3 Гбіт/с, за винятком H3, де вона становить 1 Гбіт/с.

Максимальна пропускна здатність для всіх з’єднань від віртуальних машин у кожному регіоні проекту до місць призначення за межами мережі VPC визначається квотами вихідної пропускної здатності Інтернету проекту.

Google Virtual NIC (gVNIC) — це інтерфейс віртуальної мережі, розроблений спеціально для Compute Engine. gVNIC є альтернативою драйверу Ethernet на основі virtIO.

Будучи мережевим інтерфейсом наступного покоління, який прийшов на зміну VirtIO, gVNIC замінює VirtIO-Net як єдиний підтримуваний мережевий інтерфейс у Compute Engine для всіх нових типів машин (покоління 3 і далі)[18]. Для нових машин і мережевих функцій потрібен gVNIC замість VirtIO. Використання gVNIC як сучасного інтерфейсу введення-виведення з віртуальними машинами Compute Engine дозволяє:

* + забезпечити кращу продуктивність;
  + покращення узгодженості шляхом зменшення проблем із шумними сусідами;
  + представлення нових мережевих функціональних можливостей, на які не здатний virtio;
* gvnic підтримується та рекомендовано для всіх сімейств машин, типів машин і поколінь.

gVNIC потрібен для підтримки:

* + пропускної здатності від 50 до 200 гбіт/с із віртуальними машинами, які підтримують продуктивність мережі tier\_1 для кожної віртуальної машини;
  + пропускної здатності від 50 до 100 гбіт/с із віртуальними машинами, які мають підключені графічні процесори, що допомагає покращити продуктивність розподілених робочих навантажень;
  + конфіденційні віртуальні машини, віртуальні машини серії машин третього покоління та віртуальні машини, які працюють на платформі центрального процесора arm.

**3.1.1 Оптимізація TCP для продуктивності мережі**

Сучасна архітектура мікросервісів виступає за те, що розробники повинні створювати невеликі сервіси з єдиною відповідальністю. Служби повинні обмінюватися даними за допомогою TCP або UDP, виходячи з очікуваної надійності системи. Тому для систем на основі мікросервісів важливо надійно спілкуватися з низькою затримкою.

Google Cloud забезпечує як надійність, так і низьку затримку, забезпечуючи глобальну мережу, що означає, що користувачі програми також можуть вийти на глобальний рівень. Наявність глобальної мережі означає, що можна створити мережу віртуальної приватної хмари (VPC), яка охоплює регіони та зони. Програми можуть підключатися один до одного в різних регіонах і зонах, не виходячи з мережі Google Cloud.

Програми, написані для традиційного середовища центру обробки даних, можуть демонструвати низьку продуктивність, якщо їх перемістити в гібридне хмарне середовище, тобто коли деякі компоненти програми працюють у корпоративному центрі обробки даних, а інші — у хмарі. Низька продуктивність може бути результатом ряду факторів.

TCP використовує механізм вікон, щоб запобігти швидкому відправнику перевищити повільний отримувач. Одержувач повідомляє, скільки даних має надіслати відправник, перш ніж відправник повинен буде чекати оновлення вікна від одержувача. Як наслідок, якщо програма-одержувач не може отримати дані під час з’єднання, існує обмеження на кількість даних, які можна поставити в чергу в очікуванні програми.

Вікно TCP дозволяє ефективно використовувати пам'ять у системах надсилання та отримання. Оскільки програма-одержувач споживає дані, оновлення вікон надсилаються відправнику. Найшвидше оновлення вікна відбувається за одне проходження, що призводить до такої формули 1 для одного з обмежень продуктивності масової передачі підключення TCP:

Пропускна здатність <= розмір вікна / час затримки (RTT). (1)

У оригінальному дизайні для TCP це вікно має максимальний розмір 65535 байт (64 КіБ - 1). Це був максимальний обсяг даних, який відправник міг надіслати до того, як відправник отримав оновлення вікна, щоб можна було надіслати більше даних.

З моменту появи протоколу TCP змінилися деякі ключові функції:

* + типова швидкість мережі зросла на чотири порядки;
  + типовий обсяг пам'яті в системі збільшився на чотири порядки.

Результатом першої зміни є те, що початкові розміри вікна TCP призвели до неефективного використання мережевих ресурсів. Відправник надсилав би дані за вікно з найкращою швидкістю, можливою в умовах мережі, а потім тривалий час простоював, очікуючи оновлення вікна TCP. Результатом другої зміни є те, що відправники та одержувачі можуть використовувати більше пам’яті для роботи в мережі, щоб усунути обмеження, виявлене першою зміною, як наведено у риснунку 1.

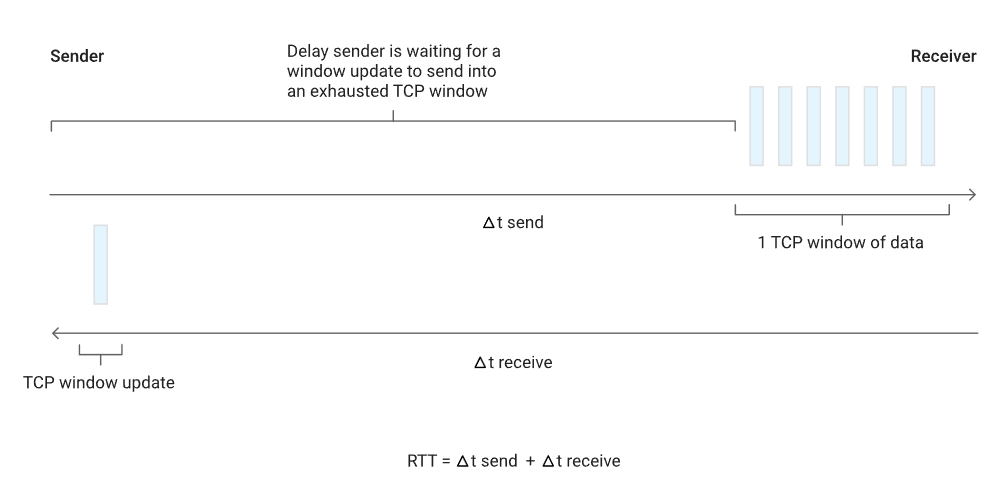
****

Рисунок 1 - Залежність затримки від розміру вікна

Відправник не може повністю використовувати мережу, оскільки він очікує оновлення вікна TCP перед надсиланням додаткових даних.

Рішення полягає в надсиланні більшої кількості даних за раз. Зі збільшенням пропускної здатності мережі більше даних може поміститися в трубу (мережу), а якщо труба стає довшою, то більше часу потрібно для підтвердження отримання даних. Це співвідношення відоме як добуток пропускної здатності на затримку (BDP). Це обчислюється як пропускна здатність, помножена на час проходження в обидві сторони (RTT), що призводить до значення, яке визначає оптимальну кількість бітів для надсилання, щоб заповнити канал, як наведено у формулі 2[17].

Обчислений BDP використовується як розмір вікна TCP для оптимізації.

Наприклад, якщо є мережа 10 Гбіт/с із RTT 30 мілісекунд, то для розміру вікна потрібно використовувати значення початкового розміру вікна TCP (65535 байт). Це значення не наближається до використання переваг пропускної здатності. Максимально можлива продуктивність TCP за цим посиланням така:

(65535 байт\* 8 біт/байт) = пропускна здатність \* 0.030 секунд  
пропускна здатність = (65535 байт\* 8 біт/байт) / 0.030 секунд  
пропускна здатність = 524280 біт / 0.030 секунд  
пропускна здатність = 17476000 біт/ секунд

Іншими словами, ці значення призводять до пропускної здатності, яка становить трохи більше 17 Мбіт/с, що є невеликою частиною можливостей мережі в 10 Гбіт/с.

Щоб усунути обмеження продуктивності, накладені початковим дизайном розміру вікна TCP, було введено розширення протоколу TCP, які дозволяють масштабувати розмір вікна до набагато більших значень. Масштабування вікон підтримує вікна розміром до 1 073 725 440 байт, або майже 1 ГіБ. Ця функція описана в RFC 7323 як параметр масштабування вікна TCP.

Розширення масштабу вікна розширюють визначення вікна TCP до використання 30 біт, а потім використовують неявний коефіцієнт масштабування для перенесення цього 30-бітного значення в поле 16-бітного вікна заголовка TCP. Щоб перевірити, чи ввімкнено цю функцію в системах на базі Linux, можна скористатися такою командою:

sudo sysctl net.ipv4.tcp\_window\_scaling

На всіх віртуальних машинах Google Cloud Linux цю функцію ввімкнено за умовчанням. Повернене значення 1 означає, що параметр увімкнено. Якщо функцію вимкнено, її можна ввімкнути за допомогою такої команди:

sudo sysctl -w net.ipv4.tcp\_window\_scaling=1

Використали попередній приклад, щоб показати переваги масштабування вікна. Як і раніше, припустимо мережу 10 Гбіт/с із затримкою 30 мілісекунд, а потім обчислимо новий розмір вікна за цією формулою 3:

Підставивши значення, отримали наступне:

(10 Гб/с \* 30мс/1000сек) / 8біт/байт= розмір вікна  
(10000 Гб/с \* 0.030 сек) / 8 біт/байт= 37.5 MB

Збільшення розміру вікна TCP до 37 МБ може збільшити теоретичну межу продуктивності масової передачі TCP до значення, що наближається до можливостей мережі. Звичайно, багато інших факторів можуть обмежувати продуктивність, включаючи накладні витрати на систему, середній розмір пакета та кількість інших потоків, які спільно використовують посилання, але, як можна побачити, розмір вікна суттєво пом’якшує обмеження, накладені попереднім обмеженим розміром вікна[6].

**3.1.2** **Варіанти балансування навантаження**

Залежно від типу трафіку, що надсилається до програми, існує кілька варіантів зовнішнього балансування навантаження, наведені у таблиці 3.2

Таблиця 2 – Характеристика Балансувальників наватнаження

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Опис | Принцип | Обсяг |
| Балансувальник навантаження зовнішніх програм | Підтримує трафік HTTP(S) і розширені функції, такі як відображення URL-адрес і розвантаження SSL  Використовується зовнішній проксі-сервер балансування мережевого навантаження для не-HTTP-трафіку на певних портах | Сеанс TCP або SSL (TLS) припиняється на передніх інтерфейсах Google (GFE), на межі мережі Google, а трафік передається на серверні частини. | Глобальний |
| Зовнішній наскрізний балансувальник навантаження мережі | Дозволяє трафіку TCP/UDP через будь-який порт проходити через балансувальник навантаження. | Доставляється за допомогою технології Maglev від Google для розподілу трафіку на сервери. | Регіональний |

Під час доступу до веб-сайту, розміщеного в us-central1, використали такі методи для перевірки затримки:

* + ping;
  + curl.

Хоча пінг ICMP є поширеним способом вимірювання доступності сервера, пінг ICMP не вимірює затримку кінцевого користувача. Щоб отримати додаткові відомості, переглянули додаткові ефекти затримки зовнішнього балансувальника навантаження програми.

Curl вимірює час до першого байта (TTFB). Повторно надіслали серверу команду curl. Порівнюючи результати, потрібно мати на увазі, що затримка оптоволоконних з’єднань обмежується відстанню та швидкістю світла в оптоволокні, яка становить приблизно 200 000 км на секунду.

Відстань між Франкфуртом, Німеччина, та Каунсіл-Блафсом, штат Айова (регіон us-central1), становить приблизно 7500 км. З прямим волокном між розташуваннями затримка в обидві сторони така:

7,500 км \* 2 / 200,000 км/с \* 1000 мс/с = 75 мс

Волоконно-оптичний кабель не йде по прямому шляху між користувачем і центром обробки даних. Світло на оптоволоконному кабелі проходить через активне та пасивне обладнання на своєму шляху. Затримка, яка приблизно в 1,5 рази перевищує ідеальну, або 112,5 мс, вказує на майже ідеальну конфігурацію[14].

Проведено порівняння балансування навантаження в таких конфігураціях:

* + відсутність балансування навантаження;
  + зовнішній наскрізний балансувальник мережевого навантаження;
  + зовнішній балансувальник прикладного навантаження.

У цьому сценарії програма складається з регіональної керованої групи примірників веб-серверів HTTP, як зазначено на рисунку 2. Оскільки програма покладається на звернення до центральної бази даних із низькою затримкою, веб-сервери мають бути розміщені в одному місці. Програма розгорнута в центральному регіоні США 1, а користувачі розподілені по всьому світу. Затримка, яку спостерігає користувач у Німеччині в цьому сценарії, ілюструє те, з чим можуть зіткнутися користувачі в усьому світі.

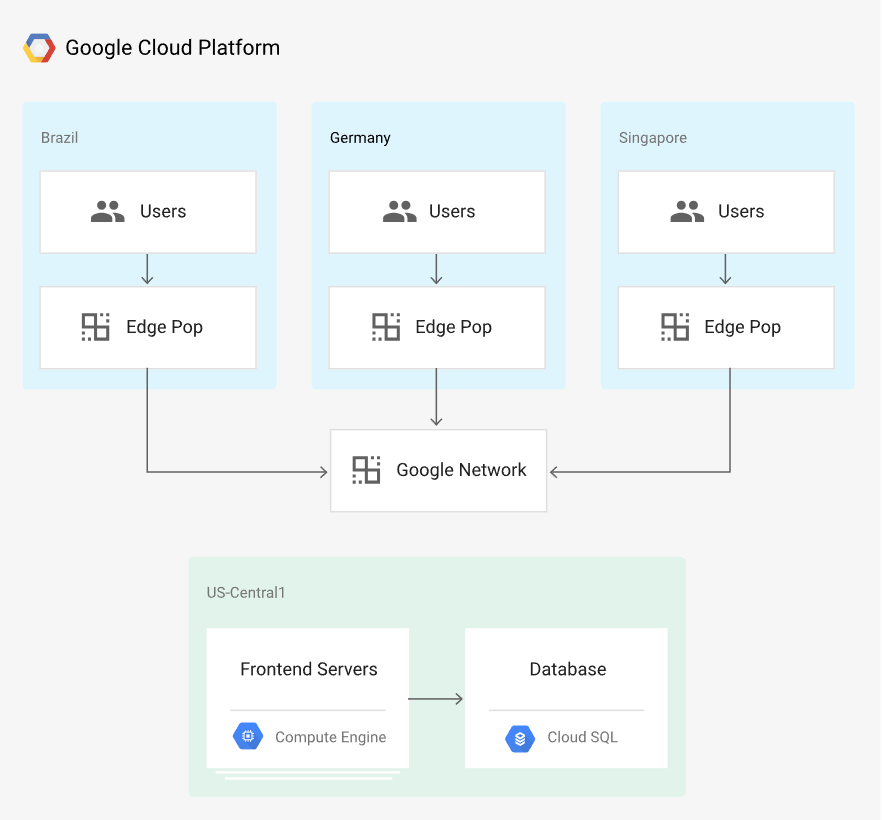
****

Рисунок 2 – Топологія мережі

Коли користувач робить HTTP-запит, якщо балансування навантаження не налаштовано, трафік переходить безпосередньо з мережі користувача на віртуальну машину (VM), розміщену на Compute Engine, як показано на рисунку 3. Для преміум-рівня трафік надходить у мережу Google у крайовій точці присутності (PoP), розташованій поблизу місцезнаходження користувача. Для стандартного рівня користувальницький трафік надходить у мережу Google через точку доступу, розташовану поблизу цільового регіону.

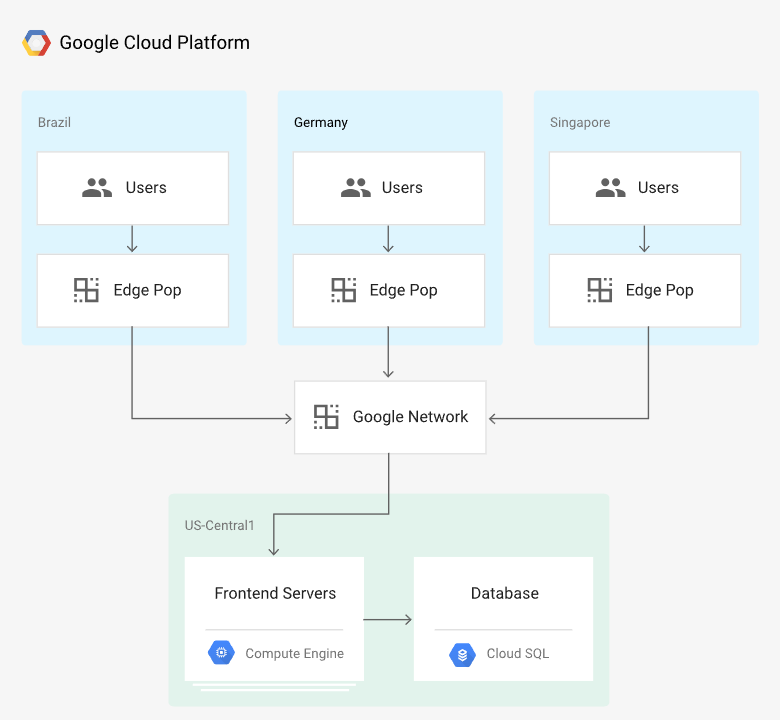
****

Рисунок 3 – Топологія мережі без балансування навантаження

У наведеній нижче таблиці показано результати тестування затримки системи без балансування навантаження:

Таблиця 3 – Тестування затримки системи

|  |  |  |
| --- | --- | --- |
| Метод | Результат | Мінімальна затримка |
| Перевірити IP-адресу віртуальної машини (відповідь надходить безпосередньо з веб-сервера) | мінімальна/середня/максимальна = 110.818/110.944/111.265 ms | 110 ms |
| TTFB **(**Час до першого байта) | 0.231, 0.232, 0.230, 0.229, 0.232 | 230 ms |

Затримка TTFB є стабільною, як показано на рисунку 4, де опрацьовано перших 500 запитів.****

Рисунок 4 – Графік затримки TTFB без балансування

Під час перевірки IP-адреси віртуальної машини відповідь надходить безпосередньо з веб-сервера[18]. Час відповіді від веб-сервера мінімальний порівняно з затримкою мережі (TTFB). Це пояснюється тим, що для кожного HTTP-запиту відкривається нове TCP-з’єднання. Початкове тристороннє рукостискання потрібне перед надсиланням відповіді HTTP, як показано на рисунку 5. Таким чином, спостережувана затримка близька до подвоєної затримки ping.

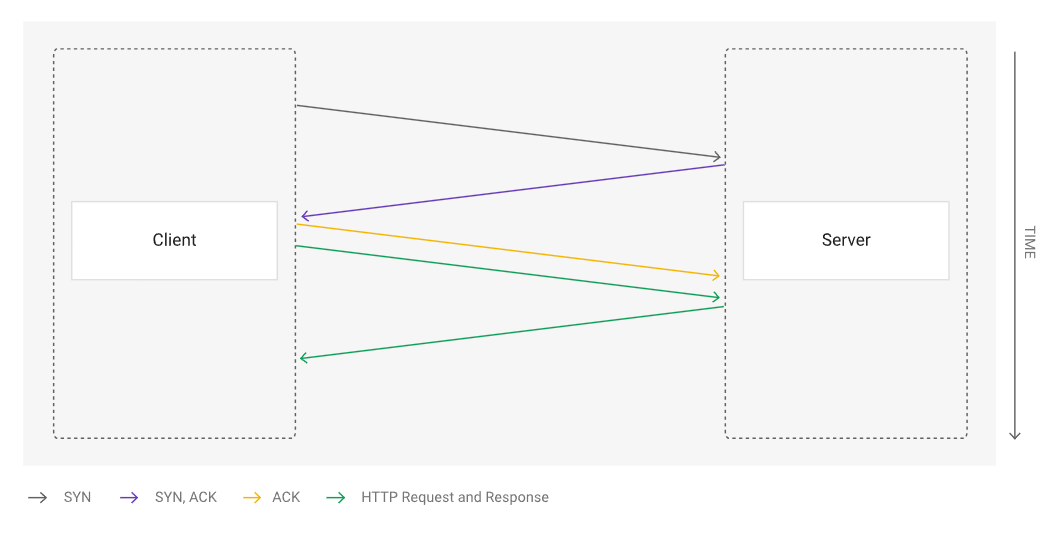
****

Рисунок 5 – Схема тристороннього рукостискання

Завдяки зовнішнім наскрізним балансувальникам мережевого навантаження запити користувачів усе ще надходять у мережу Google на найближчій межі PoP (у преміум-рівні). У регіоні, де розташовані віртуальні машини проекту, трафік спочатку проходить через зовнішній наскрізний балансувальник мережевого навантаження. Потім він пересилається без змін до цільової серверної віртуальної машини. Зовнішній наскрізний балансувальник мережевого навантаження розподіляє трафік на основі стабільного алгоритму хешування. Алгоритм використовує комбінацію портів джерела та призначення, IP-адреси та протоколу. Віртуальні машини слухають IP балансувальника навантаження та приймають трафік без змін[11].

У наведеній нижче таблиці 4 показано результати тестування затримки для конфігурації балансування мережевого навантаження.

Таблиця 4 – Тестування мережевого балансування навантаження

|  |  |  |
| --- | --- | --- |
| Метод | Результат | Мінімальна затримка |
| Команда «Ping» | мін/сер/макс/ = 110.658/110.705/110.756 мс | 110 мс |
| TTFB (Час до першого байту) | 0.231, 0.232, 0.230, 0.230, 0.232 | 230 мс |

Оскільки балансування навантаження відбувається в межах регіону, а трафік лише пересилається, немає значного впливу затримки порівняно з відсутністю балансувальника навантаження за допомогою зовнішніх балансувальників навантаження програм GFE проксі-трафік. Ці GFE знаходяться на краю глобальної мережі Google. GFE завершує сеанс TCP і підключається до серверної частини в найближчому регіоні, який може обслуговувати трафік

У наведеній нижче таблиці показано результати, тестування затримки для параметра балансування навантаження HTTP.

Таблиця 5 – Тестування HTTP балансування навантаження

|  |  |  |
| --- | --- | --- |
| Метод | Результат | Мінімальна затримка |
| Команнда «Ping» | мін/сер/макс = 1.163/1.195/1.229 мс | 1 мс |
| TTFB (Час до першого байту) | 0.123, 0.124, 0.126 | 124 мс |

Результати для зовнішнього балансувальника навантаження програм значно відрізняються. Під час перевірки зовнішнього балансувальника навантаження програми затримка в обидві сторони трохи перевищує 1 мс. Цей результат представляє затримку до найближчого GFE, який знаходиться в тому ж місті, що й користувач. Цей результат не відображає фактичну затримку, яку відчуває користувач під час спроби отримати доступ до програми, розміщеної в регіоні us-central1. Експерименти з використанням протоколів (ICMP), які відрізняються від протоколу зв’язку програми (HTTP), можуть ввести в оману[15].

Під час вимірювання TTFB початкові запити показують однакову затримку відповіді. Деякі запити досягають нижчої мінімальної затримки 123 мс, як показано на наступному графіку:

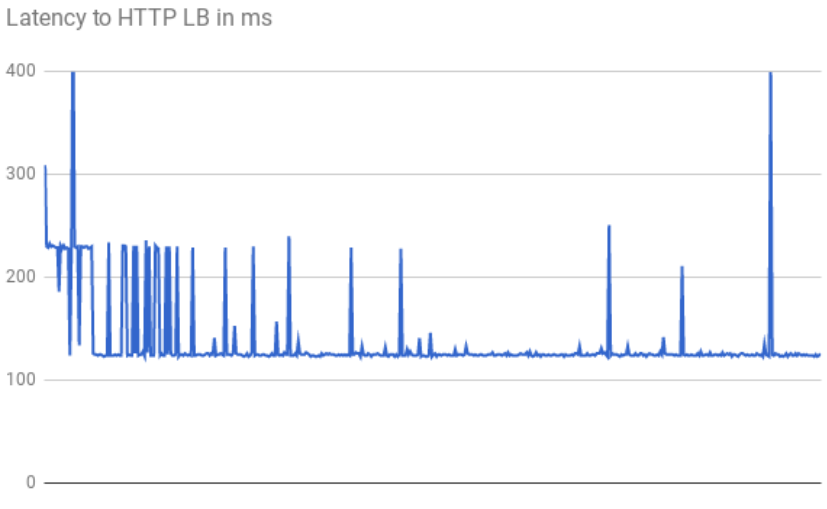
****

Рисунок 6 – Графік затримки HTTP балансувальника навантаження

Дві передачі між клієнтом і віртуальною машиною займають більше 123 мс навіть із прямим оптоволокном. Затримка менша, оскільки GFE проксі трафіку. GFE підтримують постійні підключення до серверних віртуальних машин. Таким чином, лише перший запит від певного GFE до певного сервера потребує тристороннього рукостискання.

Зовнішній балансувальник навантаження програми має меншу затримку для складних активів, ніж зовнішній наскрізний балансувальник навантаження мережі, тому що для завершення відповіді потрібно менше двосторонніх передач. Наприклад, коли користувач у Німеччині вимірює затримку через те саме з’єднання, багаторазово завантажуючи файл розміром 10 МБ, середня затримка для балансувальника навантаження зовнішньої мережі становить 1911 мс. Із зовнішнім балансувальником навантаження програми середня затримка становить 1341 мс. Це економить приблизно 5 поїздок туди й назад на запит. Постійні з’єднання між GFE та обслуговуючими серверними частинами зменшують вплив повільного запуску TCP.

Зовнішній балансувальник навантаження застосунків значно зменшує додаткову затримку для рукостискання TLS (зазвичай 1–2 додаткові зворотні передачі). Це пов’язано з тим, що зовнішній балансувальник навантаження застосунку використовує розвантаження SSL, і релевантною є лише затримка до межі PoP. Для користувача в Німеччині мінімальна спостережувана затримка становить 201 мс при використанні зовнішнього балансувальника навантаження програми проти 525 мс при використанні HTTP(S) через зовнішній наскрізний балансувальник навантаження мережі.

Зовнішній балансувальник навантаження застосунку дозволяє автоматично оновлювати сеанс користувача до HTTP/2. HTTP/2 може зменшити кількість необхідних пакетів, використовуючи вдосконалення бінарного протоколу, стиснення заголовків і мультиплексування з’єднань. Ці вдосконалення можуть зменшити спостережувану затримку навіть більше, ніж спостережувану при переході на зовнішній балансувальник навантаження програми. HTTP/2 підтримується поточними браузерами, які використовують SSL/TLS. Для користувача з Німеччини мінімальна затримка зменшилася з 201 мс до 145 мс при використанні HTTP/2 замість HTTPS.

Якщо частину трафіку, який ви обслуговуєте, можна кешувати, ви можете інтегруватися з Cloud CDN. Cloud CDN зменшує затримку, обслуговуючи ресурси безпосередньо на межі мережі Google. Cloud CDN також використовує оптимізацію TCP і HTTP від HTTP/2, згадану в розділі Додаткові ефекти затримки зовнішнього балансувальника навантаження програми.

Якщо вміст є статичним, можна зменшити навантаження на веб-сервери, обслуговуючи вміст безпосередньо з Cloud Storage через зовнішній балансувальник навантаження програм. Ця опція поєднується з Cloud CDN.

Розгортання веб-серверів у кількох регіонах поблизу користувачів може зменшити затримку, оскільки балансувальник навантаження автоматично спрямовує користувачів до найближчого регіону. Однак, якщо програма частково централізована, потрібно створити її так, щоб зменшити кількість міжрегіональних передач даних туди й назад.

Щоб зменшити затримку в програмах, слід перевірити будь-які віддалені виклики процедур (RPC), які обмінюються даними між віртуальними машинами. Ця затримка зазвичай виникає, коли програми обмінюються даними між рівнями або службами. Такі інструменти, як Cloud Trace, можуть допомогти зменшити затримку, спричинену запитами додатків.

Оскільки балансувальники навантаження мережі зовнішнього проксі базуються на GFE, вплив на затримку такий самий, як і зовнішній балансувальник навантаження програми. Оскільки зовнішній балансувальник навантаження програми має більше можливостей, ніж зовнішній балансувальник навантаження мережі проксі-сервера, ми рекомендуємо використовувати зовнішні балансувальники навантаження програми для трафіку HTTP(S)[16].

**3.2 Дизайн, орієнтований на масштабованість**

Цей розділ визначає стратегії дизайну, спрямовані на підвищення продуктивності мережевої архітектури у хмарному середовищі. Запропоновані принципи ставлять за мету оптимізацію швидкодії, зниження затримок та максимізацію використання ресурсів.

Принципи оптимізації швидкодії передбачають використання спеціалізованих апаратних засобів, таких як асиметричні процесори та апаратні прискорювачі, для обробки специфічних завдань мережевого трафіку. Додатково, розглядається можливість розподілу завдань між вузлами для підвищення паралелізму та збільшення загальної продуктивності.

Принципи зниження затримок включають у себе використання алгоритмів маршрутизації з мінімальними часами відгуку, розміщення ресурсів ближче до кінцевих користувачів, а також розгляд використання технологій кешування для зменшення часу доступу до даних[7].

Принципи максимізації використання ресурсів орієнтовані на оптимальне використання обчислювальної потужності та мережевих ресурсів. Це включає ефективну оркестрацію та управління віртуальними ресурсами, а також використання автоматизованих механізмів моніторингу та аналізу, що дозволяють реагувати на зміни використання ресурсів у реальному часі

Зокрема, управління ресурсами може включати в себе динамічне налаштування кількості ресурсів, призначених для конкретних сервісів, в залежності від попиту. Крім того, розглядається можливість використання контейнеризації та оркестрації контейнерів для ефективного розгортання та керування мікросервісами, забезпечуючи тим самим гнучкість та продуктивність системи.

Якщо у проекті, який потребує розгортання ресурсів у кількох ізольованих мережах VPC (наприклад, екземпляри віртуальної машини з кількома мережевими інтерфейсами), хост-проект має містити всі мережі VPC (Рисунок 7), які надають послуги. Це пов’язано з тим, що проект служби може приєднуватися лише до одного головного проекту.

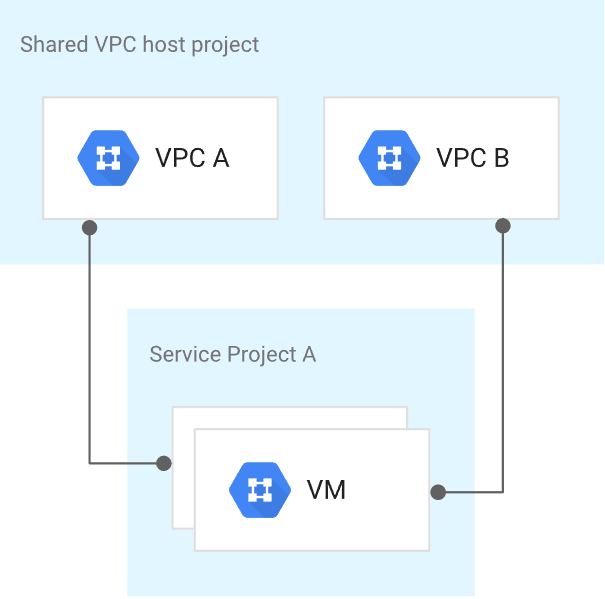
****

Рисунок 7 – Використання Shared VPC

Групування програм в меншу кількість підмереж із більшими діапазонами адрес визначається як стратегічний підхід до розподілу ресурсів в хмарному середовищі Google Cloud. Такий підхід дозволяє оптимізувати управління адресами та забезпечує більшу гнучкість у виборі діапазонів для програм.

Традиційно, корпоративні мережі розділяються на невеликі діапазони адрес з різних міркувань, таких як ідентифікація, ізоляція програм або збереження широкомовного домену. Проте, рекомендація полягає в групуванні програм одного типу в меншу кількість підмереж, з більшими діапазонами адрес в регіонах, де це є доцільним[12].

Важливо зауважити, що на відміну від інших мережевих середовищ, Google Cloud використовує програмно-визначений мережевий підхід (SDN), який забезпечує повну доступність мережі між усіма віртуальними машинами в глобальній мережі VPC. Кількість підмереж не впливає на маршрутизацію, оскільки вона реалізується через програмне забезпечення.

Для застосування конкретних політик маршрутизації чи правил брандмауера можна використовувати облікові записи служб або мережеві теги, а ідентифікація в Google Cloud не обмежується лише IP-адресами підмережі.

Окремі функції VPC, такі як хмарний NAT, приватний доступ Google, журнали потоків VPC і псевдонімні діапазони IP-адрес, можуть бути налаштовані для кожної підмережі, щоб надати додатковий контроль.

Для оптимізації створення, підтримки та розуміння мережі рекомендується почати з однієї мережі VPC для ресурсів із спільними вимогами. Це допомагає встановити кордон мережі VPC як периметр для потенційних проблем.

Додаткові мережі VPC можуть бути створені в залежності від таких факторів, як масштаб, безпека, фінансові міркування, операційні вимоги та керування ідентифікацією та доступом. Використання спільного VPC для адміністрування кількох робочих груп надає зручний інструмент для розширення простоти архітектури на кілька робочих груп.

Такий підхід може бути особливо ефективним для організацій із кількома командами, де Shared VPC надає централізований контроль та ефективне керування мережевими ресурсами.

Використання кількох хост-проектів визначається як основний підхід у випадках, коли вимоги до ресурсів перевищують квоту, надану одному проекту. У таких випадках рекомендується використовувати архітектуру з кількома хост-проектами, де кожен хост-проект має одну спільну мережу VPC. Цей підхід сприяє управлінню ресурсами та врахуванню квот на рівні проекту.

Наприклад, якщо сукупні вимоги до ресурсів перевищують квоту одного проекту, можна використовувати кілька хост-проектів, кожен із яких має власну спільну мережу VPC. Це дає можливість ефективно масштабувати ресурси та управляти ними.

Використання кількох хост-проектів також виправдане, якщо потрібні окремі політики адміністрування для кожної мережі VPC. Оскільки кожен проект має власну квоту, використовувати окремий хост-проект VPC для кожної мережі дозволяє ефективно розподіляти дозволи IAM та управляти безпекою на рівні проекту.

Наприклад, якщо розгортається дві мережі VPC (мережу VPC A та мережу VPC B) в одному хост-проекті, роль адміністратора мережі (networkAdmin) буде застосована до обох мереж VPC.

Для ефективного виходу за межі квот ресурсів VPC рекомендується створювати одну мережу VPC для кожного проекту. Це спрощує відображення збільшення квоти на рівні проекту для кожної мережі VPC.

**3.2.1** **Підключення кількох мереж VPC**

Одним із важливим кроком після прийняття рішення про створення кількох мереж VPC є підключення цих мереж VPC. Мережі VPC — це ізольовані простори орендарів у Andromeda SDN Google, але є кілька способів створити зв’язок між ними. Переваги та недлоліки методів підключення наведено в таблиці 6.

Таблиця 6 - Переваги та недоліки методів підключення

|  |  |  |
| --- | --- | --- |
| Метод підключення | Переваги | Недоліки |
| Використання Глобальних IP-адрес | - простота та ефективність;  - забезпечує глобальну доступність;  - підтримує різні моделі комунікації | - ціна;  - ризик кібератак;  - обмежена масштабованість. |
| Використання VPN (Віртуальної Приватної Мережі) | - забезпечує захист і шифрування;  - достатньо витратна ефективність;  - підтримує конфігурації "точка-точка" та "точка-багато". | - мала продуктивність;  - складність конфігурації. |
| Використання Interconnect (Підключення до Мережі) | - забезпечує високу продуктивність;  - підтримує великі об'єми трафіку;  - надає знижені затримки та більшу надійність. | -вимагає фізичного підключення до мережі google;  - вартість . |
| Використання Shared VPC (Спільна Мережа VPC) | - централізоване управління та контроль;  - забезпечує безпеку та конфігураційну стандартизацію. | - вимагає обережного управління дозволами та правами доступу;  - може потребувати більше часу на конфігурацію. |

Можна навети наступні практики для вибору методу підключення VPC[17]:

* використання глобальних ip-адрес, якщо важлива глобальна доступність;
* використання vpn, якщо потрібен захист і шифрування;
* використання interconnect, якщо потрібна висока продуктивність та надійність;
* використання shared vpc, якщо потрібно централізоване управління та контроль.

Потрібно обирати метод, який краще відповідає конкретним вимогам щодо вартості, продуктивності та безпеки. У кожному випадку важливо враховувати контекст проекту та визначати пріоритети відповідно до них.

**3.2.2 Використання мережевого пірингу VPC**

При необхідності підключення мережі VPC та неможливості використання спільного VPC слід використовувати мережевий піринг VPC.

Мережевий піринг VPC дає змогу двом мережам VPC внутрішньо з’єднуватися одна з одною через SDN Google, незалежно від того, належать вони до одного проекту чи однієї організації. Мережевий піринг VPC об’єднує площину керування та розповсюдження потоку між кожним вузлом, забезпечуючи ті самі характеристики пересилання, якби всі віртуальні машини були в одній мережі VPC. Під час однорангового аналізу мереж VPC доступні всі підмережі, діапазони псевдонімів IP-адрес і внутрішні правила переадресації, а кожна мережа VPC підтримує власний розподілений брандмауер. Піринг мережі VPC не є транзитивним.

Піринг мережі VPC є кращим методом підключення до мереж VPC з таких причин, адже вирішується проблема вузького місця шлюза (так зване вузьке горличко). Пересилання трафіку відбувається так, ніби віртуальні машини знаходяться в одній мережі VPC. Додатковою перевагоє є безкоштовне використання пірингу.

**3.2.3 Використання зовнішньої маршрутизації**

Якщо не потрібно використовувати приватні IP-адреса, можна використовувати зовнішню маршрутизацію із зовнішніми IP-адресами або шлюз NAT.

Під час розгортання мережі VPC маршрут до інтернет-шлюзу Google за замовчуванням надається з пріоритетом 1000. Якщо цей маршрут існує та віртуальній машині надається зовнішня IP-адреса, віртуальні машини можуть надсилати вихідний (вихідний) трафік через інтернет-шлюз Google. Також можна розгорнути служби за однією з багатьох загальнодоступних пропозицій балансування навантаження від Google, що дає змогу отримувати доступ до служб ззовні.

Віртуальні машини із зовнішніми адресами спілкуються одна з одною приватно через магістраль Google, незалежно від регіону та рівнів мережевих послуг. Слід використовуватиправила брандмауера Google Cloud, щоб контролювати зовнішній вхідний трафік до віртуальних машин.

Зовнішня маршрутизація є хорошим варіантом для цілей масштабування, але важливо розуміти, як публічна маршрутизація впливає на витрати.

**3.2.4 Використання Cloud VPN для підключення мереж VPC**

HA VPN надає керовану службу для підключення мереж VPC шляхом створення тунелів IPsec між наборами кінцевих точок. Якщо налаштовуються власні хмарні маршрутизатори за допомогою власних оголошень про маршрути, можна ввімкнути транзитивну маршрутизацію в мережах VPC і топології концентратора та стрілки. За допомогою Network Connectivity Center можна використовувати тунелі HA VPN як транзитну мережу між локальними мережами, як пояснюється в документації Cloud VPN.

Однак Cloud VPN може ввести обмеження продуктивності: Cloud VPN вимагає нижчого максимального блоку передачі (MTU) у мережі VPC через додаткові тунельні інкапсуляції, що обмежує пропускну здатність одного потоку.

**3.2.5 Використання Cloud Interconnect**

Cloud Interconnect розширює локальну мережу на мережу Google за допомогою високодоступного з’єднання з малою затримкою. Можна використовувати Dedicated Interconnect для підключення безпосередньо до Google або Partner Interconnect для підключення до Google через підтримуваного постачальника послуг.

Dedicated Interconnect надає високошвидкісну послугу L2 між Google і постачальником спільного розташування або локальним розташуванням. Це дозволяє використовувати локальне обладнання маршрутизації для маршрутизації між мережами VPC та використовувати існуючі локальні служби безпеки та перевірки для фільтрації всього трафіку між мережами VPC. Весь трафік між двома мережами VPC має додаткову затримку через додаткову зворотну передачу через локальну систему.

Partner Interconnect надає подібні можливості, а також здатність безпосередньо взаємодіяти з постачальником на рівні 3 і мати маршрут постачальника між мережами VPC, як показано на рисунку 8. Це забезпечує доступ до внутрішніх IP-адрес у локальній мережі та мережах Google Cloud VPC без пристрою NAT або тунелю VPN[11].

Оскільки багато корпоративних пристроїв безпеки можна використовувати в Google Cloud за допомогою віртуальних машин із кількома мережевими інтерфейсами, використовувати Cloud Interconnect для маршрутизації трафіку між мережами VPC немає необхідності, за винятком дуже небагатьох випадків, коли весь трафік має протікати через локальний пристрій через корпоративні політики.

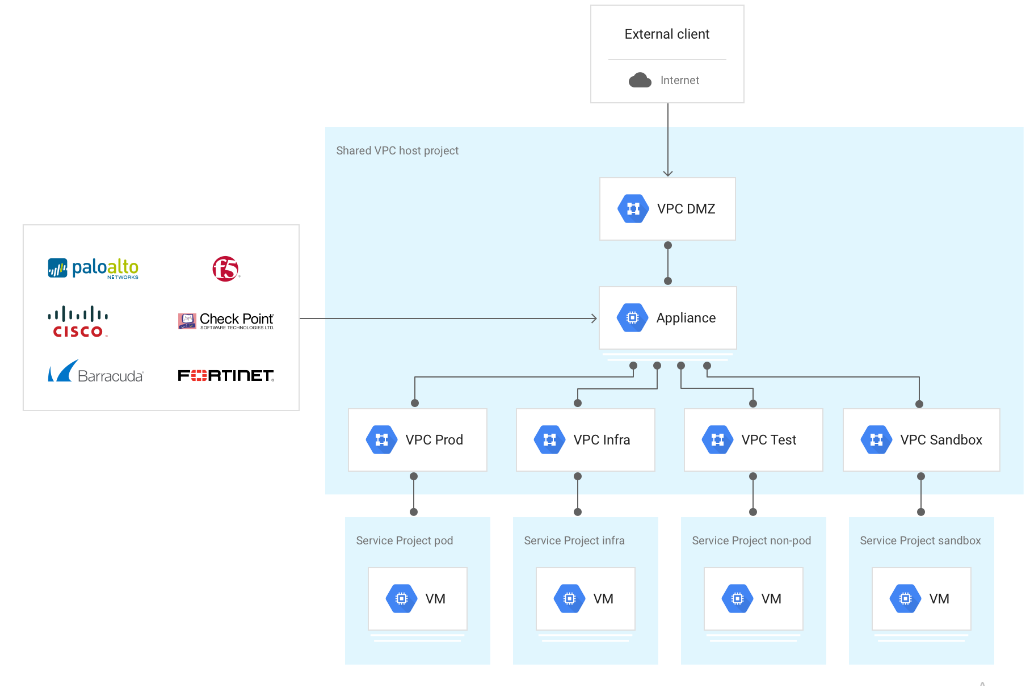
****

Рисунок 8 – Використання сторонніх постачальників

**3.2.6 Характеристика віртуальних пристроїв для керування трафіком**

Віртуальні машини з кількома мережевими інтерфейсами є звичайним явищем для мереж VPC, які потребують додаткової безпеки або служб між собою, оскільки кілька віртуальних машин з мережевим інтерфейсом дають змогу екземплярам віртуальних машин з’єднувати зв’язок між мережами VPC.

Віртуальній машині дозволено мати лише один інтерфейс для кожної мережі VPC, до якої вона підключається. Щоб розгорнути віртуальну машину в кількох мережах VPC, потрібно мати відповідний дозвіл IAM для кожної мережі VPC, до якої підключається віртуальна машина.

Під час розгортання віртуальної машини з кількома мережевими адаптерами потрібно пам’ятати, що віртуальна машина з кількома мережевими адаптерами може мати максимум 8 інтерфейсів і діапазони підмереж цих інтерфейсів не повинні перекриватися.

**3.3 Дизайн, орієнтований на безпеку**

Цей розділ визначає принципи та стратегії дизайну, спрямовані на підвищення безпеки архітектури хмарної мережі. Акцент робиться на впровадженні надійних механізмів шифрування та аутентифікації, використанні брандмауерів та систем виявлення вторгнень для моніторингу загроз, а також встановленні принципів, пов'язаних з управлінням даних та контролем доступу.

Принципи щодо шифрування та аутентифікації включають впровадження енд-ту-енд шифрування для даних у русі та спокої, вимагання багатофакторної аутентифікації для користувацького доступу та створення надійної системи управління ключами шифрування для запобігання несанкціонованому доступу до зашифрованих даних.

Принципи брандмауерів та систем виявлення вторгнень передбачають впровадження сегментації мережі для ізоляції різних сегментів, використання систем виявлення та запобігання вторгнень для моніторингу підозрілого поведінки та проведення регулярних аудитів забезпечення безпеки для оцінки ефективності заходів забезпечення безпеки.

Принципи управління даними та контролю доступу включають впровадження контролю доступу на основі ролей (RBAC) для управління правами користувачів, класифікацію даних залежно від чутливості та використання відповідних міток, а також встановлення механізмів моніторингу та забезпечення відповідності галузевим стандартам та законодавчим нормам.

Google Cloud забезпечує надійні функції безпеки в усій своїй інфраструктурі та службах, від фізичної безпеки центрів обробки даних і спеціалізованого апаратного забезпечення безпеки до спеціалізованих груп дослідників. Однак захист ресурсів Google Cloud є спільною відповідальністю. Потрібно вжити відповідних заходів, щоб забезпечити захист ваших програм і даних.

Перш ніж оцінювати засоби безпеки, вбудовані в хмару або сумісні з хмарою, потрібно почати з набору чітких цілей безпеки, які погоджуються всі зацікавлені сторони як фундаментальну частину продукту. Ці цілі мають наголошувати на досяжності, документуванні та ітерації, щоб на них можна було посилатися та вдосконалювати їх протягом розробки.

При створенні ресурсу Google Cloud, який використовує мережу VPC, потрібно обрати мережу та підмережу, де буде розміщено ресурс. Ресурсу призначається внутрішня IP-адреса з одного з діапазонів IP-адрес, пов’язаних із підмережею. Ресурси в мережі VPC можуть спілкуватися між собою через внутрішні IP-адреси, якщо це дозволяють правила брандмауера.

Важливим кроком є обмеження доступу до Інтернету лише тим ресурсам, які цього потребують. Ресурси лише з приватною внутрішньою IP-адресою можуть отримати доступ до багатьох API і служб Google через Private Service Connect або Private Google Access. Приватний доступ дає змогу ресурсам взаємодіяти з ключовими службами Google і Google Cloud, залишаючись ізольованим від публічного Інтернету.

Крім того, потрібно використовувати організаційні політики, щоб ще більше обмежити ресурси, яким дозволено використовувати зовнішні IP-адреси.

Однак перш ніж блокувати доступ до Інтернету, потрібно подуматипро вплив на екземпляри віртуальної машини. Блокування доступу до Інтернету може зменшити ризик викрадення даних, але також може заблокувати законний трафік, зокрема важливий трафік для оновлень програмного забезпечення та сторонніх API і служб. Без доступу до Інтернету можна отримати доступ до своїх екземплярів віртуальної машини лише через локальну мережу, підключену через хмарний VPN-тунель, з’єднання Cloud Interconnect або проксі-сервер із підтримкою ідентифікації. Використовуючи Cloud NAT, віртуальні машини можуть ініціювати вихідні з’єднання з Інтернетом для певного важливого трафіку, не відкриваючи публічні вхідні з’єднання[16].

Google Cloud VPC містить брандмауер рівня L3/L4, який масштабується горизонтально та застосовується до кожної віртуальної машини розподіленим способом. Цей брандмауер налаштовано за допомогою ієрархічних політик брандмауера, глобальних і регіональних мережевих політик брандмауера та правил брандмауера VPC. Докладніше див. в огляді Cloud Firewall.

Зазвичай трафік спрямовується до цих віртуальних машин, вказуючи маршрути з однаковим пріоритетом (для розподілу трафіку за допомогою хешу з 5 кортежів) або з різними пріоритетами (для створення надлишкового шляху), як показано на кількох шляхах до Dev- підмережу на рисунку 9.

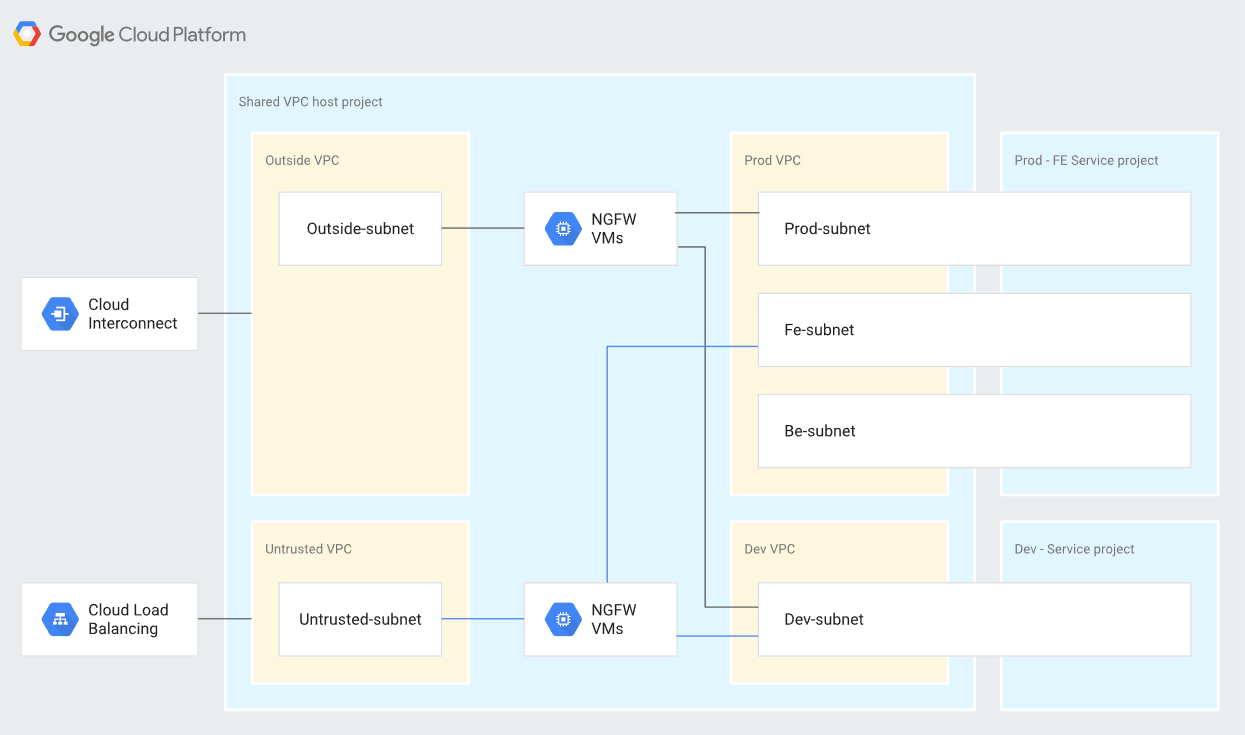
****

Рисунок 9 – Схема розподілу трафіку

Більшість рішень вимагають кількох віртуальних машин мережевого інтерфейсу. Оскільки віртуальна машина не може мати більше одного інтерфейсу на мережу VPC, при створенні віртуальної машини з декількома мережевими інтерфейсами, кожен інтерфейс потрібно підключити до окремої мережі VPC.

Масштаб також є важливим фактором під час розгортання сторонніх рішень у мережі VPC. Більшість пристроїв на основі віртуальних машин необхідно вставляти в шлях даних. Для цього потрібна віртуальна машина з декількома мережевими інтерфейсами, яка з’єднує кілька мереж VPC, які знаходяться в одному проекті. Оскільки квоти ресурсів VPC встановлюються на рівні проекту, сукупні потреби в ресурсах у всіх мережах VPC можуть стати обмеженими. введення єдиної точки дроту на основі віртуальної машини в атрибути повністю горизонтальної масштабованості мережі VPC суперечить методології хмарного проектування. Щоб пом’якшити це, можна розмістити кілька мережевих віртуальних пристроїв (NVA) у керованій групі екземплярів за внутрішнім наскрізним балансувальником мережевого навантаження.

### Щоб врахувати ці фактори у високомасштабних архітектурах вимог, слід надіслати елементи керування безпекою на кінцеві точки. Можна почати з посилення віртуальних машин і використання правил брандмауера Google Cloud. Ця стратегія також може передбачати впровадження агентів перевірки кінцевих точок на основі хоста, які не змінюють архітектуру пересилання вашої мережі VPC через кілька віртуальних машин мережевого інтерфейсу.

### На будь-якій віртуальній машині можна запрограмувати лише певну кількість правил. Однак є можливість об’єднати багато правил в одне складне визначення правила. Наприклад, якщо для всіх віртуальних машин у мережі VPC потрібно явно дозволити 10 вхідних портів TCP, інсує два варіанти: написати 10 окремих правил, кожне з яких визначає окремий порт, або визначити єдине правило, яке включає всі 10 портів. Визначення єдиного правила, яке включає всі 10 портів, є більш ефективним варіантом.

### Слід створити загальний набір правил, який застосовуватиметься до всієї мережі VPC, а потім використовувати більш конкретні набори правил для менших груп віртуальних машин, які використовують цілі. Іншими словами, можна почати з визначення широких правил і поступово визначати правила більш вузькими за потреби:

### Рекомендується застосовувати правила брандмауера, загальні для всіх віртуальних машин у мережі VPC.

### Рекомендується застосовувати правила брандмауера, які можна згрупувати в кількох віртуальних машинах, як-от групу екземплярів служби або підмережу.

### Рекомендується застосовувати правила брандмауера до окремих віртуальних машин, таких як шлюз NAT або бастіонний хост.

Багато організацій мають середовища, які вимагають певних правил для підмножини віртуальних машин у мережі VPC. У цих випадках можна застосувати два поширені підходи: ізоляція підмережі та цільова фільтрація.

Завдяки ізоляції підмережі підмережа формує межу безпеки, до якої застосовуються правила брандмауера Google Cloud. Цей підхід поширений у локальних мережевих конструкціях і у випадках, коли IP-адреси та мережеве розміщення є частиною посвідчення віртуальної машини[21].

Можна ідентифікувати віртуальні машини в певній підмережі, застосувавши до цих екземплярів унікальний тег, тег мережі або обліковий запис служби. Це дає змогу створювати правила брандмауера, які застосовуються лише до віртуальних машин у підмережі — тих, що мають пов’язаний тег, тег мережі, обліковий запис служби. Наприклад, щоб створити правило брандмауера, яке дозволяє будь-який зв’язок між віртуальними машинами в одній підмережі, слід використати таку конфігурацію правила на сторінці правил брандмауера:

* **ціль: зазначені цільові теги;**
* **цільові теги: subnet-1;**
* **фільтр джерела: підмережа;**
* **підмережа:** **subnet-1**.

За допомогою цільової фільтрації всі віртуальні машини або знаходяться в одній підмережі, або є частиною довільного набору підмереж. При такому підході членство в підмережі не вважається частиною ідентичності екземпляра для правил брандмауера. Натомість можна використовувати теги, мережеві теги або облікові записи служб, щоб обмежити доступ між віртуальними машинами в одній підмережі. Кожна група віртуальних машин, які використовують однакові правила брандмауера, має однаковий мережевий тег.

Щоб проілюструвати це, розглянемо трирівневу (веб-програма, програма, база даних) програму, для якої всі екземпляри розгорнуті в одній підмережі. Веб-рівень може спілкуватися з кінцевими користувачами та рівнем програми, а рівень програми може спілкуватися з рівнем бази даних, але ніякий інший зв’язок між рівнями не дозволяється. Екземпляри, на яких запущено веб-рівень, мають мережевий тег web, екземпляри, на яких запущено рівень програми, мають мережевий тег app, а екземпляри, на яких запущено рівень бази даних, мають мережевий тег db, як зазначено у таблиці 7.

Таблиця 7 – Реалізація правил брандмауера

|  |  |
| --- | --- |
| Правило | Пояснення |
| Дозволити кінцевим користувачам → web ярус | **Ціль: Зазначені цільові теги Цільовий тег: web Фільтр діапалоза:**Діапазони **IP  Діапазони IP: 0.0.0.0/0** |
| Дозволити web ярусу → app ярус | **Ціль: Зазначені цільові теги Цільовий гет: app Фільтр діапазона: Source tags Тег джерела: web** |
| Дозволити app ярус→ database ярус | **Ціль: Зазначені цільові теги Цільовий тег: db Фільр діапазона: цільовий тег Тег джерела: app** |

Однак, незважаючи на те, що можна використовувати мережеві теги для цільової фільтрації таким чином, рекомендовано використовувати теги або облікові записи служби, де це можливо. Цільові теги не контролюються доступом і можуть бути змінені особою з роллю instanceAdmin, поки віртуальні машини працюють. Теги та облікові записи служб контролюються доступом, тобто певний користувач має бути явно авторизований для використання облікового запису служби. Для кожного екземпляра може бути лише один обліковий запис служби, тоді як тегів може бути кілька. Крім того, службові облікові записи, призначені віртуальній машині, можна змінити, лише коли віртуальну машину зупинено.

Якщо використовуються мережеві теги, слід пам’ятати, що адміністратор екземпляра може змінити ці теги. Це може обійти політику безпеки. Тому, якщо використовуються мережеві теги у виробничому середовищі, слід використовувати інфраструктуру автоматизації, щоб допомогти подолати відсутність керування мережевими тегами IAM. Наприклад, можна використовуйте Terraform, який дає уявлення про зміни у конфігурації та може допомогти усунути проблеми.

Окрім правил брандмауера, можна використовувати ці додаткові інструменти, щоб захистити свої програми:

* глобальний балансир навантаження http(s) google cloud для підтримки високої доступності та нормалізації протоколу.
* google cloud armor із балансувальником навантаження http(s), щоб забезпечити захист від ddos і можливість блокувати або дозволяти ip-адреси на межі мережі.
* IAP для перевірки особи користувача та контексту запиту, щоб визначити, чи потрібно надавати користувачеві доступ.

**3.4 Інтеграція принципів проектування**

У першу чергу при створенні мережі VPC слід визначити ключові аспекти, такі як основні вирішуючі особи, часові обмеження та необхідну попередню роботу, щоб забезпечити відповідність вимогам зацікавлених сторін. Зацікавлені сторони можуть включати власників додатків, архітекторів безпеки, архітекторів рішень і менеджерів операцій, і ці особи можуть змінюватися в залежності від мети мережі VPC, будь то для конкретного проекту, напрямку діяльності або для всієї організації.

Серед завдань попередньої роботи також важливо включити ознайомлення команди з концепціями та термінологією, пов’язаною з проектуванням мережі VPC[14].

Потрібно розглядати проектування мережі VPC як першу та надзвичайно важливу складову проекту в хмарі. Недбале планування на цьому етапі може призвести до потенційних проблем для організації у майбутньому, особливо при внесенні значущії змін, такі як перегляд сегментації мережі або переміщення робочих завдань.

Різні конфігурації мережі VPC можуть значно впливати на маршрутизацію, масштабування та безпеку. Тому важливо ретельно розробити план та мати глибоке розуміння вашого конкретного контексту, щоб створити надійну архітектурну основу для майбутніх робочих завдань.

Збереження простої топології вашої мережі VPC — це найкращий спосіб забезпечити керовану, надійну та добре зрозумілу архітектуру.

Розробка правил найменування ресурсів та компонентів мережі є важливим етапом. Слід робити їх простими, інтуїтивно зрозумілими та узгодженими. Це гарантує чітке розуміння призначення кожного ресурсу, його розташування, залежність та відмінність від інших ресурсів.

Загальноприйняті скорочення довгих слів допомагають скоротити назву та зберегти призначення ресурсу. Використання знайомої термінології, де це можливо, сприяє читабельності.

Нижче наведено приклад правил найменувань ресурсів:

* назва компанії: «global solutions»: «gs »;
* департмент: business analysis: bi;
* код регіону: united states central 1 : us-c1, europe-west1: eu-w1;
* код оточення: dev, test, uat, stage, prod.

У цьому прикладі середовище розробки для бізнес аналітиків називається gs-bi-eu-w1-dev.

Для інших поширених мережевих ресурсів розглянемо шаблони, наведені у таблиці 8.

Таблиця 8 – Правила найменування мережевих ресурсів

|  |  |  |
| --- | --- | --- |
| Ресурс | Синтакс | Приклад |
| VPC мережа | {назва компанії}-{опис(App or BU)- мітка }-{середовище- мітка }-{ послідовність #} | acmeco-hr-dev-vpc-1 |
| Підмережа | {назва компанії}-{опис(App or BU)-мітка}-{регіон/зона- мітка } | acmeco-hr-na-ne1-dev-subnet |
| Правило брандмауера | {назва компанії}-{опис(App or BU)-мітка}{ джерело-мітка }-{ мітка призначення }-{протокол}-{порт}-{дія} | acmeco-hr-internet-internal-tcp-80-allow-rule |
| IP шлях | {приоритет}-{VPC-мітка}-{тег}-{ наступний стрибок } | 1000-acmeco-hr-dev-vpc-1-int-gw |

При створенні нового проекту, використовується мережу за замовчуванням, яка є мережею VPC автоматичного режиму, і має назву "default". Мережі в автоматичному режимі автоматично створюють підмережі та відповідні маршрути для цих підмережей. Основний діапазон IP-адрес для кожного регіону Google Cloud становить /20 CIDR і використовує передбачуваний набір діапазонів адрес RFC 1918. Мережа за замовчуванням також містить попередньо встановлені правила брандмауера.

Хоча мережі в автоматичному режимі можуть бути корисними для початкового дослідження, мережі VPC у спеціальному режимі краще підходять для більшості виробничих середовищ. Рекомендовано підприємствам із самого початку використовувати мережі VPC у спеціальному режимі з таких причин:

* не можна з’єднати дві мережі vpc в автоматичному режимі за допомогою однорангового зв’язку мережі vpc, оскільки їхні підмережі використовують однакові основні діапазони ip-адрес;
* усі підмережі автоматичного режиму мають однакову назву з мережею. можна вибрати унікальні описові назви для підмереж спеціального режиму, що зробить мережі vpc більш зрозумілими та придатними для обслуговування;
* коли з’являється новий регіон google cloud, мережі vpc в автоматичному режимі автоматично отримують нову підмережу в цьому регіоні. мережі vpc у спеціальному режимі отримують нові підмережі, лише якщо вказати їх. це може бути важливо як для суверенітету, так і для керування ip-адресою.

Деякі великі корпоративні розгортання включають автономні групи, кожній з яких потрібен повний контроль над відповідними мережами VPC. Можна задовольнити цю вимогу, створивши мережу VPC для кожного бізнес-підрозділу зі спільними службами в спільній мережі VPC (наприклад, аналітичні інструменти, конвеєр CI/CD і комп’ютери для збирання, служби DNS/каталогів).

Для компаній, які мають справу з ініціативами відповідності, конфіденційними даними або строго регламентованими даними, пов’язаними стандартами відповідності, такими як HIPAA або PCI-DSS, додаткові заходи безпеки часто мають сенс. Один із методів, який може покращити безпеку та спростити підтвердження відповідності, — це ізолювати кожне з цих середовищ у власну мережу VPC.

Мережа VPC забезпечує повну сітку глобальної доступності. З цієї причини спільні служби та конвеєри безперервної інтеграції, що знаходяться в одній мережі VPC, не вимагають особливої уваги, коли мова йде про підключення — вони за своєю суттю доступні. Спільний VPC розширює цю концепцію, дозволяючи загальним службам перебувати в ізольованому проекті, забезпечуючи підключення до інших служб або споживачів.

Споживачі можуть використовувати власні внутрішні IP-адреси для доступу до послуг, не виходячи зі своїх мереж VPC або використовуючи зовнішні IP-адреси. Трафік залишається повністю в Google Cloud. Private Service Connect забезпечує сервіс-орієнтований доступ між споживачами та виробниками з детальним контролем доступу до послуг.

У моделі однорангового зв’язку мережі VPC кожна мережа VPC створює одноранговий зв’язок із спільною мережею VPC зі спільними послугами, щоб забезпечити доступність. Мережевий піринг VPC вводить міркування щодо масштабування, оскільки обмеження масштабування застосовуються до сукупного використання ресурсів усіма одноранговими вузлами, як наведено на рисунку 10.

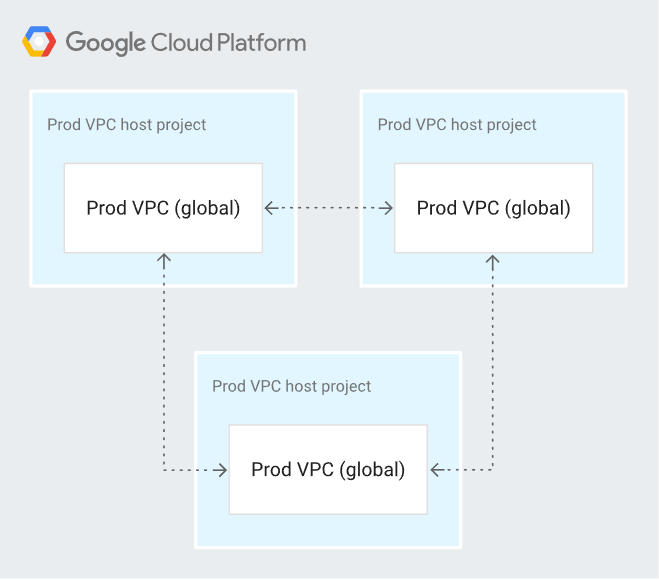
****

Рисунок 10 - Однорангова мережа

Мережевий піринг VPC також можна використовувати в поєднанні з приватним доступом до служби та API служби мереж. За допомогою Service Networking API модна дозволити клієнтам у тій самій організації чи іншій організації користуватися послугами, якінадаються, але дозволити їм вибрати діапазон IP-адрес, які підключаються за допомогою спільного доступу до мережі VPC[15].

Хмарний VPN – ще одна альтернатива. Оскільки Cloud VPN встановлює доступність через керовані тунелі IPsec, він не має сукупних обмежень VPC Network Peering. Cloud VPN використовує шлюз VPN для підключення та не враховує сукупне використання ресурсів однорангового IPsec. До недоліків Cloud VPN належать підвищені витрати (тунелі VPN і вихід трафіку), накладні витрати на керування, необхідні для підтримки тунелів, і накладні витрати на продуктивність IPsec. Схему взаємодії через VPN наведено на рисунку 11.

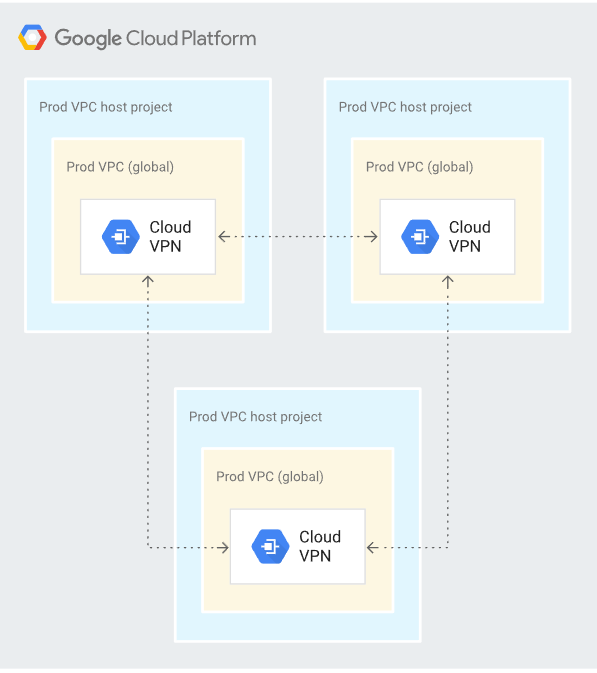
****

Рисунок 11 - Інтеграція ВПН

**4 АНАЛІЗ РЕЗУЛЬТАТІВ ПРОЕКТУВАННЯ ХМАРНОЇ МЕРЕЖІ**

**4.1 Оцінка продуктивності**

Збільшення продуктивності пропускної здатності мережі можливо шляхом зміни максимальної одиниці передачі (MTU) до 8896 байт, що дозволяє передавати більше даних за меншу кількість пакетів і, отже, зменшує використання пропускної здатності заголовками TCP/IP. Це призводить до вищої ефективної пропускної здатності мережі.

Вихідна пропускна здатність віртуальних машин Compute Engine залежить від їх типу. Враховуючи це при розробці архітектури, можна вибрати тип машини, який відповідає очікуваному трафіку. Наприклад, віртуальні машини серій N2, N2D, C2 і C2D, які використовують мережу Tier\_1, підтримують вихідну пропускну здатність до 100 Гбіт/с. Віртуальні машини H3 здатні до 200 Гбіт/с, а C3 і C3D мають обмеження вихідної смуги пропускання до 200 Гбіт/с з мережею Tier\_1. Таким чином, використовуючи машини серії C3 або H3, можна подвоїти пропускну здатність мережі

Розташування серверної частини додатку ближче до користувачів зменшує затримку мережі. Якщо користувачі розташовані в різних регіонах, можна розгорнути обчислювальні ресурси в цих регіонах і використовувати сервіс "Load Balancing" для маршрутизації трафіку залежно від локації. Вибір типу балансувальника навантаження може впливати на затримку, яку відчувають користувачі. Дослідження показують, що мережевий балансувальник навантаження на рівні 7 моделі OSI може бути більш продуктивним, ніж на рівні 4 моделі OSI.

Google автоматично шифрує трафік між балансувальниками навантаження та серверними частинами на рівні пакетів. Це дозволяє уникнути зайвого шифрування на рівні 7-го рівня HTTPS для більшості випадків. Рекомендується використовувати HTTP замість HTTPS або HTTP/2 для трафіку між балансувальником навантаження та серверними частинами, щоб зменшити використання ЦП серверних віртуальних машин. Однак для безпеки трафіку в загальнодоступному Інтернеті слід використовувати HTTPS або HTTP/2, коли серверною частиною є група кінцевих точок мережі Інтернет (NEG). Рекомендується також порівняти шаблони трафіку вашої програми для досягнення оптимальної продуктивності.

**4.2 Оцінка масштабованості**

Масштабованість у хмарній мережі є однією з ключових переваг, яка дозволяє організаціям гнучко адаптуватися до змінних вимог та обсягів даних. Хмарні мережі забезпечують високу ступінь еластичності, дозволяючи легко масштабувати ресурси вгору або вниз відповідно до потреб користувачів. Це означає, що підприємства можуть швидко розширювати мережеву пропускну здатність або обчислювальні ресурси під час сплесків попиту, а також зменшувати їх, коли потреба в ресурсах знижується. Така гнучкість не тільки оптимізує витрати, але й забезпечує високу доступність та продуктивність, адже системи можуть швидко адаптуватися до будь-яких змін у робочому навантаженні. Хмарні мережі також спрощують розгортання нових додатків та служб, пропонуючи широкий спектр автоматизованих інструментів для управління масштабованістю, що дозволяє підприємствам швидко реагувати на ринкові зміни та інноваційні виклики.

Оцінка масштабованості висвітлює успіх архітектури у ефективній адаптації до динамічних робочих навантажень та ефективному управлінні сценаріями високого стресу. Ці позитивні результати становлять основу для майбутніх покращень та оптимізацій масштабованості в мережевій архітектурі[12].

Проектувати мережу в хмарному провайдері потрібно саме так, щоб створені ресурси могли без перешкод та великих затрат комунікувати між собою. Цього можна досягти за допомогою використання Shared VPC у поєднанні з іншими мережевими сервісами за необхідністю. Ця VPC дозволяє керувати усіма компонентами мережі централізовано з одного хост-проекта та розповсюджувати для проектів-користувачів. Причому, ліміти кількості логічних груп визначаються тільки кількості підмереж.

Для досягнення регіонального масштабування по всьому світі слід створювати підмережі для цільових регіонів, щоб зменшити затримку та збільшити продуктивність мережі. Групування мережі в меншу кількість підмереж із більшими діапазонами адрес дозволяє оптимізувати управління адресами та забезпечує більшу гнучкість у виборі діапазонів для програм. Додаткові мережі VPC можуть бути створені в залежності від таких факторів, як масштаб, безпека, фінансові міркування, операційні вимоги та керування ідентифікацією та доступом. Використання спільного VPC для адміністрування кількох робочих груп надає зручний інструмент для розширення простоти архітектури на кілька робочих груп.

Створивши декілька Shared VPC для різних середовищ (Середовище розробки, середовище тестування, середовище продакшн) можна створити зв’язок між ними за допомогою мережевих шлюзів, таких як Interconnect. При масштабуванні мережі або її об’єднанні з мережою другої організації або компанії слід використовувати VPN. Цей тип з’єднання дозволяє передавати данні безпечно, без впливу на продуктивність мережі. Така система дозволяє масштабувати мережу в організаційному плані, роблячи керування простим, а роботу передачі даних надійним та безперебійним.

Одним з ключовим аспектом при масштабуванні ресурсів є балансувальник навантаження. Даний сервіс дозволяє перенаправляти трафік на обчислювальні ресурси в залежності від навантаження, регіону, певного правила або доступності ресурсу. Результати показали, що різниця в затримці при використанні балансувальника навантаження та прямого з’єднання до віртуальної машини є мінімальні і становлять близько 1-2 %, що є статистичною похибкою. В іншому сценарії при використанні балансувальника по протоколу HTTP отримали затримку в 100 раз меншу, ніж при прямому з'єднанні.

**4.3 Оцінка безпеки**

Урахування безпеки є важливою частиною при проектуванні мережі. Для захисту мережевої архітектури, додатків та даних потрібно чітко слідувати вимогам до безпеки, законів країн, та політики організації.

Одним із головних інструментом для забезпечення безпеки в мережі є правило брандмауера. Цей засіб блокує та дозволяє вхідний або вихідний трафік. Практика показує, що слід створювати цільове правило брандмауера для конкретного ресурсу або системи та включати в нього усі правила, які дозволяють або забороняють з’єднання за усіма протоколами чи портами, які вимагає реалізація.

Щоб запобігти несанкціонованому втручанню у систему потрібно слідувати принципу мінімальних привілей. Це означає, що потрібно блокувати усе потенційне з’єднання окрім тих, які вимагає система.

Такий самий принцип потрібно використовувати в подібному до брандмауеру сервісі – менеджер ідентифікації та доступу (IAM). Даний сервіс дозволяє комунікувати між сервісами за допомогою пари публічного та приватного ключа. Ця пара закріплена за обліковим записом та може бути закріплення за декількома ресурсами. Запис має список ресурсів та дій, які він може виконувати. Таким чином можна побудувати систему, яка взаємодіє між собою за чіткими правилами та забезпечує необхідний рівень безпеки.

Для забезпечення захисту системи важливим фактором є використання захищених протоколів передачі даних. Якщо система є з виходом до інтернету, як наприклад веб-сайт, то таке з’єднання слід захистити протоколом HTTPS. Захищені протоколи шифрують дані при передачі трафіку, тому потенційний зловмисник не зможе перехопити персональні дані. В деяких випадках допускається використання незахищених протоколів, якщо трафік передається всередині системи та ізольовані від зовнішнього втручання. Дослід показав, що при використанні стандартних протоколів замість захищених в ізольованій системи дає ріст продуктивності в 5%, в цей же час не впливає на загальну захищеність системи[17].

Ще одним аспектом безпеки є використання глобального сервісу захисту від DDoS атак. Інтеграція Google Cloud Armor разом з балансувальником навантаження HTTP(S) блокує вхідний трафік ззовні та зменшує навантаження на внутрішні системи мережі.

**4.4 Результати дослідження**

Ітеративний підхід до дизайну та впровадження виявився вирішальним чинником, що дозволяє здійснювати постійне вдосконалення на основі відгуків користувачів, результатів тестування та змінюючихся операційних потреб. Цей ітеративний процес відіграв значущу роль у загальному успіху архітектури.

Комплексні заходи безпеки, включаючи міцні протоколи шифрування, механізми аутентифікації та дотримання вимог до відповідності, були інструментальними факторами, що забезпечили високий рівень захисту для чутливих даних. Впровадження комплексного підходу до безпеки виявилося ключовим фактором успіху.

Виявлені ці фактори відігравали важливу роль у успіху реалізованої архітектури. У наступних розділах буде розглянуто наслідки цих факторів та запропоновано рекомендації для подальших удосконалень, забезпечуючи постійний фокус на ітеративне вдосконалення та оптимізацію.

Виявлення можливостей для подальшого удосконалення вказало на кілька напрямків, не зважаючи на позитивні результати.

Постійне Вдосконалення Моніторингу було визначено як область для поліпшення, незважаючи на позитивні результати. Є потенціал для протидії масштабованості та викликів безпеки, впроваджуючи більш продуктивні інструменти та стратегії моніторингу. Такі удосконалення можуть подальше підвищити адаптивність та реагування мережі[4].

Вдосконалення Плану Комунікацій визнається як область із потенціалом для удосконалення. Уточнення плану комунікацій сприятиме ще більш чіткій та ефективній комунікації з зацікавленими сторонами, сприяючи кращому розумінню та співпраці між всіма зацікавленими сторонами.

Стратегії Адаптивного Дизайну, хоча і проявляють адаптивність, вимагають подальшого вдосконалення для врахування змінюючихся операційних робочих процесів. Майбутні впровадження повинні надавати пріоритет ще більш гнучким принципам дизайну, щоб забезпечити неперервну адаптованість до змінних вимог.

Покращені Програми Навчання Користувачів, хоча успішні, представляють область для можливого удосконалення. Майбутні впровадження повинні досліджувати можливості розширення та адаптації програм навчання для різних груп користувачів, забезпечуючи більш загальне розуміння системи та оптимізацію користувацької кваліфікації.

Визначення цих областей для удосконалення створює передумови для подальших вдосконалень та оптимізацій у наступних ітераціях архітектури мережі. Наступні розділи розглянуть наслідки цих областей та запропонують рекомендації для подальшого уточнення та постійного вдосконалення.

Успіх архітектури сильно пов'язаний із її гнучкістю та адаптивністю. Майбутні впровадження повинні надавати пріоритет таким елементам дизайну, які безперешкодно дозволяють адаптуватися до нових технологій і змінюваних бізнес-вимог, забезпечуючи тривалу актуальність та ефективність.

Встановлення культури постійного вдосконалення виявилося ключовим висновком. Створення культури, де отримані уроки з кожної фази інформують подальші ітерації, є суттєвим. Цей культурний аспект внесе значний вклад у постійне вдосконалення та оптимізацію архітектури мережі, сприяючи адаптивності та стійкості.

Досягнення балансу між продуктивністю, масштабованістю та безпекою ставить перед собою постійний виклик. Майбутні впровадження повинні уважно розглядати витончену взаємодію цих елементів, визнаючи можливі уступки та ідентифікуючи синергії для забезпечення гармонізованого та оптимізованого мережевого середовища[5].

**4.5 Рекомендації**

Рекомендації для майбутніх дизайнів включають у себе прийняття ітеративного підходу до дизайну та впровадження. Це передбачає постійне збирання відгуків користувачів, проведення тестувань та удосконалення архітектури на основі реального використання. Ітеративний процес, як вже доведено в поточній реалізації, відіграє ключову роль у досягненні успіху та забезпеченні того, щоб архітектура залишалася відзивчивою до змінюючихся потреб.

Забезпечення того, щоб архітектура відповідала потребам та уподобанням користувачів, вносить значний внесок у загальне задоволення користувачів та підвищення продуктивності. Підхід, орієнтований на користувача, повинен бути невід'ємною частиною розгляду дизайну, що дозволяє архітектурі ефективно відповідати очікуванням та вимогам користувачів.

Впровадження комплексних заходів безпеки є критичним, і рекомендація підкреслює надійні протоколи шифрування, механізми аутентифікації та планування реагування на інциденти. Мультишаровий підхід до безпеки є ключовим для захисту конфіденційної інформації та забезпечення безпечного середовища мережі. Підкреслюючи заходи безпеки, майбутні дизайни можуть зменшити потенційні ризики та підвищити загальну стійкість мережі.

Рекомендацією також є прийняття стратегій адаптивного дизайну, які дозволяють гнучко враховувати змінюючіся операційні робочі процеси. Здатність адаптуватися до змінних бізнес-вимог та технологічного середовища є ключовою для довгострокового успіху архітектур мереж хмар. Ця рекомендація сприяє перегляду у майбутнє, забезпечуючи, що дизайни залишаються готовими до впровадження нових тенденцій та викликів.

Для майбутніх дизайнів слід включати підвищену увагу до розширення системи постійного моніторингу для проактивного вирішення можливих проблем масштабованості та безпеки. Це передбачає впровадження високорівневих засобів та стратегій моніторингу, які відповідають динаміці операцій у хмарних середовищах. Приділяючи пріоритет системі постійного моніторингу, майбутні дизайни можуть забезпечити більш прогнозований та оперативний підхід до потенційних викликів.

Також рекомендується покращення та розширення програм навчання користувачів. Це включає в себе адаптацію ініціатив навчання для різних груп користувачів, врахування конкретних потреб та стимулювання ефективного використання хмарної мережі. Загальне розуміння системи серед користувачів є ключовим для максимізації переваг архітектури, а індивідуалізоване навчання може значно сприяти вивченню користувачами системи.

Особлива увага повинна бути приділена компромісам між масштабованістю та ефективністю з точки зору витрат. Майбутні дизайни повинні прагнути досягти балансу, що забезпечить оптимальну масштабованість при обережному врахуванні пов'язаних витрат та відповідності бюджетним обмеженням організації. Ця рекомендація підкреслює важливість врахування витрат при прийнятті рішень щодо масштабування.

Слід пам’ятати про вивчення можливостей інтеграції новітніх технологій, таких як штучний інтелект та обчислення на межі, в хмарні архітектури. Слідкування за технологічними новаціями та співпраця з відповідними спільнотами може допомогти використовувати передові рішення для покращення можливостей хмарних мереж. Цей погляд у майбутнє дозволяє дизайнам залишатися на передовому рівні технологічних інновацій.

Розвивати культуру постійного удосконалення в організації може покращити майбутні дизайни архітектури. Це включає стимулювання зворотного зв'язку, вивчення з досвіду та адаптацію принципів дизайну на основі змінюючихся стандартів галузі та вимог користувачів. Культура постійного удосконалення сприяє довгостроковому успіху та актуальності архітектур хмар.

Рекомендації для майбутніх впроваджень включають у себе прийняття факторів успіху, визначених у дослідженні, таких як ітеративний дизайн, принципи, орієнтовані на користувача, і комплексні заходи безпеки, як напрямних принципів. Ці фактори виявилися ключовими для досягнення успіху і повинні бути вбудовані в основу майбутніх дизайнів.

Надавати пріоритет гнучкості в дизайні важливо для адаптації до змін потреб користувачів, операційних потоків та нових технологій. Акцент на гнучкості забезпечує, що архітектура залишається адаптивною до змінних вимог і може легко інтегрувати нові технології по мірі їх з'явлення.

Ще однією ключовою рекомендацією є активне управління масштабованістю. Це включає врахування витрат, впровадження стратегій динамічного виділення ресурсів та впровадження ефективних механізмів балансування навантаження. Проактивне управління масштабованістю гарантує оптимальну продуктивність під час змінних робочих навантажень, враховуючи пов'язані витрати.

Надання користувачам можливостей через комплексні програми навчання, адаптовані до їхніх конкретних ролей і обов'язків у хмарній мережі, є важливим. Навчання користувачів підвищує їхню кваліфікацію та гарантує, що вони ефективно використовують хмарну мережу, що сприяє загальній продуктивності та задоволенню.

**ВИСНОВОК**

При проектуванні мережі в хмарному постачальнику обчислювальних потужностей слід чітко дотримуватись вимог продуктивності, масштабованості та безпеки. Результати показали, що всі ці три фактори тісно пов’язані між собою та впливають один на одного. При розробці мережі слід слідувати відношенню: 33.3% виділити на масштабування системи, 33.3% виділити на продуктивність мережі та 33.3% виділити на безпеку.

По-перше, продуктивність мережі залежить від кількості виділених обчислювальних ресурсів, таких як CPU і RAM, тому що збільшення потужності апаратних засобів покращує пропускну здатність в 3-4 рази.

По-друге, зміна максимальної одиниці передачі (MTU) до 8896 байт збільшує продуктивності пропускної здатності мережі, що дозволяє передавати більше даних за меншу кількість пакетів і, отже, зменшує використання пропускної здатності заголовками TCP/IP.

По-трете, розташування ресурсів в географічний локації цільових користувачів зменшує затримку, оскільки фізичні можливості передачі даних обмежені і відстань може пропорціонально збільшити затримку.

По-четверте, аналіз безпеки мережі підтверджує важливість інтегрованих систем захисту, включаючи шифрування даних, аутентифікацію користувачів та моніторинг мережевого трафіку. Ефективні механізми безпеки не тільки захищають важливі дані, але й підвищують загальну надійність хмарної мережі.

По-п'яте, розглядаючи масштабованість, встановлено, що автоматизовані рішення для розподілу навантаження та еластичного управління ресурсами дозволяють оптимально використовувати обчислювальні потужності залежно від змінних потреб користувачів.

По-шосте, дослідження показало, що використання новітніх технологій віртуалізації сприяє підвищенню продуктивності мережі, забезпечуючи ефективніше використання апаратних ресурсів та знижуючи витрати на обслуговування.

По-сьоме, результати дослідження підкреслюють важливість використання захищених протоколів, таких як TLS/SSL, HTTPS для шифрування даних. Це забезпечує конфіденційність та цілісність переданих даних, запобігаючи втручанням та перехопленню інформації.

По-восьме, дослідження показує, що розподіл мережевих ресурсів за допомогою підмереж дозволяє підвищити безпеку та ефективність управління мережею. Підмережі дозволяють ізолювати групи ресурсів, знижуючи ризики безпеки та полегшуючи адміністрування.

По-дев’яте, належні заходи захисту від DDoS-атак є критично важливими для забезпечення стабільності та доступності хмарних сервісів. Імплементація систем виявлення та нейтралізації DDoS-атак є необхідною для запобігання перевантажень мережі та забезпечення неперервності сервісу.

По-десяте, ефективне налаштування правил брандмауера є ключовим для забезпечення безпеки мережевих систем. Правильно сконфігуровані брандмауери дозволяють контролювати вхідний та вихідний мережевий трафік, захищаючи систему від несанкціонованого доступу.

Останнє, Використання Shared VPC в архітектурі хмарної мережі дозволяє централізовано керувати мережевими ресурсами, забезпечуючи легку масштабованість та гнучкість управління. Цей підхід поліпшує використання мережевих ресурсів та спрощує управління безпекою на масштабі всієї організації.

**ПЕРЕЛІК ПОСИЛАННЬ**

1. Journal of Cloud Computing. 2022. "Title of Article." Vol.3, no. 613. P.54. URL:https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00362-x (date of access: 21.10.2023).
2. Erl, Thomas, Ricardo Puttini, and Zaigham Mahmood. "Cloud Computing: Concepts, Technology & Architecture." P.177.
3. Kavis, Michael J. "Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)." P. 29.
4. Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. "Cloud Computing: Principles and Paradigms." P.34.
5. Rafaels, Ray J. "Cloud Computing: From Beginning to End.". P.102.
6. Kleppmann, Martin. "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems.". P.39.
7. Vaquero, L.M., Rodero-Merino, L., et al. "A Break in the Clouds: Towards a Cloud Definition." ACM SIGCOMM Computer Communication Review. P.243.
8. Vacca, John R., ed. "Cloud Computing Security: Foundations and Challenges." P. 19.
9. Armbrust, Michael, et al. "Above the Clouds: A Berkeley View of Cloud Computing." P.55.
10. Takabi, Hassan, James B.D. Joshi, and Gail-Joon Ahn. "Security and Privacy Challenges in Cloud Computing Environments.". P.49.
11. Li, Ming, et al. "Scalable and Secure Sharing of Personal Health Records in Cloud Computing.". P.17.
12. Google Cloud. "VPC Specifications." URL: https://cloud.google.com/vpc/docs/vpc#specifications (date of access: 21.12.2023).
13. Google Cloud. "Global Load-Balancing Architectures for DNS Routing Policies." URL: https://cloud.google.com/architecture/global-load-balancing-architectures-for-dns-routing-policies (date of access: 21.12.2023).
14. Google Cloud. "Networking in Google Cloud: A Comprehensive Guide." URL: https://cloud.google.com/architecture/framework/system-design/networking (date of access: 21.12.2023).
15. Google Cloud. "VPC Specifications." URL: https://cloud.google.com/vpc/docs/vpc#specifications (date of access: 21.12.2023).
16. Google Cloud. "Best Practices for VPC Design and Implementation." URL: https://cloud.google.com/architecture/best-practices-vpc-design#network\_security (date of access: 21.12.202)
17. Kharb L., Chahal D. Cloud Access Security Brokers: Strengthening Cloud Security. International Journal of Research Publication and Reviews. 2023. Vol. 4, no. 8. P. 642–644. URL: <https://doi.org/10.55248/gengpi.4.823.50412> (date of access: 21.12.2023).
18. Cabianca D. Managing Network Operations. Google Cloud Platform (GCP) Professional Cloud Network Engineer Certification Companion. Berkeley, CA, 2023. P. 373–419. URL: <https://doi.org/10.1007/978-1-4842-9354-6_8> (date of access: 21.12.2023).
19. Савченко О. С. Хмарні технології в бізнесі : thesis. 2013. URL: <http://essuir.sumdu.edu.ua/handle/123456789/32774> (дата звернення: 21.12.2023).
20. An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN / Z. Zhang et al. *Photonic Network Communications*. 2004. Vol. 7, no. 3. P. 213–225. URL: <https://doi.org/10.1023/b:pnet.0000026887.35638.ce> (date of access: 21.12.2023).
21. Mogul J. C. Internet subnets. RFC Editor, 1984. URL: <https://doi.org/10.17487/rfc0917> (date of access: 21.12.2023).

**ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ**

