

Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки та комп'ютерних систем
Кафедра комп'ютерної інженерії

**«Дослідження безкоштовних студентських програм постачальників
хмарних ресурсів для застосування у лабораторних роботах при
підготовці фахівців комп'ютерної інженерії спеціальності Мережевий
адміністратор»**

Кваліфікаційна робота бакалавра
Студента спеціальності
123 «Комп'ютерна інженерія»
Павла КОЧАКОВА
_____ (підпис)

Науковий керівник,
канд. техн. наук, асистент
Олександр БОРЕЦЬКИЙ
_____ (підпис)

Рецензент
к.ф.-м.н., асистент
_____ (підпис)

До захисту допускаю
Протокол засідання кафедри від
“__” _____ 2022р. № _____

Завідувач кафедри
к.ф.-м.н., доцент
Юрій БОЙКО

КИЇВ 2022

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра містить 39 сторінок, 5 рисунків, 16 використаних джерела.

Метою роботи є практичний аналіз та порівняння безкоштовних студентських програм постачальників хмарних ресурсів для застосування у лабораторних роботах при підготовці фахівців комп'ютерної інженерії спеціальності Мережевий адміністратор.

Основні постачальники хмарних ресурсів: Amazon Web Services (AWS), Microsoft Azure, Google Cloud.

Актуальність роботи полягає у дослідженні можливостей хмарних обчислень, їх можливу інтеграцію в навчальний процес.

ЗМІСТ

РЕФЕРАТ	2
ВСТУП.....	4
1. КОМП'ЮТЕРНА ІНЖЕНЕРІЯ.....	5
1.1. Системні вимоги для віртуальних машин.....	6
1.2. Бази даних для мережевої інфраструктури.....	6
1.3. Інтернет сервіси.....	8
1.4. Технологічна практика.....	9
1.5. Системне програмне забезпечення.....	9
1.6. Вимоги до хмарних рішень.....	10
2. Аналіз хмарних рішень для виконання лабораторних робіт.....	11
2.1 . Типи хмарних обчислень.....	12
2.1.1 IaaS.....	13
2.1.2 PaaS.....	16
2.1.3 SaaS.....	18
2.1.4 SaaS проти PaaS проти IaaS.....	20
2.1.5 Безсерверні обчислення.....	21
2.1.6 Публічні хмари.....	23
2.1.7 Приватні хмари.....	24
2.1.8 Гібридні хмари.....	25
2.1.9 Мульти-хмара та гібридна мульти-хмара.....	25
2.2. Amazon Web Services.....	26
2.3. Microsoft Azure.....	29
2.4. Google cloud.....	32
2.4.1. Google Cloud Platform GCP.....	33
ВИСНОВКИ.....	37
ВИКОРИСТАНІ ДЖЕРЕЛА.....	38

ВСТУП

В Україні хмарні сервіси активно почали використовувати у 2014 році. Найбільшого піку популярності очікують на основі війни, яка почалася в лютому 2022 року. Після анексії Криму та Донбасу багато компаній зі сходу переїхали до Києва або інших міст і вирішили не будувати знову свої дата-центри, так як це потребує значних фінансових вкладень, а орендувати середовище у хмарних сервісах. Іншого варіанту зберегти бізнес у них не було. За попередніми даними компанії IDC за підсумками 2018 року, основними постачальниками хмарних послуг в Україні є іноземні компанії. Згідно з даними IDC, на Amazon Web Services (AWS) припадає 29% нашого ринку хмарних ресурсів (\$4,7 млн), на Microsoft Azure – 27% (\$4,4 млн). Українські постачальники хмарних ресурсів займають значно меншу частину ринку: «Де Ново» – 11% (\$1,9 млн), GigaCloud – 3%(\$0,6млн).

У зв'язку з зазначеним вище, початком воєнних дій, збільшується кількість пропозицій на ринку і з'явилась проблема фільтрування постачальників.

Мета роботи: проаналізувати основних постачальників хмарних ресурсів, які можуть стати у нагоді при виконанні лабораторних та практичних робіт фахівців комп'ютерної інженерії спеціальності «мережевий адміністратор».

1. КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

“Комп'ютерна інженерія” - це поєднання апаратного та програмного забезпечення серед комп'ютерних систем і мереж. Спеціальність об'єднує в собі електроніку та комп'ютерні науки, від вивчення як працює обчислювальна техніка, до того як працюють сучасні процесори. При здобутті даної спеціальності, майбутні адміністратори чи інженери профілюються за двома спеціальностями, системні та мережеві адміністратори

Мережеві адміністратори – спеціалісти що підтримують функціонування мереж, контролюють функції та операції в межах мережі, та дбають про її технічний стан.

Системні адміністратори – спеціалісти що дбають про роботоздатність комп'ютерної системи, як на апаратному, так і на програмному рівні.

Навчальний план забезпечує баланс між початковими курсами з комп'ютерних систем, такі як курс “Апаратне та програмне забезпечення” і сукупністю фундаментальних дисциплін з фізико-математичної підготовки та включає низку спеціалізованих курсів.

Основні професійні дисципліни що вивчаються протягом здобуття освітнього рівня: програмування, алгоритми та методи обчислень, схемотехніка, прикладна теорія цифрових автоматів, комп'ютерні мережі, інтернет сервіси, комп'ютерні системи та бази даних для мережевої інфраструктури.

Серед переліку дисциплін, можна відокремити декілька практичних курсів, таких як: “Бази даних для мережевої інфраструктури”, “Інтернет сервіси”, “Системне програмне забезпечення” та “Технологічна практика”. Вони потребували значної обчислювальної потужності наприклад: як декілька віртуальних машин, чи пару інтегрованих середовищ розробки, тощо.

1.1. Системні вимоги для віртуальних машин

Виконання лабораторних робіт передбачає в собі використання різних віртуальних машин. В загальному використовуються віртуальні машини на базі Windows Server та Linux.

Мінімальні системні вимоги для віртуальних машин Windows Server:

- 64-розрядний процесор із тактовою частотою 1.4 ГГц
- 2 ГБ оперативної пам'яті для режиму роботи з робочим столом
- Мінімальний об'єм жорсткого диску повинен бути 30ГБ

Мінімальні системні вимоги для віртуальних машин на базі ОС Linux:

- Процесор із тактовою частотою 1ГГц
- 1ГБ оперативної пам'яті
- Мінімальний об'єм жорсткого диску 10ГБ

Дані вимоги можна вважати рекомендованими для працездатності віртуальних машин, і надалі в роботі будуть згадуватись як типова конфігурація для тієї чи іншої операційної системи.

1.2. Бази даних для мережевої інфраструктури

Дисципліна вивчається протягом року. І розпочинається вона на 4 семестрі навчання, як тільки відбудеться профілювання.

Перший семестр розглядав налаштування DNS та DHCP серверів, на операційних системах CentOS і Windows Server, а також налаштування Proху-сервера і слідом вивчали TOR мережу.

Другий семестр вивчення розглядав уже мережеві протоколи: SNMP, LDAP, Kerberos, RADIUS сервер.

Із наведених вище тем лабораторних робіт можна зробити висновок, що вивчення даного лабораторного практикума, дає фундаментальні знання того як працюють мережеві пристрої.

Хоч в першій частині опрацьовували лише частину, що відповідала в загальному за адресацію в мережі, але отримали наступні знання:

- DNS - сервер його призначення і налаштування
- DHCP - сервер як налаштовується динамічна видача ір-адреса та процес встановлення з'єднання з сервером
- Проху - дозволяє нам змінювати власний ір-адрес, і виступає проміжним вузлом між двома точками
- TOR - як спосіб приховування даних по відкритій мережі інтернет, шляхом різних шарів шифрування, які то з'являються, то зникають на кожному з вузлів маршрутизації.

В другому семестрі розглядалися такі поняття, як облік користувачів та їх авторизація і аутентифікація. Вивчалися поняття шифрування та хешування.

Список лабораторних робіт із курсу «Бази даних для мережової інфраструктури»:

- Linux DNS
- Windows DNS
- Linux DHCP
- Windows DHCP
- Proху
- TOR

Для успішного їх виконання потрібна пара віртуальних машин, перша на ОС Linux, інша на Windows Server. Системні характеристики кожної з машин повинні бути на рівні рекомендованої конфігурації, або вище. Обов'язкова вимога – наявність спільної мережі між віртуальними машинами. На виконання кожної з лабораторних робіт відведено по 2 академічних годин, на самостійне опрацювання ще по 2 академічних години, і по 1 години на захист лабораторної роботи. Сумарно кожна віртуальна машина повинна працювати щонайменше 24 години.

1.3. Інтернет сервіси

Дана дисципліна познайомила студентів з тим, як працюють веб-сервери, поштові та файлові сервери, а також контейнеризація Docker

Зараз складно уявити життя без перелічених вище серверів.

Веб-сервер це майже основа сучасного інтернету, тому що він зберігає всю інформацію від файлів що відповідають за сам веб-ресурс, а й можливі файли що зберігаються там в якості сховища.

Була вивчена історія поштового сервісу та засвоєна, під час виконання лабораторних робіт, архітектура поштової мережі та який шлях долає лист щоб потрапити до адресата. Розглянуті методи боротьби зі спамом.

На лабораторному практикумі розібрали, по яких протоколах передачі даних працюють файлові сервери, а також познайомились із зовсім новим поняттям контейнеризація. Дізнались її переваги та можливості. Адже зараз вона широко використовується в хмарних технологіях, через властивість бути ізольованою від основної системи, що дозволяє завантажувати контейнер на обчислювальний сервер, і після того як він уже не потрібний, просто видалити його, і сервер залишився без змін.

Курс лабораторних робіт з предмету «Інтернет сервіси» складається з 8 лабораторних робіт

- Робота з SSH
- Веб-сервер Apache
- Веб-сервер Nginx
- Знайомство з протоколом SMTP
- Інсталяція поштового сервісу
- Налаштування файлового сервісу FTP
- Налаштування та дослідження файлових сервісів NFS та Samba
- Налаштування та створення Docker контейнерів

Виконання кожної з лабораторних робіт передбачає використання віртуальної машини на ОС Linux, а саме з дистрибутивом CentOS. Віртуальна машина повинна відповідати характеристикам типової конфігурації. Сумарний час роботи віртуальної машини повинен складати щонайменше 32 години, з міркувань що на виконання і опрацювання виділено по 4 а.г. і на захист роботи по 1 годині.

1.4. Технологічна практика

На 6 семестрі навчання, студенти отримують певне завдання, щоб виконати його їм необхідно самостійно освоїти за семестр, весь наданий викладачами матеріал.

Фактично технологічна практика являється однією з форм самоосвіти, але з проміжним контролем такі як тести, і фінальний захист проекту.

Для створення і захисту студенти об'єднуються в групи, тим самим звикаючи до спільної розробки додатків і роботи в команді. Мета технологічної практики була вивчення мови Python і розгортання веб ресурсу на фреймворку Django.

Але, на мій погляд, такий написаний веб-додаток - це теж саме, що й звичайний додаток, але який відкривається в браузері. Щоб він став справжнім веб-додатком його необхідно завантажити на будь-який веб-сервер, що можливо зробити на будь-яких стадіях розробки за допомогою будь-якого хмарного сервісу.

1.5. Системне програмне забезпечення

Даний курс частково перетинається з вище розглянутими дисциплінами. Кожна лабораторна робота даного практикума має відповідного близнюка в інших дисциплінах. Але є між ними і різниця, тоді більшість сервісів налаштовували виключно на ОС Linux, а зараз настала черга Windows Server.

Теми лабораторних робіт хоч і були однакові, але завдання зовсім відрізнялись від попередніх робіт, тому що в цьому курсі розглядалось поняття як адміністрування, а не конфігурація. Адміністрування виконувалось за допомогою вбудованих в ОС засобів.

Розглянули LDAP та Active Directory від Microsoft. Створювали користувачів. Змінювали їхні ролі та групи, та можливості.

За допомогою вбудованого в ОС серверу IIS(Internet Information Services) розгортали власний веб-сервер налаштовували порти підключення, а також додавали сертифікат для можливості підключатись за допомогою протоколу HTTPS.

Курс лабораторних робіт із «Системного програмного забезпечення» включає в себе виконання 6 лабораторних робіт виключно на віртуальній машині з ОС Windows Server типової конфігурації та вище. На виконання відведено по 4 академічних години, і на самостійне опрацювання також 4 а.г.. Враховуючи те, що на захист кожної з робіт відведена 1 година, можна стверджувати що віртуальна машина повинна в загальному працювати 42 години.

Даний лабораторний практикум виконувався на віртуальних машинах Windows Server, що знаходяться на факультеті, тому не було помітно дискомфорту від нехватки ресурсів під час запуску другої ОС. Особливим є те, що виконувались лабораторні дистанційно, а підключення до ВМ за допомогою протоколу RDP.

1.6.Вимоги до хмарних рішень

З аналізу лабораторних робіт, можна дійти висновку, що хмарні ресурси повинні забезпечувати можливість створення декількох віртуальних машин, які повинні відповідати типовій конфігурації, і час життя машини повинен знаходитись в межах часу, який потрібен на виконання лабораторних робіт.

2. Аналіз хмарних рішень для виконання лабораторних робіт

Хмарні обчислення (англ. cloud computing) — це модель забезпечення повсюдного та зручного мережевого доступу на вимогу до загального пулу конфігуруємих обчислювальних ресурсів (наприклад, мереж передачі даних, серверів, пристроїв зберігання даних, додатків і сервісів - як разом, так і окремо) , які можуть бути оперативно надані і звільнені з мінімальними експлуатаційними витратами і / або зверненнями до провайдера.

Великі хмарні постачальники часто мають функції, розподілені в різних місцях, кожне місце є дата-центром. Хмарні сервіси покладаються на спільне використання ресурсів для досягнення когерентності та зазвичай використовують властивості так-званих «еластичних обчислень», тобто модель «оплати по мірі використання». Завдяки цьому вдається зменшити капітальні витрати, але ця технологія може призвести до неочікуваних операційних витрат для недосвідчених користувачів. Національним інститутом стандартів та технологій США зафіксовані наступні обов'язкові характеристики хмарних обчислень:

- Самообслуговування на вимогу - споживач самостійно визначає свої обчислювальні потреби: серверний час, швидкості доступу та обробки даних, обсяг даних, що зберігаються - без взаємодії з представником постачальника послуг;
- Універсальний доступ до мережі — послуги доступні споживачам через мережу передачі даних незалежно від термінального пристрою, що використовується;
- Об'єднання ресурсів - постачальник послуг об'єднує ресурси для обслуговування великого числа споживачів в єдиний пул для динамічного перерозподілу потужності між споживачами в умовах постійної зміни запиту на потужність; при цьому користувачі керують тільки основними параметрами послуг (наприклад,

об'ємом даних, швидкість доступу), але фактичний розподіл ресурсів, наданих споживачам, здійснює постачальник (у деяких випадках користувачі все-таки можуть керувати деякими фізичними параметрами перерозподілу, наприклад, вказують бажаний центр обробки даних з міркувань географічної близькості);

- Еластичність — послуги можуть бути надані, розширені, звужені у будь-який час, без додаткових витрат за взаємодію з постачальником, зазвичай, у автоматичному режимі;
- Облік споживання — постачальник послуг автоматично обчислює спожиті ресурси на певному рівні абстракції (наприклад, обсяг даних, що зберігаються, пропускна спроможність, кількість користувачів, кількість транзакцій) і на основі цих даних оцінює обсяг наданих споживачам послуг

2.1. Типи хмарних обчислень.

Типи хмарних обчислень — це моделі розгортання служб, які забезпечують різний рівень контролю над інформацією та різні типи служб, що надаються. Існує три основних типи хмарних обчислень, котрі зображені на рисунку 1 (іще декілька типів знаходяться на стадії розробки): програмне забезпечення як послуга (SaaS) для веб-додатків, інфраструктура як послуга (IaaS) для доступу до сховища та обчислювальних ресурсів через інтернет, платформа як послуга (PaaS), яка надає розробникам засоби для розробки та розміщення веб-додатків. А також існує без серверна модель обчислення.



Рисунок 1- Ієрархія хмарних послуг

2.1.1 IaaS

IaaS — це доступ на вимогу до хмарної обчислювальної інфраструктури — серверів, ємності сховища та мережевих ресурсів — які клієнти можуть надавати, налаштовувати та використовувати майже так само, як і на власному локальному обладнанні. Різниця полягає в тому, що постачальник хмарних послуг розміщує, керує та обслуговує апаратні та обчислювальні ресурси у своїх власних центрах обробки даних. Клієнти IaaS використовують апаратне забезпечення через Інтернет-з'єднання і оплачують це використання на основі підписки або оплати за використання ресурсів (запити, кількість обрахунків і т.п.).

Зазвичай клієнти IaaS можуть вибирати між віртуальними машинами (VM), розміщеними на спільному фізичному обладнанні (постачальник хмарних послуг керує віртуалізацією) або відкритими серверами на виділеному (не спільному доступі) фізичному обладнанні. Клієнти можуть надавати, налаштовувати та керувати серверами та ресурсами інфраструктури за допомогою графічної інформаційної панелі або програмно за допомогою інтерфейсів прикладного програмування (API).

IaaS можна розглядати як традиційне рішення, але надане як послугу.

Кожний великий постачальник хмарних послуг — Amazon Web Services, Google Cloud, IBM Cloud, Microsoft Azure — пропонує певні форми IaaS.

2.1.1.1 Переваги IaaS

Порівняно з традиційними IT, IaaS надає клієнтам більшу гнучкість, створюючи обчислювальні ресурси за потреби, а також їх масштабування вгору або вниз у відповідь на стрибки або уповільнення трафіку. IaaS дозволяє клієнтам уникнути авансових витрат і накладних витрат на придбання та обслуговування власного локального центру обробки даних. Це також усуває постійний компроміс між витратами на закупівлю надлишкових локальних потужностей для врегулювання стрибків у порівнянні з низькою продуктивністю або відключеннями, які можуть бути наслідком недостатньої потужності для непередбачуваних сплесків або зростання трафіку.

Інші переваги IaaS включають:

- Вища доступність: за допомогою IaaS компанія може легко створювати резервні сервери і навіть створювати їх в інших регіонах, щоб забезпечити доступність під час локальних відключень електроенергії або фізичних катастроф.
- Менша затримка, покращена продуктивність. Оскільки постачальники послуг IaaS зазвичай керують центрами обробки даних у кількох регіонах, клієнти IaaS можуть розташовувати програми та послуги ближче до користувачів, щоб мінімізувати затримки та максимізувати продуктивність.
- Покращена чутливість. Клієнти можуть надавати ресурси за лічені хвилини, швидко тестувати нові ідеї та швидко розгорнути нові ідеї для більшої кількості користувачів.
- Комплексна безпека. Завдяки високому рівню безпеки на місці, в центрах обробки даних і за допомогою шифрування, клієнти часто можуть скористатися перевагами більш розширеного захисту та

захисту, який вони могли б забезпечити, якби вони розташували хмарну інфраструктуру всередині власної компанії.

- Швидший доступ до найкращих технологій. Хмарні постачальники конкурують один з одним, надаючи своїм користувачам новітні технології, клієнти IaaS можуть скористатися цими технологіями набагато раніше (і за набагато меншою ціною), ніж вони можуть впроваджували б їх на місці.

2.1.1.2 Випадки використання IaaS

- Аварійне відновлення: замість того, щоб налаштовувати резервні сервери в кількох місцях, IaaS може розгорнути своє рішення для аварійного відновлення в існуючій територіально розподіленій інфраструктурі постачальника хмарних послуг.
- Електронна комерція: IaaS є відмінним варіантом для онлайн-роздрібних продавців, які часто бачать стрибки трафіку. Здатність розширюватися в періоди високого попиту та високоякісна безпека є надзвичайно важливими в сучасній цілодобовій роздрібній індустрії.
- Інтернет речей (IoT), обробка подій, штучний інтелект (AI) : IaaS полегшує налаштування та масштабування сховищ даних та обчислювальних ресурсів для цих та інших програм, які працюють із величезними обсягами даних.
- Стартапи: стартапи не можуть дозволити собі вкладати капітал у локальну IT-інфраструктуру. IaaS надає їм доступ до можливостей центрів обробки даних корпоративного класу без авансових інвестицій в обладнання та накладні витрати на керування.
- Розробка програмного забезпечення: за допомогою IaaS інфраструктуру для тестування та середовища розробки можна налаштувати набагато швидше,

ніж локальну. (Однак цей варіант використання краще підходить для PaaS, як ви прочитаєте в наступному розділі.)

2.1.2 PaaS

PaaS надає хмарну платформу для розробки, запуску та керування додатками. Постачальник хмарних послуг розміщує, керує та підтримує все апаратне та програмне забезпечення, включене в платформу - сервери (для розробки, тестування та розгортання), програмне забезпечення операційної системи (ОС), сховище, мережі, бази даних, проміжне програмне забезпечення, середовища виконання, фреймворки, засоби розробки, а також супутні послуги з безпеки, оновлення операційної системи та програмного забезпечення, резервне копіювання тощо.

Користувачі отримують доступ до PaaS через графічний інтерфейс користувача (GUI), де команди розробників або DevOps можуть співпрацювати над усією своєю роботою протягом усього життєвого циклу програми, включаючи кодування, інтеграцію, тестування, доставку, розгортання та зворотний зв'язок.

Приклади рішень PaaS: AWS Elastic Beanstalk, Google App Engine, Microsoft Windows Azure.

2.1.2.1 Переваги PaaS

Основна перевага PaaS полягає в тому, що вона дозволяє клієнтам створювати, тестувати, розгортати запускати, оновлювати та масштабувати програми швидше та рентабельніше, ніж вони могли б, якби їм довелося створювати та керувати власною локальною платформою. Інші переваги включають:

- Швидший час виходу на ринок: PaaS дозволяє командам розробників розширювати середовище розробки, тестування та

виробництва за кілька хвилин, а не за тижні чи місяці.

- Тестування з низьким рівнем ризику та впровадження нових технологій: платформи PaaS, як правило, включають доступ до широкого спектру найновітніших ресурсів у стеку додатків. Це дозволяє компаніям тестувати нові операційні системи, мови та інші інструменти без значних інвестицій в них або в інфраструктуру, необхідну для їх запуску.
- Спрощена співпраця: як хмарна служба, PaaS надає спільне середовище розробки програмного забезпечення, надаючи командам розробників та операційних груп доступ до всіх необхідних інструментів з будь-якого місця, де є підключення до Інтернету.
- Більш масштабований підхід: за допомогою PaaS організації можуть придбати додаткову потужність для створення, тестування, встановлення та запуску додатків, коли їм це потрібно.
- Менше керування: PaaS передає управління інфраструктурою, виправлення, оновлення та інші адміністративні завдання на постачальника хмарних послуг.

2.1.2.2 Випадки використання PaaS

- Розробка API та керування ним: завдяки вбудованим фреймворкам PaaS полегшує командам розробку, запуск, керування та захист API для обміну даними та функціями між додатками.
- Інтернет речей (IoT): PaaS підтримує низку мов програмування (Java, Python, Swift тощо), інструментів і середовищ додатків, які використовуються для розробки додатків IoT та обробки даних із пристроїв IoT в режимі реального часу.
- Гнучка розробка та DevOps: рішення PaaS зазвичай покривають усі вимоги ланцюга інструментів DevOps і забезпечують вбудовану автоматизацію для підтримки безперервної інтеграції та безперервної доставки (CI/CD).

- Хмарна розробка та гібридна хмарна стратегія: рішення PaaS підтримують технології розробки на основі хмари — мікросервіси , контейнери , Kubernetes , безсерверні обчислення — що дають змогу розробникам створювати один раз, а потім розгорнути й послідовно керувати у приватних хмарах , загальнодоступних хмарах та локальних середовищах.

2.1.3 SaaS

SaaS (іноді його називають *хмарними додатками*) — це готове до використання програмне забезпечення, розміщене в хмарі. Користувачі сплачують щомісячну або річну плату за використання повної програми з веб-переглядача, клієнта для настільних комп'ютерів або мобільного додатка. Програма та вся інфраструктура, необхідна для її доставки — сервери, сховище, мережа, проміжне програмне забезпечення, прикладне програмне забезпечення, сховище даних — розміщуються та керуються постачальником SaaS.

Постачальник керує всіма оновленнями та виправленнями програмного забезпечення, зазвичай непомітно для клієнтів. Як правило, постачальник забезпечує рівень доступності, продуктивності та безпеки в рамках угоди про рівень обслуговування (SLA). За потреби клієнти можуть додати більше користувачів і сховище даних за додаткову плату.

Сьогодні кожен, хто користується смартфоном, використовує якусь форму SaaS. Електронна пошта, соціальні мережі та рішення для зберігання файлів у хмарі (наприклад, Dropbox або Google Drive) є прикладами програм SaaS, які люди використовують щодня у своєму особистому житті.

2.1.3.1 Переваги SaaS

Основна перевага SaaS полягає в тому, що він передає всю інфраструктуру та керування додатками постачальнику SaaS. Все, що

користувач повинен зробити, це створити обліковий запис, сплатити комісію та почати користуватися програмою. Постачальник займається всім іншим, від обслуговування апаратного та програмного забезпечення сервера до керування доступом і безпекою користувачів, зберігання та керування даними, впровадження оновлень і виправок тощо.

Інші переваги SaaS включають:

- **Мінімальний ризик:** багато продуктів SaaS пропонують безкоштовний пробний період або низьку щомісячну плату, що дозволяє клієнтам випробувати програмне забезпечення, щоб побачити, чи воно відповідатиме їхнім потребам, з невеликим фінансовим ризиком або без нього.
- **Продуктивність у будь-який час і будь-де:** користувачі можуть працювати з додатками SaaS на будь-якому пристрої з браузером та підключенням до Інтернету.
- **Легка масштабованість:** додавання користувачів так само просто, як реєстрація та оплата за нові місця; клієнти можуть придбати більше сховища даних за деяку плату.

Деякі постачальники SaaS навіть дозволяють налаштувати свій продукт, надаючи супутнє рішення PaaS. Одним із відомих прикладів є Heroku, рішення PaaS для Salesforce.

2.1.3.2 Випадки використання SaaS

Сьогодні майже будь-яка програма підвищення продуктивності особистого або службового персоналу доступна як SaaS; конкретні випадки використання занадто численні, щоб їх згадувати (деякі з них перераховані вище). Якщо кінцевий користувач або організація може знайти рішення SaaS з необхідною функціональністю, у більшості випадків воно надасть значно простішу, масштабнішу та більш економічно ефективну альтернативу локальному програмному забезпеченню.

2.1.4 SaaS проти PaaS проти IaaS

SaaS, PaaS, IaaS не є взаємовиключними; більшість організацій використовують більше одного, і багато великих організацій сьогодні використовують усі три, часто в поєднанні з традиційним підходом.

Очевидно, що рішення як послуга, яке вибирає клієнт, залежить насамперед від функціональності, яку потребує клієнт, і досвіду персоналу, який він має. Наприклад, організація без власних ІТ-спеціалістів для налаштування та експлуатації віддалених серверів погано підходить для IaaS; організація без команди розробників не потребує PaaS.

Але в деяких випадках будь-яка з трьох моделей «як-послуга» запропонує життєздатне рішення. У цих випадках організації зазвичай порівнюють альтернативи на основі простоти управління, яку вони пропонують, з контролем, від якого вони відмовляються.

Наприклад, припустимо, що велика організація хоче надати програму керування взаємовідносинами з клієнтами (CRM) своїй команді продажів. Це може:

- *SaaS CRM* , передавши все повсякденне управління стороннім постачальникам, але також відмовившись від усього контролю над функціями та функціональністю, зберіганням даних, доступом користувачів і безпекою.
- *PaaS і створення спеціальної програми CRM* . У цьому випадку компанія переклала б управління інфраструктурою та ресурсами розробки додатків до постачальника хмарних послуг. Замовник збереже повний контроль над функціями програми, але також візьме на себе відповідальність за керування програмою та пов'язаними з нею даними.
- Створення серверної ІТ-інфраструктури в хмарі за допомогою IaaS і використання її для створення власної платформи розробки та

програми. IT-команда організації матиме повний контроль над операційними системами та конфігураціями серверів, але також буде нести тягар їх керування та підтримки, а також платформи розробки та програм, які на них працюють.

2.1.5 Безсерверні обчислення

Безсерверна модель виконання в хмарі, яка дає змогу створювати та керувати хмарними додатками простішим та рентабельнішим. Дана модель характерна тим, що:

- Автоматично надає обчислювальні ресурси, необхідні для виконання коду програми на вимогу або у відповідь на певну подію;
- Автоматично збільшує або зменшує ці ресурси у відповідь на збільшення або зменшення попиту;
- Автоматично масштабує ресурси до нуля, коли програма припиняє роботу.

Безсерверна модель передає всю відповідальність за керування серверною хмарною інфраструктурою та операційними завданнями на постачальника хмари. Це дає розробникам більше часу для розробки та оптимізації коду інтерфейсу програми та бізнес-логіки. А без сервера клієнти ніколи не платять за вільну потужність. Вони платять лише за ресурси, необхідні для запуску їхніх програм, і лише тоді, коли ці програми запущені.

Незважаючи на назву, в безсерверних обчисленнях безсумнівно є сервери. Термін «безсерверний» описує досвід клієнта з цими серверами: вони невидимі для клієнта, який не бачить їх, не керує ними і не взаємодіє з ними будь-яким чином.

Як і кожна модель безсерверні обчислення мають свою переваги і недоліки, до переваг можна віднести наступне:

Безсерверні обчислення надають ряд індивідуальних технічних і бізнес-вигод:

- Як зазначалося вище, безсерверність дозволяє командам розробників зосередитися на написанні коду, а не на управлінні інфраструктурою. Це дає розробникам набагато більше часу для інновацій та оптимізації функціональних можливостей програм.
- Крім того, як зазначалося вище, клієнти без сервера платять лише за виконання. Лічильник запускається, коли зроблено запит, і закінчується, коли виконання закінчується. Порівняніючи з IaaS, коли клієнти платять за віртуальні машини (VM) та інші ресурси, необхідні для запуску програм, з моменту надання цих ресурсів до моменту, коли клієнт явно виводить ці ресурси з експлуатації.
- Безсерверне середовище – це поліглотове середовище, що дозволяє розробникам використовувати будь-яку мову або фреймворк – Java , Python, node.js – з якими їм зручно.
- Безсерверний режим спрощує розгортання і, в більшому сенсі, спрощує цикли DevOps , оскільки розробникам не потрібно описувати інфраструктуру, необхідну для інтеграції, тестування, доставки та розгортання коду, вбудованого у виробництво.
- Для певних робочих навантажень, наприклад, які вимагають паралельної обробки, безсерверний підхід може бути швидшим і рентабельнішим, ніж інші форми обчислень.
- Безсерверні платформи розробки додатків забезпечують майже повне бачення часу системи та користувачів і можуть систематично збирати цю інформацію.

Недоліки безсерверних обчислень:

- Стабільні або передбачувані робочі навантаження: оскільки безсерверне масштабування збільшується та зменшується на вимогу у відповідь на робоче навантаження, це забезпечує значну економію витрат для різких робочих навантажень. Але це не дає такої ж економії для робочих навантажень, які характеризуються

передбачуваними, постійними або тривалими процесами; у цих випадках традиційне серверне середовище може бути простішим і рентабельнішим.

- **Холодний старт:** оскільки безсерверні архітектури відмовляються від довготривалих процесів на користь масштабування вгору та зменшення до нуля, їм також іноді потрібно запуснутися з нуля, щоб обслуговувати новий запит. Для певних програм ця затримка запуску не є помітною або шкідливою для користувачів. Але для інших – наприклад, для фінансової торгової програми – затримка неприпустима.
- **Моніторинг і налагодження.** Ці операційні завдання є складними в будь-якій розподіленій системі, але архітектура без сервера лише посилює складність.
- **Блокування постачальника:** безсерверні архітектури розроблені, щоб скористатися перевагами екосистеми керованих хмарних служб і, з точки зору архітектурних моделей, максимально відокремити робоче навантаження від чогось більш портативного, наприклад віртуальної машини (VM) або контейнера Docker . Для деяких компаній глибока інтеграція з власними керованими службами певної хмарної платформи – це те, де можна знайти велику частину цінності хмари; для інших ця хмара призводить до матеріальних ризиків, які необхідно пом'якшити.

2.1.6 Публічні хмари

Публічна хмара — це тип хмарних обчислень, у якому постачальник хмарних послуг робить доступними користувачам обчислювальні ресурси — будь-що, від додатків SaaS до окремих віртуальних машин (VM), до лише комп'ютерного обладнання, до повних інфраструктур корпоративного рівня та

платформ розробки, через мережу Інтернет. Ці ресурси можуть бути доступними безкоштовно, або доступ може продаватись відповідно до моделей ціноутворення на основі підписки або плати за використання.

Постачальник загальнодоступної хмари володіє, керує та бере на себе всю відповідальність за центри обробки даних, обладнання та інфраструктуру, на якій виконуються робочі навантаження своїх клієнтів, і зазвичай забезпечує високошвидкісне мережеве підключення для забезпечення високої продуктивності та швидкого доступу до програм і даних.

Публічна хмара — це середовище з багатьма орендарями — інфраструктура центру обробки даних хмарного постачальника використовується всіма клієнтами загальнодоступної хмари. У провідних загальнодоступних хмарах — Amazon Web Services (AWS), Google Cloud, IBM Cloud, Microsoft Azure та Oracle Cloud — таких клієнтів може бути мільйони.

Багато підприємств переміщують частину своєї обчислювальної інфраструктури в загальнодоступну хмару, оскільки загальнодоступні хмарні послуги еластичні та легко масштабуються, гнучко пристосовуються до мінливих вимог робочого навантаження. Інших приваблює обіцянка більшої ефективності та меншої втрати ресурсів, оскільки клієнти платять лише за те, що вони використовують. Треті намагаються зменшити витрати на обладнання та локальну інфраструктуру.

2.1.7 Приватні хмари

Приватна хмара – це хмарне середовище, в якому вся хмарна інфраструктура та обчислювальні ресурси призначені та доступні лише одному клієнту. Приватна хмара поєднує багато переваг хмарних обчислень, зокрема еластичність, масштабованість та простоту надання послуг, із контролем доступу, безпекою та налаштуванням ресурсів локальної інфраструктури.

Приватна хмара зазвичай розміщується локально в центрі обробки даних

клієнта. Але приватна хмара також може бути розміщена в інфраструктурі незалежного хмарного постачальника або побудована на орендованій інфраструктурі, розміщеній у центрі обробки даних за межами сайту.

Багато компаній вибирають приватну хмару, а не загальнодоступну, тому що приватна хмара є простішим способом (або єдиним способом) задовольнити їхні вимоги щодо відповідності нормативним вимогам. Інші вибирають приватну хмару, оскільки їх робоче навантаження стосується конфіденційних документів, інтелектуальної власності, особистої інформації, медичних записів, фінансових даних або інших конфіденційних даних.

Створюючи архітектуру приватної хмари відповідно до принципів, розроблених для хмари, організація дає собі гнучкість, щоб легко переносити робочі навантаження в загальнодоступну хмару або запускати їх у середовищі гібридної хмари (див. нижче), коли вони готові

2.1.8 Гібридні хмари

Гібридна хмара — це саме те, як це звучить — поєднання загальнодоступного та приватного хмарного середовища. Зокрема, і в ідеалі, гібридна хмара з'єднує приватні хмарні служби організації та загальнодоступні хмари в єдину гнучку інфраструктуру для виконання програм і робочих навантажень організації.

Мета гібридної хмари полягає в тому, щоб створити поєднання загальнодоступних і приватних хмарних ресурсів — з можливістю диригувати ними — що надає організації гнучкості у виборі оптимальної хмари для кожної програми або робочого навантаження та вільно переміщати робочі навантаження між хмарами в залежності від навантаження. Це дозволяє організації досягати своїх технічних та бізнес-цілей більш ефективно та рентабельно, ніж це було б із загальнодоступною чи приватною хмарою.

2.1.9 Мульти-хмара та гібридна мульти-хмара

Мультихмарність — це використання двох або більше хмар від двох або більше різних хмарних постачальників. Наявність мультихмарного середовища може бути настільки простим, як використання електронної пошти SaaS від одного постачальника та редагування зображень SaaS від іншого. Але коли підприємства говорять про багатохмарність, вони зазвичай говорять про використання кількох хмарних сервісів, включаючи послуги SaaS, PaaS та IaaS, від двох або більше провідних постачальників загальнодоступних хмар.

Гібридна мультихмара – це використання двох або більше загальнодоступних хмар разом із приватним хмарним середовищем.

Організації вибирають мультихмарний режим, щоб уникнути прихильності до постачальників, мати більше послуг на вибір та отримати доступ до більшої кількості інновацій. Але чим більше хмар ви використовуєте — кожна зі своїм набором інструментів керування, швидкості передачі даних і протоколів безпеки — тим складніше може бути керувати вашим середовищем. Багатохмарні платформи керування забезпечують видимість у різних хмарах постачальників через центральну інформаційну панель, де команди розробників можуть бачити свої проекти та розгортання, оперативні групи можуть стежити за кластерами та вузлами, а персонал із кібербезпеки може відстежувати загрози.

2.2. Amazon Web Services

AWS(Amazon Web Services)[1] - це хмарна платформа із широкими можливостями, кількість повністю функціональних сервісів для обробки даних сягає більше 200 одиниць. Перевагою AWS є найбільш обширна хмарна інфраструктура по всьому світу, що забезпечує високу пропускну здатність із низькою затримкою.

Постачальник хмарних ресурсів пропонує новим користувачам ознайомитись із продуктами надаючи безкоштовний термін на 1 рік, і 200 кредитів за допомогою яких можна використовувати хмарні сервіси. Пропозиція для студентів із зменшенням вартості користування хмарним середовищем відсутня, тому в подальшому розділі було розглянуто безкоштовний пакет послуг, та пакети послуг на платній основі, де вони були необхідними.

Проаналізувавши пропозиції від даної хмарної платформи, можна відокремити певні рішення за допомогою яких, можна виконувати лабораторні роботи. Amazon EC2 є рішенням, що забезпечує створення віртуальних машин, для курсів «Інтернет сервіси», «Бази даних для мережевої інфраструктури» та «Системного програмного забезпечення». Amazon VPC обов'язковий компонент для лабораторних робіт в яких обов'язкова внутрішня мережа між машинами. Для розробки веб-застосунку який розробляється групою студентів під час технологічної практики є декілька інструментів, які можуть збільшити продуктивність виконання і тестування застосунку : AWS Amplify, AWS Device Farm, Amazon Lightsail.

Amazon EC2 (Elastic Compute Cloud) сервіс що може запропонувати широкий набір сучасних процесорів, сховищ, мереж, операційних систем забезпечуючи при цьому підтримку робочого налаштування. EC2 підтримує роботу процесорів Intel, AMD, Arm, а також єдине хмарне рішення що може запропонувати віртуальні машини на MacOS і за потреби може забезпечити швидкість з'єднання з мережею Інтернет в 400 Гбіт/с.

Образи операційних систем AMI(Amazon Machine Image) попередньо налаштовані. AMI постійно поповнюються на теперішній момент доступні Windows, Amazon Linux 2, Ubuntu, Red Hat Ent., CentOs, SUSE, Debian.[2]

EC2 входить в безкоштовний пакет послуг. Провайдер надає можливість використовувати віртуальну машину протягом 750 годин. Системні характеристики безкоштовної VM:

- 1 ядро процесора Intel Xeon E5-2686 v4 2.3GHz

- 1 ГБ оперативної пам'яті
- 30ГБ на SSD-накопичувачі

Ці системні характеристики чудово підпадають під типову конфігурацію ОС Linux. І їх буде достатньо для виконання лабораторних робіт з курсу «Інтернет сервіси». Для решти курсів необхідна наявність віртуальної машини на базі Windows Server. Враховуючи системні вимоги типової конфігурації і час на виконання курсів, а це близько 70 годин на решту курсів, то вартість її використання буде складати до 5 доларів США, не враховуючи те що 3.5 \$ потрібно платити що місяця за використання певного об'єму сховища. В загальному витрати студента навіть при використанні двох платних віртуальних машин не будуть перевищувати 50 доларів США за семестр.

AWS VPC – це сервіс, який дає можливість запускати ресурси AWS у логічно ізольованій віртуальній мережі, що визначається користувачем. Цей сервіс є необхідний для лабораторних робіт з курсу «Бази даних для мережевої інфраструктури», щоб забезпечити спільну мережу між віртуальними машинами. Адже цей сервіс дозволяє повністю контролювати середовище віртуальної мережі, у тому числі вибирати власний діапазон IP-адрес, створювати підмережі, а також налаштовувати таблиці маршрутизації та мережеві шлюзи. Для більшості ресурсів VPC можна використовувати і IPv4, і IPv6, і таким чином отримати безпечний і зручний доступ до ресурсів і додатків.[3]

Інструменти безкоштовного пакету для розробки веб-застосунків:

- AWS Amplify 15 Гб вихідного трафіку на місяць, протягом 1 року
- AWS Device Farm лише 1000 хвилин використання пристроїв для тестування
- Amazon Lightsail надається 750 годин протягом 3 місяців безкоштовно

AWS Amplify - це комплекс спеціальних інструментів які дозволяють розробникам швидко і просто створювати програми для Інтернету чи для мобільних пристроїв. Також за допомогою цього сервісу можна

налаштовувати серверну частину для додатку, і швидко підключити його. Сервіс також має візуальну середу розробки, що пришвидшує заповнення додатка контентом.[4]

AWS Device Farm - це сервіс для тестування програм, призначений щоб покращити швидкодію мобільних та веб-застосунків. Перевага даного сервісу заключається в тому, що не має необхідності створювати власне фізичне середовище для тесту, так як всі тести виконуються на реальних пристроях і веб-браузерах. І більш того користувач має змогу вибрати та налаштувати пристрої під власні потреби, тим самим моделюючи реальні умови, такі як тип мережевого підключення, геолокація, налаштування мови і т.д. AWS Device Farm має вбудований пакет для тестування(тобто без створення сценарію), але він може бути модифікований за бажанням користувача, або користувач може провести тестування вручну за допомогою віддаленого доступу до пристрою[3]

Amazon Lightsail може бути корисним в якості приватного сервера, контейнера, сховища, бази даних. Приклади використання:

- Запуск інтернет додатків. Попередньо сконфігуровані стеки розробки такі як Node.js, Nginx. Але не має обмеження розгорнути додаток на іншому фреймворку наприклад на python Django
- Створення користувацьких веб-сайтів. Попередньо налаштовані такі CMS(система керування змістом): Wordpress, Magento, Prestashop, Joomla
- Запуск середовища для тестування. Можна переходити від одного середовища до іншого, це можуть бути як середовища тестування так і розробки. Середовища ізольовані одне від одного.[5]

2.3.Microsoft Azure

Microsoft Azure - це хмарна платформа яка включає приблизно 200 продуктів та хмарних служб[6]. В загальному обчислювальні функції Azure PaaS і IaaS.

Постачальник хмарних ресурсів має програму для студентів, в якій надається 100 кредитів без прив'язки банківської карти, і якщо кредити закінчується, то надаються ще більше 40 безкоштовних сервісів. Серед яких є 2 віртуальні машини, на базі Linux і Windows Server. Їх системні характеристики складаються з 1 vCPU 1Gb Ram і 4 gb тимчасового сховища, що не є задовільним для машин на базі Windows. Але 100 безкоштовних кредитів вистачає для забезпечення роботи двох віртуальних машин, що відповідають типовій конфігурацію, протягом семестру. Або використовувати одну безкоштовну і одну платну віртуальну машину протягом року. Після 12 місяців, є можливість заново зареєструватись і отримати підписку Azure for students при умові, що користувач залишається студентом.[7]

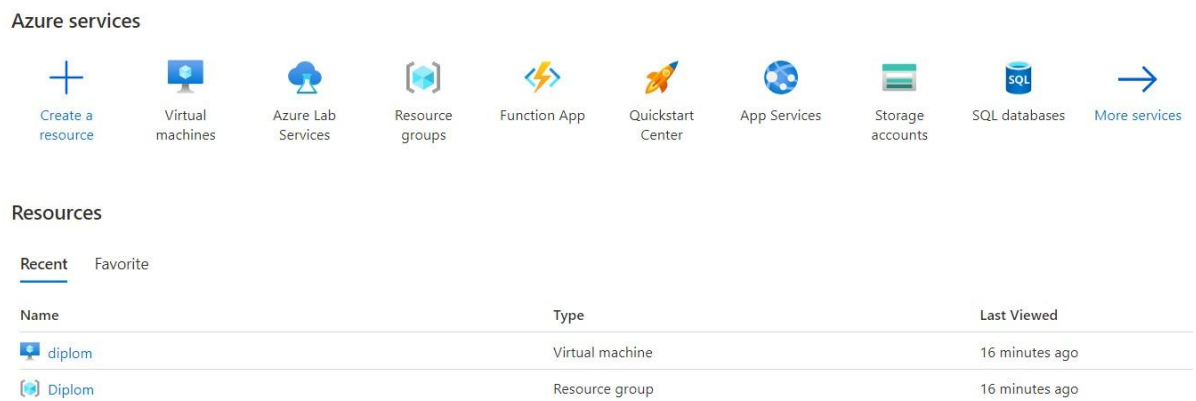


Рисунок 2 – Інтерфейс користувача Azure

Для виконання лабораторних робіт з курсів «Інтернет сервіси», «Системне програмне забезпечення» та «Бази даних для мережевої інфраструктури» при умові що використовуються віртуальні машини що відповідають типовій конфігурації і знаходяться в одній підмережі, то достатньо наступних рішень:

- Azure VMs[8]

- Windows VMs[9]
- Linux VMs[10]

Azure VMs - це чудовий приклад концепції IaaS. IaaS дає змогу майже повністю керувати інфраструктурою. Віртуальні машини підтримують два типа дисків: диск ОС та диск даних. Надійність яких забезпечується за допомогою служби сховища Azure . Диск ОС зберігає в собі лише операційну систему, а диск даних - сховище де можна зберігати різні дані, програми і т.д. Через те що користувач має широкий спектр можливостей, диски є постійними.

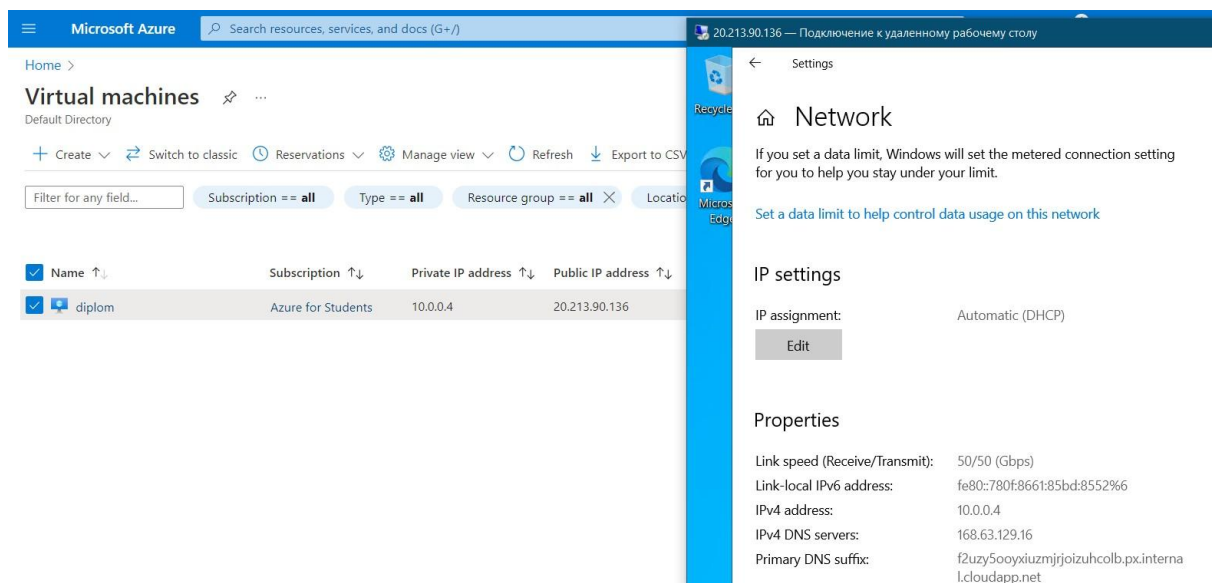


Рисунок 3 – Менеджер віртуальних машин і віртуальна машина

Для виконання розробки веб-застосунку на технологічній практиці, в рамках підписки Azure for students є завжди безкоштовні сервіси:

- Azure application service[11]
- Static Web application[12]

Azure application service - це мережева служба, що дозволяє швидко і без додаткових зусиль створювати мобільні та веб-застосунки. Підтримує можливість працювати з .NET, .NET Core, Node.js, Java, Python чи PHP в контейнерах, або власне на активній віртуальній машині Windows чи Linux.

Сервіс має як і безкоштовну так і платну основу:

- Безкоштовна основа включає в себе можливість створити 10 додатків, дисковий простір 1 Гб. Тип обчислень спільний.
- Максимальний пакет платних послуг в регіоні Північна Європа коштує 0.3\$/годину, чого вистачить на 660 годин. В нього входить необмежена кількість додатків; дисковий простір в 1Тб; максимальна кількість серверних екземплярів: до 100; підтримка власного домену; автомасштабування; гібридне підключення; власна віртуальна мережа; наявність мережевого інтерфейсу з віртуальної мережі; та тип обчислень ізольований.

Static web application - підтримує більшість фреймворків для роботи з веб-додатками: Angular, React, Vue, і т.д. Окрім створення інтерфейсу користувача є можливість створювати безсерверні інтерфейси API Azure, на основі наступних мов: JavaScript, TypeScript, Python і C#.

Має лише два можливі варіанти:

- Безкоштовний, завжди. Максимальна пропускна здатність 100Гб за підписку. 2 домена на веб-додаток. Наявність SSL сертифіката. Максимальний об'єм сховища 0.5ГБ на додаток. І можливість використовувати функції Azure
- Платний тариф відрізняється від безкоштовного можливістю перевищити ліміт пропускної здатності, 5 доменів на додаток, сховище в 2Гб на додаток, і можливістю використовувати власні API. Максимальна плата в місяць становить \$17.52 за додаток в місяць, а базова вартість \$9 за додаток в місяць

2.4. Google cloud

Google cloud - це набір сервісів і програмного забезпечення в які входять Google workspace(G suite), промислові версії Android і Chrome OS, різні API, машинне навчання(нейронні мережі) і Google maps[12]

Google workspace(G Suite) - це набір інструментів для хмарного обчислення, колаборації, розробки програмного забезпечення і продукти розробки. До складу G Suite входять:

- Засоби спілкування і колаборації
 - Gmail - поштовий сервіс.
 - Google meet - як сервіс аудіо- відеозв'язку.
 - Безпечний сервіс для спілкування у командах.
 - Календарі якими можна ділитись
- *Функціонал для спільної роботи*
 - Редактор документів, презентацій та таблиць.
 - Спрощене середовище розробки програм.
- *Рішення для зберігання інформації та її пошуку*
 - Необмежений простір для зберігання.
 - Пошук контенту в сервісах G Suite за допомогою Cloud Search.
- Інструменти керування і підтримка
 - Цілодобова підтримка за телефоном та електронною поштою.
 - Інструменти безпеки та управління доменом.

2.4.1. Google Cloud Platform GCP

Хмарна інфраструктура Google Cloud Platform. Є базою для таких високошвидкісних глобальних сервісів, як Gmail, Карти, YouTube та Пошук, а

також дозволяє працювати з різними інструментами для обчислень та хостингу.

Google Cloud Platform (GCP) розвивається в трьох самостійних напрямках:

- IaaS
- PaaS
- Безсерверні обчислення

GCP має безкоштовний пакет послуг, який надається протягом трьох місяців, і на цей період надаються 300 безкоштовних кредитів. І в загальному надається безкоштовна віртуальна машина на базі Лінукс із обчислювальною здатністю 0.25vCPU 1Gb RAM. Віртуальні машини на базі Windows Server не обхідно оплачувати безкоштовними кредитами або власними коштами. Тому не можна розцінювати дану платформу, як повністю безкоштовне рішення для виконання лабораторних робіт протягом семестру. Але вартість 2 віртуальних машин на 1 семестр складає до 35 доларів США, що менше чим в конкурентів таких як Amazon чи Microsoft. В загальному віртуальні машини на Google Cloud мають все необхідне для виконання лабораторних робіт

GCP як IaaS – інфраструктура як сервіс, надає змогу кінцевому користувачу взяти сервер в оренду, для подальшого розгортання власної інфраструктури. Під цю задачу найкраще підходить Google Compute Engine[13] в поєднанні із власним API. Даний сервіс дозволяє на серверах створювати віртуальні машини, на базі наступних ОС: Windows Server, Ubuntu, Red Hat Ent., CentOS, SUSE, Debian.

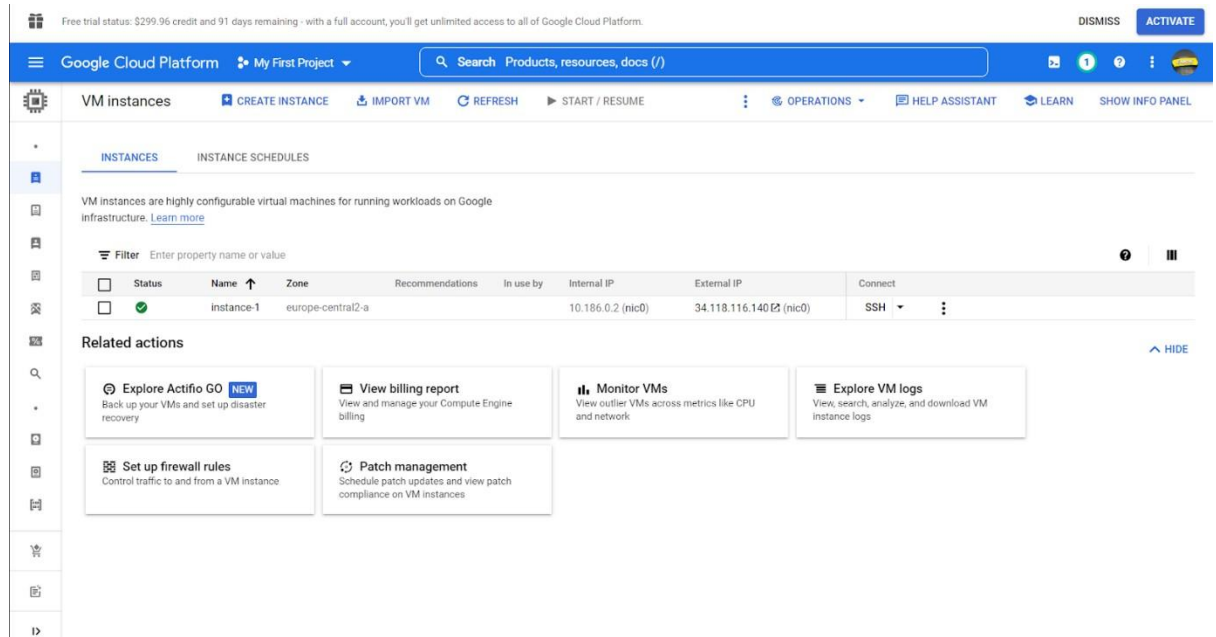


Рисунок 4 – Розділ віртуальних машин в сервісі GCP

GCP як PaaS – платформа як сервіс, включає в себе різні засоби моніторингу, контролю та адміністрування, і різноманітні засоби розробки і тестування. Найкращим рішенням для проходження лабораторних курсів по розробці різного роду програмного забезпечення стане сервіс App Engine[14]. Він дозволяє створювати програмні продукти використовуючи наступні мови програмування: Python, Java, Node.js, Go, PHP, .NET та інші.

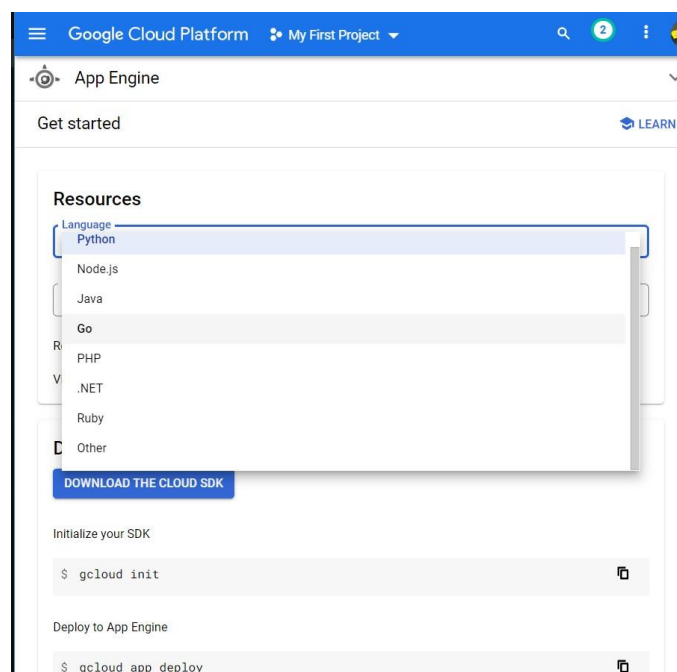


Рисунок 5 – Етап створення проекту в сервісі App Engine

Безсерверні обчислення на GCP, можна назвати своєрідною організацією PaaS. Головна задача якого залишається така ж сама як і в PaaS, а саме “функція як послуга”. Але відмінність полягає в тому що при виконанні кожного запиту функції, створюється окремий контейнер або віртуальна машина, на якій і виконується дана функція, а після виконання контейнер чи VM знищується.

ВИСНОВКИ

Виконуючи дипломну роботу, були проаналізовані лабораторні курси за весь період навчання, і були знайдені шляхи виконання лабораторних практикумів, не навантажуючи власну обчислювальну техніку, і окрім того, такий підхід дає змогу студентам, виконувати лабораторні та практичні роботи будь-де там де є досить швидке з'єднання з мережею.

Також було досліджено типи хмарних ресурсів, їх різноманіття, особливості, і сфери реалізації послуги.

Було обрано три лідируючі компанії, що постачають хмарні ресурси, та проаналізовано їхні рішення, можливості та пропозиції, за допомогою яких можливо виконувати лабораторні роботи. Кожне рішення в кожній з компаній відповідає типовим вимогам для виконання робіт. Різниця лише в тому, що кожний провайдер має різну цінову політику, і різне розміщення серверів, але в загальному вони між собою схожі.

Після аналізу можна зробити висновок, що Amazon Web Services і Microsoft Azure як лідери, мають приблизно однакову цінову політику, а GCP трішки дешевшу. Тому в якості економії фінансів рекомендується обрати рішення від компанії Google, адже якість і надання послуг відносно однакова, і питання стоїть лише в ціні.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Amazon – What is AWS? [Електронний ресурс]. - Режим доступу: URL https://aws.amazon.com/what-is-aws/?nc1=h_ls
2. Amazon – EC2 [Електронний ресурс]. - Режим доступу: URL https://aws.amazon.com/ec2/?did=ft_card&trk=ft_card
3. Amazon - AWS VPC [Електронний ресурс]. - Режим доступу: URL https://aws.amazon.com/vpc/features/?nc1=h_ls
4. Amazon – AWS Device farm [Електронний ресурс]. - Режим доступу: URL https://aws.amazon.com/device-farm/?did=ft_card&trk=ft_card
5. Amazon – AWS Amplify [Електронний ресурс]. - Режим доступу: URL <https://aws.amazon.com/amplify/hosting/>
6. Amazon – Lightsail [Електронний ресурс]. - Режим доступу: URL https://aws.amazon.com/lightsail/?did=ft_card&trk=ft_card
7. Microsoft cloud services – What is Azure? [Електронний ресурс]. - Режим доступу: URL <https://azure.microsoft.com/en-us/overview/what-is-azure/>
8. Microsoft cloud services – Azure for student [Електронний ресурс]. - Режим доступу: URL <https://azure.microsoft.com/en-us/free/students/>
9. Microsoft cloud services – Virtual Machines [Електронний ресурс]. - Режим доступу: URL <https://azure.microsoft.com/en-us/services/virtual-machines/>
10. Microsoft cloud services – Linux Virtual Machines [Електронний ресурс]. - Режим доступу: URL <https://azure.microsoft.com/en-us/services/virtual-machines/linux/>
11. Microsoft cloud services – Windows Virtual Machines [Електронний ресурс]. - Режим доступу: URL <https://azure.microsoft.com/en-us/services/virtual-machines/windows/>

12. Microsoft cloud services – App Service[Электронный ресурс]. - Режим доступа: URL <https://azure.microsoft.com/en-us/services/app-service/>
13. Microsoft cloud services – Static Web Application[Электронный ресурс]. - Режим доступа: URL <https://azure.microsoft.com/en-us/services/app-service/static/>
14. Google cloud – Why Google Cloud[Электронный ресурс]. - Режим доступа: URL <https://cloud.google.com/why-google-cloud/>
15. Google cloud – Compute Engine[Электронный ресурс]. - Режим доступа: URL <https://cloud.google.com/compute>
16. Google cloud – App Engine[Электронный ресурс]. - Режим доступа: URL <https://cloud.google.com/appengine>