

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Іфокомунікацій
Кафедра Інформаційно-мережної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Навчальна платформа для розвитку навичок роботи з хмарними сервісами

Виконав:	
студент	2 курсу, групи ІМІм-21-2
	Канівець В. І.
	(прізвище, ініціали)
Спеціальність	172 – Телекомунікації та радіотехніка
Тип програми	Освітньо-наукова
Керівник	доцент каф. ІМІ Скорик Ю. В.
	(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри

В.М. Безрук

2023 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____ / *В. І. Канівець* /

Керівник _____ / *Ю. В. Скорик* /

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)

Кафедра Інформаційно-мережної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 172 Телекомунікації та радіотехніка
(код і повна назва)

Тип програми Освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційно-мережна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ІМІ _____
(підпис)

“ _____ ” _____ 2023 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Канівцю Віталію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Навчальна платформа для розвитку навичок роботи з хмарними сервісами»

затверджена указом університету від « 27 » 04 _____ 20 23 р. №398Ст

2. Термін подання студентом роботи до екзаменаційної комісії «18» травня 2023р.

3. Вихідні дані до роботи дослідити навчальну платформу для вивчення сервісів GCP, Terraform, Jenkins, проаналізувати сучасні методи навчання роботи з хмарними сервісами, спроектувати навчальну платформу на базі Cloud Mentor.

4. Перелік питань, що потрібно опрацювати в роботі _____
Вступ

1. Місце хмарних технологій у сучасному світі

2. Аналіз сучасних методів навчання роботи з хмарними сервісами

3. Огляд технологій для розробки навчальної платформи, проектування рішення.

4. Проектування рішення удосконалення навчальної платформи CLOUD MENTOR

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускної кафедри). Слайди у форматі Power Point (назва, мета і задачі роботи, дослідження можливості та актуальності інтегрування в навчальну платформу Cloud Mentor хмарного провайдера GCP та подальше його використання для розвитку навичок з сервісами GCP, Отриманий результат роботи у вигляді працюючого Jenkins pipeline, висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання та літературних джерел	05.04.2023	вик.
2	Огляд існуючих рішень	10.04.2023	вик.
3	Проектування рішення удосконалення навчальної платформи	15.04.2023	вик.
4	Оформлення пояснювальної записки	25.04.2023	вик.
5	Оформлення презентаційного матеріалу	15.05.2023	вик.
6	Підготовка до захисту	18.05.2023	вик.

Дата видачі завдання 1.04 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доцент каф. ІМІ Скорик Ю.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка 59 с., 8 джерела, 40 рис., 2 додатки, 1 таблиця

Об'єкт роботи – навчальна платформа, яка надає можливості навчання і практичного впровадження хмарних сервісів.

Метою даної роботи є покращення навчальної платформи Cloud Mentor для розвитку навичок роботи з хмарними сервісами. Основною метою є збільшення конкурентоспроможності студентів та працівників на ринку праці, шляхом засвоєння необхідних навичок та автоматизації процесу навчання.

Методи розробки базуються на таких технологіях як Jenkins, Terraform, GCP.

У результаті роботи пропонується удосконалення навчальної платформи Cloud Mentor, шляхом розробки рішення для навчання сервісів у GCP. Удосконалення навчальної платформи Cloud Mentor дозволить вирішити проблеми, пов'язані з недостатньою експертизою в роботі з хмарними сервісами GCP, що збільшить ефективність навчання та розширить спектр навчання.

GCP, AWS, JENKINS, TERRAFORM, GITLAB, НАВЧАЛЬНА ПЛАТФОРМА, PIPELINE.

THE ABSTRACT

Explanatory slip 59 p., 8 sources, 40 fig., 2 app., 1 table

The object of the work is an educational platform that provides opportunities for training and practical implementation of cloud services.

The purpose of this work is to improve the Cloud Mentor educational platform for developing skills for working with cloud services. The main goal is to increase the competitiveness of students and workers in the labor market by acquiring the necessary skills and automating the learning process.

Development methods are based on such technologies as Jenkins, Terraform, GCP.

As a result of the work, it is proposed to improve the Cloud Mentor training platform by developing a solution for training services in GCP. The improvement of the Cloud Mentor training platform will allow solving the problems associated with insufficient expertise in working with GCP cloud services, which will increase the effectiveness of training and expand the range of training.

GCP, AWS, JENKINS, TERRAFORM, GITLAB, LEARNING PLATFORM, PIPELINE.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП.....	9
1 МІСЦЕ ХМАРНИХ ТЕХНОЛОГІЙ В СУЧАСНОМУ СВІТІ.....	10
1.1 Формування нелокальних обчислень та перспективи розвитку.....	10
1.3 Основні типи хмарних сервісів.....	15
1.4 Послуги, що здійснюються за допомогою хмарних сервісів	18
2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ НАВЧАННЯ РОБОТИ З ХМАРНИМИ СЕРВІСАМИ.....	21
2.1 Потік навчання онлайн	21
2.2 Udeму.....	22
2.2.1 Структура Udeму	22
2.2.2 Порівняння між Udeму та іншими платформами електронного навчання	23
3 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ НАВЧАЛЬНОЇ ПЛАТФОРМИ.....	26
3.1 Основні інструменти.....	26
3.1.1 Terraform	26
3.1.2 Jenkins.....	28
3.1.3 Gitlab.....	28
4 ПРОЕКТУВАННЯ РІШЕННЯ УДОСКОНАЛЕННЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ CLOUD MENTOR	29
ВИСНОВОК.....	42
ПЕРЕЛІК ПОСИЛАНЬ	43
ДОДАТОК А – СЛАЙДИ ПРЕЗЕНТАЦІЇ	44
ДОДАТОК Б – НАУКОВІ ПУБЛІКАЦІЇ.....	52

ПЕРЕЛІК СКОРОЧЕНЬ

GCP (Google Cloud Platform) – запропонований компанією Google набір хмарних служб, які виконуються на тій же самій інфраструктурі, яку Google використовує для своїх продуктів призначених для кінцевих споживачів;

AWS (Amazon Web Services) – є дочірньою компанією Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки;

PaaS (Platform as a Service) – модель, при якій розробники отримують доступ до платформи для створення, розгортання та управління веб-додатками без необхідності в управлінні інфраструктурою;

SaaS (Software as a Service) – модель, при якій програмне забезпечення надається користувачам через інтернет;

IaaS (Infrastructure as a Service) – модель, при якій користувачам надаються віртуальні обчислювальні ресурси через Інтернет.

ВСТУП

В сучасному світі, хмарні сервіси стали необхідністю для багатьох організацій та приватних користувачів. Їхня популярність зростає щодня завдяки високій мобільності та доступності, які вони надають. Проте, багато людей ще не мають достатньої експертизи в роботі з хмарними сервісами. Це може становити проблему для компаній, які потребують спеціалістів, що можуть ефективно працювати з цими сервісами.

Одним з можливих рішень цієї проблеми є створення навчальної платформи для розвитку навичок роботи з хмарними сервісами.

Актуальність дослідження навчальної програми для розвитку навичок роботи з хмарними сервісами полягає в необхідності засвоїти студентами та працівниками необхідні навички та стати більш конкурентоспроможними на ринку праці в цій області і зробити цей процес автоматизованим, щоб досягти максимальної ефективності.

У роботі розглядатиметься існуюча навчальна платформа Cloud Mentor для її покращення, розширивши її спектр використання. Розглядатимуться наступні задачі: теоретичні аспекти хмарних сервісів та онлайн-навчання, проведення аналізу існуючих платформ для навчання роботи з хмарними сервісами; аналіз технологій для навчальної платформи; удосконалення навчальної платформи Cloud Mentor, вдосконалення можливості навчання з використанням сервісів Google Cloud Platform (GCP).

1 МІСЦЕ ХМАРНИХ ТЕХНОЛОГІЙ В СУЧАСНОМУ СВІТІ

1.1 Формування нелокальних обчислень та перспективи розвитку

Інформація пронизує наш світ, і вільний доступ до неї став однією з визначальних подій у сучасній історії. Ми живемо в епоху хмарних технологій. Хмарні обчислення – одна з найбільш інноваційних технологій сучасності. Вони пропонують швидкість, меншу вартість та кращий доступ до ресурсів у глобальному масштабі. Соціальні переваги хмарних технологій все ще неочевидні, оскільки ми продовжуємо досліджувати, як розвивається наша нова технологічна ера.

Нелокальна технологія зберігання та обробки даних в 1950-х роках розвивалася у військовій сфері і була розроблена для підключення комп'ютерних терміналів через внутрішні мережі. Проте швидкого розвитку вона набула лише коли потрапила в наукове співтовариство. Це було важливим питанням у той час, коли обчислення коштували кілька мільйонів доларів, і доступ багатьох людей до цієї технології став необхідністю. Сам термін хмарні обчислення був введений у 1996 році у внутрішньому документі компанії Compaq. Термін хмара спочатку був пов'язаний з концепцією розподілених обчислень, яка стала загальноприйнятою у створеній Apple компанії General Magic на початку 1990-х років. В 1969 році Боб Тейлор та Ларрі Робертс розробили ARPANET (Мережі Агентства перспективних дослідницьких проєктів) і, врешті-решт, стали попередниками того, що ми називаємо Інтернетом [1].

До середини 1990-х для позначення цієї нової цифрової сфери використовувався термін "хмара". Незабаром Google і Microsoft почали гонку з метою збільшення своєї частки ринку в цьому віртуальному середовищі [2].

PaaS (Platform-as-a-Service), SaaS та IaaS (Infrastructure-as-a-Service) стали абсолютно новою галуззю зі своїм відгалуженням - підприємством з кібербезпеки. Усі мали доступ до хмари - для розваг, охорони здоров'я, фінансів та управління і золота лихоманка приєднання до цього нового сектору йшла прискореними темпами. Хмара створювала культурний зсув, якого ще не було в історії людства: бар'єри знань були усунені, а доступ до інформації став більш звичним і простим. Невеликі стартапи змінювали світ, а творчість та інновації стали сферою особистості [1].

25 серпня 2006 року Amazon Web Services запустила Elastic Compute Cloud (EC2), що дозволяє людям орендувати віртуальні комп'ютери та використовувати власні програми та програми в Інтернеті; за цим швидко пішли служби Google Docs. Потім Amazon створила просту службу зберігання даних (S3) [3].

Наведемо декілька прогнозів того, що ми можемо очікувати у майбутньому хмарних обчислень:

- хмарні обчислення стануть ще більш помітними у найближчі роки, завдяки швидкому, постійному зростанню основних глобальних хмарних центрів обробки даних;
- більше 50% усіх ІТ послуг мігрують у хмари протягом наступних 5 - 10 років;
- хмарні технології будуть широко застосовуватися в цілому на ринках, що розвиваються, це буде особливо відчутним в Азії, де вже існує тенденція існувати на межі новітніх технологій;
- дані для компаній та особистого користування будуть доступні скрізь у стандартизованих форматах, що дозволить нам легко взаємодіяти між собою на ще вищому рівні;
- безпека та надійність хмарних обчислень будуть продовжувати розвиватися, забезпечуючи ще більший захист даних;

- майбутнє хмарних обчислень далеко не певне, швидкі темпи зміни технологій за останні 5 років роблять наступні 5 майже неможливими для прогнозування. Однак слід сказати, що зрештою хмара зростає в геометричній прогресії і буде продовжувати це робити ще деякий час.

1.2 Проблеми хмарних технологій

За останні кілька років хмарні технології переросли з перспективної бізнес-концепції в один із найбільш швидкозростаючих сегментів ІТ-галузі. Зараз компанії все частіше усвідомлюють, що використовуючи хмару, вони можуть отримати швидкий доступ до найкращих у своїй галузі бізнес-додатків або різко збільшити свої ресурси, і все це має незначну вартість. Але в міру того, як дедалі більше інформації про приватних осіб та компаній розміщується в хмарі, починає зростати занепокоєння з приводу того, наскільки безпечним є це середовище.

Одні стверджують, що дані клієнтів є більш безпечними при внутрішньому управлінні, тоді як інші стверджують, що постачальники хмарних послуг зацікавлені підтримувати довіру і використовують більш високий рівень безпеки. Однак не слід забувати, що у хмарі дані користувачів будуть розподілені між окремими комп'ютерами, незалежно від того, де в кінцевому підсумку зберігається базовий файл. Загрози, такі як злом веб-сайтів та вірусні атаки, є найбільшими проблемами безпеки даних хмарних технологій, тому що хакери можуть вторгнутися практично на будь-який сервер [1].

Недостатність ресурсів та досвіду одна з ключових проблем. Відповідно до звіту RightScale, майже 75% респондентів відзначили це як проблему, тоді як 23% сказали, що це серйозна проблема. Незважаючи на те, що багато працівників ІТ вдосконалюють свої знання, роботодавцям складно знайти співробітників з необхідним досвідом.

Деякі організації також розраховують на рішення проблеми переходу до хмарних обчислень шляхом працевлаштування більшої кількості працівників, які мають сертифікати або навички в галузі хмарних технологій. Фахівці також пропонують забезпечити навчання нинішніх працівників, щоб зробити їх більш продуктивними та здатними використовувати найсучасніші технології [1].

ІТ працівники не мають повного контролю над обладнанням, доставкою інфраструктури та її функціонуванням у хмарному світі. Щоб викоринити різні невизначеності та труднощі при переході до хмари, ІТ спеціалісти повинні впровадити звичайні процедури контролю та управління. Основні ІТ компанії відіграють все більшу роль у посередництві, постачанні та контролі над хмарними послугами.

Відповідно до звіту RightScale, організації втрачають майже 30% грошей, які вони інвестують у хмару. Компанії допускають кілька помилок, які можуть збільшити їх витрати. Іноді ІТ-спеціалісти, такі як розробники, вмикають хмарний сервіс, який передбачається використовувати деякий час, і забувають його знову вимкнути. А деяким компаніям заважають приховані пакети з витратами на хмару, які не використовуються. Допомогти організаціям можуть кілька технічних рішень: автоматизація, концепції для управління хмарними витратами, безсерверні послуги, функції автоматичного масштабування та численні інструменти управління, що надаються постачальниками хмарних послуг.

В наш час більшість компаній працюють не лише з однією хмарою. Згідно зі звітом RightScale, майже 84% компаній дотримуються мультихмарної стратегії, а 58% вже мають свою гібридну тактику хмар, поєднану із використанням державних та приватних хмар. Щоб перемогти цей виклик, професіонали використовують навчання персоналу, активне управління відносинами з продавцями та проведення досліджень [1].

На відміну від традиційної обчислювальної моделі, хмарні обчислення використовують технологію віртуальних обчислень, особисті дані користувачів можуть розкидатися по різних віртуальних центрах обробки даних, а не залишатися в одному і тому ж фізичному місці, навіть передаватися через національні кордони. В цей час захист конфіденційності даних зіткнеться з суперечками різних правових систем. З іншого боку, користувачі можуть отримувати приховану інформацію під час доступу до послуг хмарних обчислень.

Сервери в хмарі мають такі ж проблеми, як і власні сервери-резиденти користувачів. Хмарні сервери також чутливі до простоїв та уповільнень, різниця полягає в тому, що користувачі мають більшу залежність від постачальника хмарних послуг. Існує велика різниця в моделях обслуговування, і коли клієнт обирає конкретного постачальника, то можете потрапити в залежність і тим самим створити потенційний ризик для бізнесу.

Кожного разу, коли організація передає дані зі свого внутрішнього сховища в хмару, вона повинна притримуватись законів та правил галузі, які регулюються Загальним регламентом захисту даних (GDPR). Багато організацій потребують спеціаліста із захисту даних, який може забезпечити конфіденційність інформації відповідно до вимог закону.

Слід бути впевненим, що дані, які ви вводите в хмару, ніколи не будуть втрачені, навіть якщо ваш постачальник хмарних обчислень буде ліквідований або його поглине більша компанія.

Нарешті, компанії, особливо ті, що мають гібридне хмарне середовище, повідомляють про проблеми, пов'язані з наявністю їхніх локальних додатків та інструментів та загальнодоступної хмари для спільної роботи. Згідно з опитуванням, 62% респондентів назвали інтеграцію застарілих систем найбільшим викликом у мультихмарі [1].

Не дивлячись на існуючі проблеми, значення хмарних технологій зростає в геометричній прогресії. За прогнозами, до 2022 року 90% організацій використовуватимуть хмарні сервіси.

1.3 Основні типи хмарних сервісів

Існує три основні сервісні моделі хмарних обчислень - Інфраструктура як послуга (IaaS), Платформа як послуга (PaaS) та Програмне забезпечення як послуга (SaaS). Є чіткі відмінності між цими сервісами і тим, що вони можуть запропонувати користувачам з точки зору зберігання та об'єднання ресурсів. Ці сервіси ще називають шарами хмари і вони також можуть взаємодіяти між собою, формуючи одну всеосяжну модель хмарних обчислень (рис. 1.1):



Рисунок 1.1 – Піраміда хмарних сервісів

Інфраструктура як послуга (IaaS). Як основа піраміди хмарних обчислень, IaaS є найбільш повним та гнучким видом хмарних сервісів. Це найпоширеніша модель обслуговування хмарних технологій, оскільки вона пропонує фундаментальну інфраструктуру віртуальних серверів, мереж, операційних систем

та накопичувачів даних.

По суті IaaS забезпечує повністю віртуалізовану обчислювальну інфраструктуру, яка забезпечується та управляється через Інтернет. Постачальник послуг IaaS управляє фізичною ланкою інфраструктури (сервери, простір для зберігання даних тощо), яка розташована в центрі обробки даних, але дозволяє клієнтам повністю налаштувати ці віртуалізовані ресурси відповідно до своїх конкретних потреб. За допомогою IaaS клієнт може купувати, встановлювати, налаштовувати та керувати будь-яким програмним забезпеченням, включаючи такі речі, як операційні системи, проміжне програмне забезпечення, програми, бізнес-аналітика та засоби розробки [3].

Компанії платять лише за ту інфраструктуру, яку вони використовують, що дозволяє їм масштабувати свої обчислювальні потреби без необхідності нарощувати додаткові потужності. IaaS виключає капітальні витрати на створення власної інфраструктури. Це чудовий варіант для невеликих компаній та стартапів, які не мають ресурсів для придбання обладнання та програмного забезпечення, необхідного для створення власної мережі. Повсякденний тягар управління обчислювальною інфраструктурою також беруть на себе ІТ-відділи. Оскільки постачальник послуг IaaS постійно оновлює свою систему за допомогою найновішого програмного забезпечення, стає простіше запускати нові програми. IaaS пропонує найновіші засоби захисту та, як правило, пропонує такі послуги, як аварійне відновлення.

Приклади IaaS: Microsoft Azure, Amazon Web Services (AWS), Google Compute Engine (GCE) [3].

РaaS (Платформа як послуга). РaaS розташований трохи вище у піраміді хмарних обчислень. У той час як IaaS постачає всі інструменти, доступні через хмару, і залишає замовникам можливість створювати те, що відповідає їхнім потребам, РaaS трохи більш спеціалізований. Замість чистої інфраструктури, РaaS забезпечує структуру, необхідну для побудови, тестування, розгортання,

управління та оновлення програмних продуктів.

Сервіс використовує ту саму базову інфраструктуру, що і IaaS. PaaS надзвичайно корисний для будь-якої компанії, яка розробляє програмне забезпечення та веб-додатки. Постачальники хмарних обчислень розгортають інфраструктуру та програмне забезпечення, але підприємства можуть озробляти та запускати і власні програми

Багато інструментів, необхідних для розробки для платформ (комп'ютери, мобільні пристрої, браузері тощо), є досить дорогими. Використовуючи PaaS, клієнти можуть отримати доступ до інструментів розробки, коли вони потребують, без необхідності їх купувати. Оскільки платформа доступна через Інтернет, усі команди віддалених IT-працівників можуть отримати доступ до тих самих ресурсів, щоб пришвидшити розробку продукту. Рішення PaaS є масштабованими та ідеально підходить для бізнес-середовищ, де кілька розробників працюють над одним проектом. Це також зручно для ситуацій, коли потрібно використовувати існуюче джерело даних (наприклад, інструмент CRM) [4].

Приклади PaaS: AWS Elastic Beanstalk, Apache Stratos, Google App Engine, Microsoft Azure.

Програмне забезпечення як послуга (SaaS). Для більшості людей SaaS є найбільш звичною формою хмарних обчислень. SaaS - це повністю розроблене програмне рішення, готове до придбання та використання через Інтернет і розташоване на вершині піраміди хмарних технологій.

Постачальник SaaS надає програму «в оренду», управляє інфраструктурою, операційними системами, проміжним програмним забезпеченням та даними, гарантуючи доступність програмного забезпечення, коли і де це потрібно клієнтам. Багато додатків SaaS працюють безпосередньо через веб-браузери, усуваючи необхідність завантажувати чи встановлювати додаткові програми. Це значно зменшує проблеми з управлінням програмним забезпеченням і дозволяє компаніям оптимізувати свою діяльність за допомогою гібридних та багатохмарних рішень.

Додатки SaaS дозволяють компаніям швидко розпочинати роботу, а також швидко масштабувати операції. Не потрібно купувати або розгортати апаратне та програмне забезпечення, яке використовується для надання бізнес-послуг. Навіть складні додатки, такі як програми управління взаємовідносинами з клієнтами (CRM) або програми планування корпоративних ресурсів (ERP), можуть бути легко доступні найменшим організаціям [1].

Приклади SaaS: Microsoft Office 365, Salesforce, Google Apps.

Для наочності, узагальнимо сервіси архітектури «хмара» в одну схему, на якій наведено класифікацію сервісів за типом послуг (рис 1.2):

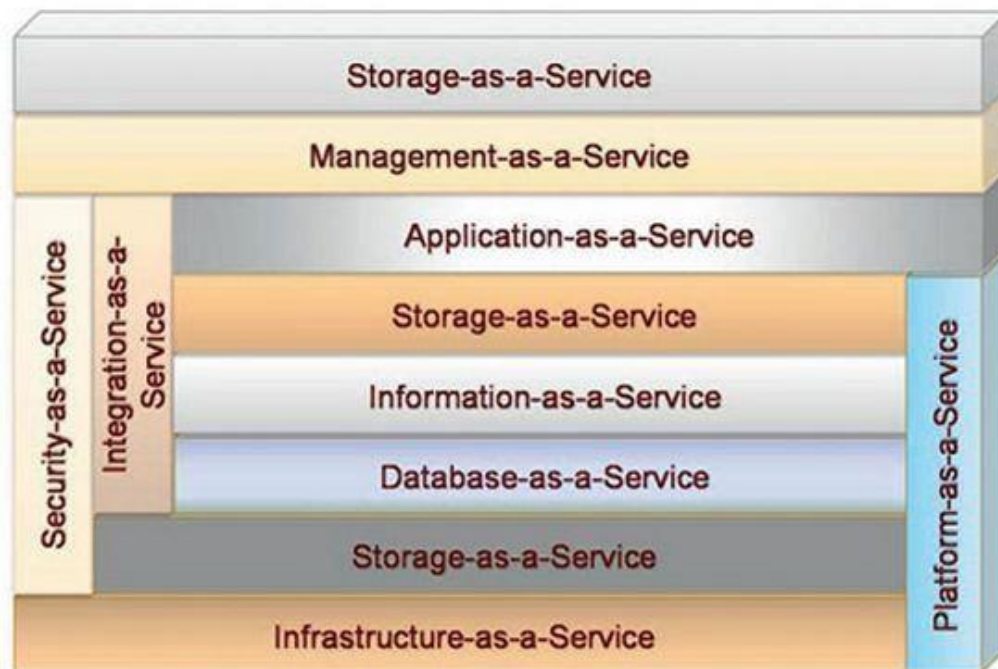


Рисунок 1.2 – Архітектура хмарних сервісів

1.4 Послуги, що здійснюються за допомогою хмарних сервісів

Функція як послуга (FaaS). Часто відома як безсерверні обчислення, FaaS дозволяє клієнтам швидко реагувати на зміни без необхідності розподіляти ресурси задалегідь. Хмарний провайдер обробляє інфраструктуру, дозволяючи клієнту

зосередитися на розгортанні коду програми. Функції масштабуються автоматично, що робить їх ідеально пристосованими для динамічних навантажень. Клієнти платять лише за ресурси, якими вони користуються, що робить FaaS найоптимальнішою формою хмарних обчислень. Більшість програм FaaS досить прості і можуть бути розгорнуті дуже швидко. Клієнту хмари просто потрібно завантажити відповідний код функції та повідомити платформі, як ресурси необхідні під час її виконання. Нові види функцій можна масштабувати за запитом, і коли функція не використовується, вона не споживає жодних ресурсів. Основним недоліком FaaS є час виконання.

Оскільки функції повинні надавати ресурси при кожному їх запуску, можуть бути незначні відставання в продуктивності, якщо програма вимагає великої обчислювальної потужності або виконується під час пікового навантаження. Більшість служб FaaS доступні через великих хмарних провайдерів, таких як AWS та Azure, що може призвести до блокування постачальника [4].

Storage as a Service (зберігання як сервіс) – це бізнес-модель, при якій велика компанія здає в оренду меншій компанії або окремій особі простір в своїй інфраструктурі сховища. На підприємстві постачальники SaaS орієнтуються на додаткові програми зберігання даних, просуваючи SaaS як зручний спосіб управління резервними копіями. Ключова перевага SaaS полягає в економії витрат на персонал, апаратному забезпеченні та фізичному просторі зберігання. Наприклад, замість того, щоб підтримувати велику кількість копій файлів і організувати їх зберігання за межами сайту, мережевий адміністратор, який використовує SaaS для резервного копіювання, може вказати, для яких даних в мережі слід створювати резервні копії та як часто їх слід створювати.

Його компанія підпише угоду про рівень обслуговування (SLA), згідно з якою постачальник SaaS погоджується надати простір для зберігання на основі ціни за гігабайт, що зберігається, і ціни за передачу даних, і дані компанії будуть автоматично передаватися в зазначений час через власну глобальну мережу

(WAN) провайдера сховища або Інтернет. Якщо дані компанії коли-небудь пошкодяться або загубляться, адміністратор мережі може зв'язатися з постачальником SaaS та запросити копію даних.

DBaaS (також відомий як сервіс керування баз даних) - це одна з найбільш затребуваних технологій в галузі управління інформаційними ресурсами, хмарний підхід до зберігання та управління структурованими даними. DBaaS дозволяє користувачам отримувати доступ до хмарної системи баз даних і користуватися нею без придбання та налаштування власного обладнання, встановлення власного програмного забезпечення для баз даних чи управління ними. Суть концепції DBaaS в тому, що користувачеві не потрібно встановлювати і підтримувати базу даних, йому досить зробити запит і отримати по ньому базу даних. Хмарний провайдер піклується про все - від періодичного оновлення і резервного копіювання до забезпечення того, щоб система баз даних залишалася доступною та безпечною цілодобово.

Ринок DBaaS та хмарних баз даних є одним із найбільш швидкозростаючих ринків SaaS, який, як очікується, зросте до 320 млрд. доларів США до 2025 року. Постачальники баз даних та сховищ даних приєдналися до відомих хмарних постачальників, пропонуючи своє програмне забезпечення, що дозволяє клієнтам використовувати багато переваг хмарних обчислень для зберігання даних, пошуку та доступу до своїх додатків. Користувач отримує з інформаційних ресурсів необхідну інформацію, зібрану згідно з параметрами свого запиту, і може працювати з нею як зі звичайною базою даних: зберігати, редагувати, відправляти іншим користувачам, об'єднувати з іншими базами і інше. Як приклад можна назвати Amazon Web Services [1].

2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ НАВЧАННЯ РОБОТІ З ХМАРНИМИ СЕРВІСАМИ

2.1 Потік навчання онлайн

Щоб отримати можливість на онлайн-навчання в різних платформах, користувач має пройти наведені нижче кроки (рис. 2.1):

Крок 1: Створення облікового запису.

Крок 2. Додайте спосіб оплати.

Крок 3: Пошук у категоріях і темах.

Крок 4: Читання відгуків.

Крок 5. Виберіть курс або декілька.

Крок 6: Купуйте.

Крок 7: Розпочніть курс або кілька курсів.

Крок 8: Завершити курс.

Крок 9: Скласти іспит і отримати диплом.

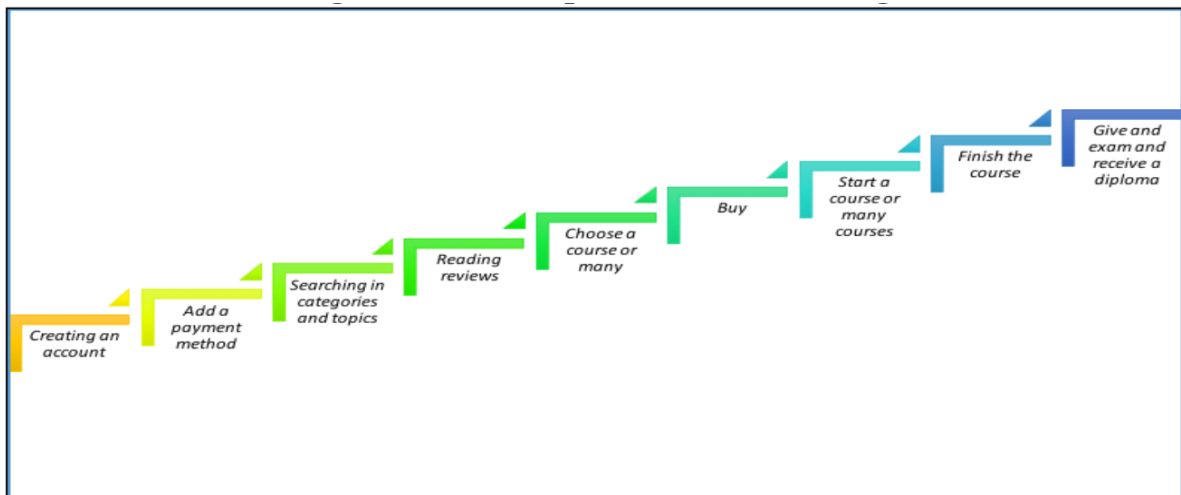


Рисунок 2.1 – Кілька кроків онлайн-навчання

На різних платформах різний функціонал, обсяг курсів для навчання, цінова політика та якість реалізації платформи. Далі розглянемо одні з найпопулярніших навчальних платформ на ринку відштовхуючись від платформи Udemy.

2.2 Udemy

2.2.1 Структура Udemy

Udemy є однією з найважливіших онлайн-платформ для навчання у світі. Нижче наведені деякі статистичні дані Udemy (www.udemy.com):

- 20 мільйонів студентів;
- 65000 онлайн-курсів;
- 14 категорій;
- понад 142 теми.

Найважливіші категорії Udemy:

- розвиток,
- бізнес,
- ІТ та програмне забезпечення,
- офісна продуктивність,
- особистісний розвиток,
- дизайн,
- маркетинг та інше.

2.2.2 Порівняння між Udemy та іншими платформами електронного навчання

Таблиця 2.1 Порівняння між Udemy та іншими онлайн-платформами

Платформа	Ціна	Якість	Охоплення
Udemy	ціни/11,99/курс	Гарна якість	Велике
Coursera	середні ціни/49/місяць	Дуже гарна якість	Середнє
Lynda	невеликі ціни/19,99/місяць	Гарна якість	Середнє
Udacity	великі ціни/499/місяць	Дуже гарна якість	Середнє
Khan Academy	Безкоштовно	Добре	Велике
Codecademy	середні ціни/199/курс	Добре	Велике
Blackboard	великі ціни/8500/програма	Гарна якість	Дві програми
Iversity	великі ціни/399	Добре	Маленьке

Як ми бачимо в таблиці 2.1 і на рисунку, Udemy має найнижчу ціну серед курсів, хорошу якість і великі території. Udemy має чотирнадцять категорій із понад 140 темами [5].

Практика Coursera середня ціна, 49 USD/міс. Якість курси дуже хороші, оскільки на платформі працюють кращі університетські художники. Сфера курсів середня, від IT до бізнесу. Лінда також має невеликі ціни, від 19,99 USD/місяць, гарну якість і середню площу категорій.

Порівняно з Udemy та іншими платформами, Udacity має високі ціни,

починаючи з 499 доларів США за курс і 199 доларів США за місяць. Курси Udacity дуже хороші із середньою площею.

Курси Khan Academy «Безкоштовні. Для всіх. Назавжди». Курси дуже хороші, і вони перекладені більш ніж на 36 мов. Khan Academy велика територія і курси розроблені спеціально для школярів і студентів. Платформа базується на волонтерстві та пожертвах.

Codecademy – це платформа із середньою ціною, 199/курс, гарною якістю та невеликою кількістю курсів, лише для навчання програмуванню.

Віос є найдорожчою платформою, 8500 доларів США за одну програму. У них лише дві програми, для веб-розробників і для дизайнерів. Якість програм дуже хороша. Останнім варіантом є Iversity з високою ціною, близько 399 євро за курс і невеликою площею [5].

Що стосується кількості курсів, UdeMy є платформою з найбільшою кількістю онлайн-курсів. Coursera має лише 2000 курсів, а Lynda – 6417 (рис. 2.2)

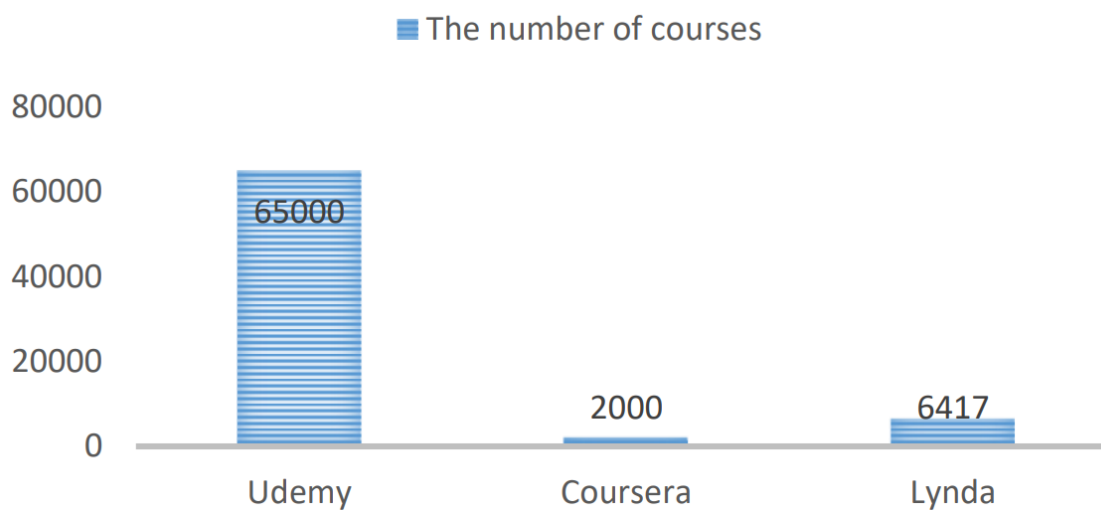


Рисунок 2.2 – Кількість ресурсів

Найбільше студентів у Codecademy, а точніше 45 мільйони, як і всі інші платформи разом.

Udacity має 160 000 учнів, Coursera 25 мільйонів і Udemy 20 мільйонів (рис. 2.3).

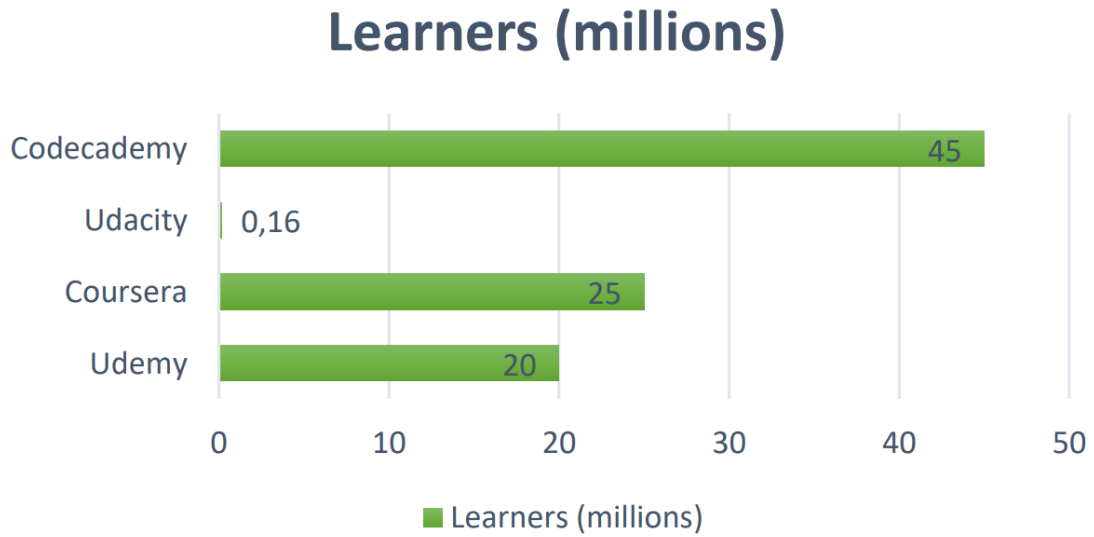


Рисунок 2.3 – Кількість користувачів

3 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ НАВЧАЛЬНОЇ ПЛАТФОРМИ

3.1 Основні інструменти

Оскільки можливість навчатися використовуючи сервіси Google Cloud Platform буде відбуватися шляхом удосконалення платформи Cloud Mentor, то базуватися будемо вже на основних наявних там інструментах, а саме:

- Terraform;
- Jenkins;
- Gitlab.

3.1.1 Terraform

HashiCorp Terraform – це інструмент кодування інфраструктури. Це дозволяє визначати хмару та локальну мережу у зрозумілих конфігураційних файлах, з яких можна створювати різні версії конфігурацій, використовувати їх повторно та ділитися ними. Використовуючи одні й ті самі файли, узгоджений робочий процес можна використовувати для керування всією інфраструктурою протягом її життєвого циклу. Terraform розроблено для керування компонентами низького рівня, такими як обчислення, сховище та мережеві ресурси [6].

За допомогою Terraform можна створювати та керувати ресурсами на багатьох хмарних платформах та інших сервісах за допомогою вбудованих інтерфейсів прикладного програмування (API). Провайдери дозволяють Terraform працювати практично з будь-якою платформою або сервісом із доступним API.

Серед таких загальнодоступних провайдерів у реєстрі Terraform є такі як Amazon Web Services (AWS), Google Cloud Platform (GCP), Azure, Helm, GitHub,

Kubernetes, Splunk, DataDog та інші. Основний процес роботи у Terraform поділено на три етапи.

Write: визначення ресурсів, які можуть бути у різних хмарних постачальників і служб. Наприклад, можна створити конфігурацію, яка виконуватиме розгортання програми на декількох віртуальних машинах у мережі віртуальної приватної хмари (VPC) з групами безпеки, підмережами, базою даних та балансувальником навантаження [6].

Plan: Terraform створює план виконання конфігурації, що повністю описує інфраструктуру, яку він може створити, оновити або видалити на основі файлів, як описують наявну інфраструктуру та наявної конфігурації.

Apply: після підтвердження в консолі Terraform виконує запропоновані на минулому етапі операції в правильному порядку, дотримуючись залежностей від ресурсів, прописаних в конфігурації. Наприклад, якщо потрібно оновити властивості VPC і змінити кількість віртуальних машин у тому самому VPC, Terraform спочатку відтворить VPC перед тим як виконувати масштабуванням віртуальних машин.

Провайдери визначають окремі одиниці інфраструктури, такі як обчислювальні екземпляри, їх тип або групи безпеки, як ресурси. Ми маємо змогу складати ресурси різних постачальників у багаторазові конфігурації, які називаються модулями, і керувати цими модулями за допомогою узгодженої мови та робочого процесу. Мова конфігурації Terraform є декларативною. Це означає опис бажаного кінцевого стану інфраструктури, на відміну відпроцедурних мов програмування, які вимагають детальних покрокових інструкцій для виконання поставлених завдань. Постачальники Terraform автоматично обчислюють всі наявні залежності між ресурсами, для їх створення або знищення у правильному порядку.

Terraform відстежує реальну інфраструктуру за допомогою файлу стану, який є джерелом інформації про поточний стан середовища. Terraform використовує цей файл, щоб визначити які зміни потрібно внести у вашу інфраструктуру, щоб вона

відповідала бажаній конфігурації [6].

Таким чином цей інструмент допомагає створювати потрібну інфраструктуру на хмарі у користувача, та після виконання завдання знищити цю інфраструктуру мінімізуючи витрати на ресурси.

3.1.2 Jenkins

Jenkins – це безкоштовний сервер автоматизації з відкритим кодом. Це допомагає автоматизувати частини розробки програмного забезпечення, пов'язані зі створенням, тестуванням та розгортанням, полегшуючи постійну інтеграцію та безперервну доставку [7].

Основним поняттям в Jenkins є збірки (Builds), що є процесом створення артефакту та доставки його у потрібне місце з можливістю розвертання на сервері.

У конвеєрі Jenkins кожна робота або подія має свого роду залежність як мінімум від одного або декількох подій.

У випадку Cloud Mentor, Jenkins грає одну з головних ролей. За його допомогою відбуваються зв'язка всіх інструментів між собою, щоб реалізувати процес автоматичного підняття інфраструктури відповідно до завдання, провести тестування правильності виконаного завдання та запустити автоматичне знищення піднятих ресурсів на хмарі.

3.1.3 Gitlab

Для керування версіями файлів та можливістю в подальшому налаштувати автоматичне стягування з репозиторію коду використовується Gitlab.

Gitlab – є популярним веб-сервісом для роботи з командою над будь-яким проектом зв'язаним з написанням програмного коду [8].

4 ПРОЕКТУВАННЯ РІШЕННЯ УДОСКОНАЛЕННЯ НАВЧАЛЬНОЇ ПЛАТФОРМИ CLOUD MENTOR

Для удосконалення навчальної платформи шляхом додавання платформи для вивчення сервісів GCP ми використовуватиме terraform в якості інструменту який буде піднімати на хмарних сервісах ресурси. Ці ресурси в свою чергу будуть покривати окреме завдання, яке треба виконати. В завданні детально описано інфраструктура, що було створено та що потрібно зробити для успішного завершення завдання.

Наприклад завдання знайти вразливість у вказаній архітектурі, яка буде розгорнута за допомогою terraform.

Користувач має підготувати свого хмарного провайдера для того, щоб платформа мала змогу піднімати на ньому ресурси для виконання завдання та перевірки. Після цього на Jenkins користувач може вибрати потрібне йому завдання та запустити його.

Вибравши завдання на його хмарі піднімуться відповідні ресурси з неправильною конфігурацією і користувачу потрібно буде вирішити всі недоліки шляхом виправлення тих ресурсів відповідно до завдання (рис. 4.1).

Як тільки користувач буде впевнений, що його завдання виконане, він може запустити перевірку на Jenkins та переконатися вірно чи ні він виконав завдання. Якщо завдання виконано не вірно, Jenkins надасть відповідний результат, що не так, та що потрібно виправляти і попросить виконати знову.

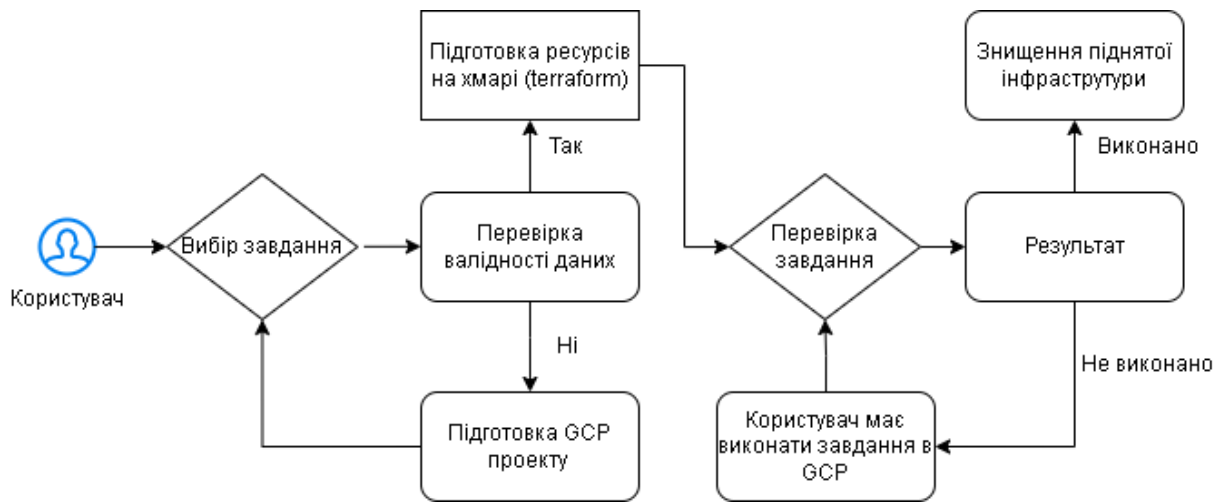


Рисунок 4.1 – Схема проходження завдання

Користувач може продовжити і повторювати перевірку доки не виконає завдання. Коли все буде вірно всі ресурси автоматично будуть знищені з хмари, що буде гарантувати мінімальні грошові затрати. Кількість створених завдань може постійно поповнюватися для покриття знань по всіх сервісах хмарних провайдерів.

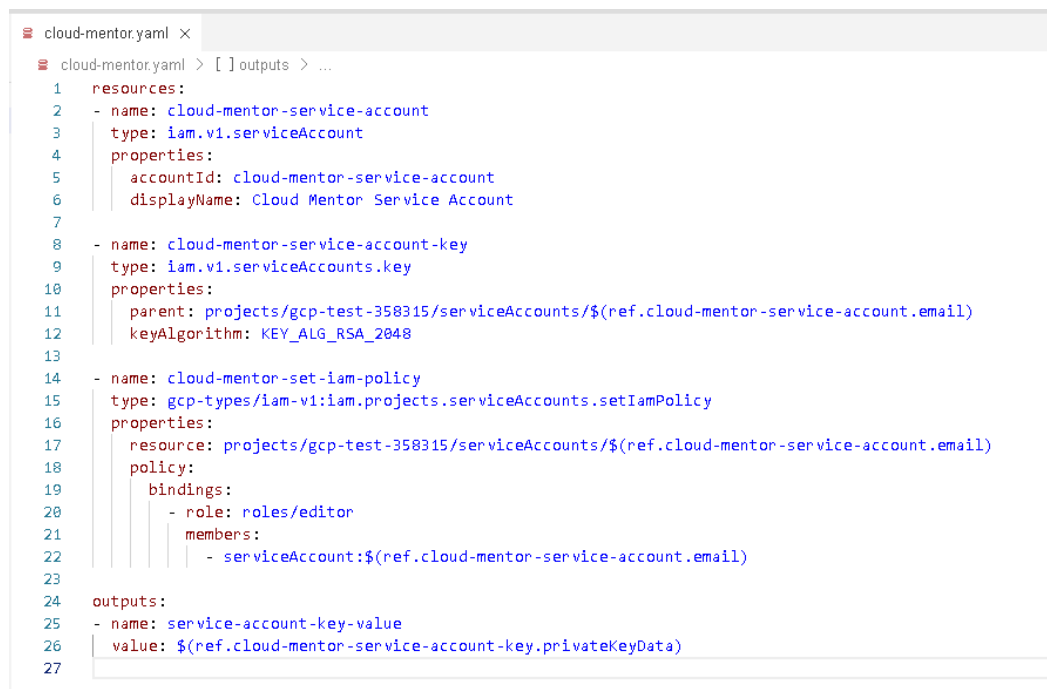
5 РЕАЛІЗАЦІЯ РІШЕННЯ

5.1 Підготовка хмарного провайдера

Для того щоб підготувати Google Cloud Platform для подальшого використання в платформі Cloud Mentor, потрібно створити сервіс акаунт за допомогою якого в хмарі будуть створюватися та видалятися ресурси.

На рис. 5.1 показаний конфігураційний файл для сервісу GCP Cloud Deployment Manager. Якщо його запустити на хмарі, то буде створено сервіс акаунт `cloud-mentor-service-account` з роллю `roles/editor`, яка дозволить створювати та видаляти більшість ресурсів в GCP. Також буде створено ключ `cloud-mentor-service-account-key`, який потрібен Cloud Mentor для взаємодії з хмарою.

Варто зауважити, що назва проекту `gcp-test-358315` є змінною і треба змінювати її в залежності до якого проекту ви хочете приєднати Cloud Mentor.



```
cloud-mentor.yaml x
cloud-mentor.yaml > [ ] outputs > ...
1 resources:
2   - name: cloud-mentor-service-account
3     type: iam.v1.serviceAccount
4     properties:
5       accountId: cloud-mentor-service-account
6       displayName: Cloud Mentor Service Account
7
8   - name: cloud-mentor-service-account-key
9     type: iam.v1.serviceAccounts.key
10    properties:
11      parent: projects/gcp-test-358315/serviceAccounts/$(ref.cloud-mentor-service-account.email)
12      keyAlgorithm: KEY_ALG_RSA_2048
13
14   - name: cloud-mentor-set-iam-policy
15     type: gcp-types/iam-v1:iam.projects.serviceAccounts.setIamPolicy
16     properties:
17       resource: projects/gcp-test-358315/serviceAccounts/$(ref.cloud-mentor-service-account.email)
18       policy:
19         bindings:
20           - role: roles/editor
21             members:
22               - serviceAccount:$(ref.cloud-mentor-service-account.email)
23
24 outputs:
25   - name: service-account-key-value
26     value: $(ref.cloud-mentor-service-account-key.privateKeyData)
27
```

Рисунок 5.1 – Cloud deployment конфігураційний файл

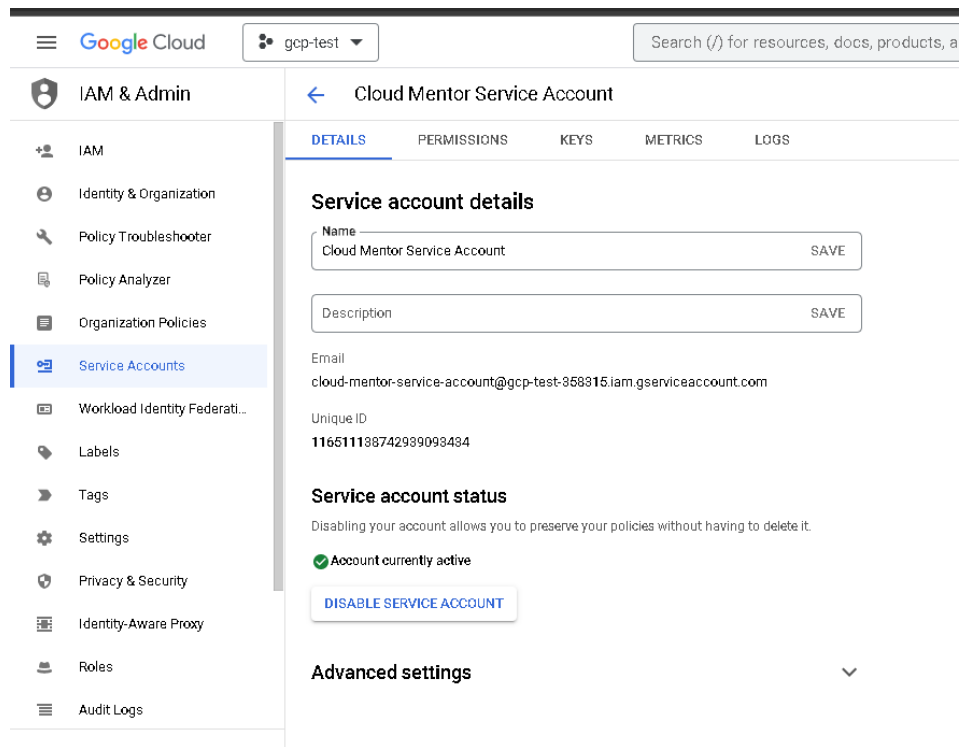


Рисунок 5.3 – Створений сервіс акаунт

5.2 Перевірка роботи

Для подальшого тестування було створено просте завдання, де потрібно змінити Cloud bucket таким чином, щоб створити статичний сайт. На terraform було написано код, який буду піднімати цей bucket не до кінця налаштованими конфігураціями і він же буде верифікувати зміни в хмарі.

В Jenkins ми налаштовуємо pipeline, який буде автоматично піднімати інфраструктуру в хмарі використовуючи вище вказаний ключ та terraform файли (рис. 5.4).

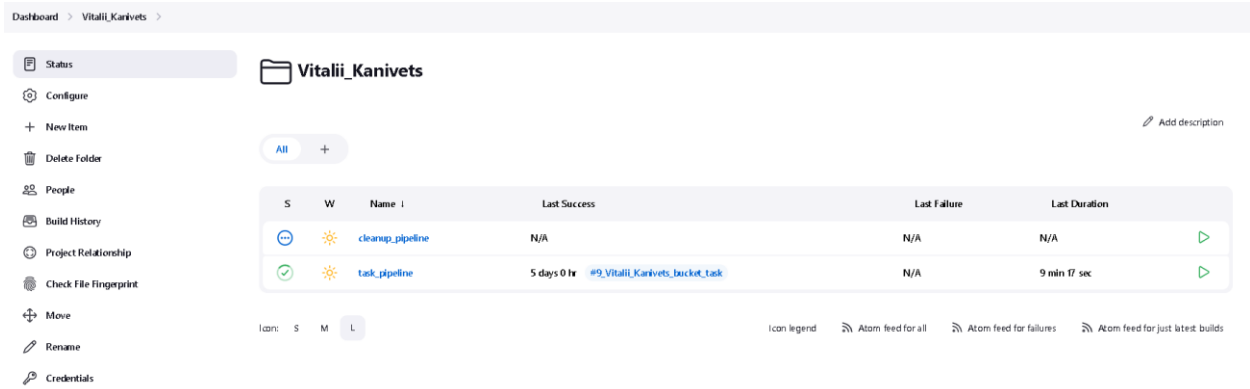


Рисунок 5.4 – Налаштований pipeline

Надалі ми запускаємо pipeline та вибираємо завдання, яке будемо проходити (наразі реалізоване одне тестове завдання). На рис. 5.5 видно всі етапи pipeline, а саме стягування коду з Gitlab. Потім ініціалізація, де пропонується вибрати завдання з доступних. Як тільки користувач вибере завдання, Jenkins запустить terraform для підняття інфраструктури в хмарі.

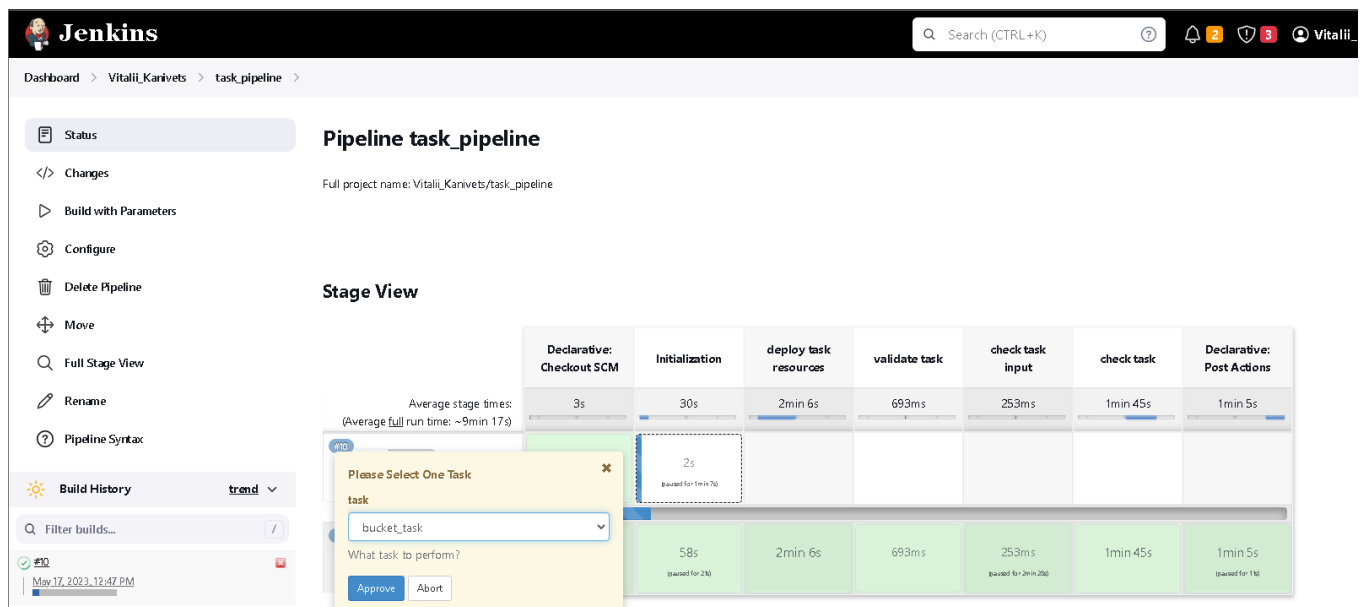


Рисунок 5.5 – Запуск pipeline

Як всі ресурси будуть підняті ми можемо виконувати завдання та після виконання тиснути на перевірку (рис. 5.6).

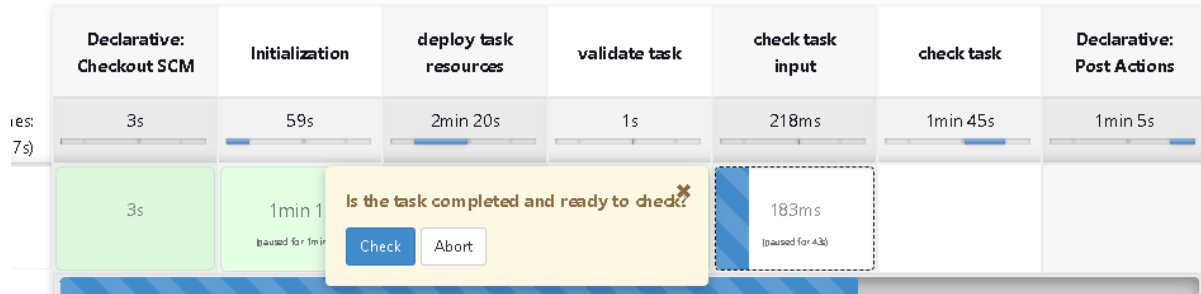


Рисунок 5.6 – Кнопка перевірки завдання

Опис завдання можна побачити натиснувши на джобі, яку ми запустили (рис. 5.7) та натиснути task definition.

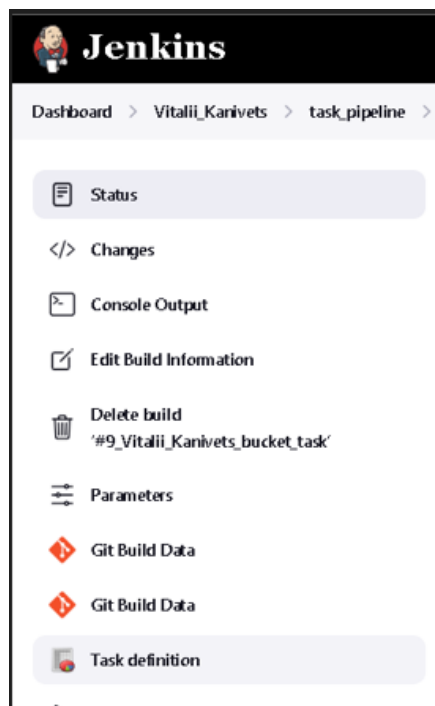


Рисунок 5.7 – Кнопка Task definition

Перейшовши до опису ми дізнаємося, що було створено cloud bucket в якому

розмістили `index.html` та потрібно зробити так щоб була можливість бачити `index.html` в якості статичного сайту для проходження завдання (рис. 5.8).

[Back to #10 Vitalii Kanivets bucket task](#) Task definition

Cloud bucket

Disclaimer

This task is designed to be resolved utilising GCP Free Tier eligible resources. Cloud Mentor team is not responsible for any costs, which may occur in case of not following instructions.

Hosting a Static Website Using Cloud Bucket

The goal of this task is to host a static website using Cloud bucket by configuring bucket properties and permissions.

In this task, you should work with the following resources:

- a bucket with a name starting with `cloudmentor-cloud-bucket-static-site-116745` - this bucket contains one file `index.html`

In two moves, you must configure a bucket as a static website.

One move is creating, updating, or deleting an GCP resource. Some validation steps can be passed without any action applied, to complete the task make sure that all the steps are passed.

To make sure everything has been done correctly, check that the website endpoint of the cloud bucket opens `index.html` in a browser.

Рисунок 5.8 – Опис завдання

Перейшовши в консоль GCP за посиланням [9], ми переконаємося, що такий bucket існує та в ньому є цей файл `index.html` (рис. 5.9).

The screenshot shows the 'Bucket details' page for 'cloudmentor-cloud-bucket-static-site-116745'. The bucket is located in 'us-central1 (Iowa)', has a 'Standard' storage class, and 'Not public' access. The 'OBJECTS' tab is active, showing a table of objects:

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expi
<code>index.html</code>	228 B	text/html	May 17, 2023, 5:14:47 PM	Standard	May 17, 2023, 5:14:47 PM	Not public	—	Google-managed key	—

Рисунок 5.9 – Створений cloud bucket та файл `index.html`

Щоб перевірити чи є можливість дивитися цей файл, як статичний сайт, треба перейти в деталі `index.html`, де ми побачимо URL (рис. 5.10).

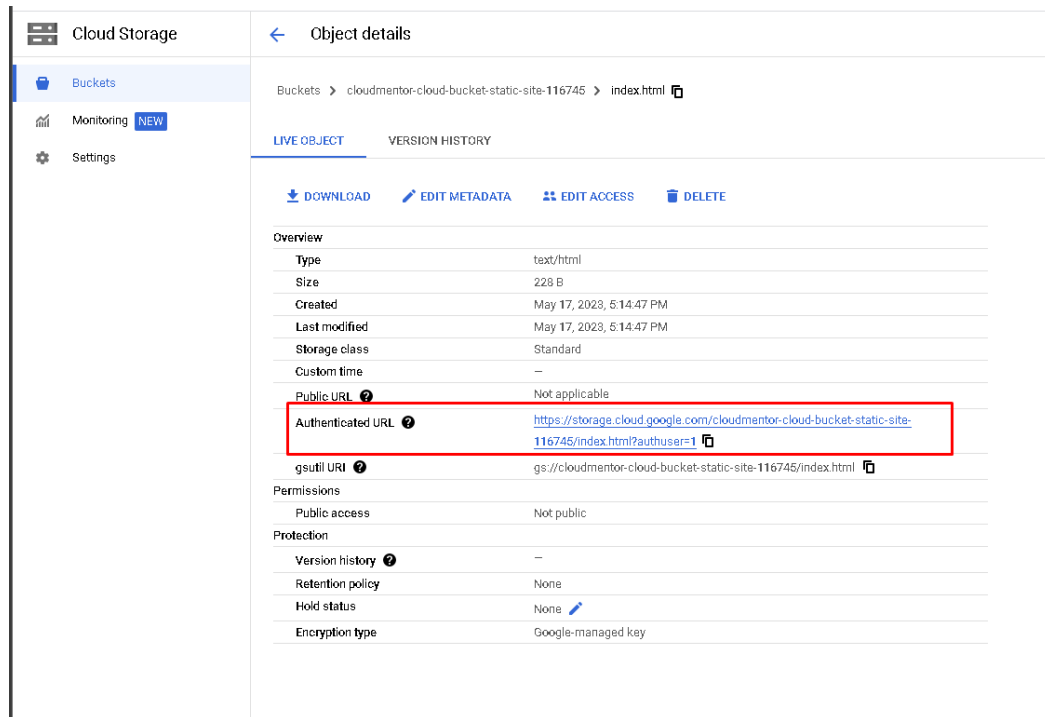


Рисунок 5.10 – URL файлу index.html

Використавши цю URL в приватному вікні ми переконаємося, що доступу до файлу не має. Щоб його побачити треба авторизуватися під акаунтом у якого є доступ (рис. 5.11), що є не вірним і треба налаштувати загальний доступ до bucket файлів.

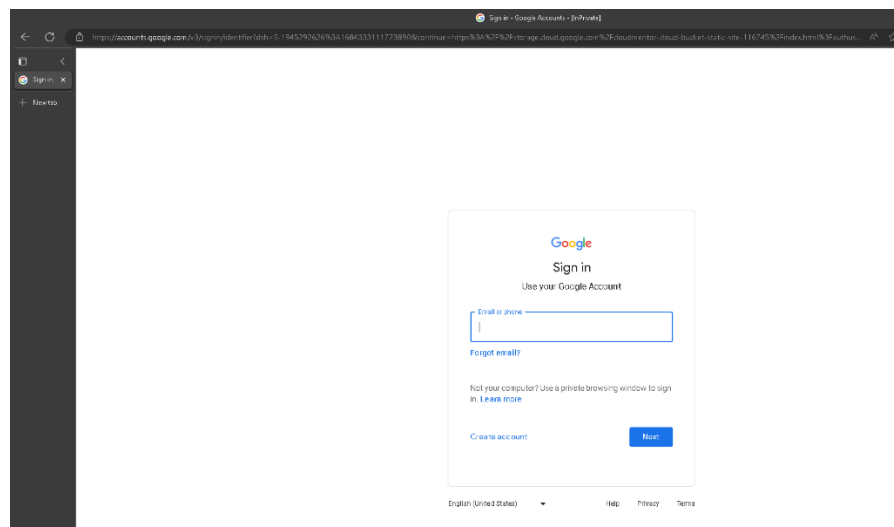


Рисунок 5.11 – Перехід по URL в приватному вікні

Для надання доступу всім потрібно виконати дві умови. Прибрати заповігання бакету на надання публічного доступу та надати публічний доступ до нього. Бажано надавати права тільки на перегляд файлів, щоб гарантувати, що ніхто не змінить або видалить файли у бакеті.

Переходимо назад до спуску бакетів та вибираємо редагувати доступ для потрібного бакету (рис. 5.12).

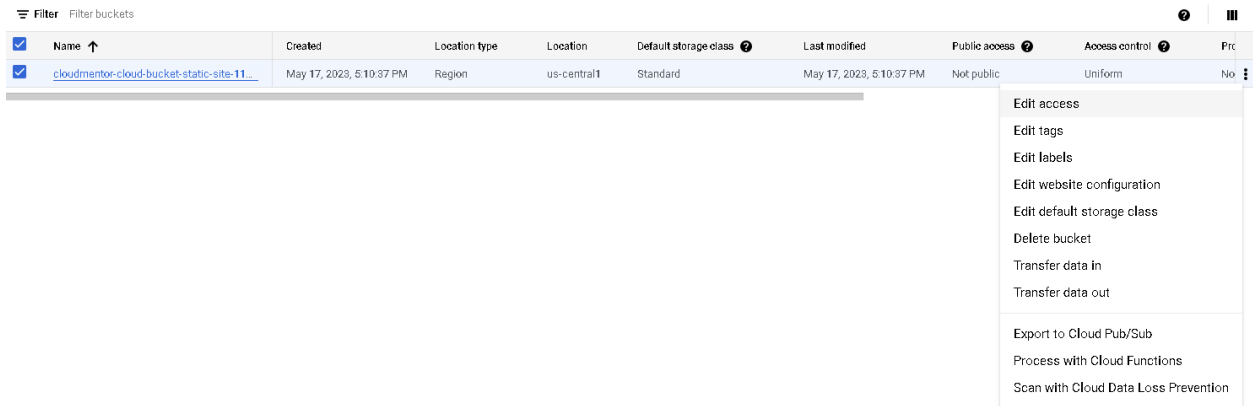


Рисунок 5.12 – Редагування доступу bucket файлів

Підтверджуємо, що ми хочемо прибрати заповігання на надання публічного доступу, як показано на рис. 5.13.

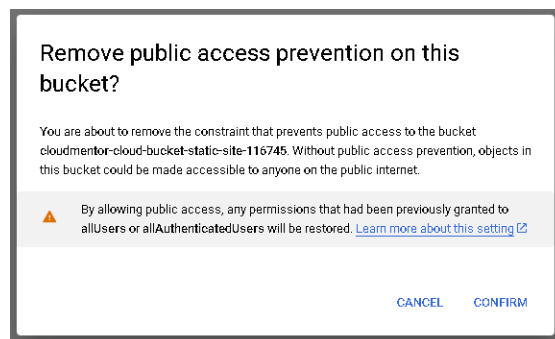


Рисунок 5.13 – Надання дозволу на присвоєння публічного доступу

Далі надаємо публічний доступ вибравши в якості principals allUsers, що гарантуватиме надання доступу всім користувачам навіть без аутентифікації в своєму акаунті. Також встановлюємо доступ перегляду до об'єктів бакету (рис. 5.14).

Grant access to "cloudmentor-cloud-bucket-static-site-116745"

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. [Learn more about IAM conditions](#)

Resource
cloudmentor-cloud-bucket-static-site-116745

Add principals
Principals are users, groups, domains, or service accounts. [Learn more about principals in IAM](#)

New principals
allUsers

Assign roles
Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role *
Storage Object Viewer
Read access to GCS objects.

IAM condition (optional)
[+ ADD IAM CONDITION](#)

[+ ADD ANOTHER ROLE](#)

[SAVE](#) [CANCEL](#)

Рисунок 5.14 – Надання публічного доступу

GCP нас перепитає чи згодні ми з такими змінами, тому треба підтвердити їх (рис. 5.15).

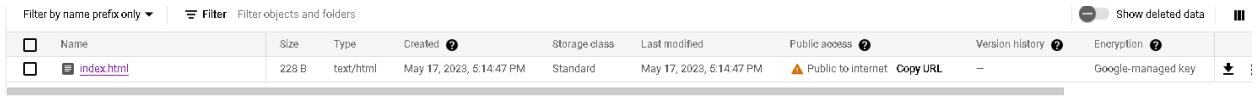
Are you sure you want to make this resource public?

Adding allUsers or allAuthenticatedUsers to this resource will make it publicly accessible to anyone on the internet. If this resource contains data that should not be made public to everyone, cancel this action to prevent public access. [Learn more](#)

[CANCEL](#) [ALLOW PUBLIC ACCESS](#)

Рисунок 5.15 – Підтвердження надання доступу

Після проведених налаштувань ми побачимо у файлі `index.html` позначку, що він став публічний і можна скопіювати URL (рис. 5.16).



Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption
<code>index.html</code>	228 B	text/html	May 17, 2023, 5:14:47 PM	Standard	May 17, 2023, 5:14:47 PM	Public to internet Copy URL	-	Google-managed key

Рисунок 5.16 – Публічна URL

Перейшовши по URL в приватному вікні ми переконаємося, що тепер в нас налаштований статичний сайт в Cloud Storage (рис. 5.17).

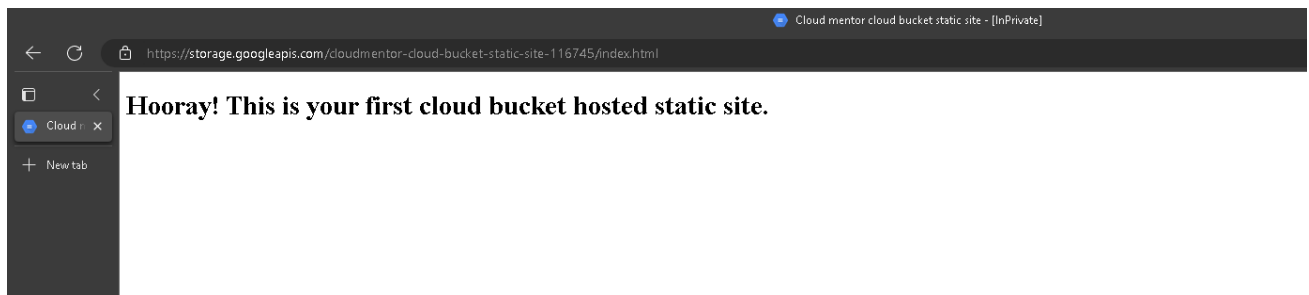


Рисунок 5.17 – Публічна URL у приватному вікні

Тепер ми можемо натиснути в Jenkins pipeline на кнопку перевірки завдання, після чого почнеться його валідація на наявність недоліків. Коли перевірка пройде pipeline покаже нам позитивний результат без помилок та пропонуватиме перевірити по новій наше завдання, що буде свідчити в успішно пройденому завданні (рис. 5.18).

Jenkins Search (CTRL+K) Vitalii_Karivets log out

Dashboard > Vitalii_Karivets > task_pipeline

Pipeline task_pipeline
Full project name: Vitalii_Karivets/task_pipeline

Build History: **#9 Vitalii_Karivets_bucket_task** (May 12, 2023, 12:15 PM)
student_id: Vitalii_Karivets
task: bucket_challenge_task
grades: 100

Stage View

	Declarative: Checkout SCM	Initialization	deploy task resources	validate task	check task input	check task	Declarative: Post Actions
Average stage times: (Average full run time: ~9min 17s)	3s	58s	2min 6s	693ms	253ms	1min 45s	1min 5s
#9 Vitalii_Karivets_bucket_task May 12 12:15 No Changes	3s	58s <small>(based for 216)</small>	2min 6s	693ms	253ms <small>(based for 2m 12.26s)</small>	1min 45s	1min 5s <small>(based for 13)</small>

Seed job: [create_user_workspace](#)
⚠ This item has been changed manually since it was generated by the seed job.

Permalinks

- [Last build \(#9_Vitalii_Karivets_bucket_task\)](#), 5 days 0 hr ago
- [Last stable build \(#9_Vitalii_Karivets_bucket_task\)](#), 5 days 0 hr ago

Рисунок 5.18 – Пройденный pipeline

ВИСНОВОК

У цій роботі була розглянута розробка навчальної платформи для розвитку навичок роботи з хмарними сервісами. Навчальна платформа є актуальною і важливою для забезпечення якісної підготовки фахівців у галузі хмарних технологій.

У ході дослідження проаналізовані різні технології, що можуть бути використані для розробки навчальної платформи. До таких технологій належать terraform, Jenkins, хмарні платформи, Gitlab.

Результатом роботи удосконалення навчальної платформи Cloud Mentor, яка дозволяє користувачам розвивати навички роботи з хмарними сервісами, такими як AWS та тепер і GCP. Навчальна платформа може містити різноманітні матеріали та вправи для розвитку навичок використання хмарних сервісів. Кожне завдання може бути розроблено в залежності від потреб, що надає високу гнучкість цієї платформи. Проте з гнучкістю постає і складність реалізації цих задач. Навіть з доволі простим завданням потрібно попрацювати.

Перспективним варіантом подальшого розвитку дослідження є розробка та запровадження мультихмарних завдань, що надало б можливість користувачам не тільки окремо вивчати хмари, а працювати з ними разом, бо часто потрібні знання і з декількох хмар для їх інтеграції чи міграції.

Отже, розробка навчальної платформи для розвитку навичок роботи з хмарними сервісами є важливим кроком у забезпеченні якісної підготовки фахівців у галузі хмарних технологій. Розроблена навчальна платформа може бути використана як у професійній освіті, так і в самостійному навчанні. Дана робота може стати основою для подальшого розвитку та вдосконалення навчальних платформ у галузі хмарних технологій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ількевич Н.С. Хмарні технології в освіті. Навчально-методичний посібник для студентів фізико-математичного факультету. – Житомир: вид-во ЖДУ, 2021. – 88 с.
2. Azure [Електронний ресурс] – Режим доступу до ресурсу: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>.
3. AWS [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/>.
4. Google Cloud [Електронний ресурс] – Режим доступу до ресурсу: <https://cloud.google.com/>.
5. UDEMY: A CASE STUDY IN ONLINE EDUCATION AND RAINING. – 2018. – С. 54.
6. Terraform [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.hashicorp.com/terraform/intro>.
7. Rafal L. Continuous Delivery with Docker and Jenkins / Leszko Rafal. – Livery Place, 2017. – 521 с. – (Packt Publishing Ltd.)
8. Gitlab [Електронний ресурс] – Режим доступу до ресурсу: <https://about.gitlab.com/>.