

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

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

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Дослідження динамічного сайту для кафе _____
_____ (тема)

Виконав:
студент 2 курсу, групи ІМІм-22-2
_____ Валюх Д.М. _____
(прізвище, ініціали)

Спеціальність 172 «Телекомунікації
_____ та радіотехніка» _____
(код і повна назва спеціальності)

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

Освітня програма «Інформаційно-мережна
_____ інженерія» _____
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри _____
(підпис)

Безрук В.М.
(прізвище, ініціали)

2024 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____

Керівник _____

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 _____
(підпис)
«__» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Валюх Дмитру Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження динамічного сайту для кафе

затверджена наказом університету від 18 березня 2024р. № 232 Ст_

2. Термін подання студентом роботи до екзаменаційної комісії 12 червня 2024р.

3. Вихідні дані до роботи: _____

Провести аналіз доступних сервісів хмарних обчислень для застосування в технологіях динамічного сайту. Основні послуги, що надаються хмарними системами. Мови реалізації моделі – Python3, HTML.

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

навчання машин: основні поняття та методи;

обробка природної мови NLP;

методи вилучення тексту з інтернету;

модель використання Beautiful Soup;

результати роботи моделі.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 16 слайдів у форматі PowerPoint _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Кривенко С.А.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	<i>Ознайомлення із завданням. Уточнення ТЗ.</i>	<i>21.03</i>	ВИК
2	<i>Підбір літератури за темою роботи.</i>	<i>21.03-28.03</i>	ВИК
3	<i>Виконання розділу 1</i>	<i>29.03-04.04</i>	ВИК
4	<i>Виконання розділу 2</i>	<i>05.04-11.04</i>	ВИК
5	<i>Виконання розділу 3</i>	<i>12.04-28.04</i>	ВИК
6	<i>Виконання розділів 4, 5</i>	<i>29.04-05.05</i>	ВИК
7	<i>Оформлення презентаційного матеріалу, підготовка до захисту у ЕК</i>	<i>06.05-12.06</i>	ВИК

Дата видачі завдання 14 березня 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Кривенко С.А
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка випускної магістерської кваліфікаційної роботи містить: 94 стор., 64 рис., 25 джерел.

AMAZON ELASTIC COMPUTE CLOUD, AMAZON MACHINE IMAGE, AWS, DISASTER RECOVERY.

Об'єкт дослідження – динамічний веб-сайт.

Предмет дослідження – робота побудована навколо вигаданого бізнес-кейсу. Бізнес-кейс надає спосіб досліджувати теми хмарних обчислень у контексті відповідних бізнес-потреб.

Метою даної роботи є створення динамічного веб-сайту кафе за допомогою Amazon EC2 для розробки моделі дослідження можливостей та вразливостей цього веб-сайту.

Методи досліджень – методи сервісів AWS.

ABSTRACT

The explanatory note of the final master's qualification work contains: 94 pages, 64 figures, 25 sources.

AMAZON ELASTIC COMPUTE CLOUD, AMAZON MACHINE IMAGE, AWS, DISASTER RECOVERY.

The research object is a dynamic website.

The subject of research is the work built around a fictional business case. A business case provides a way to explore cloud computing topics in the context of relevant business needs.

The purpose of this work is to create a dynamic website of a cafe using Amazon EC2 to develop a model for investigating the capabilities and vulnerabilities of this website.

Research methods - methods of AWS services.

ЗМІСТ

Перелік скорочень, умовних позначень, символів, одиниць і термінів	5
Вступ.....	6
1 Огляд додавання рівня обчислення.....	10
1.1 Додавання обчислень за допомогою Amazon EC2.....	10
1.2 Огляд обладнання та завдання	15
2 Підготовка екземпляра EC2 для розміщення веб-сайту.....	17
2.1 Аналіз існуючого екземпляра EC2.....	17
2.2 Підключення до IDE на примірнику EC2.....	18
2.3 Аналіз середовища стеку LAMP і підтвердження того, що веб-сервер доступний	20
3 Встановлення динамічної програми веб-сайту на примірнику EC2.....	24
3.1 Завдання 4: Встановлення програми café.....	24
3.2 Тестування веб-додатку	29
3 Створення веб-сайтів для розробки та виробництва в різних регіонах AWS	30
3.3 Створення АМІ та запуск іншого екземпляра EC2	30
3.4 Перевірка нового екземпляра café.....	35
Висновки	36
Список літератури	37
Додаток А. Слайди презентації.....	40
Додаток Б. Код.....	49
Додаток В. Публікації за темою роботи.....	55

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AMI – Amazon Machine Image – образ машини Amazon — це особливий тип віртуального пристрою, який використовується для створення віртуальної машини в EC2

API – Application Programming Interface – інтерфейс прикладного програмування

AWS – Amazon Web Services – дочірня компанія Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки

DR – disaster recovery – аварійне відновлення

EC2 – Amazon Elastic Compute Cloud – веб сервіс, котрий надає обчислювальні потужності в хмарі

HTML – HyperText Markup Language – мова розмітки гіпертексту

IDE – integrated development environment – Інтегроване середовище розробки

LAMP – аббревіатура набору вільного програмного забезпечення з відкритим кодом, в який входять Linux, веб сервер Apache, MySQL та інтерпретатор Perl/PHP/Python — основні компоненти для побудови життєздатного багатоцільового веб сервера[

URL – Uniform Resource Locator – єдиний вказівник на ресурс

S3 – Amazon Simple Storage Service – сервіс-сховище даних

SysOps – system operator – Системний оператор

ВСТУП

Робота побудована навколо вигаданого бізнес-кейсу. Бізнес-кейс надає спосіб досліджувати теми хмарних обчислень у контексті відповідних бізнес-потреб.

Цей сценарій має на меті надати модель застосовності технічних концепцій, які досліджуються, у реальному світі.

Після того, як кафе запустило першу версію свого веб-сайту, клієнти розповіли персоналу кафе, як гарно виглядає веб-сайт. Однак, окрім похвали, клієнти часто запитували, чи можна зробити онлайн-замовлення.

Власники бізнесу і менеджер Софія обговорили ситуацію. Вони погодилися, що їхня бізнес-стратегія та рішення повинні зосереджуватися на тому, щоб задовольнити своїх клієнтів і забезпечити їм найкращі враження від кафе.

Бізнес-запит для кафе: Запуск динамічного веб-сайту.

Кав'ярня хоче запровадити онлайн-замовлення для клієнтів, а також можливість персоналу кафе переглядати зроблені замовлення. Їх поточна архітектура веб-сайту, де веб-сайт розміщено на Amazon S3, не підтримує нові бізнес-вимоги.

У першій частині цієї роботи виконується роль Софії та за допомогою Amazon EC2 створено динамічний веб-сайт для кафе [1].

Метою даної роботи є створення динамічного веб-сайту кафе за допомогою Amazon EC2 для розробки моделі дослідження можливостей та вразливостей цього веб-сайту.

У цій роботі виконані такі завдання: аналіз наявного екземпляра EC2; підключення до IDE на примірнику EC2; аналіз середовища стеку LAMP і підтвердження того, що веб-сервер доступний; встановлення програми café; тестування веб-додатку; створення AMI та запуск іншого екземпляра EC2; перевірка нового екземпляра café [2].

Другий розділ пояснювальної записки присвячений розв'язанню першої проблеми. Бізнес-запит для кафе це – підготовка екземпляра EC2 для розміщення веб-сайту.

Кав'ярня хоче запровадити онлайн-замовлення для клієнтів, а також можливість персоналу кафе переглядати зроблені замовлення. Їх поточна архітектура веб-сайту, де веб-сайт розміщено на Amazon S3, не підтримує нові бізнес-вимоги. У першій частині цієї роботи розглядається роль Софії. Налаштовано екземпляр Amazon EC2 так, щоб він був готовий розмістити веб-сайт для кафе. У першому завданні записані деталі про наявний екземпляр EC2, створений в обліковому записі AWS. Завдання 2 – це підключення до IDE на примірнику EC2. AWS Cloud9 — це сервіс, який може працювати на екземплярі EC2. Він забезпечує інтегроване середовище розробки (integrated development environment - IDE), яке включає такі функції, як редактор коду, інструменти налагодження і термінал. Використовуючи середовище AWS Cloud9, не потрібно завантажувати пару ключів і підключатися до примірника EC2 за допомогою PuTTY або подібного програмного забезпечення Secure Shell (SSH). Використовуючи AWS Cloud9, також не потрібно використовувати інструменти редагування тексту командного рядка (наприклад, vi або nano) для редагування файлів на примірнику Linux. Завдання 3 – це аналіз середовища стеку LAMP і підтвердження того, що веб-сервер доступний. Мета цієї роботи — це налаштувати примірник EC2 для розміщення нового динамічного веб-сайту для кафе. У цьому завданні виконано аналіз того, що вже встановлено.

Третій розділ реалізує нову бізнес-вимогу: встановлення динамічної програми веб-сайту на примірнику EC2 (Проблема №2).

У попередньому завданні налаштовано примірник EC2. Тепер відомо, що PHP інстальовано, а середовище програми має запущену реляційну базу даних. Крім того, середовище має запущений веб-сервер, до якого можна отримати доступ з Інтернету. Тепер є основні налаштування для розміщення динамічного веб-сайту для кафе. У другій частині цієї роботи виконана роль Софії та встановлена

програма `café` на примірнику EC2. Завдання 4 – це встановлення програми `café`. Завдання 5 – це тестування веб-додатку.

Четвертий розділ описує методику вирішення третьої проблеми: створення веб-сайтів для розробки та виробництва в різних регіонах AWS.

Усі в кафе вражені новим динамічним веб-сайтом, який створила Софія! Клієнти в захваті від того, що тепер вони можуть розміщувати онлайн-замовлення та планувати вивезення десертів. Задоволеність клієнтів зросла завдяки скороченню часу очікування. Однак, крім похвали, виникає ще одна вимога до бізнесу. Власники хотіли б мати два веб-сайти кафе: один веб-сайт, який можна використовувати як середовище розробки для моделювання нових функцій і веб-дизайну, перш ніж вони будуть доступні клієнтам; та окремий веб-сайт, на якому розміщено виробниче середовище, яким користуються клієнти. Софія обговорила нову вимогу з досвідченим інженером AWS SysOps, коли він одного ранку зайшов у кафе випити кави. Він припустив, що в ідеалі ці два середовища повинні існувати в різних регіонах AWS. Така конструкція матиме додаткову перевагу, забезпечуючи більш надійне аварійне відновлення (`disaster recovery - DR`) у малоімовірному сценарії, коли регіон AWS стає тимчасово недоступним. Софія зараз дуже зайнята! Коли вона виконує більш вражаючу роботу, її навички стають більш затребуваними. Завдання 6 – це створення AMI та запуск іншого екземпляра EC2. Оскільки веб-сайт кафе вже добре працює на існуючому екземплярі EC2, Софія вирішує дублювати його, створивши з нього AMI. Потім вона запусить новий екземпляр із нового AMI. Для виконання цього завдання виконувалась роль Софії. Перед створенням AMI з цього екземпляра, була створена нова пара ключів, яку важливо мати в цій роботі. Завдання 7 – це перевірка нового екземпляра `café`.

Софія тепер герой у кафе! Вона створила динамічний веб-сайт, а також дублікат того самого веб-сайту, який працює в другому регіоні AWS.

Софія вирішила призначити перший екземпляр EC2, який вона створила — той, що знаходиться в регіоні `us-east-1` — як екземпляр розробки. Другий

примірник, який вона створила — той, що знаходиться в Орегоні (region us-west-2) — буде робочим примірником.

Таким чином, Софія та будь-які інші розробники додатків можуть тестувати вдосконалення додатків на сайті розробки, не впливаючи на робочий сайт. Потім, коли розробники вирішають, що вдосконалення виглядають добре і вони повністю їх перевірили, вони зможуть перенести код на робочий сайт.

Софія пояснила власникам, що вона зробила. Хоча вони не повністю розуміли всі технічні терміни, які використовувала Софія, вони були раді дізнатися, що веб-сайт тепер може приймати онлайн-замовлення. Вони також були раді почути, що тепер вони можуть тестувати нові вдосконалення веб-сайту, не повідомляючи про ці зміни клієнтам.

1 ОГЛЯД ДОДАВАННЯ РІВНЯ ОБЧИСЛЕННЯ

AWS пропонує кілька варіантів обчислення для задоволення різних потреб. Коли проектується архітектура для підтримки певного типу робочого навантаження, важливо розуміти доступні параметри обчислення. Ключові варіанти обчислень під час виконання можна згрупувати в чотири категорії моделі хмарних обчислень: віртуальні машини (VM); контейнери; платформа як послуга, яка також відома як PaaS; і без сервера. Крім того, можна використовувати спеціалізовані рішення для вирішення конкретних випадків використання комп'ютера.

1.1 Додавання обчислень за допомогою Amazon EC2.

1.1.1 Вибір середовища роботи прикладних програм

У категорії віртуальних машин AWS пропонує дві основні послуги. Перший сервіс — Amazon Elastic Compute Cloud (Amazon EC2). Він забезпечує безпечні віртуальні сервери в хмарі зі змінним розміром. Другий сервіс – Amazon Lightsail. Він надає віртуальні приватні сервери для виконання простих робочих навантажень економічно ефективним способом.

У категорії контейнерів AWS пропонує Amazon Elastic Container Service (Amazon ECS). Це дає змогу запускати програми-контейнери Docker на AWS.

Категорія PaaS включає AWS Elastic Beanstalk. Це рішення, яке запускає веб-додатки та служби, розроблені такими мовами, як Java, .NET, PHP, Node.js, Python, Ruby, Go та Docker.

Категорія без сервера включає AWS Lambda, яке є обчислювальним рішенням без сервера, яке виконує Java, Go, PowerShell, Node.js, C#, Python або

Ruby. Ця категорія також включає AWS Fargate, яка надає обчислювальну платформу без сервера для контейнерів.

Для спеціалізованих рішень AWS Outposts надає можливість запускати інфраструктуру та служби AWS локально, а AWS Batch — це служба, яка виконує пакетні завдання будь-якого масштабу.

Коли обирається обчислювальне середовище виконання AWS для робочого навантаження, необхідно враховувати, що віртуальні машини та служби на основі контейнерів забезпечують більше контролю над інфраструктурою та забезпечують більший ступінь налаштування. PaaS і послуги без сервера дозволяють зосередитися більше на програмі, а не на інфраструктурі. Вони також забезпечують швидке розгортання. Сервіси в категорії спеціалізованих рішень стосуються конкретних типів робочих навантажень або гібридної хмари та пакетної обробки. Ці спеціалізовані служби добре працюють для таких випадків використання, оскільки ними також повністю керує AWS. У цій дослідницькій роботі основна увага зосереджена на Amazon EC2.

1.1.2 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) забезпечує обчислення в хмарі. Можна використовувати Amazon EC2 для надання віртуальних серверів і повністю контролювати обчислювальні ресурси цих серверів. Можна отримати та запустити нові екземпляри сервера за лічені хвилини. Можна швидко збільшувати та зменшувати ємність у міру зміни вимог до комп'ютера. З точки зору вартості, ви можна платити лише за ємність, яка використовується.

Чому EC2 називається Elastic Compute Cloud?

Еластичний, тому що можна легко збільшити або зменшити кількість серверів, які ви запускаєте для автоматичної підтримки програми. Також можна збільшити або зменшити розмір існуючих серверів

Обчислення, тому що більшість користувачів запускають сервери для розміщення запущених програм або обробки даних, які потребують обчислювальних ресурсів. Ці ресурси включають процесорну потужність CPU і пам'ять RAM.

Хмара, оскільки екземпляри EC2, які ви запускаються, розміщені в хмарі.

1.1.3 Екземпляри EC2

Екземпляри Amazon EC2 працюють як віртуальні машини на хост-комп'ютерах, розташованих у зонах доступності AWS. На кожній віртуальній машині працює операційна система, наприклад Amazon Linux або Microsoft Windows. Можна встановлювати та запускати програми на операційній системі на кожній віртуальній машині. Навіть можна запускати корпоративні програми, які охоплюють кілька віртуальних машин.

Віртуальні машини працюють на рівні гіпервізора, який підтримується AWS. Гіпервізор — це рівень операційної платформи, який надає екземпляру EC2 доступ до фактичних фізичних апаратних ресурсів, необхідних для роботи, наприклад процесорів, пам'яті та сховища.

Деякі примірники EC2 використовують сховище примірників. Сховище екземплярів також відоме як ефемерне сховище. Це сховище, яке фізично приєднане до головного комп'ютера та надає примірнику тимчасове сховище на рівні блоку.

Багато екземплярів EC2 використовують Amazon Elastic Block Store (Amazon EBS) для завантажувального диска та інших потреб у сховищі. Amazon EBS надає постійні блокові томи зберігання, що означає, що дані зберігатимуться. Наприклад, дані зберігаються в цьому екземплярі, навіть якщо екземпляр EC2 перебуває в зупиненому стані.

Екземпляри, оптимізовані для EBS, забезпечують швидший доступ до підключеного тому Amazon EBS, зводячи до мінімуму конфлікти введення/виведення між томом та іншим трафіком від екземпляра.

Екземпляри EC2 можуть мати мережеве підключення до інших ресурсів, таких як інші екземпляри EC2, служби AWS та Інтернет. Можна налаштувати рівень доступу до мережі відповідно до своїх потреб і збалансувати потреби в доступності та вимоги безпеки. Різні типи екземплярів забезпечують різні рівні продуктивності мережі.

1.1.4 Кейси використання EC2

Amazon EC2 надає віртуальні машини, на яких можна розміщувати ті самі програми, що й на традиційному локальному сервері. Загальні випадки використання екземплярів EC2 включають веб-сервери, сервери додатків, сервери баз даних і медіа-сервери.

Зокрема, рекомендується Amazon EC2 як обчислювальний вибір у ситуаціях, коли потрібен повний контроль над обчислювальними ресурсами Amazon EC2 дає змогу налаштовувати та налаштовувати все, що стосується екземплярів, від операційної системи до програм. Наприклад, можна використовувати різні операційні системи, які включають Microsoft Windows і багато варіантів Linux. Крім того, можна вибрати екземпляри з процесорною архітектурою x86 або Advanced RISC Machine (ARM).

Варіанти оптимізації витрат на обчислення – Amazon EC2 пропонує кілька способів оплати інстансів EC2, зокрема інстанси на вимогу, плани заощаджень, зарезервовані інстанси та спотові інстанси. Також можна заплатити за виділені хости, які надають вам ємність екземплярів EC2 на фізичних серверах, призначених для особливого використання.

Віртуальний сервер для роботи будь-якого типу.

1.1.5 Налаштування Amazon EC2

Щоб створити примірник EC2, необхідно прийняти ключові рішення щодо деталей його конфігурації. Основні параметри, які потрібно вказати для запуску захищеного екземпляра, включають наступне.

Образ машини Amazon AMI – AMI визначає базову конфігурацію програмного забезпечення для примірника та використовується Amazon EC2 для запуску примірника.

Тип екземпляра – тип екземпляра визначає комбінацію центрального процесора, пам'яті, сховища та мережевої ємності, яка забезпечує певний рівень обчислювальних можливостей для запуску програми.

Розміщення в мережі та адресація – коли запускається екземпляр, можна вказати відповідне розміщення в мережі та адресацію, щоб забезпечити потрібний доступ до мережі та безпеку.

Передбачувана роль – за бажанням можна приєднати роль AWS Identity and Access Management (IAM), яка надає дозволи на доступ до служб AWS програмам, які працюють на екземплярі, або користувачам, підключеним до екземпляра.

Дані користувача – можна додатково виконати ініціалізацію або налаштувати конфігурацію екземпляра, вказавши сценарій даних користувача. Цей сценарій автоматично запускається під час запуску примірника.

Сховище – необхідно вказати тип сховища, яке використовуватиметься для зберігання кореневого або завантажувального тому примірника.

Група безпеки – також потрібно налаштувати нову групу безпеки або використати наявну. Група безпеки визначає, на яких портах дозволений мережевий трафік.

Пара ключів – пара ключів зазвичай також вказується під час запуску. Пара ключів використовується для підключень Secure Shell (SSH) або для встановлення доступу до екземпляра за протоколом віддаленого робочого стола RDP.

Деякі ключові висновки з цього підрозділу включають:

Amazon EC2 дозволяє запускати віртуальні машини Microsoft Windows і Linux у хмарі.

Можна використовувати екземпляр EC2, коли потрібен повний контроль над обчислювальними ресурсами та бажано виконати будь-який тип робочого навантаження.

Коли запускається примірник EC2, необхідно вибрати АМІ та тип примірника. Запуск екземпляра передбачає вказівку параметрів конфігурації, включаючи налаштування мережі, безпеки, сховища та даних користувача.

1.2 Огляд обладнання та завдання

У цій дослідницькій роботі необхідно розгорнути прикладну програму в примірнику Amazon Elastic Compute Cloud (Amazon EC2). Прикладна програма дозволяє кафе приймати онлайн-замовлення. Перевірити, чи програма працює належним чином у першому регіоні AWS (середовищі розробки), створите образ машини Amazon (АМІ) з примірника EC2. Також розгорнути другий екземпляр тієї самої програми як робоче середовище в іншому регіоні AWS.

Під час виконання цієї дослідницької роботи необхідно виконати наступні завдання:

- підключатися до AWS Cloud9 IDE на наявному екземплярі EC2 ;
- аналізувати середовище екземпляра EC2 і підтверджувати доступність веб-сервера;
- встановити веб-програму на примірник EC2, який також використовує сховище параметрів AWS Systems Manager;
- тестувати веб-програму;
- створити АМІ;
- розгорнути другу копію веб-додатку в іншому регіоні AWS.

Перед початком дослідження, деякі ресурси вже були створені в обліковому записі AWS (рис. 1.1).

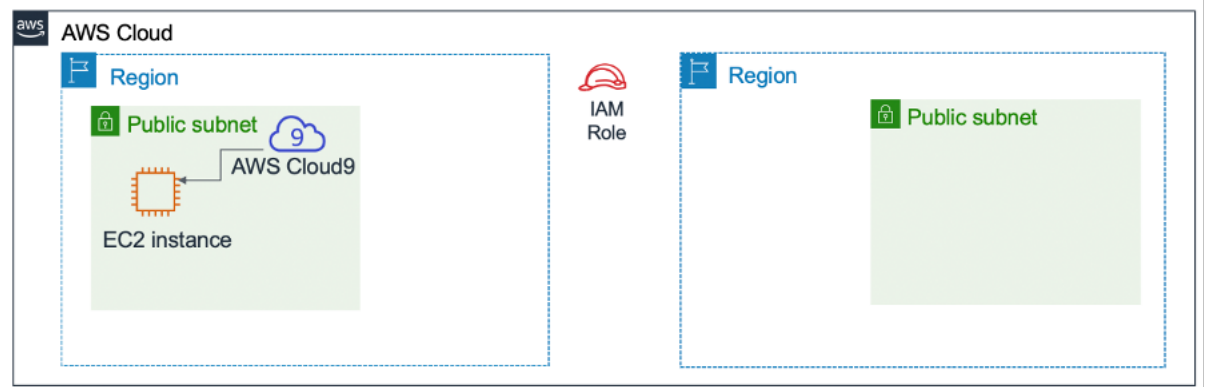


Рисунок 1.1 – Початкова інфраструктура

Наприкінці цієї дослідницької роботи архітектура має виглядати як наведений нижче приклад (рис.1.2).

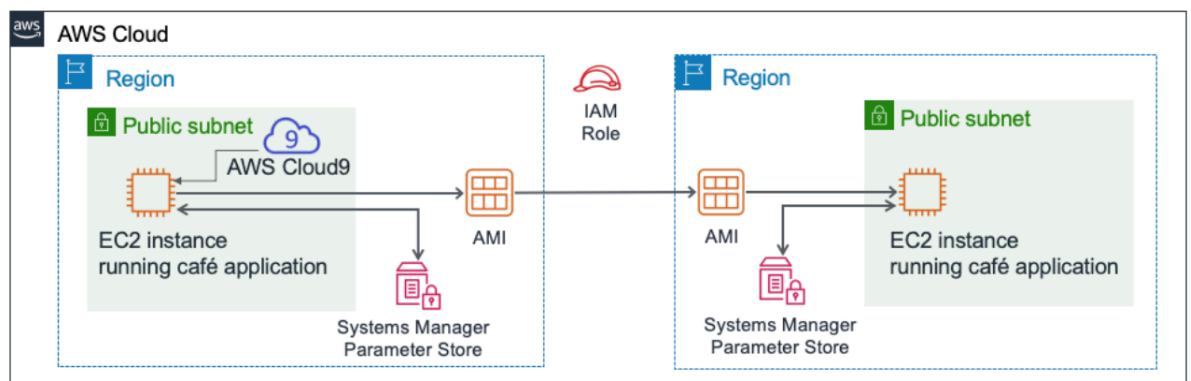


Рисунок 1.2 – Інфраструктура після проведення дослідження

2 ПІДГОТОВКА ЕКЗЕМПЛЯРА EC2 ДЛЯ РОЗМІЩЕННЯ ВЕБ-САЙТУ

Кав'ярня хоче запровадити онлайн-замовлення для клієнтів, а також можливість персоналу кафе переглядати зроблені замовлення. Їх поточна архітектура веб-сайту, де веб-сайт розміщено на Amazon S3, не підтримує нові бізнес-вимоги.

У першій частині цієї дослідницької роботи аналізується роль Софії. Проблема номер один бізнес запиту для кафе – налаштувати екземпляр Amazon EC2 так, щоб він був готовий розмістити веб-сайт для кафе.

2.1 Аналіз існуючого екземпляра EC2

У цьому завданні номер один було записано деталі про наявний екземпляр EC2, створений в обліковому записі AWS.

У полі пошуку поруч із пунктом «Сервіси» було вибрано EC2, щоб перейти до консолі EC2. Потім вибрано Примірники Instances.

Проаналізовано запущений екземпляр під назвою aws- cloud9- CafeWebServer-a7.... Цей екземпляр EC2 було створено до початку дослідження (рис.2.1).

Сформовано відповіді на перші чотири запитання дослідження:

- запитання 1: чи знаходиться примірник в загальнодоступній підмережі? Відповідь – так, subnet-099c1de0bc13f054a (Public Subnet);
- запитання 2: чи має примірник EC2 загальнодоступну IP-адресу IPv4? Відповідь – так, 3.239.39.193 (Public IPv4 address);
- питання 3: які номери вхідних портів TCP відкриті для цього екземпляра? Відповідь – TCP port 22 only, open to a specific range of IP addresses;

- запитання 4: чи пов'язана з примірником EC2 роль AWS Identity and Access Management (IAM)? Відповідь – ні.

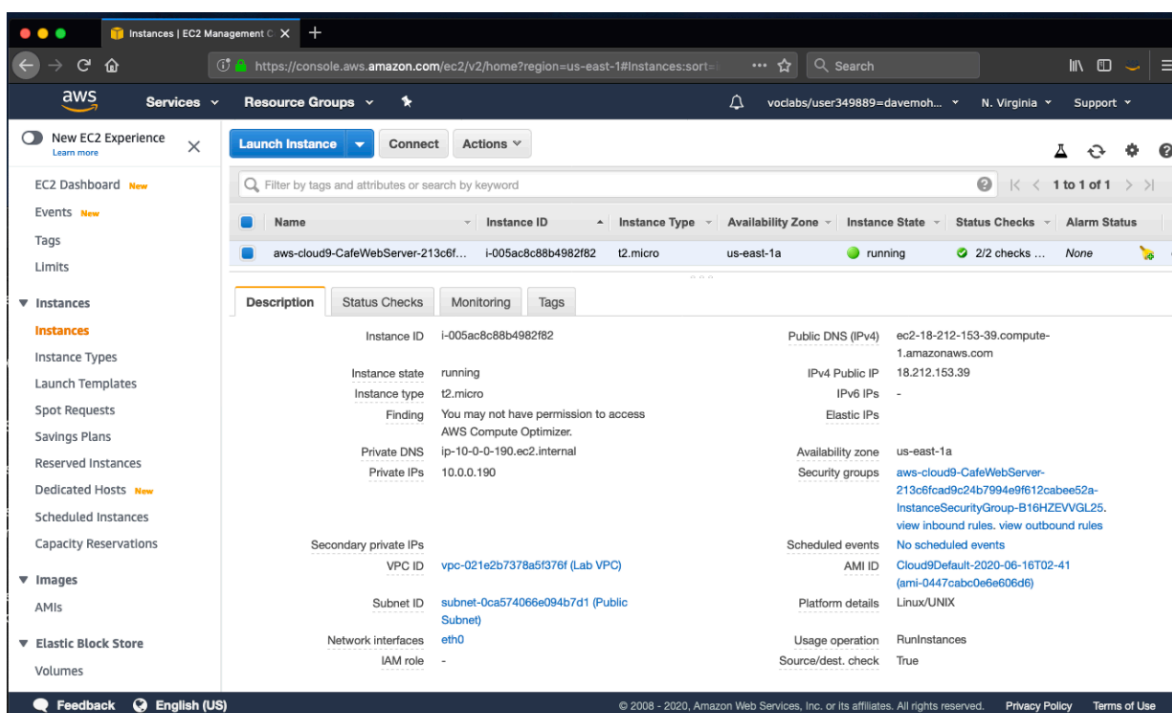


Рисунок 2.1 – Екземпляр EC2

2.2 Підключення до IDE на примірнику EC2

AWS Cloud9 — це сервіс, який може працювати на екземплярі EC2. Він забезпечує інтегроване середовище розробки (integrated development environment - IDE), яке включає такі функції, як редактор коду, інструмент налагодження і термінал.

Використовуючи середовище AWS Cloud9 (завдання номер два), не потрібно завантажувати пару ключів і підключатися до примірника EC2 за допомогою PuTTY або подібного програмного забезпечення Secure Shell (SSH). Використовуючи AWS Cloud9, також не потрібно використовувати інструменти редагування тексту командного рядка (наприклад, vi або nano) для редагування

файлів на примірнику Linux.

У полі пошуку поруч із пунктом «Сервіси» вибрано Cloud9, щоб перейти до консолі AWS Cloud9.

На сторінці «Середовища» Environments для середовища CafeWebServer зазначено тип – це екземпляр типу EC2. Для відкриття середовища вибрано Open.

Тепер менеджер (Софія) підключена до AWS Cloud9 IDE, яка працює на примірнику EC2, який описано раніше.

IDE містить (рис.2.2):

- термінал Bash у нижній правій панелі;
- переглядач файлів на панелі ліворуч, який показує файли в каталозі /home/ec2-user/environment екземпляра;
- редактор файлів у верхній правій панелі. Якщо двічі клацнути файл у файловому браузері, наприклад файл README.md, він відобразиться в редакторі.

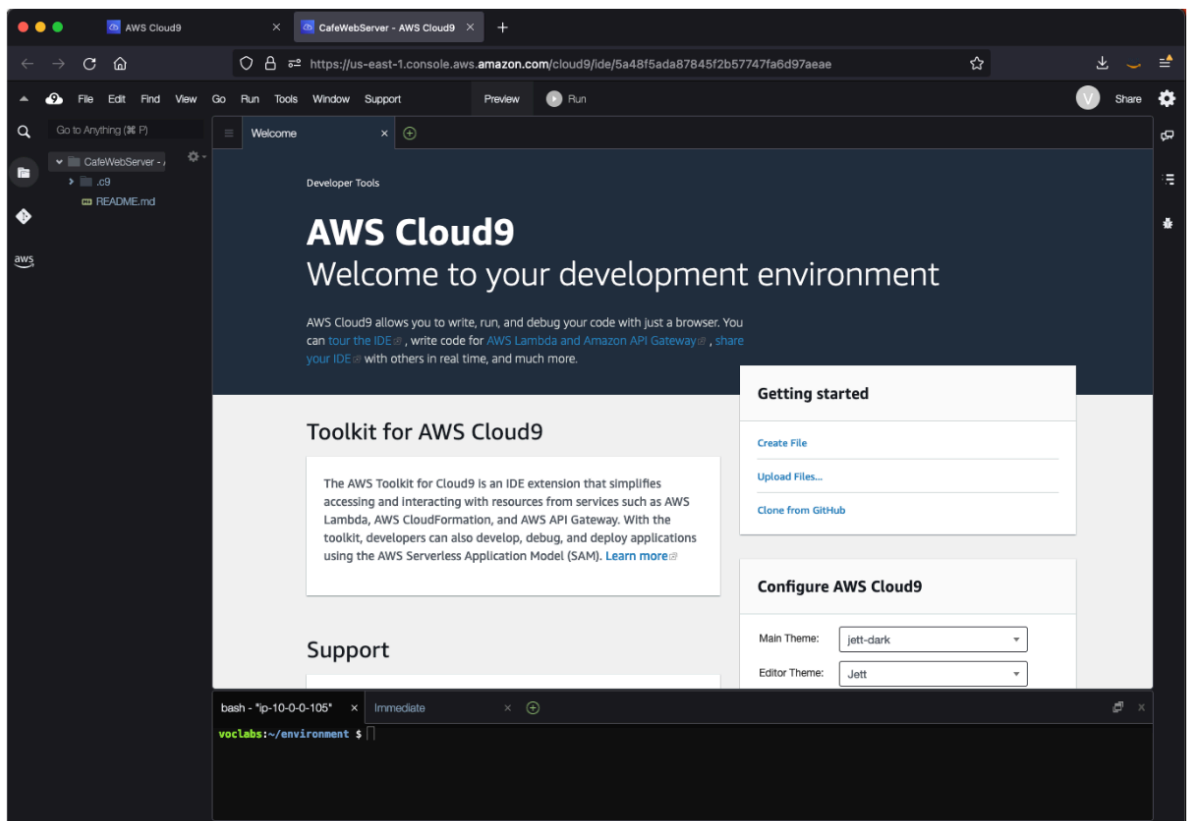


Рисунок 2.2 – AWS Cloud9 IDE

2.3 Аналіз середовища стеку LAMP і підтвердження того, що веб-сервер доступний

Мета цієї дослідницької роботи — налаштувати примірник EC2 для розміщення нового динамічного веб-сайту для кафе. У цьому завданні номер три проаналізовано те, що вже встановлено.

Зафіксовано версію операційної системи.

У терміналі bash AWS Cloud9 виконано команду:

```
cat /proc/version
```

Вихідні дані вказують, що це екземпляр Amazon Linux, приблизно аналогічний Red Hat 7.

Зафіксовано деталі веб-сервера, бази даних, PHP і стан сервера.

У терміналі виконано ці команди:

```
sudo httpd -v
```

```
service httpd status
```

```
mysql --version
```

```
service mysqld status
```

```
php --version
```

Вихідні дані відображають версії веб-сервера та бази даних, а також показують, що вони наразі не запущені.

```
voclabs:~/environment $ sudo httpd -v
```

```
Server version: Apache/2.4.59 ()
```

```
Server built: Apr 22 2024 13:07:26
```

```
voclabs:~/environment $ service httpd status
```

Redirecting to /bin/systemctl status httpd.service

- httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Drop-In: /usr/lib/systemd/system/httpd.service.d

└─php-fpm.conf

Active: inactive (dead)

Docs: man:httpd.service(8)

voclabs:~/environment \$

voclabs:~/environment \$ mysql --version

mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper

voclabs:~/environment \$ service mysqld status

Redirecting to /bin/systemctl status mysqld.service

Unit mysqld.service could not be found.

voclabs:~/environment \$

voclabs:~/environment \$ php --version

PHP 8.2.19 (cli) (built: May 21 2024 22:55:32) (NTS)

Copyright (c) The PHP Group

Zend Engine v4.2.19, Copyright (c) Zend Technologies

with Xdebug v3.2.2, Copyright (c) 2002-2023, by Derick Rethans

voclabs:~/environment \$

Далі було виконано крок 12, запущено веб-сервер і базу даних, а також налаштовано їх автоматичний запуск після будь-якого майбутнього перезапуску екземпляра EC2.

У терміналі виконано ці команди:

```
sudo chkconfig httpd on
```

```
sudo service httpd start
```

```
sudo service httpd status
sudo chkconfig mysqld on
sudo service mysqld start
sudo service mysqld status
```

Далі було виконано крок 13, налаштовано примірник EC2, щоб можна було використовувати редактор AWS Cloud9 для редагування файлів веб-сервера.

Браузер файлів AWS Cloud9 спочатку не відображає стандартний веб-каталог веб-сервера Apache.

У терміналі виконано ці дві команди:

```
ln -s /var/www/ /home/ec2-user/environment
sudo chown ec2-user:ec2-user /var/www/html
```

Перша команда створила символічне посилання з робочої області редактора AWS Cloud9 за замовчуванням на каталог /var/www, який містить файли веб-сервера.

Друга команда змінила право власності на підкаталог html, щоб користувач ec2-user (під яким увійшов менеджер Софія) міг редагувати та створювати нові файли в ньому.

Далі було виконано крок 14, створена проста тестова веб-сторінка:

- у браузері файлів розгорнуто каталог CafeWebServer > www і видалено каталог html;
- вибрано File > New File;
- на вкладці текстового редактора вставлено такий рядок: <html>Hello from the café web server!</html>;
- вибрано «Файл» > «Зберегти» File > Save та збережено файл у каталозі html як index.html.

Далі було виконано крок 14, зроблено веб-сайт доступним з Інтернету.

На цьому кроці було перевірено та оновлено конфігурації, які роблять веб-сторінки (розміщені на веб-сервері) доступними з Інтернету.

Знайдено загальнодоступну IP-адресу IPv4 екземпляра EC2 і завантажено `http://<public-ip>` у новій вкладці браузера. При цьому зафіксовано, що у браузері не завантажується повідомлення, яке було введено в index.html.

Щоб дозволити вхідний трафік HTTP на TCP-порт 80 з будь-якого місця, було оновлено групу безпеки екземпляра EC2.

3 ВСТАНОВЛЕННЯ ДИНАМІЧНОЇ ПРОГРАМИ ВЕБ-САЙТУ НА ПРИМІРНИКУ EC2

У попередньому завданні було налаштовано примірник EC2. На ньому інстальовано РНР, а середовище програми має запущену реляційну базу даних. Крім того, середовище має запущений веб-сервер, до якого можна отримати доступ з Інтернету. Тобто підготовлені основні налаштування для розміщення динамічного веб-сайту для кафе.

У другій частині цієї дослідницької роботи для нової бізнес вимоги (проблема №2) формується модель ролі менеджера Софії та встановлюється програма café на примірнику EC2.

3.1 Завдання 4: Встановлення програми café

В рамках завдання 4 було виконано наступні кроки.

Завантажено та розпаковано файли програми веб-сервера (крок 16).

У терміналі Bash виконано такі команди:

```
cd ~/environment
```

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/setup.tar.gz
```

```
tar -zxvf setup.tar.gz
```

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/db.tar.gz
```

```
tar -zxvf db.tar.gz
```

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/cafe.tar.gz
```

```
tar -zxvf cafe.tar.gz
```

Файловий браузер показує три файли .tar.gz, які було завантажено.

Також розпакувані ці архівні файли, які створили каталоги `safe`, `db`, і `setup` у робочому середовищі манеджера.

На 17-му кроці скопійовано файли `safe` до кореневої папки документа веб-сервера.

У терміналі Bash виконано команду:

```
mv safe /var/www/html/
```

Для спостереження за тим, як програма працює (крок18) виконані наступні дії.

Відкрито вихідний код `html/safe/index.php` у редакторі AWS Cloud9, двічі клацнувши його.

Цей файл містить HTML-код, але він також містить розділи, укладені в елементи. Ці елементи здійснюють виклики до інших систем і ресурсів.

Наприклад, у рядку 18 код PHP посилається на файл під назвою `getAppParameters.php`.

Файл `getAppParameters.php` відкритий у редакторі коду.

Рядок 3 цього файлу, викликається `AWSSDK`.

Крім того, у рядках 10–33 веб-програма створює клієнта, який підключається до служби `ssm`, яка є `AWS Systems Manager`. Потім програма отримує сім параметрів із `Systems Manager`. Ці параметри ще не створено в `AWS Systems Manager`, але це описується далі.

На дев'ятнадцятому кроці у сховищі параметрів `AWS Systems Manager` налаштовано параметри програми.

У терміналі Bash виконано такі команди:

```
cd setup
```

```
./set-app-parameters.sh
```

Запущений сценарій оболонки видавав команди інтерфейсу командного рядка AWS (`AWS CLI`). Ці команди додали параметри до сховища параметрів `Parameter Store`, які використовує програма.

У консолі керування AWS у меню «Служби» Services виберіть «Диспетчер систем» Systems Manager (крок 20).

На 21-му кроці на панелі ліворуч виберіть Parameter Store.

Зверніть увагу, що тут зберігається сім параметрів.

PHP-код програми `café` посилається на ці значення (наприклад, щоб він міг отримати інформацію про підключення до бази даних MySQL).

Вибрано параметр `/café/dbPassword` і скопійовано значення Value в буфер обміну, використання якого описано далі.

На кроці двадцять два налаштовано базу даних MySQL для підтримки програми `café`.

Повернувшись у термінал `bash` AWS Cloud9, виконано такі команди:

```
cd ../db/  
./set-root-password.sh  
./create-db.sh
```

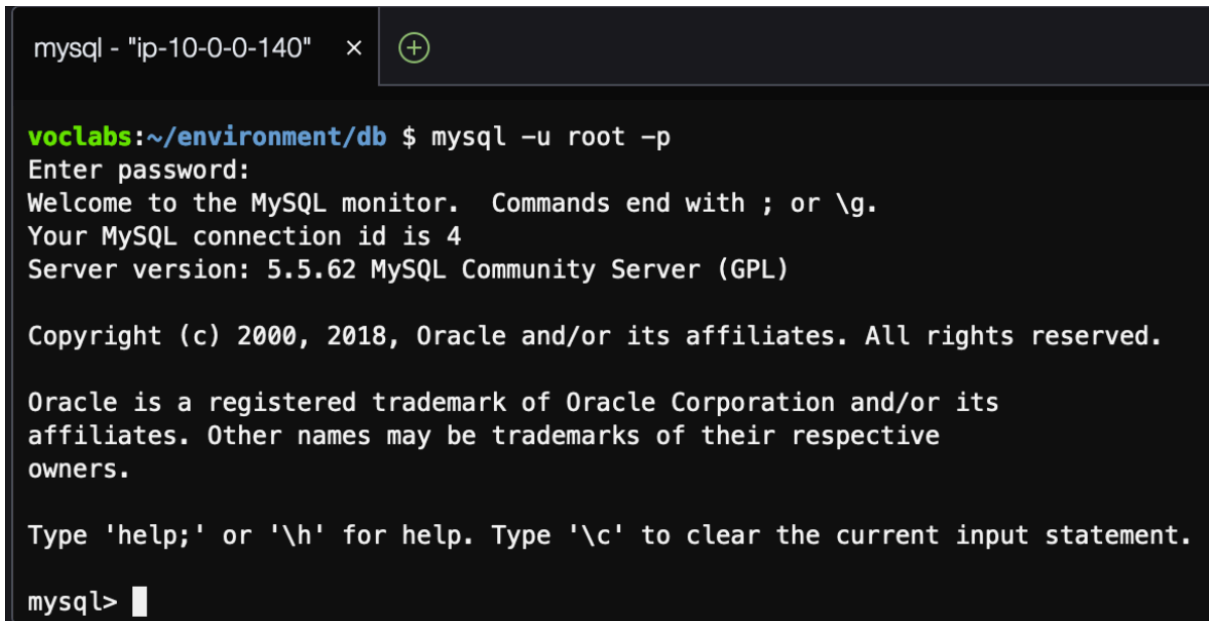
Для спостереження за створеними таблицями бази даних (крок 23) виконано наступні дії.

У терміналі Bash виконано команду, щоб підключити термінальний клієнт MySQL до бази даних:

```
mysql -u root -p
```

На запрошення Коли ввести пароль бази даних, введено скопійоване значення параметра `dbPassword`.

При цьому формується підказка `mysql>`, яка вказує на те, що менеджер підключений до бази даних MySQL, яка працює на цьому екземплярі EC2 (рис. 3.1).



```
mysql - "ip-10-0-0-140" x (+)
voclabs:~/environment/db $ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

Рисунок 3.1 – Екран терміналу

Щоб переглянути вміст бази даних (зокрема, таблиці, які підтримують веб-програму café), введено такі команди:

```
show databases;
use cafe_db;
show tables;
select * from product;
exit;
```

На кроці 24 оновлено конфігурацію часового поясу в PHP. (Update the timezone configuration in PHP).

У терміналі Bash виконано такі команди:

```
sudo sed -i "2i date.timezone = \"America/New_York\" " /etc/php.ini
sudo service httpd restart
```

Перша команда, налаштувала часовий пояс у програмному забезпеченні PHP.

Друга команда, перезапустила веб-сервер, щоб веб-сервер помітив оновлення конфігурації.

На 25-му кроці перевірено, чи веб-сайт кав'ярні працює та чи можна отримати до нього доступ з Інтернету.

У новій вкладці браузера завантажено програму за адресою `http://<public-ip>/cafe`, де `<public-ip>` — загальнодоступна IP-адреса IPv4 примірника EC2.

Але завантажується лише заголовний банер веб-сайту. Решта веб-сторінки не завантажується належним чином.

Крок 26 вирішив проблему з веб-сайтом.

На цьому кроці вам потрібно було з'ясувати, як зробити так, щоб веб-сайт кафе працював правильно.

Ось список того, що працює:

Тестова сторінка за адресою `http://<public-ip>/loads`, відомо, що веб-сервер працює та доступний в Інтернеті

Також відомо, що база даних MySQL запущена та містить таблиці та дані для підтримки програми

У чому може бути проблема?

Отримано наступні відповіді.

Немає нічого поганого в написаному коді PHP. Проблема пов'язана з дозволами.

Тоді виникають наступні запитання.

Чи залежить веб-програма від будь-яких інших служб або ресурсів, крім веб-сервера та бази даних? Наприклад, чи потрібен доступ до іншого сервісу AWS?

Виконано перехід до служби IAM і знайдено роль IAM під назвою `CafeRole`. Зафіксовано дозволи, які надає ця роль. Чи було б корисно надати ці права примірнику EC2? Як можна надати ці дозволи веб-програмі `café`?

Вибрано примірник EC2 на консолі Amazon EC2 і переглянуто параметри в меню «Дії» `Actions`. Чи допомогли б якісь дії вирішити проблему?

Коли проблему вирішено, знову завантажено сторінку `http://<public-ip>/cafe`. Вона повністю завантажується, щоб можна бачити пункти меню кафе.

3.2 Тестування веб-додатку

Для тестування веб-додатку (завдання 5) виконано наступний крок.

Крок 27 перевіряє оформлення замовлення.

На вкладці браузера, де відкрито сторінку <http://<public-ip>/cafe>, вибрано «Меню » Menu.

Надіслано замовлення принаймні на один із показаних пунктів меню.

Щоб знайти кнопку «Надіслати замовлення» Submit Order необхідно прокрутити вниз.

Після повернення на сторінку меню та розміщення іншого замовлення було виконано перехід на сторінку Order History «Історія замовлень», щоб переглянути деталі всіх замовлень, які було розміщено.

3 СТВОРЕННЯ ВЕБ-САЙТІВ ДЛЯ РОЗРОБКИ ТА ВИРОБНИЦТВА В РІЗНИХ РЕГІОНАХ AWS

Усі в кафе вражені новим динамічним веб-сайтом, який створила Софія! Клієнти в захваті від того, що тепер вони можуть розміщувати онлайн-замовлення та планувати вивезення десертів. Задоволеність клієнтів зросла завдяки скороченню часу очікування.

Однак, крім похвали, виникає ще одна вимога до бізнесу (проблема 3). Марта та Френк хотіли б мати два веб-сайти кафе:

- Один веб-сайт, який можна використовувати як середовище розробки для моделювання нових функцій і веб-дизайну, перш ніж вони будуть доступні клієнтам
- Окремий веб-сайт, на якому розміщено виробниче середовище, яким користуються клієнти

Софія обговорила нову вимогу з Матео, коли він одного ранку зайшов у кафе випити кави. Він припустив, що в ідеалі ці два середовища повинні існувати в різних регіонах AWS. Така конструкція матиме додаткову перевагу, забезпечуючи більш надійне аварійне відновлення (disaster recovery - DR) у малоймовірному сценарії, коли регіон AWS стає тимчасово недоступним.

Софія зараз дуже зайнята! Коли вона виконує більш вражаючу роботу, її навички стають більш затребуваними.

3.3 Створення АМІ та запуск іншого екземпляра EC2

Оскільки веб-сайт кафе вже добре працює на існуючому екземплярі EC2, Софія вирішує дублювати його, створивши з нього АМІ. Потім вона запустить новий екземпляр із нового АМІ (завдання 6).

Дослідження присвячене розробці методики виконання ролі Софії для виконання цього завдання. Перш ніж створити АМІ з цього екземпляра, слід створити нову пару ключів, яку може бути важливо мати пізніше в цій дослідницькій роботі.

На кроці 28 встановлене статичне внутрішнє ім'я хоста та створена нова пара ключів на примірнику EC2.

У терміналі bash виконано такі команди:

```
sudo hostname cafeserver
```

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa
```

Два рази, коли вам було запропоновано ввести парольну фразу, виконано натиснення клавіши ENTER.

Щоб зробити новий ключ доступним для утиліт SSH, у терміналі Bash виконано таку команду:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

У консолі керування AWS (крок 29) виконано перехід до зони обслуговування EC2 і вибрано екземпляр.

На 30-му кроці вибрано Actions > Images and templates > Create Image.

Діалогове вікно «Створити зображення» Create Image було залишено відкритим на вкладці браузера, поки формувалися відповіді на запитання про АМІ.

Відповіді на питання про АМІs (кроки 31, 32).

Запитання 5: коли створюється АМІ з примірника, чи буде він перезавантажений? Відповідь – так завжди.

Запитання 6: як можна змінити властивості кореневого тому під час створення АМІ з примірника? Відповідь – You can edit the size and volume type, but not the 'delete on termination' setting.

Запитання 7: чи можете ви додати більше томів до АМІ, створеного з примірника, який має лише один том? Відповідь – так.

У консолі керування AWS, на екрані «Створити образ» Create Image створено новий АМІ (крок 33):

- Image name: CafeServer;
- Виберіть Create Image.

У навігаційному меню на 34-му кроці вибрано АМІs і виконано очікування, поки статус зображення став доступним Available. Процес зазвичай займає близько 2 хвилин. Краще розгорнути зображення Images, щоб знайти АМІ.

На 35-му кроці створено АМІ в іншому регіоні AWS.

На цьому кроці мета — створити новий екземпляр EC2 з АМІ, який був захоплений. Однак необхідно створити новий екземпляр у регіоні AWS Орегона Oregon (us-west-2).

Скопійовано ідентифікатор АМІ створеного зображення. Спроба знайти його в регіоні us-west-2 не дала результат.

Чи існують АМІ на рівні облікового запису?

Було вибрано АМІ, який був створений перед цим в іншому регіоні AWS. Далі вибрано меню Дії Actions. Чи існують якісь дії можуть допомогти зробити АМІ доступним у Західному регіоні США (Орегон)? Вибрано відповідну дію. Після того, як вона була ініційована, для виконання знадобилось 5 хвилин. Час від часу виконувалось натискання піктограми оновлення, щоб швидше знати, коли воно завершиться.

На кроці 36 створено новий екземпляр café з створеного АМІ. Новий екземпляр, відповідає наступним критеріям.

- Region: Oregon
- Instance Size: t2.small
- Network: Lab VPC Region 2, Public Subnet
- IAM Role: CafeRole
- Tag:
 - Key: Name

- Value: ProdCafeServer
- Security Group:
 - створена нова з іменем safeSG, з TCP-портом 22, відкритим будь-де
 - встановлено TCP-порт 80 так, щоб він також був відкритим будь-де
- продовжено без пари ключів Proceed without a key pair (пара ключів, яка була створена раніше в цій дослідницькій роботі, має працювати для підключення до неї, якщо це необхідно)

Виконано очікування (крок 37), доки новому екземпляру було призначено значення Public DNS, навіть якщо статус екземпляра все ще був недоступний not Available.

На 38-му кроці було скопійовано значення Public DNS.

На 38-му кроці створені необхідні параметри AWS Systems Manager parameters у новому регіоні AWS, тобто виконано кроки.

Виконано перехід до AWS Cloud9 IDE у регіоні Північна N. Virginia (us-east-1).

Відкрито файл CafeWebServer/setup/set-app-parameters.sh у текстовому редакторі.

Відредаговано рядок 12 файлу відповідно до цього параметра:

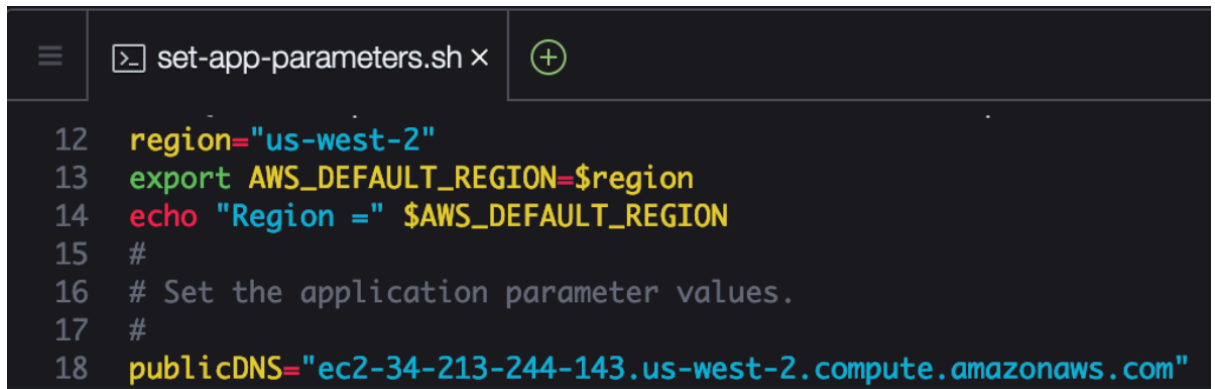
```
region="us-west-2"
```

Відредаговано рядок 18 відповідно до цього параметра (де <public-dns-of-ProdCafeServer-instance> є фактичним DNS примірника ProdCafeServer):

```
publicDNS="<public-dns-of-ProdCafeServer-instance>"
```

Рядок все ще має містити лапки, але не має містити кутові дужки (< >).

Цей рис.4.1 показує, як має виглядати рядок 12 і як має бути відформатований рядок 18. Однак значення загальнодоступного DNS може бути іншим.



```
12 region="us-west-2"
13 export AWS_DEFAULT_REGION=$region
14 echo "Region =" $AWS_DEFAULT_REGION
15 #
16 # Set the application parameter values.
17 #
18 publicDNS="ec2-34-213-244-143.us-west-2.compute.amazonaws.com"
```

Рисунок 3.1 – Рядок 12

Зміни збережені File > Save.

Для запуску цього сценарію, виконано перехід у верхній частині IDE та вибрано кнопку «Виконати» Run.

У терміналі Bash під текстовим редактором з'явився вихід, відформатований у нотації об'єктів JavaScript (JavaScript Object Notation -JSON). Цей вихід вказує на те, що сценарій параметрів виконано успішно.

Якщо вихідні дані команди вказують на те, що термін дії облікових даних минув, можна спробувати запусити ці команди в терміналі Bash (а потім знову запусити set-app-parameters.sh):.

```
wget      https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-
ACACAD-20-EN/mod4-challenge/cred-update.sh
chmod +x cred-update.sh
./cred-update.sh
```

Змінівши деталі регіону AWS і повторно запусивши цей сценарій, було створено ті самі параметри, які були створені раніше в регіоні us-east-1 у сховищі параметрів AWS Systems Manager. Однак цього разу ці параметри створені в регіоні Орегон.

3.4 Перевірка нового екземпляра café

Для виконання завдання 7 виконані наступні кроки.

На кроці 40 виконано повернення до консолі EC2 у регіоні Орегон і підтверджено, що новий екземпляр ProdCafeServer працює.

Далі скопійовано загальнодоступну IP-адресу IPv4 і завантажено її у веб-браузер (крок 41).

Відобразилося повідомлення The Hello from the cafe web server!

Завантажено URL-адресу `http://<public-ip>/cafe/` на вкладці браузера (крок 42).

Відображається весь веб-сайт кафе.

На 43-му кроці завантажено сторінку меню Menu.

Завантажилася повна сторінка меню Menu, а функція розміщення замовлення працює.

На 44-му кроці було розміщено замовлення, щоб перевірити, чи веб-сайт працює належним чином.

Може ще знадобитися крок 45 у разі виникнення несправності.

Можна підключитися до нового екземпляра EC2 в Орегоні (us-west-2) для усунення несправності та запустити цю команду з AWS Cloud9 IDE в us-east-1:

```
ssh -i ~/.ssh/id_rsa ec2-user@<public-ip-of-ProdCafeServer>
```

Важливо що `<public-ip-of-ProdCafeServer>` є фактичною публічною IP-адресою екземпляра ProdCafeServer.

ВИСНОВКИ

Софія тепер герой у кафе! Вона створила динамічний веб-сайт, а також дублікат того самого веб-сайту, який працює в другому регіоні AWS.

Софія вирішила призначити перший екземпляр EC2, який вона створила — той, що знаходиться в регіоні us-east-1 — як екземпляр розробки. Другий примірник, який вона створила — той, що знаходиться в Орегоні (регіон us-west-2) — буде робочим примірником.

Таким чином, Софія та будь-які інші розробники додатків можуть тестувати вдосконалення додатків на сайті розробки, не впливаючи на робочий сайт. Потім, коли розробники вирішать, що вдосконалення виглядають добре і вони повністю їх перевірили, вони зможуть перенести код на робочий сайт.

Софія пояснила своїм батькам, Френку та Марті, що вона зробила. Хоча вони не повністю розуміли всі технічні терміни, які використовувала Софія, вони були раді дізнатися, що веб-сайт тепер може приймати онлайн-замовлення. Вони також були раді почути, що тепер вони можуть тестувати нові вдосконалення веб-сайту, не повідомляючи про ці зміни клієнтам.

СПИСОК ЛІТЕРАТУРИ

- [1] AWS academy [Електронний ресурс], «Set up demo,» - 2024. - режим доступу до ресурсу: https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-100-ACFNDS-20-EN/Module_6_EC2+v2.0.mp4.
- [2] AWS CLI Command Reference [Електронний ресурс] – Режим доступу до ресурсу:
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/index.html>
- [3] Gobinda G. Chowdhury. “Natural Language Processing”. In: Annual Review of Information Science and Technology, 2003. – С. 51-89.
- [4] J. Le. (2018). “The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I)” [Online]. Available: <https://heartbeat.fritz.ai/the-7-nlp-techniques-that-will-change-how-youcommunicate-in-the-future-part-i-f0114b2f0497/>
- [5] Ted Briscoe. Introduction to Linguistics for Natural Language Processing. Tech. rep. 2013, pp. 1–37.
- [6] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu. Deep Learning Based Text Classification: A Comprehensive Review // 2005.
- [7] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.
- [8] Jurafsky D, Martin J H (2002) Speech and Language Processing – An Intro to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education Asia, ISBN 81-7808-594-1
- [9] R. Socher, “Recursive deep learning for natural language processing and computer vision” Stanford University, 2014, pp. 8-120.
- [10] Daniel Jurafsky. Classification: Naive Bayes, Logistic Regression, Sentiment / Daniel Jurafsky, James H. Martin // Speech and Language Processing / Daniel

- Jurafsky, James H. Martin., 2015. – С. 1–28.
- [11] Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. – СПб.: Питер, 2019. – 368 с.
- [12] Хобсон Лейн, Ханнес Хапке, Коул Ховард Обработка естественного языка в действии. – СПб.: Питер, 2020. – 576 с
- [13] Manaswi N.K. (2018) Developing Chatbots. In: Deep Learning with Applications Using Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3516-4_11
- [14] What is Amazon Transcribe? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/transcribe/latest/dg/transcribe-what-is.html>
- [15] Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных: навч. посіб. – М.: НИУ ВШЭ, 2017. – 269 с.
- [16] What is Amazon Polly? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/polly/latest/dg/what-is.html>
- [17] What is Amazon Translate? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/translate/latest/dg/what-is.html>
- [18] Klopfenstein, L., Delpriori, S., Malatini, S., Bogliolo, A.: The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In: Proceedings of the 2017 Conference on Designing Interactive Systems, pp. 555–565. Association for Computing Machinery (2017).
- [19] Poibeau T., Saggion H., Piskorski J., Yangarber R. Multi-source, Multilingual Information Extraction and Summarization. Theory and Applications of Natural Language Processing. Springer, Berlin, Heidelberg, 2013. – 257 с.
- [20] What is Amazon Comprehend? [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html>
- [21] M. Bates (1995). “Models of natural language understanding” [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC40721/>

- [22] Abdul-Kader, S., & Woods, J. (2015). Survey on Chatbot Design Techniques in Speech Conversation Systems. *International Journal of Advanced Computer Science and Applications*, 6(7). <http://doi.org/10.14569/ijacsa.2015.060712>
- [23] J. Chai and J.Lin, “The role of natural language conversational interface in online sales: a case study,” *International Journal of Speech Technology.*, vol. 4, pp. 285–295, Nov. 2001.