МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Інформаційна технологія інтеграції моделей в системі підтримки прийняття рішень при управлінні гібридною енергомережою

Здобувача групи __ІТм-24__ _____Заіка Юрій Сергійович_____

(шифр групи) (прізвище, ім'я, по батькові)

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

_____ Юрій ЗАЇКА

(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник професор кафедри ІТ, д.т.н., доцент Тимчук С.О. _____

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2023

Сумський державний університет

Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО
«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентові

Заіка Юрій Сергійович

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Інформаційна технологія інтеграції моделей в системі підтримки прийняття рішень при управлінні гібридною енергомережою

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «_15_» ___грудня__ 2023 р.

3 Вхідні дані до кваліфікаційної роботи перелік вимог на розробку програмного модуля активації прогнозних моделей для системи підтримки прийняття рішень при управлінні гібридною енергомережо

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

аналіз предметної області, постановка задачі та методи дослідження, проектування інформаційної технології інтеграції моделей в системі підтримки прийняття рішень при управлінні гібридною енергомережою, практична реалізація технології інтеграції

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) Мета і задачі, аналіз існуючих методів активації прогнозних моделей в системах управління енергомережами, порівняльна таблиця систем прогнозування енергопотреб, аналіз алгоритмів вибору технологій, порівняльна таблиця алгоритмів інтеграцій, функціональні вимоги до технології інтеграції, інструменти реалізації технології, контекстна діаграма процесу інтеграції моделей, декомпозиція діаграми процесу інтеграції моделей, діаграма варіантів використання технології інтеграції, діаграма послідовності інтеграції моделей, діаграма послідовності інтеграції даних для прогнозування виробництва, діаграма послідовності інтеграції з зовнішніми даними, реалізація технології, демонстрація роботи технології, висновки.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Формування технічного завдання	13.10.23 – 18.10.23	
2	Аналіз ринку	18.10.23 – 20.10.23	
3	Розробка функціональних вимог	20.10.23 – 28.10.23	
4	Проектування технології інтеграції	28.10.23 – 12.11.23	
5	Розробка технології інтеграції	12.11.23 – 22.11.23	
6	Інтеграція моделей та даних	22.11.23 – 30.11.23	
7	Тестування технології інтеграції	30.11.23 – 07.12.23	
8	Розгортання модуля	07.12.23 – 09.12.23	
9	Технічна документація	09.12.23 – 13.12.23	
10	Завершення та аналіз проекту	13.12.23 – 15.12.23	

Магістрант _____ **Юрій ЗАІКА**

Керівник роботи _____ **д.т.н., доц. Сергій ТИМЧУК**

АНОТАЦІЯ

Кваліфікаційна робота магістра на тему «Інформаційна технологія інтеграції моделей в системі підтримки прийняття рішень при управлінні гібридною енергомережею» є актуальною через зростаючу потребу в ефективному управлінні ресурсами в енергетичній галузі. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел на 41 найменувань, та додатків, загальним обсягом 66 сторінок, з яких 35 сторінок становить основний текст, 6 сторінки списку використаних джерел, 6 сторінок додатків.

Метою дослідження є розробка технології інтеграції даних та моделей, які використовуються в системі підтримки прийняття рішень для управління гібридними енергомережами. Робота включає аналіз існуючих методів, розробку концептуальної моделі, проектування та реалізацію технології. Особлива увага приділяється автоматизації процесу інтеграції, що дозволяє забезпечити точність і ефективність прогнозування.

Одним із ключових результатів є створення надійної системи, здатної ефективно збирати, аналізувати та використовувати дані для управління енергомережами. Розробка має практичне значення для сектору енергетики, оскільки сприяє підвищенню ефективності управління ресурсами та підтримці стабільності енергосистем.

Щодо впровадження, розроблена технологія може бути інтегрована у різні системи управління енергомережами, що відкриває шлях для подальшого дослідження та розширення функціональності.

Ключові слова: інтеграція, енергетика, енергомережа, модель, моделі, дані, система підтримки прийняття рішень, гібридні системи, управління ресурсами, стабільність енергосистем.

ЗМІСТ

ВСТУП

З початку промислової революції людство інтенсивно використовувало викопне паливо, що стало основою для енергетики, транспорту та промисловості. Незважаючи на те, що це забезпечило значний розвиток та комфорт, одночасно призвело до безпрецедентного викиду парникових газів, що стало причиною глобальних кліматичних змін.

У відповідь на ці виклики, світова спільнота шукає шляхи зменшення викидів. Оскільки енергетичний сектор відповідає за значну частку глобальних викидів, відновлювані джерела енергії виходять на передній план як ключовий елемент у боротьбі з кліматичними змінами. Джерела сонячної та вітрової енергія, не викидають парникових газів, тому їх використання може зіграти вирішальну роль у запобіганні найгіршим наслідком зміни клімату. В той же час, через природне походження виробництво енергії з цих джерел не є добре контрольованими, і зі збільшенням їх частки в енергетичному балансі зростатимуть і проблеми, пов’язані з балансуванням споживання, виробництва, та прогнозування.

Контролювати ці процеси в реальному часі неможливо без використання сучасних методів інформаційних технологій. Енергетичні системи з відновлюваними джерелами енергії повинні стати більш “розумними”, тобто здатними реагувати на зміни оточуючого середовища, зміни в потребі енергозабезпечення, дотримання енергетичного балансу. Дані та моделі прогнозування рівня генерації, чи потреб споживання, повинні працювати у реальному часі. Тому питання інтеграції даних і моделей інформаційної підтримки прийняття рішень є досить актуальною задачею.

Тому, ця кваліфікаційна робота магістра спрямована на розробку інформаційної технології інтеграції моделей в системі підтримки прийняття рішень (СППР) при управлінні гібридною енергомережею.

Об’єкт дослідження: інтеграція моделей в системі підтримки прийняття рішень при управлінні гібридною енергомережею.

Предмет дослідження: методи та засоби інтеграції моделей у системі підтримки прийняття рішень при управлінні гібридною енергомережею

Мета: розробка інформаційної технології інтеграції моделей, яка буде відповідати наступним вимогам:

- Ефективність - інформаційна технологія повинна забезпечувати ефективну взаємодію різних моделей, що використовуються в СППР управління гібридною енергомережею.
- Надійність - інформаційна технологія повинна бути надійною та стійкою до помилок, відмов та втручань.
- Масштабованість - інформаційна технологія повинна бути масштабованою, щоб її можна було використовувати для управління великими гібридними енергомережами чи кластерами мікромереж.

Для досягнення цієї мети в роботі будуть вирішені такі завдання:

- Аналіз об'єкта дослідження;
- Аналіз існуючих методів інтеграції в інформаційних системах;
- Розробка методів та засобів інтеграції моделей;
- Оцінка ефективності інформаційної технології інтеграції моделей.

Результатами даної роботи буде інформаційна технологія інтеграції моделей, яка може бути використана, як при управлінні новими гібридними енергомережами, так і для підвищення ефективності управління існуючих енергомереж.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Актуальність

Енергетичний сектор зазнає значної трансформації, переходячи від викопного палива до відновлюваних джерел. Гібридні енергомережі, що об'єднують кілька джерел відновлюваної енергії, таких як сонячна, вітрова та інша, з традиційними системами генерації та зберігання, стають ключовими гравцями в цьому процесі [1]. Однак управління цими складними системами вимагає надійних та інтелектуальних інструментів для забезпечення оптимальної роботи всіх компонентів енергомережі та дотримання енергетичного балансу. Саме тут у гру вступають системи підтримки прийняття рішень (СППР) [2]. Ці системи забезпечують підтримку інформацією особу, яка приймає рішення щодо управління гібридною електромережею, і є багатогранним інструментом, що має вирішальне значення для збалансування мінливості відновлюваних джерел енергії зі стабільністю традиційних методів виробництва електроенергії [3]. Такі системи поєднують у собі велику кількість підсистем та забезпечують постійний моніторинг, прогнозування, оптимізацію виробництва і розподілу енергії з урахуванням таких факторів, як погодні умови, попит на енергію та стан обладнання. Безперебійний потік даних, інформації та знань у СППР для управління гібридною електромережею є запорукою її ефективності, надійності та успішності всього проекту [4].

У загальному вигляді СППР включають в себе підсистеми, що забезпечують:

- оперативний збір даних і попередня обробка;
- моделювання та імітацію процесів на різних рівнях функціонування гібридної енергомережі у будь які інтервали часу, а також у реальному часі;
- оптимізацію процесів функціонування та прийняття рішень щодо управління гібридною енергомережею;

- інтерфейс користувача з розподілом прав доступу до інформації та візуалізацією поточного стану енергомережі та її прогнозованих станів;
- комунікацію між всіма компонентами сппр та інтеграцію даних та моделей, що забезпечують інформаційну підтримку.
- безпека зберігання інформації та аудит поточного стану безпеки, детекція можливих втручань та вразливостей [5].

Тож постає проблема інтеграції даних та моделей СППР у спільному безпечному середовищі, що забезпечує ефективний обмін даними.

1.2. Аналіз рівнів інтеграції

1.2.1 Інтеграція на рівні даних

На цьому етапі відбувається оперативний збір та попередня обробка різноманітних даних з різних джерел у гібридній мережі у реальному часі. Збираються дані про поточний стан відновлюваних джерел енергії (сонячні панелі, вітрові турбіни тощо), дані про виробництво енергії, та погодні умови, стан зовнішньої енергомережі, системи зберігання енергії (акумулятори, тощо, що надають дані про рівень накопиченої енергії та швидкість заряджання/розряджання), датчики "розумної" мережі (моніторинг стану мережі, включаючи напругу, струм, потік потужності та продуктивність обладнання) [6].

В цей етап також включено попередню обробку даних, таку як очищення, нормалізація та агрегація, забезпечуючи узгодженість та сумісність даних для подальшого аналізу.

Зберігання даних поділено на декілька рівнів зберігання. Оперативна база даних зберігає дані про поточний стан, які обробляються у реальному часі. Це дані моніторингу та результати оперативного прогнозування рівнів генерації та споживання. Центральне сховище даних зберігає та керує величезною кількістю даних, зібраних із різних джерел за попередні періоди [7]. Також це сховище даних

забезпечує, розподіл ролей користувачів, рівень доступності моделей, ефективний доступ до даних, пошук і аналіз даних, знань та моделей різними модулями СППР, задіяними в управлінні мережею.

Для оптимального спільного використання та обміну даними повинні існувати механізми, що сприяють безперебійному обміну даними між різними підсистемами, модулями, базами даних, сховищем даних та додатками, задіяними в управлінні гібридною мережею [8]. Це включає зв'язок із зовнішніми системами, такими як служби прогнозування погоди та енергетичні ринки, для ефективного прийняття рішень.

1.2.2 Інтеграція на рівні моделей

Цей рівень передбачає розробку різних моделей, адаптованих до конкретних потреб у гібридній мережі, зокрема:

- фізичні моделі: імітація електричної поведінки мережі, включаючи напругу, струм і потік потужності через різні лінії та трансформатори.
- моделі оптимізації: оптимізація диспетчеризації енергії з урахуванням таких факторів, як доступність відновлюваної енергії, попит на енергію та ємність зберігання.
- статистичні моделі та моделі машинного навчання: прогнозування попиту на енергію, прогнозування погодних умов і виявлення потенційних збоїв у мережі.

Інтеграція цих моделей полегшує обмін даними та співпрацю між різними моделями, надаючи повну інформацію для оптимізації роботи мережі [9].

Різним моделям часто потрібно взаємодіяти й обмінюватися даними для комплексного аналізу. Механізми з'єднання забезпечують безперебійну взаємодію, а стандарти сумісності полегшують зв'язок між моделями, розробленими на різних платформах [10].

Управління версіями моделей, управління ними та відстеження змін має вирішальне значення для забезпечення прозорості та підзвітності у прийнятті рішень. Це дозволяє користувачам зрозуміти еволюцію моделей та їхній вплив на рішення.

1.2.3 Інтеграція на рівні системи

Зручний інтерфейс дозволяє операторам ефективно взаємодіяти з СППР та отримувати доступ до її функцій. Інструменти візуалізації, такі як інформаційні панелі та діаграми, дають чітке уявлення про продуктивність мережі, що дозволяє приймати обґрунтовані рішення [12].

Управління робочими процесами та автоматизація є важливим етапом розробки. СППР має автоматизувати рутинні завдання та робочі процеси, пов'язані з управлінням мережею, звільняючи час операторів для зосередження на стратегічних рішеннях та аналізі. Це може включати автоматичне керування генерацією, управління реагуванням на попит та виявлення несправностей і реагування на них.

Надійні заходи безпеки захищають конфіденційні дані та забезпечують санкціонований доступ до СППР. Аутентифікація користувачів, шифрування даних та механізми контролю доступу мають вирішальне значення для підтримки цілісності системи та запобігання несанкціонованому доступу [13].

1.2.4 Інтеграція на рівні систем

СППР має безперешкодно інтегруватися з існуючими системами управління енергомережами, а також з системи комерційного обліку електроенергії. Така інтеграція забезпечує цілісне уявлення про енергетичні операції організації та полегшує прийняття рішень на основі даних на різних рівнях управління.

Механізми співпраці та обміну знаннями між різними зацікавленими сторонами, що беруть участь в управлінні енергосистемою, мають вирішальне значення для підвищення загальної ефективності. Вони можуть включати спільні інформаційні панелі, комунікаційні платформи та сховища знань для полегшення співпраці та передачі знань [14].

Постійний моніторинг та оцінка ефективності роботи системи підтримки прийняття рішень мають важливе значення для визначення сфер, які потребують вдосконалення, та забезпечення його довгострокової ефективності. Такі показники, як ефективність виробництва енергії, стабільність мережі та точність прийняття рішень, можна відстежувати, щоб оцінити роботу системи та визначити сфери для вдосконалення.

В загальному вигляді система підтримки прийняття рішень може мати такі компоненти (рисунок 1.1):

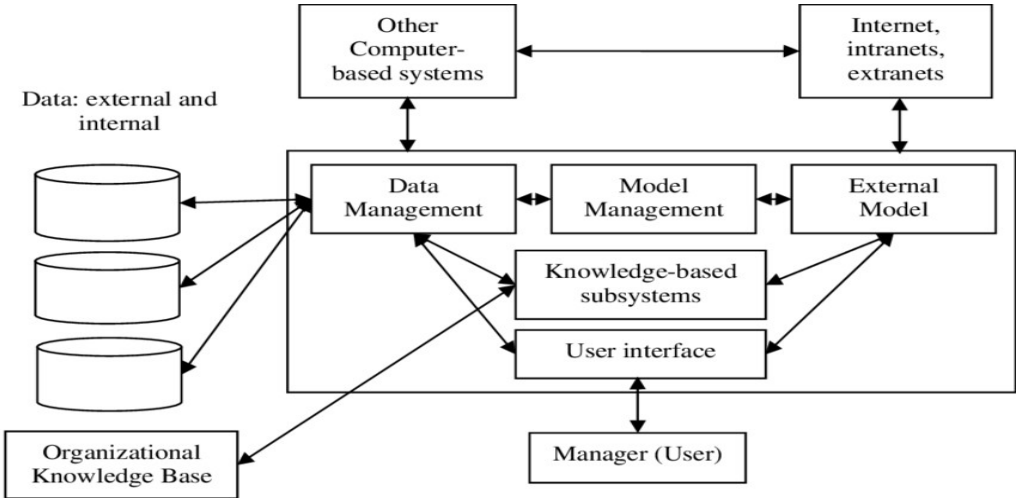

Рисунок 1.1 - Компоненти СППР [15]

В результаті роботи ці складні системи генерують величезні обсяги різноманітних даних, які необхідно обробляти та аналізувати для прийняття ефективних рішень щодо управління мережею [15].

У загальному вигляді можна показати входи та виходи системи (рисунок 1.2):

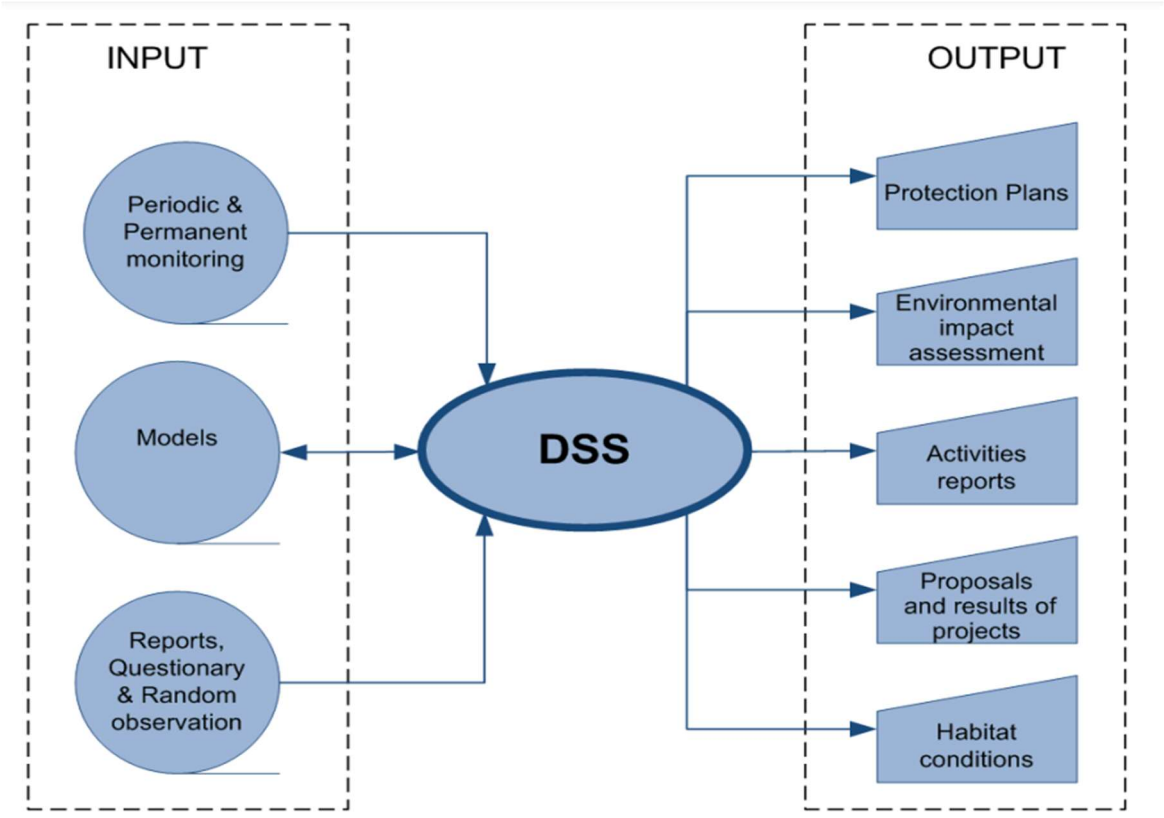

Рисунок 1.2 - Вхідні та вихідні дані СППР[15]

В ситуації, коли дані повинні бути передані крізь декілька підсистем, ключову роль починає грати концепція конвеєра даних. Конвеєр даних – це процес, в якому дані проходять довгий шлях від збору з багатьох джерел, включаючи бази даних, зовнішні статистичні та потокові дані, різні сенсори та інші, та до моменту їх використання в аналітичних і бізнес-процесах [16]. Протягом цього шляху дані піддаються різноманітним видам обробки: від очищення та стандартизації до інтеграції, агрегації та аналізу, що підвищує їх цінність і робить дані придатними для ефективного використання в ухвалених рішеннях.

Таким чином, конвеєр даних стає не просто механізмом передачі даних, але й інструментом їх перетворення, збагачення та підготовки до подальшого аналітичного

використання. У всіх цих задачах і викликах, концепція конвеєра даних набуває особливого значення у сучасному бізнесі, ставши ключовим елементом у стратегії управління.

Ретельно побудований конвеєр даних закладає фундамент системи, орієнтованої з їхньої обробку. Компанії можуть автоматизувати потік даних між системами, роблячи їх доступними для аналізу та прийняття рішень [17].

Конвеєр даних також стоїть у центрі багатьох передових технологій та бізнес-процесів. Наприклад, вони відіграють ключову роль у обробці даних у реальному часі та аналітиці, дозволяючи компаніям отримувати миттєві оцінки та приймати негайні рішення. Вони також потрібні для додатків машинного навчання та штучного інтелекту, яким потрібні великі обсяги даних високої якості для навчання моделей та алгоритмів [18].

В епоху цифровізації та інформаційних технологій конвеєри даних займають ключове місце в архітектурі сучасних інформаційних систем. Це не просто набір інструментів і процесів, а скоріше каркас, що забезпечує життєвий цикл даних від початку до кінцевого споживача.

Конвеєр даних є комплексною системою, завданням якої є обробка та передача даних від місць їх збору до точок аналізу та використання. Ця система включає в себе етапи збору, очищення, трансформації та завантаження даних. Вона забезпечує не тільки їхній фізичний потік, але й гарантує їхню якість, відповідність та готовність до подальшого аналітичного застосування [19].

Конвеєр даних знаходить своє застосування в різних областях від бізнес-аналітики, де він допомагає перетворювати зібрані дані в цінні інсайти, до машинного навчання і штучного інтелекту, де якісні і добре оброблені дані є ключем до успішного навчання моделей. Важливу роль data pipeline грає у обробці даних у реальному часі, що критично у таких сферах, як інтернет речей, фінансові послуги та логістика [20].

Розвиток конвеєрів даних стикається з низкою викликів, включаючи необхідність масштабування системи для обробки зростаючих обсягів даних,

забезпечення безпеки та конфіденційності інформації, а також відповідності нормативним вимогам. У той же час, перспективи розвитку цієї галузі пов'язані з інтеграцією нових технологій, таких як штучний інтелект та машинне навчання, для автоматизації та оптимізації процесів обробки даних.

Основою конвеєрів даних є джерела даних, які можуть бути різноманітними - від баз даних до потокових джерел (рисунок 1.3). Потім йде етап обробки даних, який включає їх очищення, нормалізацію та трансформацію. Кінцевими точками маршруту є системи зберігання, такі як Data Lakes і Data Warehouses, де дані стають доступними для аналітики і прийняття рішень. Загальний вигляд конвеєра даних представлено на рисунку 1.3 [21].

Рисунок 1.3 - Приклад конвеєру даних [21]

Інструментарій конвеєрів даних включає програмне забезпечення для ETL-процесів, таке як Apache NiFi, Talend і Informatica, які забезпечують структурування і підготовку даних (рисунок 1.4).

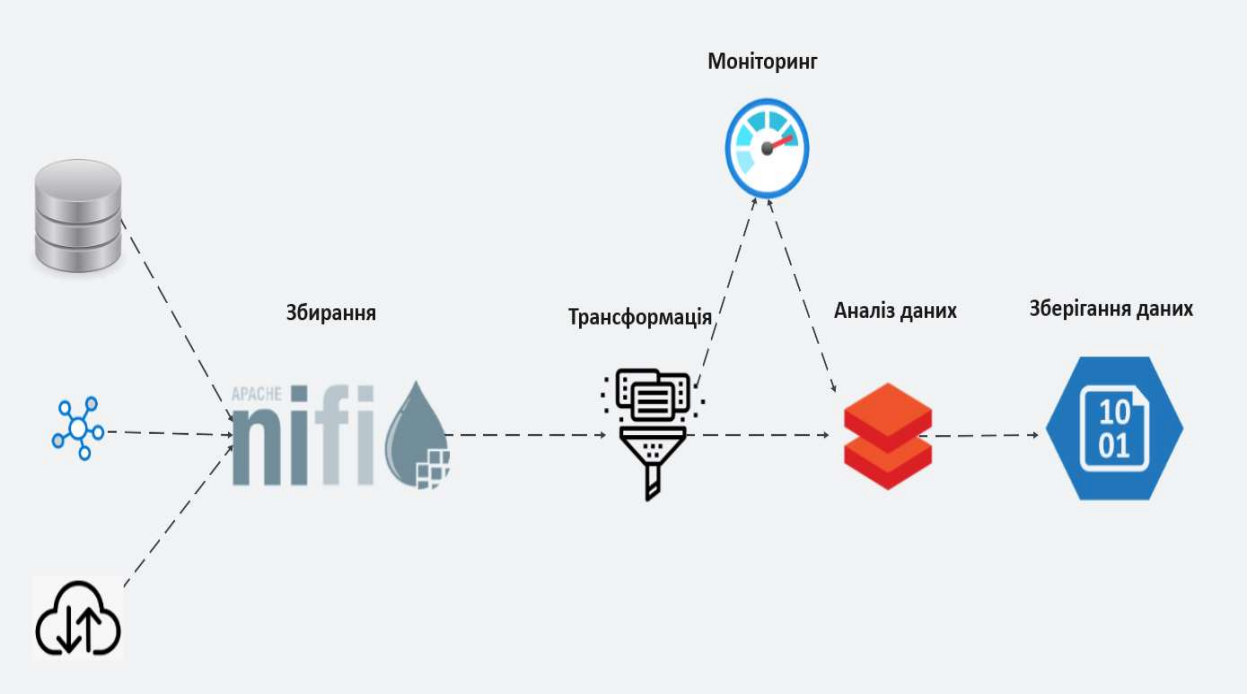

Рисунок 1.4 - Приклад використання інструментарію конвеєра даних[21]

Традиційні системи підтримки прийняття рішень часто не справляються з цими завданнями. Вони можуть бути занадто дорогими, складними в управлінні та не гнучкими.

Хмарні сервіси пропонують ряд переваг, які роблять їх ідеальним рішенням для СППР що використовуються у гібридних енергетичних мережах.

1.3. Хмарні сервіси

1.3.1 Аналіз переваг хмарних сервісів

Робота гібридних енергетичних мереж тягне за собою постійне зростання обсягу даних з різних джерел, таких як системи відновлюваної енергетики, прогноз генерації електроенергії та моделі споживання, в то же час хмарні провайдери оснащені передовими інструментами інтеграції даних, які можуть безперешкодно агрегувати та синхронізувати ці розрізнені дані. Таке уніфіковане представлення даних має вирішальне значення для прийняття обґрунтованих рішень в системах підтримки прийняття рішень [22]. Крім того, хмарні платформи забезпечують необхідну гнучкість і масштабованість для інтеграції різноманітних обчислювальних моделей, включаючи статистичне прогнозування, алгоритми оптимізації розподілу ресурсів і прогностичне обслуговування на основі штучного інтелекту. Масштабованість хмарної інфраструктури забезпечує адаптивність у міру зміни мережевих вимог і конфігурацій [23].

Ще однією ключовою перевагою реалізації СППР у хмарі є розширена співпраця та доступність, яку вона пропонує для різного роду стейкхолдерів. Завдяки даним і моделям, розміщеним у хмарі, зацікавлені сторони з різних областей або регіонів можуть отримати доступ до узгодженої інформації та інструментів, що сприяє прийняттю більш узгоджених і ефективних управлінських рішень [24]. Крім того, хмарні провайдери пропонують найсучасніші сервіси аналітики і машинного навчання, які можна інтегрувати в систему підтримки прийняття рішень. Ці передові технології мають вирішальне значення для аналізу складних моделей даних, прогнозування поведінки мережі та оптимізації розподілу енергії, підвищуючи таким чином ефективність і передбачуваність гібридних енергетичних мереж [25].

Крім того, використання хмарних провайдерів для СППР в гібридних енергетичних мережах є економічно вигідним та ефективним підходом. Він усуває

необхідність значних попередніх інвестицій в інфраструктуру та зменшує операційні витрати, пов'язані зі зберіганням, обробкою даних та управлінням моделями [26]. Моделі ціноутворення за принципом "оплата за фактом" забезпечують оптимізований розподіл ресурсів, оскільки мережі платять лише за ті послуги, якими вони користуються.

Надійність і безперервність роботи є критично важливими для енергетичних мереж, а хмарні сервіси відомі своєю високою надійністю і безвідмовністю. Це гарантує стабільну роботу СППР з мінімальним часом простою, що має важливе значення для енергетичного сектору. Окрім експлуатаційної надійності, першорядне значення мають безпека та відповідність нормативним вимогам [27]. Хмарні провайдери інвестують у надійні заходи безпеки, включаючи шифрування даних і надійний контроль доступу, а також стежать за змінами в законодавстві, забезпечуючи відповідність системи підтримки прийняття рішень найновішим стандартам.

Таким чином, хмарні провайдери пропонують надійне, масштабоване та ефективне рішення для інтеграції даних та моделей в управлінні гібридними енергетичними мережами. Використовуючи хмарні обчислення, СППР може значно підвищити операційну ефективність, надійність і можливості прийняття рішень в цих складних мережах [28]. А скільки глобальний енергетичний ландшафт продовжує зміщуватися в бік більш інтегрованих і стійких рішень, роль хмарних технологій в управлінні гібридними енергетичними мережами буде ставати все більш важливою.

1.3.2 Аналіз провідних хмарних сервісів

Оскільки існує дуже багато хмарних сервісі зупинимось на добре відомих світових флагманах. Платформи хмарних обчислень, такі як Amazon Web Services (AWS), Microsoft Azure і Google Cloud, пропонують ряд інструментів, які задовольняють специфічні потреби систем підтримки прийняття рішень [29].

Amazon Web Services (AWS)

AWS (рисунок 1.5), піонер хмарних обчислень, пропонує широкий спектр послуг, які можна використовувати для СППР в гібридних енергетичних мережах.

AWS IoT Core забезпечує безперебійне підключення та взаємодію з широким спектром пристроїв Інтернету речей, що має вирішальне значення для збору даних в гібридних мережах.

Amazon SageMaker забезпечує повне середовище для машинного навчання, що дозволяє проводити предиктивну аналітику, необхідну для прогнозування попиту та пропозиції енергії.

Такі сервіси, як Amazon S3 та DynamoDB, пропонують масштабовані та безпечні варіанти зберігання даних.

AWS Greengrass розширює можливості AWS на периферійні пристрої, дозволяючи їм працювати з даними локально, одночасно використовуючи хмару для управління, аналітики та зберігання.

Крім цього AWS пропонує потужний та багатофункціональний набір інструментів та сервісів для створення та управління конвеєрами, що робить її одним із провідних провайдерів хмарних рішень у галузі обробки та аналізу даних. Ці послуги охоплюють весь спектр завдань, пов'язаних з конвеєрами даних, від збору та зберігання даних до їх обробки та аналізу.

Microsoft Azure

Azure відома своїми хмарними сервісами, орієнтованими на підприємства, та потужними можливостями Інтернету речей, що робить її серйозним конкурентом для СППР у гібридних енергетичних мережах (рисунок 1.6).

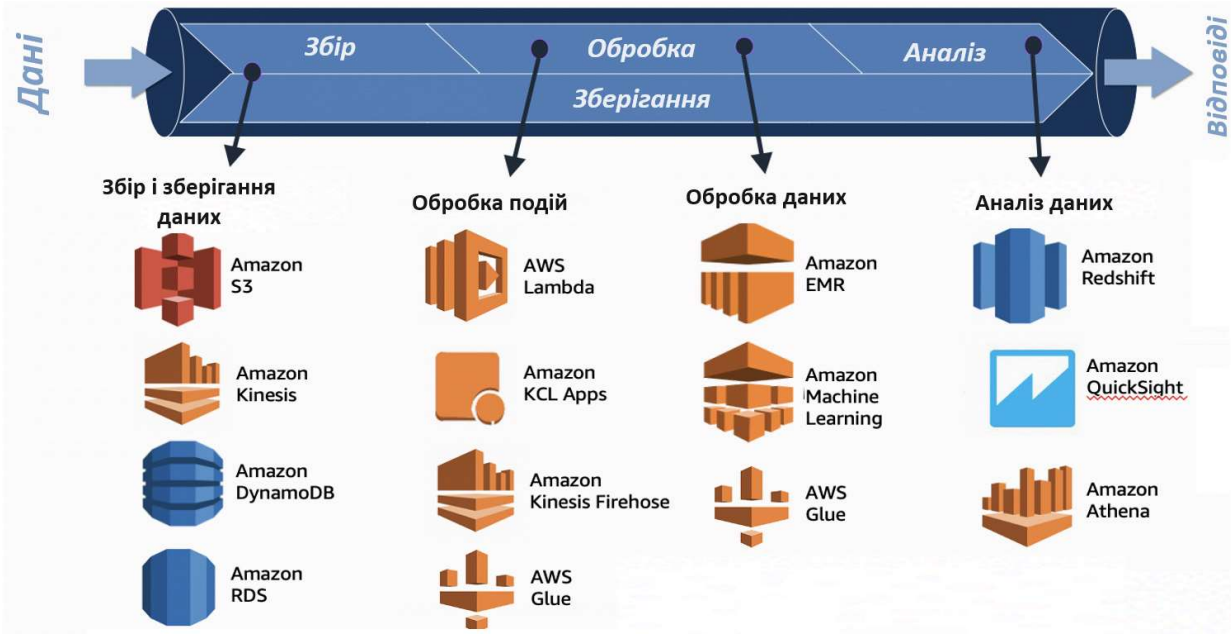

Рисунок 1.5 - Реалізація обробки даних у хмарному середовищі Amazon Web Services (AWS)[29]

IoT та периферійні рішення: Azure IoT Hub забезпечує центральний хаб для управління пристроями IoT, а Azure IoT Edge дозволяє запускати хмарні сервіси безпосередньо на пристроях IoT.

Штучний інтелект і аналітика: Azure Machine Learning та Azure Synapse Analytics підтримують створення, навчання та розгортання моделей машинного навчання в масштабах [30].

Azure пропонує надійні рішення для гібридних хмарних розгортань, що є вигідним для енергетичних мереж з локальними та хмарними компонентами.

Служби безпеки та відповідності вимогам від Azure є першокласними, забезпечуючи цілісність і захист даних у критично важливих об'єктах енергетичної інфраструктури.

Реалізація конвеєру даних від Microsoft - це набір сервісів та інструментів, що пропонуються в рамках хмарної платформи Azure, які допомагають у створенні, управлінні та оптимізації конвеєру даних. Ці інструменти та послуги полегшують процес збору, перетворення та передачі даних для різних бізнес-цілей [31].

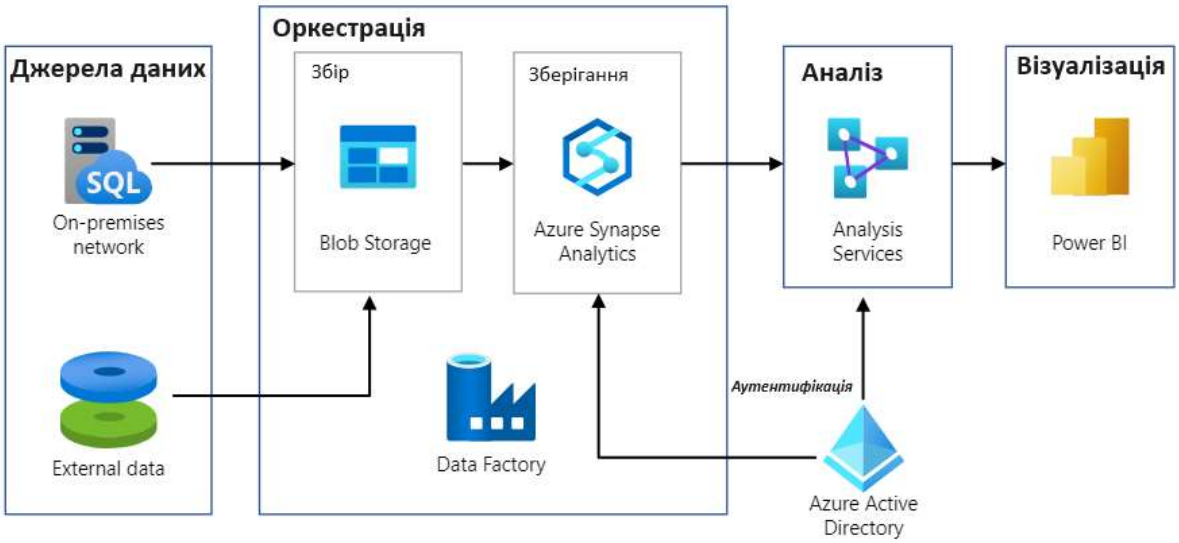

Рисунок 1.6 - Реалізація обробки даних у хмарному середовищі Microsoft Azure [31]

Google Cloud

Google Cloud (рисунок 1.7) відомий своїми сильними сторонами в аналітиці даних і машинному навчанні, пропонуючи інноваційні рішення для СППР в гібридних енергетичних мережах.

BigQuery, безсерверне, високомасштабоване та економічно ефективне мультихмарне сховище даних Google, ідеально підходить для великомасштабного аналізу даних.

TensorFlow та Google AI Platform надають передові інструменти для машинного навчання та штучного інтелекту.

Хмарне сховище Google пропонує масштабовані та гнучкі рішення для зберігання даних.

Google Cloud Dataflow надає аналітику в режимі реального часу та оперативну гнучкість, що є важливим для динамічних середовищ енергетичних мереж [32].

Як і інші хмарні сервіси Google Cloud також пропонує свій набір сервісів та інструментів для створення та управління конвеєрами даних, забезпечуючи потужні та гнучкі рішення для обробки та аналізу даних у хмарному середовищі, які можуть

використовуються для різних бізнес-завдань, включаючи аналіз даних, машинне навчання, дослідження, створення звітів і багато іншого. Це потужний інструмент для компаній, які прагнуть максимально використати потенціал своїх даних у хмарному середовищі [33].

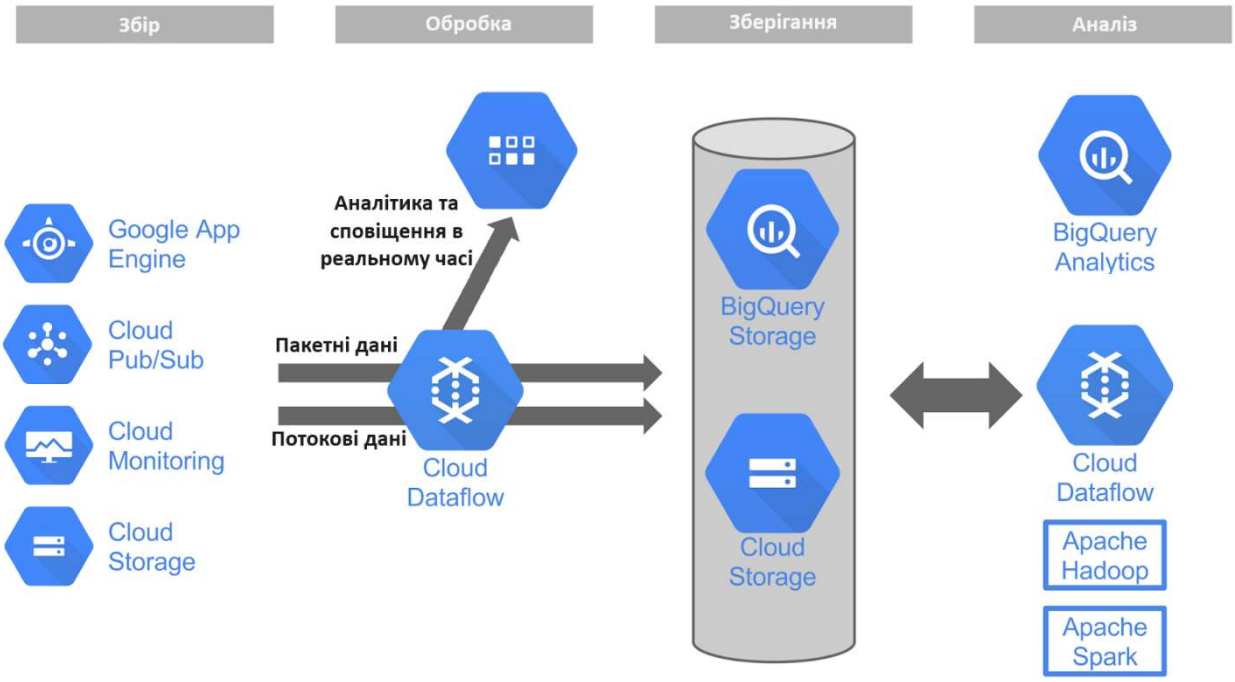

Рисунок 1.7 - Реалізація обробки даних у хмарному середовищі Google Cloud [33]

Проведемо порівняльний аналіз хмарних середовищ. Представимо результати порівняння у таблиці 1.1.

Вибір хмарного провайдера для СППР в гібридних енергетичних мережах залежить від конкретних вимог, таких як можливості Інтернету речей, аналітика даних, безпека та масштабованість.

AWS виділяється своїми комплексними послугами Інтернету речей та периферійних обчислень, які мають вирішальне значення для управління даними в режимі реального часу в гібридних енергетичних мережах[34].

Amazon SageMaker від AWS забезпечує надійну платформу для штучного інтелекту та машинного навчання, необхідну для предиктивної аналітики в управлінні енергією.

Її глобальна інфраструктура забезпечує високу доступність та надійність, що є ключовим фактором для безперебійної роботи енергетичних мереж.

Широкі можливості зберігання даних та масштабовані обчислювальні ресурси AWS можуть задовольнити потреби гібридних енергетичних мереж у величезних обсягах даних та їх обробці.

Крім того, прихильність AWS до безпеки та відповідності відповідає критичній потребі в захисті даних і дотриманні нормативних вимог в енергетичному секторі.

Таким чином, хоча Azure і Google Cloud пропонують величезні можливості, широкий набір сервісів AWS, особливо в області Інтернету речей, штучного інтелекту і глобальної інфраструктури, робить його особливо підходящим вибором для систем підтримки прийняття рішень в гібридних енергетичних мережах[35]. Здатність надавати комплексні, масштабовані та безпечні рішення позиціонує AWS як провідного конкурента в цій галузі.

Таблиця 1.1 - Порівняння хмарних середовищ

Функція	AWS	Microsoft Azure	Google Cloud
IoT та периферійні обчислення	AWS IoT Core та Greengrass для IoT та периферійних обчислень.	Azure IoT Hub та IoT Edge для інтегрованих рішень IoT.	Google Cloud IoT Core з менш комплексними периферійними рішеннями.
Аналітика даних та ШІ	Amazon SageMaker та інші інструменти для ШІ/ML та аналітики.	Azure Machine Learning та Synapse Analytics для AI/ML.	BigQuery та TensorFlow для розширеної аналітики та ML.
Рішення для зберігання даних	Масштабоване сховище з Amazon S3 і DynamoDB.	Azure Blob Storage та Table Storage для різноманітних потреб.	Google Cloud Storage та Bigtable для високопродуктивних потреб.
Масштабованість і гнучкість	Високомасштабована інфраструктура з Lambda для безсерверних обчислень.	Масштабовані рішення; Azure Kubernetes Service для оркестрування контейнерів.	Google Kubernetes Engine та масштабовані хмарні сервіси.
Безпека та відповідність вимогам	Комплексні функції безпеки та відповідності вимогам.	Особлива увага приділяється безпеці та відповідності вимогам, особливо для підприємств.	Просунуті інструменти безпеки з можливістю ручного налаштування.
Енергетичні рішення	Спеціалізовані послуги для енергетичного сектору.	Рішення, адаптовані для застосування в енергетичному секторі.	Менше уваги приділяється специфічним рішенням для енергетики.
Глобальна інфраструктура	Розгалужена глобальна мережа дата-центрів.	Широке глобальне охоплення з численними центрами обробки даних.	Глобальна інфраструктура, трохи поступається AWS та Azure.
Інтеграція та сумісність	Потужні можливості інтеграції з іншими службами AWS.	Безшовна інтеграція з іншими продуктами Microsoft.	Ефективна інтеграція з набором інструментів Google.
Ціноутворення та економічна ефективність	Конкурентоспроможне ціноутворення за моделлю "оплата за фактом".	Як правило, вища вартість, але пропонує економічно вигідні варіанти.	Вважається економічно вигідним для обробки та зберігання даних.
Підтримка та спільнота	Широка підтримка та велика спільнота розробників.	Широка мережа підтримки та орієнтація на великі підприємства.	Хороша підтримка та зростаюча спільнота аналітиків даних.

2. ПОСТАНОВКА ЗАДАЧІ

2.1 Мета та задачі дослідження

Метою кваліфікаційної роботи є дослідження технології інтеграції даних в систему прийняття рішень. У світлі актуальності проблеми, особлива увага приділяється доступності та простоті одержання інформації, забезпечуючи таким чином ефективнішу взаємодію між різноманітними підсистемами.

Для успішної реалізації потрібно зосередити увагу на всіх важливих аспектах, щоб основні функції були завжди легкодоступними та інтуїтивно зрозумілими для користувачів. Також буде використана база для збереження та оброблення даних параметрів мікромережі.

Представлено перелік функціоналу, що потрібно реалізувати при розробці:

1. Отримання результатів обробки зібраних даних
2. Перегляд інформації про роботу системи
3. Інформування про не відсутність синхронізації між підсистемами

Представлено перелік функціоналу, що потрібно реалізувати при розробці технології:

1. Адміністрування параметрів моделей та даних
2. Моніторинг проходження даних між підсистемами
3. Перегляд інформації про роботу процедур перетворення

Також перелік функціоналу обробки інформації:

1. Взаємодія між підсистемами
2. Отримання через певні проміжки часу наборів даних
3. Аналіз даних та очищення даних
4. Валідація результатів

3. ПРОЕКТУВАННЯ ІНТЕГРАЦІЇ МОДЕЛЕЙ В СИСТЕМІ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ УПРАВЛІННІ ГІБРИДНОЮ ЕНЕРГОМЕРЕЖОЮ

3.1. Структурно-функціональне моделювання процесу

Для системи поставлена конкретна задача – реалізувати інтеграцію моделей та даних з різних підсистем в систему підтримки прийняття рішень. Для побудови буде використано вхідні та вихідні дані, механізми аналізу та валідації, інструменті керування.

Вихідними даними запит на розгортання інформаційної системи:

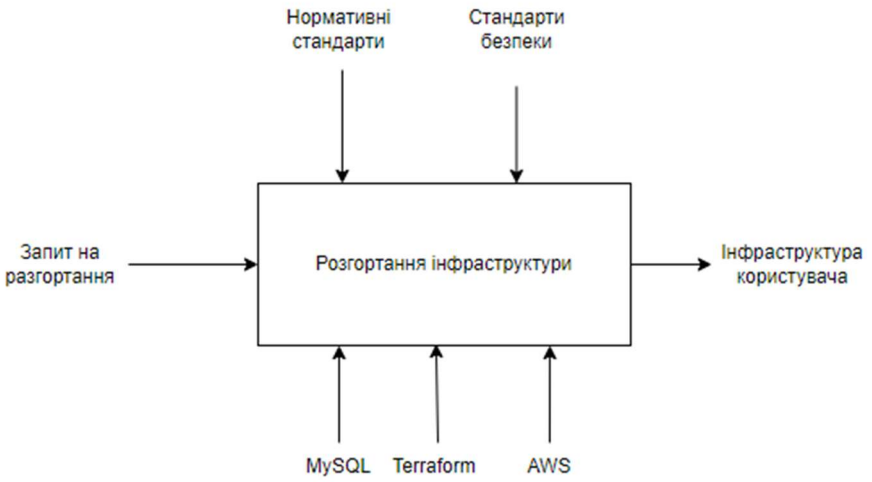

Рисунок 3.1 - Контекстна діаграма у нотації IDEF0

Джерело: побудовано автором

Виконали декомпозицію першого рівня (рис. 3.2).

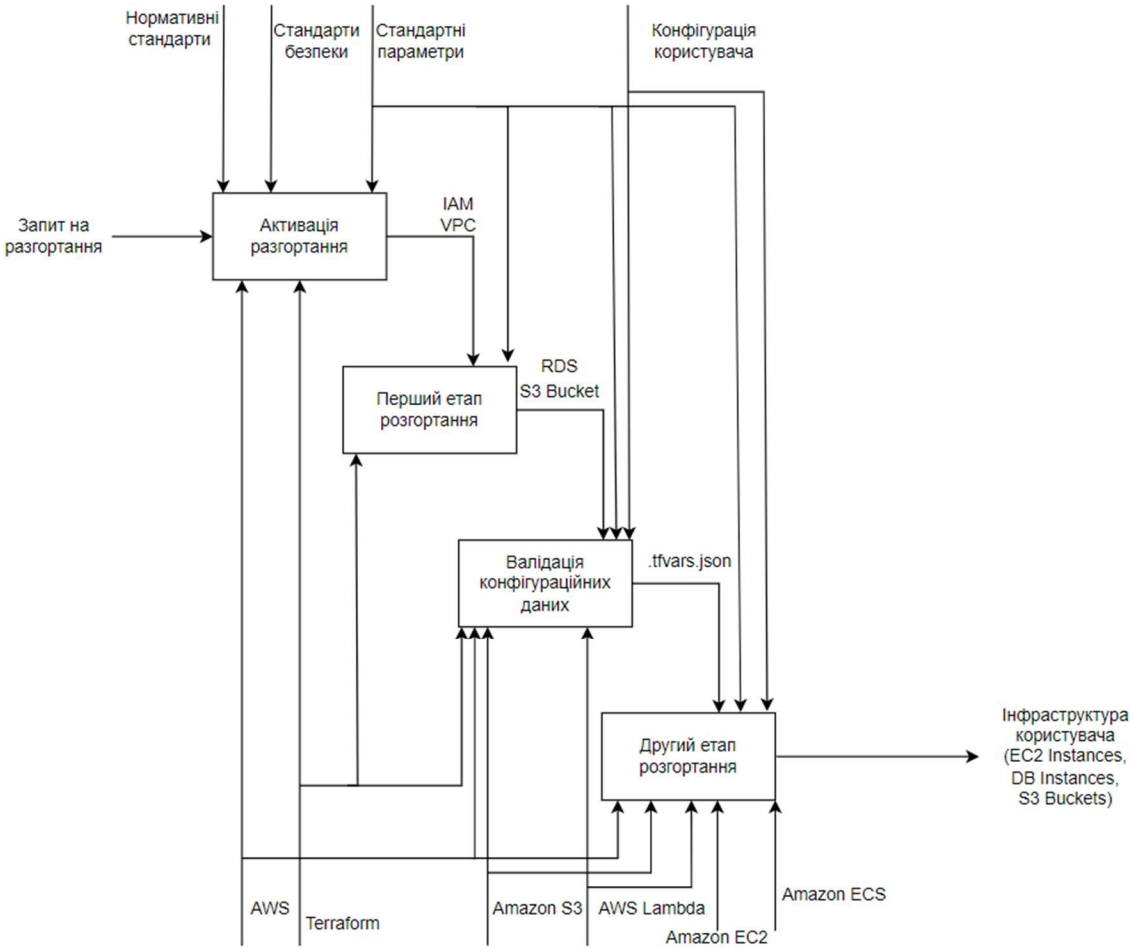

Рисунок 3.2 - Декомпозиція першого рівня

Джерело: побудовано автором

4. РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ІНТЕГРАЦІЇ

Інтеграція моделей та даних в систему підтримки прийняття рішень виконується методом розгортання необхідної інфраструктури в хмарному середовищі AWS з використанням інструментів Terraform, Amazon IAM, Amazon VPC, Amazon S3, AWS Lambda, Amazon EC2, Amazon ECS.

Побудуємо інтеграцію за наступною схемою:

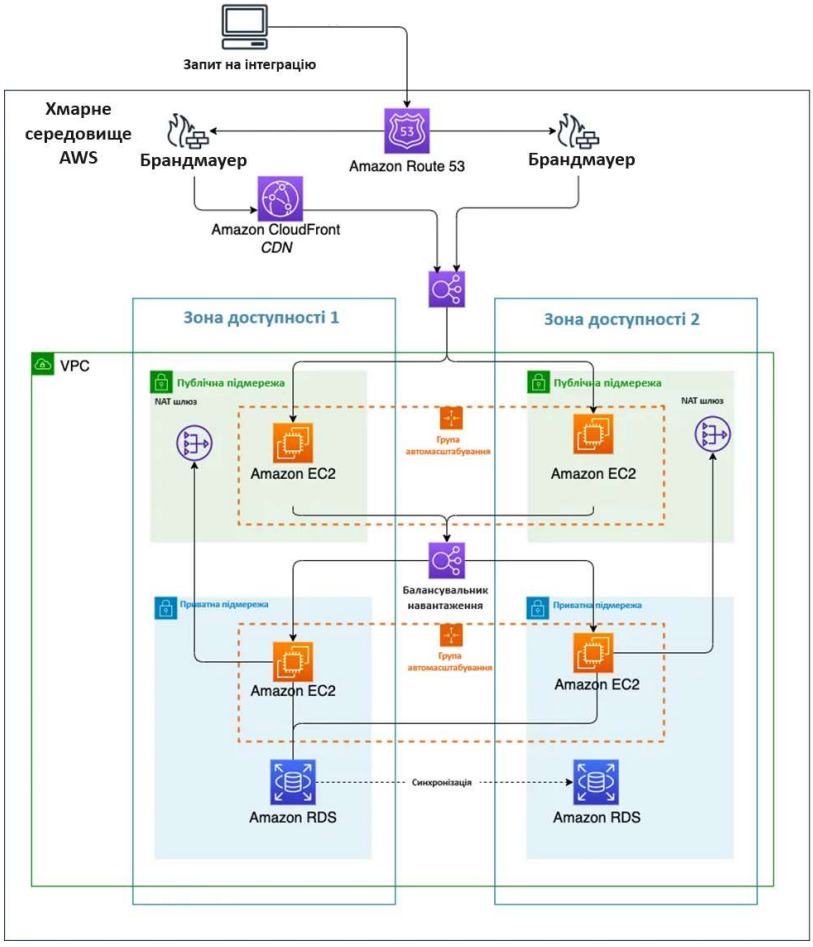

Рисунок 4.1 Схема інтеграції [45]

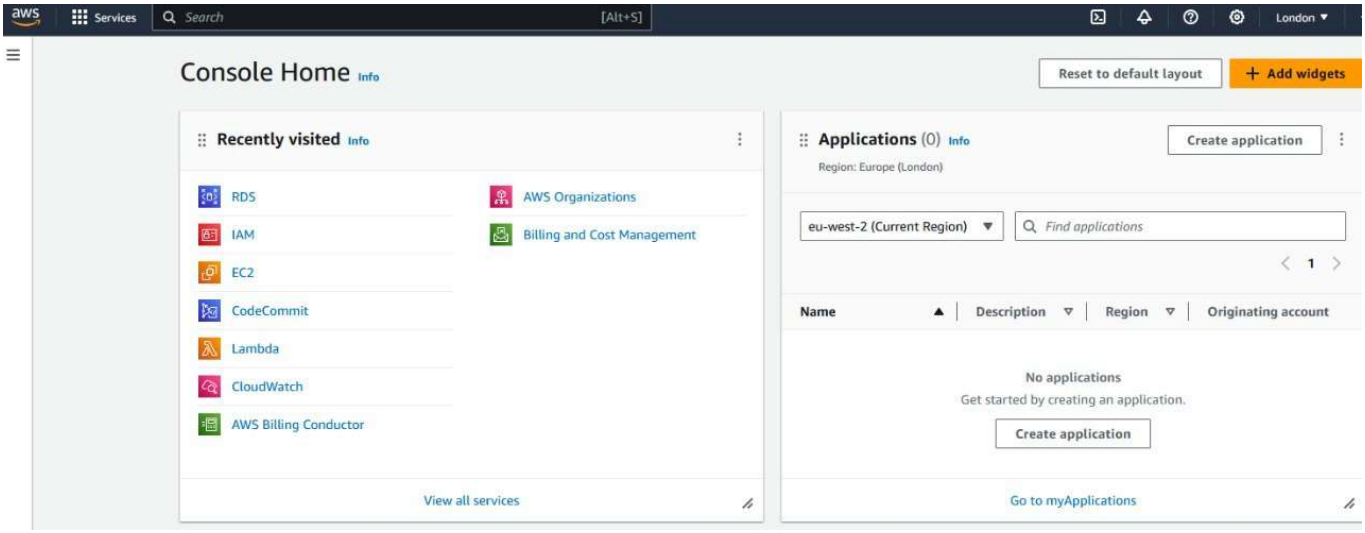

Рисунок 4.1.1 Основний дашборд AWS

Джерело: побудовано автором (знімок з екрану)

4.1. Технологія створення облікових записів

При роботі з Amazon Web Services (AWS) починаємо з налаштування управління ідентифікацією та доступом (IAM) - це не просто рекомендація, а критично важливий крок. Це початок шляху до створення безпечного та ефективно керованого хмарного середовища. IAM забезпечує точний контроль над тим, хто і як може використовувати ресурси AWS, що є основою для захисту даних і додатків.

Процес починаємо зі створення облікових записів IAM для кожного користувача. Такий підхід дозволяє уникнути використання спільних облікових записів і підвищує безпеку, гарантуючи, що дії кожного користувача можна відстежити. Потім визначаються ролі та відповідні правила доступу, які гарантують, що кожен користувач або служба має лише ті дозволи, які необхідні для виконання своєї роботи. Управління доступом спрощується завдяки групуванню користувачів зі схожими дозволами. Створення груп для різних відділів або команд допомагає централізувати управління правилами доступу. Важливою частиною безпеки є

включення багатофакторної автентифікації (MFA) для всіх користувачів, що значно знижує ризик несанкціонованого доступу. Принцип найменших привілеїв відіграє ключову роль у безпечній роботі з AWS. Він передбачає надання користувачам і сервісам лише тих дозволів, які є абсолютно необхідними для виконання їхніх завдань. Такий підхід допомагає запобігти потенційним ризикам, пов'язаним з надмірними привілеями. Нарешті, моніторинг та аудит діяльності IAM за допомогою таких інструментів, як AWS CloudTrail, відіграє важливу роль у підтримці безпеки та дотриманні політик. Це дозволяє своєчасно виявляти та реагувати на будь-яку підозрілу активність.

Реалізувавши за допомогою Terraform автоматизацію процесу конфігурації IAM, не тільки забезпечуємо високий рівень безпеки, але й закладаємо основу для гнучкого та масштабованого використання хмарних ресурсів.

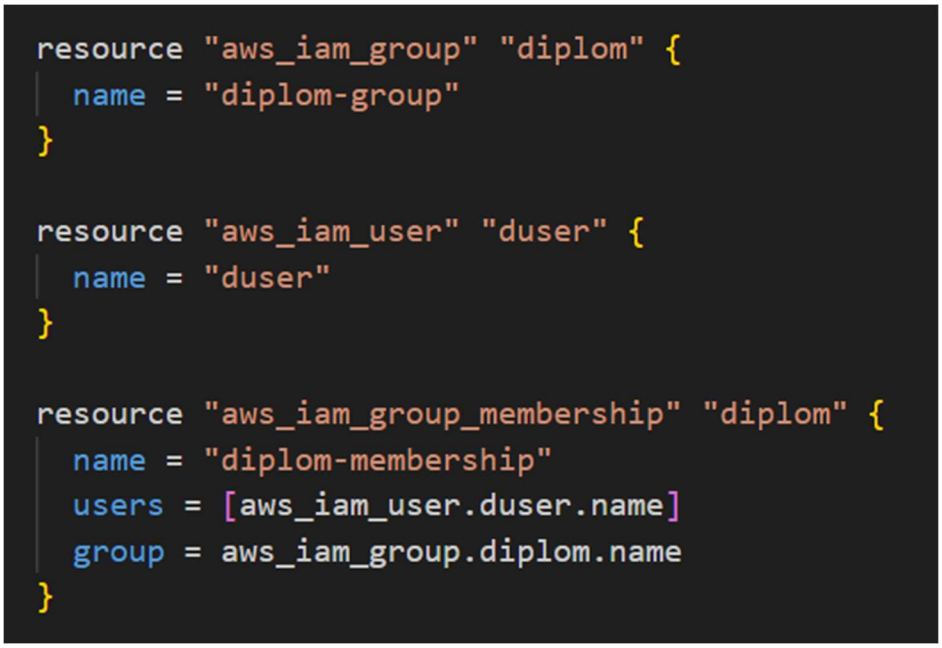

Рисунок 4.2. Автоматизація створення користувача, групи та додавання користувача до групи

Джерело: побудовано автором (знімок з екрану)

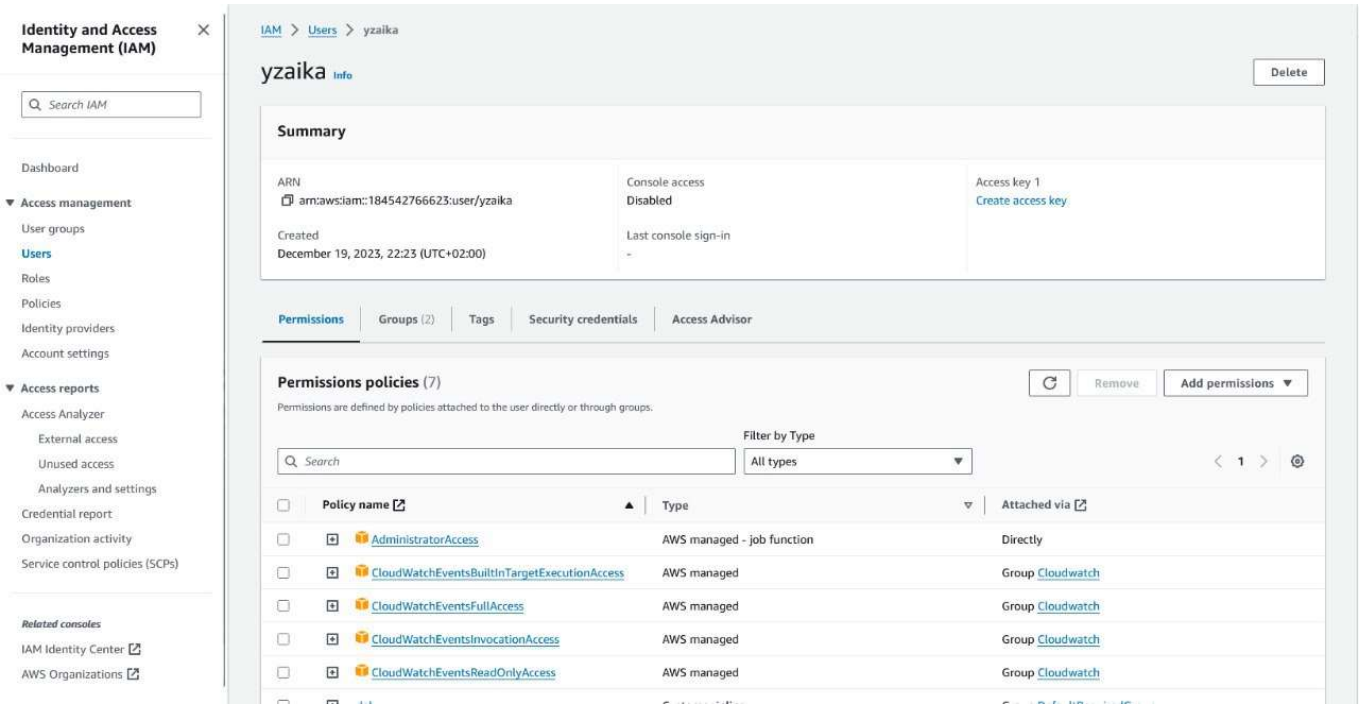

Рисунок 4.2.1. Дашборд IAM

Джерело: побудовано автором (знімок з екрану)

4.2. Технологія створення та налаштування мережі

Наступним кроком інтеграції виконуємо конфігурацію віртуальної приватної хмари (VPC) яка є важливим аспектом створення ефективної, безпечної та масштабованої хмарної інфраструктури. VPC в AWS надає потужні можливості для ізоляції та контролю мережевого середовища, що є ключовим для управління доступом до мережі, безпекою даних та загальною продуктивністю системи.

Створення VPC - це не лише вибір інструментів і сервісів, але й розуміння того, як ці ресурси повинні взаємодіяти і бути організовані для оптимальної продуктивності. Створюючи власний VPC замість попередньо встановлених опцій AWS, користувачі мають повний контроль над мережевими аспектами своєї хмарної інфраструктури, від IP-адресації до детальної конфігурації мережевих шлюзів і таблиць маршрутизації. Наріжним каменем ефективної VPC є правильне планування мережевої структури. Це включає створення підмереж, які можна ізолювати для

різних цілей, наприклад, для розділення ресурсів між загальнодоступними та приватними сегментами мережі. Такий поділ дозволяє точніше контролювати доступ до ресурсів, підвищуючи безпеку та продуктивність. Крім того, налаштування таблиць маршрутизації для кожної підмережі в VPC має вирішальне значення для управління потоками трафіку. Правильно налаштовані таблиці маршрутизації забезпечують ефективний і безпечний потік трафіку як всередині VPC, так і між VPC і зовнішнім світом. Шлюзи, такі як інтернет-шлюзи для публічних підмереж і NAT-шлюзи для приватних підмереж, забезпечують зв'язок між VPC і зовнішньою мережею. Ці елементи відіграють ключову роль у контролі доступу до ресурсів і захисті внутрішніх мережевих ресурсів. Крім того, конфігурація груп безпеки і списків контролю доступу (ACL) для VPC забезпечує додатковий рівень контролю доступу до ресурсів. Групи безпеки використовуються для контролю доступу до екземплярів EC2, в той час як ACL застосовуються на рівні підмережі, контролюючи доступ до мережевих ресурсів в більш широкому сенсі. Таким чином, ретельно спланована і налаштована VPC забезпечує необхідний рівень ізоляції, безпеки і контролю над мережевими ресурсами в AWS. Це не тільки підвищує загальну безпеку і продуктивність хмарної інфраструктури, але також забезпечує гнучкість масштабування і адаптацію до мінливих потреб бізнесу. Тому конфігурація VPC є критично важливим кроком у будь-якій стратегії хмарної архітектури.

Рисунок 4.3. Створення VPC та підмережі

Джерело: побудовано автором (знімок з екрану)

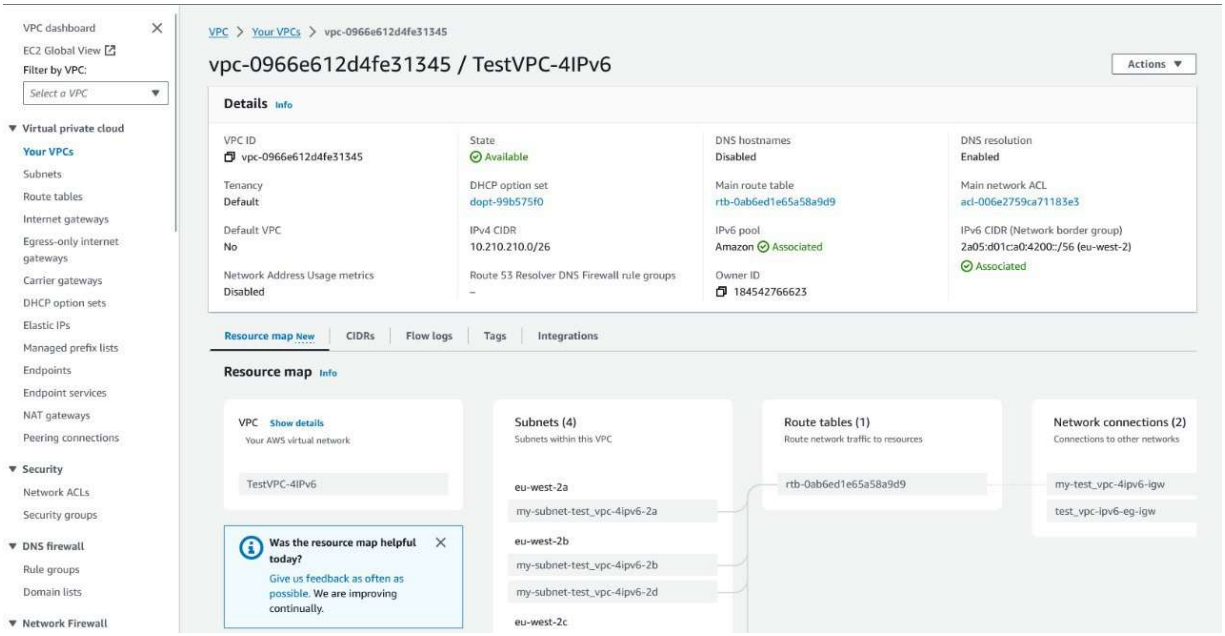

Рисунок 4.3.1 Створення VPC та підмережі

Джерело: побудовано автором (знімок з екрану)

4.3. Сервіси зберігання та керування базами даних

Успішно автоматизувавши управління ідентифікацією та доступом (IAM) і налаштування віртуальної приватної хмари (VPC) на Amazon Web Services (AWS), наступним етапом є встановлення та налаштування ключових сервісів: Amazon Relational Database Service (RDS) та Amazon Simple Storage Service (S3). Ці сервіси є невід'ємними компонентами багатьох додатків і систем, що працюють у хмарі AWS. Нашим наступним кроком виконуємо створення RDS. RDS - це рішення для керованої бази даних, яке значно спрощує завдання встановлення, обслуговування та масштабування баз даних. Спираючись на вже налаштований VPC, можемо інтегрувати нашу базу даних RDS в ізольовану мережу, забезпечуючи таким чином підвищену безпеку та продуктивність. Крім того, за допомогою IAM маємо механізми контролю доступу до бази даних, що дозволяє точно контролювати, хто може бачити або змінювати дані. Далі зосередимося на Amazon S3, хмарному сервісі зберігання даних, який пропонує відмінну масштабованість і надійність. S3 буде

використовуватися для зберігання різних типів даних від статичних файлів веб-сторінок до великих наборів даних, що використовуються нашими додатками. Налаштовані політики IAM забезпечують детальний контроль доступу досховища S3, що дозволить нам точно визначити, хто і як може отримати доступ до цих даних. Важливо, що інтеграція RDS і S3 з налаштованим VPC покращує загальну архітектуру безпеки.

Використовуючи VPC, можемо обмежити доступ до наших баз даних RDS і кошиків S3, налаштувавши їх так, щоб вони були доступні тільки з нашої внутрішньої мережі або з певних служб AWS. Отже, після налаштування IAM та VPC створюємо основу для безпечного та ефективного використання AWS. Наступні кроки включають впровадження та налаштування RDS та S3, які дозволять повністю використати потужність хмарних обчислень для AWS. Ці кроки мають вирішальне значення для розвитку хмарної інфраструктури.

```
resource "aws_s3_bucket" "diplom_s3_bucket" {
  bucket = "diplomproject-dev"
  acl    = "private"

  tags = {
    Name        = "DiplomProjectDevelopment"
    Environment = "Development"
  }
}
```

Рисунок 4.4. Створення AWS S3 Bucket

Джерело: побудовано автором (знімок з екрану)

4.4. Створення бази даних

Оскільки підсистеми потребують бази даних, то переходимо до наступного важливого кроку - створення екземпляру MySQL на Amazon Relational Database Service (RDS). Для забезпечення гнучкості та безпеки конфігурації планується використовувати зовнішній конфігураційний файл для зберігання критичних параметрів конфігурації. На цьому кроці зосередимося на створенні екземпляру MySQL в RDS. Amazon RDS надає зручні інструменти для налаштування та управління реляційними базами даних, автоматизації таких завдань, як резервне копіювання, виправлення та масштабування. Однак, щоб забезпечити безпеку та відповідність найкращим практикам, важливо ретельно керувати налаштуваннями бази даних. Для цього ми використовуємо зовнішній конфігураційний файл. Такий підхід дозволяє ізолювати конфіденційні дані, такі як паролі та налаштування конфігурації, від основного коду Terraform. Конфігураційний файл буде містити ключову інформацію, таку як ім'я бази даних, ім'я користувача та пароль. Це підвищує безпеку, оскільки конфіденційні дані не будуть зберігатися безпосередньо в інфраструктурному коді. Потім, під час процесу конфігурації, Terraform буде зчитувати дані з цього файлу, щоб створити і налаштувати екземпляр RDS. Це означає, що кожного разу, коли змінюються параметри бази даних, нам потрібно буде лише оновити цей файл, не впливаючи на основний код. Це також спрощує процес оновлення та підтримки нашої інфраструктури. Після створення екземпляру MySQL в RDS він буде інтегрований в наш існуючий VPC, забезпечуючи ізольоване і безпечне середовище для бази даних. Використовуючи налаштовані групи безпеки та

мережеві правила, ми можемо точно контролювати доступ до екземпляру бази даних, обмежуючи його лише необхідними службами та користувачами.

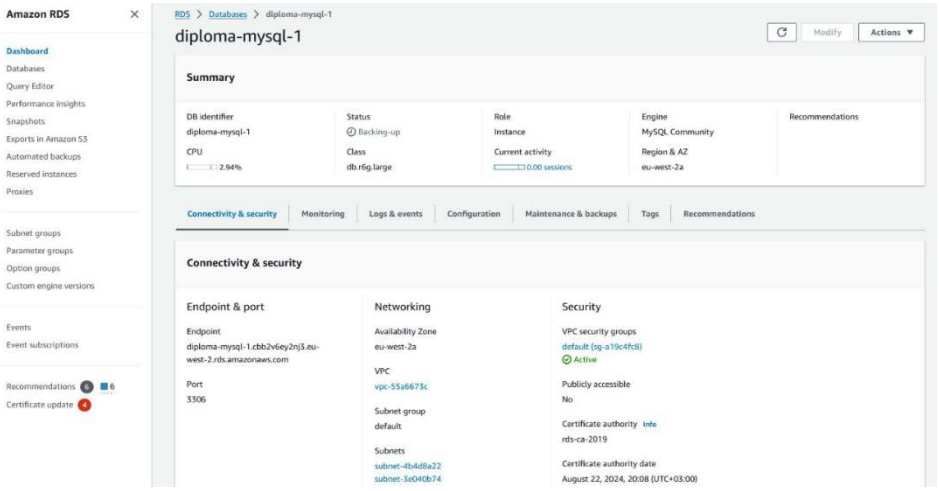

Рисунок 4.4.1. Дашборд RDS

Джерело: побудовано автором (знімок з екрану)

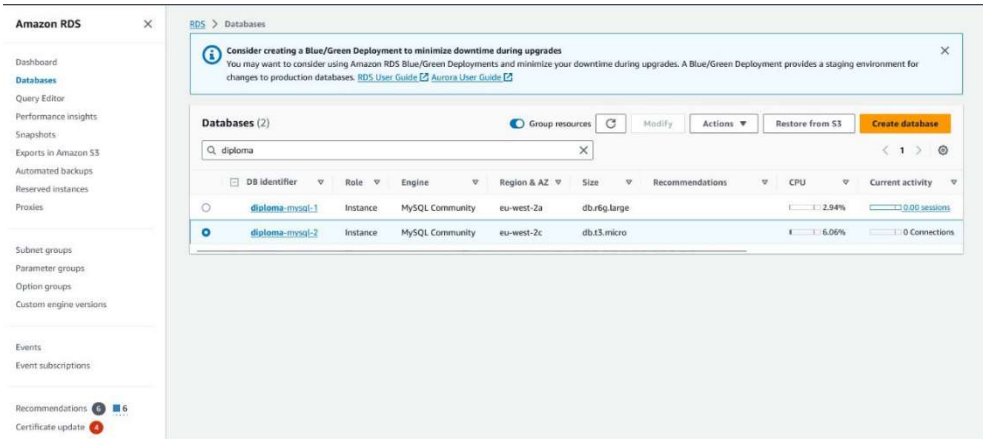

Рисунок 4.4.2. Дашборд RDS Database

Джерело: побудовано автором (знімок з екрану)

4.5. Створення EC2 екземплярів

Встановивши основи, включаючи управління ідентифікацією та доступом (IAM), віртуальну приватну хмару (VPC) та налаштувавши Amazon Relational Database Service (RDS) і Amazon Simple Storage Service (S3), було зроблено наступний крок, створено і налаштовано екземпляр Amazon Elastic Compute Cloud (EC2). Потрібно створити два екземпляри EC2, один з яких буде працювати з Docker, а інший без нього, оскільки вони виконуватимуть різні ролі в нашій системі.

Перший екземпляр EC2 налаштовуємо без Docker. Цей екземпляр буде призначений для підсистем, які не потребують контейнеризації, і буде використовуватися для завдань, які вимагають прямого доступу до операційної системи та апаратних ресурсів. Налаштування цього екземпляру буде зосереджено на оптимізації продуктивності та безпеки, враховуючи його пряму інтеграцію з рештою нашої інфраструктури.

Для налаштування другого екземпляру EC2 виконуємо налаштування за допомогою Docker. Цей екземпляр призначений для підсистем, які потребують ізоляції та гнучкості контейнеризації. Docker дозволить нам легко розгортати, масштабувати та керувати контейнерними додатками, забезпечуючи при цьому необхідний рівень ізоляції. Це особливо важливо для тестування та розробки, де нам потрібно швидко та ефективно створювати та видаляти різні середовища.

Вибір двох різних типів екземплярів EC2 відображає потребу створити масштабовану та гнучку хмарну архітектуру, яка може підтримувати різноманітні потреби підсистем. Використання екземпляру EC2 без Docker ідеально підходить для традиційних додатків і завдань, тоді як екземпляр з Docker підтримуватиме контейнерні додатки, що вимагають високого ступеня мобільності та пакування. При налаштуванні цих екземплярів було зосереджено на безпеці, продуктивності та інтеграції з уже налаштованими компонентами нашої хмарної інфраструктури. Це забезпечить надійну та ефективну роботу всієї системи, що дозволить нам гнучко масштабувати та адаптувати наші сервіси до мінливих вимог та викликів.

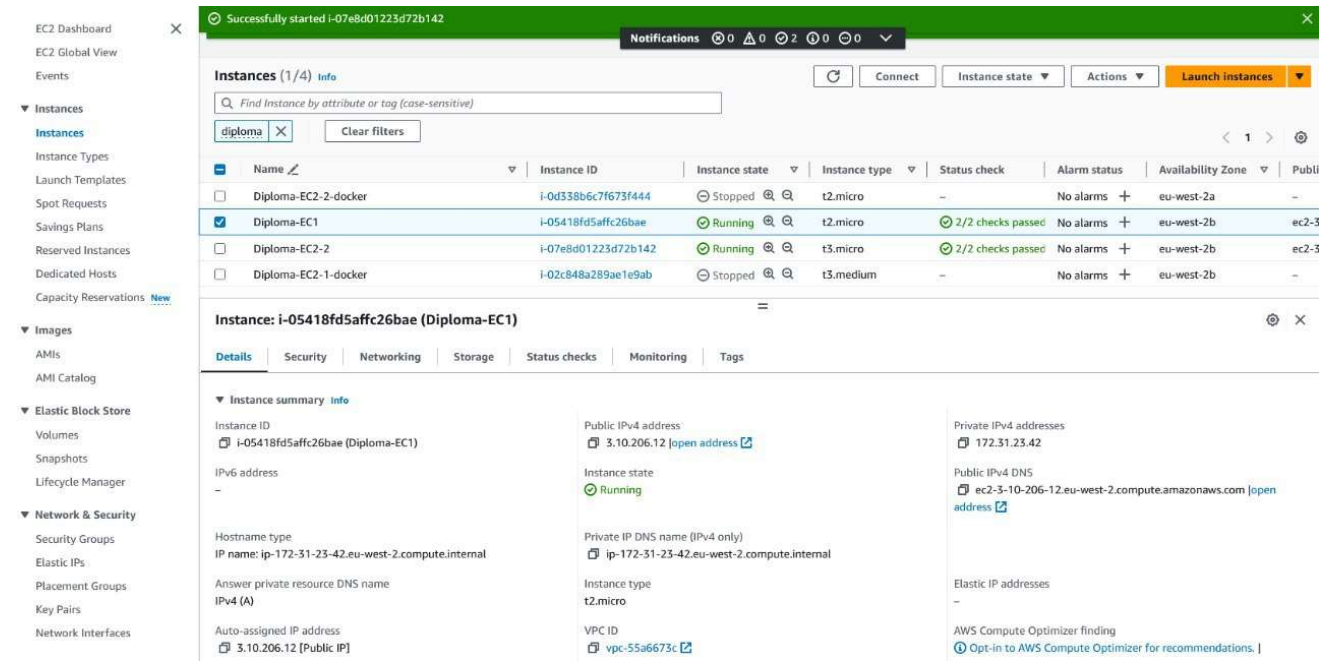

Рисунок 4.5.1. Дашборд EC2

Джерело: побудовано автором (знімок з екрану)

4.6. Створення AWS Lambda

У рамках постійного розвитку системної інфраструктури наступним завданням є створення функції AWS Lambda для автоматичного розгортання баз даних MySQL. Цей крок не тільки спрощує управління ресурсами, але й вносить елемент автоматизації та гнучкості в процес розгортання. Налаштовуємо окрему лямбда-функцію в якої є здатність зчитувати параметри конфігурації з файлу, який буде створено та оновлено під час роботи системи. Мета цієї функції - автоматизувати процес конфігурації та розгортання екземплярів MySQL в нашому хмарному середовищі, що особливо важливо для динамічно мінливих потреб проектування та розробки. Лямбда-функція активується, коли новий конфігураційний файл надходить у певне сховище S3, забезпечуючи його своєчасне виконання з актуальними даними.

Процес роботи лямбда-функції починається, коли вона отримує конфігураційний файл, що містить усі необхідні дані для створення бази даних: ім'я бази даних, розмір, версію MySQL, параметри безпеки тощо. Функція аналізує цей файл, витягує з нього параметри та ініціює процес створення нового екземпляру бази даних в Amazon RDS. Це дозволяє нам забезпечити високий рівень кастомізації та гнучкості при розгортанні баз даних, а також скоротити час і ресурси, необхідні для ручної конфігурації.

Безпека та надійність цього процесу є ключовими. Lambda була розроблена з урахуванням найкращих практик безпеки AWS, включаючи використання ролей IAM для мінімізації прав доступу та використання шифрування для захисту конфіденційних даних. Також були передбачені механізми реєстрації та моніторингу для відстеження процесу розгортання та швидкого реагування на потенційні помилки або зміни в системі.

Впровадження цієї функціональності Lambda в хмарну архітектуру відкриває нові можливості для оптимізації та автоматизації, роблячи процес управління базами

даних більш ефективним і гнучким. Це підкреслює нашу орієнтацію на інновації та прагнення до постійного вдосконалення процесів у нашій хмарній інфраструктурі.

```python
import json
import boto3

def lambda_handler(event, context):
    s3_client = boto3.client('s3')

    bucket_name = event['Records'][0]['s3']['bucket']['name']
    file_key = event['Records'][0]['s3']['object']['key']

    config_file = s3_client.get_object(Bucket=bucket_name, Key=file_key)
    config_params = json.loads(config_file['Body'].read().decode('utf-8'))

    rds_client = boto3.client('rds')

    try:
        response = rds_client.create_db_instance(
```

Рисунок 4.6. Частина коду Lambda-функції

Джерело: побудовано автором (знімок з екрану)

У процесі роботи було впроваджено впровадили низку ключових елементів, кожен з яких відіграє важливу роль у забезпеченні функціональності та продуктивності системи. Створення служб управління ідентифікацією та доступом (IAM) і віртуальної приватної хмари (VPC) заклало основу для безпечної та ізольованої роботи нашої системи. Було розроблено та розгорнуто екземпляри Amazon Relational Database Service (RDS) та Amazon Simple Storage Service (S3), які забезпечують надійне та масштабоване зберігання даних та управління ними. Крім того, екземпляри Amazon Elastic Compute Cloud (EC2), налаштовані як з Docker, так і без нього, забезпечують гнучку обчислювальну потужність, необхідну для різноманітних завдань і додатків нашої системи. Було приділено особливу увагу інтеграції та автоматизації процесів. Використання AWS Lambda для автоматичного розгортання баз даних MySQL є чудовим прикладом нашої стратегії автоматизації, що дозволяє нам швидко адаптуватися до мінливих потреб і спростити процеси

управління. Тепер, коли основні компоненти системи готові і повністю функціонують, можна перейти до фази безперервного розвитку та оптимізації. Подальші компоненти та вдосконалення будуть додаватися в міру необхідності, залежно від конкретних вимог кожної інсталяції системи. Такий підхід гарантує, що система не тільки відповідає поточним потребам наших користувачів, але й легко адаптується до майбутніх вимог і змін. Таким чином, наша система підтримки прийняття рішень готова до роботи, з гнучкістю і масштабованістю для зростання і розвитку. Це важливий крок вперед на шляху до високоефективної та гнучкої хмарної інфраструктури.

ВИСНОВКИ

Гібридна енергетична мережа - це складна система, що складається з різних джерел енергії, таких як сонячні панелі, вітрові електростанції, теплові електростанції тощо. Кожен тип джерела енергії має свої особливості, які необхідно враховувати при управлінні гібридною мережею.

Традиційні методи управління гібридними енергетичними мережами не завжди дозволяють їм працювати ефективно. Це пов'язано з тим, що вони не враховують всі фактори, які впливають на роботу гібридної енергетичної мережі.

Новим підходом до управління гібридними енергосистемами є інформаційна технологія інтеграції моделей. Ця технологія передбачає використання двох моделей: моделі управління гібридною мережею та моделі прийняття рішень.

Модель управління гібридною мережею відповідає за оптимізацію роботи мережі. Вона враховує всі фактори, що впливають на роботу гібридної мережі, такі як

Доступні джерела енергії

Попит на електроенергію

Ціни на електроенергію

Фактори навколишнього середовища

Модель прийняття рішень дозволяє користувачеві приймати рішення на основі даних, отриманих від керуючої моделі. Ці рішення можуть стосуватися режиму роботи гібридної мережі, інвестицій в нові джерела енергії тощо.

Проведено розробку інформаційної технології інтеграції моделей та даних, та показано, що розроблена технологія дозволяє підвищити ефективність управління гібридною енергосистемою. Таким чином, інформаційна технологія інтеграції

моделей є ефективним інструментом управління гібридними електричними мережами. Вона дозволяє підвищити ефективність гібридної енергосистеми за рахунок оптимізації режиму її роботи. Ця технологія може бути використана для управління гібридними електромережами різних типів.

Для підвищення ефективності розробленої ІТ-технології необхідні подальші дослідження та вдосконалення. Це може бути досягнуто шляхом розробки більш складних моделей управління гібридними мережами, які враховують більше факторів, більш ефективних моделей прийняття рішень, які дозволяють користувачеві приймати більш оптимальні рішення, а також більш комфортного користувацького інтерфейсу, який забезпечує більш зручну взаємодію між користувачем і системою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Soroudi A. & Amraee T. (2013). Decision making under uncertainty in energy systems: State of the art. Renewable and Sustainable Energy Reviews, 28, 376-384. doi: 10.1016/j.rser.2013.08.039

2. Polikarpova, I. & Rosa, M. (2017). Energy reduction potential of the district heating company introducing energy management systems. Energy Procedia, 128, 66-71. doi: 10.1016/j.egypro.2017.09.016

3. Парфененко Ю. В. Інформаційна технологія моніторингу функціонування системи теплопостачання підвищеної надійності [Текст] / Ю. В. Парфененко, В. Г. Неня // Східно-європейський журнал передових технологій. – 2010. – № 4/9 (46). – С. 22–25.

4. Інформаційно-аналітична система моніторингу та прогнозування теплозабезпечення будівель / Ю. В. Парфененко, В. В. Шендрик, В. Г. Неня, Р. П. Окопний // Вісник Східноукраїнського національного університету імені Володимира Даля. – 2013. – № 743. – Ч. 1. – С. 38– 43.

5. Peulic A. Flexible GPS/GPRS based System for Parameters Monitoring in the District Heating System / A. Peulic, S. Dragicevic, M. Snezana, Z. Jovanovic, R. Krneta // International Journal of Computers Communications & Control . – 2013. – Vol. 8. – No. 1. – P. 105–110.

6. Wojciech Grega. Information Technologies Supporting Control and Monitoring of Power System / Grega Wojciech // Przeglad Elektrotechniczny (Electrical Review). – 2012. –No. 5a. – P. 193–197.

7. From Industry 4.0 to Energy 4.0: Future Business Models and Legal Relations. – [Електронний ресурс]. – Режим доступу : http://www.germanenergyblog.de/wp-

content/uploads/2016/04/160317_Lang_IBE_Bochum_From_Industry_4_0_to_Energy_4_0_fin.pdf

8. Тимчук С. О., Шендрик В. В., Шендрик С. О., Шулима О. В. Прийняття оптимальних рішень при керуванні гібридними електричними мережами з відновлювальними джерелами енергії. Електромеханічні і енергозберігаючі системи. 2016. №34. С. 55-61.

9. Шендрик С. О., Тимчук С. О. Аналіз предметної області прийняття рішень при управлінні гібридними енергомережами. Автоматика – 2017: матеріали XXIV Міжнародна конференція з автоматичного управління, Київ, 13–15 вересня 2017 р. Київ. 2017. С. 221-222

10. Ahmad, F., Alam, M. S., and Asaad, M. (2017). Developments in xEVs charging infrastructure and energy management system for smart microgrids including xEVs. Sustain. cities Soc. 35, 552–564. doi:10.1016/j.scs.2017.09.008

11. Zamfir, M., Florian, V., Stanciu, A., Neagu, G., Preda, Ş., and Militaru, G. (2016). "Towards a platform for prototyping IoT health monitoring services," in International conference on exploring services science (Cham: Springer), 522–533

12. Elmouatamid, A., Ouladsine, R., Bakhouya, M., El Kamoun, N., Zine-Dine, K., and Khaidar, M. (2019). "A model predictive control approach for energy management in micro-grid systems," in 2019 international conference on smart energy systems and technologies (SEST) (IEEE)

13. Sedhom, B. E., El-Saadawi, M. M., El Moursi, M. S., Hassan, M. A., and Eladl, A. A. (2021). IoT-based optimal demand-side management and control scheme for smart microgrid. Int. J. Electr. Power & Energy Syst. 127

14. Sylcloud Smart Micro Grid Sylcloud smart micro grid. Доступно за посиланням: https://sylcloud. com/smartmicrogrid (Accessed March 26 2022).

15. https://www.researchgate.net/figure/A-Schematic-view-of-DSS_fig1_50417997, режим доступу: відкритий, доступ на (20.10.2023)

16. Ali, S. S., and Choi, B. J. (2020). State-of-the-art artificial intelligence techniques for distributed smart grids: A review. Electronics 9 (6), 1030.

17. Moghimi, M., Jamborsalamati, P., Hossain, J., Stegen, S., and Lu, J. (2018). "A hybrid communication platform for multi-microgridEMSoptimization," in 2018 IEEE 27th international symposium on industrial electronics (ISIE) (IEEE), 1215–1220.

18. Ghiasi, M., Wang, Z., Mehrandezh, M., Jalilian, S., and Ghadimi, N. (2022). Evolution of smart grids towards the Internet of energy: Concept and essential components for deep decarbonisation. IET Smart Grid.

19. Marinakis, V., and Doukas, H. (2018). An advanced IoT-based system for intelligent energy management in buildings. Sensors 18 (2), 610.

20. Zhuang, J., Shen, G., Yu, J., Xiang, T., and Wang, X. (2017). The design and implementation of intelligent microgrid monitoring system based on WEB. Procedia Comput. Sci. 107, 4–8.

21. https://www.researchgate.net/figure/General-system-diagram-The-main-tasks-of-DSS-is-verification-formatting-storing_fig1_221916051, режим доступу: відкритий, доступ на (20.10.2023)

22. Khoa, N. M., Dai, L. V., Tung, D. D., and Toan, N. A. (2021). An advanced IoT system for monitoring and analyzing chosen power quality parameters in microgrid solution. Archives Electr. Eng., 70.

23. https://blog.det.life/apache-nifi-in-action-data-integration-and-analysis-in-azure-be999ed170f7, режим доступу: відкритий, доступ на (20.10.2023)

24. Ali, M. A., Barakat, M. M., Abokhalaf, M. M., Fadel, Y. H., Kandil, M., Rasmy, M. W., et al. (2021). Micro-grid monitoring and supervision: Web-based SCADA approach. J. Electr. Eng. Technol. 16 (5), 2313–2331.

25. Karthick, T., Chandrasekaran, K., and Jeslin., D. N. J. (2021). Design of IoT based smart compact energy meter for monitoring and controlling the usage of energy and power quality issues with demand-side management for a commercial building. Sustain. Energy, Grids Netw. 26.

26. Chen, Y. Y., Lin, Y. H., Kung, C. C., Chung, M. H., and Yen, I. H. (2019). Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. Sensors 19 (9), 2047.

27. Mehrizi-Sani, A., and Iravani, R. (2010). Potential-function based control of a microgrid in islanded and grid-connected modes. IEEE Trans. Power Syst. 25 (4), 1883–1891.

28. Mohanty, S. P., Choppali, U., and Kougianos, E. (2016). Everything you wanted to know about smart cities: The Internet of things is the backbone. IEEE Consum. Electron. Mag. 5 (3), 60–70.

29. https://twitter.com/awscloud/status/694932117164150784, режим доступу: відкритий, доступ на (20.10.2023)

30. Ghiasi, M., Dehghani, M., Niknam, T., Kavousi-Fard, A., Siano, P., and Alhelou, H. H. (2021). Cyber-attack detection and cyber-security enhancement in smart DC microgrid based on blockchain technology and Hilbert Huang transform. Ieee Access 9, 29429–29440.

31. Vuddanti, S., and Salkuti, S. R. (2021). Review of energy management system approaches in microgrids. Energies 14 (17).

32. Zheng, J., Gao, D. W., and Lin, L. (2013). “Smart meters in smart grid: An overview,” in 2013 IEEE green technologies conference (GreenTech) (IEEE), 57–64.

33. Guo, Y., Tan, C. W., Hu, S., and Weaver, W. W. (2015). “Modeling distributed denial of service attack in advanced metering infrastructure,” in 2015 IEEE power & energy society innovative smart grid technologies conference (ISGT) (IEEE), 1–5.

34. Barai, G. R., Krishnan, S., and Venkatesh, B. (20152015). IEEE electrical power and energy conference (EPEC). IEEE, 138–145.Smart metering and functionalities of smart meters in smart grid-a review.

35. Avancini, D. B., Rodrigues, J. J., Rabêlo, R. A., Das, A. K., Kozlov, S., and Solic, P. (2021). A new IoT-based smart energy meter for smart grids. Int. J. Energy Res. 45 (1), 189–202.

36. ПОПАДЧЕНКО, С. А.; САВЧЕНКО, О. А.; АБРАМОВ, М. А. Підвищення ефективності технологій Smart Grid на основі моніторингу параметрів електричної мережі. 2019, 20-21.

37. Кириленко А. В. Інтелектуальні електроенегретичні системи: елементи та режими; под загальною редакцією акад. НАН України А. В. Кириленко. Київ : Інститут електродинаміки НАН України, 2014, 408.

38. Попадченко С. А. Аналіз світових тенденцій модернізації електричних підстанцій на сучасному етапі розвитку. Енергетика та електрифікація. 2016. №9, 46-49.

39. Model Predictive Control for the Energy Management of A Hybrid PV/Battery /Fuel Cell Power Plant [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.researchgate.net/publication/335566238_Model_Predictive_Control_for_the_Energy_Management_of_A_Hybrid_PVBattery_Fuel_Cell_Power_Plant (дата звернення: 01.12.2023)

40. Thurner, L, Scheidler, A, Schäfer, F, Menke, J-H, Dollichon, J, Meier, F, Meinecke, S and Braun, M 2017 pandapower – an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems. Preprint. [Електронний ресурс]. – 2017.

41. Shin, H.C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging 2016, 35, 1285–1298

42. Docker development best practices [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://docs.docker.com/develop/dev-best-practices/ (дата звернення: 25.11.2023).

43. Wang, Z.X.; He, L.Y.; Zheng, H.H. Forecasting the residential solar energy consumption of the United States. Energy 2019, 178, 610–623

44. Interfacing Power System and ICT Simulators: Challenges, State-of-the-Art, and Case Studies [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://ieeexplore.ieee.org/document/7444194 (дата звернення: 01.12.2023)

45. Corrado, E.M., & Frederick, K. (2008). Free and Open Source Options for Creating Database-Driven Subject Guides. Code4Lib Journal.

46. https://oyunbilegdav.medium.com/3-tier-application-deployment-on-aws-using-terraform-and-docker-2483df995686, режим доступу: відкритий, доступ на (20.10.2023)

ДОДАТОК А

ПЛАНУВАННЯ РОБІТ

для розробки кваліфікаційної роботи магістра

«Інформаційна технологія інтеграції моделей в системі підтримки прийняття рішень при управлінні гібридною енергомережою»

А.1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

У рамках розробки інформаційної технологій інтеграції даних, SMART-методологія дає можливість чітко визначити та структурувати цілі проекту. Розшифровка термінів SMART допомагає нам зосередитися на ключових аспектах мети проекту: вона має бути конкретною (Specific), вимірюваною (Measurable), досяжною (Achievable), реалістичною (Relevant) та обмеженою в часі (Time-framed).

S: Мета проекту полягає у розробці сервісу, який забезпечуватиме інтеграцію даних. Цей сервіс включатиме функціонал для моніторингу стану мікромережі, генерації/споживання/накопичення електроенергії, та генерації звітів в режимі реального часу, моделей прогнозування генерації та споживання, моделей прийняття рішень, бази даних, модулі, підсистеми та мікросервіси, що сприятиме оптимізації управління енергією та ефективному контролю за системою в цілому.

M: Ефективність системи можна вимірювати через здатність підготувати та сформувати наявну інформацію, та здатність відобразити її з використанням мінімуму технічних засобів. Критерії оцінки ефективності включають покращення інформування кінцевого користувача.

A: В наявності є всі необхідні технології та ресурси для розробки цього сервісу, включаючи доступ до бази даних параметрів та інструментів для їх обробки та аналізу.

R: Проект відповідає поточним тенденціям в сфері управління енергетичними системами та відображає актуальні потреби галузі. Команда розробників має відповідні знання та досвід для виконання цього проекту.

T: Проект має чітко визначені терміни виконання, які відповідають плановому графіку розробки та впровадження модуля в систему управління гібридною енергомережею.

Таблиця А.1 – Деталізація мети методом SMART

Категорія SMART	**Опис**
Specific (конкретна)	Розробка інформаційної технології інтеграції даних.
Measurable (вимірювана)	Оцінка ефективності інтеграції повнотою обробки інформації.
Achievable (досяжна)	Доступ до необхідних технологій та ресурсів.
Relevant (реалістична)	Проект відповідає актуальним потребам галузі, команда має відповідні компетенції.
Time-framed (обмежена у часі)	Чітко визначені терміни для розробки та впровадження модуля.

А.2 ПЛАНУВАННЯ ЗМІСТУ СТРУКТУРИ РОБІТ ІНФОРМАЦІНОЇ СИСТЕМИ

У контексті розробки технології інтеграції, структура робіт (Work Breakdown Structure, WBS) є ключовою для організації та планування проекту. WBS допомагає визначити ієрархію та залежності між різними етапами робіт, що сприяє чіткому розумінню обсягу та змісту проекту.

WBS організовує та визначає весь зміст проекту. На верхньому рівні WBS розташовується кінцевий продукт – сервіс інтеграції. Кожен наступний рівень деталізує конкретні завдання та етапи, необхідні для досягнення цієї мети. Це дозволяє визначити, які саме роботи є частиною проекту та які – ні.

Формування технічного завдання: Включає визначення функціональних та технічних вимог технології, вибір технологій та платформ для розробки. Також тут відбувається визначення цілей та можливостей проекту.

Планування проекту: Розробка OBS (Organizational Breakdown Structure) та матриці відповідальності, планування ресурсів, оцінка ризиків, розробка календарного плану включаючи діаграму Ганта.

Реалізація проекту: Поділяється на кілька основних етапів, включаючи проектування модуля, розробку, тестування та оптимізацію. Ключова увага приділяється якості та ефективності роботи модуля.

Впровадження та закриття проекту: Включає фінальне тестування, виправлення помилок, підготовку документації та передачу модуля в експлуатацію.

Діаграми WBS та OBS:

Діаграма WBS: На рисунку А.1 наведено діаграму WBS, яка ілюструє ієрархічну структуру робіт в проекті.

Діаграма OBS: На рисунку А.2 представлено OBS, що демонструє організаційну структуру проекту та розподіл відповідальності між учасниками.

Цей підхід до планування змісту структури робіт забезпечує чітке розуміння та ефективне управління проектом, від початкових етапів формування технічного завдання до фінального етапу впровадження та закриття проекту.

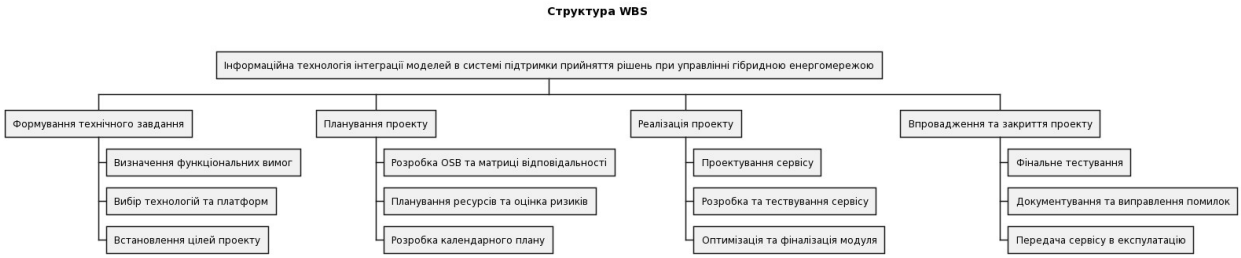

Рисунок А.1 Структура WBS

Рисунок А.2 Структура OBS

А.3 ПОБУДОВА КАЛЕНДАРНОГО ГРАФІКУ ВИКОНАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Для того, щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі будують календарний графік робіт.

Діаграма Ганта – горизонтальна лінійна діаграма, на якій задачі проекту представляються протяжними в часі відрізками, що характеризуються датами початку та закінчення, затримками і, можливо, іншими тимчасовими параметрами.

Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

На наступному рисунку представлено діаграму Ганта розроблюваного проекту. На рисунку А.3 представлено побудовану діаграму Ганта.

Етап проекту	Початок	Тривалість (дні)
Формування технічного завдання	25.09.2023	5
Аналіз ринку	30.09.2023	2
Розробка функціональних вимог	02.10.2023	8
Проектування архітектури сервісу	10.10.2023	15
Інтеграція бази даних	25.10.2023	10
Інтеграція сервісу	04.11.2023	8
Тестування системи	12.11.2023	7
Розгортання системи	19.11.2023	2
Технічна документація	21.11.2023	4
Завершення та аналіз проекту	25.11.2023	2

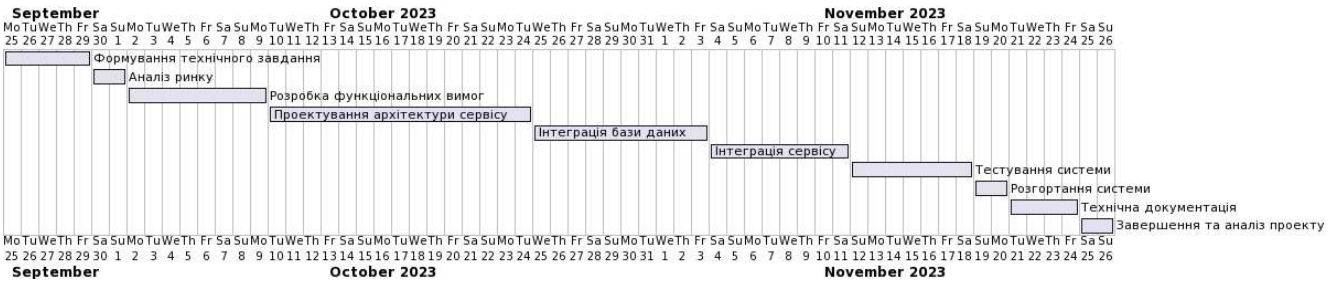

Рисунок А.3 - Діаграма Ганта

А.4 ПЛАНУВАННЯ РИЗИКІВ ПРОЕКТУ

Управління ризиками є критично важливим елементом у розробці інформаційної технології інтеграції моделей та даних в системі управління гібридною енергомережею. Цей процес включає ідентифікацію потенційних ризиків, оцінку їх впливу на проект, розробку стратегій реагування та постійний моніторинг.

Процес управління ризиками включає:

- **Ідентифікація ризиків**: Виявлення потенційних загроз проекту, включаючи технічні, економічні, юридичні та екологічні ризики.
- **Оцінювання ризиків**: Використання методу експертних оцінок для визначення ймовірності та впливу кожного ризику.
- **Планування реагування на ризики**: Розробка заходів щодо попередження ризиків або мінімізації їх наслідків.
- **Моніторинг ризиків**: Постійне відстеження ідентифікованих ризиків та ефективності заходів реагування.

План по усуненню ризиків:

- Підвищення безпеки системи: Впровадження сучасних технологій захисту даних та регулярний аудит безпеки.
- Резервне зберігання даних: Створення надійної системи резервного копіювання та відновлення даних.
- Оновлення та підтримка системи: Забезпечення своєчасного оновлення програмного забезпечення та апаратної частини.
- Забезпечення контингентності: Розробка плану дій на випадок виходу з ладу ключового обладнання, щоб забезпечити неперервність робіт над проектом.

Таблиця А.2 – Ймовірність втрат

Ймовірність виникнення	**Величина втрат**
Слабоймовірно	Мінімальна
Малоймовірно	Низька
Ймовірно	Середня
Вельми ймовірно	Висока
Майже можливо	Максимальна

Таблиця А.3 – Класифікація за ступенем впливу та за рівнем ризику

Ризик	**Ступінь впливу**	**Рівень ризику**
Кібератаки	4	Ігноровані
Пошкодження даних	4	Ігноровані
Серверні проблеми	3	Ігноровані
Вихід з ладу обладнання	6	Незначні

Ці таблиці та плани дій є основою для планування ризиків проекту, дозволяючи команді антисипативна відповідати на можливі проблеми та забезпечувати стабільність проекту.

ДОДАТОК Б

Використані скрипти

```
provider "aws" {
  region = "us-west-2" }

resource "aws_db_instance" "mysql" {
  allocated_storage    = 20
  engine               = "mysql"
  engine_version       = "5.7"
  instance_class       = "db.t2.micro"
  name                 = "mydatabase"
  username             = "admin"
  password             = "mysecurepassword"
  parameter_group_name = "default.mysql5.7"
  skip_final_snapshot  = true
}

resource "aws_ssm_parameter" "db_username" {
  name  = "/config/mysql/username"
  type  = "String"
  value = aws_db_instance.mysql.username
}

resource "aws_ssm_parameter" "db_password" {
  name  = "/config/mysql/password"
  type  = "SecureString"
  value = aws_db_instance.mysql.password
}

resource "aws_ssm_parameter" "db_endpoint" {
  name  = "/config/mysql/endpoint"
  type  = "String"
  value = aws_db_instance.mysql.endpoint
}

resource "aws_instance" "app_server" {
  ami           = "ami-0abcdef1234567890"
  instance_type = "t2.micro"

  user_data = <<-EOF
              #!/bin/bash
              yum update -y
              yum install docker -y
              service docker start
              echo "DB_USERNAME=$(aws ssm get-parameter --name /config/mysql/username --
query 'Parameter.Value' --output text)" >> /etc/environment
              echo "DB_PASSWORD=$(aws ssm get-parameter --with-decryption --name
/config/mysql/password --query 'Parameter.Value' --output text)" >> /etc/environment
              echo "DB_ENDPOINT=$(aws ssm get-parameter --name /config/mysql/endpoint --
query 'Parameter.Value' --output text)" >> /etc/environment
```

```
              EOF

  tags = {
    Name = "EC2 with Docker"
  }
}
provider "aws" {
  region = "us-west-2"
}

resource "aws_db_instance" "mysql" {
  allocated_storage    = 20
  engine               = "mysql"
  engine_version       = "5.7"
  instance_class       = "db.t2.micro"
  name                 = "mydatabase"
  username             = "admin"
  password             = "mysecurepassword"
  parameter_group_name = "default.mysql5.7"
  skip_final_snapshot  = true
}

resource "aws_ssm_parameter" "db_username" {
  name  = "/config/mysql/username"
  type  = "String"
  value = aws_db_instance.mysql.username
}

resource "aws_ssm_parameter" "db_password" {
  name  = "/config/mysql/password"
  type  = "SecureString"
  value = aws_db_instance.mysql.password
}

resource "aws_ssm_parameter" "db_endpoint" {
  name  = "/config/mysql/endpoint"
  type  = "String"
  value = aws_db_instance.mysql.endpoint
}

resource "aws_instance" "app_server" {
  ami           = "ami-0abcdef1234567890"
  instance_type = "t2.micro"

  user_data = <<-EOF
              #!/bin/bash
              echo "DB_USERNAME=$(aws ssm get-parameter --name /config/mysql/username --
query 'Parameter.Value' --output text)" >> /etc/environment
              echo "DB_PASSWORD=$(aws ssm get-parameter --with-decryption --name
/config/mysql/password --query 'Parameter.Value' --output text)" >> /etc/environment
              echo "DB_ENDPOINT=$(aws ssm get-parameter --name /config/mysql/endpoint --
query 'Parameter.Value' --output text)" >> /etc/environment
```

```
              EOF
}
provider "aws" {
  region = "us-west-2"
}

resource "aws_db_instance" "mysql_instance" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "mysql"
  engine_version       = "5.7"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "dbuser"
  password             = "mysecurepassword"
  parameter_group_name = "default.mysql5.7"
  skip_final_snapshot  = true
}


resource "aws_ssm_parameter" "db_username" {
  name  = "/myapp/database/username"
  type  = "String"
  value = aws_db_instance.mysql_instance.username
}

resource "aws_ssm_parameter" "db_password" {
  name  = "/myapp/database/password"
  type  = "SecureString"
  value = aws_db_instance.mysql_instance.password
}

resource "aws_ssm_parameter" "db_endpoint" {
  name  = "/myapp/database/endpoint"
  type  = "String"
  value = aws_db_instance.mysql_instance.endpoint
}
resource "aws_iam_user" "example_user" {
  name = "example-user"
  path = "/system/"
}

resource "aws_iam_user_policy" "example_policy" {
  name = "example_policy"
  user = aws_iam_user.example_user.name

  policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Action = [
          "s3:*"
```

```
        ],
        Effect = "Allow",
        Resource = "*"
      },
    ]
  })
}
resource "aws_subnet" "main_subnet" {
  count                   = length(var.subnets)
  vpc_id                  = aws_vpc.main_vpc.id
  cidr_block              = var.subnets[count.index].cidr_block
  availability_zone       = var.subnets[count.index].availability_zone

  tags = {
    Name = "subnet-${count.index}"
  }
}
provider "aws" {
  region = "us-west-2" # Замените на ваш регион
}

resource "aws_vpc" "main_vpc" {
  cidr_block = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true

  tags = {
    Name = "main-vpc"
  }
}

resource "aws_subnet" "public_subnet" {
  vpc_id     = aws_vpc.main_vpc.id
  cidr_block = "10.0.1.0/24"
  map_public_ip_on_launch = true

  tags = {
    Name = "public-subnet"
  }
}

# Групи безпеки
resource "aws_security_group" "ec2_sg" {
  name        = "ec2-security-group"
  vpc_id      = aws_vpc.main_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

```
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_security_group" "rds_sg" {
  name        = "rds-security-group"
  vpc_id      = aws_vpc.main_vpc.id

  ingress {
    from_port   = 3306
    to_port     = 3306
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

# IAM конфігурація
resource "aws_iam_user" "user1" {
  name = "user1"
}

resource "aws_iam_user" "user2" {
  name = "user2"
}

resource "aws_iam_group" "group1" {
  name = "group1"
}

resource "aws_iam_user_policy" "policy1" {
  name   = "policy1"
  user   = aws_iam_user.user1.name
  policy = jsonencode({
```

```
    Version = "2012-10-17",
    Statement = [
      {
        Action = ["s3:*"],
        Effect = "Allow",
        Resource = "*"
      }
    ]
  })
}

resource "aws_iam_group_membership" "membership1" {
  name = "example-membership1"
  users = [aws_iam_user.user1.name]
  group = aws_iam_group.group1.name
}

resource "aws_db_instance" "mysql_instance1" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "mysql"
  engine_version       = "8.0"
  instance_class       = "db.t2.micro"
  name                 = "mydb1"
  username             = "user1"
  password             = "yourpassword1"
  parameter_group_name = "default.mysql8.0"
  skip_final_snapshot  = true
  vpc_security_group_ids = [aws_security_group.rds_sg.id]
  db_subnet_group_name = "my-db-subnet-group"
}

resource "aws_db_instance" "mysql_instance2" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "mysql"
  engine_version       = "8.0"
  instance_class       = "db.t2.micro"
  name                 = "mydb2"
  username             = "user2"
  password             = "yourpassword2"
  parameter_group_name = "default.mysql8.0"
  skip_final_snapshot  = true
  vpc_security_group_ids = [aws_security_group.rds_sg.id]
  db_subnet_group_name = "my-db-subnet-group"
}


resource "aws_instance" "linux_instance1" {
  ami                    = "ami-0abcdef1234567890"
  instance_type          = "t2.micro"
  vpc_security_group_ids = [aws_security_group.ec2_sg.id]
```

```
  subnet_id              = aws_subnet.public_subnet.id
  key_name               = "my-key-name"

  user_data = <<-EOF
              #!/bin/bash
              yum update -y
              yum install docker -y
              service docker start
              usermod -a -G docker ec2-user
              EOF

  tags = {
    Name = "Linux Instance 1 with Docker"
  }
}

resource "aws_instance" "linux_instance2" {
  ami                    = "ami-0abcdef1234567890"
  instance_type          = "t2.micro"
  vpc_security_group_ids = [aws_security_group.ec2_sg.id]
  subnet_id              = aws_subnet.public_subnet.id
  key_name               = "my-key-name"

  user_data = <<-EOF
              #!/bin/bash
              yum update -y
              yum install docker -y
              service docker start
              usermod -a -G docker ec2-user
              EOF

  tags = {
    Name = "Linux Instance 2 with Docker"
  }
}

resource "aws_instance" "linux_instance3" {
  ami                    = "ami-0abcdef1234567890"
  instance_type          = "t2.micro"
  vpc_security_group_ids = [aws_security_group.ec2_sg.id]
  subnet_id              = aws_subnet.public_subnet.id
  key_name               = "my-key-name"

  tags = {
    Name = "Linux Instance 3"
  }
}

resource "aws_instance" "linux_instance4" {
  ami                    = "ami-0abcdef1234567890"
  instance_type          = "t2.micro"
  vpc_security_group_ids = [aws_security_group.ec2_sg.id]
```

```
  subnet_id              = aws_subnet.public_subnet.id
  key_name               = "my-key-name"

  tags = {
    Name = "Linux Instance 4"
  }
}

# S3 Bucket
resource "aws_s3_bucket" "my_bucket" {
  bucket = "my-unique-bucket-name"
  acl    = "private"
}
```