Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп’ютерних наук_____
(повна назва)

Кафедра ___Комп'ютерного моделювання та інтелектуальних технологій____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти _____другий (магістерський)_____

Дослідження та порівняльний аналіз методів аварійного відновлення систем в хмарних середовищах
(тема)

Виконав:
Здобувач ___2_____ року навчання,
Групи _____СПРм-24-1_____
_____Борис ГУНЬКО_____
(власне ім’я, прізвище)

Спеціальність _122 – Комп’ютерні науки____
_____
(код і повна назва спеціальності)
Тип програми ___освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)
Освітня програма _Системне проектування _
_____
(повна назва освітньої програми)

Керівник __доц. каф. КМІТ Тетяна БІЛОВА___
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри _____ _____ ____Ігор ГРЕБЕННІК_____
(підпис) (власне ім’я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп’ютерних наук_____

Кафедра__Комп'ютерного моделювання та інтелектуальних технологій_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____122 – Комп’ютерні науки_____

(код і повна назва)

Тип програми _____освітньо-наукова _____

(освітньо-професійна або освітньо-наукова)

Освітня програма_____Системне проектування_____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____»_____ 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві_____Гуньку Борису Геннадійовичу_____

(прізвище, ім’я, по батькові)

1. Тема роботи _Дослідження та порівняльний аналіз методів аварійного відновлення систем в хмарних середовищах_____

Затверджена наказом університету від _06__ ____04_____2026 р. № _628Ст____

2. Термін подання здобувачем роботи до екзаменаційної комісії _14_ травня 2026 р.

3. Вихідні дані до роботи Розробити методику економічно обґрунтованого вибору стратегії аварійного відновлення для хмарних інформаційних систем різного рівня критичності на основі моделі сукупної вартості володіння, очікуваних річних втрат та аналізу точки байдужості._____

4. Перелік питань, що потрібно опрацювати в роботі _4.1 Теоретичні основи та проблематика аварійного відновлення в хмарних середовищах. 4.2. Сутність та складові аварійного відновлення та безперервності бізнесу. 4.3. Метрики оцінки ефективності відновлення. 4.4. Модель розподіленої відповідальності. 4.5. Методи аварійного відновлення даних та інфраструктури в хмарних середовищах. 4.6. Постановка задачі. 4.8. Аналіз механізмів реплікації даних. 4.9. Аналіз моделей відновлення інфраструктури. 4.10. Автоматизація DR. 4.11. Порівняльний аналізу сервісів аварійного відновлення провідних хмарних провайдерів. 4.12. Розробка моделі сценарного економіко-архітектурного моделювання DR-стратегій. 4.13. Проведення моделювання сукупної вартості володіння (ТСО). 4.14. Розробка методології оцінки економічної доцільності вибору стратегій 4.15. Аналіз балансу між витратами на інфраструктуру та ризиками втрат від простою систем. 4.16. Визначення оптимальної вартості простою для вибору стратегії.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп’ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 5.1 Співвідношення метрик аварійного відновлення між собою. 5.2 Графік залежності витрат на підтримання працездатності інфраструктури в разі інциденту та вартості простою систему від

показників RTO та RPO. 5.3 Схема синхронної реплікації даних. 5.4 Схема асинхронної реплікації даних, 5.5 Схема напівсинхронної реплікації даних. 5.6 Структура витрат стратегії Backup and Restore. 5.7 Структура витрат стратегії Pilot Light. 5.8 Структура витрат стратегії Warm Standby. 5.9 Структура витрат стратегії Multisite Active/Active. 5.10 Витрати на DR-стратегії. 5.11 Лінійні залежності сумарних витрат стратегій DR від вартості години простою.

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
Кваліфікаційна робота	Доц. каф. КМІТ Білова Т.Г.		

Календарний план

№	Назва етапів роботи	Строк / термін Виконання етапів роботи	Примітка
1.	Отримання завдання атестаційної роботи	26.01.2026	Виконано
2.	Аналіз завдання, літератури та аналогів з теми кваліфікаційної роботи	28.02.2026-04.03.2026	Виконано
3.	Формулювання цілей дослідження та постановки задач	05.03.2026-08.03.2026	Виконано
4.	Аналіз існуючих підходів до аварійного відновлення систем	10.03.2026-19.03.2026	Виконано
5.	Аналіз рішень аварійного відновлення в хмарних середовищах різних провайдерів	20.03.2026-03.04.2025	Виконано
6.	Виконання TCO-моделювання	04.04.2026-15.03.2026	Виконано
7.	Аналіз економічної доцільності стратегій аварійного відновлення	15.03.2026-17.04.2026	Виконано
8.	Аналіз оптимальної вартості стратегій аварійного відновлення	19.04.2026-05.05.2026	Виконано
9.	Оформлення пояснювальної записки	06.05.2026	Виконано
10	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захист	11.05.2026	Виконано
11.	Представлення на рецензування	10.05.2026	Виконано
12.	Представлення атестаційної роботи в ДЕК	14.05.2026	Виконано

Дата видачі завдання 26 січня 2026 р.

Здобувач _____ Гунько Б. Г.
(підпис)

Керівник роботи _____ _____ доц. каф. КМІТ Тетяна БІЛОВА
(підпис) (посада, власне ім'я, прізвище

РЕФЕРАТ

Кваліфікаційна робота: 92 с., 21 табл., 11 рис., 33 джерел інформації.

АВАРІЙНЕ ВІДНОВЛЕННЯ, БЕЗПЕРЕВНІСТЬ БІЗНЕСУ, ЗАГАЛЬНА ВАРТІСТЬ ОБСЛУГОВУВАННЯ, РЕЗЕРВНЕ КОПІЮВАННЯ ДАНИХ, РЕПЛІКАЦІЯ ДАНИХ, ЦІЛЬОВА ТОЧКА ВІДНОВЛЕННЯ, ЦІЛЬОВИЙ ЧАС ВІДНОВАЛЕННЯ,.

У роботі досліджено методи аварійного відновлення систем. Об’єктом є архітектури, що забезпечують безперервність роботи за умов навантаження та відмов, предметом – підходи до побудови стійких до збоїв систем. Об'єктом дослідження є процес планування і реалізації аварійного відновлення інформаційних систем у хмарних середовищах.

Предметом дослідження є аналіз існуючих методів, технологій та архітектурних підходів до аварійного відновлення інфраструктури та збереження даних.

Метою є комплексне дослідження та порівняльний аналіз методів і технологій аварійного відновлення систем у хмарних середовищах з метою визначення методів для забезпечення мінімальних показників цільового часу відновлення та цільової точки відновлення.

Методи дослідження включають теоретичний аналіз існуючих підходів аварійного резервування даних в хмарних середовищах, огляд хмарних реалізацій методі аварійного відновлення даних в хмарних середовищах, а також вивчення методів вибору оптимальних стратегій.

У роботі досліджено теоретичні засади та проблематику забезпечення доступності систем під час катастрофічних збоїв. Проаналізовано ключові метрики ефективності відновлення та класифіковано загрози хмарній інфраструктурі. Проведено комплексний порівняльний аналіз базових стратегій аварійного відновлення інфраструктури. Досліджено роль автоматизації процесу аварійного відновлення. Виконано порівняння хмарних рішень від різни провайдерів.

Рекомендовано застосовувати результати для розробки веб-застосунків, хмарних сервісів, корпоративних систем, що потребують надійності роботи та стійкості до аварій.

ABSTRACT

Qualification work: 92 p., 21 tab., 11 fig., 33 sources.

BUSINESS CONTINUITY, DATA BACKUP, DATA REPLICATION, DISASTER RECOVERY, RECOVERY POINT OBJECTIVE, RECOVERY TIME OBJECTIVE, TOTAL COST OF OWNERSHIP.

The thesis investigates methods for system disaster recovery. The object of the research is the processes of disaster recovery of systems in cloud environments. The subject of the research is the process of planning and implementation of distater recovery in cloud environments.

The goal is a comprehensive study and comparative analysis of disaster recovery methods and technologies in cloud environments to determine optimal approaches for achieving minimal Recovery Time Objective (RTO) and Recovery Point Objective (RPO) targets.

Research methods include a theoretical analysis of existing approaches to data disaster redundancy in cloud environments, an overview of cloud-native implementations of disaster recovery, and the study of methods for selecting optimal strategies.

The work explores the theoretical foundations and challenges of ensuring system availability during catastrophic failures. Key recovery efficiency metrics are analyzed, and threats to cloud infrastructure are classified. A comprehensive comparative analysis of baseline infrastructure disaster recovery strategies is conducted. The role of disaster recovery process automation is examined, and cloud solutions from various providers are compared.

It is recommended to apply the results to the development of web applications, cloud services, and enterprise systems that require high reliability and disaster resilience.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8

Вступ 9

1 Теоретичні основи та проблематика аварійного відновлення в хмарних середовищах 11

1.1 Сутність та складові аварійного відновлення та безперервності бізнесу 11

1.2 Класифікація загроз та ризиків 13

1.3 Метрики оцінки ефективності відновлення 15

1.4 Модель розподіленої відповідальності в хмарних середовищах 17

1.5 Методи аварійного відновлення даних та інфраструктури в хмарних середовищах та їх проблеми 19

1.6 Постановка задачі 21

2 Аналіз методів та технологій аварійного відновлення 23

2.1 Дослідження та порівняння методів резервного копіювання даних у хмарних середовищах 23

2.2 Аналіз механізмів реплікації даних та їх вплив на цільові показники disaster recovery 26

2.3 Порівняльний аналіз моделей відновлення інфраструктури: Backup & Restore, Pilot Light, Warm Standby, Multi-Site Active/Active 30

2.4 Автоматизація процесу аварійного відновлення 34

3 Порівняльний аналіз DR-сервісів провідних хмарних провайдерів 37

3.1 Резервне копіювання та незмінність даних у провайдерів 37

3.2 Сервіси аварійного відновлення з блочною реплікацією 42

3.3 Реплікація керованих баз даних 47

3.4 Маршрутизація трафіку і механіка failover 50

4 Розробка підходу для вибору стратегії аварійного відновлення 53

4.1 Класифікація систем за критичністю та визначення базових показників для обчислення вартості 53

4.2 Моделювання загальної вартості утримання чотирьох стратегій аварійного відновлення на платформі AWS 56
4.2.1 Структура витрат компонентів аварійного відновлення на AWS 56
4.2.2 Розрахунок загальної вартості утримання для стратегії Backup and Restore 59
4.2.3 Розрахунок загальної вартості утримання для стратегії Pilot Light 61
4.2.4 Розрахунок загальної вартості утримання для стратегії Warm Standby 64
4.2.5 Розрахунок структури витрат для Multi-Site Active/Active 67
4.3 Аналіз економічної доцільності стратегій авайного відновлення 71
4.3.1 Методологія розрахунку очікуваних втрат 71
4.3.2 Розрахунок очікуваної річної втрати та сумарної вартості стратегії 73
4.3.3 Визначення оптимальної вартості простою для вибору стратегії 74
4.4 Проблеми та особливості використання підходу 78
Висновки 80
Перелік джерел посилань 82
Додаток А Графічний матеріал кваліфікаційної роботи 86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AWS – Amazon Web Services – Хмарні сервіси Amazon

ALE – Annualized Loss Expectancy – Очікувані річні втрати

ARO – Annual Rate of Occurrence – Річна частота виникнення

BCP – Business Continuity Planning – Планування безперервності бізнесу

BIA – Business Impact Analysis – Аналіз впливу на бізнес

CMEK – Customer-Managed Encryption Keys – Ключі шифрування, керовані клієнтом

Draas – Disaster Recovery as a Service – Аварійне відновлення як послуга

GCP – Google Cloud Platform – Хмарна платформа Google

GFS – Grandfather-Father-Son – Схема ротації резервних копій «дід-батько-син»

GSLB – Global Server Load Balancing – Глобальне балансування серверного навантаження

I/O – Input/Output – Введення/виведення

Iac – Infrastructure as Code – Інфраструктура як код

IAM – Identity and Access Management – Керування ідентифікацією та доступом

NAT – Network Address Translation – Перетворення мережевих адрес

OLTP – Online Transaction Processing – Оперативна обробка транзакцій

OPEX – Operating Expenses – Операційні витрати

PITR – Point-In-Time Recovery – Відновлення на момент часу

RBAC – Role-Based Access Control – Контроль доступу на основі ролей

RPO – Recovery Point Objective – Цільова точка відновлення

RTO – Recovery Time Objective – Цільовий час відновлення

TCO – Total Cost of Ownership – Сукупна вартість володіння

VPC – Virtual Private Cloud – Віртуальна приватна хмара

WAF – Web Application Firewall – Мережевий екран веб-застосунків

WORM – Write Once Read Many – Запис один раз, читання багато разів

WRT – Work Recovery Time – Час повернення до повноцінної роботи

ВСТУП

Сучасна цифрова інфраструктура вимагає доступності, що вимірюється не годинами, а секундами. Бізнес все частіше віддає перевагу хмарним рішення через їх гнучкість, масштабованість та можливості економії коштів. Міграція до хмари не усуває ризики, а лише частково їх мінімізує. В умовах повномасштабного вторгнення Російської Федерації в Україну і постійних кібератак на критично важливу інфраструктуру проблема аварійного відновлення систем стоїть особливо гостро як для великих бізнесів, так і для державних органів. Крім того важливо гарантувати захист від критичних помилок конфігурації, неполадок мережі або апаратних збоїв на боці провайдера хмарних послуг.

Актуальність цього дослідження полягає в визначенні найбільш ефективних підходів до аварійного відновлення систем, які забезпечують їхню стабільну роботу та збереження даних навіть у разі критичних збоїв. Порівняльний аналіз сучасних методів і технологій, таких як резервне копіювання, реплікація баз даних, автоматизоване розгортання інфраструктури та мультихмарні сервіси, дозволяє знаходити ефективні та гнучкі рішення для підтримки надійності та швидкого відновлення систем у хмарних середовищах з урахуванням фінансової складової кожного.

Метою кваліфікаційної роботи є дослідження та порівняльний аналіз методів аварійного відновлення інформаційних систем у хмарних середовищах із розробкою методики обґрунтованого вибору DR-стратегії для систем різного рівня критичності. Для досягнення цієї мети в роботі поставлено наступні завдання: проаналізувати теоретичні основи аварійного відновлення та класифікувати загрози для хмарних інформаційних систем; виконати порівняльний аналіз методів резервного копіювання, реплікації даних та архітектурних моделей відновлення інфраструктури; провести порівняння DR-сервісів провідних хмарних провайдерів за функціональним складом та

сценаріями застосування; розробити модель розрахунку сукупної фінансової доцільності стратегій аварійного відновлення.

Для реалізації дослідницьких завдань у роботі застосовано методи системного аналізу, порівняльного аналізу архітектурних моделей і керованих DR-сервісів хмарних провайдерів, кількісного моделювання сукупної вартості володіння, методики ризик-менеджменту для оцінки очікуваних річних втрат, аналізу точки байдужості для зіставлення альтернативних стратегій з різною структурою витрат.

Наукова новизна роботи полягає у запропонованій методиці економічно обґрунтованого вибору DR-стратегії для хмарних інформаційних систем, що поєднує розрахунок сукупної вартості володіння з кількісною оцінкою очікуваних втрат від простою через метрику ALE. Вперше для аналізу DR-стратегій застосовано метод аналізу точки байдужості, який дозволяє визначити для кожної стратегії діапазон вартості простою, у межах якого вона є економічно оптимальною. Це створює універсальний інструмент вибору, що може бути застосований до будь-якої організації без необхідності повного перерахунку балансу TCO та ALE для всіх стратегій..

Практичне значення одержаних результатів полягає у можливості використання запропонованої методики при плануванні аварійного відновлення для хмарних інформаційних систем різних класів критичності. Модель розрахунку TCO та діапазонів економічної оптимальності для чотирьох DR-стратегій може бути адаптована до конкретного навантаження організації шляхом підстановки власних параметрів вартості інфраструктури та вартості простою.

Результати кваліфікаційної роботи апробовано у вигляді тез доповідей на 30-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» [1].

1 ТЕОРЕТИЧНІ ОСНОВИ ТА ПРОБЛЕМАТИКА АВАРІЙНОГО ВІДНОВЛЕННЯ В ХМАРНИХ СЕРЕДОВИЩАХ

Забезпечення високої доступності та безперервності роботи є фундаментальною вимогою до архітектури сучасних інформаційних систем. Еволюція обчислювальних інфраструктур і масовий перехід від локальних центрів обробки даних до хмарних середовищ кардинально змінили підходи до забезпечення відмовостійкості. Традиційні парадигми вимагали значних капітальних інвестицій у створення резервних апаратних потужностей, які простоювали більшу частину часу, а також супроводжувалися тривалими процесами фізичного відновлення. Хмарні технології, дозволяють динамічно розгортати ресурси. Проте ця гнучкість породжує нові класи архітектурних ризиків, зумовлених складністю управління конфігураціями, розподіленою природою хмарних сервісів та еволюцією кіберзагроз.

1.1 Сутність та складові аварійного відновлення та безперервності бізнесу

Забезпечення надійності інформаційних систем спирається на процеси безперервності бізнесу (Business Continuity, BC) та аварійного відновлення (Disaster Recovery, DR). Business Continuity – це здатність організації підтримувати надання послуг на заздалегідь визначеному прийнятному рівні після настання аварійного інциденту [2]. Це комплексне поняття, що охоплює управління ризиками, кризові комунікації, логістику, запобігання загрозам, виявлення, відновлення та поновлення операцій у разі інциденту. Аварійне відновлення є підмножиною поняття безперервності бізнесу, що фокусується виключно на відновленні працездатності IT-інфраструктури, телекомунікацій та цілісності даних після їхньої відмови. Аварійне відновлення включає в себе набір політик, практик, процесів для відновлення, метою яких є мінімізація негативного впливу катастрофи на бізнес шляхом швидкого та впорядкованого

відновлення критично важливих технологічних активів, часто в альтернативному середовищі.

У процесі архітектурного проєктування важливо розмежовувати високу доступність (High Availability, HA) та аварійне відновлення (DR). Технології HA впроваджуються для забезпечення стійкості системи до рутинних локальних збоїв, таких як відмова окремого сервера, комутатора чи контейнера, за рахунок надлишковості компонентів та балансування навантаження. Висока доступність забезпечує безперебійну роботу в штатному режимі. Аварійне відновлення розробляється для подолання наслідків глобальних катастроф, що призводять до повної недоступності обчислювального середовища або регіонального дата-центру.

Процес планування аварійного відновлення можна розділити на послідовні етапи [2]:

– розробка політики планування (Policy Statement). Формалізація повноважень, виділення бюджетів та призначення відповідальних осіб на рівні вищого керівництва організації;

– проведення аналізу впливу на бізнес (Business Impact Analysis, BIA). Ідентифікація та пріоритезація інформаційних систем на основі їхньої критичності для бізнес-процесів;

– визначення превентивних заходів (Preventive Controls). Впровадження механізмів, що знижують ймовірність настання системних збоїв та зменшують загальну вартість життєвого циклу відновлення;

– розробка стратегій відновлення (Recovery Strategies). Вибір технологічних рішень для швидкого відновлення IT-операцій на основі визначених під час BIA метрик;

– розробка плану аварійного відновлення (Data recovery plan). Створення детального документа з покроковими інструкціями щодо відновлення систем;

– тестування та навчання (Testing and Training). Регулярна перевірка ефективності плану шляхом симуляцій та практичних навчань персоналу;

– підтримка та актуалізація плану (Plan Maintenance). Безперервний процес оновлення документації відповідно до змін в IT-інфраструктурі та бізнес-процесах.

Найважливішим елементом цього життєвого циклу є аналіз впливу на бізнес (BIA). Процес BIA вимагає проведення глибокої інвентаризації активів та визначення інформаційних систем із критичними бізнес-функціями. Під час BIA аналізуються наслідки від простою систем у розрізі конфіденційності, цілісності та доступності [3]. На основі зібраних даних формується кількісна оцінка допустимого часу простою для кожного процесу, що є основою для визначення технічних цілей, числових показників відновлення та обґрунтування витрат на резервну інфраструктуру. Відсутність коректно проведеного BIA призводить до неефективного розподілу ресурсів: надмірних витрат на некритичні системи або недостатнього резервування критичних

1.2 Класифікація загроз та ризиків

Види загроз у хмарних обчисленнях суттєво відрізняється від традиційних локальних інфраструктур через розподілену архітектуру, мультитенантність та складність ієрархії рівнів доступу. Загрози доступності інформаційних систем класифікуються за кількома основними напрямами, кожен з яких вимагає специфічних механізмів протидії [4].

Апаратні збої та системні помилки інфраструктури провайдера є загрозою, коли через проблеми на стороні провайдера хмарних інформаційні системи не мають змоги коректно працювати. Хмарні платформи не є застрахованими від глобальних відмов. Збої часто виникають не через фізичне руйнування обладнання, а внаслідок алгоритмічних помилок у площині управління, збоїв у системах балансування навантаження, багів у вихідному коді хмарних сервісів або у підсистемах маршрутизації DNS [4]. Такі інциденти можуть паралізувати цілі регіони, викликаючи розгортання ресурсів в межах кількох зон доступності одного регіону та вимагаючи побудови інфраструктури в кількох регіонах.

Кіберзагрози та атаки на ланцюги постачання є типом загрози, коли через втручання шкідливих програм відбувається паралізація інформаційних системи бізнесу. Актуальними векторами є цілеспрямовані атаки з використанням програм-вимагачів та експлуатація вразливих API [4]. Сучасні програми-вимагачі перед початком шифрування виробничих даних здійснюють пошук та знищення ланцюжків резервних копій. Ефективне відновлення за таких умов можливе лише за наявності криптографічно незмінних резервних копій та ізольованих сховищ. Крім того, зростає кількість інцидентів, пов’язаних із вразливостями в ланцюгах постачання програмного забезпечення.

Проблеми управління ідентифікацією та доступом, IAM, може завдати значної шкоди, коли користувач з високим рівнем доступу до системи може знищувати або компроментувати ресурсу в хмарі. Некоректне налаштування політик IAM та надмірне надання привілеїв є провідною причиною інцидентів у хмарі. Відсутність багатофакторної автентифікації або компрометація токенів доступу дозволяє зловмисникам видаляти критичні ресурси, бази даних та резервні зліпки. Нерідко компрометація даних відбувається зі сторони акторів всередині системи, які мають завеликі права доступу.

Помилки конфігурації та дрейф інфраструктури можуть паралізувати інфраструктуру, коли через людську помилку можуть переписуватись або знищуватись критично важливі ресурсу в хмарі. Хмарні середовища характеризуються високою динамічністю. Регулярні зміни налаштувань груп безпеки, політик доступу до об’єктних сховищ та параметрів віртуальних мереж створюють ризик внесення помилок. Некоректна конфігурація може призвести до порушення зв’язку між мікросервісами, видалення критично важливих вузлів системи, повної ізоляції бази даних або, навпаки, до ненавмисного публічного розкриття корпоративної інформації.

1.3 Метрики оцінки ефективності відновлення

В основі будь-якого плану аварійного відновлення лежать вимірювані технічні метрики, які безпосередньо впливають на архітектуру системи та визначають її бюджет.

Recovery Point Objective (RPO), цільова точка відновлення – це метрика, яка визначає максимально допустимий обсяг даних, який організація може втратити внаслідок інциденту [5]. Значення RPO вимірюється в часовому еквіваленті у зворотному напрямку: від моменту збою до часу створення останньої резервної копії або транзакції реплікації.5 Значення RPO є визначальним фактором для вибору технології збереження: високі значення дозволяють використовувати періодичні зліпки, тоді як значення, близькі до нуля, вимагають безперервної синхронної реплікації [6].

Recovery Time Objective (RTO), цільовий час відновлення – це метрика, яка визначає максимальний проміжок часу, протягом якого IT-система повинна бути відновлена та повернена до нормального функціонування після виникнення збою від моменту виникнення інциденту [5]. RTO диктує вибір інфраструктурної моделі: від «холодного» резерву, повільне розгортання, до «гарячого», миттєве перемикання трафіку на паралельний кластер.

Maximum Time of Downtime (MTD) – це максимально допустимий період переривання. Це абсолютна верхня часова межа, протягом якої бізнес-процес може залишатися недоступним до настання незворотних фінансових або репутаційних втрат [2]. MTD формує загальні рамки для всіх технічних і організаційних процесів відновлення.

Work Recovery Time (WRT), час відновлення роботи – це тривалість часу, необхідного бізнес-підрозділам для верифікації цілісності відновлених даних, тестування бізнес-логіки та ручного введення транзакцій, втрачених за період RPO, після того, як IT-відділ досягнув показника RTO.

Service Delivery Objective (SDO). Рівень надання послуг, який повинен підтримуватися в альтернативному (резервному) режимі роботи до моменту

повного усунення наслідків катастрофи [2]. Перехід від альтернативного режиму до штатного обмежується параметром Maximum Tolerable Outage (MTO).

На рисунку 1.1 можна побачити, як метрики співвідносяться між собою, і в за який етап інциденту вони відповідають.

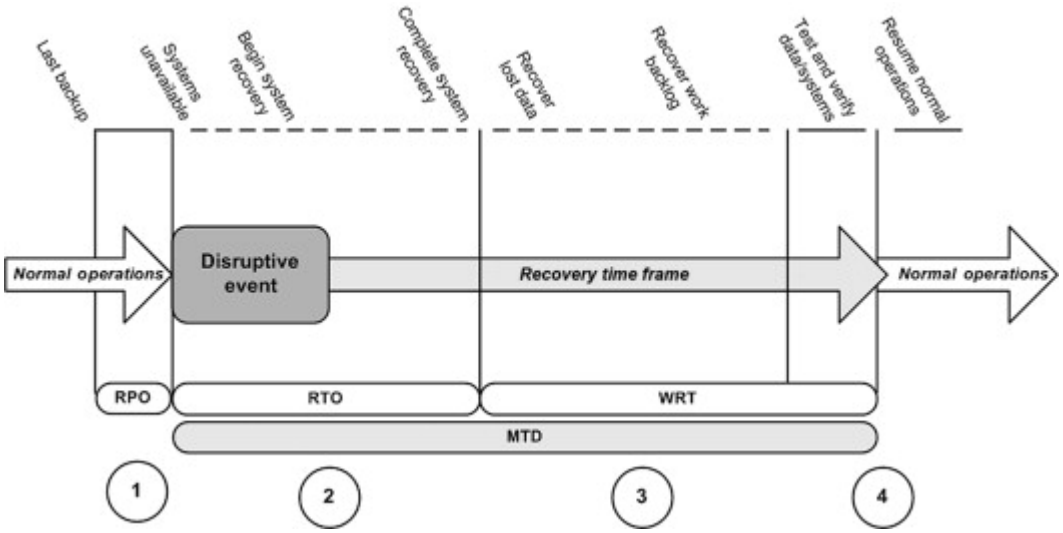

Рисунок 1.1 – Співвідношення метрик аварійного відновлення

Існує пряма залежність між зменшенням цільових значень (RTO/RPO) та зростанням вартості інфраструктури. Наближення RTO до нуля вимагає переходу на архітектуру Multi-Site Active/Active, що передбачає повне дублювання обчислювальних потужностей та використання дорогих каналів зв'язку з низькою затримкою. При розробці DRP необхідно знайти баланс між витратами на підтримання інфраструктури та показниками RTO/RPO. Оптимальний показник RTO визначається в точці перетину кривої вартості простою системи, яка зростає з часом простою, та кривої вартості рішень для відновлення, яка зростає при зменшенні часу.

Середні значення вартості години простою суттєво варіюються залежно від галузі та масштабу організації, а також від часу коли стався збій системи.

На рисунку 1.2 зображений графік залежності витрат на підтримання працездатності інфраструктури в разі інциденту та вартості простою систему від показників RTO та RPO.

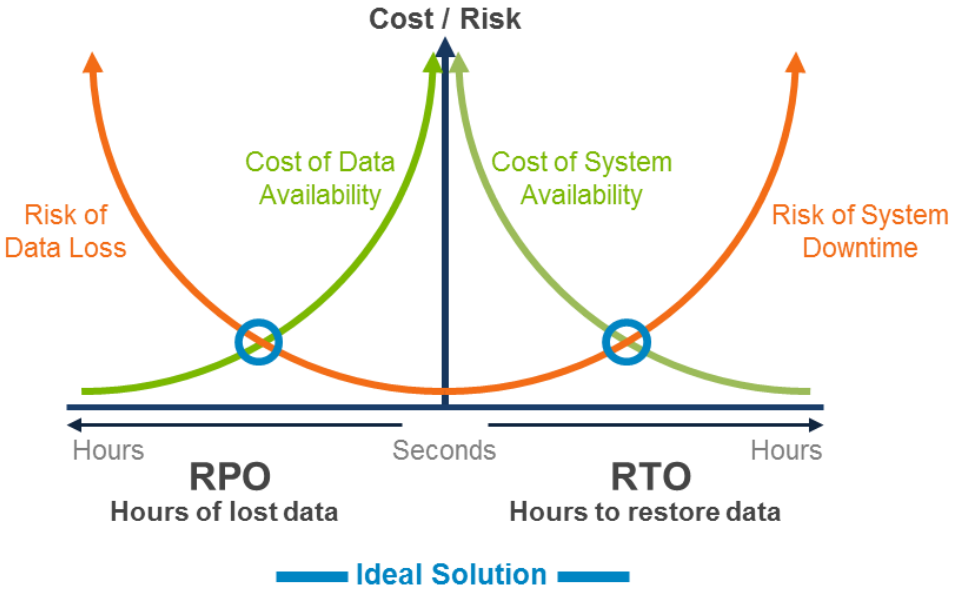

Рисунок 1.2 – Графік залежності витрат на підтримання працездатності інфраструктури в разі інциденту та вартості простою систему від показників RTO та RPO

З метою оптимізації витрат додатки розділяють на рівні критичності [6]:

– рівень 1, критичні системи: RTO до 15 хвилин, RPO близьке до нуля;

– рівень 2, важливі системи: RTO до 4 годин, RPO до 2 годин;

– рівень 3, некритичні системи: RTO 8–24 години, RPO 4 години і більше.

1.4 Модель розподіленої відповідальності в хмарних середовищах

Модель розділеної відповідальності – це концепція, що регулює безпеку та процеси аварійного відновлення у хмарі [7]. Вона розділяє зони відповідальності між провайдером надання хмарних послуг та клієнтом-споживачем послуг, зони відповідальності можуть змінюватись залежно від обраної моделі надання хмарних сервісів.

Модель інфраструктури як послуги (iaas) полягає в тому, що провайдер несе відповідальність виключно за базову інфраструктуру – фізичну безпеку датацентрів, апаратне забезпечення серверів, дискові масиви та гіпервізор. Клієнт бере на себе відповідальність за гостьові операційні системи,

конфігурацію віртуальних мереж, управління доступом користувачів, розгортання додатків та за повний цикл резервного копіювання і відновлення даних [7].

Модель платформи як послуги (paas) полягає в тому, що провайдер додатково бере на себе обслуговування операційних систем, середовищ виконання та базового мережевого стеку [7]. Клієнт залишається відповідальним за налаштування безпеки на рівні додатків, управління ідентифікацією та захист своїх даних.

Модель програмне забезпечення як послуга (saas) полягає в тому, що провайдер забезпечує повну працездатність додатку та інфраструктури [7]. Клієнт відповідає лише за конфігурацію політик доступу користувачів, класифікацію даних та їхнє резервне копіювання від логічних пошкоджень або випадкового видалення.

Спільним знаменником для всіх моделей є те, що управління даними та забезпечення їхнього резервного копіювання завжди залишається в зоні відповідальності клієнта.

У таблиці 1.1 показано зони відповідальності сторін при різних моделях обчислення в хмарах.

Таблиця 1.1 – Зони відповідальності сторін в моделях

Responsibility area	Iaas	Paas	Saas
Клієнтські дані	Клієнт	Клієнт	Клієнт
Налаштування та конфігурації	Клієнт	Клієнт	Клієнт
Identities and users	Клієнт	Клієнт	Клієнт
Клієнтські пристрої	Клієнт	Клієнт	Спільна
Застосунки	Клієнт	Спільна	Спільна

Кінець таблиці 1.1

Responsibility area	Iaas	Paas	Saas
Контроль мережі	Клієнт	Спільна	Хмарний провайдер
Операційна система	Клієнт	Хмарний провайдер	Хмарний провайдер
Physical hosts	Хмарний провайдер	Хмарний провайдер	Хмарний провайдер
Фізична мережа	Хмарний провайдер	Хмарний провайдер	Хмарний провайдер
Фізичний датацентр	Хмарний провайдер	Хмарний провайдер	Хмарний провайдер

1.5 Методи аварійного відновлення даних та інфраструктури в хмарних середовищах та їх проблеми

Для збереження стану системи використовується механізм снепшотів віртуальних сховищ. На відміну від традиційного файлового копіювання, блочні снепшоти фіксують стан на рівні фізичних блоків жорсткого диска. Такі зліпки є інкрементальними, наприклад, на основі алгоритмів Copy-on-Write або Redirect-on-Write, тобто копіюються лише ті блоки даних, які були змінені з моменту попереднього зліпка. Це дозволяє суттєво збільшити швидкість створення копій, зменшити навантаження на підсистему вводу-виводу та мінімізувати обсяги споживаного мережевого трафіку.

У контексті хмарних обчислень також виникає складна проблематика відмінності між відновленням даних та відновленням інфраструктури. Наявність консистентного зліпка бази даних гарантує виконання RPO, але є недостатньою для досягнення RTO [8]. Для відновлення сервісу необхідно повністю відбудувати інфраструктуру системи у резервному регіоні: створити віртуальні приватні мережі, налаштувати таблиці маршрутизації, відтворити складну ієрархію ролей доступу, розгорнути контейнери та ініціалізувати балансувальники навантаження.

Відсутність автоматизації цього процесу робить швидке відновлення неможливим.

Необхідність швидкого розгортання сотень взаємопов’язаних інфраструктурних компонентів під час аварії зумовлює використання підходу Інфраструктура як код (Infrastructure as Code, iac). Застосування інструментів iac, таких як Terraform або Ansible, дозволяє описувати інфраструктуру у вигляді декларативного коду, що забезпечує версіонування, повторюваність та можливість розгортання резервного середовища без ручного втручання. Проте ефективність цього підходу критично підривається явищем неконтрольованих змін [9].

Важливо розрізняти поняття інфраструктурний дрейф та дрейф конфігурацій. Інфраструктурний дрейф пов’язаний зі структурними розбіжностями – створенням або видаленням цілих ресурсів (віртуальних машин, баз даних) поза межами пайплайнів iac, що призводить до появи некерованих «сирітських» об’єктів [10].

Дрейф конфігурацій означає непомітне відхилення внутрішніх параметрів та атрибутів вже існуючих ресурсів від їхнього задекларованого стану у коді. Це відбувається, коли адміністратори вносять екстрені виправлення через веб-консоль для вирішення інцидентів, таких як зміна правил безпеки, розширення лімітів пам’яті або зміна змінних середовища, і не реплікують ці зміни назад у вихідний код [10].

Наслідком накопичення дрейфу конфігурацій є асиметрія відновлення. В умовах інциденту процес технічного відновлення масивів даних відбувається автоматизовано, швидко та прогнозовано. Проте, коли паралельно запускаються скрипти iac для створення обчислювального середовища, вони розгортають застарілу, еталонну конфігурацію, яка не містить актуальних, змін, які були зробленій вручну. В результаті нові версії розгорнутих сервісів не можуть запуститися, з’єднатися з базами даних або коректно обробити трафік через відсутність необхідних мережевих дозволів чи змінених параметрів доступу.

1.6 Постановка задачі

Об’єктом досліджень є процеси аварійного відновлення систем в хмарних середовищах.

Предметом дослідження є аналіз існуючих методів, технологій та архітектурних підходів до аварійного відновлення інфраструктури та збереження даних.

Проблема дослідження полягає у тому, що в умовах зростаючої складності хмарних архітектур та збільшення кількості кіберзагроз виникає значна різниця між швидкістю відновлення масивів даних та готовністю обчислювальної інфраструктури до їх обробки, а також суттєве збільшення витрат на підтри. Втрата доступності або тривалі простої призводять до значних фінансових та репутаційних втрат для бізнесу. Водночас традиційні підходи до аварійного відновлення виявляються недостатньо ефективними через явище дрей конфігурацій та необхідність ручного повторного налаштування складних мережевих топологій під час інцидентів. Тому необхідно комплексно дослідити сучасні методи аварійного відновлення, проаналізувати інструменти автоматизації інфраструктури та визначити оптимальні рішення, що дозволять мінімізувати цільовий час відновлення та цільову точку відновлення за умови забезпечення високої надійності та оптимізації вартості зберігання даних.

Задачі дослідження:

– дослідити сучасні методи резервного копіювання даних та механізми реплікації (синхронна та асинхронна), оцінити їхню надійність та вплив на збереження цілісності даних, а також проаналізувати підходи до оптимізації вартості зберігання резервних копій у хмарі;

– провести порівняльний аналіз базових стратегій відновлення інфраструктури: Backup & Restore, Pilot Light, Warm Standby, Multi-Site Active/Active, в контексті балансу між вартістю утримання та швидкістю розгортання;

– вивчити процеси автоматичного перемикання (Failover) та повернення (Failback) у розподілених архітектурах для забезпечення безперервного обслуговування трафіку;

– розглянути підходи до масштабування системи, зокрема горизонтальне та вертикальне масштабування, та оцінити їхній вплив на продуктивність і стійкість веб-застосунків до навантаження;

– проаналізувати можливості підходу «Інфраструктура як код» (iac) в усуненні проблеми розсинхронізації конфігурацій та забезпеченні детермінованого розгортання резервного середовища без ручного втручання;

– виконати зведену порівняльну оцінку досліджених методів та технологій для формування висновків щодо вибору оптимальних архітектурних рішень залежно від рівня критичності інформаційних систем;

– порівняти сервіси аварійного відновлення

– розробити практичні рекомендації щодо вибору ефективних рішень для впровадження плану аварійного відновлення.

2 АНАЛІЗ МЕТОДІВ ТА ТЕХНОЛОГІЙ АВАРІЙНОГО ВІДНОВЛЕННЯ

Забезпечення швидкого та надійного відновлення інформаційних систем є критично важливим завданням, яке вимагає ретельного проєктування та використання сучасних технологічних рішень. Відмова окремих компонентів, перевантаження або непередбачені збої на рівні дата-центрів можуть негативно вплинути на функціонування системи, викликаючи зупинку бізнес-процесів та фінансові втрати. Саме тому під час проєктування архітектури особлива увага приділяється стратегіям, що дозволяють мінімізувати час простою, автоматично усувати наслідки збоїв та ефективно керувати резервними ресурсами. Після визначення метрик ефективності (RTO та RPO) та класифікації архітектурних ризиків, виникає необхідність детального аналізу конкретних технічних інструментів. Теоретичні моделі аварійного відновлення вимагають практичної реалізації за допомогою механізмів, які пропонують сучасні провайдери хмарних послуг. Оптимальний вибір технологічного стека безпосередньо впливає на здатність організації виконати вимоги щодо безперервності бізнесу в умовах жорстких фінансових та часових обмежень.

2.1 Дослідження та порівняння методів резервного копіювання даних у хмарних середовищах

Резервне копіювання залишається невід’ємною частиною будь-якої стратегії аварійного відновлення.

Повне копіювання передбачає збереження всього масиву даних незалежно від того, чи були вони змінені з моменту попередньої операції. Цей метод вимагає найбільше часу та дискового простору, створюючи значне навантаження на мережу, проте він забезпечує найвищу швидкість відновлення, оскільки для відтворення еталонного стану потрібен лише один, цілісний файл копії [11].

Диференціальне копіювання зберігає лише ті дані, які були модифіковані з моменту останнього базового повного бекапу [11]. Час створення копії та її обсяг лінійно зростають з кожним днем після повного бекапу, але відновлення потребує застосування лише двох компонентів: останньої повної копії та останньої диференціальної.

Інкрементальне копіювання зберігає виключно ті дані, що були змінені з моменту створення будь-якої останньої копії: як повної, так і попередньої інкрементальної [11]. Цей метод є найшвидшим у створенні та найменш ресурс ємним з точки зору використання дискового простору. Однак процес відновлення є найповільнішим і найбільш вразливим, оскільки вимагає послідовного та безпомилкового застосування повної копії та всього ланцюжка інкрементальних доповнень. Корупція хоча б одного проміжного інкрементального файлу унеможливлює подальше відновлення.

Застосування зліпків на рівні блоків є значно ефективнішим для резервного копіювання. На відміну від файлового копіювання, яке вимагає звернення до файлової системи та зчитування файлу цілком, блочні зліпки виконуються на рівні апаратної абстракції. Вони фіксують стан на рівні фізичних блоків зберігання. Після створення початкового повного зліпка наступні операції використовують алгоритми копіювання під час запису (Copy-on-Write) або перенаправлення під час запису (Redirect-on-Write) [12]. Наприклад, перенесення з EBS у дешевший клас Amazon S3. Це дозволяє створювати резервні копії даних майже миттєво, без блокування операцій вводу-виводу продуктивної системи та з мінімальними витратами на зберігання в об’єктних сховищах. Ця технологія реалізована сервісах AWS Elastic Block Store (EBS), Azure Managed Disks та Google Cloud Persistent Disk.

Для транзакційних реляційних баз даних застосовується метод відновлення на заданий момент часу (Point-in-Time Recovery, PITR). Технологія PITR базується на комбінації періодичних базових зліпків та безперервного архівування журналів транзакцій (Write-Ahead Logs, WAL). У разі втрати даних, система автоматично розгортає останній базовий зліпок у новому ізольованому

екземплярі, після чого послідовно застосовує до нього всі збережені транзакції з журналу до вказаного моменту часу [13]. Це забезпечує виконання жорстких вимог щодо мінімізації RPO, дозволяючи відновити систему за мить до моменту внесення помилки. Ця технологія реалізована у Amazon RDS, Azure SQL Database, Google Cloud SQL.

Для надійності збереження резервних копій даних застосовується правило «3-2-1». Його суть полягає в наявності трьох копій даних (одна робоча та дві резервні), які зберігаються на двох різних типах носіїв, причому одна копія обов’язково повинна знаходитися на віддаленому географічному середовищі [14]. Зі зростанням ризиків компрометації, зокрема, цілеспрямованого знищення бекапів програмами-вимагачами, правило “3-2-1” розширилось в парадигму “3-2-1-1-0”. Його суть полягає в наступному: 3 копії даних загалом для мінімізації ризику одночасної втрати; 2 різні типи середовищ зберігання для уникнення апаратних вразливостей одного типу; 1 копія на віддаленому середовищі для захисту від фізичного знищення локального дата-центру; 1 копія в незмінному (immutable) або фізично ізольованому середовищі; 0 помилок під час відновлення, що досягається шляхом регулярного автоматизованого тестування цілісності бекапів та верифікації контрольними сумами [14].

Реалізація вимоги щодо незмінності даних та їх зберігання у віддаленому місці необхідне для гарантування безпеки даних в разі несанкціонованого доступу до системи зловмисниками або спроб пошкодження даних за допомогою шкідливого програмного забезпечення.

Зберігання у віддаленому місці хмарній парадигмі реалізується через географічну та логічну ізоляцію. Географічна ізоляція полягає в асинхронній реплікації резервних копій у віддалений регіон хмарного провайдера для захисту системи від глобальних збоїв хмарного провайдера або природних катастроф. Логічна ізоляція полягає в розміщенні бекапів у окремому хмарному акаунті без спільних облікових даних з основним середовищем. Такий підхід гарантує збереження інформації у випадку цілеспрямованої кібератаки, під час якої зловмисники отримують права адміністратора до основної виробничої системи.

Реалізація вимоги щодо незмінності даних реалізовуються через технології, засновані на моделі WORM (Write Once, Read Many). Цей механізми дозволяє встановити жорсткі політики зберігання даних, які забороняють зміну або видалення об’єктів протягом заздалегідь визначеного періоду часу. Важливою характеристикою цієї технології є наявність режиму суворої відповідності. У цьому режимі блокування застосовується на рівні ядра хмарної платформи і не може бути зняте жодним користувачем, включаючи адміністраторів із найвищими привілеями. Це робить бекапи захищеними від компрометації облікових записів або внутрішніх загроз. Object Lock в Amazon S3, Immutable Storage в Azure Blob Storage та Retention Policies в Google Cloud Storage реалізують цей механізм.

2.2 Аналіз механізмів реплікації даних та їх вплив на цільові показники disaster recovery

Для досягнення показників безперервності, які не можуть бути задоволені традиційним резервним копіюванням, застосовується реплікація – процес безперервного копіювання даних з основного вузла на один або кілька резервних вузлів на рівні транзакцій або блоків. Вибір між типом реплікації є надзвичайно важливим архітектурним рішенням, яке визначає компроміс між консистентністю даних, продуктивністю та географічною топологією розгортання. В контексті DR важливо розуміти, що реплікація має відбуватись в межах різних регіонів, або, принаймні, різних зон доступності для забезпечення надійності збереження даних

Синхронна реплікація гарантує, що дані записуються на первинний і вторинний носій одночасно. З точки зору архітектури це означає, що клієнтський додаток не отримує від бази даних підтвердження про успішне завершення транзакції доти, доки резервний вузол не підтвердить фізичний запис даних на свої диски [16]. Цей підхід забезпечує точну консистентність та повну відсутність втрат даних під час аварійного перемикання, гарантуючи нульовий

RPO. Однак, технологічна необхідність очікування підтвердження від віддаленого вузла створює мережеву затримку, яка сумується з часом обробки самого запиту. Зі збільшенням географічної відстані між вузлами ця затримка зростає, що призводить до значного зменшення продуктивності транзакційних систем з високим навантаженням на запис. Враховуючи фізичні обмеження швидкості передачі сигналів, у хмарних середовищах синхронна реплікація застосовується переважно локально в межах одного регіону, але між різними зонами доступності, де провайдери володіють виділеними оптоволоконними каналами зв'язку з мінімальними затримками.

Такий тип реплікації даних найбільше підходить для критично важливих систем обробки транзакцій, таких банківський білінг або системи управління ланцюгами постачання, де консистентність даних має вищий пріоритет над швидкістю відгуку.

На рисунку 2.1 зображено схему синхронної реплікації даних.

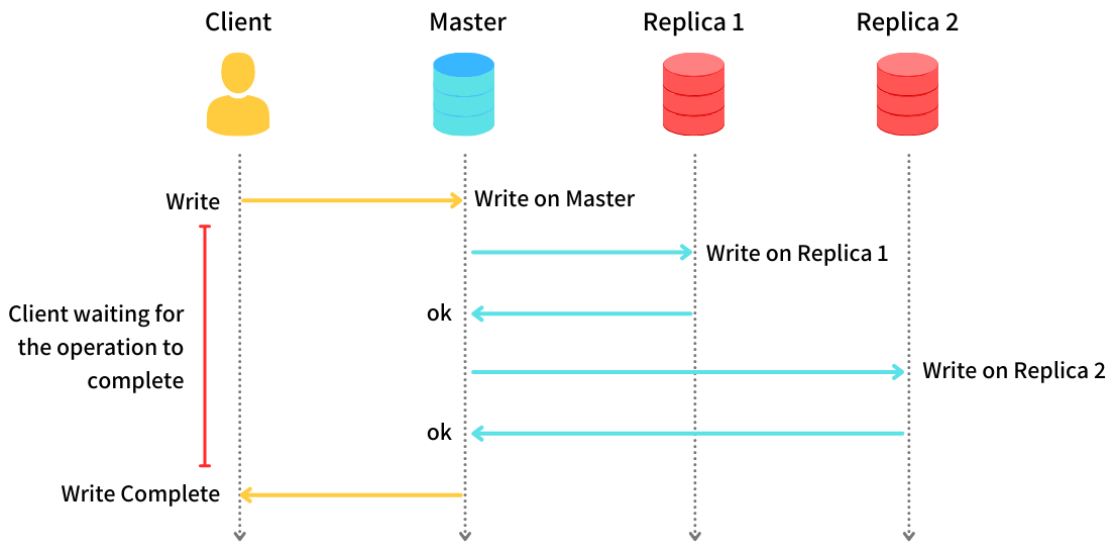

Рисунок 2.1 – Схема синхронної реплікації даних

Асинхронна реплікація нівелює проблему мережевих затримок шляхом зміни послідовності підтверджень. У цій моделі операція запису вважається успішною та підтверджується клієнтському додатку відразу після її фіксації на

первинному вузлі бази даних, а процес передачі блоків даних на резервний вузол відбувається асинхронно у фоновому режимі [16]. Такий архітектурний дизайн забезпечує максимальну продуктивність додатку при інтенсивних операціях запису та дозволяє географічно розносити центри обробки даних на великі трансконтинентальні відстані. Головним недоліком цього підходу є наявність вікна вразливості: якщо основний вузол зазнає катастрофічної відмови до того, як останні фонові транзакції будуть передані мережею, ці дані будуть безповоротно втрачені. Рівень втрат прямо пропорційний швидкості каналу зв'язку та обсягу змін. Завдяки оптимізованим протоколам хмарних сервісів, таких як Amazon Aurora Global Database, затримка асинхронної реплікації між географічно віддаленими регіонами у штатному режимі зазвичай становить менше секунди, що дозволяє досягати мінімальних показників цільової точки відновлення при збереженні глобальної продуктивності.

Такий тип реплікації даних найбільше підходить для систем аналітики великих даних, контентні платформи та другорядні сервіси, де географічна розподіленість та загальна швидкодія системи є критичнішими за втрату останніх даних.

На рисунку 2.2 зображено схему асинхронної реплікації даних.

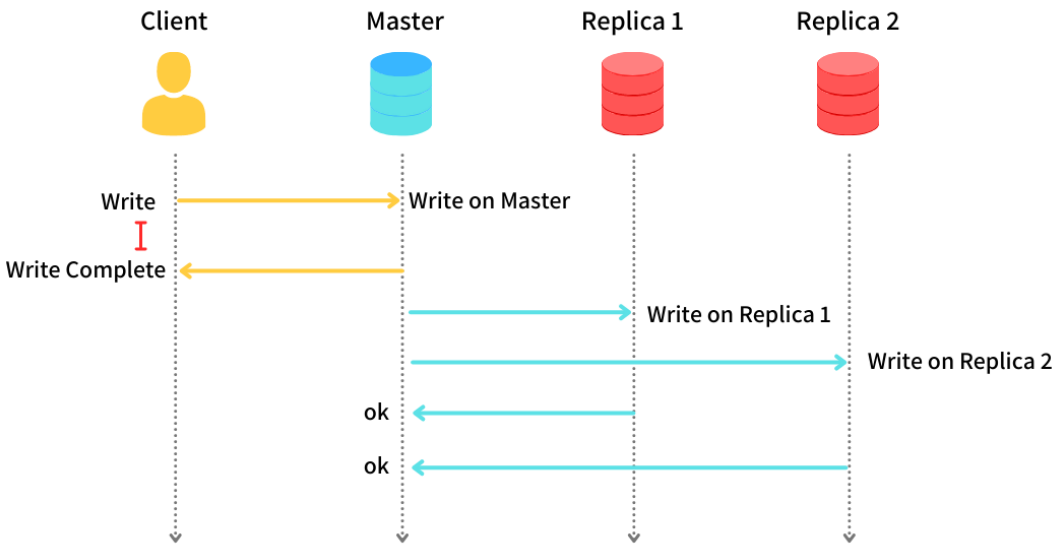

Рисунок 2.2 – Схема асинхронної реплікації даних

Напівсинхронна реплікація виступає компромісним рішенням між двома попередніми підходами. У цій архітектурі основний вузол очікує підтвердження успішного запису принаймні від одного з віддалених резервних вузлів перед тим, як підтвердити транзакцію клієнту [17]. Це дозволяє суттєво знизити загальну мережеву затримку порівняно зі строгою синхронною реплікацією, гарантуючи при цьому, що дані збережено як мінімум у двох незалежних локаціях наближаючи RPO до нуля. Порівняно з асинхронною реплікацією, напівсинхронна реплікація забезпечує покращену цілісність даних, оскільки коли фіксація завершується успішно, відомо, що дані існують щонайменше у двох місцях. Доки напівсинхронне джерело не отримає підтвердження від необхідної кількості реплік, транзакція залишається на паузі та не фіксується. Цю технологію підтримують AWS RDS в режимі Multi-AZ DB cluster, а також GCP Cloud SQL та Azure Database, але тільки для mysql.

На рисунку 2.3 зображено схему напівсинхронної реплікації даних.

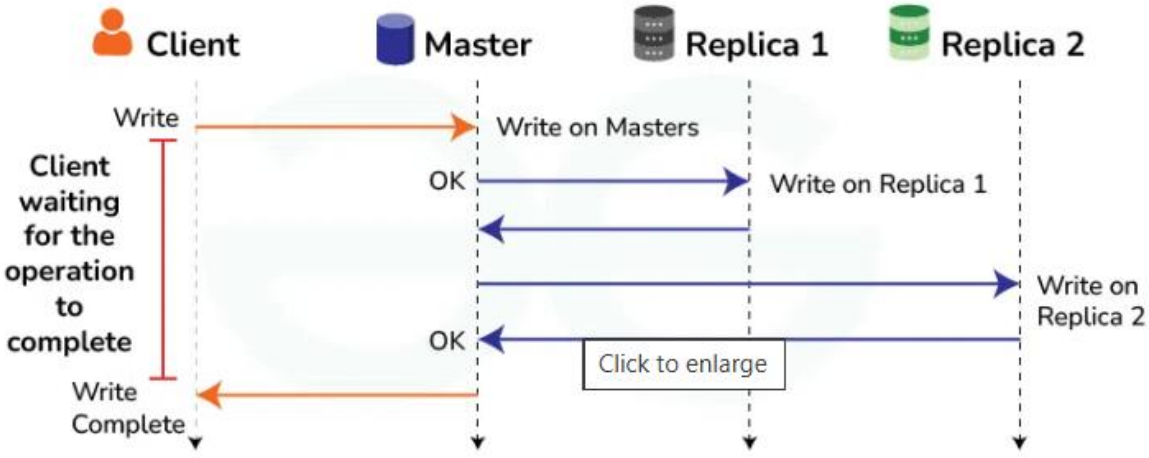

Рисунок 2.3 – Схема напівсинхронної реплікації даних

Кворумна реплікація у хмарних архітектурах широко підтримується у сучасних розподілені хмарні бази даних, таких як Amazon Aurora, Google Cloud Spanner, Azure Cosmos DB, використовують кворумні моделі консенсусу на основі протоколів Paxos або Raft для подолання обмежень традиційної

синхронної реплікації. Замість очікування відповідей від усіх вузлів, система вимагає підтвердження запису від визначеної більшості реплік. Наприклад, у Amazon Aurora дані реплікуються у 6 копій (по дві у кожній з трьох зон доступності), і транзакція вважається успішною після отримання підтвердження лише від 4 із 6 вузлів [18]. Щоб мінімізувати навантаження на мережу та уникнути ефекту посилення трафіку, через мережу передаються лише записи журналів транзакцій, а не повні сторінки даних, збірка блоків відбувається безпосередньо на вузлах зберігання.

Вибір конкретного механізму реплікації завжди є строгим архітектурним компромісом між показником RPO, глобальною продуктивністю системи та фінансовими витратами на інфраструктуру. Синхронна реплікація залишається безальтернативним вибором для транзакцій найвищого рівня критичності в межах одного регіону, тоді як повноцінне географічно розподілене аварійне відновлення здебільшого спирається на асинхронні або гібридні (кворумні та напівсинхронні) моделі. Обраний механізм реплікації баз даних формує технологічний фундамент, на якому базуватимуться стратегії відновлення всієї інфраструктури. При цьому важливо розуміти, що синхронна реплікація в межах різних зон доступності належить до категорії High Availability, оскільки захищає від локальних інфраструктурних відмов та забезпечує нульовий показник RPO з автоматичним перемиканням. Натомість асинхронна географічна реплікація є елементом стратегії Disaster Recovery, призначеним для виживання системи при втраті цілого хмарного регіону. Водночас жоден із механізмів реплікації не захищає від логічної корупції даних, таких як віруси-шифрувальники або помилка адміністратора БД, тому фундаментальним компонентом DR для баз даних залишається резервне копіювання.

2.3 Порівняльний аналіз моделей відновлення інфраструктури: Backup & Restore, Pilot Light, Warm Standby, Multi-Site Active/Active

Окрім відновлення, резервування і реплікації даних з оптимізації показника RPO, надзвичайно важливо забезпечити ефективне відновлення

обчислювальної та мережевої інфраструктури для забезпечення оптимального значення RTO. Процес побудови стійкої хмарної інфраструктури вимагає вибору оптимальної моделі відновлення.

У хмарних середовищах застосовуються чотири основні патерни аварійного відновлення, які формують спектр рішень від найдешевших з високими затримками до найдорожчих з нульовим часом простою: Backup and restore, Pilot Light, Warm Standby та Multi-Site Active/Active.

Backup and Restore є найбільш економічно вигідним підходом. Його суть полягає у тому, що у цій моделі резервна інфраструктура фізично не існує у нормальному режимі. Дані, зліпки блочних дисків та файли конфігурацій періодично експортуються у дешеве об’єктне сховище (Amazon S3 Standard-IA або Azure Blob Archive) в іншому географічному регіоні. Обчислювальні ресурси, такі як віртуальні машини, кластери контейнерів, балансувальники, не розгорнуті, тому компанія не несе операційних витрат на їх утримання [15]. Під час ініціалізації аварійного відновлення інженерам необхідно згенерувати інфраструктуру з нуля, налаштувати мережеві топології, а потім імпортувати резервні копії даних з об’єктного сховища на нові блочні накопичувачі. Для управління цими процесами використовуються інструменти оркестрації бекапів, такі як AWS Backup або Azure Backup.

У моделі Pilot Light критичне ядро інформаційної системи: сервери баз даних та ключові сховища, постійно розгорнуті та активні у резервному регіоні [15]. Ці системи безперервно отримують репліковані дані з основного середовища за допомогою асинхронних механізмів, гарантуючи низький рівень RPO. Однак рівень обробки додатків, що включає веб-сервери та сервери бізнес-логіки, залишається у вимкненому стані або розгорнутий у формі статичних образів віртуальних машин. При настанні інциденту скрипти автоматизації розгортають ці вимкнені обчислювальні ресурси, динамічно масштабують та підключають їх до вже синхронізованої бази даних. Для реалізації цієї стратегії провайдери пропонують спеціалізовані сервіси, такі як AWS Elastic Disaster Recovery (DRS) та Azure Site Recovery (ASR). Вони здійснюють безперервну

блочну реплікацію стану серверів на дешеві диски і дозволяють швидко конвертувати їх в активні обчислювальні інстанси під час переключення на резервне середовище.

У моделі Warm Standby в резервному середовищі функціонує повноцінна, але немасштабована (scaled-down) копія всієї екосистеми. Активними є як репліковані бази даних, так і мінімальний, але функціональний пул серверів додатків [15]. Ця зменшена інфраструктура завжди увімкнена, повністю налаштована і здатна обробляти мінімальний відсоток тестового трафіку. У момент аварії глобальний маршрутизатор миттєво перенаправляє весь клієнтський трафік на Warm Standby середовищі. Оскільки сервери вже запущені, додаткам не потрібен час на ініціалізацію. Після отримання трафіку хмарна інфраструкутра автоматично масштабується для обробки 100% навантаження, що надходить з основного регіону.

Метод Multi-Site Active/Active забезпечує найвищий рівень відмовостійкості. Інфраструктура розгортається у повному обсязі в кількох географічних регіонах одночасно [15]. Кожен регіон є повноцінним production середовищем, і трафік користувачів динамічно розподіляється між усіма середовищами за допомогою алгоритмів глобального балансування на основі географічної близькості або затримок. У разі повної відмови одного регіону трафік автоматично та безперервно перенаправляється на інші стабільні регіони. Користувачі практично не відчувають переривання трафіку. Однак ця архітектура вимагає впровадження складних алгоритмів вирішення конфліктів у розподілених базах даних та налаштування балансувальника навантажень, оскільки операції запису відбуваються одночасно у кількох регіонах.

У таблиці 2.1 наведена порівняльна характеристика описаних вище методів аварійного відновлення інфраструктури.

Крім того, для спрощення впровадження стратегій DR провідні хмарні провайдери пропонують спеціалізовані керовані сервіси аварійного відновлення.

Таблиця 2.1 – Порівняльна характеристика методів аварійного відновлення інфраструктури

Архітектурна стратегія	RTO	RPO	OPEX	Складність налаштування та управління
Backup and Restore	Години	Години	Низька. Оплачується виключно дисковий простір у дешевих класах об'єктного сховища.	Низька. Базове налаштування політик резервного копіювання та написання стандартних скриптів розгортання.
Pilot Light	Від десятків хвилин до годин	Хвилин	Помірна. Постійна оплата за синхронізацію сховищ та функціонування мінімального кластера баз даних.	Середня. Необхідність підтримки консистентної реплікації та механізму масштабування ресурсів.
Warm Standby	Хвилини	Секунди / Хвилини	Висока. Постійна оплата активних зменшених копій серверів додатків та інфраструктури балансування.	Висока. Жорсткий контроль за дрейфом конфігурацій, налаштування Auto Scaling, тестування автоматичного failover.
Multi-Site Active/Active	Близький до нуля	Близький до нуля	Дуже висока. Повноцінна оплата обчислювальних ресурсів в усіх регіонах, дуже високі витрати на міжрегіональний трафік.	Дуже висока. Складна маршрутизація GSLB, забезпечення транзакційної консистентності розподілених баз даних, оптимізація затримок.

Сервіс AWS Elastic Disaster Recovery реалізує архітектуру Pilot Light, здійснюючи безперервну реплікацію даних на рівні блоків з первинних серверів у резервний регіон AWS, підтримуючи мінімальну кількість активних ресурсів до моменту активації. Microsoft Azure пропонує рішення Azure Site Recovery, яке глибоко інтегроване в екосистему та дозволяє автоматизувати створення планів відновлення з підтримкою складних залежностей між додатками. Google Cloud Platform надає сервіс Google Cloud Backup and DR, який використовує підхід централізованого управління життєвим циклом копій даних. Кожен з цих інструментів оптимізує процеси перемикання трафіку, але вибір конкретної платформи залежить від технологічного стеку та вимог до зберігання даних конкретної організації.

Хмарні сервіси пропонують користувачам власні рішення для DR-стратегій(Disaster revovery as a Service, draas). Draas – це сервіси швидкого аварійного відновлення роботи віртуальних серверів на резервному платформі у разі часткової чи повної відмови продуктивного. Вони є основою для стратегій Pilot Light і Warm Standby.

Хмарні сервіси надають великий вибір інструменти для досягнення необхідних показників RPO та RTO. Вибір методу аварійного відновлення буде залежати від бюджету на операційну діяльність, людських ресурсів, критичності та складності інформаційної системи. Оптимальна архітектурна стратегія DR дозволить забезпечити функціонування в разі інциденту і дозволить уникнути фінансових втрат від простою системи.

2.4 Автоматизація процесу аварійного відновлення

Для досягнення оптимальних показників RTO необхідно забезпечити автоматизацію процесу аварійного відновлення (DR) і гарантувати стабільне розгортання інфраструктури в резервному середовищі. Застарілі, несинхронізовані мережеві налаштування або політики доступу (IAM) у резервному середовищі унеможливлюють коректну обробку відновлених даних і спричиняють інфраструктурний дрейф та дрейф конфігурацій.

Інфраструктурний дрейф виникає, коли реальний стан ресурсів у хмарі перестає збігатися з тим, що описано в конфігураційних файлах. Найтиповіші причини цього явища – ручні зміни через консоль провайдера, екстрені виправлення без оновлення репозиторію або паралельні конвеєри, які змінюють ті самі ресурси. Для стратегії DR це є критичною вразливістю: якщо резервне середовище будується з коду, а сам код розійшовся з основною робочою системою, після аварійного перемикання трафіку система отримає конфігурацію, яка не відповідає актуальним вимогам застосунків.

Єдиним ефективним рішенням для вирішення проблеми дрейфу є тотальне впровадження підходу «Інфраструктура як код» (Infrastructure as Code, iac), методології gitops та розширеної концепції «Аварійне відновлення як код» (Disaster Recovery as Code, draac).

Підхід iac полягає у використанні спеціалізованих інструментів для опису всієї архітектури: мереж, підмереж, кластерів, політик доступу та баз даних – у вигляді декларативного машинозчитуваного коду [9]. До таких інструментів належать платформо-незалежні Terraform чи Ansible, а також нативні рішення на кшталт AWS cloudformation або Azure Resource Manager. Декларативний підхід означає, що інженер описує бажаний кінцевий стан системи, а інструмент автоматично обчислює послідовність API-викликів для досягнення цього стану.

Ключовим аспектом iac є використання системи контролю версій як єдиного джерела істини, логічним продовженням якого є методологія gitops. Будь-яка зміна в інфраструктурі вноситься виключно через модифікацію коду (pull request) у Git-репозиторії [21]. Окремий агент (gitops-контролер) постійно порівнює стан, описаний у Git, зі станом у хмарі та приводить їх до відповідності. Цей цикл має назву reconciliation loop (цикл узгодження) і складається з чотирьох кроків:

– агент зчитує бажаний стан із Git;

– агент зчитує фактичний стан із API хмари або кластера Kubernetes;

– агент ідентифікує розбіжності;

– агент застосовує необхідні зміни для усунення розбіжностей.

Якщо ресурс змінюється, агент помічає це і повертає конфігурацію до еталонного значення з Git. Завдяки цьому дрейф автоматично відкочується або примусово фіксується в коді. Найпопулярніші реалізації gitops для Kubernetes, такі як argocd та Flux, мають вбудований режим самовідновлення для миттєвого повернення кластера до задекларованого стану. Для DR-сценаріїв це дає дві беззаперечні переваги: резервне середовище гарантовано збігається з основним, оскільки розгортається з того самого репозиторію, а після повернення системи в

нормальний стан (Failback) нічого не втрачається через забуті ручні налаштування.

Еволюцією цього підходу є Disaster Recovery as Code (draac) – концепція, що інтегрує процеси аварійного відновлення безпосередньо у конвеєри безперервної інтеграції та розгортання (CI/CD) [19]. Пайплайн для резервного середовища відрізняється від стандартного тим, що він розгортає ресурси в інший регіон, містить перевірки відповідності резервної конфігурації основній та має окремий алгоритм для автоматичного перемикання (failover).

Запуск такого конвеєра ініціюється трьома типами тригерів:

– коміт-тригер – запуск під час злиття коду у захищену гілку;

– плановий тригер – періодичний запуск за розкладом для перевірки здатності резервного середовища зібратися без помилок;

– тригер – автоматичний запуск за сигналом від системи моніторингу про падіння основного регіону (або запуск інженером у разі катастрофи).

Типовий процес виконання CI/CD-пайплайну для резервного середовища включає валідацію синтаксису, статичний аналіз безпеки, побудову плаун змін, перевірку Policy-as-Code, розгортання у резервному регіоні, оркестрацію та Smoke-тести, канаркове перемикання.

Окремим і надзвичайно важливим сценарієм використання такого конвеєра є періодична перевірка готовності DR-сценарію. У повністю ізольованій віртуальній мережі розгортається копія резервного середовища, симулюється процес аварії, після чого система автоматично вимірює фактичні метрики RTO і RPO та знищує тестове середовище.

Поєднання технологій iac, gitops та конвеєрів CI/CD дозволяє перетворити аварійне відновлення з непередбачуваної ручної операції, вразливої до людського фактора, на детермінований, повторюваний та відтестований алгоритм.

3 ПОРІВНЯЛЬНИЙ АНАЛІЗ DR-СЕРВІСІВ ПРОВІДНИХ ХМАРНИХ ПРОВАЙДЕРІВ

Розглянуті в попередніх розділах архітектурні патерни і теоретичні метрики отримують практичне втілення через конкретні сервіси хмарних провайдерів. Перехід від теоретичних стратегій аварійного відновлення до їх практичної реалізації вимагає детального аналізу інструментарію, який пропонують хмарні провайдери. Одні й ті самі концепції, такі як снепшоти, реплікація, failover, незмінність, реалізовані з помітними відмінностями в AWS, Microsoft Azure та Google Cloud Platform, і ці відмінності впливають на досяжні значення RTO/RPO, вартість DR-рішення та обсяг ручної роботи для підтримки резервного середовища.

Вимагається глибока декомпозиція та порівняння конкретних підсистем управління даними, оркестрації відновлення та глобальної маршрутизації за стандартизованими критеріями. Такий підхід дозволяє виявити архітектурні переваги та обмеження кожного провайдера в контексті мінімізації цільового часу (RTO) та цільової точки відновлення (RPO).

3.1 Резервне копіювання та незмінність даних у провайдерів

Резервне копіювання є інфраструктурним фундаментом для забезпечення базових показників RPO систем усіх рівнів критичності. У всіх трьох провайдерів центральною одиницею є керований сервіс, що уніфікує бекапи різних типів ресурсів і надає WORM (Write Once, Read Many) сховище для реалізації незмінності. Їхні архітектурні підходи до забезпечення незмінності даних, політик утримання та криптографічного захисту суттєво різняться.

AWS Backup – це сервіс для керування резервними копіями в AWS. AWS Backup поширюється на понад двадцять типів ресурсів: EBS-томи, екземпляри EC2, бази даних RDS, Aurora, dynamodb, EFS, fsx, Neptune, documentdb, об’єкти S3, а також гібридні середовища через Storage Gateway і vmware Cloud on AWS

[22]. Політики резервного копіювання задаються декларативно і включають розклад створення копій, тривалість зберігання та правила переміщення копій між гарячим (warm) і холодним (cold) сховищем для оптимізації витрат.

Незмінність реалізована через AWS Backup Vault Lock у двох режимах. Режим керування (governance mode) дозволяє адміністраторам із відповідними правами IAM змінювати політики чи достроково видаляти копії, він підходить для сценаріїв з внутрішнім контролем, де потрібен жорсткий, але оборотний бар'єр. Режим відповідності (compliance mode) є незворотним. Після активації відлічується пільговий період мінімум три дні, протягом якого блокування ще можна скасувати. Після його завершення жоден користувач AWS незалежно від прав і ролей не може видалити сховище чи скоротити термін зберігання копій до моменту їх закінчення [22]. Режим відповідності забезпечує захист проти атак з компрометацією привілейованих облікових записів.

Крос-регіональне копіювання налаштовується через параметр copy_actions у плані резервного копіювання (backup plan), крос-акаунтне налаштовується через AWS Organizations і довірче сховище в окремому обліковому записі. Шифрування виконується через AWS Key Management Service (KMS) з підтримкою ключів, керованих клієнтом (Customer-managed keys, CMK).

Окремо від AWS Backup існує Amazon S3 як універсальне сховище для альтернативних стратегій резервного копіювання. Функція S3 Object Lock реалізує модель WORM на рівні окремих об'єктів і працює лише в кошиках із увімкненим версіонуванням (S3 Versioning), який дозволяє заблокувати конкретну версію об'єкта неможливо. Object Lock має два режимиж governance і compliance, які працюють за логікою, аналогічною Vault Lock. Окремо від цих режимів існує механізм юридичного утримання (legal hold), який блокує видалення об'єкта без часового обмеження і знімається лише вручну [23]. Поєднання S3 Versioning, S3 Object Lock і крос-регіональної реплікації (S3 Cross-Region Replication) дозволяє реалізувати правило 3-2-1 з географічно рознесеними копіями на рівні об'єктного сховища.

Azure Backup – це сервіс для реалізації резервного копіювання в Azure. Він керує копіями віртуальних машин Azure, SQL Server і SAP HANA на ВМ, файлових ресурсів Azure Files, об’єктів Azure Blob Storage, баз даних postgresql Flexible Server і томів Kubernetes [24]. Сервіс використовує два типи сховищ копій. Recovery Services Vault – це більш старе сховище для віртуальних машин, SQL Server, файлів. Backup Vault – це новіше сховище для Blob Storage, postgresql Flexible Server і керованих дисків. Політики резервного копіювання підтримують схему GFS (grandfather-father-son), яка задає різну тривалість зберігання для копій. Це спрощує довготривалі плани збереження даних, оскільки не потрібно вручну керувати окремими копіями для кожного рівня. Незмінність забезпечує функція Immutable Vault. Її можна активувати у звичайному або незворотному режимі. У незворотному режимі після завершення короткого вікна для перевірки конфігурації не можна видалити копії або скоротити терміни зберігання. Заборона діє навіть для власника підписки (Owner). За логікою це аналог режиму compliance у AWS Vault Lock.

Шифрування даних виконується через Azure Key Vault з підтримкою ключів, керованих клієнтом (customer-managed keys, CMK), або ключів, керованих платформою, які використовуються за замовчуванням.

Крос-регіональні копії реалізовані через геонадлишкове сховище (geo-redundant storage, GRS). Кожен регіон Azure має жорстко закріплену пару, в яку Azure автоматично копіює дані. Така прив’язка обмежує гнучкість, оскільки клієнт не може обрати довільний регіон для копії. Натомість функція Cross Region Restore працює без додаткових налаштувань. Це дозволяє адміністраторам отримувати доступ до резервних копій та ініціювати відновлення у резервному регіоні навіть у випадку повної відмови первинного регіону.

Для об’єктного сховища в Azure існує окрема функція Immutable Blob Storage. Вона має два механізми незмінності, які працюють незалежно і можуть застосовуватися разом. Time-based retention блокує видалення на визначений період і не дозволяє зняти обмеження до його завершення. Legal hold блокує

видалення без часового обмеження і знімається лише вручну, через видалення відповідних тегів.

Google Cloud Backup and DR Service (BDR) – це централізований сервіс управління життєвим циклом резервних копій у GCP. BDR надає централізоване управління життєвим циклом копій для віртуальних машин (Compute Engine), баз даних та окремих блочних дисків. Він покриває віртуальні машини Compute Engine, інстанси Google Cloud vmware Engine, бази даних SQL Server, Oracle і SAP HANA, окремі блочні диски та файлові системи. Архітектурно сервіс складається з трьох компонентів. Керуюча консоль, через яку конфігуруються політики резервного копіювання і контролюється стан середовища. Backup/Recovery Appliances – віртуальні машини у VPC клієнта, які виконують безпосередню роботу зі створення копій та їх відновлення. Backup Vault – це сховище створених копій, яке знаходиться окремо від проєкту клієнта.

BDR використовує підхід вічно інкрементального копіювання (incremental-forever). Перший бекап є повним, усі подальші є інкрементальними. Функцію моментального монтування (Instant Mount) дозволяє клієнту запустити віртуальну машину безпосередньо з резервної копії, без фактичного відновлення даних на окремий диск. Це суттєво скорочує RTO для великих ВМ, оскільки старт не залежить від обсягу даних, які треба перенести.

Backup Vault у GCP відрізняється архітектурою від аналогів в AWS і Azure. Vault є повністю керованим Google сервісом і розташовується не в проєкті клієнта, а у внутрішній інфраструктурі провайдера, ізольованій від користувацького середовища [25]. Така модель реалізує air-gap на рівні інфраструктури, суть якого полягає у повній ізоляції сховища копій від користувацького середовища без прямих мережевих з’єднань. У разі компрометації облікового запису клієнта зловмисник не отримує доступу до Vault, оскільки сховище знаходиться поза зоною його контролю. GCP виносить сховище резервних копій в окремий проєкт Google і додає рівень ізоляції на випадок, коли зловмисник отримує максимальні права у основному обліковому записі.

Дані у Vault неможливо змінити чи видалити протягом заданого терміну зберігання, це поширюється на будь-яких користувачів GCP. Шифрування підтримує ключі, керовані клієнтом (customer-managed encryption keys, CMEK). Ключі залишаються під контролем клієнта навіть для бекапів у керованому Google сховищі. Для довгострокового зберігання поза BDR, у Cloud Storage, окремо використовується функція Bucket Lock з утриманням на рівні політик (policy-level retention).

В таблиці 3.1 наведено порівняльний аналіз сервісів резервного копіювання в найпопулярніших хмарних провайдерів.

Таблиця 3.1 – Порівняльний аналіз сервісів резервного копіювання в найпопулярніших хмарних провайдерів

Критерій порівняння	AWS (AWS Backup / Amazon S3)	Azure (Azure Backup / Blob Storage)	GCP (Cloud Backup and DR / Cloud Storage)
Основний керований сервіс	AWS Backup, Amazon S3	Azure Backup (Recovery Services / Backup Vault)	Google Cloud Backup and DR Service
Механізм незмінності даних (модель WORM)	Vault Lock / S3 Object Lock. Має два режими: Governance (змінний) та Compliance (незворотній).	Immutable Vault / Immutable Blob. Підтримує Time-based retention (незворотний на час дії) та Legal Hold.	Backup Vault Immutability / Bucket Lock. Дані повністю immutable та indelible (незмінні й невидальні).
Крос-регіональне копіювання	Гнучке: налаштовується довільно через параметр copy_actions у плані резервного копіювання.	Фіксоване: реалізовано через Geo-redundant storage із жорсткою прив'язкою до парного регіону (paired region).	Гнучке/Нативне: реалізується через реплікацію між регіонами у Backup and DR або використання Multi-region / Dual-region бакетів у Cloud Storage.
Крос-акаунтне копіювання та ізоляція	Через AWS Organizations до довіреного сховища в іншому акаунті клієнта (найгнучкіша модель).	Ізоляція досягається через розміщення бекапів у відокремлених підписках (Subscriptions) із жорстким розмежуванням прав (RBAC) та використанням Azure Lighthouse.	Air-gap на рівні інфраструктури: Vault розташований поза проєктом клієнта (у внутрішній інфраструктурі Google).

Кінець таблиці 3.1

Критерій порівняння	AWS (AWS Backup / Amazon S3)	Azure (Azure Backup / Blob Storage)	GCP (Cloud Backup and DR / Cloud Storage)
Шифрування (CMEK)	Повна інтеграція з AWS KMS для використання ключів клієнта.	Повна підтримка CMEK, через інтеграцію з Azure Key Vault.	Підтримується використання CMEK для Google Storage.
Ключові архітектурні особливості	Найбільший перелік ресурсів, найгнучкіша модель крос-акаунтних копій та політик.	Підтримка GFS-політик «з коробки», спрощене налаштування гео-резервування через парні регіони.	Підхід incremental-forever та функція Instant Mount (миттєвий запуск ВМ з бекапу), вимагає розгортання Appliance у VPC.

Три сервіси функціонально близькі, але відрізняються деталями. AWS Backup має найбільший перелік підтримуваних типів ресурсів і найгнучкішу модель копіювання між обліковими записами. Azure Backup тісніше інтегрований з гео-надлишковим зберіганням через парні регіони, що спрощує типові сценарії, але обмежує вибір цільового регіону. GCP Backup and DR виділяється функцією миттєвого монтування з майже миттєвим доступом до резервних даних. Натомість він вимагає розгортання власного Appliance у VPC клієнта, що ускладнює початкову конфігурацію. Усі три провайдери підтримують незворотний режим WORM, необхідний для захисту від кібератак з компрометацією привілейованих облікових записів. GCP Backup виносить сховище за межі проєкту клієнта. Це забезпечує додатковий рівень ізоляції на випадок, коли зловмисник отримує повні права в основному обліковому записі.

3.2 Сервіси аварійного відновлення з блочною реплікацією

Для забезпечення безперервності на рівні інфраструктури (iaas) та досягнення низьких показників RTO і RPO застосовуються сервіси аварійного відновлення як послуги (draas). Їхня головна мета – це забезпечити безперервну реплікацію стану серверів на рівні блоків пам’яті (Continuous Block-level Replication), що дозволяє відтворювати не лише дані, а й повний стан

віртуальних машин, включно з драйверами та мережевою конфігураціє, оминаючи етап довготривалого відновлення з об’єктних архівів.

AWS Elastic Disaster Recovery (DRS) – це сервіс для реплікацій на рівні блоку в AWS. На кожний source сервер встановлюється агент AWS Replication Agent, який зчитує зміни з локальних дисків на рівні блоків і потоково передає їх у цільовий регіон [15]. Агентна модель вимагає супроводу агента у гостьовій операційній системі, але натомість робить сервіс універсальним за типом джерела. DRS працює з фізичними серверами, віртуальними машинами Vmware, Hyper-V, KVM та з ВМ інших хмарних провайдерів, оскільки залежить лише від наявності операційної системи, а не від конкретного гіпервізора чи API хмари.

Ключовою особливістю DRS є архітектура з відкладеною конвертацією, яка реалізує підхід Pilot Light. У штатному режимі в цільовому регіоні функціонує Staging Area з мінімальним набором ресурсів, до якого входять кілька низькопродуктивних серверів реплікації і недорогі диски, що приймають вхідні потоки змін з агентів. Повноцінні віртуальні машини з характеристиками основного середовища у штатному режимі не створюються. Завдяки цьому місячні витрати на обмежуються вартістю Staging Area, що є дешевше за вартість дублювання повної продуктивної інфраструктури

Перетворення реплікованих даних на повноцінні цільові ресурси відбувається лише в момент запуску перемикання (Failover) або тестового запуску (Drill). Оркестратор DRS автоматично зупиняє запис на staging-диски, створює з них високопродуктивні томи Elastic Block Store необхідного типу, запускає цільові екземпляри EC2 у заданій підмережі, приєднує до них сформовані томи та повертає керування.

DRS забезпечує RPO у діапазоні секунд для стабільних мереж і RTO у межах 5-20 хвилин залежно від розміру ВМ і кількості томів. Агентна архітектура дозволяє працювати з джерелами з будь-яких середовищ: фізичні сервери, Vmware, Hyper-V, KVM, інші хмари. DRS підтримує наступні ОС: Windows Server від 2008 R2 і основні дистрибутиви Linux.

Azure Site Recovery (ASR) – це draas-сервіс Microsoft, який покриває три основні сценарії: захист локальних серверів Vmware або Hyper-V з реплікацією у хмару Azure, реплікація віртуальних машин між регіонами Azure та реплікація фізичних серверів у Azure [26].

ASR працює у двох режимах: з агентом і без агента. Агентний режим використовується для реплікації між регіонами Azure і для захисту фізичних серверів. Його суть полягає в тому, що у гостьову операційну систему встановлюється агент Mobility Service, який перехоплює всі записи на диск і дублює їх у цільовий регіон. Безагентний режим доступний для Vmware vsphere і Hyper-V. У ньому ASR підключається не до окремих віртуальних машин, а напряму до гіпервізора. Завдяки тому, що гіпервізор відстежує зміни блоків на дисках для власних задач, ASR має можливість зчитувати ці зміни і пересилати у цільовий регіон. Перевагою безагентного режиму є те, що не потрібно встановлювати, оновлювати і відстежувати агента всередині кожної ВМ.

Потік в ASR даних побудований через проміжний буфер. Зміни блоків спочатку записуються у проміжне сховище в основному регіоні у обліковий запис кешу (Cache Storage Account). Звідти вони асинхронно переносяться на керовані диски в цільовому регіоні. ASR використовує проміжне сховище з двох причин: згладжування мережевих сплесків, що забезпечує продуктивність основного середовища, навіть коли канал тимчасово завантажений; утримання черги змін, якщо канал у цільовий регіон коротко обривається. Без такого буфера будь-яка проблема з мережею миттєво зупиняла б реплікацію і збільшувала RPO.

При тестовому перемиканні ASR піднімає клони реплікованих віртуальних машин у окремій ізольованій віртуальній мережі (Vnet). Реплікація основного середовища при цьому не зупиняється, основні робочі ВМ і тестові клони існують паралельно і не бачать одне одного. Це дозволяє регулярно перевіряти готовність плану відновлення без жодного впливу на роботу користувачів.

ASR гарантує RPO від 30 секунд для crash-consistent snapshot і від 5 хвилин для application-consistent snapshot з інтеграцією VSS на Windows. RTO зазвичай 15-30 хвилин для середніх ВМ.

Google Cloud Platform не пропонує прямого аналога AWS DRS чи Azure ASR з безперервною блочною реплікацією. Backup and DR Service через механізм Instant Mount частково покриває функцію draas. Сервіс монтує резервну копію ВМ як пристрій блочного рівня безпосередньо з Backup Vault. ВМ стартує і отримує доступ до даних, у фоні дані переносяться на повноцінний диск через операцію Migrate, після чого ВМ перемикається на нього. Це дає короткий RTO, але RPO визначається розкладом бекапів.

Для захисту від зональних відмов GCP пропонує регіональні Persistent Disks, які синхронно реплікують записи між двома зонами одного регіону. Для крос-регіонального сценарію є окремий сервіс Persistent Disk Asynchronous Replication з цільовим RPO близько 1 хвилини, що покриває саме реплікацію даних на рівні дисків, але не оркестрацію перемикання. Повноцінне аварійне перемикання з автоматичним розгортанням ВМ, налаштуванням мережі та запуском плану відновлення реалізується через власні скрипти або сторонні інструменти.

В таблиці 3.2 наведено порівняльний аналіз draas рішень найпопулярніших провайдерів.

Таблиця 3.2 – Порівняльний аналіз архітектури draas

Критерій порівняння	AWS Elastic Disaster Recovery (DRS)	Azure Site Recovery (ASR)	Google Cloud Platform (GCP)
Базовий механізм реплікації	Безперервна асинхронна реплікація на рівні блоків пам’яті	Безперервна реплікація через Storage REST API (маршрутизація у кеш, потім на керовані диски)	Відсутній прямий аналог. Частково покривається Backup and DR Service (Instant/Sync Mount) та Regional Persistent Disks
Тип агента на джерелі	Легкий агент на рівні ОС (AWS Replication Agent) для кожного сервера	Гібридний: безагентний для підтримуваних гіпервізорів, або агентний для фізичних серверів	Залежить від сервісу бекапування (агент резервного копіювання) або кастомні рішення
Нативна інтеграція з гіпервізором	Відсутня. Реплікація відбувається на рівні гостьової ОС (через встановлений агент)	Пряма інтеграція на рівні гіпервізорів vmware vsphere та Microsoft Hyper-V	Відсутня для довільних ВМ, частково підтримується міграція через Migrate for Compute Engine

Кінець таблиці 3.2

Критерій порівняння	AWS Elastic Disaster Recovery (DRS)	Azure Site Recovery (ASR)	Google Cloud Platform (GCP)
Архітектура staging-середовища	Pilot Light: Staging Area з мінімальними ВМ (серверами реплікації) та дешевими дисками	Проміжний обліковий запис кешу (Cache Storage Account) та цільові керовані диски	Відсутня спеціалізована архітектура постійної реплікації; використання об’єктних сховищ для бекапів
Тестове перемикання (Test Failover)	Підтримується. Автоматичне розгортання повноцінних ВМ у цільовій підмережі (Recovery VPC) без переривання реплікації	Підтримується. Створення клонів віртуальних машин в ізольованій віртуальній мережі (vnet) без зупинки реплікації	Вимагає ручного розгортання або використання кастомних скриптів автоматизації
Failback (реверсна реплікація)	Підтримується. Наявний вбудований механізм повернення в основне середовище	Підтримується. Наявний вбудований механізм зворотної синхронізації	Реалізується вручну або за допомогою додаткових інструментів міграції
Recovery plans та оркестрація	Базові плани для упорядкованого запуску залежних систем	Розвинені (Глибока оркестрація): pre/post скрипти powershell, Azure Automation, ручні точки паузи, інтеграція з Traffic Manager	Відсутні готові плани для iaas, вимагається написання власних CI/CD пайплайнів чи скриптів

AWS DRS і Azure ASR – це готові draas-рішення з блочною реплікацією, тестовим перемиканням, планами відновлення і реверсною реплікацією. DRS відрізняється простішою агентною моделлю, яка працює з будь-якими джерелами без додаткової проміжної інфраструктури; ASR відрізняється більш розвиненою оркестрацією через плани відновлення, що зручно для складних багаторівневих застосунків. GCP не має повноцінного draas для довільних ВМ, що обмежує його застосовність для сценаріїв перенесення робочих навантажень і сповільнює процес аварійного відновлення. DRS тарифікується за годину реплікації одного сервера окремо від проміжного середовища, ASR тарифікується за захищений екземпляр на місяць з різною ціною для локальних сценаріїв і реплікації між регіонами Azure.

3.3 Реплікація керованих баз даних

Реплікація на рівні керованих БД є необхідним доповненням DR. Перемикання маршрутизації на рівні DNS або глобальних балансувальників не має практичного сенсу, якщо цільова резервна інфраструктура не має негайного доступу до синхронізованої БД або її репліки, тому механізми реплікації БД є критичним ядром DR-стратегії.

AWS пропонує декілька інструментів для забезпечення безперервності баз даних. Вони покривають сценарії від внутрішньо-регіональної високої доступності до повноцінного крос-регіонального реплікації. В контексті DR буде розглядатись лише міжрегіональна реплікація.

AWS RDS Cross-Region Read Replicas використовується міжрегіональні реплікації. Це асинхронні репліки, які створюються в іншому регіоні та працюють у режимі «тільки читання». При відмові основного регіону репліка вручну підвищується до повноцінної основної бази. Підвищення займає кілька хвилин, після чого застосунки треба переналаштувати на нову кінцеву точку. RPO залежить від мережевої затримки і обсягу транзакцій. У типових сценаріях він становить кілька секунд.

Amazon Aurora Global Database – це спеціалізоване рішення для postgresql- і mysql-сумісних кластерів. Воно підтримує один основний регіон і до п’яти резервних. Реплікація виконується на рівні сховища Aurora (Aurora storage layer), а не через журнали транзакцій SQL, що є суттєвою відмінністю від read replicas. Завдяки цьому підходу типовий RPO становить менше секунди [41]. Global Database також підтримує перенаправлення запитів на запис (write forwarding). Застосунок може звертатися до резервного регіону, а запити на запис автоматично передаються в основний без участі коду застосунку.

AWS dynamodb Global Tables – це рішення для nosql баз даних. Dynamodb Global Tables дозволяє налаштовувати таблиці з multi-master реплікацією між регіонами. Сервіс не має одного основного регіону. Усі регіональні копії

приймають запис одночасно, а зміни поширюються між регіонами за 1-2 секунди. Конфлікти при одночасних записах вирішуються за правилом «останній виграє» (last writer wins) на рівні елемента [27]. Це робить Global Tables зручними для глобально розподілених застосунків. Для систем зі строгими вимогами до транзакційної консистентності цей механізм не підходить, оскільки можливі ситуації, коли частина оновлень буде втрачена при одночасних записах.

Azure для крос-регіонального захисту реляційних баз даних пропонує дві технології з різним рівнем автоматизації: Active Geo-Replication та Auto-Failover Groups.

Azure Active Geo-Replication створює до чотирьох асинхронних копій бази даних у обраних регіонах. Копії доступні для читання і можуть використовуватися для розвантаження аналітичних запитів від основної бази. При відмові основного регіону репліка вручну підвищується до повноцінної основної бази. З недоліків можна відмітити, що цей підхід вимагає повторного налаштування застосунків після перемикання і не має автоматизованого механізму failover.

Azure Auto-Failover Groups об’єднує одну або кілька баз даних у групу зі спільною політикою автоматичного перемикання. Група має окрему точку доступу – listener, який сам направляє трафік до поточної основної копії. Застосунки не потребують повторного налаштування при перемиканні: вони продовжують підключатися до того самого імені, а listener сам спрямовує запити до правильного регіону. Ключовою особливістю Auto-Failover Groups є період очікування (grace period). Це час, протягом якого Azure чекає після виявлення збою перш ніж запустити автоматичне перемикання. Період очікування потрібен для того, щоб не реагувати на короткочасні мережеві проблеми, які зникають самі без участі DR-механізму. Типовий RPO для крос-регіональної асинхронної реплікації становить до 5 секунд, RTO становить 30-60 секунд після завершення періоду очікування.

Нереляційна СУБД Azure Cosmos DB має вбудовану підтримку реплікації між регіонами. Сервіс автоматично реплікує дані між обраними регіонами і може

приймати запис у кожному з них (multi-region writes). Це принципово відрізняє його від реляційних БД з єдиним основним регіоном. Особливістю Cosmos DB є вибір рівня консистентності клієнтом. Доступні п’ять рівнів:

– Strong (повна синхронна консистентність між усіма регіонами);

– Bounded staleness (відставання реплік обмежене за часом або кількістю операцій);

– Session (за замовчуванням, послідовна консистентність у межах сесії);

– Consistent prefix (зберігається порядок записів);

– Eventual (тільки фінальна консистентність).

Чим суворіша гарантія, тим більша затримка операцій через необхідність синхронізації між регіонами. Перемикання на інший регіон виконується автоматично при втраті основного. У режимі multi-region writes окрема процедура перемикання не потрібна, оскільки решта регіонів продовжують приймати запис без перерви.

GCP Cloud SQL і Cloud Spanner надають можливості для міжрегіональної реплікації даних. Для nosql рішень GCP пропонує Cloud Bigtable.

Для крос-регіональної реплікації даних Cloud SQL пропонує асинхронні read replicas в інших регіонах. Логіка аналогічна AWS RDS Cross-Region Read Replicas: репліка доступна для читання, при відмові основного регіону її потрібно вручну підвищити (promote), що підвищує RTO. RPO у типових сценаріях становить кілька секунд. На відміну від AWS і Azure, GCP не має інтегрованого механізму автоматичного перемикання на рівні Cloud SQL, тому оркестрація failover лежить на стороні замовника.

Cloud Spanner – це глобально розподілена реляційна БД із синхронною реплікацією через протокол консенсусу Paxos на основі truetime, сервісу Google для отримання точного часу на синхронізованих атомних годинниках. Multi-region інстанси автоматично реплікують дані у три або більше регіонів із синхронним кворумом, що дає нульове RPO при записі в кілька вузлів. Перемикання при відмові основного регіону відбувається автоматично і займає

секунди. Основним недоліком Cloud Spanner є обмежена кількість SQL-діалектів, СУБД підтримує лише діалекти googlesql та postgresql.

Bigtable підтримує крос-регіональну реплікацію через механізм кластерів: один інстанс може мати до восьми кластерів у різних регіонах, які асинхронно реплікують дані між собою. У режимі multi-cluster routing запити автоматично направляються до найближчого здорового кластера і при відмові прозоро переходять на інший. Це робить Bigtable функціонально схожим на dynamodb Global Tables. Конфлікти при одночасних записах вирішуються за правилом «останній виграє» (last writer wins), між кластерами підтримується лише eventual consistency.

AWS Aurora Global Database та Azure SQL з функцією Auto-Failover Groups мають низькі RTO та RPO для реляційних БД, в той час як GCP Cloud SQL має обмеження через відсутність вбудованої оркестрації автоматичного перемикання. Для нереляційних рішень AWS dynamodb Global Tables, Azure Cosmos DB і GCP Cloud Bigtable забезпечують зіставну ефективність завдяки підтримці запису між регіонами, однак відрізняються архітектурними підходами до узгодженості. Cosmos DB надає п’ять рівнів консистентності, тоді як dynamodb і Bigtable використовують модель із вирішенням конфліктів за принципом «перемагає останній запис». Можливостей базових рішень від AWS та Azure цілком достатньо для більшості типових корпоративних навантажень, однак для систем найвищого рівня критичності з високими вимогами уникнення втрати даних найкращим вибором є GCP Cloud Spanner, який, попри високу вартість та архітектурну складність, гарантує нульовий показник RPO завдяки синхронній крос-регіональній реплікації.

3.4 Маршрутизація трафіку і механіка failover

Після відновлення або реплікації даних та розгортання інфраструктури в резервному регіоні необхідно перевести трафік на нову точку входу. Цю роль виконують сервіси глобальної маршрутизації, ефективність яких визначається

двома параметрами: швидкістю виявлення відмови та швидкістю поширення нового маршруту до кінцевих клієнтів.

Усі три провайдери реалізують глобальну маршрутизацію через DNS-based і anycast-based моделі. У DNS-based моделі сервіс не пропускає трафік через себе, а при запиті DNS повертає клієнту IP-адресу обраної кінцевої точки, після чого з'єднання встановлюється напряму. Це економічно ефективна модель, але швидкість перемикання обмежена часом життя (TTL) DNS-запису і кешуванням на рівні проміжних DNS-серверів. У anycast-based моделі трафік проходить через магістральну мережу провайдера. Клієнт підключається до найближчої кінцевої точки через anycast-IP, який не змінюється при перемиканні, тому реакція на відмову регіону відбувається на мережевому рівні за секунди без залежності від DNS-кешу. Кожен провайдер пропонує реалізації обох моделей, але зі своїми специфічними відмінностями.

В AWS DNS-based маршрутизація реалізується через Amazon Route 53. Сервіс підтримує кілька режимів маршрутизації, але для DR ключовим є failover-режим з парою основного і резервного кінцевих точок та перевіркою доступності основного [28]. Health checks виконуються з понад 15 географічно розподілених локацій через HTTP, HTTPS або TCP з інтервалом 10 або 30 секунд і порогом від трьох послідовних невдач, що дає час детекції 30 або 90 секунд відповідно. Особливістю Route 53 є stйring-match – це перевірка не лише коду HTTP-відповіді, а й її вмісту. Це корисно у випадках, коли застосунок повертає HTTP 200 з повідомленням про внутрішню помилку. TTL зазвичай встановлюється на 60-120 секунд як компроміс між швидкістю перемикання і навантаженням на проміжні DNS-сервери. Anycast-based маршрутизацію в AWS забезпечує Global Accelerator, який працює на мережевому рівні (L4) і підтримує будь-які TCP/UDP-протоколи.

В Azure DNS-based рішенням є Azure Traffic Manager з режимом priority як основним для DR. Health checks виконуються з декількох регіонів Azure з інтервалом 30 секунд в стандартному режимі або 10 секунд в режимі швидкого аварійного перемикання, що дає час детекції до 90 або до 30 секунд відповідно.

Anycast-based маршрутизація реалізується через Azure Front Door. Azure Front Door – це балансувальник рівня застосунку (L7), який крім самого перемикання надає WAF, CDN-кешування і термінування SSL. Front Door працює лише з HTTP/HTTPS-трафіком, тоді як Traffic Manager універсальний за протоколом завдяки своїй DNS-природі.

В GCP anycast-based маршрутизацію виконує Global External Application Load Balancer. Global External Application Load Balancer – це балансувальник L7, інтегрований у базову мережу Google. Для не-HTTP TCP-трафіку GCP додатково пропонує Global External proxy Network Load Balancer, який працює на L4 і використовує той самий anycast-механізм; це функціональний аналог AWS Global Accelerator у частині TCP/SSL-проксіювання. Health checks виконуються з множини глобальних локацій з інтервалом 1-10 секунд і підтримують HTTP, HTTPS, HTTP/2, TCP, SSL та grpc. Ключова особливість GCP в тому, що anycast є поведінкою стандартного глобального балансувальника без додаткової тарифікації. DNS-based маршрутизація в GCP реалізується через Cloud DNS з підтримкою політик маршрутизації. Failover-політики Cloud DNS використовуються для реалізації DR за логікою, аналогічною Route 53 і Traffic Manager.

Вибір рішення залежить від профілю навантаження. Для систем з різнорідними протоколами (TCP, UDP, ігрові, кастомні) AWS Global Accelerator залишається найбільш універсальним завдяки L4. GCP надає окремі L4 і L7 сервіси, однак їх потрібно налаштовувати незалежно. Для веб-навантажень з потребою у WAF, CDN і термінуванні SSL зручніший Azure Front Door, який покриває ці функції з коробки. GCP Global External Application Load Balancer є кращим вибором, коли застосунок уже розгорнутий у GCP, оскільки anycast-маршрутизація доступна за замовчуванням, а ширший набір протоколів health checks, включно з grpc і HTTP/2, корисний для мікросервісних архітектур. На рівні DNS Route 53 залишається кращим вибором завдяки string-match. Коли для секундних RTO жодне DNS-based рішення не підходить незалежно від провайдера, обов’язковим є перехід на anycast.

4 РОЗРОБКА ПІДХОДУ ДЛЯ ВИБОРУ СТРАТЕГІЇ АВАРІЙНОГО ВІДНОВЛЕННЯ

Вибір між стратегіями аварійного відновлення (Backup and restore, Pilot light, Warm standby, Multisite Active/Active) не зводиться лише до зіставлення RTO та RPO. Кожна стратегія DR генерує постійні операційні витрати на утримання резервної інфраструктури, обсяг яких може відрізнятись. Стратегія, що задовольняє цільові показники tier-класу за найменшої сумарної вартості, є фінансово оптимальним вибором. Вибір стратегії DR потребує економічної моделі, яка зіставляє сукупну вартість володіння (Total Cost of Ownership, TCO) кожної стратегії з очікуваними річними втратами від простою (Annualized Loss Expectancy, ALE).

Моделювання буде виконуватись на платформі Amazon Web Services.

4.1 Класифікація систем за критичністю та визначення базових показників для обчислення вартості

RTO і RPO є базовими технічними параметрами вибору стратегії DR. Додатково на вибір впливають бюджетні обмеження, регуляторні вимоги і операційна спроможність організації. Тому перед моделюванням загальної вартості володіння системи поділяються на класи критичності з прив’язкою до конкретних значень RTO і RPO.

Визначено три класи систем (Tiers). Кожен клас має зафіксовані вимоги до RTO та RPO, які визначають допустимі межі простою сервісу під час аварії.

Tier 1 – системи, для яких будь-який простій спричиняє пряме переривання основної бізнес-функції. До цього класу належать критичні внутрішні OLTP-системи банків середнього рівня, процесингові компоненти fintech-сервісів, real-time білінг телеком-операторів регіонального масштабу. Цільовий RTO для Tier 1 становить 15 хвилин, RPO наближається до нуля [6].

Tier 2 – системи, які забезпечують основні бізнес-операції, але припускають короткочасний простій без фатальних наслідків. До цього класу належать платформи електронної комерції середнього масштабу, корпоративні CRM- і ERP-системи, бекенди мобільних додатків. Цільовий RTO становить 4 години, RPO – 2 години [6].

Tier 3 – системи, які підтримують допоміжні функції та не блокують операційну діяльність при відмові. Це сховища архівних логів, аналітичні data-warehouse без вимог реального часу, внутрішні портали відповідності. Допустимий RTO становить 8-24 години, RPO – 4 години [6].

Для коректного зіставлення TCO між стратегіями необхідно визначити єдине еталонне навантаження на систему. Параметри навантаження наведено в таблиці 4.1.

Таблиця 4.1 – Параметри еталонного навантаження

Параметр	Значення
Кількість віртуальних машин	50
Обсяг блочного сховища на одну ВМ	500 ГБ
Сумарний виділений блочний простір	25 ТБ
Ефективний обсяг даних після стискання	10 ТБ
Щоденна дельта змін даних	~300 ГБ (3% від ефективного обсягу)
Базовий тип екземпляру EC2	M5.xlarge (4 vcpu, 16 ГБ)
Регіональна пара	Us-east-1 (основний) / us-west-2 (резервний)
Вихідний трафік (egress) в Інтернет	1 ТБ/місяць

Конфігурація 50 ВМ по 500 ГБ – це умовне середнє корпоративне навантаження, обране для фіксації порівняльної бази. Модель розрахунку лінійна за обсягом інфраструктури, тому заміна на 100 ВМ по 500 ГБ або 25 ВМ по 1 ТБ пропорційно змінить абсолютні цифри TCO, але збереже відносні співвідношення між стратегіями.

Ефективний обсяг даних 10 ТБ при 25 ТБ виділеного блочного простору відображає типове неповне заповнення дисків у промислових середовищах і визначає реальний обсяг кросрегіональної реплікації.

Екземпляр m5.xlarge представляє клас загального призначення. Зміна класу впливає на абсолютну вартість EC2, не змінюючи структури порівняння.

Регіональна пара us-east-1/us-west-2 обрана за принципом географічної ізоляції від спільних інфраструктурних ризиків [15].

Базовий вихідний трафік 1 ТБ/місяць відповідає низькій інтенсивності користувацького навантаження. У порівнянні стратегій між собою цей параметр не вносить різниці, оскільки він тарифікується однаково в усіх чотирьох сценаріях.

Зв’язок між класом критичності та обраною стратегією не є однозначним: одні стратегії можуть не виконувати цільові показники; інші навпаки, можуть перевиконувати їх, там самим забираючи надмірну кількість фінансових ресурсів; деякі ж покривають вимоги RTO/RPO лише частково на нижній межі. У таблиці 4.2 технічної відповідності RTO і RPO.

Таблиця 4.2 – Відповідність tier-класів стратегіям аварійного відновлення

Стратегія	RTO стратегії	RPO стратегії	Tier 1 (15 хв/~0)	Tier 2 (4 год/2 год)	Tier 3 (8-24 год / 4 год)
Backup and Restore	Години	24 год	Не задовольняє	Не задовольняє	Мінімально достатня
Pilot Light	Десятки хвилин	Секунди	Граничний випадок	Достатня	Перевиконує
Warm Standby	Хвилини	Секунди	Достатня	Перевиконує	Перевиконує
Multi-Site Active/Active	Секунди	Приблизно 0	Перевиконує	Перевиконує	Перевиконує

Стратегія Backup and Restore сумісна з усіма tier-класами, проте показник RTO на рівні 12-24 годин робить її доцільною виключно для Tier 3. Модель Pilot

Light із RTO 10-30 хвилин є оптимальною для систем класу Tier 2, а за певних умов задовольняє нижню межу вимог Tier 1. Своєю чергою, стратегія Warm Standby (1-5 хвилин) відповідає вимогам Tier 1, проте для Tier 2 створює економічно невиправдану надлишковість. Застосування Multi-Site Active/Active із секундним RTO та близьким до нуля RPO є обґрунтованим виключно для систем найвищого рівня критичності, де відсутність простоїв є жорсткою бізнес-вимогою.

4.2 Моделювання загальної вартості утримання чотирьох стратегій аварійного відновлення на платформі AWS

Технічна відповідність DR-стратегії вимогам Tier-класу є необхідною, але не достатньою умовою архітектурного вибору. Серед стратегій, які задовольняють однакові цільові RTO і RPO, місячна вартість може суттєво відрізнятися. Розрахунок сукупної вартості володіння (Total Cost of Ownership, TCO) перетворює якісне порівняння на кількісне і дозволяє обґрунтовано обрати стратегію за критерієм мінімальних витрат серед технічно прийнятних варіантів.

Розрахунок виконано для чотирьох DR-стратегій: Backup and Restore, Pilot Light, Warm Standby і Multi-Site Active/Active. Усі розрахунки виконано на основі тарифів AWS у регіоні us-east-1 з реплікацією до резервного регіону us-west-2. Кожну стратегію оцінено у двох цінових режимах: з оплатою обчислювальних ресурсів за моделлю on-demand та з застосуванням однорічного плану економії Compute Savings Plan разом, який забезпечує приблизно 28% знижки на обчислювальну частину витрат, разом з Database Savings Plan, який 20% знижки на обчислення для Aurora.

4.2.1 Структура витрат компонентів аварійного відновлення на AWS

Перед тим як розрахувати TCO для конкретних стратегій, необхідно розглянути ціни на компоненти, спільних для всіх чотирьох моделей.

Сукупна вартість володіння DR-рішенням на платформі AWS складається з повторюваних місячних витрат на сервіси, які утримують резервну інфраструктуру у готовому до активації стані. Саме структура цих витрат, а не їхні абсолютні значення, визначає поведінку моделі при зміні вхідних параметрів робочого навантаження. Тому подальший аналіз зручно вести в категоризованому вигляді, виділяючи чотири основні групи: сховище, мережа, обчислення та реплікація з допоміжними сервісами.

Витрати на сховище у DR-сценаріях формуються з дисків віртуальних машин, знімків блочних томів та архівних копій в об’єктному сховищі. Тарифні ставки для регіону us-east-1 наведено у таблиці 4.3.

Таблиця 4.3 – Компоненти сховища DR на AWS (us-east-1, квітень 2026)

Сервіс	Ставка	Призначення в DR
EBS gp3	0,08 $/ГБ-міс	Активні диски ВМ
EBS sc1	0,015 $/ГБ-міс	Staging-диски DRS
EBS Snapshots Standard	0,05 $/ГБ-міс	Денні (інкрементальні) знімки
EBS Snapshots Archive	0,0125 $/ГБ-міс	Довгострокове утримання знімків
S3 Glacier Deep Archive	0,00099 $/ГБ-міс	Архівні бекапи
Aurora storage	0,10 $/ГБ-міс	Кластери реляційних БД

Архітектурне призначення кожного сервісу визначає, у якій стратегії він використовується. Дешевий клас Glacier Deep Archive є основою Backup and Restore, тоді як швидкий gp3 застосовується лише там, де є постійно увімкнені обчислювальні потужності.

Мережеві витрати у DR-моделях формуються переважно міжрегіональним трафіком, який тарифікується за фіксованою ставкою незалежно від програми зобов’язань, а також керованими сервісами балансування навантаження та маршрутизації. Ставка 0,02 $/ГБ за міжрегіональний трафік є рушійним фактором витрат для стратегій з безперервною реплікацією: при щоденній дельті 300 ГБ навіть мінімальна реплікація DRS генерує близько 9-11 ТБ вихідного трафіку щомісяця. Ставки наведено у таблиці 4.4.

Таблиця 4.4 – Мережеві компоненти DR

Компонент	Ставка
Міжрегіональний трафік (us-east-1 до us-west-2)	0,02 $/ГБ
Application Load Balancer (ALB)	16,43 $/міс + LCU за тарифом 0,008 $/LCU-год
NAT Gateway	0,045 $/год + 0,045 $/ГБ обробки
Global Accelerator	18,25 $/міс fixed + 0,015 $/ГБ DT-Premium

Обчислювальна частина витрат охоплює віртуальні машини додатків, екземпляри баз даних та допоміжні сервери реплікації. Наведено три типи екземплярів віртуальних серверів, які найчастіше використовуються у досліджуваних стратегіях, у двох цінових режимах: on-demand і з застосуванням двох планів економії (Compute Savings Plan та Database Savings Plan) [29]. Ставки наведено у таблиці 4.5.

Таблиця 4.5 – Обчислювальні компоненти у двох цінових режимах

Інстанс	On-demand($/год)	On-demand ($/місяць)	З 1-уг Compute
M5.xlarge	0,192 $/год	140,16 $/міс	≈101 $/міс
T3.small	0,0208 $/год	15,18 $/міс	≈11 $/міс
Db.r8g.large	0,276 $/год	201,48 $/міс	≈161 $/міс (Database SP ≈20 %)

Окрему категорію формують реплікаційні та допоміжні сервіси. Ліцензія AWS Elastic Disaster Recovery тарифікується посервереводинно і не залежить від обсягу даних, реплікаційний потік Aurora Global Database вимірюється в мільйонах операцій вводу-виводу, сервіси шифрування ключів і агрегування журналів працюють за змішаною моделлю: фіксована ставка плюс плата за обсяг операцій. Ставки цієї категорії наведено у таблиці 4.6.

Таблиця 4.6 – Реплікаційні та допоміжні сервіси

Сервіс	Ставка
AWS Elastic Disaster Recovery (DRS)	0,028 $/server-год = 20,44 $/server/міс
Aurora Global Database replicated I/O	0,20 $ за мільйон операцій
AWS Key Management Service (KMS)	1 $/ключ/міс + 0,03 $ за 10 тис. Операцій
Cloudwatch Logs	0,50 $/ГБ ingestion + 0,03 $/ГБ archival

4.2.2 Розрахунок загальної вартості утримання для стратегії Backup and Restore

У стратегії Backup and Restore у резервному регіоні постійно існує лише сховище. Обчислювальні ресурси розгортаються виключно у момент активації плану відновлення. Архітектурно це реалізовано через сервіс AWS Backup, який створює інкрементальні знімки томів EBS і кластерів Aurora. Політика життєвого циклу автоматично переміщує копії у клас Glacier Deep Archive через тридцять днів зберігання. Міжрегіональне копіювання конфігурується через параметр copy_actions у плані резервного копіювання, який спрямовує знімки до резервного регіону us-west-2. Інфраструктура додатків у DR-регіоні описана у шаблонах cloudformation або Terraform і у штатному режимі не існує як працюючі ресурси.

Структура місячного TCO обмежується вартістю архівного сховища і міжрегіонального трафіку. Розрахунки TCO для стратегії Backup and Restore подані в таблиці 4.6.

Таблиця 4.6 – Місячні витрати стратегії Backup and Restore

Стаття	Розрахунок	$/міс
Glacier Deep Archive у DR-регіоні	10000 ГБ × $0.00099 × 1 регіон	9,90
Міжрегіональне копіювання (амортизована initial copy + щомісячна дельта)	(дельта × 30 днів + initial/12) × $0.02/ГБ	196,67
EBS Snapshots Archive у source-регіоні	10000 ГБ × $0.0125/ГБ-міс	125,00

Кінець таблиці 4.6

Стаття	Розрахунок	$/міс
KMS, Backup API, cloudwatch	Фіксовано $30/міс	30,00

Місячний TCO складає 361,57 $/міс.

Структура витрат для стратегії Backup and Restore зображена на рисунку 4.1.

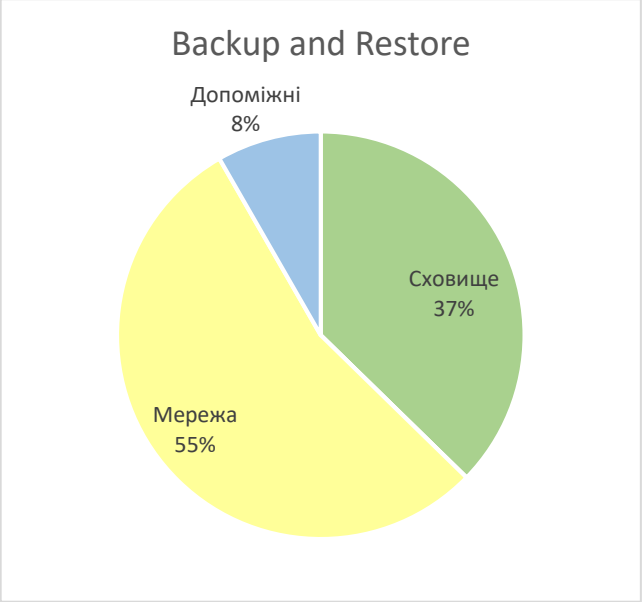

Рисунок 4.1 – Структура витрат стратегії Backup and Restore

Структура витрат повністю визначається сховищем і мережею. Міжрегіональне копіювання даних до резервного регіону формує 54,4% TCO, архівне сховище – 37,3%, допоміжні сервіси – 8,3%. Compute, Database та Replication відсутні, оскільки у штатному режимі активної інфраструктури в DR-регіоні не існує.

Основна копія 10 ТБ у Glacier Deep Archive є найдешевшим класом, S3($0,00099/ГБ-міс), розрахований саме на рідко відновлювані бекапи, і тому коштує $9,90. Передача даних до DR-регіону ($196,67) – це основна стаття цієї моделі, оскільки щомісяця дельта(300 ГБ) реплікується, даючи 9 ТБ щомісячного трафіку. Частина повної початкової копії амортизується, приблизно 833 ГБ трафіку на місяць. Таким чином виходить 9833 ГБ міжрегіонального трафіку за

місяць за тарифом $0,02/ГБ-міс. Окремо в основному регіоні утримується паралельна копія в EBS Snapshot Archive ($125) – це опційний шар захисту від кібер-атак на основну інфраструктуру. Він не обов'язковий для DR, але стандартний для виробничих систем. Решта $30 – це фіксована оцінка KMS-ключів, викликів Backup API і cloudwatch-логів. Compute і Database Savings Plans не дають економії взагалі, оскільки у DR-регіоні немає ні постійних серверів, ні працюючих БД.

Разом з місячним TCO виникають витрати, які реалізуються лише у момент реальної аварії і тому не входять до планового бюджету утримання. До них належать одноразова плата за отримання даних з Glacier Deep Archive по тарифу Standard Retrieval (0,02$/ГБ) у розмірі 200$, restore time fee і тимчасове зберігання у класі S3 Standard на час відновлення вартістю $50-100, а також обчислювальні витрати на запуск 50 інстансів m5.xlarge у DR-регіоні протягом одного-двох тижнів роботи системи у резервному середовищі, що дає $1750-3500 додатково. Сумарна разова витрата при реальному інциденті оцінюється приблизно у $2000.

4.2.3 Розрахунок загальної вартості утримання для стратегії Pilot Light

Pilot Light будується навколо двох постійно активних компонентів у резервному регіоні: реплікацій AWS DRS та вторинного кластера Aurora Global Database. Сервіс AWS DRS реплікує всі 50 серверів у staging area DR-регіону. Дані безперервно записуються на дешеві диски EBS sc1, а реплікаційні сервери запускаються у мінімальній конфігурації t3.small на сім екземплярів. Aurora Global Database secondary cluster, що складається з двох екземплярів db.r8g.large, працює постійно і приймає реплікований потік від основного кластера через storage-layer replication. Усі інші application-сервери у штатному режимі вимкнені, їхні образи готові до запуску з реплікованих дисків у момент активації плану відновлення.

Така архітектура породжує більш складну структуру TCO, у якій уже представлені всі чотири категорії витрат. Розрахунки TCO для стратегії Backup and Restore подані в таблиці 4.7.

Таблиця 4.7 – Місячний TCO стратегії Pilot Light

Стаття	Розрахунок	$/міс
Ліцензія DRS (50 серверів)	50 servers × $0.028/server-год × 730 год	1 022,00
Staging EBS sc1 (25 ТБ)	25000 ГБ × $0.015/ГБ-міс	375,00
Реплікаційні сервери t3.small (7 шт)	7 шт × $0.0208/год × 730 год	106,29
EBS Snapshots staging (worst-case unique blocks)	(10000 + 7×300) ГБ × $0.05/ГБ-міс (worst-case unique blocks)	605,00
Aurora secondary cluster (2× db.r8g.large)	2 × $0.276/год × 730 год	402,96
Aurora storage у DR-регіоні	10000 ГБ × $0.10/ГБ-міс	1 000,00
Aurora cross-region I/O	(дельта × 1024^2 / 8 КБ блок) × 30 днів / 1e6 × $0.20/млн	235,93
Міжрегіональний трафік DRS (+20 % overhead)	300 ГБ/д × 30 дн × 1.2 (DRS overhead) × $0.02/ГБ	216,00
KMS, cloudwatch, Route 53	50 servers × $0.028/server-год × 730 год	80,00
Місячний TCO on-demand		4 043,18
3 Compute SP 1-yr		3 932,82

Найбільша стаття витрат у Pilot Light – це ліцензія AWS DRS ($1 022). Вона тарифікується посерверно за ставкою $0,028 за сервер-год, тобто 50 серверів по 730 годин на місяць. Ставка не залежить від обсягу реплікованих даних, оскільки оплачується сам факт захисту сервера. Поряд з ліцензією йде staging-інфраструктура: 25 ТБ дисків sc1 ($375) для прийому потоку даних реплікаціїї та 7 екземплярів t3.small ($106).

Інкрементальні знімки EBS staging-області з семиденним терміном утримання дають $605, оскільки розрахунок виконано за консервативною оцінкою, коли всі змінені блоки за тиждень вважаються унікальними. Реальна

вартість завдяки блочній дедуплікації EBS зазвичай нижча на 30-50%, проте для моделі використано песимістичний сценарій.

Окремо постійно увімкнений вторинний кластер бази даних Aurora ($403) – це два екземпляри типу db.r8g.large, оскільки база даних має бути готова до підвищення до основної за хвилини. Aurora Storage коштує $1 000 (10 ТБ за тарифом $0,10 за ГБ-міс). Aurora cross-regional I/O становить $236 і тарифікується не за обсяг даних, а за кількість операцій запису. Розрахунок іде через сторінки Aurora розміром 8 КБ: 300 ГБ дельти ділиться на 8 КБ та множиться на 30 днів, що дає приблизно 1 180 млн операцій запису на місяць.

Міжрегіональний трафік сервісу DRS складає $216. Щоденна дельта даних 300 ГБ за 30 днів дає 9 ТБ передачі на місяць. До цього обсягу додається близько 20% службових накладних витрат, які генерує протокол реплікації – це метадані операцій, перевірочні пакети та контрольні суми. Фактичний обсяг переданих даних становить близько 10,8 ТБ, який за тарифом $0,02 за ГБ дає $216. Решта $80 – це фіксована оцінка вартості KMS-ключів шифрування, cloudwatch-логування та перевірок доступності Route 53.

Структура витрат для стратегії Pilot light зображена на рисунку 4.2.

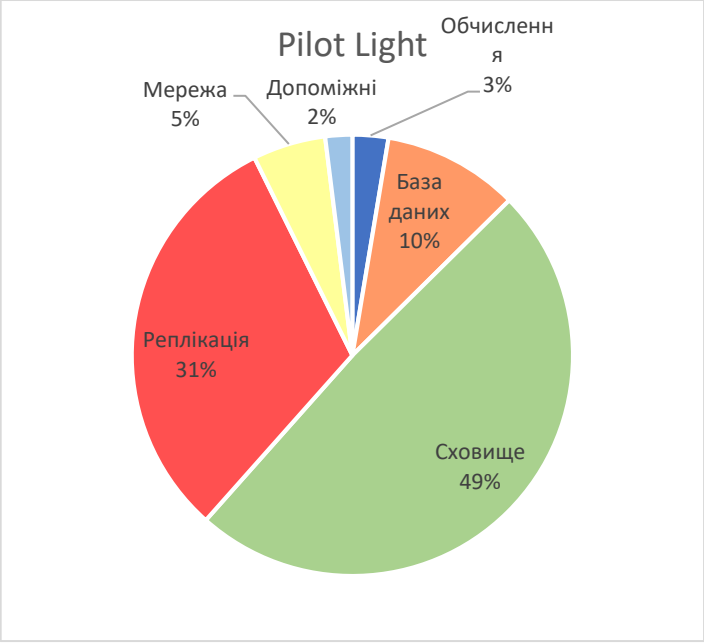

Рисунок 4.2 – Структура витрат стратегії Pilot Light

Як видно з рис. 4.2, найбільшу долю витрат займає сховище (49,0%) – staging-диски DRS, EBS-знімки та Aurora storage. Реплікаційний шар (DRS-ліцензія та Aurora cross-region I/O) формує 31,1%. Бази дани (Aurora secondary cluster) складають 10,0%. Обчислення(replication servers t3.small) слкдають лише 2,6%, мережевеві сервіси – 5,3%, допоміжні сервіси – 2,0%. Структура є орієнтованою на реплікацію даних в сховище, що відображає суть стратегії: безперервна реплікація даних без активного прикладного шару.

Compute Savings Plans зі знижкою 28% поширюється на вартість реплікаційних серверів, а Database Savings Plans зі знижкою 20% на кластер Aurora secondary. Сумарна економія від обох програм становить близько $110 на місяць, або 2,7% від TCO. Оскільки часта обчислювальної складової та складової БД невеликі, Saving Plans не мають значного впливу на TCO.

Крім того варто враховувати, що тестові перемикання вимагають виконання окремого DRS launch action і генерують додаткову вартість обчислювальних ресурсів на час тесту: один drill для 50 ВМ протягом чотирьох годин коштує близько $40.

4.2.4 Розрахунок загальної вартості утримання для стратегії Warm Standby

Архітектура Warm Standby базується на тому самому реплікаційному шарі DRS, що й Pilot Light, проте додатково передбачає постійне утримання частини прикладного шару у резервному регіоні. У базовій конфігурації стратегії 30% обчислювальних потужностей працюють безперервно разом з вторинним кластером Aurora, балансувальником навантаження ALB у warm-режимі і налаштованою Auto Scaling Group, яка у момент перемикання здатна швидко масштабувати ВМ з 30% до 100%. Ключова архітектурна різниця між Pilot Light і Warm Standby зводиться саме до утримання активного, хоч і неповного, прикладного шару у DR-регіоні.

Розрахунок місячних витрат для еталонного навантаження наведено у таблиці 4.8.

Таблиця 4.8 – Місячний TCO стратегії Warm Standby

Стаття	Розрахунок	$/міс
DRS license	50 servers × $0.028 × 730 год	1 022,00
Staging EBS sc1	25000 ГБ × $0.015	375,00
Replication servers t3.small	7 × $0.0208 × 730	106,29
EBS Snapshots staging	(10000 + 7×300) × $0.05 (worst-case unique blocks)	605,00
Активні ВМ у DR (m5.xlarge)	15 × $0.192 × 730 (30% capacity warm)	2 102,40
EBS gp3 для активних ВМ	15 ВМ × 500 ГБ × $0.08	600,00
Aurora secondary cluster (db.r8g.large)	2 × $0.276 × 730	402,96
Aurora storage	10000 × $0.10	1 000,00
Aurora I/O (cross-region writes + WS read I/O)	Write I/O (cross-region, 8 КБ блок) ~1180 млн + WS read I/O ~50 млн × $0.20/млн	245,93
ALB у DR	$16.43/міс + ~50 LCU × $0.008 × 730 год	308,43
NAT Gateway	$0.045 × 730 год + 200 ГБ × $0.045 (processing)	41,85
Inter-region трафік	15000 ГБ × $0.02	300,00
KMS + CW + Route 53 + WAF	Фіксовано	150,00
Місячний TCO on-demand		7 259,86
3 Compute SP 1-yr		6 560,83

Реплікаційний блок Warm Standby повністю ідентичний Pilot Light. Архітектурна та вартісна різниця починається там, де у Pilot Light був сплячий прикладний шар. Активні віртуальні машини у DR-регіоні дають $2102 – це 15 інстансів (30% capacity) m5.xlarge по ціні $0,192/год.

Швидкі диски EBS gp3 для них вартують $600 (7,5 ТБ по ціні $0,08 за ГБ-міс). Aurora secondary cluster ($403) і її сховище ($1000) аналогічні Pilot Light, оскільки база даних в обох стратегіях працює в однаковому режимі прийому реплікації. Aurora cross-regional I/O ($246) вищий, ніж у Pilot Light, оскільки до

1180 млн операцій запису додається близько 50 млн операцій читання від warm-навантаження.

Мережева частина суттєвіша через те, що частина трафіку реально проходить. ALB у DR коштує $308, це фіксована плата $16,43 плюс 50 LCU по ціні 0,008 $/год протягом місяця. NAT Gateway ($42) тарифікується подвійно: $0,045 за годину роботи плюс $0,045 за ГБ обробленого трафіку (200 ГБ на місяць). Міжрегіональний трафік складає 15 ТБ ($300 за тарифом $0,02 за ГБ) за рахунок додаткової синхронізації стану додатків і моніторингу між регіонами. Допоміжні сервіси на $150.

Структура витрат для стратегії Warm Standby зображена на рисунку 4.3.

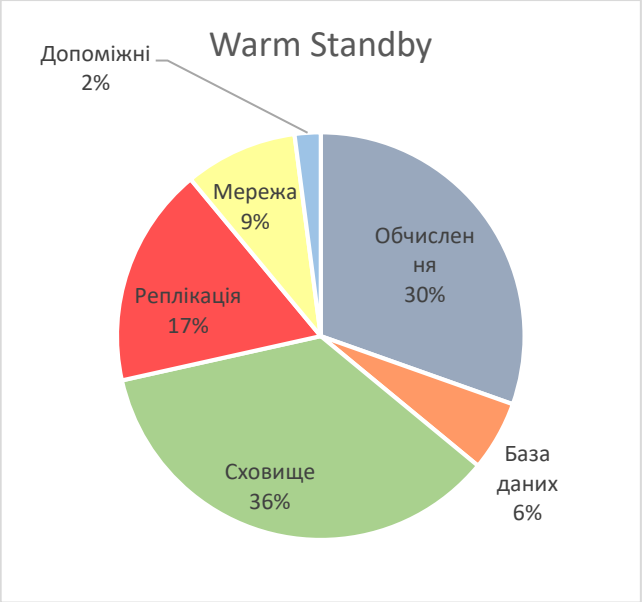

Рисунок 4.3 – Структура витрат стратегії Warm Standby

Витрати розподілені рівномірніше без явно вираженої домінуючої статті розходів. Поява значної частки витрат на обчислювальні потужності відображає ключову особливість стратегії – постійне утримання 30% обчислювальних потужностей у резервному регіоні.

Compute Savings Plans поширюється на 15 активних ВМ і 7 реплікаційних серверів t3.small, даючи економію близько $620. Database Savings Plans знижує

вартість Aurora secondary на $80. Сумарна економія близько $700 на місяць, або 10% від TCO.

4.2.5 Розрахунок структури витрат для Multi-Site Active/Active

Multi-Site Active/Active передбачає одночасне функціонування двох повноцінних production-середовищ у регіонах us-east-1 і us-west-2. Aurora Global Database у режимі write forwarding дає змогу приймати запити на запис в обох регіонах і делегувати навантаження primary-кластеру, Global Accelerator розподіляє вхідний трафік між регіонами через anycast IP-адреси, що забезпечує перемикання без зміни IP-адреси клієнта.

Розрахунок місячного TCO для Active/Active розрахований у таблиці 4.9.

Таблиця 4.9 – Місячний TCO стратегії Multi-Site Active/Active

Стаття	Розрахунок	$/міс
EC2 (100 інстансів m5.xlarge у двох регіонах)	100 × 140,16	14 016,00
EBS gp3 (50 ТБ, два регіони)	100 × $0.192 × 730 (50 в кожному регіоні)	4 000,00
Aurora primary cluster (4× db.r8g.large)	50000 × $0.08 (в обох регіонах)	805,92
Aurora secondary cluster (4× db.r8g.large)	4 × db.r8g.large × $0.276 × 730	805,92
Aurora storage (2 регіони × 10 ТБ)	4 × db.r8g.large × $0.276 × 730	2 000,00
Aurora cross-region I/O (multi-master, ×2)	10000 ГБ × 2 регіони × $0.10	511,86
Міжрегіональний трафік (non-Aurora)	2 × (дельта × 1024² / 8 КБ) × 30 днів / 1e6 + 200 млн	100,00
ALB × 2	5000 ГБ × $0.02 (EC2 cross-region traffic, Aurora виключена)	1 200,86
Global Accelerator	2 × ($16.43 + 100 LCU × $0.008 × 730)	93,25
KMS, cloudwatch, Route 53, WAF	$18.25/міс + 5000 ГБ egress × $0.015 DT-Premium	350,00

Кінець таблиці 4.9

Стаття	Розрахунок	$/міс
Місячний TCO on-demand		23 883,81
3 Compute SP 1-yr		19 636,96

Найбільша стаття витрат – це обчислювальні ресурси. 100 екземплярів (2 регіони 50 екзмплярів) m5.xlarge × $0,192 за годину × 730 годин дають $14 016.

EBS gp3 для всіх 100 ВМ дає $4 000, тобто 50 ТБ виділеного простору в обох регіонах за повною ставкою $0,08 за ГБ-міс.

Aurora primary cluster – це 4 екземпляри db.r8g.large (один writer плюс три readers) на $806, тобто 4 екземпляри по ціні $0,276/год протягом місяця. Aurora secondary cluster симетричний primary і теж коштує $806.

Сховище Aurora подвоюється, оскільки повна копія даних має бути в обох регіонах, що дає вартість $2000.

У Aurora cross regional I/O через активні запити на запис в обох регіонах обсяг реплікованих операцій подвоюється. При сумарному навантаженні у 2,56 млрд операцій за тарифом $0,20 за мільйон операцій вартість цього компонента становить $512.

Міжрегіональний трафік некластерної природи дає $100, це 5 ТБ × $0,02 за ГБ для координації EC2 між регіонами.

ALB подвоюється до $1201, оскільки потрібно по одному балансувальнику в кожному регіоні з активним трафіком близько 100 LCU. Global Accelerator коштує $93 як фіксована плата $18,25 плюс $0,015 за ГБ DT-Premium за пропуск 5 ТБ трафіку через магістральну мережу AWS. Допоміжні сервіси коштують фіксовано $350.

Структура витрат для стратегії Multisite Active/Active зображена на рисунку 4.4.

Рисунок 4.4 – Структура витрат стратегії Multisite Active/Active

Найбільшу долю витрат займають обчисленні – 59%. Compute (100 активних m5.xlarge у двох регіонах) формує 58,7% TCO, сховище – 25,1%, бази даних (8 інстансів Aurora) – 7%, мережеві сервіси – 6%, реплікація – 2,1%, допоміжні сервіси – 1%. Це принципово відрізняє Multisite Active/Active від решти стратегій, оскільки дублювання повного прикладного шару у двох регіонах робить обчислення основним фактором витрат, тоді як реплікаційні та мережеві компоненти стають другорядними.

Оскільки обчислення і бази даних формують 65% всіх витрат Compute Saving plans суттєво знижують TCO. Compute Savings Plans знижує вартість 100 ВМ на $3924, Database Savings Plans знижує на $322. Сумарна економія $4247 на місяць або 18%.

4.2.6 Аналіз розрахунків TCO

Розрахунки попередніх підпунктів дозволяють провести пряме порівняння чотирьох стратегій за єдиним набором метрик місячне TCO у двох цінових режимах, річна сукупна вартість і фактичні значення RTO та RPO для еталонного навантаження. Зведення цих показників представлено у таблиці 4.10.

Таблиця 4.10 – Зведене порівняння TCO та цільових метрик чотирьох DR-стратегій

Стратегія	TCO/міс on-demand, $	TCO/міс з SP, $	TCO/рік з SP, $	Фактичний RTO	Фактичний RPO
Backup and Restore	362	362	4 339	12–24 год	24 год
Pilot Light	4 043	3 933	47 194	10–30 хв	Секунди
Warm Standby	7 260	6 561	78 730	1–5 хв	Секунди
Multi-Site Active/Active	23 884	19 637	235 644	Секунди	<1 с

Стратегії DR мають різні структури витрат. Backup and Restore формується майже виключно зі сховища і міжрегіональної реплікації даних – ці дві категорії дають 92% всіх витрат. У Pilot Light переважають реплікація 31% і сховище 49% від витрат. Warm Standby має змішану структуру витрат. В Active/Active Compute-частина становить 14 016$ (59%), Database-частина становить 1 612$ (7%). Це означає, що при збільшенні навантаження на різні компоненти стратегії в плані вартості реагуватимуть по різному.

Savings Plans діють по-різному на різні стратегії. Backup and Restore не отримує економії взагалі, оскільки не утримує постійних обчислювальних ресурсів. Pilot Light економить 110 $/міс, або близько 2,8%, через мінімальну частку обчислень у структурі. Warm Standby економить 700 $/міс, що відповідає 10%. Active/Active отримує найбільший розмі економі – 4247 $/міс, або 17,8%. Стратегії з переважанням обчислювальної склажової виграють найбільше, причому SP не змінюють відносного позиціонування Backup and Restore і Pilot Light, проте суттєво скорочують розрив між Warm Standby і Active/Active: з 3,26 раза в режимі on-demand до 2,92 раза з SP, тобто розрив між двома найдорожчими стратегіями зменшується приблизно на 10%.

Порівняння Saving Plain для стратегій DR зображено на стовпчиковій діаграмі на рисунку 4.5.

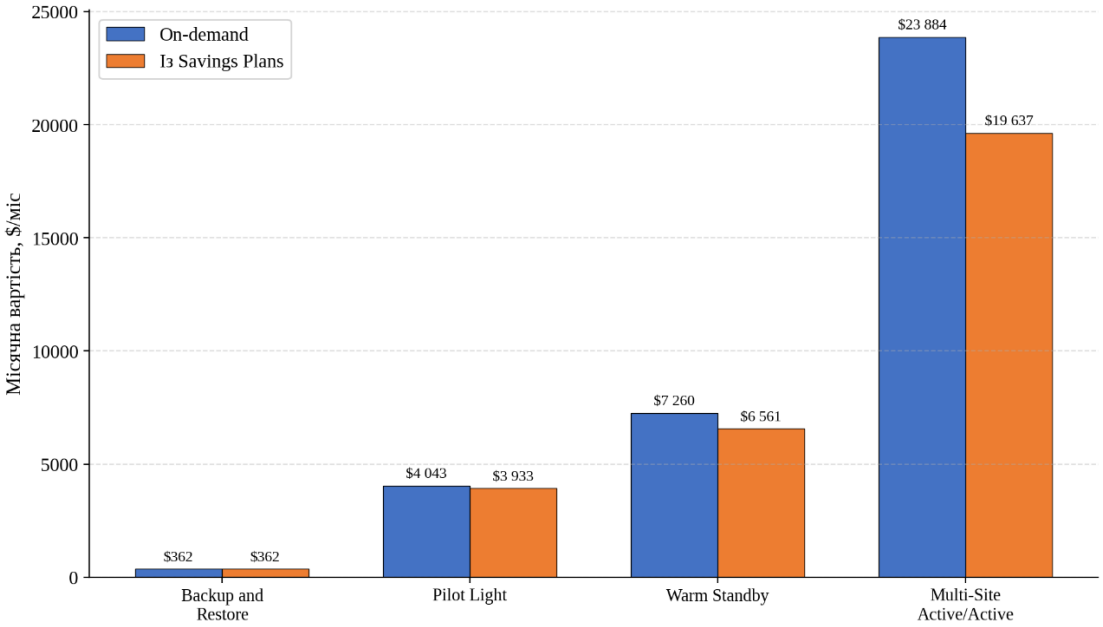

Рисунок 4.5 – Витрати на DR-стратегії

4.3 Аналіз економічної доцільності стратегій авайного відновлення

Було визначено, які стратегії задовольняють вимоги конкретного tier-класу за RTO/RPO, для деяких класів задовільними були декілька стратегій, проте крім технічних показників важливу роль для бізнесу відіграє і фінансова складова. Розрахунки TCO зафіксували діапазон можливих витрат для кожної стратегії. Однак самих значень витрат не достатньо, щоб обрати найоптимальнішу стратегію відновлення, оскільки такі розрахунки не враховують фінансові ризики від простою інфраструктури в разі інциденту.

4.3.1 Методологія розрахунку очікуваних втрат

Для розрахунку очікуваних втрат буде використовуватись метрика Annualized loss expectancy (ALE). ALE – це метрика, що використовується в управлінні ризиками для оцінки річної вартості потенційних збитків від виявленого ризику [30].

Annualized loss expectancy розраховується за формулою 4.1:

$$ALE = ARO \times SLE, \quad (4.1)$$

де *ARO* – річна частота виникнення (Annual Rate of Occurrence);

SLE – очікувана вартість одного збитку (Single Loss Expectancy).

Annual Rate of Occurrence, являє собою очікувану частоту, з якою виникатимуть певні загрози або події протягом одного року.

Очікувана тривалість збитків (SLE) – це очікувані грошові збитки або вплив від одноразового виникнення конкретної загрози або події.

Для визначення ARO буде використано оцінку з публічної історії регіональних інцидентів AWS за останні 10 років [31]. За період з 2026 по 2016 рік зафіксовано 5 регіональних інцидентів в регіоні us-east-1: відмова S3 28 лютого 2017 року, відмова Amazon Kinesis Event 25 листопада 2020 року, відмова AWS Service Event в 7 грудня 2021 року, відмова Amazon Kinesis Data Streams Service 30 липня 2024 року, відмова Amazon dynamodb 19 жовтня 2025 року. Таким чином, ARO становить 0,5.

Single Loss Expectancy буде розраховуватись за формулою 4.2:

$$SLE = Cost_{downtime} \times T_{downtime}, \quad (4.2)$$

де $Cost_{downtime}$ – вартість однієї години простою;

$T_{downtime}$ – час простою.

Значення вартості простою для tier-систем оцінено за діапазонами щорічного індустріального дослідження [32]. Конкретні числа моделі обрані в межах опублікованих діапазонів і узгоджені з прикладами систем. Параметри для трьох tier-класів подано у таблиці 4.11.

Таблиця 4.11 – Вартість простою для tier-систем

Tier	Cost_downtime, $/год	Положення в діапазоні індустріальних досліджень
Tier 1	115 000	High-criticality enterprise (середина)

Кінець таблиці 4.11

Tier 1	115 000	High-criticality enterprise (середина)
Tier 2	20 000	Mid-tier (верхня межа)
Tier 3	100	Low-criticality (continued IT payroll)

4.3.2 Розрахунок очікуваної річної втрати та сумарної вартості стратегії

На основі прийнятих параметрів обчислено очікувані річні втрати для кожної комбінації стратегії та tier-класу. Результати наведено у таблиці 4.12.

Таблиця 4.12 – ALE для трьох tier-класів за чотирма стратегіями

Стратегія	Tier 1 ($115 000/год)	Tier 2 ($20 000/год)	Tier 3 ($100/год)
Backup and Restore (T = 18 год)	1 035 000	180 000	900
Pilot Light (T = 0,33 год)	18 975	3 300	17
Warm Standby (T = 0,05 год)	2 875	500	3
Active/Active (T = 0,008 год)	460	80	0

ALE не достатній для прийняття рішення, оскільки він показує лише очікувану вартість простою, але не враховує витрат на підтримку інфраструктури. Рішення формується сумою TCO та ALE для знаходження вартості технічного рішення. Для TCO будуть братись розрахунки з Saving Plan. Сумарні витати подано у таблиці 4.13.

Таблиця 4.13 – Сумарні витрати ALE та TCO

Стратегія	TCO/рік	Tier 1	Tier 2	Tier 3
Backup and Restore	4 339	1 039 339	184 339	5 239 (мін.)
Pilot Light	47 194	66 169 (мін.)	50 494 (мін.)	47 211
Warm Standby	78 730	81 605	79 230	78 733

Кінець таблиці 4.13

Стратегія	TCO/рік	Tier 1	Tier 2	Tier 3
Active/Active	235 644	236 104	235 724	235 644

Оптимальна вартість для tier 2 і tier 3 збігається з технічно прийнятним варіантом з таблиці 4.1. Для tier 3 оптимальним вибором є Backup and Restore, для tier 2 – це Pilot Light. Для tier 1 ситуація відрізняється, оскільки найоптимальнішою стратегією з точки зору витрат виявилась Pilot Light, яка лише частково покриває потреби даного класу. Якщо організація не готова приймати ризик роботи оптимальним вибором є Warm Standby. Перехід від Warm Standby до Active/Active для Tier 1 є економічно невиправданим за поточних параметрів моделі.

4.3.3 Визначення оптимальної вартості простою для вибору стратегії

Розрахунок суми ALE та TCO у 4.3.2 показав оптимальну стратегію для трьох конкретних значень вартості простою, проте важливо розуміти, в якому діапазоні вартості простою кожна стратегія залишається оптимальною. Для визначення діапазонів треба знайти точку байдужості для кожної пари стратегій.

Точка байдужості – це точка, в якій дві різні альтернативи рівні [33].

Для визначення точки байдужості необхідно скласти рівняння витрат 4.3, де дві сторони представляють загальні витрати за кожним методом:

$$FC_a + VC_a \times Q = FC_b + VC_b \times Q, \quad (4.3)$$

де FC_a – постійні витрати методу *a* (fixed costs);

FC_b – постійні витрати методу *b* (fixed costs);

VC_a – змінні витрати на одиницю методу *a* (variable costs per unit);

VC_a – змінні витрати на одиницю методу *b* (variable costs per unit);

Q – обсяг або інший параметр, від якого залежать змінні витрати.

Розв’язання рівняння (4.3) відносно Q дає значення параметра, при якому сумарні витрати двох альтернатив збігаються, що є точкою байдужостості.

У контексті аналізу DR-стратегій річну TCO можна вважати фіксованою складовою, а ALE змінною складовою. Таким чином витрати можна вивести формулу 4.4:

$$C_{total} = TCO + C_{downtime} \times T_{downtime} \times ARO, \quad (4.4)$$

де C_{total} – загальна сума витрат;

$C_{downtime}$ – вартість простою однієї години;

TCO – загальна вартість утримання інфраструктури;

$T_{downtime}$ – час простою;

ARO – річна частота виникнення інцилентів.

Зіставляючи формулу (4.4) із рівнянням (4.3), у контексті аналізу DR-стратегій роль постійних витрат відіграє TCO стратегії, роль змінних витрат на одиницю відіграє добуток часу простою та ARO, а роль параметра обсягу – вартість простою бізнесу. Підставляючи ці позначення у метод складання рівнянь сумарних витрат для двох DR-стратегій a та b, можна отримати рівняння 4.5:

$$TCO_a + C_{downtime} \times T_a \times ARO = TCO_b + C_{downtime} \times T_b \times ARO, \quad (4.5)$$

де $C_{downtime}$ – загальна сума витрат;

TCO_a – загальна вартість утримання інфраструктури стратегії *a*;

T_a – час простою для стратегії *a*;

TCO_b – загальна вартість утримання інфраструктури стратегії *b*;

T_b – час простою для стратегії *b*;

ARO – річна частота виникнення інцидентів.

Якщо з цього рівняння винести вартість простою, можна отримати формулу 4.6:

$$C_{downtime} = \frac{TCO_b - TCO_a}{(T_a - T_b) \times ARO}. \quad (4.6)$$

Якщо підставити у формулу (4.6) конфігурації стратегій для обчисленого раніше ARO, можна отримати точки байдужості для трьох сусідніх пар стратегій. Результати наведено у таблиці 4.13.

Таблиця 4.13 – Точки байдужості для пар DR-стратегій

Стратегія нижчого рівня	Стратегія вищого рівня	Різниця TCO, $/рік	Різниця орієнтованого часу простою, Год	Вартість простою, $/год
Backup and Restore	Pilot Light	42 855	17,670	4 851
Pilot Light	Warm Standby	31 536	0,280	225 258
Warm Standby	Multisite Active/Active	156 914	0,042	7 472 073

На рисунку 4.6 зображені графіки лінійної залежності сумарних витрат стратегій DR від вартості години простою. Перетини прямих є точками байдужості для пар стратегій. На рисунку можна побачити, що сумарна вартість зростає з різною швидкістю для різних стратегій DR. Backup and Restore стартує з найнижчої точки, але має сумарні витрати зростають найшвидше при збільшенні вартості простою, що робить стратегію придатною лише для систем з низькою вартістю простою. Pilot Light, Warm Standby і Multi-Site Active/Active мають дедалі вищі стартові точки, але дедалі меншу швидкість зростання. Вони дорогі в утриманні, проте слабко реагують на зростання вартості просто. завдяки малим RTO.

Маючи точки байдужості можна побудувати діапазони, в яких кожна стратегія є фінансово виправданою. Розподіл наведено у таблиці 4.14.

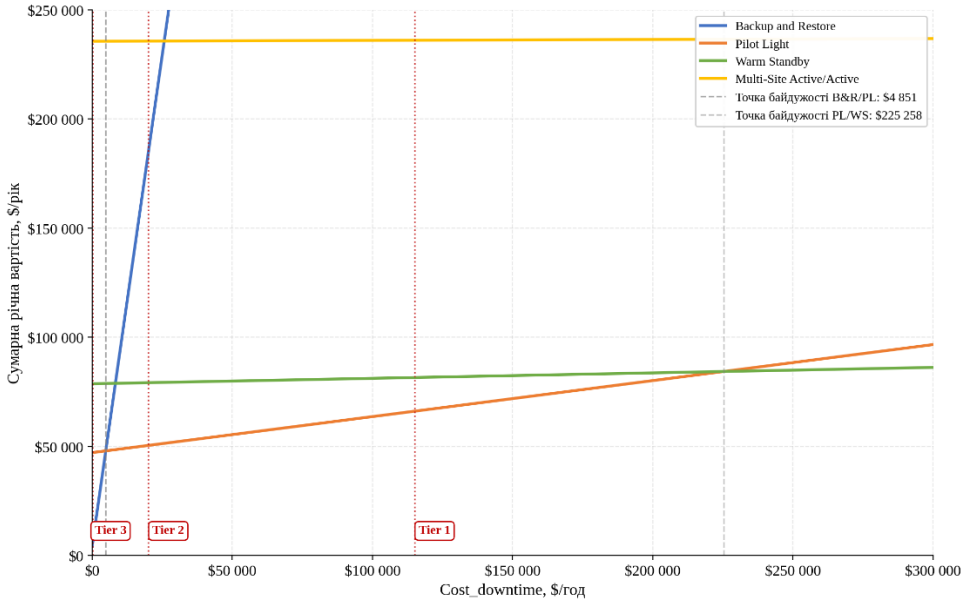

Рисунок 4.6 – Лінійні залежності сумарних витрат стратегій DR від вартості години простою

Таблиця 4.14 – оптимальна вартість простою для DR-стратегій

Стратегія	Діапазон вартості простою, $/год
Backup and Restore	0 - 4 851
Pilot Light	4 851 - 225 258
Warm Standby	225 258 - 7 471 143
Multi-Site Active/Active	Понад 7 471 143

Результати аналізу точки байдужості підтвердили висновки моделювання за критерієм мінімального сумарного балансу TCO та ALE. Для tier 3 оптимальною стратегією є Backup and Restore, для tier 2 є Pilot Light, для tier 1 оптимальними є Pilot Light або Warm Standby за умови строгого дотримання вимог RTO.

Multi-Site Active/Active не потрапляє в діапазон оптимальності для жодного з трьох tier-класів і її вибір спирається на регуляторні, репутаційні та контрактні фактори, що виходять за межі ALE-моделі.

4.4 Проблеми та особливості використання підходу

Запропонований підхід вибору DR-стратегії показав свою працездатність на еталонному навантаженні, проте має особливості і обмеження, що визначають межі його застосування і формують напрями подальших досліджень.

Підхід базується за двоступеневою схемою прийняття рішення. На першому етапі здійснюється технічна фільтрація стратегій за відповідністю RTO та RPO вимогам tier-класу системи, на другому виконується економічна оптимізація серед прийнятних варіантів через мінімізацію сумарних витрат TCO та ALE. На етапі підготовки моделі до конкретної організації ключовим є коректне визначення вартості простою для систем різного класу критичності. На відміну від TCO, який однозначно обчислюється з тарифів хмарного провайдера, вартість простою бізнесу включає прямі втрати виручки, втрачений час персоналу, штрафи SLA та репутаційні ризики, що не мають єдиного методу розрахунку і потребують окремого аналізу для кожної системи. Точність рекомендації залежить від точності цієї оцінки.

У підходу є обмеження, які треба враховувати при використанні результатів. По-перше, всі розрахунки зроблено для AWS, що дозволяє працювати з однією тарифною системою, але не дає прямого порівняння з Microsoft Azure та Google Cloud Platform. По-друге, числа отримано для конкретного еталонного робочого навантаження. Якщо збільшити або зменшити масштаб, співвідношення між стратегіями збережуться, але при зміні самої структури навантаження розрахунки треба робити заново. По-третє, вхідні значення вартості простою та ARO – це значення з публічних джерел, а не дані конкретних компаній. ARO оцінено лише по історії регіональних відмов AWS і не враховує інших ризиків, таких як кібератаки або помилки персоналу. По-четверте, підхід дає тільки економічне обґрунтування, але при остаточному рішенні треба враховувати регуляторні вимогами, SLA та репутаційні ризик.

Перспективними напрямами подальших досліджень є розширення моделі мульти-хмарними сценаріями з порівняльним аналізом вартості DR-рішень для

AWS, Azure та GCP, включно з гібридними рішеннями. Ще одним напрямком дослідження є стохастичне моделювання аварій з урахуванням різних класів загроз та тривалості простою. Це дозволить зробити розрахунок очікуваних щорічних втрат більш точним, тим самим покращуючи точність вибору оптимальної стратегії. Також можливим є дослідження з урахуванням регуляторних обмежень як окремого параметра вибору стратегії. Потенціал має застосування практик хаос-інжиніринг і внесення несправностей як механізмів визначення фактичних RTO та RPO, що дозволить знизити розрив між декларованими та реальними показниками відновлення стратегій.

ВИСНОВКИ

У процесі виконання кваліфікаційної магістерської роботи було здійснено комплексне дослідження методів аварійного відновлення інформаційних систем у хмарних середовищах із розробкою методики економічно обґрунтованого вибору DR-стратегії для систем різного рівня критичності. Поставлена мета – проаналізувати сучасні методи аварійного відновлення в хмарі та сформувати кількісно обґрунтований підхід до вибору стратегії з урахуванням сукупної вартості володіння і очікуваних втрат від простою – досягнута шляхом поєднання теоретичного аналізу, порівняльного огляду DR-сервісів провідних хмарних провайдерів і сценарного економіко-архітектурного моделювання.

У першому розділі обґрунтовано значущість аварійного відновлення як критичного компонента стратегії безперервності бізнесу для систем, розгорнутих у хмарних середовищах. Розглянуто базові поняття DR і безперервності бізнесу, наведено класифікацію загроз. Формалізовано метрики оцінки ефективності відновлення. Розкрито модель розподіленої відповідальності між хмарним провайдером і клієнтом, що визначає межі обов'язків сторін у питаннях резервного копіювання, реплікації та відновлення.

У другому розділі проведено порівняльний аналіз сучасних методів забезпечення аварійного відновлення. Розглянуто типи резервного копіювання, механізми блочних знімків, методи реплікації даних і їх вплив на цільові показники RTO та RPO. Виконано порівняльний огляд стратегій аварійного відновлення з аналізом переваг, обмежень і типових сфер застосування. Розглянуто інструменти автоматизації DR.

У третьому розділі здійснено порівняльний аналіз DR-сервісів провідних хмарних провайдерів: Amazon Web Services, Microsoft Azure та Google Cloud Platform. Проведено зіставлення сервісів резервного копіювання та забезпечення незмінності даних, рішень draas з блочною реплікацією, механізмів реплікації керованих баз даних та інструментів маршрутизації трафіку при failover.

У четвертому розділі розроблено сценарну економіко-архітектурну модель вибору DR-стратегії на платформі AWS. Виконано класифікацію інформаційних систем за трьома рівнями критичності з прив'язкою до конкретних значень RTO та RPO. Для еталонного навантаження обчислено сукупну вартість володіння для чотирьох DR-стратегій. На основі моделі ALE обчислено очікувані річні втрати для трьох tier-класів і визначено стратегію з мінімальним сумарним балансом TCO та ALE для кожного класу. Застосовано метод аналізу точки байдужості для встановлення діапазонів вартості простою, у яких кожна стратегія є економічно оптимальною. Отримані пороги переходу між сусідніми стратегіями.

Узагальнюючи результати, можна констатувати, що оптимальний вибір DR-стратегії суттєво відрізняється від технічно максимального. Для типового корпоративного навантаження більшість систем потрапляє у діапазон оптимальності Pilot Light – стратегії, що поєднує прийнятні показники RTO та RPO з помірною вартістю утримання інфраструктури. Стратегія Multi-Site Active/Active, що часто рекомендується для критичних систем, у моделі чистого економічного обґрунтування не отримує: її вибір виправданий лише при надзвичайно високій вартості простою або зумовлений регуляторними вимогами, репутаційними ризиками, контрактними SLA-зобов'язаннями.

Практична цінність роботи полягає у можливості застосування розробленої методики для планування DR в організаціях різного масштабу. Модель розрахунку TCO та діапазонів економічної оптимальності для чотирьох DR-стратегій може бути адаптована до конкретних умов організації шляхом підстановки власних параметрів вартості інфраструктури і вартості простою. Результати дослідження можуть слугувати основою для прийняття обґрунтованих архітектурних рішень фахівцями з хмарної архітектури, спеціалістами з інформаційної безпеки та керівниками IT-відділів при формуванні бюджетів на DR.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Гунько Б.Г. Аналіз ефективності методів аварійного відновлення систем в хмарних середовищах // 30-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6. Харків: ХНУРЕ. 2026. С. 425-426

2. Contingency planning guide for federal information systems / M. Swanson та ін. Gaithersburg, MD : National Institute of Standards and Technology, 2010. URL: https://doi.org/10.6028/nist.sp.800-34r1 (дата звернення: 01.03.2026).

3. Business Impact Analysis. *Plan Ahead for Disasters | Ready.gov*. URL: https://www.ready.gov/business/planning/impact-analysis (дата звернення: 01.03.2026).

4. Cloud Security Alliance (2023). Treacherous 12: Top Threats to Cloud Computin. Https://cloudsecurityalliance.org/research/top-threats/treacherous-12/ (дата звернення 06.02.26)

5. RTO vs RPO: Key Differences in Disaster Recovery Planning. *Sentinelone*. URL: https://www.sentinelone.com/cybersecurity-101/cloud-security/rto-vs-rpo/ (дата звернення: 11.03.2026).

6. Establishing RPO and RTO Targets for Cloud Applications | Amazon Web Services. *Amazon Web Services*. URL: https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-of-on-premises-applications-to-aws/recovery-objectives.html (дата звернення: 10.03.2026).

7. Shared responsibility in the cloud - Microsoft Azure. *Microsoft Learn: Build with answers in reach*. URL: https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility (дата звернення: 13.03.2026).

8. REL13-BP04 Manage configuration drift at the DR site or Region – Reliability Pillar. URL: https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/rel_planning_for_recovery_config_drift.html (дата звернення: 17.03.2026).

9. What is IAC? - circleci. *Circleci*. URL: https://circleci.com/topics/infrastructure-as-code-iac/ (date of access: 14.03.2026).

10. Drift Detection in iac: Prevent Your Infrastructure from Breaking | env zero. *Env zero: Cloud Governance Platform for Faster Infrastructure Delivery*. URL: https://www.env0.com/blog/drift-detection-in-iac-prevent-your-infrastructure-from-breaking (дата звернення: 16.03.2026).

11. Differential backup - SQL Server. *Microsoft Learn: Build with answers in reach*. URL: https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/create-a-differential-database-backup-sql-server?View=sql-server-ver17 (дата звернення: 18.03.2026).

12. Block-level Backup. Documentation. URL: https://documentation.commvault.com/11.40/commcell-console/block_level_backup.html (дата звернення: 17.03.2026)

13. Point-in-time recovery (PITR) overview | Spanner | Google Cloud Documentation. *Google Cloud Documentation*. URL: https://docs.cloud.google.com/spanner/docs/pitr (дата звернення: 20.03.2026).

14. Стратегії резервного копіювання: що таке метод 3-2-1?. *Experience - Discover How Dropbox Empowers Teams - Dropbox*. URL: https://experience.dropbox.com/uk-ua/resources/3-2-1-backup-strategy (дата звернення: 20.03.2026).

15. AWS Well-Architected Framework Disaster Recovery of Workloads on AWS: Recovery in the Cloud. URL: https://docs.aws.amazon.com/pdfs/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-workloads-on-aws.pdf#disaster-recovery-options-in-the-cloud (дата звернення 21.03.2026)

16. C H A P T E R 2 - Replication and Synchronization Modes. *Moved*. URL: https://docs.oracle.com/cd/E19359-01/819-6148-10/chap2.html (дата звернення: 23.03.2026)

17. Mysql :: mysql 8.4 Reference Manual :: 19.4.10 Semisynchronous Replication. *Semisynchronous Replication | dev.mysql.com*.

URL: https://dev.mysql.com/doc/refman/8.4/en/replication-semisync.html (дата звернення: 20.03.2026).

18. What is a quorum in distributed databases?. *Milvus | High-Performance Vector Database Built for Scale*. URL: https://milvus.io/ai-quick-reference/what-is-a-quorum-in-distributed-databases (дата звернення: 25.03.2026).

19. Disaster Recovery as Code: Automating Business Continuity | qodequay. *Human-Centered Design Thinking Consulting Services*. URL: https://www.qodequay.com/disaster-recovery-code-automating-business-continuity?Srsltid=afmbooqwc5nhijjn9stsac_Mf1LSw35xz6mRpG7wwo4kRVY36yKkXrhv (дата звернення: 27.03.2026).

20. Franke U. IT service outage cost: case study and implications for cyber insurance. The Geneva Papers on Risk and Insurance - Issues and Practice. 2020. Т. 45, № 4. С. 760–784. URL: https://doi.org/10.1057/s41288-020-00177-4 (дата звернення: 02.04.2026).

21. Gitops. *Gitops*. URL: https://www.gitops.tech/ (дата звернення: 24.04.2026).

22. What is AWS Backup? - AWS Backup. URL: https://docs.aws.amazon.com/aws-backup/latest/devguide/whatisbackup.html (дата звернення: 24.04.2026).

23. Locking objects with Object Lock - Amazon Simple Storage Service. URL: https://docs.aws.amazon.com/amazons3/latest/userguide/object-lock.html (date of access: 15.04.2026).

24. What is Azure Backup? - Azure Backup. *Microsoft Learn: Build with answers in reach*. URL: https://learn.microsoft.com/en-us/azure/backup/backup-overview (дата звернення: 16.04.2026).

25. Backup vaults for immutable and indelible backups | Backup and DR | Google Cloud Documentation. *Google Cloud Documentation*. URL: https://cloud.google.com/backup-disaster-recovery/docs/concepts/backup-vault (дата звернення: 16.04.2026).

26. About Azure Site Recovery - Azure Site Recovery. *Microsoft Learn: Build with answers in reach*. URL: https://learn.microsoft.com/en-us/azure/site-recovery/site-recovery-overview (дата звернення: 18.04.2026).

27. Global tables - multi-active, multi-Region replication - Amazon dynamodb.
URL: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/globaltables.html (дата звернення: 19.04.2026).

28. Creating Amazon Route 53 health checks - Amazon Route 53. URL: https://docs.aws.amazon.com/Route53/latest/developerguide/dns-failover.html (дата звернення: 20.04.2026).

29. Compute Savings Plans. *Amazon Web Services, Inc*. URL: https://aws.amazon.com/savingsplans/compute-pricing (дата звернення: 22.04.2026).

30. What Is Annual Loss Expectancy. *Apache2 Ubuntu Default Page: It works*. URL: https://reviewbooku.com/review/what-is-annual-loss-expectancy-4951969 (дата звернення: 27.04.2026).

31. AWS Post-Event Summaries. *Amazon Web Services, Inc*. URL: https://aws.amazon.com/premiumsupport/technology/pes/ (date of access: 27.05.2026).

32. ITIC 2024 Hourly Cost of Downtime Part 2 – Information Technology Intelligence Consulting. Information Technology Intelligence Consulting – Information Technology Intelligence Consulting. URL: https://itic-corp.com/itic-2024-hourly-cost-of-downtime-part-2/ (дата звернення: 27.05.2026).

33. Indifference curves and the marginal rate of substitution. *The Economy 2.0*. URL: https://books.core-econ.org/the-economy-v1/book/text/leibniz-03-02-01.html (дата звернення: 04.05.2026).