ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти магістр

на тему: **«Методологія перенесення бінарних файлів із корпоративних баз даних у хмарні сховища (на прикладі Amazon S3)»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи та технології
спеціальності 126 Інформаційні системи та технології
ступеня вищої освіти магістр
групи 126ІСТ_мд_2023
Онищенко Ростислав Русланович
Керівник: Флегантов Леонід Олексійович
Рецензент: Ковальчук Станіслав Богданович

Полтава – 2024 року

ВСТУП

Актуальність теми. Перенесення бінарних файлів із локальних корпоративних баз даних у хмарні сховища є актуальним рішенням, що обумовлене низкою сучасних вимог до ефективності, безпеки та масштабованості даних в умовах зростання їх обсягу та необхідності швидкого доступу. Зокрема, Amazon S3 як хмарне сховище є популярним вибором завдяки своїм технічним можливостям, високому рівню надійності та економічній доцільності.

Сучасні підприємства постійно генерують значні обсяги бінарних даних, таких як документи, медіафайли, звіти, результати вимірювань тощо. Збереження цих даних на локальних серверах створює ризики, пов'язані з браком фізичного простору, збільшенням витрат на адміністрування серверів і технічним обслуговуванням. Перенесення бінарних файлів у Amazon S3 дозволяє зняти навантаження з локальної інфраструктури, оскільки S3 забезпечує практично необмежене сховище даних, яке автоматично масштабується.

Хмарні сховища, такі як Amazon S3, забезпечують високу доступність даних у будь-який час і з будь-якого місця. Це особливо важливо для підприємств, які мають розподілену інфраструктуру, працюють у віддаленому режимі або співпрацюють із міжнародними партнерами. Дані в Amazon S3 легко інтегруються з іншими сервісами AWS, що дає можливість автоматизувати обробку файлів, скорочуючи час на їх отримання та обробку.

Надійне зберігання і захист інформації є одним із важливих завдань у корпоративному середовищі. Хмарні сховища, зокрема Amazon S3, підтримують шифрування даних як під час передавання, так і при зберіганні файлів. S3 надає можливість налаштовувати доступ до файлів, використовуючи політики доступу, шифрування та контрольовані дозволи для підвищення захисту даних. Це відповідає стандартам інформаційної безпеки та дозволяє підприємствам дотримуватись вимог щодо конфіденційності.

Утримання та обслуговування локальних серверів часто є дорогим і потребує значних ресурсів, включаючи енергоспоживання, технічне

обслуговування і оновлення обладнання. Amazon S3 пропонує різні класи зберігання, такі як S3 Standard, S3 Standard-IA (Infrequent Access), і S3 Glacier для архівних даних. Ці класи дозволяють гнучко управляти витратами в залежності від частоти доступу до файлів, тим самим оптимізуючи витрати на зберігання в порівнянні з локальними серверами.

Amazon S3 забезпечує динамічну масштабованість і гнучкість, що дозволяє адаптуватися до зростання обсягів даних без необхідності додаткових інвестицій у локальне обладнання. Це важливо для підприємств, що розширюються, адже S3 дозволяє зберігати великі масиви бінарних даних без затримок і з високою продуктивністю.

Таким чином, актуальність теми перенесення бінарних файлів із корпоративних баз даних у хмарні сховища, такі як Amazon S3, обумовлена прагненням підприємств до зменшення витрат, підвищення доступності даних, забезпечення безпеки та масштабованості інфраструктури зберігання.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконана у відповідності до науково-дослідної ініціативної теми «Організаційно-методологічні аспекти впровадження інформаційно-комунікаційних систем і технологій в управлінні діяльністю сучасних організацій та підприємств за умов переходу до цифрової економіки» ДРН 0123U105060.

Метою роботи є визначення ефективних підходів до перенесення бінарних файлів із корпоративних баз даних у хмарні сховища на прикладі Amazon S3.

Завдання роботи:

- розглянути основні підходи до зберігання бінарних файлів у корпоративних базах даних та їх обмеження;
- проаналізувати сучасні хмарні рішення для зберігання бінарних файлів, зокрема можливості Amazon S3, та порівняти їх з традиційними методами зберігання;
- дослідити методи інтеграції корпоративних баз даних (на прикладі PostgreSQL) з хмарними сховищами даних;

- визначити оптимальні алгоритми та інструменти для перенесення бінарних файлів у Amazon S3, враховуючи безпеку та економічну ефективність;

- розробити практичні рекомендації щодо використання методології перенесення бінарних файлів у корпоративному середовищі.

Об'єкт дослідження: хмарні технології оптимізації зберігання бінарних файлів у корпоративних базах даних.

Предмет дослідження: технології та методи перенесення бінарних файлів із локальних корпоративних баз даних у хмарне сховище Amazon S3.

Методи дослідження: аналіз науково-технічної літератури з питань зберігання даних у хмарних сховищах, емпіричне дослідження методів перенесення даних, моделювання процесів інтеграції корпоративних баз даних з Amazon S3, а також порівняльний аналіз ефективності різних підходів до зберігання бінарних файлів.

Інформаційна база дослідження: науково-технічна література – книги та посібники з питань управління даними, хмарних технологій і баз даних, що висвітлюють сучасні підходи до зберігання бінарних файлів та їх перенесення; наукові статті та публікації з тематики хмарних обчислень, оптимізації сховищ даних та інтеграції корпоративних баз даних з хмарними платформами, доступні в наукових базах даних Google Scholar, IEEE Xplore, SpringerLink, ScienceDirect, Scopus, ResearchGate; офіційна документація, ресурси та довідкові матеріали Amazon Web Services (AWS); національні та міжнародні стандарти з інформаційної безпеки та зберігання даних; аналітичні матеріали, блоги та статті провідних експертів з інформаційних технологій та управління даними з питань ефективності перенесення файлів у хмару; дослідження з питань інтеграції корпоративних баз даних з хмарними платформами; приклади використання Amazon S3 для зберігання великих обсягів даних у різних галузях.

Елементи наукової новизни: порівняльний аналіз технологій та підходів до зберігання бінарних файлів у хмарних сховищах на прикладі Amazon S3 дозволив визначити найбільш ефективні методи для перенесення корпоративних даних із баз даних, таких як PostgreSQL, з урахуванням частоти доступу та вимог до

безпеки; дослідження можливостей використання різних класів зберігання Amazon S3, таких як Standard, Intelligent-Tiering, та Glacier, показало потенціал для оптимізації витрат і підвищення відмовостійкості; розробка методології, що включає інтеграцію AWS SDK, CLI та інструментів моніторингу, дозволила підвищити ефективність процесу переносу, гнучкість управління доступом і автоматизацію життєвого циклу даних.

Практична значущість: робота надає практичні рекомендації щодо вибору методів і технологій для перенесення бінарних файлів із корпоративних баз даних у хмарні сховища (на прикладі Amazon S3), що дозволяє оптимізувати процес управління даними; запропоновані підходи до структурування, шифрування та організації bucket-ів у хмарному сховищі сприяють зниженню витрат на зберігання, підвищенню доступності та захисту даних; результати дослідження можуть бути використані для створення гнучкої та економічної системи зберігання бінарних файлів у корпоративному середовищі.

Апробація результатів дослідження. За результатами проведеного дослідження опубліковано тези доповіді: «Автоматизація сповіщень при перенесенні файлів у хмарне сховище Amazon S3», Матер. XXI щорічного міждисциплінарного семінару «Студентські роботи за науковою тематикою кафедри інформаційних систем та технологій ННІ ЕУП та ІТ ПДАУ», 20 листопада 2024 року, м. Полтава.

Структура та обсяг кваліфікаційної роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Основний текст роботи викладений на 68 сторінках, містить 24 рисунки і 29 таблиць. Список використаних джерел налічує 50 найменувань.

РОЗДІЛ 1
ТЕОРЕТИЧНІ АСПЕКТИ ПЕРЕНЕСЕННЯ БІНАРНИХ ФАЙЛІВ У ХМАРУ

1.1 Визначення та основні типи бінарних файлів

Бінарні файли – це файли, які зберігають дані в двійковому форматі (наборі 0 і 1), що відрізняє їх від текстових файлів, які зберігають інформацію в форматі символів ASCII або Unicode. Бінарні файли використовуються для зберігання різних типів даних, таких як зображення, аудіо, відео, документи тощо. Оскільки ці файли мають закодовану структуру, їх не можна легко читати або редагувати у текстовому редакторі без спеціального програмного забезпечення [1, 2].

Приклад двійкового файлу, який складається з ряду послідовних байтів, вишикуваних один за одним, і може бути інтерпретований лише цільовою платформою (у цьому випадку – Unix), представлений на рисунку 1.1.

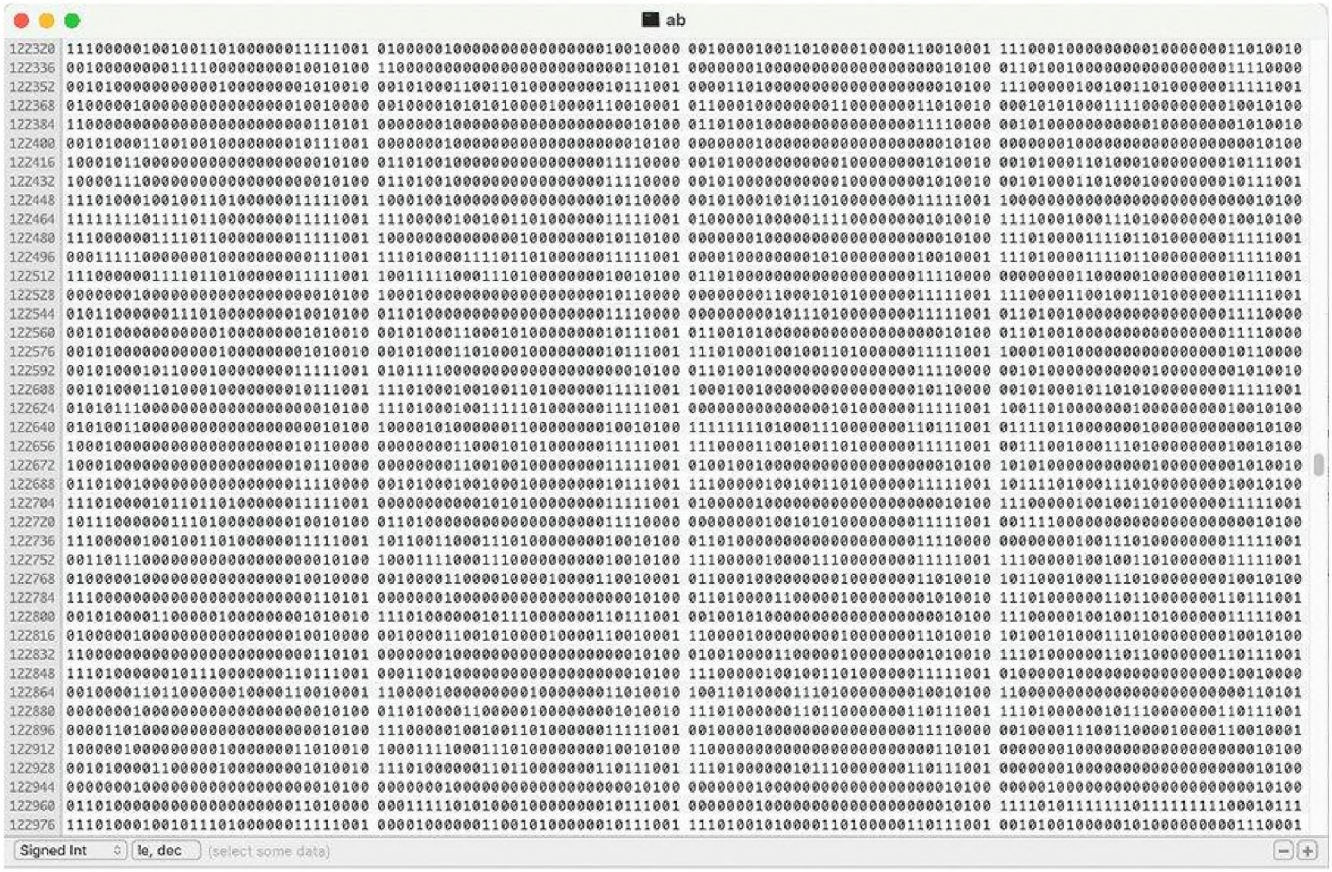

Рисунок 1.1 – Приклад двійкового файлу на платформі Unix [3]

На рисунку 1.2 представлений шістнадцятковий дамп 318-байтової піктограми Вікіпедії: замість кожного байту файлу виводиться його значення у шістнадцятковій системі числення; перший стовпець нумерує початкову адресу рядка, а знак (*) вказує на повторення.

Рисунок 1.2 – Шістнадцятковий дамп 318-байтової піктограми Вікіпедії [4]

Загальні відомості про основні типи бінарних файлів, які використовуються в різних сферах, таких як мультимедіа, програмування, наукові дослідження тощо, представлені у таблиці 1.1.

Таблиця 1.1 – Загальні відомості про основні типи бінарних файлів

Категорія файлів	Тип файлу	Розширення	Опис
Документи	PDF, Word, Excel	.pdf, .docx, .xlsx	Зберігають структуровані дані та текстову інформацію, включаючи зображення та таблиці
Зображення	JPEG, PNG, GIF	.jpg, .jpeg, .png, .gif	Використовуються для зберігання графічних зображень у різних форматах
Аудіофайли	MP3, WAV, FLAC	.mp3, .wav, .flac	Формати для зберігання аудіоданих з різними рівнями якості та стиснення
Відеофайли	MP4, AVI, MKV	.mp4, .avi, .mkv	Зберігають відеоінформацію, часто містять декілька потоків, зокрема аудіо та субтитри
Архіви	ZIP, RAR, 7z	.zip, .rar, .7z	Містять стислі дані або декілька файлів у одному контейнері, що економить місце

Продовження таблиці 1.1

Програмні файли	EXE, DLL, SO	.exe, .dll, .so	Виконувані файли та бібліотеки, які містять код для запуску програм та додатків
Наукові дані	HDF5, NetCDF, FITS	.hdf5, .nc, .fits	Спеціалізовані формати для зберігання та обробки великих наборів наукових даних
Бази даних	SQLite, MDB, ACCDB	.sqlite, .mdb, .accdb	Формати для зберігання даних у структурованих базах, зокрема для SQL-запитів
Логи та системні дані	LOG, BIN, CFG	.log, .bin, .cfg	Файли, що зберігають логи, конфігурації та системні дані для моніторингу та налаштування систем

Наступна діаграма описує розподіл різних типів бінарних файлів за частотою їх використання. Основну частку займають документи (25%) та зображення (20%), тоді як аудіо- та відеофайли становлять по 15% кожен. Архіви та програмні файли займають по 10%, наукові дані – 5% (рисунок 1.3).

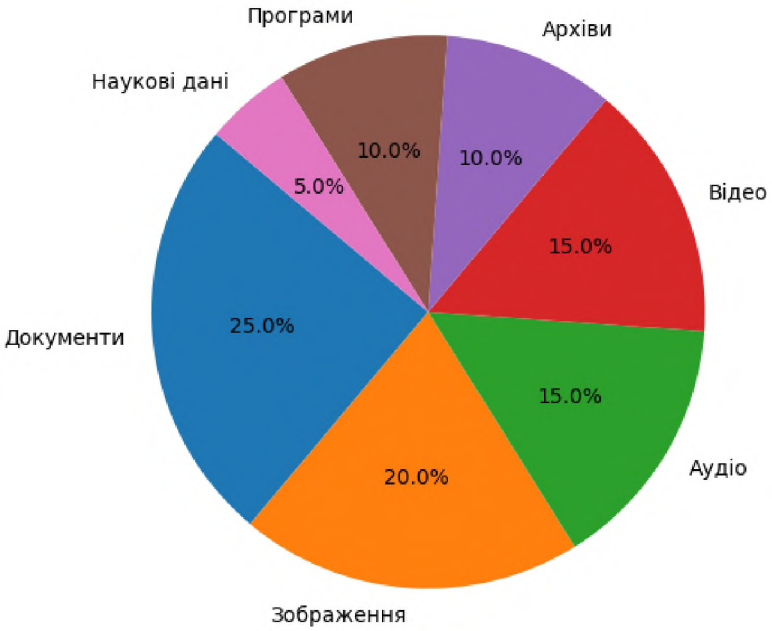

Рисунок 1.3 – Розподіл типів бінарних файлів за частотою їх використання

Отже, бінарні файли є важливою частиною корпоративних процесів, забезпечуючи зберігання мультимедійної інформації, документів та спеціалізованих даних. Оптимізація їх зберігання буде корисною для зменшення витрат, підвищення доступності та покращення управління даними [5, 6].

1.2 Проблєми зберігання бінарних файлів у корпоративних базах даних

Бінарні файли у корпоративних базах даних включають документи, мультимедійні файли, архіви та журнали (логи), що використовується для забезпечення ділових процесів (таблиця 1.2.).

Таблиця 1.2 – Основні типи бінарних файлів у корпоративних базах даних

Тип бінарного файлу	Опис	Приклади
Документи	Файли документів, що використовуються для бізнес-даних, звітності та ін.	PDF, DOCX, XLSX, TXT
Зображення	Графічні файли для візуального контенту	JPG, PNG, SVG
Аудіо- та відеофайли	Мультимедійні файли, які забезпечують аудіо- та відео контент	MP3, MP4, AVI
Архіви	Стислі файли для збереження та передачі великих обсягів даних	ZIP, RAR
Логи	Логи операцій для відстеження дій користувачів та процесів у системі	LOG, JSON

Типова структура зберігання бінарних файлів у корпоративних базах даних така, що основну частку займають документи та мультимедійні файли, а інші типи даних, зазвичай, наявні в менших обсягах [7, 8] (рисунок 1.4).

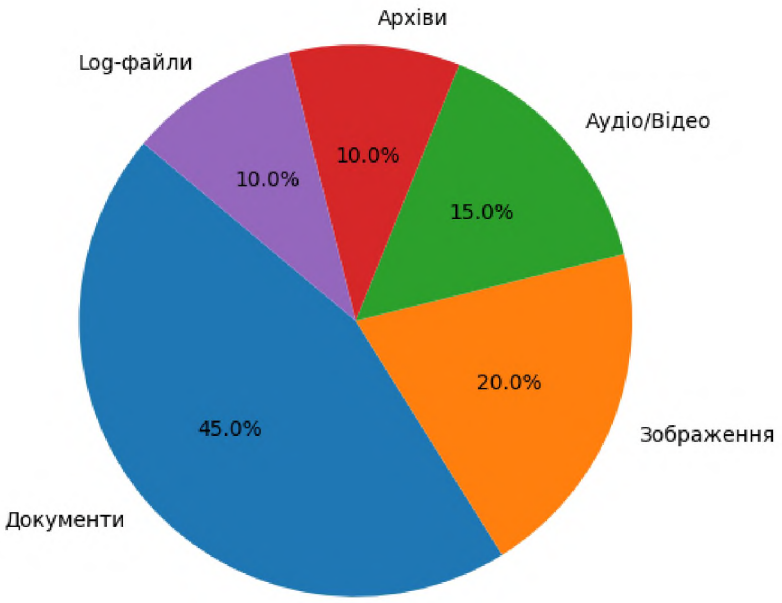

Рисунок 1.4 – Типова структура зберігання бінарних файлів у корпоративній базі даних

На діаграмі (рисунок 1.2) представлено типовий розподіл бінарних файлів у корпоративній базі даних, де основну частку займають документи (45%) та зображення (20%). Решту становлять аудіо- та відеофайли (15%), архіви (10%) та логи (10%). Така структура відображає стандартний набір бінарних даних, які зберігаються в корпоративних базах і створюють виклики для продуктивності й ефективності зберігання.

Зберігання великих обсягів бінарних файлів у традиційних локальних базах даних супроводжується низкою проблєм, що знижують ефективність роботи системи. Основні труднощі включають збільшення обсягу бази даних, зменшення продуктивності і додаткові витрати на адміністрування (таблиця 1.3).

Таблиця 1.3 – Проблєми зберігання великих обсягів бінарних файлів у традиційних базах даних

Проблєма	Опис	Наслідки
Великий обсяг зберігання	Збереження бінарних файлів значно збільшує розмір бази даних, що потребує додаткових ресурсів	Підвищені вимоги до пам'яті, збільшення витрат на обладнання та обслуговування
Зниження продуктивності запитів	Збільшення обсягу даних ускладнює обробку запитів і призводить до затримок у доступі до даних	Триваліший час обробки запитів, зниження швидкості взаємодії з системою
Проблєми з резервним копіюванням	Бінарні файли збільшують час і обсяг для резервного копіювання баз даних, що ускладнює процеси відновлення	Затримки під час резервного копіювання, збільшення часу відновлення після збоїв
Обмеження на типи зберігання даних	Традиційні реляційні бази даних не оптимізовані для роботи з великими бінарними об'єктами	Використання складних підходів для інтеграції зберігання даних
Безпекові ризики	Зберігання чутливих файлів у базі даних збільшує ризики порушення безпеки, оскільки усі дані знаходяться у спільному середовищі	Високий ризик втрати конфіденційності, потреба в шифруванні та контролі доступу

Отже, зберігання бінарних файлів у традиційних базах даних потребує оптимізації через високе навантаження на сервери, поступове зниження продуктивності та постійне зростання витрат [9, 10].

Використання хмарних сховищ, зокрема Amazon S3, може значно покращити ефективність зберігання бінарних даних за рахунок зниження витрат, підвищення доступності та забезпечення масштабованості [11, 12].

1.3 Сучасні рішення для зберігання даних

Сучасні підприємства та організації все частіше обирають хмарні сховища для зберігання даних, адже вони пропонують гнучкі, масштабовані та економічно ефективні рішення.

Вибір між хмарними сховищами та локальними серверами залежить від багатьох факторів, серед яких обсяги даних, вимоги до безпеки та доступу, потреба в автоматизації і розподіленій обробці даних [13-17].

На ринку хмарних послуг існує декілька основних постачальників хмарних сховищ, які пропонують різні функціональні можливості (таблиця 1.4).

Таблиця 1.4 – Популярні хмарні сховища для зберігання даних

Хмарне сховище	Провайдер	Важливі можливості	Особливості
Amazon S3	Amazon Web Services	Масштабоване сховище об'єктів, підтримка класів зберігання (S3 Standard, S3 Glacier), інтеграція з іншими сервісами AWS	Підходить для зберігання великих масивів даних з можливістю автоматизації
Azure Blob Storage	Microsoft	Зберігання об'єктів для всіх типів файлів, підтримка Hot, Cool та Archive шарів для управління витратами	Глибока інтеграція з сервісами Microsoft, хороша підтримка аналітики
Google Cloud Storage	Google	Гнучке сховище для будь-яких типів даних, класи Coldline та Archive для оптимізації зберігання	Інтеграція з BigQuery, висока продуктивність для обробки великих даних
IBM Cloud Object Storage	IBM	Широкий вибір класів зберігання, підтримка багатообласного зберігання, вбудована підтримка шифрування	Висока надійність, оптимізоване для використання у великих підприємствах
DigitalOcean Spaces	DigitalOcean	Простий у використанні інтерфейс, оптимізоване для малого бізнесу, сумісне з S3 API	Простота налаштувань, обмежена функціональність для великих даних

Найбільш популярними серед представлених варіантів є хмарні сховища Amazon S3, Microsoft Azure Blob Storage та Google Cloud Storage. Діаграма, яка відображає частку популярності провідних хмарних рішень, засновану на загальній кількості користувачів і інтеграцій, представлена на рисунку 1.5.

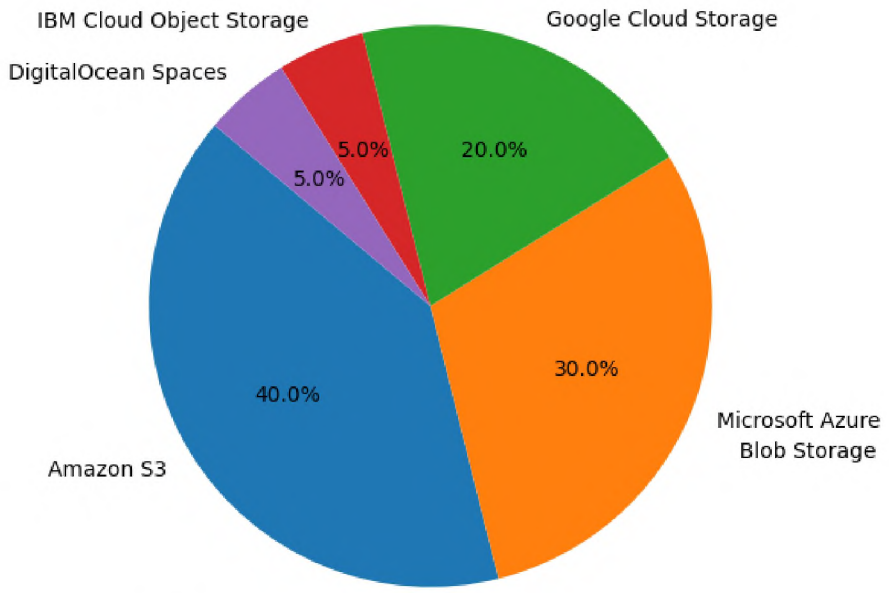

Рисунок 1.5 – Діаграма популярності хмарних сховищ даних

Таким чином, Amazon S3 займає лідируючу позицію (40%), що обумовлено широкими можливостями та гнучкістю інтеграції з іншими сервісами AWS. Microsoft Azure Blob Storage посідає друге місце (30%) завдяки інтеграції з сервісами Microsoft. Google Cloud Storage, IBM Cloud Object Storage та DigitalOcean Spaces користуються меншою популярністю, але кожне з них має свої унікальні особливості [18-20]. Порівняння переваг і недоліків хмарних сховищ та локальних серверів показує, що хмара є ефективним рішенням для масштабованості, безпеки та зменшення витрат (таблиця 1.2).

Таблиця 1.2 – Порівняння характеристик хмарних сховищ даних та локальних серверів

Критерій	Хмарне сховище	Локальні сервери
Масштабованість	Гнучка автоматична масштабованість	Обмежена, потребує фізичного розширення
Економічна ефективність	Плата тільки за використані ресурси, мінімізація витрат на обслуговування	Висока початкова вартість обладнання та обслуговування
Доступність	Доступ до даних з будь-якого місця і пристрою	Доступ обмежений внутрішньою мережею
Безпека	Високий рівень шифрування та політик доступу, що залежить від провайдера	Повний контроль над фізичною безпекою даних

Продовження таблиці 1.2

Відновлення після збоїв	Автоматичне резервне копіювання, реплікація	Потребує додаткових налаштувань резервування
Затримки	Можливі затримки в передачі даних через мережу	Мала затримка завдяки локальному доступу
Обмеження доступу	Доступ до даних може бути обмежений через політики хмарного провайдера	Повний контроль доступу на місцевому рівні

Таким чином, сучасні хмарні сховища мають широкий спектр можливостей для зберігання, доступу та управління даними. Amazon S3, Microsoft Azure Blob Storage і Google Cloud Storage – найбільш популярні хмарні платформи, що надають можливість зберігання та обробку даних для підприємств різних розмірів та галузей. Ці платформи пропонують різні класи зберігання, що дозволяє оптимізувати витрати залежно від частоти доступу до даних та їхньої важливості.

Amazon S3 забезпечує гнучке управління завдяки класам Standard, Intelligent-Tiering та Glacier, які відповідають різним потребам – від швидкого доступу до архівного зберігання. Microsoft Azure Blob Storage дозволяє зберігати великі обсяги неструктурованих даних з можливістю інтеграції в існуючі системи через API та інструменти аналітики. Google Cloud Storage відзначається високою продуктивністю, глобальною доступністю та підтримкою автоматизації через політики життєвого циклу даних.

1.4. Методологічні аспекти переносу бінарних файлів у хмарні сховища

Перенесення бінарних файлів у хмарні сховища, такі як Amazon S3, вимагає чіткої методології та відповідних інструментів для забезпечення надійності, безпеки та продуктивності. Основні методологічні аспекти включають вибір методів переносу, використання інтеграційних технологій і врахування викликів безпеки. Існує декілька методів переносу бінарних файлів із корпоративних баз даних у хмарні сховища. Вибір методу залежить від обсягу даних, частоти їх оновлення, доступності та вимог до швидкості [21-27] (таблиця 1.6).

Таблиця 1.6 – Основні методи та інструменти для переносу даних

Метод переносу	Опис	Інструменти
Резервне копіювання та відновлення	Створення резервних копій даних та завантаження їх у хмару для подальшого використання.	AWS Backup, AWS Storage Gateway
Автоматична реплікація	Автоматичне віддзеркалення даних у хмару для безперервного збереження та доступу.	AWS DataSync, Amazon S3 Replication
Пакетний експорт і імпорт	Експорт файлів з бази даних у вигляді архівів або пакетів для їх подальшого завантаження.	AWS Snowball, AWS CLI
Прямий переніс за допомогою API	Програмне завантаження файлів у хмару безпосередньо з баз даних.	AWS SDK, Amazon S3 API
Гібридне зберігання	Поєднання локального та хмарного зберігання для резервування даних з обмеженим доступом у хмарі.	AWS Storage Gateway

Amazon Web Services (AWS) пропонує різноманітні засоби для інтеграції з Amazon S3, що дозволяє автоматизувати завантаження, керування та доступ до бінарних файлів у хмарі.

Комплекс інтеграційних інструментів AWS для роботи з Amazon S3 представлено на діаграмі, що відображає взаємозв'язок між інструментами та сервісом Amazon S3 (рисунок 1.6).

Рисунок 1.6 – Інтеграційні інструменти AWS для роботи з Amazon S3

AWS SDK (Software Development Kit) надає можливість програмно взаємодіяти з Amazon S3 через різні мови програмування, такі як Python, Java, та Node.js. Це дозволяє автоматизувати завантаження файлів, організацію bucket-ів і контроль доступу [28].

Amazon S3 API забезпечує прямий доступ до функцій S3, дозволяючи виконувати операції завантаження, оновлення та видалення файлів. Це особливо корисно для автоматизації процесів та інтеграції з іншими корпоративними системами[29, 30] .

AWS DataSync автоматизує перенесення великих обсягів даних з локальних серверів у Amazon S3, забезпечуючи при цьому шифрування та моніторинг процесу [31].

AWS CLI (Command Line Interface) – інструмент AWS, що забезпечує управління bucket-ами та виконання команд завантаження безпосередньо через командний рядок [32].

Як показано вище, AWS CLI підключається безпосередньо до Amazon S3 для завантаження, видалення або керування даними. AWS SDK здійснює інтеграцію з API Amazon S3, що дозволяє програмно взаємодіяти з S3 через різні мови програмування. DataSync забезпечує автоматичну реплікацію та моніторинг переносу даних між локальними сховищами та Amazon S3. API виступає посередником між SDK та S3 для виконання запитів і операцій з даними.

Безпека є критично важливим аспектом при перенесенні бінарних файлів у хмару. Хоча Amazon S3 забезпечує потужні інструменти для захисту даних, важливо дотримуватися існуючих практик, що мінімізують ризики витоку або несанкціонованого доступу.

До ключових практик належать використання шифрування даних як на етапі передачі (TLS/SSL), так і на етапі зберігання (SSE – Server-Side Encryption), а також керування доступом через IAM (Identity and Access Management), де ролі й дозволи налаштовуються відповідно до принципу найменших привілеїв. Крім цього, важливо впроваджувати політики моніторингу та сповіщення, використовуючи AWS CloudWatch [33] та AWS CloudTrail [34] для відстеження

операцій і своєчасного виявлення потенційних загроз. Налаштування політик життєвого циклу дозволяє автоматизувати процес видалення або архівації старих даних, що знижує ризики накопичення застарілої інформації (таблиця 1.7).

Таблиця 1.7 – Основні загрози безпеки та рішення щодо їх врахування

Загроза безпеки	Опис	Рішення
Неавторизований доступ	Ризик доступу до файлів особами, які не мають дозволу	Використання політик IAM, шифрування, S3 Access Control Lists
Перехоплення даних при передачі	Ризик перехоплення даних під час передачі їх з локального середовища в хмару	SSL/TLS шифрування для безпечної передачі даних
Вразливість об'єктів даних	Відсутність шифрування даних на рівні bucket-а може спричинити витоки	Використання Amazon S3 Server-Side Encryption (SSE)
Відсутність логування	Неможливість відстежити доступ до файлів	Включення Amazon CloudTrail для логування доступу
Незахищене резервне копіювання	Ризик втрати даних у випадку відсутності належного резервного копіювання	Налаштування автоматичного резервування з AWS Backup

На діаграмі (рисунок 1.7) показані основні загрози безпеці та їх рівні під час перенесення бінарних файлів у хмару [35-40].

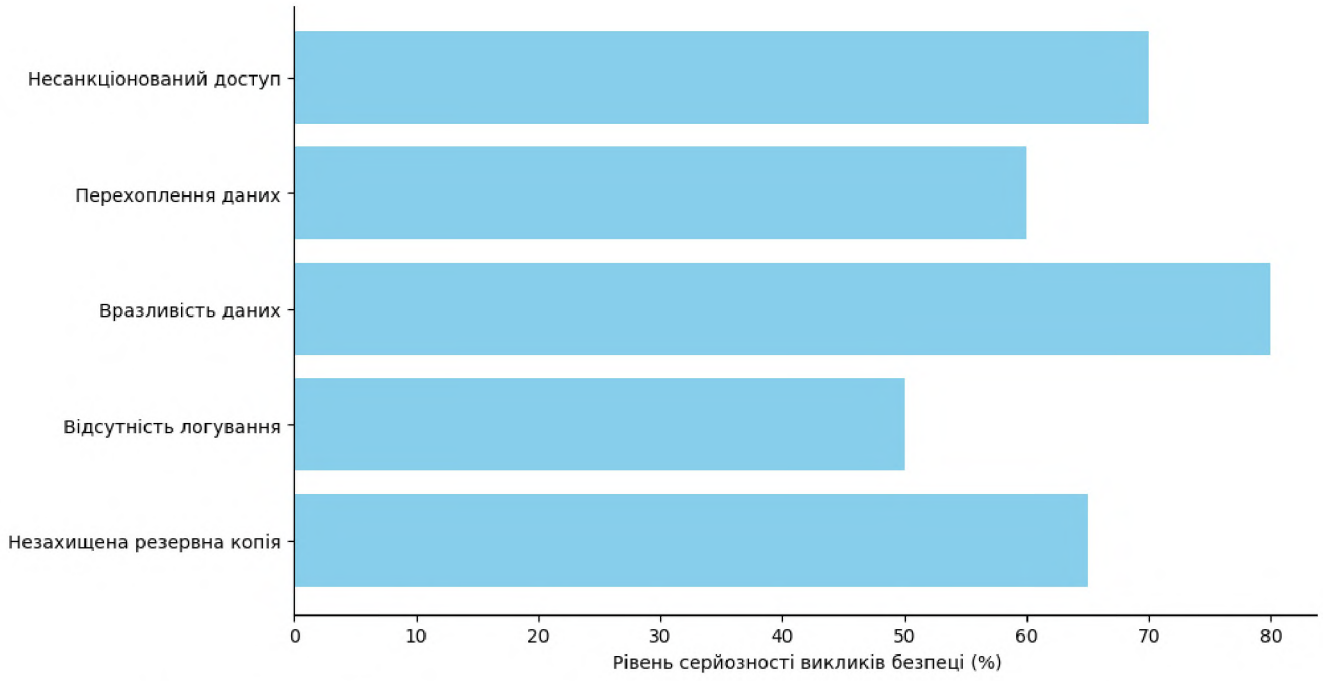

Рисунок 1.7 – Рівні загроз безпеці даних при перенесенні у хмару

Найвищий рівень ризику має вразливість об'єктів даних (80%), що вимагає обов'язкового шифрування та контролю доступу. Перехоплення даних і неавторизований доступ також представляють високі ризики, які можна мінімізувати за допомогою SSL/TLS шифрування та політик IAM.

Таким чином, методологія перенесення бінарних файлів у хмарні сховища включає вибір відповідних методів і інструментів для забезпечення продуктивності та безпеки. Інтеграційні технології AWS, такі як SDK, API, DataSync, і CLI, дозволяють автоматизувати процеси та захищати дані. Окрім цього, важливо враховувати загрози безпеки, впроваджуючи шифрування, логування та налаштування політик доступу для захисту даних у хмарі.

Висновки до розділу 1

Бінарні файли є важливою частиною корпоративних систем для зберігання мультимедіа, документів та спеціалізованих даних. Їх ефективне зберігання потребує оптимізації для зниження витрат, підвищення доступності та спрощення управління. Традиційні бази даних мають низку проблєм із зберіганням бінарних файлів. Використання хмарних сховищ, таких як Amazon S3, дозволяє значно підвищити ефективність за рахунок масштабованості, доступності та зменшення витрат. Сучасні хмарні сховища, зокрема Amazon S3, Microsoft Azure Blob Storage і Google Cloud Storage, забезпечують зручне зберігання та управління даними для підприємств різних масштабів. Порівняння з локальними серверами демонструє, що хмарні рішення є більш ефективними з точки зору безпеки, витрат та масштабування. Методологія перенесення бінарних файлів у хмару включає вибір інструментів для продуктивності та безпеки. Інтеграційні технології AWS (SDK, API, DataSync, CLI) дозволяють автоматизувати процеси, а заходи, такі як шифрування, логування та налаштування політик доступу, забезпечують надійний захист даних у хмарному середовищі.

РОЗДІЛ 2
МЕТОДИ ПЕРЕНОСУ БІНАРНИХ ФАЙЛІВ У ХМАРНІ СХОВИЩА

2.1. Характеристики корпоративних баз даних

Для даного дослідження обрано систему управління базами даних PostgreSQL як одну з популярних і гнучких баз даних, що широко використовується у корпоративному середовищі. PostgreSQL підтримує складні реляційні зв'язки і забезпечує високу продуктивність при зберіганні бінарних файлів за допомогою спеціалізованих типів даних. Розглянемо основні особливості PostgreSQL у контексті зберігання бінарних даних та інтеграції з хмарними сервісами.

PostgreSQL – реляційна СУБД з відкритим кодом, що має надійні механізми для зберігання структурованих даних. PostgreSQL надає гнучкі можливості для зберігання як структурованих, так і неструктурованих даних, включаючи бінарні об'єкти. Забезпечує підтримку транзакційності, ACID-сумісність і має велику кількість вбудованих інструментів для управління даними. Основні характеристики PostgreSQL представлені у таблиці 2.1.

Таблиця 2.1 – Характеристики СУБД PostgreSQL

Характеристика	Опис
Тип СУБД	Реляційна, з підтримкою ACID
Ліцензія	Відкрита ліцензія PostgreSQL (BSD)
Основні функції	Реляційні запити, транзакційність, підтримка PL/pgSQL
Вбудовані типи даних	Підтримка числових, текстових, JSON та бінарних типів
Сумісність з хмарою	Легка інтеграція з AWS, Azure, Google Cloud
Бінарні типи даних	BYTEA, Large Objects
Підтримка індексації	Повнотекстовий пошук, B-дерево, GiST, GIN
Придатність для великих даних	Може працювати з великими наборами даних при оптимізації

СУБД PostgreSQL використовує декілька підходів для зберігання бінарних файлів. Основні з них використовують типи BYTEA та Large Objects (LOB). Кожен з них підходить для зберігання бінарної інформації, але має свої

особливості в залежності від розміру файлів та вимог до швидкості доступу (таблиця 2.2).

Таблиця 2.2 – Методи зберігання бінарних даних у СУБД PostgreSQL

Метод зберігання	Опис	Рекомендоване використання
BYTEA	Зберігає бінарні файли безпосередньо в таблиці у вигляді рядків. Підходить для файлів невеликого розміру (до декількох МБ)	Файли невеликого розміру, швидкий доступ
Large Objects	Зберігає великі файли поза основною таблицею. PostgreSQL надає спеціальні функції для роботи з LOB, такі як lo_import і lo_export	Великі бінарні файли, які рідко використовуються

PostgreSQL легко інтегрується з популярними хмарними платформами, такими як Amazon Web Services (AWS), Microsoft Azure і Google Cloud Platform. Така інтеграція для зберігання бінарних файлів дозволяє знизити навантаження на локальну базу даних і підвищити доступність даних.

Діаграма, на рисунку 2.1 демонструє зв'язок між PostgreSQL і Amazon S3, показуючи основні методи інтеграції для забезпечення збереження та доступності бінарних файлів у хмарному сховищі.

Рисунок 2.1 – Інтеграція СУБД PostgreSQL з Amazon S3

Як показано на рисунку 2.1, СУБД PostgreSQL підключається до сервісу Amazon S3 через три основні процеси:

– резервне копіювання – регулярне зберігання копій даних для захисту від втрат;

– реплікація – синхронізація зі сховищами на резервних серверах для безперервного доступу у разі збою;

– перенос даних через API – програмна передача даних між базою даних та сховищем.

Головні аспекти інтеграції PostgreSQL з хмарними сервісами:

– PostgreSQL можна налаштувати для автоматичного зберігання резервних копій у Amazon S3 або іншому хмарному сховищі, забезпечивши цим автоматизацію резервного копіювання;

– PostgreSQL може використовувати AWS DataSync або AWS DMS (Database Migration Service) для ефективного переносу великих обсягів бінарних файлів до Amazon S3;

– хмарні сервіси підтримують механізм реплікації, що дозволяє забезпечити безперервний доступ до даних у випадку відмови локальних серверів.

Типовий сценарій інтеграції PostgreSQL з Amazon S3 для зберігання бінарних файлів є наступним:

a) експорт бінарних файлів – вибрані бінарні файли (зображення, документи тощо) екстрагуються з PostgreSQL за допомогою типу Large Object або функції BYTEA;

b) завантаження файлів у Amazon S3 – файли, експортовані з PostgreSQL, завантажуються до Amazon S3 за допомогою AWS CLI, AWS SDK або інших інструментів автоматизації (наприклад, DataSync);

c) створення резервної копії даних – для забезпечення надійності та доступності даних в Amazon S3 налаштовується політика автоматичного резервування та архівування.

Таким чином, використання PostgreSQL для зберігання бінарних файлів є ефективним рішенням для корпоративних середовищ, які потребують надійного управління даними. Інтеграція з хмарними сервісами, зокрема Amazon S3, дозволяє знизити навантаження на локальну базу даних, автоматизувати процес резервування та зберігання великих файлів, підвищуючи загальну продуктивність системи.

2.2 Альтернативні методи переносу даних

Існує декілька альтернативних методів для переносу даних із корпоративних баз даних у різні хмарні сховища. Основними з них є: резервне копіювання та відновлення, автоматична реплікація, пакетний експорт та імпорт, перенос даних через API та гібридне зберігання. Кожен метод має свої переваги та підходить для різних обсягів даних, частоти доступу та вимог до безпеки.

Резервне копіювання та відновлення передбачає створення резервних копій даних у локальній системі з подальшим завантаженням цих копій у хмару. Резервне копіювання зазвичай проводиться періодично, забезпечуючи захист від втрат. Підходить для даних, які не потребують частого оновлення. Використовує інструменти, такі як AWS Backup, Azure Backup та Google Cloud Backup.

Автоматична реплікація дозволяє автоматично синхронізувати дані між локальними серверами та хмарою в режимі реального часу або з мінімальними затримками. Цей метод корисний для забезпечення безперервного доступу до актуальних даних. Підходить для ґритично важливих даних, що постійно оновлюються. Спеціальні інструменти: AWS DataSync, Amazon S3 Replication [41], Google Cloud Storage Transfer Service [42].

Пакетний експорт та імпорт – дані експортуються з локальної бази даних у вигляді великих пакетів або архівів, які згодом завантажуються у хмарне сховище. Вважається, що це досить простий метод, особливо для початкового перенесення великих обсягів даних. Підходить для масового переносу або міграції історичних даних. Інструменти, що використовуються: AWS Snowball [43], Azure Import/Export Service [44], Google Cloud Storage Transfer Appliance [42].

Перенос даних через API. За допомогою API дані передаються безпосередньо з бази даних у хмарне сховище. Цей метод дозволяє налаштувати автоматизацію та точкове завантаження файлів, підтримуючи швидкий доступ до даних. Підходить для даних, до яких необхідний швидкий доступ, та інтеграції з програмами. Інструменти: AWS SDK, Amazon S3 API, Google Cloud API [45], Azure Blob Storage API [46, 47].

Гібридне зберігання поєднує локальне та хмарне зберігання даних: активні дані зберігаються локально, архівні – у хмарі. Гібридне зберігання доцільне для великих організацій, які хочуть знизити навантаження на локальні ресурси, зберігаючи при цьому швидкий доступ до критичних даних. Для організацій, які мають обмеження щодо зберігання чутливих даних у хмарі. Інструменти: AWS Storage Gateway [48], Azure StorSimple [49], Google Cloud Filestore [50].

Порівняння альтернативних методів переносу даних із корпоративних баз даних у хмарні сховища представлене у таблиці 2.3.

Таблиця 2.3 – Порівняння альтернативних методів переносу даних

Метод	Переваги	Недоліки	Інструменти
Резервне копіювання	Захист від втрат, простота реалізації	Потребує часу для відновлення даних	AWS Backup, Azure Backup
Автоматична реплікація	Актуальні дані в режимі реального часу	Потребує високої пропускної здатності	AWS DataSync, S3 Replication
Пакетний експорт та імпорт	Масовий перенос, ефективність для великих даних	Обмежена частота оновлення	AWS Snowball, Google Transfer Appliance
Перенос через API	Гнучкість, автоматизація, підтримка додатків	Може бути складним для великих обсягів	AWS SDK, Amazon S3 API
Гібридне зберігання	Зменшення навантаження на локальне сховище	Обмежена інтеграція між локальним та хмарним	AWS Storage Gateway, Azure StorSimple

Таким чином, кожен з методів переносу даних з локального серверу у хмару має свої переваги та підходить для різних сценаріїв використання. Вибір методу переносу даних залежить від обсягів інформації, частоти доступу, а також специфічних вимог до безпеки та відновлення даних у випадку збоїв.

2.3. Аналіз методів і технологій переносу бінарних файлів

Як було показано вище, перенесення бінарних файлів із корпоративних баз даних у хмарні сховища може здійснюватися кількома методами, серед яких основними є резервне копіювання, реплікація та автоматичний експорт. Кожен із

цих методів має свої особливості та підходить для різних типів сценаріїв залежно від обсягів даних, частоти доступу та вимог до безпеки. Переваги і недоліки основних методів переносу даних представлені у таблиці 2.4.

Таблиця 2.4 – Переваги і недоліки основних методів переносу даних

Метод переносу	Опис	Переваги	Недоліки
Резервне копіювання	Створення копій даних і завантаження їх у хмарне сховище. Використовується для захисту від втрат і відновлення у разі збою.	Простота реалізації, мінімальні затрати на підтримку	Не забезпечує актуальності даних в режимі реального часу
Реплікація	Автоматичне копіювання даних у хмару в режимі реального часу або з невеликими затримками, що дозволяє синхронізувати локальні дані з хмарою.	Актуальні дані в будь-який момент, висока надійність	Високі вимоги до пропускної здатності та з'єднання
Автоматичний експорт	Регулярне експортування даних з бази в пакетах для їх подальшого завантаження у хмарне сховище.	Ефективний для великих обсягів даних, зручний в налаштуванні	Обмежена частота оновлення, затримки в обробці великих даних

На практиці, вибір методу переносу даних залежить від специфічних вимог до частоти доступу, обсягів даних та безпеки. Нижче наведено методи, які можна вважати найефективнішими для забезпечення безперервного та безпечного переносу даних:

– реплікація – оптимальний метод для безперервного переносу даних у режимі реального часу. Використання інструментів, таких як AWS DataSync або Amazon S3 Replication, дозволяє автоматично синхронізувати локальні файли з хмарою без затримок. Це забезпечує доступ до актуальних даних і підходить для критичних систем, які потребують постійного доступу до оновленої інформації;

– резервне копіювання з регулярним оновленням – підходить для сценаріїв, де дані потребують зберігання для архівних або відновлювальних цілей. Для цього часто використовуються такі інструменти, як AWS Backup або Azure Backup, які забезпечують автоматичне резервування з налаштуванням графіка;

– автоматичний експорт – ефективний для великих обсягів даних, які можуть оновлюватися в періодичному режимі (наприклад, щодня або щотижня).

Використання AWS Snowball або аналогічних сервісів дозволяє знизити навантаження на мережу, а дані завантажуються у хмару після обробки.

Порівняння ефективності методів для безперервного та безпечного переносу представлене у таблиці 2.5.

Таблиця 2.5 – Порівняння ефективності методів для безперервного та безпечного переносу

Метод	Ефективність для безперервного доступу	Безпека даних	Відповідний інструмент
Резервне копіювання	Низька	Висока	AWS Backup, Azure Backup
Реплікація	Висока	Висока, залежить від шифрування	AWS DataSync, Amazon S3 Replication
Автоматичний експорт	Середня	Помірна, вимагає додаткових налаштувань	AWS Snowball, Azure Data Box

Основні методи переносу бінарних файлів із локальних баз даних у хмарне сховище ілюструє діаграма на рисунку 2.2.

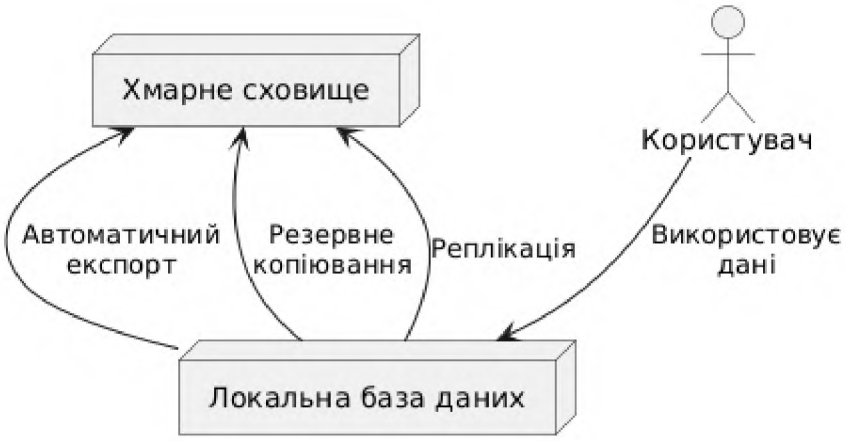

Рисунок 2.2 – Основні методи переносу бінарних файлів із локальних баз даних у хмару

Діаграма на рисунку 2.2 схематично представляє логіку використання трьох основних методів, що використовуються на практиці для перенесення бінарних файлів із локальних баз даних у хмару:

– резервне копіювання – локальна база даних створює резервні копії даних, які періодично передаються у хмарне сховище для збереження;

– реплікація – дані автоматично синхронізуються між локальною базою та хмарою, забезпечуючи актуальність у реальному часі;

– автоматичний експорт – локальна база періодично експортує дані у пакети, які завантажуються у хмару для подальшого зберігання та обробки.

Таким чином, резервне копіювання є надійним методом, що застосовується для архівування і захисту даних від втрат. Для безперервного і безпечного переносу даних найкраще підходить реплікація, яка забезпечує актуальність даних у режимі реального часу. Автоматичний експорт є оптимальним для переносу великих обсягів даних з періодичним оновленням, оскільки дозволяє уникнути перенавантаження мережі. Вибір методу залежить від вимог до доступу, безпеки та обсягів

2.4. Перенесення бінарних файлів із корпоративної бази даних у Amazon S3

Для перенесення бінарних файлів із корпоративної бази даних у хмарне сховище Amazon S3 потрібно забезпечити баланс між продуктивністю, безпекою та економічною ефективністю.

Цей баланс досягається завдяки ретельному плануванню структури зберігання даних, вибору відповідних класів зберігання та налаштуванню політик життєвого циклу файлів. Для продуктивності важливо використовувати AWS SDK або CLI для автоматизації процесів завантаження й обробки даних, а також налаштувати паралельну передачу файлів для пришвидшення міграції великих обсягів даних. З точки зору безпеки, необхідно впроваджувати шифрування на стороні сервера (SSE) та контроль доступу за допомогою IAM-політик, щоб обмежити права користувачів відповідно до принципу найменших привілеїв.

Моніторинг операцій через AWS CloudWatch і CloudTrail допомагає оперативно виявляти помилки та потенційні загрози, забезпечуючи надійність системи.

Основними умовами перенесення є надійне з'єднання та автоматизовані процеси, які забезпечують безперервність доступу, і ефективне управління витратами на зберігання (таблиця 2.6).

Таблиця 2.6 – Умови оптимізації перенесення бінарних файлів із корпоративної бази даних у Amazon S3

Умова	Опис	Інструменти та рішення
Надійне мережеве з'єднання	Забезпечує стабільність і швидкість передачі даних, мінімізує ризик переривань	Використання стабільних каналів зв'язку
Безпека даних	Шифрування даних при передачі та зберіганні, контроль доступу	SSL/TLS, Amazon S3 Server-Side Encryption (SSE)
Автоматизація процесу переносу	Налаштування автоматичного експорту, реплікації або резервного копіювання	AWS DataSync, AWS CLI, Amazon S3 API
Управління витратами	Використання різних класів зберігання для оптимізації витрат	S3 Standard, S3 Glacier, S3 Intelligent-Tiering
Логування та моніторинг	Відстеження доступу та змін для забезпечення аудиту й аналізу ефективності	Amazon CloudTrail, Amazon CloudWatch

Для оцінки ефективності існуючих підходів щодо перенесення даних до Amazon S3 розглянемо їх переваги і недоліки, представлені у таблиці 2.7.

Таблиця 2.7 – Переваги і недоліки існуючих підходів перенесення даних до Amazon S3

Підхід	Переваги	Недоліки
Резервне копіювання	Простота реалізації, надійність зберігання даних	Відсутність безперервного доступу до оновлених даних
Реплікація	Актуальність даних у режимі реального часу, висока доступність	Високі вимоги до пропускної здатності мережі
Автоматичний експорт	Ефективний для періодичного перенесення великих обсягів даних, мінімальне навантаження на мережу	Не забезпечує миттєвої актуальності даних
Використання API та SDK	Гнучкість у налаштуванні, інтеграція з іншими додатками	Може бути складним у реалізації для великих обсягів
Гібридне зберігання	Зниження навантаження на локальні ресурси, підтримка локального доступу до критичних даних	Вимагає додаткової синхронізації між локальними та хмарними сервісами

Оптимальний процес перенесення файлів із корпоративної бази даних у Amazon S3 представлений на рисунку 2.3.

Рисунок 2.3 – Оптимальний процес перенесення файлів із корпоративної бази даних у Amazon S3

На діаграмі (рисунок 2.3) відображене наступне:

– адміністратор налаштовує параметри переносу в корпоративній базі даних;

– корпоративна БД передає файли для переносу через інтеграційні інструменти, такі як AWS CLI або Amazon S3 API;

– інтеграційні інструменти здійснюють автоматичний експорт або резервне копіювання і надсилають файли до Amazon S3;

– Amazon S3 обробляє дані та, за допомогою CloudWatch, надає адміністраторам логування й сповіщення про завершення процесу.

На рисунку 2.4. представлена діаграма, що відображає ієрархію умов для успішного та ефективного переносу бінарних файлів у хмарні сховища.

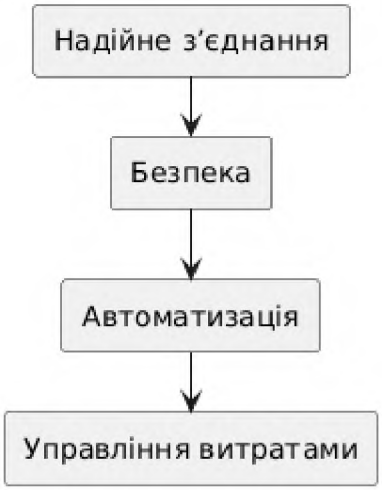

Рисунок 2.4 – Умови успішного переносу даних

Отже, надійне з'єднання є основною умовою, яка забезпечує стабільність передачі даних. Безпека має бути гарантована на кожному етапі для захисту даних під час переносу. Автоматизація процесу допомагає знизити витрати часу та ресурсів на перенесення. Управління витратами дозволяє оптимізувати витрати на зберігання і забезпечує економічну ефективність.

Порівняння основних методів переносу бінарних файлів із корпоративної бази даних дозволяє вибрати оптимальний метод залежно від специфічних вимог до переносу даних. Ці методи включають резервне копіювання, реплікацію та автоматичний експорт даних, кожен з яких має свої переваги та обмеження. Резервне копіювання є найбільш надійним методом для створення копій бінарних файлів перед їх перенесенням, що забезпечує захист від втрати даних, проте потребує додаткового часу та ресурсів. Реплікація дозволяє синхронізувати дані в реальному часі між корпоративною базою даних і хмарним сховищем, що забезпечує високу актуальність інформації. Однак цей метод вимагає значної продуктивності системи та ефективного управління потоком даних. Автоматичний експорт через інструменти, такі як AWS DataSync, AWS CLI або спеціалізовані скрипти, є найефективнішим методом для перенесення великих обсягів бінарних файлів завдяки гнучкості, можливості автоматизації та налаштуванню параметрів продуктивності. Порівняння основних методів

переносу бінарних файлів із корпоративної бази даних у Amazon S3 за важливими умовами та характеристиками кожного методу представлене у таблиці 2.8.

Таблиця 2.8 – Порівняння основних методів переносу бінарних файлів із корпоративної бази даних у Amazon S3

Метод переносу	Надійне з'єднання	Безпека даних	Автоматизація процесу	Управління витратами	Підходить для
Резервне копіювання	Потрібне стабільне з'єднання	Високий рівень захисту даних, завдяки шифруванню	Мінімальна, залежить від інструмента	Контроль витрат завдяки налаштуванню частоти копіювань	Архівування та відновлення у разі збоїв
Реплікація	Високі вимоги до пропускної здатності	Захист даних через реплікацію в реальному часі	Висока, автоматичне оновлення	Збільшені витрати через постійне зберігання актуальних даних	Безперервний доступ та актуальність даних у режимі реального часу
Автоматичний експорт	Потрібне стабільне з'єднання	Забезпечення захисту даних через SSL/TLS	Середня, налаштовується періодичність	Оптимізація витрат за рахунок обмеженої частоти оновлень	Масове перенесення великих обсягів даних у хмару
API та SDK	Вимагає стабільного з'єднання	Контроль доступу через політики IAM, шифрування	Висока, програмна автоматизація	Гнучке управління витратами за допомогою вибору класів зберігання	Інтеграція з іншими системами та додатками
Гібридне зберігання	Вимагає стабільного з'єднання	Поєднання шифрування для локальних та хмарних даних	Мінімальна	Управління витратами через комбіноване зберігання локальних і хмарних даних	Ситуації, де потрібний швидкий локальний доступ

Таким чином, вибір методу залежить від обсягу даних, вимог до швидкості переносу, а також необхідності забезпечення цілісності та безпеки файлів. Оптимальним підходом є комбінування кількох методів для досягнення балансу

між продуктивністю, надійністю та економічною ефективністю процесу перенесення даних.

Результати порівняння методів переносу даних у хмару за важливими умовами та характеристиками стосовно кожного методу представлено на рисунку 2.5.

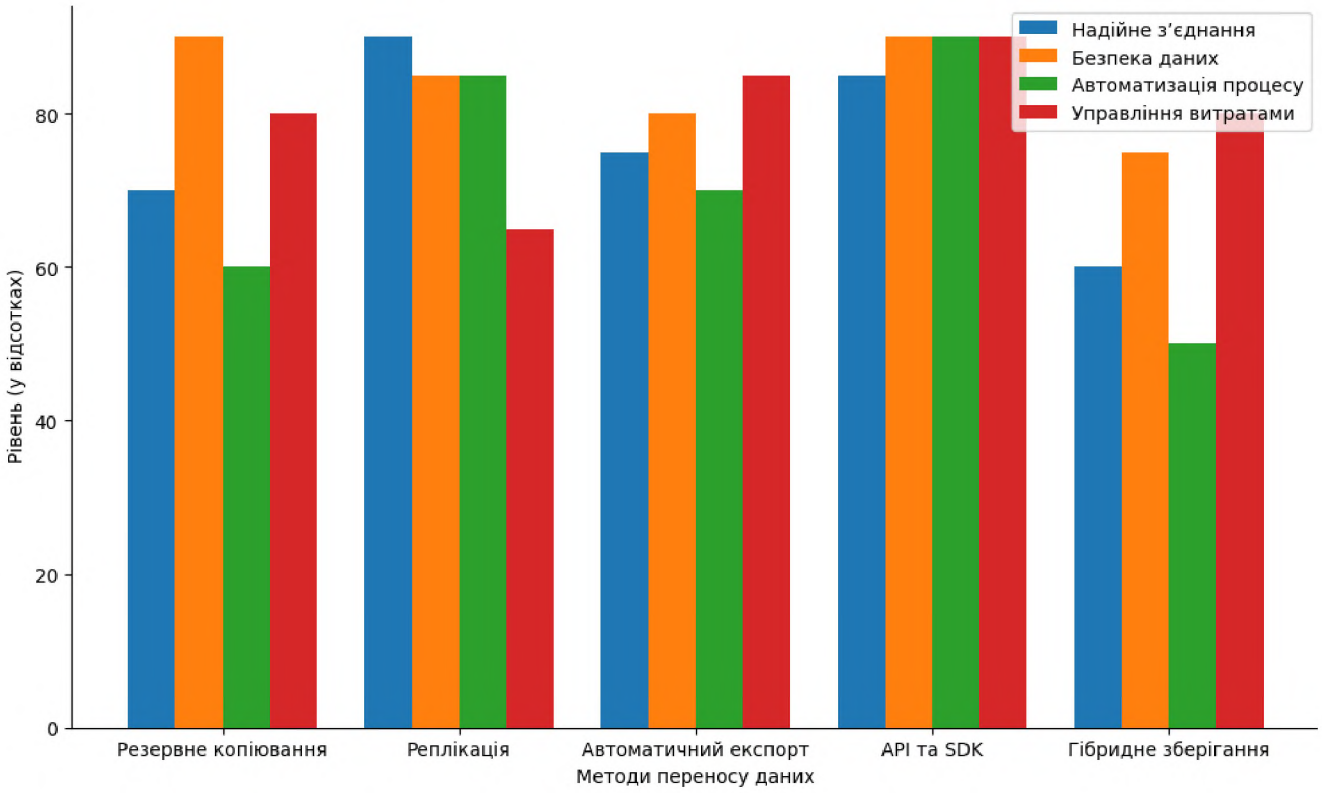

Рисунок 2.5 – Порівняння методів переносу даних у хмару

Графік на рисунку 2.5 показує, що:

– реплікація має найвищий рівень надійності з'єднання та автоматизації, що робить її оптимальною для безперервного доступу до даних;

– API та SDK забезпечують високі показники автоматизації та управління витратами, завдяки гнучким можливостям інтеграції;

– резервне копіювання має високий рівень безпеки та управління витратами, підходячи для архівного зберігання;

– автоматичний експорт є збалансованим рішенням для періодичного переносу великих обсягів даних.

Отже, реплікація забезпечує безперервний доступ до актуальних даних, але потребує високих витрат на зберігання і пропускну здатність мережі. Резервне копіювання є оптимальним для архівування даних, зберігаючи високий рівень безпеки. Автоматичний експорт та API підходять для великих обсягів даних, дозволяючи ефективно контролювати витрати, зберігаючи доступність і безпеку.

Висновки до розділу 2

Використання PostgreSQL для зберігання бінарних файлів є надійним рішенням для корпоративних середовищ. Інтеграція з Amazon S3 знижує навантаження на локальну базу даних, автоматизує резервування та зберігання великих файлів, підвищуючи продуктивність системи.

Існують різні методи переносу даних: резервне копіювання, реплікація та автоматичний експорт, кожен із яких має переваги залежно від обсягу, частоти доступу та вимог до безпеки.

Для оптимального перенесення даних в Amazon S3 важливо забезпечити надійне з'єднання, автоматизацію та ефективне управління витратами. Реплікація підходить для безперервного переносу в режимі реального часу, але вимагає значних ресурсів. Резервне копіювання є надійним для архівування і захисту даних. Автоматичний експорт та API оптимальні для великих обсягів даних завдяки контролю витрат та збереженню доступності.

Вибір методу залежить від вимог: реплікація забезпечує актуальність даних, резервне копіювання – безпеку, а автоматичний експорт – ефективний баланс між продуктивністю та витратами.

РОЗДІЛ 3
РОЗРОБКА МЕТОДОЛОГІЇ ПЕРЕНЕСЕННЯ БІНАРНИХ ФАЙЛІВ У СХОВИЩЕ AMAZON S3

3.1. Проєктування архітектури перенесення файлів із корпоративних баз даних у Amazon S3

Проєктування архітектури для перенесення бінарних файлів із корпоративної бази даних у Amazon S3 має передбачити надійну та гнучку структуру, яка забезпечує автоматизацію процесів, масштабованість для обробки великих обсягів даних та відмовостійкість системи в разі збоїв. Важливо враховувати особливості інфраструктури, зокрема оптимізацію з'єднань, захист даних та ефективний контроль доступу для забезпечення безпеки та продуктивності.

Архітектура перенесення даних використовує такі компоненти:

– інтеграційний шар для взаємодії між базою даних та Amazon S3 використовується Amazon S3 API або SDK, що дозволяє завантажувати та отримувати файли з S3;

– процес автоматизації (використовує AWS Lambda або інші серверлесс-функції, які забезпечують автоматизацію процесу переносу). Наприклад, коли новий файл додається до бази, функція автоматично завантажує його в S3;

– моніторинг та управління (за допомогою Amazon CloudWatch, що забезпечує моніторинг статусу переносу та виявлення збоїв, тоді як Amazon CloudTrail здійснює логування всіх операцій);

– резервне копіювання та відновлення для захисту даних і відновлення у випадку збою (налаштовується резервне копіювання через AWS Backup або відповідні інструменти на рівні бази даних).

Компоненти архітектури та інструменти перенесення файлів з корпоративної бази даних у Amazon S3 представлені у таблиці 3.1.

Таблиця 3.1 – Компоненти архітектури переносу даних у Amazon S3

Компонент	Опис	Інструменти
Інтеграційний шар	API або SDK для передачі даних із бази даних у S3	Amazon S3 API, AWS SDK
Процес автоматизації	Виконання автоматичних завдань для переносу нових файлів	AWS Lambda, AWS Step Functions
Моніторинг	Відстеження статусу переносу та виявлення проблєм	Amazon CloudWatch, CloudTrail
Резервне копіювання	Створення копій даних для захисту від втрат	AWS Backup

Для забезпечення стабільної роботи системи зберігання та перенесення даних необхідно також передбачити її масштабованість та відмовостійкість. Amazon S3 автоматично масштабує зберігання залежно від обсягу даних.

Архітектура переносу даних із корпоративної бази даних у Amazon S3 представлена на діаграмі (рисунок 3.1).

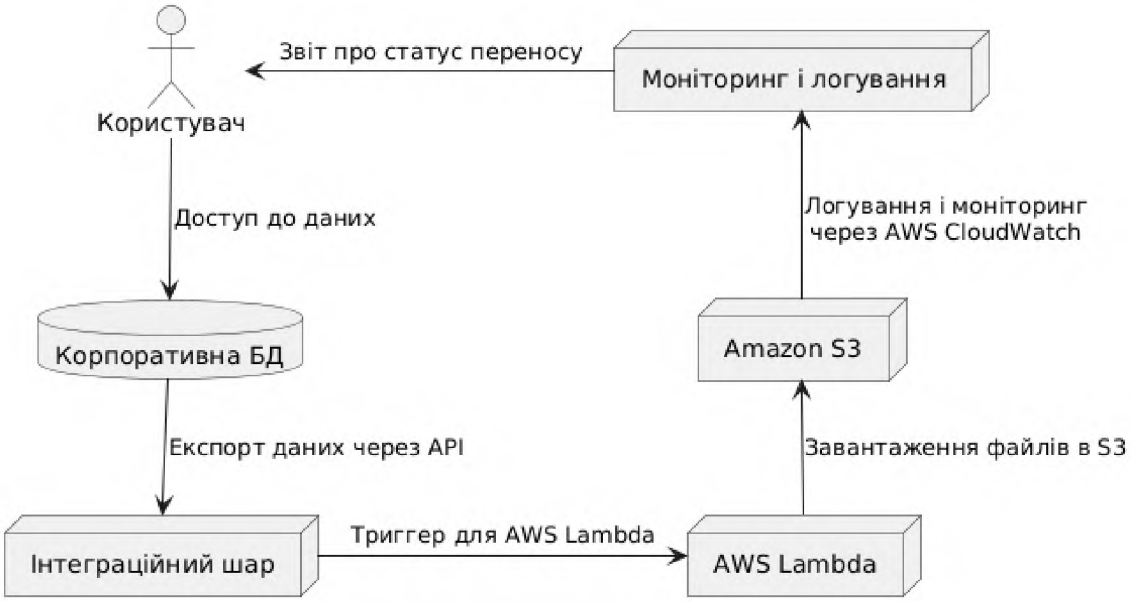

Рисунок 3.1 – Архітектура переносу даних із корпоративної бази у Amazon S3

Таким чином, користувач отримує доступ до даних у корпоративній базі; корпоративна БД використовує інтеграційний шар для перенесення файлів до Amazon S3 через API; AWS Lambda обробляє перенесення файлів у Amazon S3, автоматизуючи процес; моніторинг і логування здійснюється через CloudWatch і CloudTrail для відстеження операцій та виявлення збоїв.

Використання AWS Lambda забезпечує автоматичне виконання завдань переносу, що дозволяє обробляти більші обсяги даних. За рахунок реплікації Amazon S3 гарантує високу доступність і захист даних. Налаштування резервного копіювання та інтеграції з Amazon CloudWatch дозволяє швидко реагувати на збої, а CloudTrail забезпечує аудит і відстеження операцій.

Отже, запропонована архітектура для перенесення бінарних файлів у Amazon S3 включає автоматизований інтеграційний шар, процес моніторингу та резервування, що забезпечує стабільність, масштабованість і відмовостійкість системи. Цей підхід дозволяє ефективно керувати великими обсягами даних, забезпечуючи надійність та безперервний доступ до них у хмарному сховищі.

3.2. Розроблення алгоритму переносу даних у хмарне сховище

3.2.1 Структура алгоритму переносу даних

Розробка алгоритму переносу даних у хмарне сховище Amazon S3 включає три основні етапи: експорт файлів із корпоративної бази даних, завантаження файлів у Amazon S3 та перевірка коректності перенесення. На кожному етапі важливо впроваджувати заходи безпеки, щоб забезпечити захист даних і знизити ризики під час передачі.

Основні етапи переносу даних відображені у таблиці 3.1.

Таблиця 3.1 – Основні етапи переносу даних

Етап	Опис кроку	Заходи безпеки
1. Експорт файлів із бази даних	Файли експортуються з корпоративної бази даних за допомогою запитів SQL або спеціалізованих функцій.	Використання прав доступу, що обмежують доступ до даних
2. Завантаження файлів в Amazon S3	Експортовані файли передаються в Amazon S3 за допомогою AWS CLI, SDK або API.	Шифрування даних при передачі через SSL/TLS, IAM політики
3. Перевірка коректності перенесення	Перевірка успішності переносу за допомогою хеш-функцій (наприклад, MD5), щоб упевнитися, що файли збережені без помилок.	Зберігання журналів для аудиту, верифікація прав доступу

Візуалізація алгоритму переносу даних у Amazon S3 представлена на рисунку 3.2.

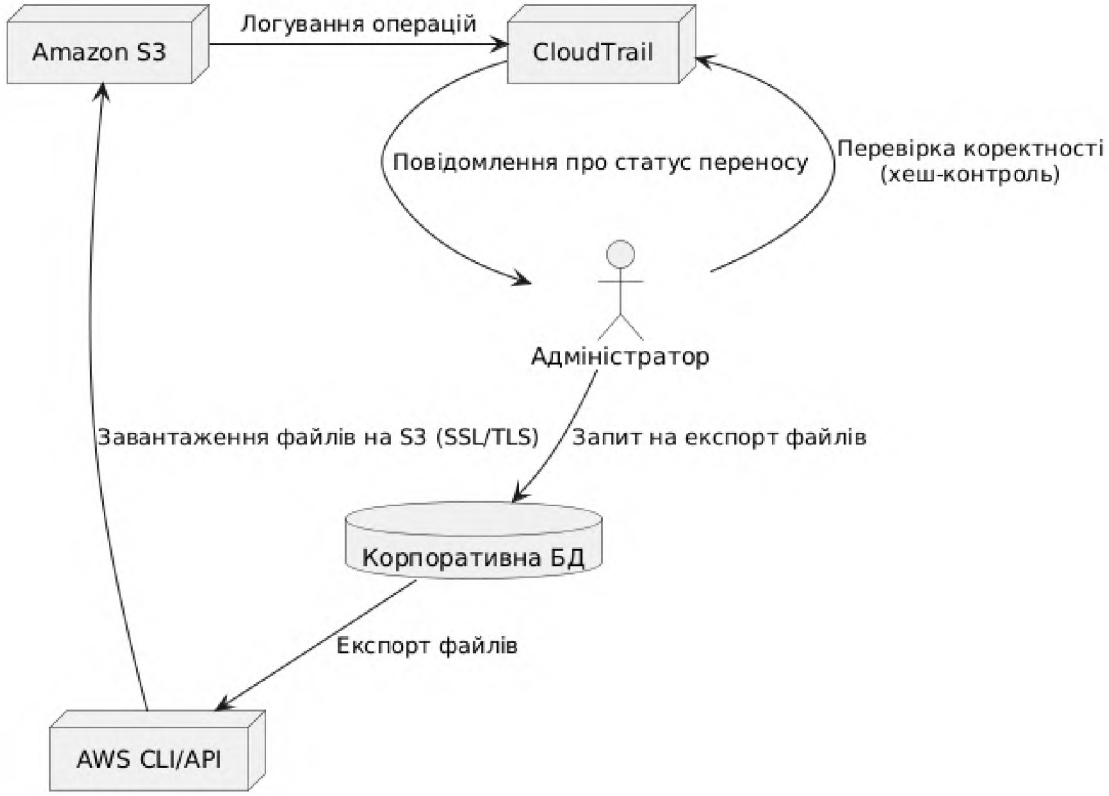

Рисунок 3.2 – Алгоритм переносу даних у Amazon S3

Етапи алгоритму переносу включають наступні деталі.

1. Експорт файлів із бази даних:

a) виконується SQL-запит для отримання бінарних даних, що потребують переносу;

b) застосовується рольова модель доступу, щоб доступ до файлів мали лише авторизовані користувачі;

c) експортовані файли конвертуються (при необхідності) у формат, сумісний з Amazon S3.

2. Завантаження файлів у Amazon S3:

a) файли передаються у S3 через API або SDK;

b) забезпечується захист передачі даних через шифрування SSL/TLS, щоб уникнути перехоплення даних;

c) налаштовуються політики доступу IAM, які обмежують доступ до конкретних bucket-ів S3 для запобігання несанкціонованому доступу.

3. Перевірка коректності перенесення:

a) після завантаження на Amazon S3 генерується контрольна сума (хеш) для кожного файлу і порівнюється з оригінальною;

b) зберігається журнал операцій у CloudTrail для подальшого аудиту та аналізу;

c) у випадку виявлення помилок або невідповідностей, файли повторно завантажуються.

3.2.2 Інтеграція заходів безпеки

На кожному етапі переносу даних впроваджуються заходи для забезпечення безпеки.

Інформація щодо заходів безпеки на етапах переносу даних узагальнена у таблиці 3.2.

Таблиця 3.2 – Заходи безпеки на етапах переносу даних

Етап переносу	Безпековий захід	Інструмент
Експорт файлів	Обмеження прав доступу	Ролі, політики IAM
Завантаження на S3	Шифрування даних, політики доступу	SSL/TLS, IAM
Перевірка коректності	Контрольні суми, логування операцій	Хеш-функції, Amazon CloudTrail

Заходи безпеки на етапі експорту файлів: обмеження доступу, яке передбачає використання ролей і політик доступу, щоб доступ до файлів мали лише авторизовані користувачі; контрольний запис – ведеться журнал доступу до файлів для подальшого аналізу.

На етапі завантаження на Amazon S3: шифрування файлів під час передачі через SSL/TLS; IAM політики (налаштування прав доступу до bucket-ів у Amazon S3, щоб обмежити доступ до файлів лише для авторизованих користувачів).

На етапі перевірки коректності перенесення: хеш-контроль (перевірка цілісності файлів за допомогою контрольних сум); логування та аудит –

використання Amazon CloudTrail для відстеження операцій, що забезпечує можливість подальшого аудиту.

3.2.3 Автоматизація перевірки коректності перенесення файлів

Перевірку коректності перенесення файлів можна автоматизувати, що значно спростить процес переносу і підвищить його надійність. Основний підхід до автоматизації полягає у використанні контрольних сум (хешів) та інструментів AWS для автоматичної перевірки файлів після їх завантаження.

Використання контрольних сум (MD5, SHA-256):

– на етапі експорту з корпоративної бази для кожного файлу обчислюється контрольна сума;

– після завантаження в Amazon S3 створюється скрипт або AWS Lambda-функція, яка автоматично обчислює контрольну суму для кожного файлу у S3 і порівнює її з оригінальною;

– у випадку невідповідності файли автоматично маркуються для повторного завантаження.

Використання AWS Lambda для автоматичної перевірки після завантаження:

– сервіс AWS Lambda може бути налаштований для запуску після кожного завантаження файлу у S3. Lambda-функція перевіряє контрольну суму та передає результат до логів або надсилає повідомлення адміністратору через Amazon SNS;

– якщо виявлено помилку, можна ініціювати повторне завантаження файлу автоматично або вручну.

Використання Amazon S3 Event Notifications:

– Amazon S3 підтримує Event Notifications, що дозволяє запускати AWS Lambda після кожної операції завантаження;

– Lambda-функція обробляє подію, порівнює хеші та записує статус операції у базу даних або відправляє звіт в Amazon CloudWatch.

Діаграма автоматизованої перевірки коректності представлена на рисунку 3.3.

Рисунок 3.3 – Діаграма автоматизованої перевірки коректності переносу даних

Перевагами автоматизації перевірки коректності є:

– підвищена надійність – автоматизація знижує ймовірність людських помилок і дозволяє виявляти некоректні файли відразу після завантаження;

– швидкість обробки – автоматичні скрипти та функції значно швидше виконують перевірку, ніж ручний контроль;

– сповіщення адміністраторів – у випадку виявлення невідповідностей автоматично надсилаються сповіщення, що дозволяє швидко вжити заходів для повторного переносу.

Автоматизована перевірка коректності є ефективним рішенням для великих обсягів даних, що знижує ризик втрати або пошкодження інформації під час переносу в Amazon S3.

3.2.4 Автоматизація системи сповіщень в Amazon S3

Для створення автоматизованої системи сповіщень в Amazon S3 можна скористатись послугами Amazon S3 Event Notifications разом із AWS Lambda та Amazon SNS (Simple Notification Service). Комбінація цих сервісів дозволяє

відстежувати події, такі як завантаження файлів, і автоматично надсилати сповіщення адміністраторам про статус переносу або про виявлені помилки.

Автоматизація системи сповіщень в Amazon S3 включає наступні кроки.

1. Налаштування Amazon S3 Event Notifications:

– вибрати bucket S3, в якому будуть зберігатися файли;

– у налаштуваннях вибраного bucket-а перейти до розділу Event Notifications та створити нове сповіщення;

– вибрати події, які потрібно відстежувати (наприклад, s3:ObjectCreated:* для сповіщення про кожне нове завантаження файлу);

– налаштувати виклик AWS Lambda або надсилання повідомлень через Amazon SNS як ціль для подій.

2. Створення AWS Lambda-функції для обробки подій:

– в AWS Lambda створити нову функцію, яка буде запускатися при кожному завантаженні файлу в Amazon S3;

– у створеній Lambda-функції реалізувати перевірку файлів на коректність (наприклад, порівняння хешів) або інші необхідні дії;

– налаштувати функцію для відправки сповіщення через Amazon SNS у разі виявлення помилок або успішної перевірки.

3. Налаштування Amazon SNS для відправлення сповіщень:

– у консолі AWS SNS створити нову тему (topic), яка буде використовуватися для надсилання сповіщень;

– додати підписників до створеної теми (це можуть бути адреси електронної пошти, SMS, HTTP/HTTPS адреси кінцевих точок або інші послуги, що підтримують отримання повідомлень);

– після цього, коли AWS Lambda викликатиме SNS, всім підписникам теми будуть автоматично надсилатись відповідні повідомлення.

4. Тестування та перевірка системи:

– завантажити тестовий файл у Amazon S3, щоб перевірити, чи запускається Lambda і надсилається повідомлення у випадку успішного або невдалого переносу даних;

– переконатись, що повідомлення доходять до всіх підписників, вказаних у темі SNS.

На рисунку 3.4 представлена діаграма автоматизованої системи сповіщень у Amazon S3.

Рисунок 3.4 – Діаграма автоматизованої системи сповіщень

Запропоновані налаштування автоматизації системи сповіщень щодо переносу даних в Amazon S3 вирішують наступні завдання:

– швидке реагування – автоматичні сповіщення дозволяють адміністраторам миттєво отримувати інформацію про статус переносу файлів;

– ефективність – у випадку невідповідностей чи помилок система негайно повідомляє відповідальних осіб, що дозволяє швидко вжити заходів для усунення проблєми;

– гнучкість – Amazon SNS підтримує різні типи повідомлень (електронна пошта, SMS, HTTP), що дає змогу підлаштувати систему сповіщень під конкретні потреби.

Загалом, автоматизована система сповіщень спрощує процес контролю за перенесенням файлів і підвищує надійність переносу, особливо при роботі з великими обсягами даних.

Таким чином, запропонований алгоритм забезпечує ефективне перенесення даних у Amazon S3 з інтеграцією заходів безпеки на кожному етапі. Використання

контрольних сум для перевірки цілісності, а також ведення журналів у CloudTrail забезпечує захист даних і прозорість усіх операцій.

3.3. Практична реалізація методології перенесення бінарних файлів у хмару

3.3.1. Оцінка поточного стану локальної системи зберігання даних

У сучасних корпоративних середовищах, де постійно зростає обсяг інформації, оптимізація зберігання даних є однією з пріоритетних задач. Можливим рішенням є впорядкування бінарних файлів у корпоративних базах даних через перенесення в хмарне сховище Amazon S3. Основна проблєма, яку вирішує це перенесення, полягає в зменшенні навантаження на локальні сховища, оптимізації витрат та підвищенні доступності файлів. Зважаючи на специфіку корпоративного зберігання даних, розглянута методологія містить не тільки технічні рекомендації, але й інструменти автоматизації.

На початковому етапі перенесення даних до хмарного сховища потрібно виконати аналіз структури даних у корпоративному сховищі, що включає:

– класифікацію файлів – визначення типів бінарних файлів (зображення, документи, мультимедійні файли тощо);

– оцінку обсягів даних – визначення обсягу файлів, частоти їх використання, необхідного рівня доступу та ступеня важливості;

– аналіз ризиків – аналіз безпекових аспектів і ризиків, пов’язаних із втратою даних, доступом до них і захистом конфіденційності при перенесенні в хмару.

На етапі класифікації визначаються основні типи файлів, які потрібно перенести в хмару. Характеристики файлів, представлені у таблиці, дозволяють передбачити та спланувати відповідні методи обробки, зберігання та доступу до файлів (таблиця 3.3).

Таблиця 3.3 – Типи файлів і їх характеристика

Тип файлу	Приклади	Обсяг файлів	Частота використання
Зображення	JPG, PNG, GIF	Середній	Висока (для зображень продукції)
Документи	PDF, DOCX, XLSX	Малий до середнього	Середня (для звітів)
Мультимедійні файли	MP3, MP4, AVI	Великий	Низька (для рекламних матеріалів)
Архіви	ZIP, RAR	Великий	Низька (для резервних копій)
Логи	LOG, JSON	Залежить від обсягів даних	Висока (для моніторингу систем)

Оцінка обсягів даних дозволяє визначити, які ресурси потрібні для перенесення та зберігання даних у хмарі. На основі цієї оцінки можна вибрати відповідний клас зберігання в Amazon S3 та налаштувати політики доступу до файлів (таблиця 3.4).

Таблиця 3.4 – Оцінка обсягів даних

Тип файлу	Середній обсяг (Гб)	Частота використання	Рівень доступу	Ступінь важливості
Зображення	5-10	Висока	Швидкий доступ	Високий
Документи	1-2	Середня	Періодичний доступ	Середній
Мультимедійні файли	50-100	Низька	Архівний доступ	Низький
Архіви	20-30	Низька	Архівний доступ	Високий
Логи	Змінний	Висока	Швидкий доступ	Середній

Аналіз ризиків полягає у визначенні потенційних проблєм при перенесенні даних у хмару, зокрема безпекових аспектів і питань конфіденційності: важливо врахувати ризики втрати або пошкодження даних, а також загрози несанкціонованого доступу.

На основі аналізу ризиків можна розробити стратегію захисту даних, що включає впровадження заходів безпеки на кожному етапі перенесення. Зокрема, необхідно використовувати шифрування даних як під час передачі (TLS/SSL), так і під час зберігання (SSE – Server-Side Encryption) в Amazon S3. Крім того, важливо налаштувати контроль доступу через IAM-політики, обмеживши права користувачів відповідно до їхніх ролей та завдань. Для забезпечення прозорості

операцій доцільно використовувати AWS CloudTrail для логування всіх дій, пов'язаних із доступом до даних і їх перенесенням (таблиця 3.5).

Таблиця 3.5 – Основні ризики та заходи безпеки при перенесенні даних у хмару

Ризик	Опис	Заходи безпеки
Втрата даних	Ризик втрати файлів під час передачі або зберігання у хмарі	Резервне копіювання, шифрування
Несанкціонований доступ	Можливість доступу до файлів сторонніх осіб	Політики доступу IAM, шифрування, багаторівнева автентифікація
Зниження продуктивності	Потреба в постійному доступі може перевантажити ресурси	Вибір відповідного класу зберігання
Порушення конфіденційності	Ризик витоку конфіденційної інформації під час передачі у хмару	Шифрування даних при передачі (SSL/TLS), контроль доступу

Діаграма аналізу структури даних у корпоративному сховищі відображена на рисунку 3.5.

Рисунок 3.5 – Діаграма для етапів аналізу структури даних

Відповідно до наведеної діаграми, аналітик даних отримує доступ до корпоративної бази даних для збору інформації. Далі, дані проходять етап класифікації файлів, де визначаються типи та призначення файлів. На етапі оцінки обсягів даних встановлюються частота використання і важливість файлів. Завершальний етап – аналіз ризиків, що дозволяє визначити необхідні заходи безпеки перед перенесенням даних у хмару.

Таким чином, головним завданням початкового етапу перенесення даних є глибокий аналіз структури даних у корпоративному середовищі, що дозволяє краще зрозуміти обсяги, частоту використання, рівень доступу та можливі ризики, пов'язані з перенесенням у хмару.

3.3.2. Обґрунтування вибору хмарного сховища

Amazon S3 є оптимальним рішенням для зберігання корпоративних даних через низку переваг, зокрема:

– гнучкість у налаштуванні – Amazon S3 підтримує різні класи зберігання даних, що дозволяє зберігати файли в залежності від частоти доступу;

– безпека – можливість використання функцій шифрування, як Amazon S3 Server-Side Encryption (SSE), що забезпечує конфіденційність даних;

– масштабованість та доступність – Amazon S3 надає можливість автоматичного масштабування обсягу даних та налаштування політик доступу, що важливо для корпоративного середовища.

Отже, хмарне сховище Amazon S3 було обрано як оптимальне рішення через низку переваг, які роблять його універсальним і надійним для використання в корпоративному середовищі. Основні переваги Amazon S3 та їх практичне значення узагальнені у таблиці 3.6.

Таблиця 3.6 – Основні переваги Amazon S3

Перевага	Опис	Практичне значення
Гнучкість у налаштуванні	Amazon S3 пропонує різні класи зберігання даних, які можна обирати залежно від частоти доступу до файлів	Оптимізація витрат за рахунок вибору класу зберігання для рідко використовуваних даних
Безпека	Підтримка шифрування, включно з S3 Server-Side Encryption (SSE), а також контроль доступу IAM	Захист конфіденційних даних і контроль доступу
Масштабованість	Amazon S3 автоматично масштабується для підтримки великих обсягів даних	Можливість зберігати будь-які обсяги даних
Доступність	Дані доступні з будь-якої точки світу, з налаштуванням політик доступу	Зручність для співробітників та клієнтів
Відмовостійкість	Дублювання даних у різних регіонах і політика відновлення даних	Мінімізація ризику втрати даних

Архітектура використання Amazon S3 представлена на рисунку 3.6.

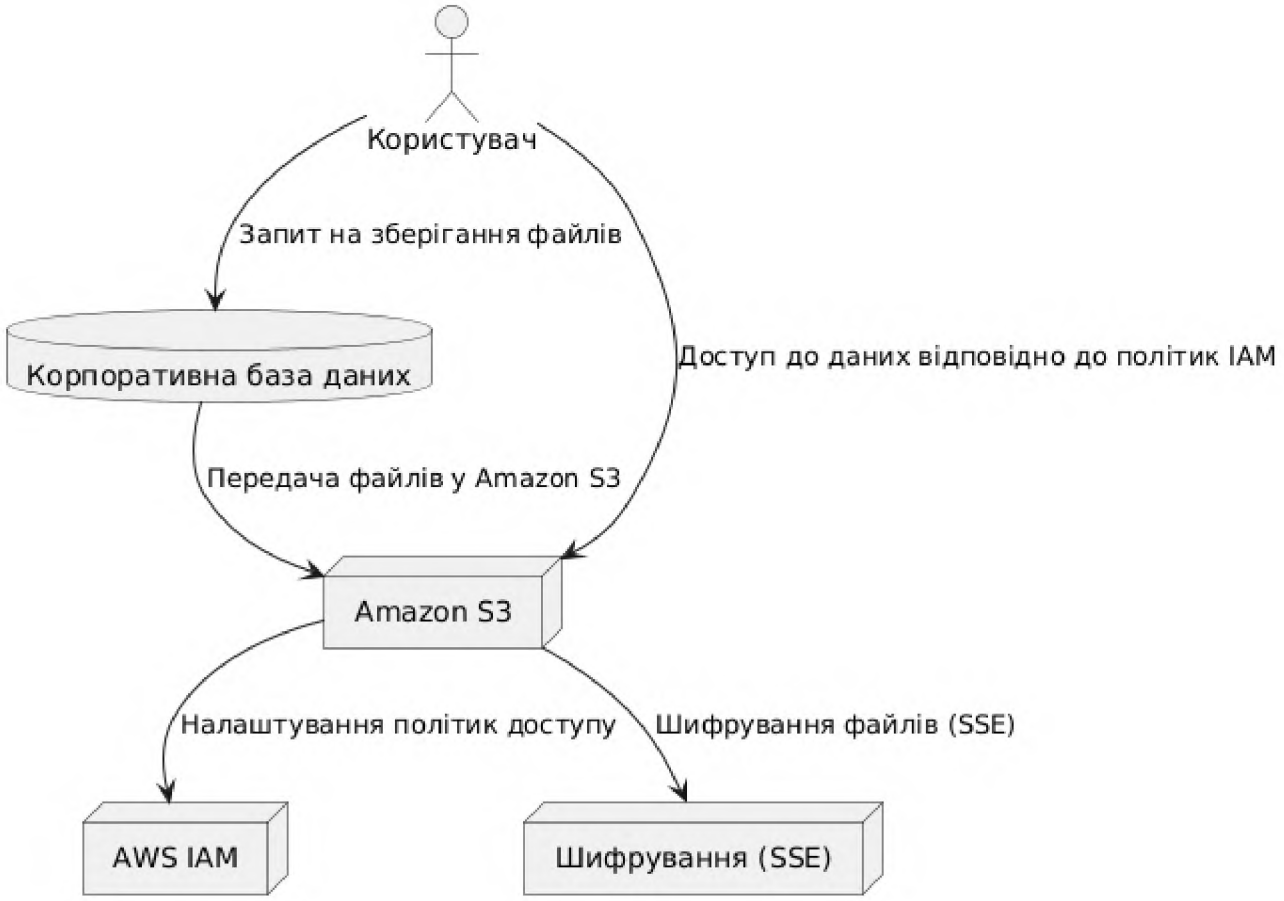

Рисунок 3.6 – Архітектура використання Amazon S3

На рисунку 3.6 користувач взаємодіє з корпоративною базою даних і може зберігати або запитувати файли для переносу в Amazon S3. Корпоративна база даних передає файли в Amazon S3 для зберігання. Шифрування виконується автоматично під час завантаження файлів у Amazon S3, забезпечуючи їх захист. AWS IAM контролює доступ до даних, застосовуючи відповідні політики доступу, що обмежує доступ до файлів лише для авторизованих користувачів.

Гнучкість у налаштуванні. Amazon S3 підтримує кілька класів зберігання даних (наприклад, S3 Standard, S3 Intelligent-Tiering, S3 Glacier), що дозволяє вибирати оптимальні параметри для зберігання даних залежно від частоти доступу. Наприклад, дані, до яких рідко звертаються, можна зберігати у класі S3 Glacier, знижуючи витрати на зберігання. Такі можливості є корисними для

корпоративних середовищ, де є дані різного типу та з різними вимогами до доступності.

Безпека. Amazon S3 пропонує багаторівневий захист даних за допомогою шифрування на серверній стороні (Server-Side Encryption, SSE) і політик доступу IAM, які дозволяють детально контролювати доступ до даних. За допомогою SSE-S3 (шифрування за замовчуванням) дані автоматично шифруються під час запису та зберігаються у зашифрованому вигляді. Це дозволяє захистити чутливу інформацію та відповідати вимогам безпеки.

Масштабованість і доступність. Amazon S3 забезпечує автоматичне масштабування для зберігання великих обсягів даних, що робить його ідеальним для середовищ, де обсяги зберігання можуть значно змінюватися. Доступність даних 99,99% у різних регіонах AWS дозволяє уникнути збоїв у доступі до інформації і забезпечує стабільну роботу.

Отже, вибір Amazon S3 як хмарного сховища обґрунтований його високою гнучкістю, безпекою та масштабованістю, що є особливо важливим у корпоративному середовищі. Можливість вибору класів зберігання, шифрування даних та гнучкі налаштування доступу дозволяють забезпечити ефективне, безпечне та економічне зберігання даних.

3.3.3 Вибір класу зберігання Amazon S3

Amazon S3 пропонує кілька класів зберігання для різних потреб, що дозволяє оптимізувати витрати залежно від частоти доступу до даних і необхідного рівня захисту та відновлення. Кожен клас зберігання має свою специфіку, призначення та ціну.

S3 Standard – це стандартний клас зберігання, який забезпечує високу продуктивність і швидкий доступ до даних. Найкраще підходить для часто використовуваних даних, де потрібна низька затримка і висока доступність.

S3 Intelligent-Tiering – інтелектуальний клас зберігання, який автоматично переміщує дані між частим і рідким доступом, залежно від частоти звернень до даних. Оптимальний для даних, частота доступу до яких може змінюватися з часом, що зменшує витрати на зберігання.

S3 Standard-IA (Infrequent Access). Призначений для рідко використовуваних даних, але які потребують швидкого доступу у разі потреби. Вартість зберігання нижча, ніж у S3 Standard, але є додаткові витрати на запити, що робить його ефективним для архівів і резервних копій.

Інформація щодо характеристик класів зберігання Amazon S3 представлена у таблиці 3.7.

Таблиця 3.7 – Основні класи зберігання в Amazon S3

Клас зберігання	Призначення	Характеристики	Приклади використання
S3 Standard	Для даних, які часто використовуються	Висока доступність (99,99%), низька затримка, три копії даних для відмовостійкості	Сайти, програми з високими вимогами до швидкого доступу
S3 Intelligent-Tiering	Для даних зі змінною частотою доступу	Автоматичне переміщення між частим і рідким доступом, зменшення витрат	Дані з непередбачуваною частотою доступу
S3 Standard-IA (Infrequent Access)	Для рідко використовуваних даних, що потребують швидкого доступу	Висока доступність, менша ціна за зберігання, але додаткові витрати на запити	Архіви звітів, резервні копії, доступ до яких потрібен рідко
S3 One Zone-IA	Для рідко використовуваних даних, що не вимагають високої відмовостійкості	Дані зберігаються в одній зоні доступності, нижчі витрати, ризик втрати у разі аварії зони	Локальні резервні копії, тимчасові або вторинні дані
S3 Glacier	Для архівного зберігання, де низька вартість зберігання є критичною	Дуже низька ціна за зберігання, дані доступні протягом кількох годин	Архіви історичних даних, які потрібні рідко
S3 Glacier Deep Archive	Найнижча ціна за зберігання, призначений для даних, які майже ніколи не використовуються	Найдешевший клас, доступ до даних потребує від 12 до 48 годин	Архіви для відповідності вимогам або юридичних цілей

S3 One Zone-IA. Аналог S3 Standard-IA, але зберігає дані лише в одній зоні доступності, що знижує вартість. Підходить для даних, які не потребують високої відмовостійкості та можуть бути відновлені з іншого джерела у разі збою.

S3 Glacier. Клас для довгострокового архівного зберігання, де дані потрібні рідко, і час доступу може сягати кількох годин. Витрати на зберігання дуже

низькі, але потрібен тривалий час для відновлення, тому він підходить для історичних архівів.

S3 Glacier Deep Archive. Найдешевший клас зберігання, призначений для даних, які майже ніколи не використовуються, наприклад, для юридичних архівів. Час доступу складає від 12 до 48 годин, що достатньо для даних, які потрібно зберігати тривалий період, але до яких майже не звертаються

Критерії вибору класу зберігання в Amazon S3 представлено на рисунку 3.7.

Рисунок 3.7 – Вибір класу зберігання в Amazon S3

Отже, Amazon S3 пропонує широкий вибір класів зберігання, що дозволяє оптимізувати витрати, забезпечуючи необхідну доступність і швидкість для різних типів даних. Правильний вибір класу зберігання залежить від частоти доступу до даних, вимог до доступності та допустимого часу відновлення.

3.3.4 Технічні аспекти процесу перенесення

Підготовка та екстракція файлів. Першим кроком процесу перенесення є розробка сценаріїв для екстракції бінарних файлів із корпоративної бази даних. Екстракція включає такі етапи:

- створення копій файлів, які підлягають перенесенню;
- конвертація даних у формат, придатний для зберігання в S3 (за потреби);
- верифікація даних, щоб уникнути помилок під час завантаження.

Підготовка та екстракція файлів є важливим кроком для переносу даних у Amazon S3. На цьому етапі створюються скрипти або сценарії, які виконують екстракцію бінарних файлів із корпоративної бази даних та підготовку їх до завантаження в хмарне сховище. Для досягнення цього процес включає кілька основних етапів (таблиця 3.8).

Таблиця 3.8 – Етапи підготовки та екстракції файлів

Етап	Опис	Задача
Створення копій файлів	Здійснюється копіювання файлів, які потрібно перенести, щоб оригінали залишалися у базі даних	Забезпечити збереження оригінальних даних
Конвертація даних	Файли перетворюються у формат, який підтримується Amazon S3, наприклад, PNG для зображень	Підготувати дані до завантаження в S3
Верифікація даних	Перевірка цілісності та якості даних перед завантаженням для мінімізації ризиків	Уникнути помилок під час завантаження і зберегти якість файлів

Створення копій файлів. На першому етапі екстракції скрипт виконує копіювання файлів із корпоративної бази, щоб запобігти впливу на оригінальні дані. Це дозволяє уникнути випадкового видалення або пошкодження файлів, зберігаючи їх у базі даних.

Конвертація даних у відповідний формат. Деякі файли можуть зберігатися у форматах, які не є оптимальними для зберігання в S3, або можуть мати надмірний розмір. Наприклад, зображення можуть бути стиснуті та перетворені у формати PNG або JPEG для зменшення обсягів зберігання. Конвертація включає перевірку підтримки S3 для відповідних форматів, що полегшує подальшу обробку та доступ до файлів.

Верифікація даних. На завершальному етапі виконується верифікація даних, яка передбачає перевірку цілісності файлів перед завантаженням. Цей етап може включати обчислення хешів (наприклад, MD5) та порівняння з оригіналами для впевненості у відсутності помилок. Верифікація допомагає мінімізувати ризик невдалого завантаження та помилок під час переносу.

Діаграма процесу підготовки та екстракції файлів корпоративної бази даних представлена на рисунку 3.8.

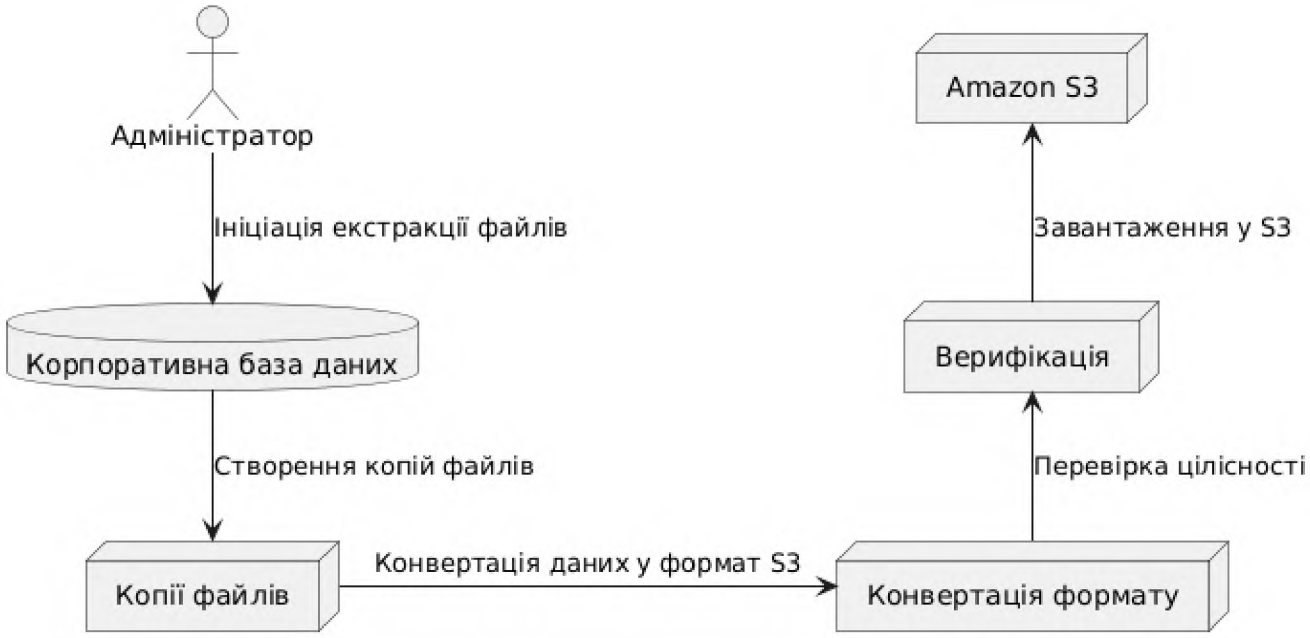

Рисунок 3.8 – Діаграма підготовки та екстракції файлів

Діаграма на рисунку 3.8 описує наступну послідовність дій: адміністратор ініціює процес екстракції файлів з корпоративної бази даних; корпоративна база даних передає файли для копіювання, щоб створити резервні копії даних перед перенесенням; копії файлів проходять етап конвертації для приведення файлів у сумісний формат з Amazon S3; верифікація перевіряє цілісність файлів перед завантаженням, щоб уникнути помилок; після завершення етапів підготовки файли завантажуються у Amazon S3.

Перевагами описаної ретельної підготовки даних є: збереження оригіналів (створення копій даних дозволяє захистити вихідну інформацію від змін і втрат); оптимізація зберігання (конвертація у придатний для S3 формат зменшує обсяги даних і підвищує швидкість доступу); захист від помилок (верифікація допомагає запобігти втратам даних під час завантаження в хмару).

Етап підготовки та екстракції файлів є важливим для ефективного переносу даних у хмару. Ретельна підготовка дозволяє зберегти цілісність даних, оптимізувати зберігання та уникнути помилок під час завантаження.

Налаштування bucket-ів Amazon S3. Наступним етапом є налаштування bucket-ів у Amazon S3, де будуть зберігатися файли. Bucket в Amazon S3 є фундаментальною структурою даних, що виконує роль віртуального контейнера для зберігання об'єктів довільного розміру. В межах сервісу S3 відро слугує основною одиницею організації даних, дозволяючи зберігати та керувати різноманітними типами файлів, від мультимедійних ресурсів (зображення, відео) до великомасштабних наборів даних та архівних матеріалів. Налаштування bucket-ів у Amazon S3 є важливим етапом для організації зберігання файлів, що забезпечує зручність доступу та високу безпеку.Основними рекомендаціями є наступні:

– структурувати bucket-и на основі типів даних або рівнів доступу;

– застосовувати політики доступу для bucket-ів, щоб обмежити доступ до файлів відповідно до вимог безпеки;

– використовувати шифрування для bucket-ів для захисту від несанкціонованого доступу.

Правильна структура bucket-ів дозволяє оптимально зберігати дані різного типу, розмежовувати доступ до файлів і забезпечувати конфіденційність. Основні рекомендації щодо налаштування bucket-ів Amazon S3 представлені у таблиці 3.9.

Таблиця 3.9 – Рекомендації щодо налаштування bucket-ів Amazon S3

Рекомендація	Опис	Переваги
Структурування bucket-ів	Організація bucket-ів на основі типів даних або рівнів доступу	Спрощений доступ до даних, ефективний контроль доступу
Застосування політик доступу	Використання політик доступу IAM та ACL для обмеження доступу до різних bucket-ів	Контроль за доступом, захист конфіденційних даних
Використання шифрування	Увімкнення шифрування для захисту даних при зберіганні та передачі	Захист від несанкціонованого доступу, відповідність вимогам безпеки

Рекомендується структурувати bucket-и на основі типів даних, які зберігаються (наприклад, зображення, документи, мультимедіа) або на основі рівнів доступу (загальнодоступні файли, внутрішні документи). Це дозволяє легше організовувати дані і забезпечує простіший контроль доступу. Наприклад, зображення продуктів можна зберігати в окремому bucket-і, що є загальнодоступним, тоді як конфіденційні документи – у bucket-і з обмеженим доступом.

Використання політик доступу (IAM Policy) і списків керування доступом (ACL) дозволяє обмежити доступ до bucket-ів. Це забезпечує можливість визначення, хто і до яких bucket-ів має доступ, що є особливо важливим для чутливих даних. Політики можуть обмежувати доступ за такими критеріями, як роль користувача, IP-адреса або географічне положення. Наприклад, внутрішні документи можуть бути доступні лише для певних груп співробітників.

Amazon S3 підтримує кілька методів шифрування, включаючи Server-Side Encryption (SSE-S3, SSE-KMS) та Client-Side Encryption. Шифрування bucket-ів забезпечує безпеку даних під час зберігання та передачі. Це особливо важливо для конфіденційних або чутливих даних, оскільки захищає їх від несанкціонованого доступу.

Діаграма процесу налаштування bucket-ів у Amazon S3 відображена на рисунку 3.8.

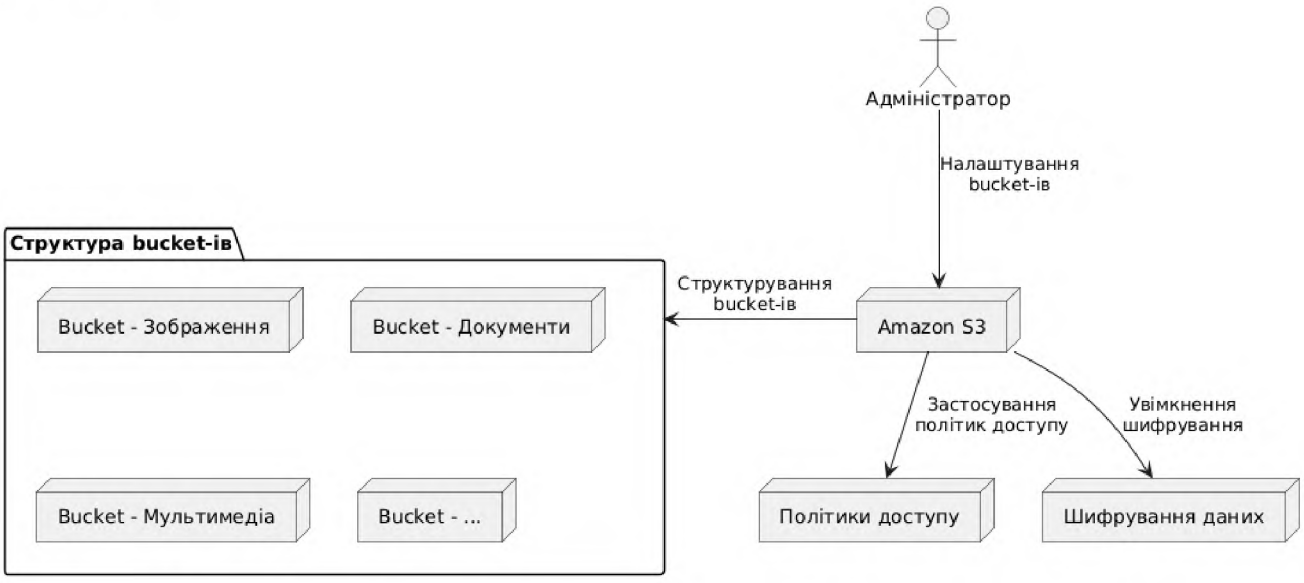

Рисунок 3.9 – Діаграма для налаштування bucket-ів у Amazon S3

Адміністратор налаштовує bucket-и в Amazon S3, визначає їх структуру та параметри безпеки. Структура bucket-ів налаштовується для кожного типу даних: зображення, документи, мультимедіа. Політики доступу, що встановлюються для контролю доступу до кожного bucket-а, визначають, хто має право на перегляд, завантаження або зміну файлів. Шифрування даних, що вмикається для bucket-ів, забезпечує конфіденційність і безпеку збережених даних (таблиця 3.10).

Таблиця 3.10 – Приклади налаштування bucket-ів у Amazon S3

Bucket	Призначення	Політика доступу	Шифрування
Зображення	Зберігання загальнодоступних зображень	Загальнодоступний доступ	Не обов'язкове
Документи	Внутрішні документи компанії	Доступ тільки для співробітників	SSE-KMS (шифрування ключами)
Мультимедіа	Рекламні матеріали	Доступ для маркетингової групи	SSE-S3

Правильне налаштування bucket-ів у Amazon S3 є важливим для безпечного та ефективного зберігання даних. Структурування на основі типів файлів або рівнів доступу, застосування політик доступу і використання шифрування дозволяють створити надійну систему зберігання, яка відповідає вимогам корпоративної безпеки і забезпечує легкий доступ до файлів.

Автоматизація процесу завантаження файлів у хмару значно спрощує управління даними. Для цього використовується AWS SDK або AWS CLI. У довідковій системі сервісу надаються приклади команд для завантаження файлів, оновлення bucket-ів і налаштування політик доступу. Зокрема, програмний код для автоматизації завантаження файлу у S3 через AWS CLI:

```
aws s3 cp /local/path/to/file s3://bucket-name/path/to/file
```

Додатково рекомендується використовувати моніторинг процесу через AWS CloudWatch, що дозволяє відстежувати статус завантаження файлів і своєчасно реагувати на помилки.

Автоматизація процесу завантаження файлів у Amazon S3 дозволяє спростити управління даними та забезпечує їх своєчасне оновлення в хмарі. Використання AWS SDK або AWS CLI дозволяє ефективно завантажувати файли, оновлювати bucket-и та налаштовувати політики доступу (таблиця 3.11).

Таблиця 3.11 – Основні компоненти автоматизації завантаження файлів

Компонент	Опис	Приклади використання
AWS CLI	Інструмент командного рядка, який дозволяє швидко завантажувати файли і виконувати інші операції	Команда для завантаження файлів aws s3 cp
AWS SDK	Програмний інтерфейс, який дозволяє інтегрувати завантаження файлів у додатки	Автоматизація завантаження на різних мовах програмування
AWS CloudWatch	Система моніторингу, що дозволяє відстежувати статус завантаження і реагувати на помилки	Сповіщення через CloudWatch у разі помилки завантаження

Приклад використання AWS CLI для завантаження файлу в S3. AWS CLI дозволяє швидко завантажити файл у bucket S3 за допомогою консольної команди aws s3 cp:

```
# Завантаження файлу у S3
aws s3 cp /local/path/to/file s3://bucket-name/path/to/file

# Параметри:
# /local/path/to/file - локальний шлях до файлу, який потрібно
завантажити
# bucket-name - ім'я bucket-а у S3, де буде зберігатися файл
# path/to/file - шлях до файлу в bucket-і S3
Приклад програмного Python-коду для завантаження файлу через AWS
SDK:
import boto3
from botocore.exceptions import NoCredentialsError

# Ініціалізація клієнта S3
s3 = boto3.client('s3')

def upload_file_to_s3(file_name, bucket, object_name=None):
    """Завантаження файлу у S3 bucket.

    Parameters:
    file_name (str): Шлях до файлу на локальному комп'ютері
    bucket (str): Назва bucket-а S3
    object_name (str, optional): Шлях і назва файлу у S3 (за
замовчуванням той же, що file_name)
```

```
    Returns:
    bool: Повертає True, якщо файл успішно завантажений, інакше
False
    """
    if object_name is None:
        object_name = file_name

    try:
        # Завантаження файлу в S3
        s3.upload_file(file_name, bucket, object_name)
        print(f"Файл {file_name} успішно завантажено у
{bucket}/{object_name}")
        return True
    except FileNotFoundError:
        print("Файл не знайдено.")
        return False
    except NoCredentialsError:
        print("Помилка аутентифікації.")
        return False

# Використання функції
upload_file_to_s3('/local/path/to/file', 'bucket-name',
'path/to/file')
```

Налаштування моніторингу через AWS CloudWatch дозволяє відстежувати статус завантаження файлів і отримувати сповіщення у разі помилок або проблєм. Налаштування сповіщень у AWS CloudWatch виконується так: спочатку створюється правило події для відстеження дій у bucket-і, таких як PutObject або CopyObject, а потім визначається сповіщення, яке надсилається у разі помилки, наприклад, через Amazon SNS (Simple Notification Service).

Приклад налаштування моніторингу з AWS CloudWatch через AWS CLI (виконується через консоль):

```
# Команда для створення правила події у CloudWatch для S3
aws events put-rule --name "S3UploadMonitoringRule" \
    --event-pattern '{
        "source": ["aws.s3"],
        "detail-type": ["AWS API Call via CloudTrail"],
        "detail": {
            "eventSource": ["s3.amazonaws.com"],
            "eventName": ["PutObject"]
        }
    }' \
    --state ENABLED

# Додавання SNS для відправки сповіщення у разі помилки
aws sns create-topic --name S3UploadNotifications
```

```
aws sns subscribe --topic-arn arn:aws:sns:us-west-
2:123456789012:S3UploadNotifications \
    --protocol email --notification-endpoint "your-
email@example.com"
```

Діаграма автоматизації процесу завантаження файлів у Amazon S3 показана на рисунку 3.10.

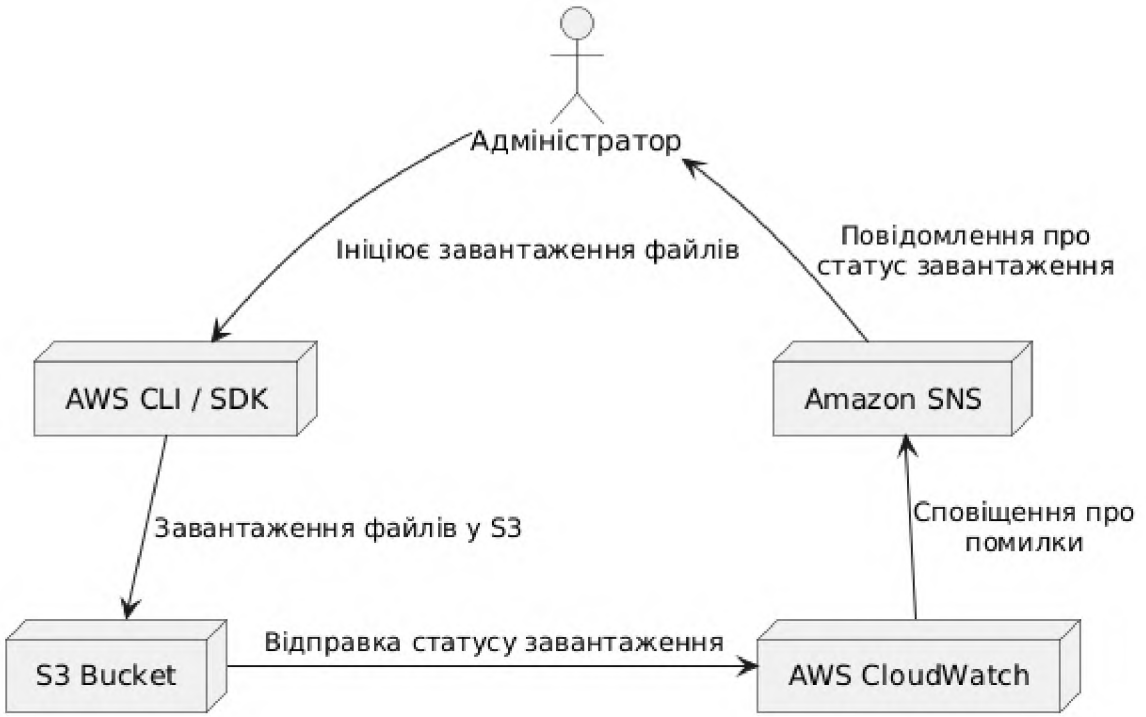

Рисунок 3.10 – Автоматизація процесу завантаження файлів у Amazon S3

Адміністратор ініціює завантаження файлів через AWS CLI або SDK. CLI / SDK завантажує файли у bucket S3. CloudWatch відстежує процес завантаження, контролюючи статус операцій. У випадку помилок CloudWatch надсилає сповіщення через Amazon SNS, яке доходить до адміністратора.

Основні переваги автоматизації завантаження файлів:

– автоматизація через CLI або SDK дозволяє швидко завантажувати дані у S3, економити час на рутинних операціях;

– AWS CloudWatch дозволяє відстежувати процес завантаження і отримувати сповіщення про помилки;

– завдяки політикам доступу IAM і моніторингу у AWS CloudWatch забезпечується захист файлів під час завантаження.

Автоматизація завантаження файлів у Amazon S3 за допомогою AWS CLI або SDK спрощує управління даними та дозволяє створити надійну систему для постійного оновлення хмарного сховища. Налаштування моніторингу в AWS CloudWatch і сповіщень у Amazon SNS допомагає швидко реагувати на помилки, забезпечуючи високий рівень контролю за завантаженням файлів.

Таким чином, етап підготовки та екстракції файлів є важливим для ефективного переносу даних у хмару. Ретельна підготовка дозволяє зберегти цілісність даних, оптимізувати зберігання та уникнути помилок під час завантаження.

Правильне налаштування bucket-ів у Amazon S3 є важливим для безпечного та ефективного зберігання даних. Структурування на основі типів файлів або рівнів доступу, застосування політик доступу і використання шифрування дозволяють створити надійну систему зберігання, яка відповідає вимогам корпоративної безпеки і забезпечує легкий доступ до файлів.

Автоматизація завантаження файлів у Amazon S3 за допомогою AWS CLI або SDK спрощує управління даними та дозволяє створити надійну систему для постійного оновлення хмарного сховища. Налаштування моніторингу в AWS CloudWatch і сповіщень у Amazon SNS допомагає швидко реагувати на помилки, забезпечуючи високий рівень контролю за завантаженням файлів.

3.3.5 Міграція даних у єдиний кластер

Для оптимізації управління файлами рекомендується забезпечити централізацію файлів, для чого потрібно здійснити міграцію всіх даних у єдиний кластер. Завдяки цьому процесу спрощується контроль за даними, прискорюється доступ до них і зменшується кількість збоїв, пов'язаних із розподілом даних між різними локаціями.

Міграція даних у єдиний кластер передбачає перенесення всіх файлів та інформаційних ресурсів в одну центральну локацію або хмарний кластер. Це забезпечує централізоване управління, спрощує доступ до файлів і покращує їхню

відмовостійкість. Централізація даних дозволяє більш ефективно контролювати ресурси, прискорює процеси обробки запитів і зменшує ризики, пов'язані з розподіленим зберіганням даних.

Переваги міграції у єдиний кластер узагальнені у таблиці 3.11.

Таблиця 3.11 – Переваги міграції у єдиний кластер

Перевага	Опис
Централізоване управління	Легше управляти даними, коли вони зберігаються в одній локації
Прискорення доступу	Доступ до файлів відбувається швидше, оскільки всі дані знаходяться в одному кластері
Зменшення збоїв	Менше помилок, оскільки немає необхідності звертатися до різних локацій для отримання файлів
Покращена відмовостійкість	Єдиний кластер дозволяє налаштувати резервування та реплікацію, що підвищує надійність

Основними етапами міграції даних у єдиний кластер є планування міграції, підготовка даних до міграції, процес міграції, оптимізація та налаштування доступу до даних.

На першому етапі планування міграції виконується визначення обсягів даних, які потрібно перенести, і потреб щодо пропускної здатності мережі, а також вибір відповідного кластеру або хмарної платформи для централізованого зберігання, наприклад, Amazon S3 з організацією у єдиний bucket або використанням Amazon S3 для архівних даних у Glacier.

Етап підготовки даних до міграції включає завданні класифікації та фільтрації файлів перед перенесенням (наприклад, зберігання рідко використовуваних файлів у архівному класі Glacier), а також налаштування шифрування та захисту для конфіденційних даних перед перенесенням у єдиний кластер.

Етап міграції включає використання AWS CLI або SDK для автоматизації перенесення даних з різних локацій у єдиний кластер, моніторинг та контроль за процесом міграції для забезпечення цілісності даних.

На етапі оптимізації та налаштування доступу відбувається налаштування прав доступу та політик для централізованого управління доступом до файлів та

оптимізація кластеру через політики життєвого циклу для зниження витрат на зберігання.

Прикладом реалізації цих етапів може слугувати використання AWS CLI для міграції файлів у єдиний кластер:

```
# Копіювання даних з локальної локації у центральний bucket
aws s3 cp /local/path/to/file s3://central-bucket-name/path/to/file
--recursive

# Параметри:
# /local/path/to/file - шлях до файлу або директорії на локальному
комп'ютері
# central-bucket-name - ім'я bucket-а у S3, де буде централізовано
зберігатися файл
# --recursive - для перенесення всіх файлів у директорії
```

Діаграма процесу міграції даних у єдиний кластер представлена на рисунку 3.11.

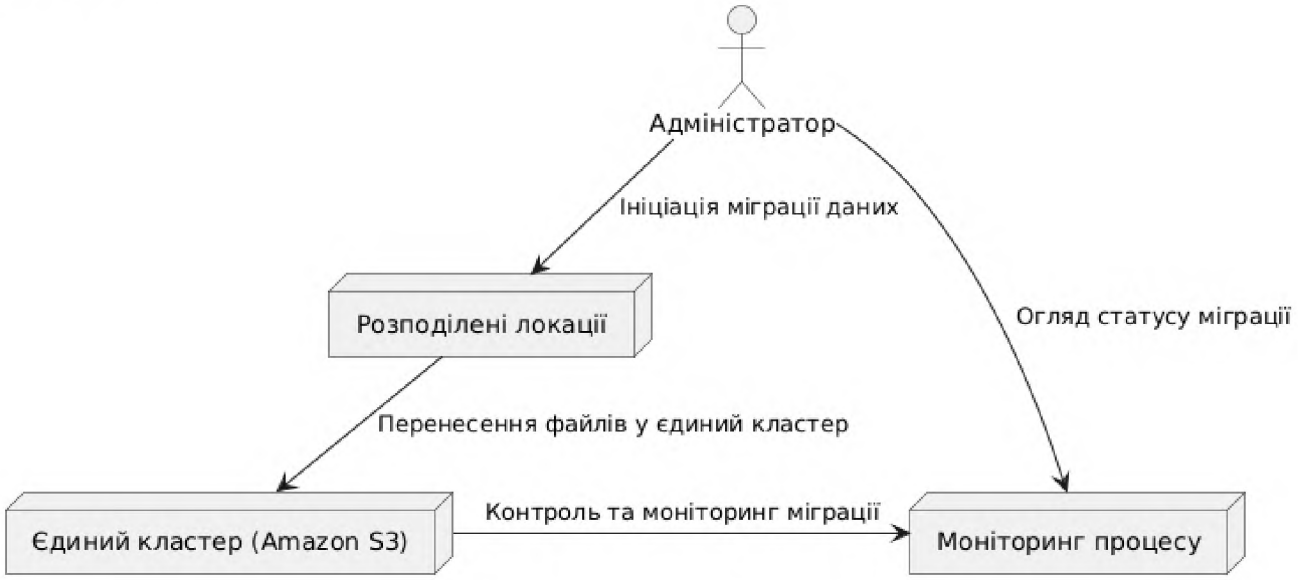

Рисунок 3.11 – Діаграма процесу міграції даних у єдиний кластер

Опис UML-діаграми: адміністратор ініціює процес міграції даних з розподілених локацій у єдиний кластер; розподілені локації передають файли у центральний кластер (Amazon S3), де відбувається централізація; моніторинг процесу дозволяє адміністратору відстежувати статус і цілісність міграції даних.

Перевагами централізації даних у єдиному кластері є: зручність управління – адміністратори отримують можливість централізовано керувати усіма файлами,

а також налаштовувати політики доступу і зберігання; ефективний контроль доступу – з єдиного кластеру простіше контролювати, хто має доступ до файлів, та налаштовувати відповідні політики безпеки; зменшення затримок і помилок – всі файли знаходяться в одному кластері, що знижує ймовірність помилок через віддалений доступ до різних локацій.

Отже, міграція даних у єдиний кластер забезпечує централізоване управління, підвищує надійність та відмовостійкість системи і знижує витрати на управління даними. Використання Amazon S3 з оптимізованими налаштуваннями і політиками життєвого циклу робить централізацію ефективною, зручною і безпечною для корпоративного середовища.

3.3.6 Оптимізація витрат та управління файлами

Оптимізація витрат на зберігання даних у Amazon S3 є важливим аспектом для ефективного використання хмарних ресурсів, особливо при великих обсягах даних. Це досягається шляхом вибору відповідних класів зберігання (S3 Standard, Glacier, Deep Archive), налаштування політик життєвого циклу для автоматичного переміщення файлів між класами залежно від частоти доступу, а також регулярного моніторингу використання ресурсів через AWS CloudWatch. Упровадження таких рішень дозволяє мінімізувати витрати, забезпечуючи доступність і надійність даних. Використання класів зберігання, налаштування термінів зберігання та політик життєвого циклу дозволяє контролювати витрати і забезпечувати оптимальне управління файлами. Рекомендації для ефективного управління витратами в Amazon S3 включають:

– використання різних класів зберігання (для файлів, до яких рідко звертаються, пропонується використовувати клас Glacier, що знижує витрати на зберігання);

– кешування та управління термінами зберігання – можна налаштувати автоматичне видалення або архівування файлів, які не використовуються;

– налаштування політик життєвого циклу – це дозволяє автоматизувати процес переміщення файлів між класами зберігання в залежності від їхньої важливості та частоти доступу.

Рекомендації для оптимізації витрат та управління файлами у Amazon S3 представлені у таблиці 3.12.

Таблиця 3.12 – Рекомендації для оптимізації витрат та управління файлами

Метод оптимізації	Опис	Переваги
Різні класи зберігання	Використання класу зберігання Glacier для архівного зберігання рідко використовуваних файлів.	Зниження витрат на зберігання.
Кешування та терміни зберігання	Автоматичне видалення або архівування файлів після певного терміну.	Ефективне управління використанням простору.
Політики життєвого циклу	Автоматизація переміщення файлів між класами зберігання на основі частоти доступу та важливості.	Зменшення витрат, зручність управління файлами.

Для файлів, які нечасто використовуються, рекомендується застосовувати клас зберігання Glacier або Glacier Deep Archive, що значно знижує витрати. Часто доступні файли можна зберігати в класах S3 Standard або S3 Intelligent-Tiering для оптимального доступу і варіативності частоти звернень.

Налаштування автоматичного видалення або архівування файлів, які не використовуються, дозволяє контролювати використання простору у Amazon S3. Наприклад, файли, до яких не зверталися більше 90 днів, можуть автоматично архівуватися або видалятися для зменшення витрат.

Політики життєвого циклу дозволяють автоматизувати переміщення файлів між класами зберігання в залежності від частоти доступу. Наприклад, дані можуть автоматично переміщатися з класу S3 Standard до S3 Standard-IA після 30 днів без доступу, а потім до Glacier через 60 днів. Цей процес налаштовується через конфігурацію політик у bucket-ах Amazon S3, що забезпечує зручність і автоматизацію. Політика автоматичного переміщення файлів між класами зберігання може виглядати наступним чином (описується у форматі JSON):

```
{
    "Rules": [
        {
            "ID": "Архівування файлів",
            "Status": "Enabled",
            "Filter": {
```

```
                "Prefix": ""
            },
            "Transitions": [
                {
                    "Days": 30,
                    "StorageClass": "STANDARD_IA"
                },
                {
                    "Days": 60,
                    "StorageClass": "GLACIER"
                }
            ],
            "Expiration": {
                "Days": 365
            }
        }
    ]
}
```

Опис політики життєвого циклу включає: Rules – правила, які визначаються для bucket-a; ID – ідентифікатор політики (наприклад, «Архівування файлів»); Status – активний стан політики («Enabled»); Transitions – переміщення файлів між класами зберігання, наприклад, через 30 днів у STANDARD_IA і через 60 днів у GLACIER; Expiration – видалення файлів через 365 днів.

Діаграма процесу автоматизованого управління файлами в Amazon S3 через налаштування політик життєвого циклу представлена на рисунку 3.12.

Рисунок 3.12 – Автоматизація управління файлами в Amazon S3 через налаштування політик життєвого циклу

Адміністратор налаштовує політики життєвого циклу в Amazon S3 для автоматизації управління файлами. Політики життєвого циклу автоматично переміщують файли між класами зберігання відповідно до встановлених правил. Файли, до яких не зверталися довгий час, переміщуються у Glacier для архівного зберігання.

Основні напрямки оптимізації витрат:

– використання класів Glacier та політик життєвого циклу знижує витрати на зберігання даних;

– політики життєвого циклу спрощують процес переміщення файлів і зменшують необхідність у ручному управлінні;

– автоматичне видалення старих файлів або архівування дозволяє ефективно використовувати простір у Amazon S3.

Таким чином, оптимізація витрат і управління файлами у Amazon S3 через налаштування класів зберігання, термінів зберігання та політик життєвого циклу дозволяє знизити витрати та забезпечити ефективне використання хмарного сховища. Ці методи забезпечують автоматизацію та контроль за зберіганням, що особливо важливо для корпоративного середовища з великим обсягом даних.

3.4. Оцінка економічної ефективності перенесення бінарних файлів із корпоративних баз даних у хмарне сховище Amazon S3

Перенесення бінарних файлів із корпоративної бази даних PostgreSQL у хмарне сховище Amazon S3 може бути економічно обґрунтованим завдяки зниженню витрат на зберігання, покращенню продуктивності бази даних і спрощенню управління файлами.

Виконаємо розрахунки для компанії, що має корпоративну базу даних PostgreSQL, у якій зберігаються бінарні файли обсягом близько 10 ТБ, і частина цих файлів (8 ТБ) активно використовується, а інша частина (2 ТБ) є архівною і до неї звертаються рідко. Компанія планує перенести бінарні файли у хмарне

сховище Amazon S3 з використанням різних класів зберігання для зниження витрат: 8 ТБ активних файлів будуть зберігатися у класі S3 Standard для забезпечення швидкого доступу; 2 ТБ архівних файлів будуть зберігатися у класі S3 Glacier для зниження витрат.

Вихідні дані для розрахунку економічних показників переносу бінарних файлів із корпоративної бази даних PostgreSQL у хмарне сховище Amazon S3 представлені у таблиці 3.13.

Таблиця 3.13 – Вихідні дані для розрахунку економічних показників із бази даних PostgreSQL в Amazon S3

Показник	Значення
Загальний обсяг файлів	10 ТБ
Активні файли (частота доступу висока)	8 ТБ
Архівні файли (частота доступу низька)	2 ТБ
Вартість зберігання PostgreSQL на локальному сервері	$0,25 за ГБ/місяць
Вартість зберігання в S3 Standard	$0,023 за ГБ/місяць
Вартість зберігання в S3 Glacier	$0,004 за ГБ/місяць
Вартість запиту на доступ до файлу у Glacier	$0,01 за запит
Кількість запитів до архівних файлів	100 на місяць
Вартість передачі даних у S3	$0,09 за ГБ (одноразово)

Кошторис проєкту перенесення бінарних файлів із корпоративної бази даних PostgreSQL у хмарне сховище Amazon S3 представлений у таблиці 3.14.

Таблиця 3.14 – Кошторис переносення бінарних файлів із корпоративної бази даних PostgreSQL у хмарне сховище Amazon S3

Категорія	Обсяг (ГБ)	Вартість за одиницю	Загальна вартість на місяць
PostgreSQL (поточне зберігання)	10 ТБ	$0,25 за ГБ	$2500
Amazon S3 Standard (активні файли)	8 ТБ	$0,023 за ГБ	$184
Amazon S3 Glacier (архівні файли)	2 ТБ	$0,004 за ГБ	$8
Доступ до архівних файлів (Glacier)	100 запитів	$0,01 за запит	$1
Передача даних у S3 (одноразово)	10 ТБ	$0,09 за ГБ	$900

Загальні витрати на зберігання після перенесення в S3:

– місячна вартість зберігання в Amazon S3 становить $184 + $8 + $1 = $193;

– економія порівняно з PostgreSQL: $2500 - $193 = $2307 на місяць.

Формули для розрахунків:

а) місячна вартість зберігання у хмарі:

Місячна вартість=(Активні файли×Вартість S3 Standard)+(Архівні файли×Вартість S3 Glacier)+(Запити до архіву×Вартість запиту Glacier);

б) загальна економія на зберіганні:

Економія=Вартість PostgreSQL − Місячна вартість Amazon S3;

в) розрахунок ROI (Return on Investment):

ROI=Економія на місяць×12/Загальні витрати на проєкт×100%;

г) термін окупності:

Термін окупності=Вартість передачі даних/Економія на місяць.

Виконання розрахунків:

а) місячна вартість зберігання у S3:

Місячна вартість=(8000 ГБ×0,023)+(2000 ГБ×0,004)+(100 запитів×0,01)=184+8+1=$193;

б) економія на місяць:

Економія=$2500 − $193=$2307;

в) ROI (річний):

ROI=($2307×12)/$900×100%=3076%;

г) термін окупності:

Термін окупності=$900/$2307 ≈ 0,39 місяців ≈ 12 днів.

Отже, перенесення даних у Amazon S3 дає змогу значно зменшити витрати на зберігання даних завдяки використанню різних класів зберігання. Розрахунок показує, що проект окупається приблизно за 12 днів, а річний ROI становить понад 3,000%, що свідчить про високу економічну ефективність цього підходу.

Висновки до розділу 3

Алгоритм перенесення бінарних файлів у хмарне сховище включає контроль цілісності даних через перевірку контрольних сум і ведення логів за допомогою CloudTrail, що гарантує прозорість операцій та захист інформації на кожному етапі. Аналіз структури даних на початковому етапі дозволяє визначити обсяги, частоту використання та рівень доступу, що допомагає мінімізувати ризики і спроєктувати ефективну стратегію зберігання в хмарі.

Доцільність вибору хмарного сервісу Amazon S3 обґрунтовано його гнучкістю, безпкою та можливістю масштабування, що критично важливо для корпоративного середовища. Вибір класів зберігання, таких як S3 Standard і Glacier, дозволяє оптимізувати витрати відповідно до потреб доступності та частоти використання даних.

Підготовка та екстракція файлів є ключовим етапом для забезпечення цілісності даних і ефективного завантаження у S3. Налаштування bucket-ів із застосуванням політик доступу, шифрування та структуризації за типами даних забезпечує безпеку та зручність управління.

Централізація даних у єдиному кластері забезпечує легке управління, знижує операційні ризики та покращує відмовостійкість системи, що робить зберігання ефективним і безпечним.

Оптимізація витрат здійснюється через автоматизацію політик життєвого циклу і використання класів зберігання, що дозволяє мінімізувати витрати та підвищити ефективність використання ресурсів.

Економічний аналіз показав, що перенесення бінарних файлів у Amazon S3 окупається приблизно за 12 днів, а річний ROI перевищує 3000%, що підтверджує високу економічну доцільність методології.

Таким чином, розроблена методологія дозволяє оптимізувати зберігання бінарних файлів, зменшити навантаження на локальні сервери, підвищити продуктивність та забезпечити гнучкість і надійність управління корпоративними даними.

ВИСНОВКИ

У результаті дослідження було розроблено методологію для перенесення бінарних файлів із корпоративної бази даних PostgreSQL у хмарне сховище Amazon S3, яка забезпечує надійність, високу продуктивність та економічну ефективність системи. Основні результати показують, що використання Amazon S3 дозволяє значно знизити витрати на зберігання даних за рахунок класів зберігання (S3 Standard, Glacier) та політик життєвого циклу.

Завдяки виведенню великих бінарних файлів у хмару знижується навантаження на базу даних, що підвищує її продуктивність і прискорює обробку запитів. Також впровадження політик безпеки IAM і шифрування в Amazon S3 забезпечує конфіденційність та відмовостійкість даних, а використання AWS CloudWatch для моніторингу сприяє своєчасному реагуванню на потенційні проблєми.

Управління даними стає гнучкішим за рахунок автоматичного переміщення файлів між класами зберігання залежно від частоти доступу. Централізоване зберігання в S3 також полегшує адміністрування і покращує контроль за доступом до файлів.

Дослідження виявило потенціал для подальших напрямків розвитку методології, зокрема:

- автоматизація процесу перенесення даних через AWS Lambda та CI/CD-підходи для актуалізації даних у реальному часі;
- інтеграція з іншими сервісами AWS для швидкого перенесення великих обсягів, а також аналітичних інструментів, таких як Amazon Athena;
- використання систем резервного копіювання AWS Backup для підвищення відмовостійкості.

Подальші дослідження щодо автоматизації та інтеграції з іншими хмарними рішеннями відкривають перспективи для вдосконалення управління інформаційними ресурсами корпоративних систем.