ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО – НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Система автоматизованого відновлення після кіберінцидентів на хмарних платформах»**

на здобуття освітнього ступеня магістр
за спеціальності 123 Комп'ютерна інженерія
(код,найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

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

_____ ___ Руслан САРАФАНЮК _____
(підпис) (ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр.КСДМ-61
_____Руслан САРАФАНЮК_____
(ім'я, ПРІЗВИЩЕ)
Керівник: _____Артем АНТОНЕНКО_____
к.т.н,доцент, (ім'я, ПРІЗВИЩЕ)

Рецензент:_____
(ім'я, ПРІЗВИЩЕ)

Київ – 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Комп'ютерної інженерії
Ступінь вищої освіти «Магістр»
Напрям підготовки 123 «Комп'ютерна інженерія»
Освітньо-професійна програма Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерної інженерії
Наталія ЛАЩЕВСЬКА
(ім'я, ПРІЗВИЩЕ)
"___" _____ 20__ року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Сарафанюку Руслану Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Система автоматизованого відновлення після кіберінцидентів на хмарних платформах»
Керівник роботи Антоненко Артем Васильович – кандидат технічних наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 15.10.24 року № 320 .

2. Строк подання студентом роботи 20.12.2024 року
3. Вхідні дані до роботи:
   3.1. Вимоги до кваліфікаційної роботи магістра з актуальних завдань.
   3.2 Нормативні матеріали (стандарти).
   3.3 Технічні вимоги.
   3.4 Науково-технічна література з питань, пов'язаних з темою роботи.

. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   4.1 Аналіз існуючих кіберінцидентів та методів відновлення;
   4.2 Автоматизовані системи відновлення;
   4.3 Опис автоматищованої системи та алгоритм відпрацювання;
   4.4 Оцінка ефективності та подальший розвиток системи;
   4.4 Висновки

5. Графічні матеріали.
   1. Презентація

6. Дата видачі завдання 01.09.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	05.09	Виконано
2	Аналіз існуючих кіберінцидентів та методів відновлення	12.09	Виконано
3	Огляд технологій та методів автоматизованого відновлення	15.09	Виконано
4	Огляд методів захисту резервних копій систем	19.10	Виконано
5	Оцінка ефективності та подальший розвиток систем	24.10	Виконано
6	Вступ, висновки, реферат	30.10	Виконано
7	Попередній захист роботи	11.12	Виконано
8	Подання роботи в деканат	23.12	Виконано

Студент _____ Сарафанюк Р.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Антоненко А.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 96 с., 2 табл., 14 рис., 1 дод., 22 джерела.

Мета роботи: оптимізувати процеси відновлення після кіберінцидентів з використанням хмарних технологій для забезпечення мінімального часу простою та втрат даних.

Об’єкт дослідження: процес автоматизованого відновлення систем.

Предмет використання: автоматизація процесів відновлення за допомогою хмарних технологій.

З розвитком кіберзлочинності та збільшенням кількості кіберінцидентів, автоматизовані системи відновлення стають критично важливими для сучасних організацій. Оскільки значно посилюється з урахуванням стрімкого зростання кількості кіберінцидентів в Україні, особливо у 2022 році. За даними, кількість кібератак на Україну зросла майже втричі порівняно з попередніми роками, досягнувши понад 4500 інцидентів. Більшість атак організовані хакерськими групами, що мають прямі зв’язки з силовими структурами Росії. Ці угруповання націлюються на критичну інфраструктуру, урядові та комерційні установи, а також енергетичний сектор.

Методи дослідження: для досягнення мети були застосовані аналітичний метод, моделювання сценаріїв відновлення, порівняльний аналіз існуючих рішень, а також експериментальне впровадження розробленої системи у хмарному середовищі.

ABSTRACT

Text part of a bachelor's thesis: 96 p., 2 tables, 14 figures, 1 appendix, 22 sources.

Purpose: to optimise cyber incident recovery processes using cloud technologies to ensure minimal downtime and data loss.

Object of study: the process of automated system recovery.

Subject of application: automation of recovery processes using cloud technologies.

With the development of cybercrime and the increase in the number of cyber incidents, automated recovery systems are becoming critical for modern organisations. This is especially true given the rapid increase in the number of cyber incidents in Ukraine, especially in 2022. According to data, the number of cyberattacks on Ukraine has almost tripled compared to previous years, reaching more than 4,500 incidents. Most of the attacks were organised by hacker groups with direct ties to Russia's security forces. These groups target critical infrastructure, government and commercial institutions, and the energy sector.

Research Methods: To achieve the goal, the analytical method, modelling of recovery scenarios, comparative analysis of existing solutions, and experimental implementation of the developed system in the cloud environment were used.

ЗМІСТ

ВСТУП

З розвитком інформаційних технологій кібербезпека стала однією з найважливіших сфер, що потребує постійної уваги з боку державних та приватних організацій. Особливо критичним цей аспект став для України, де у 2022 році кількість кібератак зросла майже втричі, досягнувши понад 4500 інцидентів. Більшість з цих атак здійснюються хакерськими угрупуваннями, що підтримуються силовими структурами Російської Федерації, і вони спрямовані на пошкодження критичної інфраструктури, урядових органів та комерційних організацій. [1]

Особливо гострою є проблема атак на енергетичну інфраструктуру України, що може призводити до катастрофічних наслідків для економіки та безпеки країни. Використання автоматизованих систем відновлення після таких інцидентів, зокрема з використанням хмарних технологій, здатне забезпечити швидке реагування та мінімізувати негативні наслідки. Це робить дослідження в цій сфері особливо актуальним і важливим для посилення кіберстійкості на національному рівні.

Завданням сучасних організацій є не лише захист від таких атак, але й швидке та ефективне відновлення після кіберінцидентів. Традиційні методи відновлення часто вимагають значних ресурсів і часу, що може призвести до значних фінансових втрат та порушень безперервності бізнесу. Зважаючи на це, використання автоматизованих систем відновлення, особливо з використанням хмарних технологій, набуває особливої актуальності. Такі системи дозволяють значно скоротити час відновлення, мінімізувати втрати даних та забезпечити безперервність бізнес-процесів, що є критично важливим у сучасних умовах.

Відновлення інфраструктури після кібератаки є складним через те, що атакуючі часто впроваджують складні та приховані механізми повторної активації шкідливого ПЗ, що ускладнює повне очищення системи. Крім того, під час внутрішніх атак зловмисники можуть мати доступ до критичних ресурсів і розуміння внутрішніх процесів, що дозволяє їм уникати виявлення на початкових етапах і підвищує складність відновлення.

Метою цієї роботи є дослідження автоматизованих систем відновлення після кіберінцидентів з використанням хмарних технологій. У роботі буде проведено аналіз існуючих рішень, розроблення концепції автоматизованої системи відновлення на основі існуючих рішеннях. Це дослідження дозволить створити ефективне рішення для забезпечення стійкості інформаційних систем та мінімізації наслідків кібератак.

РОЗІДЛ 1 АНАЛІЗ ІСНУЮЧИХ КІБЕРІНЦИДЕНТІВ ТА МЕТОДІВ ВІДНОВЛЕННЯ

1.1 Типи кібератак і їх вплив на організації

Кібератаки є однією з найбільших загроз для сучасних організацій. Кількість кібератак зростає з кожним роком, а методи атак стають все більш витонченими, що призводить до серйозних наслідків для бізнесу та організацій. Від втрати даних до порушення бізнес-процесів - такі інциденти можуть мати руйнівні наслідки для будь-якої організації. За даними аналітиків, середній час, необхідний для виявлення та реагування на серйозний кіберінцидент, може тривати місяцями, а середня вартість такого інциденту для організації обчислюється мільйонами доларів.

Автоматизація процесу відновлення стає ключовим елементом забезпечення стійкості до кіберзагроз. Традиційні методи, які часто передбачають ручну роботу та застарілі процедури відновлення, більше не відповідають вимогам сучасного світу. Вони не можуть відновлюватися достатньо швидко і не можуть ефективно протистояти новітнім методам атак.

Поява хмарних обчислень призвела до значних змін у підходах до відновлення після кіберінцидентів. Використання хмарних платформ для автоматизації процесів резервного копіювання, виявлення загроз і відновлення даних дозволяє значно підвищити швидкість і надійність таких систем. Хмарні технології надають масштабовані рішення, які дозволяють організаціям швидко адаптуватися до нових викликів і мінімізувати втрати.

Метою цього розділу є аналіз існуючих рішень для відновлення даних після кіберінцидентів, порівняння їх ефективності, виявлення переваг та недоліків автоматизованих підходів та визначення ролі хмарних технологій у цьому процесі. Особливу увагу буде приділено використанню основних хмарних платформ, таких як AWS, Azure та Google Cloud, які активно розвивають рішення для автоматизації відновлення систем після кіберінцидентів.

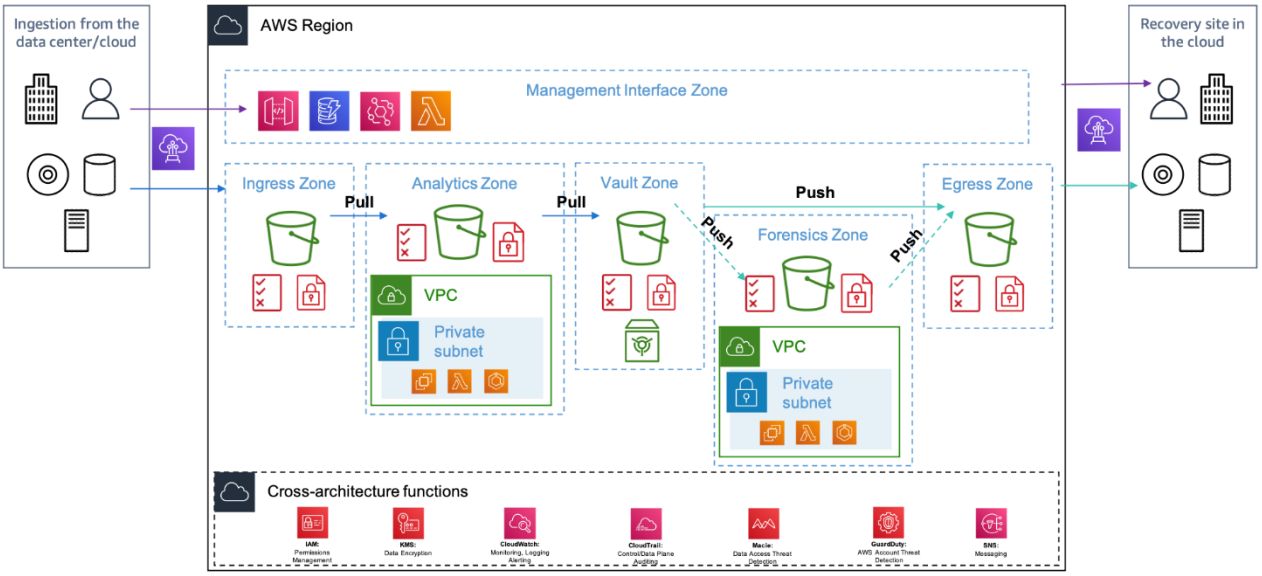

Рисунок 1.1 – Відновлення після кіберподій у сфері фінансових послуг

Зображення вище ілюструє складність сучасних кіберзагроз і підкреслює важливість використання автоматизованих рішень для відновлення систем після інциденту. Хмарні технології, в тому числі рішення від провідних постачальників, можуть відігравати ключову роль у прискоренні та спрощенні процесу відновлення.

Кібератака – це спроба викрасти, змінити, знищити, порушити або вивести з ладу інформаційні ресурси або системи в комп'ютерній мережі. Кібератаки можна розділити на дві категорії: внутрішні та зовнішні загрози. Внутрішні загрози виникають від людей, які мають законний доступ до систем і використовують вразливості, свідомо чи випадково використовуючи свої права доступу. Доступ до систем організації можуть отримати незадоволені працівники або підрядники. Зовнішні загрози надходять від осіб, не пов'язаних з системою, на яку спрямована атака, наприклад, від злочинних угруповань або хакерів.

Кіберзлочинність полягає в ефективній експлуатації вразливостей, що завжди ставить команди безпеки в невигідне становище. Це пов'язано з тим, що команді безпеки доводиться захищатися від усіх можливих точок входу, в той час як зловмиснику достатньо знайти і використати лише одну слабкість або вразливість. Така асиметрія грає на руку зловмиснику. Як наслідок, навіть великі компанії намагаються перешкодити кіберзлочинцям монетизувати доступ до їхніх

мереж. Зазвичай, доступ до мережі та комунікації повинні бути відкритими, а фахівці з безпеки намагаються захистити ресурси компанії.

Однак кібератаки становлять загрозу не лише для великих компаній. Кіберзлочинці використовують будь-який пристрій, підключений до інтернету, як зброю, мішень або і те, і інше, а малі та середні підприємства, як правило, мають менш досконалі заходи кібербезпеки, що робить їх вразливими до потенційних інцидентів безпеки.

Розглянемо 16 найшкідливіших кібератак і те, як вони працюють:

1) Атака шкідливого програмного забезпечення

Шкідливе програмне забезпечення - це загальний термін, що позначає ворожі або нав'язливі програми чи файли, призначені для використання пристрою на шкоду користувачеві та на користь зловмисника. Шкідливе програмне забезпечення буває різних форм, використовуючи хитромудрі та заплутані методи, призначені не лише для обману користувачів, але й для того, щоб обійти засоби контролю безпеки та встановити на системи та пристрої без дозволу.

Найбільше побоювань викликають програми-вимагачі - програмне забезпечення, в якому зловмисник шифрує файли жертви і вимагає сплатити викуп, щоб отримати ключ для розшифрування. Програми-здирники широко поширені і більш детально розглядаються в окремому розділі нижче. До інших поширених типів шкідливих програм належать:

- Руткіти. На відміну від інших шкідливих програм, руткіт - це набір програмних інструментів, які використовуються для відкриття чорного ходу на пристрої жертви. Це дозволяє зловмисникам встановлювати додаткові шкідливі програми, такі як програми-вимагачі або клавіатурні шпигуни, а також отримувати віддалений доступ до інших пристроїв у мережі та керувати ними. Щоб уникнути виявлення, руткіти часто відключають захисне програмне забезпечення. Отримавши контроль над пристроєм, руткіт може використовувати його для розсилки спаму, приєднання до бот-мереж або збору конфіденційних даних і відправки їх назад зловмиснику.

- Троянський кінь Троянський кінь - це програма, яка завантажується та

встановлюється на комп'ютер і виглядає нешкідливою, але насправді є шкідливою. Таке шкідливе програмне забезпечення зазвичай ховається у на перший погляд нешкідливих вкладеннях електронної пошти або безкоштовних файлах для завантаження. Коли користувач натискає на вкладення або завантажує програму, шкідливе програмне забезпечення переноситься на комп'ютерний пристрій користувача. Потрапивши всередину, шкідливий код виконує завдання, розроблені зловмисником. У багатьох випадках це призводить до негайного запуску атаки, але також може створювати чорний хід, який хакери можуть використовувати для майбутніх атак.

- Шпигунські програми. Після встановлення шпигунське програмне забезпечення відстежує дії жертви в Інтернеті, відстежує дані для входу в систему та викрадає конфіденційну інформацію. Наприклад, кіберзлочинці використовують шпигунські програми для отримання номерів кредитних карток, банківських рахунків і паролів. Урядові установи багатьох країн також використовують шпигунські програми (особливо програму Pegasus) для стеження за активістами, політиками, дипломатами, блогерами, дослідницькими центрами та союзниками.

2) Атака програм-вимагачів

Програми-здирники зазвичай встановлюються, коли користувач відвідує шкідливий веб-сайт або відкриває фальшивий вкладення електронної пошти. Зазвичай він використовує вразливості зараженого пристрою, щоб зашифрувати та вимкнути важливі файли, такі як документи Word, таблиці Excel, PDF, бази даних та системні файли. Потім зловмисник вимагає викуп в обмін на ключ розшифровки, необхідний для відновлення заблокованих файлів. Атаки можуть бути спрямовані на критичні сервери або на інші пристрої, підключені до мережі, перед запуском процесу шифрування, і намагатися вплинути на них одночасно.

Щоб посилити тиск на жертву, зловмисники часто погрожують продати або опублікувати викрадені під час атаки дані, якщо викуп не буде сплачено. Дійсно, зміни в тактиці зловмисників призвели до того, що деякі з них змушують жертву заплатити викуп, не намагаючись зашифрувати дані, покладаючись виключно на можливість крадіжки та публічного розголошення даних. Ці зміни можуть бути

причиною рекордної кількості атак зловмисників-здирників у 2023 році, про які повідомляють постачальники послуг з кібербезпеки та дослідники. Згідно з дослідженням Check Point Research, 10% організацій у всьому світі стали мішенню для програм-вимагачів.

Від приватних осіб і малого та середнього бізнесу до великих корпорацій і державних установ - будь-хто може стати мішенню вірусу-здирника. Атаки можуть мати серйозні руйнівні наслідки. Відомим прикладом є атака вірусу-здирника WannaCry у 2017 році, від якої постраждали організації у понад 150 країнах світу, а Національна служба охорони здоров'я Великої Британії зазнала збитків у розмірі близько 111 мільйонів доларів США через закриття лікарень. Зовсім недавно, у 2023 році, Королівська пошта Великобританії стала жертвою атаки вірусу-здирника, який зашифрував важливі файли і зупинив міжнародну доставку на шість тижнів. Королівська пошта відмовилася платити початкову вимогу викупу у 80 мільйонів доларів США або подальші зменшення, але заявила, що витратила майже 13 мільйонів доларів США на відновлення та покращення безпеки. Крім того, дані, викрадені під час атаки, були опубліковані в Інтернеті.

Аналогічно, у 2023 році атака на MGM Resorts International з використанням програмного забезпечення з вимогою викупу коштувала компанії готелів і казино близько 100 мільйонів доларів США, порушила її діяльність і призвела до викрадення персональних даних клієнтів. За даними Wall Street Journal, компанія Caesars Entertainment, яка зазнала подібної атаки, погодилася заплатити викуп у розмірі 15 мільйонів доларів США, щоб запобігти публікації викрадених даних в Інтернеті. Програми-вимагачі є настільки серйозною проблемою, що уряд США у 2021 році запустив веб-сайт під назвою StopRansomware, який надає ресурси, що допомагають організаціям запобігати атакам, а також контрольний список для реагування на них.

3) Атаки на паролі

Незважаючи на численні відомі недоліки, паролі залишаються найпоширенішим методом автентифікації для комп'ютерних сервісів, тому отримання пароля жертви - це простий спосіб обійти контроль безпеки та отримати

доступ до критично важливих даних і систем. Зловмисники отримують паролі шахрайським шляхом різними способами, зокрема:

- Атаки грубої сили. Зловмисники підбирають облікові дані користувача методом проб і помилок, використовуючи відомі паролі, такі як password123, або паролі, засновані на інформації, зібраній з постів жертви в соціальних мережах, наприклад, ім'я домашнього улюбленця. У деяких випадках вони також використовують автоматизовані інструменти для злому паролів, щоб спробувати всі можливі комбінації символів.
- Словникові атаки. Подібно до атак грубої сили, словникові атаки використовують заздалегідь підібрану бібліотеку загальновживаних слів і фраз, залежно від місцезнаходження та національності жертви.
- Соціальна інженерія. Зловмисники можуть легко створювати персоналізовані електронні листи та текстові повідомлення, які виглядають справжніми, збираючи інформацію про людей з постів у соціальних мережах та інших джерел. Як форма соціальної інженерії, ці повідомлення, особливо якщо вони надіслані з фейкового акаунта, який видає себе за когось із знайомих жертви, можуть бути використані для отримання облікових даних для входу в систему під хибним приводом, щоб маніпулювати ними або обманом змусити їх розкрити інформацію.
- Кейлоггери. Кейлоггер - це програма, яка таємно відстежує і записує кожне натискання користувачем клавіш з метою перехоплення паролів, PIN-кодів та іншої конфіденційної інформації, що вводиться через клавіатуру. Потім ця інформація надсилається зловмиснику через інтернет.
- Винюхування паролів. Перехоплювачі паролів - це невеликі програми, встановлені в мережі, які витягують імена користувачів і паролі, що надсилаються мережею у вигляді простого тексту. Зловмисники все ще використовують цей метод, але він вже не становить такої загрози, як раніше, оскільки більшість мережевого трафіку зараз шифрується.
- Крадіжка або купівля бази даних паролів. Хакери можуть спробувати зламати мережеву безпеку організації, щоб викрасти базу даних облікових даних

користувачів і використовувати ці дані самостійно або продати їх іншим.

4) DDoS-атака

Розподілена атака на відмову в обслуговуванні (DDoS) використовує велику кількість скомпрометованих комп'ютерних систем або мобільних пристроїв для атаки на сервери, веб-сайти або інші мережеві ресурси. Її мета - уповільнити або повністю вивести їх з ладу, надсилаючи потік повідомлень, запитів на з'єднання або неправильно сформованих пакетів, тим самим відмовляючи в обслуговуванні легальним користувачам.

Постачальник програмного забезпечення для підвищення продуктивності та безпеки Netscout повідомляє, що в першій половині 2023 року було здійснено майже 7,9 мільйона DDoS-атак, що на 31% більше, ніж у минулому році. Хоча багато з цих атак є політично або ідеологічно мотивованими, вони також використовуються з метою викупу. У деяких випадках зловмисники погрожують DDoS-атакою на організацію, якщо вимога викупу не буде виконана. Зловмисники також використовують можливості інструментів штучного інтелекту для вдосконалення своїх методів атак і дають вказівки мережам підлеглих машин здійснювати DDoS-атаки. Викликає занепокоєння той факт, що ШІ наразі використовується для посилення всіх форм кібератак, але також може застосовуватися і для кібербезпеки.

5) Фішинг

Під час фішингу зловмисники видають себе за організацію або особу, якій довіряють, і обманом змушують нічого не підозрюючих жертв передати цінну інформацію, таку як паролі, дані кредитних карток або інтелектуальну власність. Фішингові кампанії, засновані на методах соціальної інженерії, прості у виконанні та напрочуд ефективні. Хоча електронна пошта найчастіше використовується для доставки шкідливих посилань і вкладень, фішингові атаки також можуть здійснюватися за допомогою текстових повідомлень (SMS-фішинг) і телефонних дзвінків (голосовий фішинг).

Списовий фішинг націлений на конкретних осіб або компанії, тоді як китобійні атаки - це різновид списового фішингу, який націлений на керівників

організацій. Подібні атаки включають в себе шахрайство з діловою електронною поштою (BEC), коли зловмисники видають себе за топ-менеджерів або інших уповноважених осіб і просять співробітників надіслати гроші, придбати подарункові картки і т.д. Центр скарг на інтернет-злочинність ФБР класифікує фішингові атаки та BEC-атаки в окремі категорії. Центр ФБР зі скарг на інтернет-злочинність класифікує фішингові та BEC-атаки в окремі категорії. У 2022 році, останньому році, за який опубліковані дані, було зареєстровано 21 832 скарги на BEC-атаки із загальним збитком понад 2,7 мільярда доларів США, а також 300 497 скарг на фішинг із загальним збитком 52 мільйони доларів США.

6) Атаки SQL-ін'єкцій

Веб-сайти, які використовують бази даних, тобто більшість веб-сайтів, вразливі до атак SQL-ін'єкцій - SQL-запит - це запит до бази даних на виконання певної дії, і добре сконструйований зловмисний запит може створювати, змінювати або видаляти дані, що зберігаються в базі даних. Вони також можуть читати і витягувати такі дані, як інтелектуальна власність, персональні дані клієнтів і співробітників, облікові дані керівництва та особисту ділову інформацію.

SQL-ін'єкції залишаються широко використовуваним вектором атак: вони посідають третє місце серед найнебезпечніших вразливостей програмного забезпечення25 згідно зі звітом Common Weakness Enumeration (CWE) 2023, який ведеться компанією The Mitre Corp.CVEdetails За даними .com, у 2023 році до бази даних CVE, окремого каталогу поширених уразливостей, який також веде Mitre.com, було додано понад 2100 SQL-ін'єкцій. Відомими прикладами SQL-ін'єкцій є використання зловмисником однієї з цих нових уразливостей для отримання доступу до веб-додатку MoveIt Transfer від Progress Software, що призвело до витоку даних у тисячах організацій, які використовують це програмне забезпечення для передачі файлів.

7) Міжсайтовий скриптинг

Це ще один тип ін'єкційних атак, коли зловмисник додає шкідливі скрипти до вмісту легітимного веб-сайту. Атаки на міжсайтовий скриптинг (XSS) відбуваються, коли ненадійне джерело впроваджує код у веб-додаток, а шкідливий

код динамічно генерується і включається у веб-сторінки, що доставляються до браузера жертви. Це дозволяє зловмисникам виконувати скрипти, написані на таких мовах, як JavaScript, Java і HTML, у браузерах нічого не підозрюючих користувачів веб-сайтів.

Зловмисники можуть використовувати XSS для викрадення сесійних файлів cookie та видавати себе за жертву. XSS також може використовуватися для розповсюдження шкідливого програмного забезпечення, пошкодження веб-сайтів, викрадення облікових даних користувачів та інших шкідливих дій. Часто це поєднується з методами соціальної інженерії, такими як фішинг.

8) Атака “людина посередині”

Під час атаки «людина посередині» (MitM) зловмисник таємно перехоплює комунікацію між двома сторонами (наприклад, кінцевим користувачем і веб-додатком). Легітимні сторони вважають, що вони спілкуються безпосередньо одна з одною, але насправді зловмисник вступає в середину електронної розмови і контролює її. Зловмисник може прочитати, скопіювати або змінити повідомлення, включаючи дані, що містяться в ньому, перш ніж переслати його нічого не підозрюючому одержувачу.

Успішні MitM-атаки дозволяють зловмисникам отримувати або маніпулювати конфіденційною особистою інформацією, такою як облікові дані для входу в систему, деталі транзакцій, записи про рахунки та номери кредитних карток. Такі атаки часто націлені на користувачів банківських додатків і веб-сайтів електронної комерції, часто з використанням фішингових листів, щоб заманити користувачів до встановлення шкідливого програмного забезпечення, яке уможливлює атаку.

9) Інтерпретація URL-адрес/отруєння URL-адрес

Зловмисник може легко змінити URL-адресу для доступу до інформації або ресурсів. Наприклад, зловмисник може увійти в обліковий запис користувача, створений на веб-сайті, і легко змінити його на “https://www.workwebsite.com/acount?user=9210”,він може легко змінитиURL-адресу, скажімо, на URL “https://www. workwebsite.com/account?user=4892”, щоб

перевірити, чи зможе він отримати доступ до налаштувань облікового запису цього користувача. Якщо веб-сервер сайту не гарантує, що кожен користувач має відповідні дозволи на доступ до запитуваного ресурсу, особливо якщо він містить дані, надані користувачем, зловмисник з більшою ймовірністю зможе побачити налаштування облікових записів інших користувачів на сайті.

Атаки на інтерпретацію URL-адрес, також відомі як отруєння URL-адрес, можуть бути використані для збору конфіденційної інформації, наприклад, імен користувачів і записів у базі даних, або для доступу до сторінок адміністратора для керування сайтом. Якщо зловмисник може маніпулювати URL-адресами для доступу до привілейованих ресурсів, це зазвичай відбувається через уразливість прямого посилання на об'єкт, коли сайт не застосовує належним чином перевірки контролю доступу для підтвердження особи користувача.

10) Підробка DNS

DNS дозволяє користувачам отримувати доступ до веб-сайтів, зіставляючи доменні імена та URL-адреси з IP-адресами, які комп'ютери використовують для пошуку сайтів. Хакери вже давно використовують незахищену природу DNS, щоб перезаписувати збережені IP-адреси на DNS-серверах і роздільниках фальшивими записами, щоб жертви потрапляли на підконтрольний зловмисникам веб-сайт замість легітимного. Ці фальшиві сайти розроблені так, щоб виглядати точно так само, як сайти, які користувачі очікували відвідати. Як наслідок, жертви атаки підміни DNS не викликають підозр, коли їх просять ввести облікові дані для входу на справжньому, на їхню думку, сайті. Ця інформація дозволяє зловмисникам увійти в облікові записи користувачів на підроблених сайтах.

11) Тунелювання DNS

Оскільки DNS є довіреною службою, DNS-повідомлення зазвичай проходять через брандмауер організації в обох напрямках без особливого контролю. Однак це означає, що зловмисники можуть обійти засоби контролю безпеки, вбудовуючи командні повідомлення або інші шкідливі дані в запити та відповіді DNS. Наприклад, OilRig, хакерська група, яка підозрюється у зв'язках з Іраном, використовує DNS-тунелювання для підтримки зв'язку між серверами управління

та системами, що атакуються.

Атаки DNS-тунелювання використовують тунельне шкідливе програмне забезпечення, розгорнуте на веб-сервері із зареєстрованим доменним ім'ям. Після того, як зловмисник заражає комп'ютер за брандмауером організації, шкідливе програмне забезпечення, встановлене на цьому комп'ютері, намагається підключитися до сервера за допомогою програми тунелювання, яка використовує DNS-запити для визначення його місцезнаходження. Це дозволяє зловмиснику підключитися до захищеної мережі.

Тунелювання DNS також може використовуватися для інших цілей. Наприклад, виробники антивірусного програмного забезпечення надсилають оновлення профілю шкідливого програмного забезпечення у фоновому режимі через DNS-тунелювання. Тому DNS-трафік потрібно контролювати, щоб гарантувати, що через мережу може проходити лише довірений трафік.

12) Ботнет-атака

Ботнет - це група підключених до Інтернету комп'ютерів і мережевих пристроїв, які заражені шкідливим програмним забезпеченням і віддалено контролюються кіберзлочинцями. Зловмисники також компрометують вразливі пристрої Інтернету речей, збільшуючи розмір і потужність ботнету. Ботнети часто використовуються для розсилки електронного спаму, участі в кампаніях клікового шахрайства та генерування шкідливого трафіку для DDoS-атак.

Наприклад, коли у 2021 році було виявлено ботнет Meris, дослідники безпеки Cloudflare повідомили, що зловмисники використовували ботнет Meris для запуску DDoS-атак на близько 50 різних веб-сайтів щодня. Meris також відповідальний за одні з найбільших DDoS-атак в історії завдяки використанню HTTP-конвеєра та оціночному розміру близько 250 000 ботів до 2021 року. Метою бот-мереж є зараження якомога більшої кількості пристроїв і використання об'єднаних обчислювальних потужностей і ресурсів цих пристроїв для автоматизації та масштабування своєї шкідливої діяльності.

13) Атака “водопою”

У так званій drive-by атаці зловмисник використовує вразливість системи

безпеки для додавання шкідливого коду на легітимний веб-сайт, і коли користувач відвідує цей сайт, код автоматично виконується і заражає його комп'ютер або мобільний пристрій. Це тип атаки «водопою», коли зловмисники знаходять і використовують незахищені веб-сайти, які відвідують користувачі, яких вони хочуть атакувати (наприклад, працівники або клієнти певної організації, або цілі сектори, такі як фінансовий, медичний або військовий).

Оскільки користувачам важко виявити, які саме веб-сайти були скомпрометовані в результаті атаки “водопою”, це дуже ефективний спосіб інсталяції шкідливого програмного забезпечення на пристрій. Оскільки потенційні жертви довіряють веб-сайту, зловмисники можуть навіть приховати шкідливе програмне забезпечення у файлах, які користувачі навмисно завантажують. Шкідливе програмне забезпечення для атаки “водопою” часто являє собою троянську програму з віддаленим доступом, яка дозволяє зловмисникам отримати віддалений контроль над зараженими системами.

14) Внутрішня загроза

Працівники та підрядники можуть мати законний доступ до систем організації, деякі з них мають глибоке розуміння засобів захисту кібербезпеки. Цим можуть скористатися зловмисники, щоб отримати доступ до обмежених ресурсів, внести зловмисні зміни в конфігурацію системи або встановити шкідливе програмне забезпечення. Інсайдери також можуть ненавмисно спричинити проблеми через недбалість, недостатню обізнаність або недостатню підготовку щодо політик і передових практик кібербезпеки.

Колись вважалося, що інциденти, пов'язані з внутрішніми загрозами, переважають зовнішні атаки, але це вже не так. Згідно зі звітом Verizon про витоки даних за 2023 рік, понад 80% розслідуваних порушень були спричинені зовнішніми сторонами. Однак внутрішні сторони були причетні до 19% з них, або майже до кожного п'ятого. Деякі з найгучніших витоків даних були здійснені внутрішніми сторонами, які мали доступ до привілейованих облікових записів. Наприклад, Едвард Сноуден, підрядник Агентства національної безпеки з доступом до адміністративного облікового запису, стояв за одним з найбільших витоків

секретної інформації в історії США, починаючи з 2013 року.У 2023 році член повітряної національної гвардії штату Массачусетс був заарештований за публікацію надсекретних військових документів в Інтернеті і притягнутий до кримінальної відповідальності за розміщення надсекретних військових документів в Інтернеті.

15) Атака на підслуховування

Атаки підслуховування, також відомі як мережевий сніфінг або перехоплення пакетів, використовують незахищені комунікації для перехоплення трафіку в реальному часі, що надсилається мережею комп'ютерами та іншими пристроями. Для пасивного моніторингу та запису інформації, а також підслуховування незашифрованих даних з мережевих пакетів можуть використовуватися апаратні, програмні засоби або їх поєднання. Підслуховування мережі - це законна діяльність, яку здійснюють мережеві адміністратори та команди IT-безпеки для усунення мережевих проблем і перевірки трафіку. Однак зловмисники можуть використовувати такі дії для викрадення конфіденційних даних або отримання інформації для подальшого проникнення в мережу.

Під час атаки на прослуховування шкідливе програмне забезпечення може бути встановлене на пристрої, підключені до мережі, за допомогою фішингових електронних листів, або зловмисник може підключити до системи апаратне забезпечення. Атака не вимагає постійного підключення до скомпрометованого пристрою, а захоплені дані можна відновити пізніше, або фізично, або за допомогою віддаленого доступу. Виявити сніфінг-атаки може бути складно через складність сучасних мереж і велику кількість підключених пристроїв.

16) Атака на день народження

Це різновид криптографічної атаки грубої сили, в якій цифрові підписи, паролі та ключі шифрування отримують шляхом зіставлення хеш-значень, що використовуються для їх представлення. Він заснований на «парадоксі дня народження», який стверджує, що у випадковій групі з 23 осіб ймовірність того, що дві людини мають однаковий день народження, перевищує 50%. Подібна логіка може бути застосована до хеш-значень, щоб уможливити атаки на день

народження.

Важливою властивістю хеш-функцій є їхня стійкість до колізій: надзвичайно важко згенерувати однакове хеш-значення з двох різних вхідних даних. Однак, якщо зловмисник генерує тисячі випадкових вхідних даних і обчислює їх хеш-значення, особливо якщо хеш-функція слабка або пароль короткий, шанси зіставити викрадені значення і знайти облікові дані користувача зростають. Такі атаки також можуть бути використані для створення фальшивих повідомлень і підробки цифрових підписів. Тому розробники повинні використовувати надійні криптографічні алгоритми і методи, стійкі до атак на день народження, такі як коди автентифікації повідомлень і коди автентифікації повідомлень на основі хеш-функції.

1.1.1 Статистика кіберінцидентів

З початку 2021 року в Україні зареєстрували майже 14 млн інцидентів у сфері кібербезпеки. Однак масштабних кіберзагроз у перший місяць року не зафіксували.

Більшість атак (75%) з початку року була спрямована на державний сектор. Найбільша кількість кіберінцидентів була пов'язана з:

- несанкціонованим доступом до комп’ютерних систем - 70%;
- скануванням ресурсів - 10%;
- здійсненням атак типу brute-force - 7%. [2]

Також у січні зафіксували понад 400 тисяч фішингових атак. Зокрема, було виявлено масову розсилку електронних листів по державних установах нібито від адміністрації Держспецзв’язку.

Зазначається, що у більшості випадків вдалося уникнути негативних наслідків, але в деяких держустановах зловмисникам все ж вдалося отримати доступ до уражених комп’ютерів. [2]

З аналізу даних кіберзлочинів в Україні на початку 2021 року можна зробити декілька важливих висновків:

1) Масштаб кібератак: Протягом перших місяців 2021 року в Україні було

зареєстровано майже 14 мільйонів інцидентів у сфері кібербезпеки, що свідчить про значний рівень активності кіберзлочинців у країні. Це велика кількість інцидентів для такої короткої періоду, що вказує на зростання загроз у цифровому середовищі.

2) Ціль державного сектору: 75% усіх атак були спрямовані на державний сектор, що підкреслює вразливість державних установ перед кіберзлочинцями. Це також свідчить про те, що державний сектор є пріоритетною мішенню через чутливу інформацію, яку він обробляє.

3) Типи атак:

- Несанкціонований доступ: Близько 70% інцидентів були пов'язані з несанкціонованим доступом до комп'ютерних систем, що є однією з найпоширеніших форм кіберзлочинності. Це загрожує зловмисниками отримати повний контроль над системами та їхніми даними.

- Сканування ресурсів (10%) і атаки типу brute-force (7%): Такі типи атак вказують на те, що кіберзлочинці активно намагаються знайти вразливості в системах для подальшого проникнення.

4) Фішингові атаки: Понад 400 тисяч фішингових атак було зафіксовано лише у січні, що демонструє масштабність цієї загрози. Особливо варто зазначити, що такі атаки часто спрямовані на державні установи з метою доступу до конфіденційних даних або контролю над системами.

5) Наслідки атак: Хоча в більшості випадків вдалося уникнути серйозних наслідків, деякі з атак все ж призвели до успішного проникнення зловмисників у державні системи, що підкреслює необхідність покращення заходів безпеки та підвищення кіберзахисту.

Загалом ці дані вказують на високу активність кіберзлочинців і важливість посилення кібербезпеки, особливо для державних установ. Інтенсивність кібератак свідчить про постійний тиск на державні інфраструктури, і це вимагає від України стратегічного підходу до захисту критичних інформаційних ресурсів.

На основі даних, представлених у матеріалі, можна зробити кілька ключових висновків щодо стану кіберзагроз в Україні у 2022 році, а також про зміни в кількості та природі атак у порівнянні з 2021 роком.

Основні тенденції у 2022 році:

1) Значне зростання кількості кібератак:

- У 2022 році в Україні було зафіксовано у 2,8 рази більше кіберінцидентів, ніж у 2021 році, що є суттєвим збільшенням. Це зростання пов'язане з агресією з боку Росії, яка посилила свої кібератаки після повномасштабного вторгнення в Україну.

- Загалом на інформаційну інфраструктуру України було здійснено 7000 кібератак протягом року. Кількість атак, географічно пов’язаних із РФ, збільшилася на 26%, що відображає агресивні дії з боку проросійських угруповань.

2) Найбільш постраждалі сектори:

- Фінансовий сектор: 120 інцидентів були спрямовані на фінансові установи, що становить 5% від усіх атак. Хоча фінансовий сектор не є основною мішенню, його важливість для національної економіки робить його вразливим до кібератак.

- Комерційний сектор: Близько 156 інцидентів зачепили комерційні організації.

- Телекомунікаційні компанії та розробники ПЗ: 92 інциденти були зафіксовані в секторі телекомунікацій та розробки програмного забезпечення, що становить ще одну важливу групу цілей.

3) Фішинг як основний вектор атак:

- Більшість зловмисників отримували доступ до систем через таргетовані фішингові кампанії. Це підкреслює важливість захисту від соціальної інженерії та підвищення обізнаності користувачів щодо потенційних загроз.

4) Типи атак і методи:

- Зловмисники використовували шкідливе програмне забезпечення для шпигунства, збору даних та експлуатації вразливостей систем. Ці методи дозволяють хакерам втручатися в роботу систем, не залишаючи очевидних слідів, що робить такі атаки складними для виявлення.

5) Україна на глобальній арені кіберінцидентів:

- Україна посіла друге місце серед найбільш атакованих країн світу після

США. Це говорить про те, що Україна стала однією з головних мішеней для кіберзлочинців, зокрема через війну та політичну ситуацію.

6) Зміни у 2023 році:

- На початку 2023 року кількість атак з боку проросійських хакерських угруповань дещо зменшилася, але вони залишаються систематичними та інтенсивними. CERT-UA зафіксувала 549 кіберінцидентів за перший квартал 2023 року, серед яких:
- 13 атак спрямовані на фінансовий сектор;
- 23 – на комерційні організації;
- 11 – на розробників ПЗ. [3]

Збільшення кількості кібератак у 2022 році відображає посилення кібервійни між Україною та Росією. Російські хакери активно використовують кібератаки як частину гібридної війни для підриву критичних інформаційних систем та дестабілізації ситуації в країні. Основна мішень – державні, фінансові та IT-сектори. Україна показала певний рівень кіберстійкості, але постійне збільшення атак вимагає посилення кіберзахисту та більш інтенсивного реагування на кіберінциденти.[3]

У 2023 році Департамент кіберполіції Національної поліції України досяг значних результатів у боротьбі з кіберзлочинністю, зокрема завдяки міжнародній співпраці та посиленню оперативно-розшукової діяльності.

Ключові результати:

1) Загальна кількість виявлених кіберзлочинів:

- Протягом 2023 року було виявлено понад 3600 кіберзлочинів, що свідчить про активну діяльність кіберполіції у виявленні та розслідуванні кіберзлочинів.

2) Кримінальні провадження та підозрювані:

- Оголошено підозру понад 1700 особам за вчинення понад 3700 злочинів. Це на 59% більше, ніж у 2022 році.
- Направлено до суду матеріали щодо 42 організованих злочинних груп, у тому числі 7 злочинних організацій. Порівняно з 2022 роком, цей показник

зріс на 83%.

3) Міжнародні спецоперації:

- Проведено 18 міжнародних спецоперацій у співпраці з правоохоронцями з Грузії, Швейцарії, Чехії, Ізраїлю, Норвегії, Нідерландів, Франції, Німеччини та США. Ці операції демонструють ефективність співпраці України з іншими країнами у сфері кібербезпеки.

4) Обвинувальні акти та збитки:

- За оперативного супроводу кіберполіції до суду направлено обвинувальні акти щодо понад 4000 злочинів.

- Відшкодовано майже 144 мільйони гривень збитків з урахуванням арештованого та вилученого майна.

5) Блокування ворожих вебресурсів:

- Заблоковано майже 13 тисяч "ворожих" вебресурсів, що вказує на активну роботу кіберполіції з протидії інформаційній агресії, спрямованій на дестабілізацію ситуації в Україні.

Департамент кіберполіції продемонстрував суттєвий прогрес у боротьбі з кіберзлочинністю, зокрема за рахунок збільшення кількості виявлених злочинів, успішного виявлення та переслідування організованих злочинних угруповань, а також активної міжнародної співпраці. Ці показники свідчать про системний підхід до боротьби з кіберзагрозами та розвиток стратегічного партнерства з міжнародними організаціями.[4]

1.2 Огляд традиційно ручних методів відновлення

Традиційні ручні методи відновлення після кіберінцидентів є основою стратегій кібербезпеки та відновлення багатьох організацій. Ці методи покладаються на залучення кваліфікованих фахівців і передбачають низку кроків, спрямованих на відновлення нормальної роботи IT-інфраструктури після збою або атаки. Ручні методи вимагають значних витрат часу та людських ресурсів, що може суттєво вплинути на ефективність та швидкість відновлення.

Основні етапи ручного відновлення:

1) Оцінка та діагностика інциденту

Традиційні ручні методи відновлення після кіберінцидентів є основою стратегій кібербезпеки та відновлення багатьох організацій. Ці методи покладаються на залучення кваліфікованих фахівців і передбачають низку кроків, спрямованих на відновлення нормальної роботи IT-інфраструктури після збою або атаки. Ручні методи вимагають значних витрат часу та людських ресурсів, що суттєво впливає на ефективність та швидкість відновлення.

2) Ізоляція уражених систем

Наступним кроком є ізоляція заражених мережевих компонентів і систем, щоб запобігти поширенню шкідливого програмного забезпечення. Це часто передбачає вимкнення серверів, мережевих пристроїв або кінцевих точок, поки експерти проводять розслідування. Ізоляція мінімізує поширення шкоди.

3) Ідентифікація точок входу

Одним з найважливіших завдань є виявлення вразливих точок входу, через які зловмисники можуть отримати доступ до системи. Це включає ручне сканування системи, перевірку паролів, аудит прав доступу та аналіз активності користувачів.

4) Видалення загроз

Після виявлення вразливостей фахівці вручну видаляють шкідливі файли і програми, чистять реєстр, знешкоджують віруси і шкідливі програми та відновлюють пошкоджені системні файли. У деяких випадках може знадобитися повне перевстановлення операційної системи та програмного забезпечення.

5) Відновлення даних

У разі пошкодження або втрати даних фахівці використовують резервні копії для їх відновлення. Це включає відновлення даних з локальних архівів або хмарних сховищ. Процес ручного відновлення резервних копій часто займає багато часу і вимагає ретельної перевірки, щоб не допустити повторного потрапляння в систему шкідливих файлів.

6) Верифікація систем

Після усунення загроз і відновлення даних системи перевіряються на наявність нових вразливостей і залишків шкідливого коду. Це включає ручне тестування, аналіз результатів сканування безпеки та перевірку коректної роботи ключових сервісів.

7) Відновлення нормальної роботи

Під час цієї фази система поступово повертається до нормальної роботи. Це включає відновлення з'єднань із зовнішніми сервісами, перевірку мережевих з'єднань і відновлення доступу до користувачів.

8) Документування інциденту

Кожен крок ручного відновлення повинен бути задокументований для подальшого аналізу. Ця документація може включати звіт про виявлену загрозу, дії, вжиті для усунення проблеми, та оцінку впливу інциденту на бізнес-процеси. Це важливо для покращення майбутніх процесів реагування.

Недоліки ручних методів:

1) Тривалість процесу

Ручне відновлення може зайняти багато часу, особливо якщо атака зачіпає кілька компонентів інфраструктури. Це може збільшити час простою та призвести до значних фінансових втрат для бізнесу.

2) Потреба в людських ресурсах

Ці методи вимагають висококваліфікованих експертів, яких може не вистачати під час масштабних або складних атак. Відсутність автоматизації збільшує навантаження на команди кібербезпеки і може призвести до людських помилок.

3) Залежність від резервних копій

Відновлення даних часто залежить від наявності актуальних резервних копій. Цей процес може бути дуже складним, якщо резервне копіювання не було виконано належним чином або якщо дані були зашифровані в результаті атаки на шифрування.

4) Відсутність гнучкості

Ручні методи відновлення менш гнучкі, ніж автоматизовані рішення.

Наприклад, якщо атака націлена на кілька систем одночасно, зусилля, необхідні для їх одночасного відновлення вручну, можуть бути значними.

Переваги ручних методів:

1) Контроль над процесом

Однією з переваг ручної праці є те, що кожен етап процесу реставрації можна повністю контролювати.

2) Можливість детального аналізу

Ручні методи дозволяють детально проаналізувати кожен елемент системи.

Цей огляд традиційних методів відновлення ілюструє їхню важливість у сучасних процесах кібербезпеки. Однак розвиток хмарних технологій та автоматизованих рішень значно підвищив швидкість та ефективність цих процесів, що призвело до переходу до автоматизованих методів відновлення після кіберінцидентів.

1.3 Автоматизовані системи відновлення

З розвитком хмарних технологій та інструментів автоматизації процесів все більшої популярності набувають методи автоматизації відновлення після кібератак. Вони дозволяють значно пришвидшити реакцію на інцидент, мінімізувати людський фактор, знизити ймовірність помилок та прискорити процес повернення до нормальної роботи. У цьому розділі ми розглянемо основні принципи, механізми та технології, які дозволяють автоматизувати процеси відновлення.

1.3.1 Основні принципи автоматизованого відновлення

Автоматизоване відновлення ґрунтується на автоматичних діях, які запускаються у відповідь на конкретні події або інциденти. Основні принципи цього підходу включають:

- Автоматичне виявлення загроз: використання систем моніторингу та

виявлення (IDS/IPS, SIEM), які автоматично виявляють підозрілу активність та атаки в режимі реального часу.

- Швидке реагування на інциденти: автоматизовані процеси для пом'якшення наслідків інцидентів, такі як ізоляція скомпрометованих систем, блокування доступу та зупинка шкідливих процесів.
- Автоматизовані сценарії та політики: використання заздалегідь розроблених сценаріїв і політик безпеки для прискорення дій з відновлення.
- Автоматизоване резервне копіювання та відновлення: використовуйте резервні копії для автоматичного відновлення даних і конфігурацій системи в разі пошкодження або втрати даних.
- Інтеграція з хмарними сервісами: хмарна платформа надає інструменти для швидкого розгортання інфраструктури для автоматизованого відновлення системи.

1.3.2 Інструменти та технології автоматизованого відновлення

Сучасні рішення для автоматизованого відновлення використовують різноманітні інструменти, які інтегруються з системами безпеки та управління. Найпоширеніші з них включають:

- Security Orchestration, Automation and Response (SOAR): ці платформи забезпечують автоматизоване реагування на інциденти шляхом інтеграції процесів виявлення загроз, реагування та відновлення. Залежно від типу атаки, відповідні сценарії відновлення можуть бути виконані автоматично.
- Відновлення після аварій як послуга (DRaaS): хмарні сервіси, що забезпечують автоматизоване резервне копіювання та відновлення інфраструктури, часто використовуються для мінімізації часу простою під час масштабних інцидентів.
- Endpoint Detection and Response (EDR): система, яка не тільки виявляє зловмисну активність на кінцевій точці, але й може автоматично поміщати заражені пристрої в карантин і відновлювати їх до попереднього стану за допомогою

резервних копій.

- Backup and RestoreSystems: системи автоматичного резервного копіювання та відновлення, які інтегруються з інструментами моніторингу та виявлення і швидко реагують на втрату даних або пошкодження системи.

1.3.3 Етапи автоматизованого відновлення

Автоматизовані методи відновлення зазвичай поділяються на такі основні етапи:

1) Виявлення інцидентів: використання автоматизованих систем моніторингу для виявлення кібератак або системних збоїв. Це може бути система SIEM, що аналізує журнали подій, або спеціальні інструменти для моніторингу мережевої активності.

2) Аналіз загроз: автоматизовані інструменти, такі як SOAR і EDR, проводять первинний аналіз загроз, щоб визначити тип загрози і ступінь впливу.

3) Реагування: на основі результатів аналізу виконуються автоматизовані дії, такі як поміщення заражених систем в карантин, блокування доступу або зміна налаштувань безпеки.

4) Автоматизоване відновлення: використання резервних копій або «золотих образів» для відновлення пошкодженої системи та відновлення файлів і баз даних, які були зловмисно модифіковані.

5) Верифікація та тестування: автоматизуйте процедури тестування для перевірки цілісності відновлених систем і даних, щоб переконатися, що вони продовжують функціонувати належним чином після інциденту.

6) Звітність і документація: створення автоматизованих звітів про інциденти, результати відновлення та вжиті заходи для подальшого аналізу та вдосконалення процесів безпеки.

1.3.4 Переваги автоматизованого відновлення

Автоматизовані методи відновлення мають кілька важливих переваг над традиційними ручними підходами:

- Швидкість реагування: автоматизоване відновлення значно скорочує час простою і прискорює процес відновлення після інциденту.
- Зменшення людських помилок: автоматизація мінімізує вплив людського фактору на процес відновлення та підвищує надійність і точність дій.
- Масштабованість: автоматизовані системи легко масштабуються навіть у великих інфраструктурах і можуть реагувати на кілька інцидентів одночасно без втручання оператора.
- Економія ресурсів: автоматизовані процеси зменшують потребу в численних технічних спеціалістах, які беруть участь у реагуванні на інциденти та відновленні.

1.3.5 Виклики автоматизації

Незважаючи на свої значні переваги, автоматизоване відновлення також пов'язане з певними проблемами:

- Складність конфігурації: автоматизовані системи потребують ретельної конфігурації та постійного моніторингу, щоб забезпечити їхню ефективність та відповідність бізнес-процесам.
- Залежність від інфраструктури: автоматизація значною мірою залежить від наявності належної інфраструктури, резервного копіювання та інтегрованих систем моніторингу.
- Ризик помилкових дій: якщо автоматизовані процеси налаштовані неправильно, вони можуть призвести до помилкових дій, які можуть погіршити ситуацію або призвести до втрати даних.

1.3.6 Використання хмарних платформ для відновлення

Хмарні платформи, такі як Amazon Web Services (AWS), Microsoft Azure, та Google Cloud Platform (GCP), забезпечують широкий спектр інструментів для автоматизованого резервного копіювання та відновлення після кіберінцидентів. Завдяки своїм можливостям хмари дозволяють значно скоротити час простою систем, мінімізувати втрати даних та забезпечити надійність процесу відновлення. В цьому розділі розглянемо ключові можливості хмарних платформ та їхню роль у процесі відновлення після кіберінцидентів.

Масштабованість та доступність. Одна з основних переваг хмарних платформ полягає в їхній здатності до масштабування ресурсів відповідно до потреб користувача. Хмарні провайдери пропонують можливість резервного копіювання великих обсягів даних і системних компонентів у різних географічних регіонах, що забезпечує високий рівень доступності навіть під час атак або збоїв у окремих дата-центрах.

AWS S3 та інші сервіси зберігання даних дозволяють автоматично копіювати та відновлювати дані на кількох географічно розподілених серверах, забезпечуючи доступність і захист від фізичних втрат. Це робить відновлення швидшим і безпечнішим, адже навіть у разі збою в одному регіоні, інші сервери продовжують працювати.

Azure Backup пропонує інтегровані рішення для резервного копіювання та відновлення, включаючи автоматизовані правила та політики зберігання. Azure також забезпечує шифрування та контроль доступу до даних, що є важливим для збереження конфіденційності інформації під час інцидентів.

Автоматизація відновлення. Хмарні платформи використовують автоматизовані інструменти для відновлення систем після інцидентів. Це включає автоматичне резервне копіювання, розгортання нових серверів, та відновлення пошкоджених даних або конфігурацій без ручного втручання.

AWS Backup пропонує централізовану систему управління резервними копіями для різних сервісів AWS (EC2, RDS, DynamoDB), що дозволяє

автоматизувати процеси створення та відновлення резервних копій. Адміністратори можуть налаштовувати політики резервного копіювання для автоматичного створення копій на основі графіків, що зменшує людський фактор і знижує ризики.

Google Cloud Backup and DR надає можливості для резервного копіювання з мінімальним часом простою. Автоматизація включає в себе відновлення цілих систем за короткий час після збою або атаки. Системи зберігають кілька копій даних, що мінімізує втрати у разі кіберінцидентів.

Захист даних та шифрування. Хмарні платформи забезпечують високий рівень захисту даних через шифрування на всіх етапах: під час передачі, зберігання і відновлення. Це допомагає захистити конфіденційну інформацію навіть у разі успішної атаки.

AWS KMS (Key Management Service) та Azure Key Vault забезпечують безпечне зберігання та управління ключами шифрування, що використовуються для захисту резервних копій. Це дозволяє організаціям контролювати доступ до даних навіть під час відновлення після інцидентів.

Google Cloud також пропонує вбудовані механізми шифрування, що робить процес відновлення не лише швидким, але й безпечним. Шифрування дозволяє зберігати чутливі дані під час відновлення і запобігає витоку інформації.

Моніторинг та реагування на інциденти. Хмарні платформи надають інструменти для безперервного моніторингу систем, які здатні виявляти загрози у реальному часі та автоматично запускати відновлювальні процеси. Це дозволяє знизити час реагування на інциденти та мінімізувати наслідки атак.

AWS CloudWatch та Azure Monitor дозволяють відслідковувати події та аномалії у системах, автоматично надсилаючи попередження адміністраторам або запускаючи відновлення за попередньо налаштованими сценаріями. Це допомагає миттєво відреагувати на кіберзагрозу без втручання людини.

Google Cloud Operations Suite також надає можливості для моніторингу систем і автоматичного запуску скриптів для відновлення або ізоляції пошкоджених системних компонентів.

Планування і тестування відновлення. Хмарні платформи дозволяють проводити регулярне планування та тестування сценаріїв відновлення, що є критично важливим для забезпечення готовності систем до кіберзагроз.

AWS Elastic Disaster Recovery дозволяє відтворювати інфраструктуру на випадок надзвичайних ситуацій, з можливістю постійного тестування та оптимізації планів відновлення.

Azure Site Recovery надає можливість автоматично відновлювати роботу у разі катастрофи або кіберінциденту, забезпечуючи мінімальний час простою та збереження даних на основі постійного тестування.

1.4 Управління ризиками та витрати на кіберзахист

Термін “ризик” зазвичай використовується для позначення можливості втрат і/або пошкоджень, що представляється як добуток впливу на ймовірність настання. Ризик завжди супроводжується певною мірою невизначеності, а результати важко передбачити. У залежності від контексту, ризики можуть бути різних типів: бізнесові, економічні, безпекові. Також ризик можна визначити як імовірність настання несприятливої події, наприклад, ймовірність вищих економічних втрат через кібератаки, ніж очікувалося. Важливо розуміти, що ризик — це невід’ємний елемент будь-якої компанії.

У цьому розділі ризик розглядається як потенційне негативне відхилення від найбільш ймовірної суми втрат. Витрати на кібербезпеку включають не лише прямі витрати (наприклад, витрати на відновлення систем і програмного забезпечення), але й непрямі (зокрема, репутаційні втрати та судові позови) та альтернативні витрати (як-от втрачені можливості для бізнесу, через те що ресурси перенаправлені на відновлення після атаки). Ці види витрат обговорюються в даному розділі, а приклади наведено у таблиці 1.

Таблиця 1.1 – Види витрат, пов’язаних з кіберінцидентами

Тип витрат	**Витрати на запобігання**	**Витрати на наслідки**	**Витрати на реагування**
Прямі	Страхування від кіберзагроз, інструменти захисту	Втрата прибутку	Оплата юридичних послуг
Непрямі	Дослідження інциденту, навчання співробітників	Моніторинг систем	Документація та звітність
Витрати можливостей	Інвестиції в нові стратегії кібербезпеки	Втрата репутації	Перебої в роботі бізнесу

Ці витрати можуть бути вражаючими: середня вартість витоку даних для транснаціональних підприємств у 2020 році становила 1,09 мільйона доларів США, порівняно з 1,41 мільйона доларів США у 2019 році, тоді як для малих та середніх підприємств середні витрати склали 101 000 доларів США, порівняно з 108 000 доларів США у 2019 році.[source1.4.1] Бізнеси можуть зменшити вартість витоків даних різними способами. Наприклад, швидке виявлення витоку може знизити втрати на 32% для транснаціональних підприємств і на 17% для малих та середніх підприємств завдяки економії на прямих і непрямих витратах. Також, проактивне повідомлення клієнтів і зацікавлених сторін про інцидент з витоком даних може зменшити фінансові збитки на 28% для транснаціональних підприємств і на 40% для малих та середніх підприємств. [5]

Іншим фактором зниження витрат на кібератаки є оновлене програмне та апаратне забезпечення. Якщо компанія не має ефективного підходу до оновлень, це може призвести до зростання витрат на кібератаки на 47% для транснаціональних підприємств і на 54% для малих та середніх підприємств. У випадку, коли компанія постраждала від програм-вимагачів, середні витрати на відновлення становили 761 106 доларів США у 2021 році, що включає витрати, пов'язані зі зменшенням

впливу, відновленням та зупинкою бізнес-процесів. Оплата зловмисникам для відновлення даних також стає все більш поширеною практикою, причому суми платежів варіюються від кількох тисяч до мільйонів доларів, залежно від розміру компанії. У 2021 році середній платіж, здійснений однією атакованою компанією, становив 233 817 доларів США. Витрати також дуже високі на суспільному та державному рівнях, що спонукало уряд Сполучених Штатів протидіяти угрупуванням зловмисників, які використовують програми-вимагачі. У 2021 році уряд США оголосив винагороду в 10 мільйонів доларів за інформацію, яка допоможе впіймати угруповання, відоме як DarkSide. [5][6][7]

Окрім прямих витрат, пов’язаних з кібератаками, існує також багато непрямих витрат, таких як штрафи та регуляторні заходи, які накладаються до або після атак урядами та іншими організаціями на компанії, що не дотримувалися правил кібербезпеки. Наприклад, з липня 2018 по листопад 2021 року Загальний регламент про захист даних (GDPR) спричинив накладення 837 штрафів у Європі, загальна сума яких склала 1,2 мільярда євро. Найпоширеніші типи порушень включали недостатнє правове обґрунтування для обробки даних (299 штрафів), недостатні технічні та організаційні заходи для забезпечення безпеки інформації (179 штрафів) та недотримання загальних принципів обробки даних (178 штрафів). [8][9]

Таблиця 1.1 демонструє приклади найбільш поширених витрат, пов'язаних з кібератаками, які обговорювались у наукових і промислових колах за останні кілька років. Ці значення необхідно враховувати під час визначення ризику та пов'язаних із ним наслідків. Під «попереджувальними витратами» розуміються витрати, які виникають до того, як відбудеться кібератака, і включають витрати на додаткові заходи безпеки та втрачені можливості через ці дії. Витрати, пов'язані з наслідками, відображають безпосередній економічний вплив кібератаки. У свою чергу, витрати на реагування пов'язані з витратами, що виникають відразу після кібератаки, включаючи правові, технічні та економічні наслідки. Таким чином, як показано в Таблиці 1, прямі, непрямі та альтернативні витрати можуть виникати на різних етапах кібератаки — від попередження до дій після інциденту.

У першому розділі було проведено детальний аналіз типів кібератак та їх впливу на організації. Виявлено, що кількість кібератак зростає щороку, а методи атак стають все більш витонченими. Традиційні ручні методи відновлення, хоча і є основою багатьох стратегій кібербезпеки, мають значні недоліки, такі як тривалість процесу та потреба в значних людських ресурсах. Автоматизовані системи відновлення, зокрема з використанням хмарних технологій, пропонують значні переваги, включаючи швидкість реагування, зменшення людських помилок та масштабованість.

РОЗІДЛ 2 РОЗГЛЯД МОЖЛИВОСТЕЙ ХМАРНИХ ТЕХНОЛОГІЙ ДО ВІДНОВЛЕННЯ ПІСЛЕ ІНЦИДЕНТІВ

2.1 Огляд хмарних технологій і їх застосування для відновлення після інцидентів

Хмарні технології стали одним із ключових елементів сучасної стратегії кібербезпеки, забезпечуючи високу доступність, масштабованість, і широкий набір інструментів для відновлення після інцидентів. Використання хмарних платформ дозволяє організаціям швидко відновити роботу критичних систем після атак або збоїв, мінімізуючи простої і втрати даних. У цьому розділі розглянемо основні хмарні платформи та технології, які використовуються для реалізації стратегії відновлення після інцидентів.

1. Високодоступні хмарні середовища та безперервність бізнесу. Однією з головних переваг хмарних платформ, таких як AWS, Azure і Google Cloud, є їхня здатність підтримувати високодоступні середовища. Вони дозволяють використовувати географічно розподілені дата-центри для збереження резервних копій та безперервності бізнесу. За рахунок можливостей автоматичного відновлення даних та переведення навантаження на інші регіони, компанії можуть уникати простоїв, навіть у випадках великих інцидентів .

2. Резервне копіювання та відновлення. Резервне копіювання є основою стратегії відновлення після інцидентів. Хмарні провайдери пропонують надійні інструменти для резервного копіювання даних, включаючи автоматизовані політики резервного копіювання та зберігання. Наприклад, AWS Backup, Azure Backup, і Google Cloud Backup надають гнучкі налаштування, що дозволяють встановити регулярність копіювання, тривалість зберігання, та умови автоматичного видалення копій, що застаріли.

3. Автоматизація процесів відновлення. Хмарні сервіси також пропонують інструменти для автоматизації процесів відновлення, що забезпечують швидке і надійне повернення до нормальної роботи. Наприклад, за допомогою

Amazon CloudFormation, Azure Resource Manager, або Google Deployment Manager можна автоматизувати процеси відновлення, такі як створення нових інфраструктурних компонентів та налаштування служб. Цей підхід дозволяє уникнути людських помилок та знижує час відновлення у випадку інциденту .

4. Контейнеризація та оркестрація. Технології контейнеризації та оркестрації, такі як Docker та Kubernetes, дозволяють швидко розгортати і відновлювати додатки у хмарному середовищі. З використанням Kubernetes, компанії можуть створювати резервні копії стану контейнерів та їхнього зберігання, що дозволяє відновлювати додатки до попереднього стану після збою або атаки. Це також забезпечує підтримку безперервної роботи у випадку інцидентів за рахунок переведення навантаження на інші вузли або кластери .

5. Відповідність нормативним вимогам та управління доступом. Крім технічних рішень, хмарні платформи забезпечують відповідність нормативним вимогам за рахунок вбудованих засобів шифрування даних, захисту резервних копій, і керування доступом. Наприклад, такі сервіси як AWS IAM, Azure Active Directory, і Google Cloud IAM дозволяють компаніям контролювати доступ до даних та інфраструктури, що мінімізує ризики несанкціонованого доступу до резервних копій та інших критичних ресурсів під час інциденту .

Таким чином, хмарні технології надають компаніям інструменти для швидкого відновлення після кіберінцидентів, покращуючи доступність даних, забезпечуючи їхній захист, і мінімізуючи наслідки від можливих атак або технічних збоїв. Застосування хмарних платформ також сприяє оптимізації процесів відновлення та підвищенню надійності роботи додатків і даних в умовах різноманітних загроз.

2.2 Огляд хмарних технологій і їх застосування для відновлення після інцидентів

Для успішного відновлення бізнесу після кібератак або інших аварійних інцидентів, компанії потребують чітко визначеного плану відновлення після аварій

(Disaster Recovery Plan, DRP). Впровадження хмарних технологій у процес DRP забезпечує більшу гнучкість та швидкість відновлення даних, мінімізуючи втрати часу та ресурсів. Хмарні рішення дають змогу зберігати резервні копії у віддалених дата-центрах, що підвищує захищеність даних від локальних загроз та дозволяє організаціям швидше повертатися до нормальної роботи після інцидентів.

На наступній схемі показані основні етапи побудови плану відновлення після аварій у хмарному середовищі. Цей процес охоплює всі аспекти, починаючи від аналізу бізнес-ризків і закінчуючи регулярним тестуванням плану, щоб забезпечити його ефективність у кризових ситуаціях.

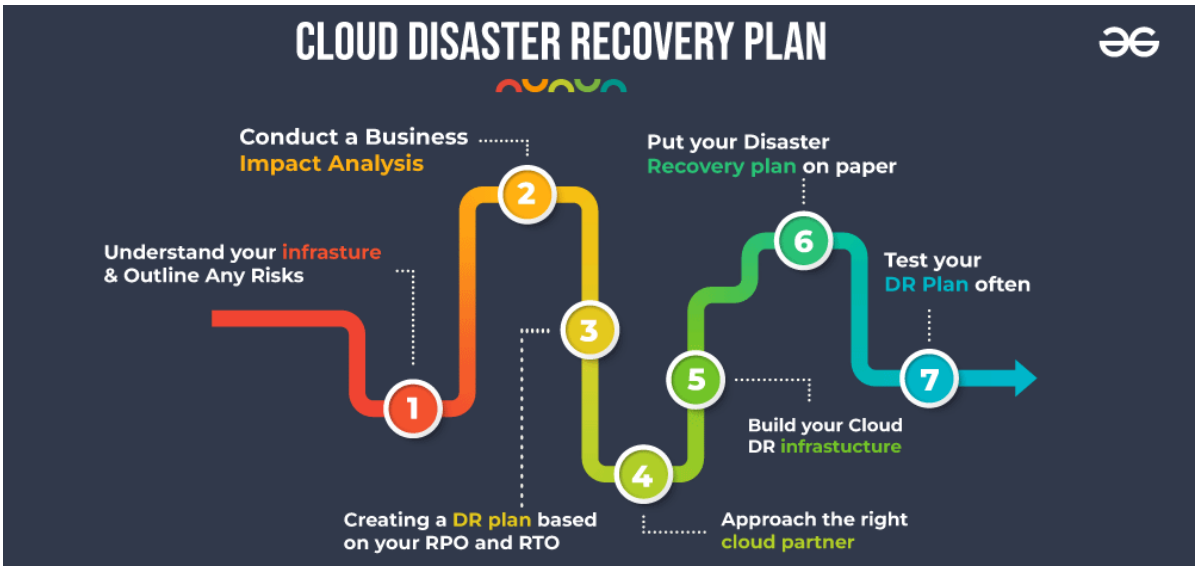

Рисунок 2.1 – Етапи створення Disaster Recovery Plan

На рисунку 2.1 показано основні етапи створення та реалізації плану відновлення після аварій у хмарному середовищі:

1. Розуміння інфраструктури та визначення ризиків. Необхідно детально вивчити поточну IT-інфраструктуру, включаючи мережеві ресурси, сервери та системи зберігання даних. Важливо окреслити всі можливі ризики, що можуть призвести до інцидентів, таких як збій обладнання, кібератаки або природні катастрофи.

2. Проведення аналізу впливу на бізнес (Business Impact Analysis). Цей крок включає оцінку критичних бізнес-процесів, розуміння, як саме збої можуть

вплинути на операції та прибутковість компанії. Під час аналізу визначаються ключові ризики для кожного процесу, щоб встановити пріоритети відновлення.

3. Створення плану відновлення на основі параметрів RPO та RTO. RPO (Recovery Point Objective) та RTO (Recovery Time Objective) – це цільові показники для відновлення. RPO визначає, скільки даних компанія може дозволити собі втратити (наприклад, за останні години до інциденту), тоді як RTO визначає максимальний час простою. План розробляється відповідно до цих параметрів, щоб знизити втрати та час відновлення.

4. Документування плану відновлення після аварій. Важливо зафіксувати план відновлення в документі, який буде містити покрокову інструкцію для співробітників. Це полегшує виконання плану під час аварії, оскільки персонал має доступ до чітких дій та рекомендацій.

5. Побудова інфраструктури для відновлення в хмарі. На цьому етапі організація створює ресурси в хмарному середовищі, необхідні для швидкого відновлення даних та систем. Це може включати резервні сервери, віддалені копії даних та оркестрацію процесів відновлення.

6. Пошук надійного партнера для хмарного DR (Disaster Recovery). Вибір партнера з досвідом у відновленні після аварій є важливим, оскільки партнери можуть надати додаткові ресурси, консультації та підтримку. Такий партнер допоможе оптимізувати план і надасть експертні рекомендації.

7. Регулярне тестування DR-плану. Періодичне тестування плану відновлення дозволяє перевірити його ефективність та виявити можливі слабкі місця. Це допомагає впевнитися, що план працює коректно та може бути реалізований швидко й без затримок у разі реального інциденту.

Кожен з цих кроків допомагає створити ефективний план відновлення після аварій у хмарному середовищі, що є важливою складовою для безперервності бізнесу.

2.3 Інфраструктура як Код (IaC) і автоматизація процесів відновлення

Інфраструктура як Код (IaC) — це підхід до управління та забезпечення інфраструктурних ресурсів за допомогою коду, що дозволяє автоматизувати процеси налаштування і управління серверними ресурсами.[12] Використовуючи IaC, організації можуть швидко та надійно відновлювати свою інфраструктуру після інцидентів. Основними інструментами для реалізації IaC є Ansible, Terraform, та AWS CloudFormation.

IaC дозволяє зберігати конфігурації всієї інфраструктури в коді, який легко версіонувати та перевіряти, що гарантує наявність необхідних резервних планів для швидкого відновлення в разі катастроф. Автоматизація відновлення може бути здійснена шляхом розробки скриптів, які запускаються одразу після інциденту, ініціюючи процес відновлення на основі описаних сценаріїв (playbooks).

Ansible — це програмне забезпечення для автоматизації IT з відкритим вихідним кодом, написане мовою Python. Він може налаштовувати системи, розгортати програмне забезпечення та організовувати розширені робочі процеси для підтримки розгортання додатків, оновлення системи тощо. Він дозволяє адміністраторам створювати та керувати інфраструктурою через прості YAML-файли (Ansible Playbooks), в яких описуються конфігурації та сценарії для керування серверами та іншими компонентами. Архітектура Ansible не вимагає агентів на серверах — він працює за допомогою протоколу SSH або WinRM для керування ресурсами.

Як працює Ansible з хмарними технологіями:

1. Playbooks описують операції з розгортання і конфігурації в хмарі.
2. За допомогою SSH або API-запитів до хмарних провайдерів, таких як AWS, Azure, Google Cloud, Ansible керує інфраструктурою.
3. Він легко інтегрується з іншими хмарними сервісами, як-от AWS CloudFormation, дозволяючи автоматизувати процес розгортання нових ресурсів та

управління існуючими.

Переваги Ansible:

- Простота використання: не вимагає встановлення агентів на клієнтських машинах.
- Універсальність: підтримує різні операційні системи та платформи, що робить його універсальним рішенням для багатохмарної або гібридної інфраструктури.
- Масштабованість: легко масштабується в великих середовищах через паралельне виконання завдань.

Недоліки Ansible:

- Швидкість виконання: порівняно з іншими інструментами, такими як Puppet або Chef, Ansible може бути повільнішим через свій безагентний підхід.
- Обмежена підтримка складних середовищ: коли мова йде про складні багатошарові середовища, Ansible може вимагати значних налаштувань і тестування.

Terraform — це інструмент «інфраструктура як код», який дозволяє вам визначати хмарні та локальні ресурси у зручних для читання конфігураційних файлах, які ви можете змінювати, повторно використовувати та ділитися ними. Потім ви можете використовувати послідовний робочий процес для забезпечення та управління всією інфраструктурою протягом її життєвого циклу. Terraform може керувати низькорівневими компонентами, такими як обчислювальні, сховища та мережеві ресурси, а також високорівневими компонентами, такими як записи DNS та функції SaaS.[13]

Архітектура Terraform:

1. Декларативний підхід: користувачі описують бажану інфраструктуру за допомогою мовою конфігурації Terraform (HCL).
2. Планування та застосування: Terraform генерує план змін перед їх застосуванням, що дозволяє бачити всі дії, які будуть виконані в інфраструктурі.
3. Підтримка багатьох провайдерів: Terraform може керувати інфраструктурою на різних платформах, включно з AWS, Azure, Google Cloud та

багатьма іншими.

4. Стан інфраструктури: інструмент відслідковує стан усіх ресурсів і синхронізує їх зі збереженою версією, забезпечуючи точне управління та відновлення ресурсів.

Переваги Terraform:

- Підтримка багатохмарних середовищ: дозволяє легко керувати інфраструктурою на різних хмарних платформах через один інтерфейс.
- Стан інфраструктури: забезпечує чіткий контроль над поточним станом інфраструктури.
- Масштабованість: Terraform ефективно масштабується для великих середовищ і дозволяє автоматизувати складні архітектури.

Недоліки Terraform:

- Час на освоєння: через декларативний підхід і унікальну синтаксис, новим користувачам може знадобитися більше часу для його освоєння.
- Обмеження в інтеграціях: хоча Terraform підтримує багато провайдерів, деякі з них можуть мати обмежені можливості або не повністю підтримувати всі функції конкретного хмарного провайдера.

Порівняння Ansible та Terraform:

- Ansible краще підходить для автоматизації конфігурації, управління системами та розгортання додатків.
- Terraform найкраще використовувати для створення та управління інфраструктурою як кодом, особливо в багатохмарних середовищах.
- Обидва інструменти можуть працювати в парі: Terraform можна використовувати для створення інфраструктури, а Ansible для управління її конфігураціями.

2.4 Хмарні сервіси для забезпечення резервного копіювання та відновлення

Резервне копіювання та відновлення даних є критично важливими

компонентами будь-якої стратегії відновлення після аварій (Disaster Recovery, DR). У сучасних умовах хмарних обчислень ці завдання можуть бути значно спрощені та автоматизовані за допомогою хмарних сервісів. Вони дозволяють забезпечити високу доступність, надійність, а також мінімізувати ризики втрати даних у випадку катастроф або інших інцидентів.

1. Amazon Web Services (AWS) Backup

AWS Backup — це керований сервіс, який дозволяє автоматизувати та централізувати процеси резервного копіювання і відновлення даних в інфраструктурі AWS. Він підтримує інтеграцію з багатьма сервісами AWS, такими як Amazon EC2 (віртуальні машини), Amazon RDS (бази даних), Amazon EFS (файлові системи), та іншими. Основні можливості включають:

- Централізоване управління процесами резервного копіювання і відновлення для всіх ресурсів в AWS через єдину консоль.
- Політики резервного копіювання для автоматизації процесу створення копій даних.
- Шифрування даних під час зберігання та передачі.
- Журналювання і аудит всіх операцій резервного копіювання через AWS CloudTrail.

Верхньорівнева модель роботи AWS Backup, представлена на рисунку 2.2, охоплює ключовий функціонал, який допомагають гарантувати надійність та безперервність у формуванні резервного копіювання компонентів.

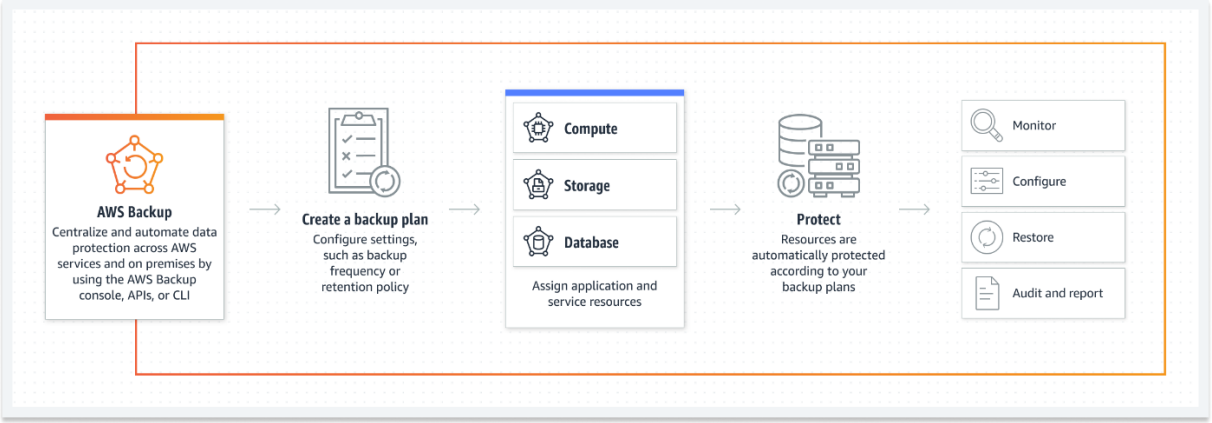

Рисунок 2.2 – Верхньорівневий опис дизайну продукту AWS Backup

2. Microsoft Azure Backup

Azure Backup є керованим рішенням для резервного копіювання даних в екосистемі Microsoft Azure. Цей сервіс дозволяє користувачам автоматично створювати резервні копії даних із віртуальних машин, баз даних SQL, а також інших типів даних у хмарі. Основні функції:

- Безагентна архітектура для захисту віртуальних машин, баз даних і файлів.
- Інкрементальні резервні копії, які зменшують використання сховища та підвищують ефективність.
- Geo-реплікація, яка забезпечує резервне копіювання в різних географічних регіонах для підвищення надійності.
- Автоматизоване управління життєвим циклом резервних копій.

Верхньорівнева модель роботи Azure Backup Service, представлена на рисунку 2.3, охоплює основні функції, які забезпечують надійність і безперервність процесів резервного копіювання. Azure Backup об’єднує як локальні ресурси (через агентів резервного копіювання), так і хмарні системи (за допомогою вбудованих рішень для резервування), дозволяючи централізовано керувати точками відновлення, керуванням сховищем, доступом, моніторингом і звітністю.

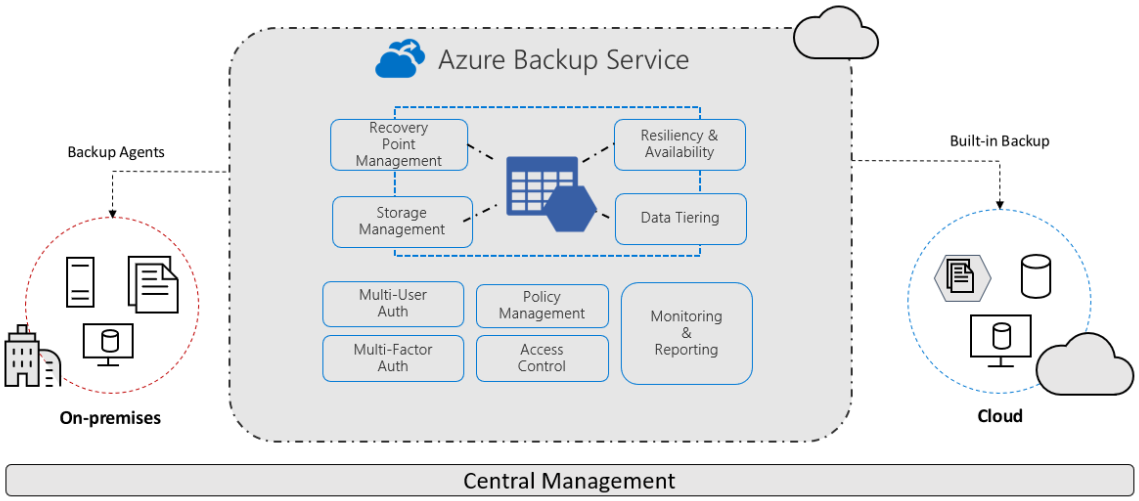

Рисунок 2.3 – Верхньорівневий опис дизайну продукту Azure Backup Service

3. Google Cloud Backup and Disaster Recovery

Google Cloud Backup and Disaster Recovery — це рішення, яке дозволяє користувачам Google Cloud автоматизувати процеси резервного копіювання даних та відновлення. Воно підтримує не лише хмарні ресурси, але й гібридні середовища, з можливістю копіювання локальних даних у хмару. Основні можливості включають:

- Підтримка мультихмарних та гібридних середовищ, що дозволяє захищати дані незалежно від їхнього розташування.
- Планування політик резервного копіювання і налаштування частоти створення копій.
- Можливість швидкого відновлення даних для мінімізації простоїв.
- Шифрування даних під час зберігання та передачі.

4. Backblaze B2 Cloud Storage

Backblaze B2 є відомим сервісом для зберігання та резервного копіювання даних в хмарі, що орієнтований на простоту використання та низьку вартість. Його можливості включають:

- Низькі витрати на зберігання у порівнянні з іншими постачальниками хмарних послуг.

- Підтримка API для інтеграції з іншими сервісами та інструментами резервного копіювання.
- Автоматичне масштабування обсягів сховища відповідно до потреб користувачів.
- Висока доступність даних завдяки розподіленому зберіганню.

Верхньорівнева архітектура Google Cloud Backup and Disaster Recovery, представлена на рисунку 2.4, охоплює ключові функції для забезпечення надійності та безперервності процесів резервного копіювання і відновлення. Архітектура підтримує резервне копіювання баз даних, віртуальних машин Compute Engine і VMware, файлових систем, а також локальних віртуальних машин VMware, забезпечуючи інкрементальні резервні копії на постійній основі.

Резервні копії для віртуальних машин Compute Engine зберігаються як знімки дисків (Persistent Disk snapshots), тоді як резервні копії для VMware і баз даних зберігаються на Persistent Disk або в Cloud Storage з варіантами зберігання, такими як Nearline та Coldline, що дозволяє обирати відповідний рівень доступності для даних.

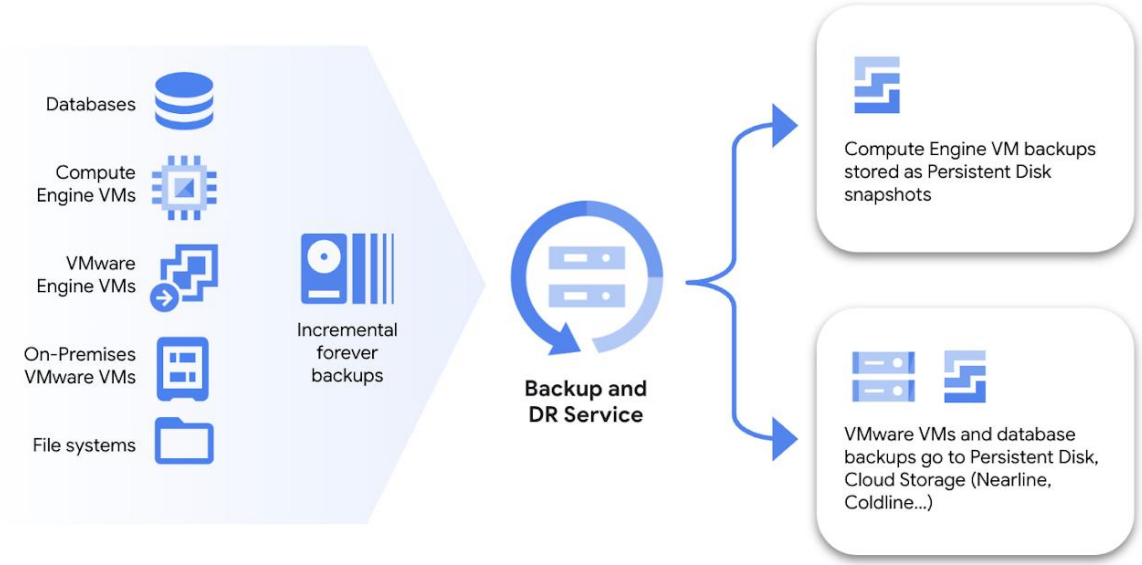

Рисунок 2.4 – Верхньорівневий опис дизайну продукту Google Cloud Backup and Disaster Recovery

2.5 Контейнеризація і оркестрація в хмарних середовищах для відновлення після інцидент

Контейнеризація — це розгортання програмного забезпечення разом з усіма необхідними компонентами: код, бібліотеки, фреймворки тощо, таким чином, щоб вони були ізольовані у власному контейнері.[14]

Контейнеризація і оркестрація стали невід'ємними елементами сучасної хмарної інфраструктури, забезпечуючи високий рівень гнучкості, масштабованості та швидке розгортання для відновлення після інцидентів. Застосування контейнерів і інструментів оркестрації дозволяє організаціям відновлювати системи в разі аварій та кіберінцидентів швидше та з мінімальними витратами ресурсів.

Головна причина розробки технології контейнеризації — поява складних і високонавантажених IT-інфраструктур. Для їхньої повноцінної роботи необхідно забезпечити надійність кожного окремого елемента та мінімізувати їхній вплив один на одного. Потрібна операційна система та ПЗ, які зможуть «завертати» ці елементи в контейнери, дробити великі та монолітні IT-інфраструктури на безліч дрібних, незалежних елементів, взаємодіючих між собою як одне ціле.[14]

Архітектура контейнеризації має кілька шарів, які дозволяють гнучко керувати контейнерами та забезпечують їх незалежність від базової інфраструктури:

1. Інфраструктура: Базовий апаратний шар, який забезпечує фізичні ресурси для виконання контейнерних додатків. Він включає сервер або комп'ютер без встановленої операційної системи. Цей рівень може бути локальним сервером або віртуальним сервером у хмарі.

2. Операційна система (ОС): Контейнеризація зазвичай використовує операційну систему Linux завдяки її сумісності та легкості інтеграції з контейнерами. ОС забезпечує середовище для контейнерного рушія та образів контейнерів і може працювати як локально, так і в хмарі.

3. Образи контейнерів: Це файли, створені розробниками, які містять всю інформацію, необхідну для запуску контейнерного додатка. Вони є статичними,

доступними тільки для читання, і створюються відповідно до стандарту Open Container Initiative (OCI). Контейнерні образи гарантують, що додаток буде працювати однаково незалежно від середовища, в якому він розгорнутий.

4. Контейнерний рушій: Програмне забезпечення, що створює та керує контейнерами на основі образів. Воно виступає як посередник між контейнерами та операційною системою, забезпечуючи необхідні ресурси та ізоляцію. Контейнерний рушій може керувати кількома контейнерами на одній ОС, зберігаючи їх незалежність від апаратного шару та один від одного.

5. Додаток і залежності: Це найвищий шар архітектури контейнера, що містить код додатка, бібліотеки, файли конфігурації та всі необхідні залежності. Цей шар дозволяє розробникам легко включати все необхідне для роботи додатка, а також, за потреби, додаткову гостьову операційну систему, що розгортається поверх основної ОС для забезпечення ізоляції.

На рисунку 2.5 наглядно зображено різницю між контейнеризацією та віртуалізацією, що пояснює простоту та ефективність контейнерів у порівнянні з традиційними віртуальними машинами. Контейнери працюють на одному ядрі операційної системи та не потребують окремих гостьових ОС для кожного додатка, що значно знижує витрати ресурсів і спрощує керування. Віртуалізація, натомість, вимагає повної копії ОС для кожної віртуальної машини, що збільшує навантаження на апаратну інфраструктуру і ускладнює підтримку та оновлення додатків.

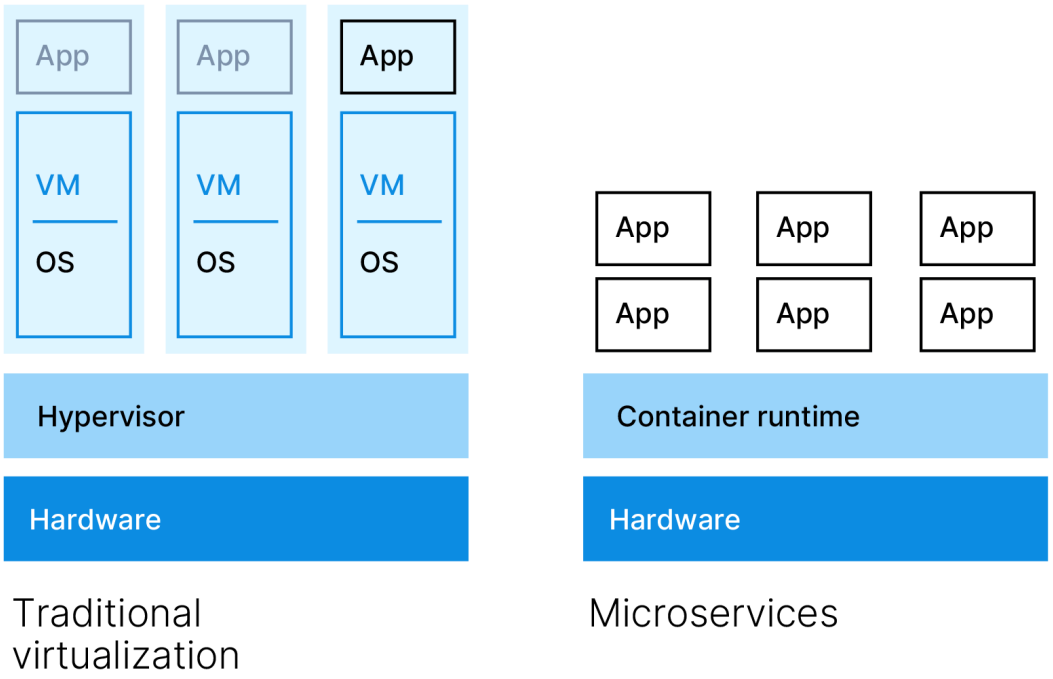

Рисунок 2.5 – Схематичне зрівняння віртуалізації та контейнеризації

Оркестрація контейнерів передбачає управління великими кількостями контейнерів для забезпечення стабільної роботи, балансування навантаження, автоматичного масштабування та відновлення. Інструменти оркестрації, такі як Kubernetes і Docker Swarm, дозволяють автоматизувати процеси запуску, управління та відновлення контейнеризованих додатків.

Щоб зрозуміти, що таке Kubernetes (від грецького "штурман" або "керманич"), варто ознайомитися з базовими поняттями. Цю платформу часто скорочено називають K8s. Kubernetes — це система оркестрації, яка автоматизує управління контейнерами, забезпечуючи їх масштабування, розгортання та контроль. Контейнери слугують ізольованими середовищами для запуску додатків і містять всі необхідні файли у вигляді образу, що дозволяє додаткам працювати незалежно від обладнання. Завдяки цьому контейнери легко переносити між фізичними серверами, хмарними платформами та різними операційними системами.

Кластер Kubernetes складається з контрольної площини (control plane) і набору робочих машин, які називаються вузлами (nodes) і виконують контейнеризовані додатки. Кожен кластер потребує щонайменше одного робочого

вузла для запуску Pod-ів.

Робочі вузли (worker nodes) розміщують Pod-и, які є компонентами робочого навантаження додатків. Контрольна площина керує робочими вузлами та Pod-ами в кластері. У виробничих середовищах контрольна площина зазвичай працює на декількох комп'ютерах, а кластер має кілька вузлів, що забезпечує відмовостійкість і високу доступність.

Для наглядного огляду архітектури розглянемо рисунок 2.5.

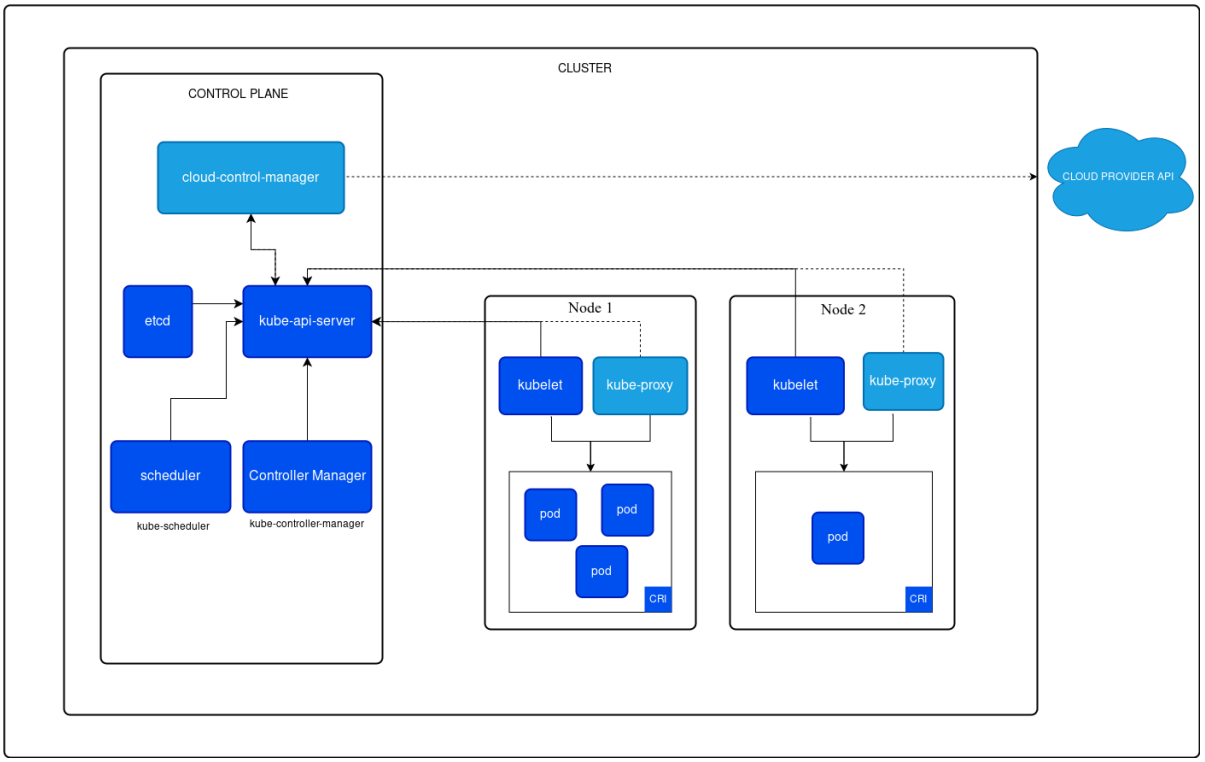

Рисунок 2.6 – Архітектура Kubernetes

Основні компоненти Kubernetes забезпечують ефективне керування та оркестрацію контейнеризованих додатків. Ось ключові з них:

1. Kubernetes Master (Контрольна площина):

- API Server: основний компонент для взаємодії з кластером Kubernetes. Він обробляє запити від користувачів, інших компонентів і зовнішніх інструментів.

- Scheduler: визначає, на яких вузлах (Node) запускатимуться нові контейнери, розподіляючи навантаження на основі ресурсів і вимог.

- Controller Manager: керує контролерами (різними процесами), які відповідають за життєвий цикл об’єктів, наприклад, відстежують відновлення відмовлених Pod-ів.
- etcd: сховище конфігурацій кластеру, що зберігає всю інформацію про його стан.

2. Вузли (Nodes):

- Kubelet: агент на кожному вузлі, що виконує команди від API-сервера, забезпечуючи запуск і моніторинг контейнерів.
- Kube-proxy: мережевий компонент, який дозволяє доступ до Pod-ів та виконує балансування навантаження.
- Container Runtime: програмне забезпечення для запуску контейнерів, наприклад, Docker або containerd.

3. Pod: Це найменша одиниця Kubernetes, яка може містити один або більше контейнерів. Pod-и працюють на одному вузлі та поділяють мережевий простір і ресурси.

4. Сервіси (Services): Служби в Kubernetes забезпечують доступ до набору Pod-ів за статичною IP-адресою та ім'ям, навіть якщо ці Pod-и переміщуються між вузлами.

5. Namespace: Логічні групи, що дозволяють розділяти ресурси Kubernetes для ізоляції та управління доступом у кластері.

6. Ingress: Компонент, який дозволяє керувати зовнішнім доступом до сервісів, часто використовується для налаштування HTTP або HTTPS маршрутизації. Детальніше можемо розглянути приклад роботи компоненту Ingress на рисунку 2.7.

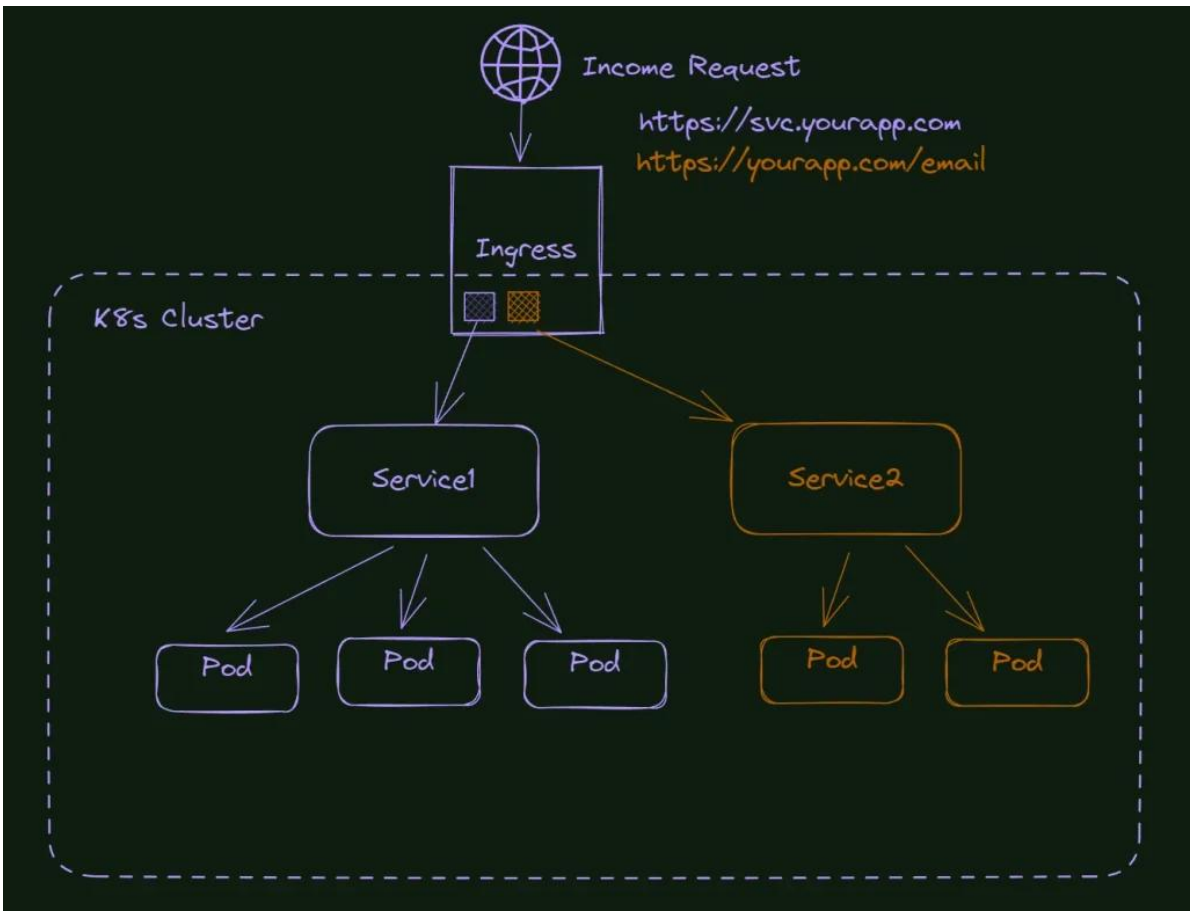

Рисунок 2.7 – Приклад роботи Ingress

Свого часу платформа Kubernetes справила справжню революцію в управлінні програмами не просто залишається затребуваною, а й демонструє стрімке зростання популярності. Свого часу вона стала одним із найбільш швидко зростаючих проєктів з відкритим кодом. Вже до 2030 року, за прогнозами Research and Markets, розміри світового ринку Kubernetes сягнуть 7,8 млрд доларів із середньорічним темпом зростання 23,4%. А за даними Oracle, цю платформу використовують у понад 50% компаній зі списку Fortune 100 (перша сотня найбільших компаній США). [15]

2.6 Захист резервних копій

Захист резервних копій необхідний для запобігання кібератакам, таким як програми-вимагачі, які шифрують системи або видаляють копії, роблячи їх невідновлюваними. Він також забезпечує конфіденційність конфіденційних даних

і знижує ризик їх витоку та пов'язаних з цим юридичних наслідків. Надійний захист забезпечує безперервність бізнесу та мінімізує час простою і пов'язані з ним витрати, захищаючи від випадкового видалення через людську помилку та гарантуючи швидке відновлення критично важливих систем після збою.

2.6.1 Захист резервних копій в Amazon Web Services

Надійне резервне копіювання та відновлення даних є ключовим аспектом забезпечення безперервності бізнесу та захисту від втрати даних. У хмарній інфраструктурі AWS надає потужні інструменти резервного копіювання, контролю доступу, шифрування, моніторингу та тестування відновлення, які допомагають організаціям дотримуватися найкращих практик безпеки та відповідати нормативним вимогам. Нижче наведено ключові рекомендації щодо організації ефективної стратегії резервного копіювання на базі AWS, яка захищає дані від несанкціонованого доступу та випадкових інцидентів.

1. Реалізація механізмів контролю доступу. Забезпечення безпеки в хмарі починається з надійного управління ідентифікацією користувачів, щоб гарантувати, що кожен користувач має лише необхідні дозволи для доступу до даних. Використання відповідних методів автентифікації та авторизації дозволяє знизити ризик виникнення інцидентів безпеки. AWS використовує модель спільної відповідальності, де клієнти мають впроваджувати політики контролю доступу. За допомогою AWS Identity and Access Management (IAM) можна створювати та керувати політиками доступу в масштабі.

Для налаштування прав доступу важливо дотримуватися принципу мінімальних привілеїв, надаючи користувачам лише необхідні дозволи для виконання їхніх завдань. Використовуючи AWS Backup, ви можете впровадити політики контролю доступу, наприклад, дозволити створювати резервні копії, але обмежити можливість видалення точок відновлення. Крім того, за допомогою політик доступу до сховища резервних копій можна поділитися сховищем з іншими обліковими записами або користувачами AWS, виходячи з бізнес-потреб.

Щоб централізовано керувати дозволами, коли ваші робочі навантаження

масштабуються або мігруються до AWS, рекомендується використовувати Service Control Policies (SCPs). Це забезпечує додатковий рівень захисту і гарантує, що користувачі дотримуються встановлених політик доступу. Крім того, AWS IAM Access Analyzer допоможе виявити будь-які ролі IAM, які можуть бути доступні для зовнішніх об'єктів, таких як інші облікові записи AWS або анонімні користувачі.

2. Шифрування даних резервних копій та сховищ. Організаціям необхідно удосконалювати стратегії захисту даних і відповідати вимогам щодо захисту інформації, які виникають при масштабуванні в хмарі. Правильне впровадження методів шифрування забезпечує додатковий захисний рівень, знижуючи ризики при можливому порушенні політик доступу.

Для максимального захисту даних слід шифрувати їх як під час передачі, так і на зберіганні. AWS використовує Transport Layer Security (TLS) для захисту даних у транзиті та AWS Key Management Service (KMS) з алгоритмом AES-256 для шифрування на зберіганні. Конфігурації шифрування можуть відрізнятися в залежності від типу ресурсів і операцій резервного копіювання. AWS Backup Audit Manager допомагає перевірити правильність налаштування шифрування відповідно до вимог безпеки та нормативних стандартів.

Щоб забезпечити мульти-регіональну підтримку резервних копій, можна використовувати багаторегіональні ключі KMS для шифрування, що дозволяє дублювати дані між регіонами для відновлення після інцидентів.

3. Захист резервних копій за допомогою незмінного сховища. Незмінне сховище WORM (Write Once Read Many) дозволяє записати дані один раз і використовувати їх багаторазово без можливості видалення або зміни. Це допомагає захистити цілісність даних і запобігає видаленню або зміні даних під час зберігання. За допомогою AWS Backup Vault Lock можна запобігти видаленню резервних копій та змінам в налаштуваннях життєвого циклу, що особливо корисно для високорегульованих галузей з підвищеними вимогами до цілісності даних.

4. Моніторинг та сповіщення про резервні копії. Невдачі у завданнях резервного копіювання можуть вплинути на наступні процеси. Інтеграція AWS

Backup з іншими сервісами, такими як Amazon CloudWatch, Amazon EventBridge, AWS CloudTrail і Amazon SNS, дозволяє відслідковувати завдання резервного копіювання, генерувати оповіщення та автоматизувати розслідування інцидентів. AWS Backup Audit Manager автоматично створює звіти про виконання резервного копіювання для перевірки відповідності бізнес-вимогам.

5. Аудит конфігурації резервного копіювання. Організації повинні регулярно перевіряти відповідність політик резервного копіювання, таких як частота резервного копіювання. AWS Backup Audit Manager надає вбудовані шаблони для перевірки дотримання політик, а AWS Security Hub дозволяє стежити за відповідністю стандартам безпеки, що включає контроль за резервними копіями.

6. Тестування можливостей відновлення даних. Надійна стратегія резервного копіювання передбачає регулярне тестування відновлення даних. Рекомендується створити повторювані процеси тестування відновлення для підвищення впевненості у працездатності резервних копій. Це дозволяє виявити можливі проблеми до того, як відбудеться реальна потреба у відновленні даних.

7. Інтеграція резервного копіювання в план реагування на інциденти. Симуляції інцидентів, такі як Security Incident Response Simulations (SIRS), дозволяють організаціям перевірити ефективність планів реагування на інциденти. AWS Backup може автоматично створювати резервні копії, які можуть бути корисними для форензичного аналізу при розслідуванні інцидентів.

2.6.2 Захист резервних копій в Microsoft Azure

Забезпечення безпеки резервних копій даних у хмарному середовищі є критичним елементом стратегії кіберзахисту для будь-якої організації. Azure Backup надає різноманітні інструменти та механізми для захисту резервних копій, що дозволяють знизити ризик несанкціонованого доступу, забезпечити цілісність даних і гарантувати їх доступність під час відновлення після інцидентів. Нижче наведені ключові рекомендації з безпеки резервного копіювання в Azure, які допоможуть захистити дані та відповідати вимогам регуляторів.

1. Реалізація контролю доступу. Для захисту даних резервного копіювання в Azure необхідно забезпечити контроль доступу на основі ролей (RBAC). Azure Role-Based Access Control (RBAC) дозволяє надавати доступ до ресурсів лише тим користувачам, яким це необхідно для виконання їхніх обов'язків, відповідно до принципа мінімальних привілеїв. Встановлюючи ролі та дозволи для резервних копій, варто забезпечити, щоб кожен користувач або система мали тільки потрібні права.

Для централізованого управління доступом у масштабі всієї організації можна використовувати Azure Policy для створення та застосування правил доступу, що гарантує відповідність вимогам безпеки. Azure Active Directory (Azure AD) також забезпечує додатковий рівень захисту, включаючи багатофакторну автентифікацію для забезпечення безпечного доступу до ресурсів резервного копіювання.

2. Шифрування даних резервних копій. Щоб забезпечити безпеку резервних копій в Azure, необхідно застосувати шифрування як під час зберігання, так і під час передачі. Azure автоматично шифрує дані резервного копіювання на зберіганні за допомогою AES-256, забезпечуючи відповідність індустріальним стандартам шифрування. Для захисту даних під час передачі Azure використовує протоколи TLS, що забезпечує безпечну передачу даних між користувачем, додатками та службами резервного копіювання Azure.

Є можливість налаштувати Azure Key Vault для управління ключами шифрування, що забезпечує додатковий рівень захисту та дозволяє централізовано контролювати доступ до ключів. За допомогою Azure Key Vault можна використовувати свої власні ключі (BYOK) для шифрування даних резервного копіювання, що може бути необхідним для відповідності вимогам регуляторів.

3. Використання незмінного сховища. Azure пропонує незмінне сховище Immutable Blob Storage, яке дозволяє створювати резервні копії в режимі Write Once, Read Many (WORM), що захищає дані від випадкового або несанкціонованого видалення, перезапису або зміни. Це особливо корисно для захисту від атак з використанням зловмисного програмного забезпечення, таких як

програми-вимагачі.

4. Моніторинг та сповіщення. Забезпечення надійного моніторингу резервних копій є важливим елементом стратегії безпеки в Azure. За допомогою Azure Monitor можна налаштувати сповіщення про успішність або невдачу завдань резервного копіювання, а також відстежувати ключові метрики, такі як час резервного копіювання та використання сховища.

Інтеграція з Azure Security Center дозволяє отримувати централізовані оповіщення про можливі інциденти безпеки та проводити автоматичне розслідування інцидентів. Використання Azure Policy для моніторингу та контролю стану резервного копіювання дозволяє впевнитися, що всі резервні копії відповідають встановленим вимогам безпеки та нормативним стандартам.

5. Аудит конфігурації резервного копіювання. Azure Backup надає можливість проводити аудит конфігурацій резервного копіювання для перевірки відповідності політикам компанії. Azure Backup Reports дозволяє автоматично створювати звіти для відстеження всіх операцій резервного копіювання та перевіряти відповідність вимогам безпеки.

Azure Policy також дозволяє відстежувати конфігурацію резервного копіювання для впевненості, що всі політики дотримуються вимог безпеки. Застосування Azure Policy для моніторингу резервного копіювання допомагає ідентифікувати можливі невідповідності у резервному копіюванні та забезпечити відповідність політикам компанії.

6. Тестування можливостей відновлення даних. Для забезпечення надійності резервного копіювання варто регулярно тестувати відновлення даних. Azure Backup дозволяє проводити тестове відновлення даних для перевірки працездатності резервних копій. Регулярне тестування гарантує, що резервні копії можна буде відновити в разі потреби, забезпечуючи ефективність стратегії резервного копіювання.

7. Інтеграція резервного копіювання в план реагування на інциденти. Azure пропонує засоби для інтеграції резервного копіювання в план реагування на інциденти, включаючи Azure Security Center для організації Security Incident

Response Simulations (SIRS), що допомагає організаціям тестувати плани реагування на інциденти та аналізувати ефективність резервного копіювання в реальних сценаріях. Використання резервних копій в симуляціях інцидентів допомагає підвищити готовність до надзвичайних ситуацій, забезпечуючи оперативне відновлення даних.

2.6.3 Захист резервних копій в Google Cloud

Для захисту резервних копій у Google Cloud рекомендується використовувати комплексний підхід, який включає кілька ключових заходів. Ось основні кроки, які забезпечать безпеку резервних даних:

- Шифрування даних на всіх етапах: У Google Cloud за замовчуванням використовується шифрування даних як при зберіганні (AES-256), так і при передачі за допомогою SSL/TLS. Також можна використовувати власні ключі шифрування за допомогою Customer-Managed Encryption Keys (CMEK) для підвищення контролю над безпекою даних.
- Обмеження доступу та IAM: Впровадження контролю доступу за принципом найменших привілеїв (PoLP) допомагає зменшити ризики несанкціонованого доступу до резервних копій. Сервіси доступу налаштовуються через Cloud Identity та управління ролями IAM.
- Сегментація мережі та міжмережеві екрани: Використання Virtual Private Cloud (VPC) дозволяє сегментувати середовище та обмежувати доступ до критичних ресурсів. Налаштування правил міжмережевих екранів гарантує, що лише необхідний трафік може досягати ресурсів у вашій мережі.
- Автоматизовані резервні копії та відновлення: Google Cloud пропонує інструменти, такі як Cloud Storage та Cloud SQL, для автоматизації процесу резервного копіювання. Ці сервіси забезпечують зберігання резервних копій, що дозволяє швидке відновлення даних у разі збою.
- Моніторинг та виявлення загроз: Використання інструментів Cloud Monitoring та Cloud Logging допомагає відстежувати активність у реальному часі,

а також виявляти підозрілу активність, що може свідчити про спроби злому або інші інциденти.

Такі практики сприяють захисту резервних копій, зменшуючи ризик витоків та забезпечуючи відповідність вимогам до безпеки даних. Для отримання більш детальної інформації про рекомендації щодо безпеки в Google Cloud можна звернутися до посібників на офіційних сайтах і оглядових статей від партнерів Google. [10][11]

2.7 Керування ідентифікацією та доступом (IAM)

Керування ідентифікацією та доступом (IAM) є важливим компонентом кібербезпеки, який контролює, хто має доступ до яких ресурсів у хмарному середовищі та з якими дозволами. IAM допомагає створювати детальні політики доступу, щоб користувачі могли отримати лише той рівень доступу, який необхідний для виконання їхніх обов’язків, що значно знижує ризики несанкціонованого доступу та забезпечує відповідність вимогам безпеки. IAM є невід'ємною частиною будь-якої хмарної платформи, оскільки дозволяє централізовано керувати доступом для різних користувачів, груп та ролей.

IAM (Identity and Access Management) — це система, яка дозволяє компаніям створювати, керувати та контролювати доступ до ресурсів. Вона включає функціональність для ідентифікації користувачів, аутентифікації (верифікації особи) та авторизації (визначення рівня доступу). Використовуючи IAM, організації можуть створювати ролі та групи з різними правами, що дозволяє уникнути надмірних привілеїв та забезпечує безпеку даних. [16]

Кожна з основних хмарних платформ — AWS, Azure та Google Cloud — має власну IAM систему з різними можливостями та практиками налаштувань.

В AWS IAM (Identity and Access Management) надає можливість створювати користувачів, групи, ролі та політики, які визначають доступ до ресурсів AWS.

Основні принципи безпеки IAM для AWS:

1. Принцип мінімальних привілеїв: користувачам і ролям надаються лише

ті дозволи, які необхідні для виконання їхніх завдань. Наприклад, можна налаштувати права для групи розробників, щоб вони могли запускати, але не видаляти ресурси.

2. Мультифакторна аутентифікація (MFA): для захисту критично важливих ресурсів рекомендується використовувати MFA для адміністративних акаунтів.

3. Управління ролями для сервісів: для автоматизації використовується спеціальний тип ролей, які надають доступ сервісам (наприклад, EC2 для доступу до S3), а не користувачам.

4. Моніторинг і аудит: за допомогою AWS CloudTrail можна відстежувати дії, які виконуються з використанням IAM ролей і політик.

Приклад налаштування для адміністраторів застосунку:

- Група адміністраторів застосунку може мати повний доступ до певних сервісів (наприклад, до S3, Lambda, RDS) з обмеженням на створення та видалення інфраструктурних ресурсів (наприклад, обмеження на IAM та VPC).
- Вони можуть отримати дозволи на перегляд журналів через CloudWatch та доступ до ресурсів у специфічних VPC.
- Додатково, для підвищення безпеки, MFA обов’язковий для всіх користувачів цієї групи.

Azure Active Directory (Azure AD) — це служба ідентифікації та управління доступом від Microsoft, яка пропонує широкий спектр інструментів для захисту доступу до ресурсів. Azure AD інтегрується з багатьма іншими сервісами Microsoft та дозволяє централізовано управляти доступом для користувачів і додатків.

Основні принципи безпеки IAM для Azure:

- Role-Based Access Control (RBAC): Azure використовує рольову модель контролю доступу, де дозволи надаються на основі ролей (наприклад, Читач, Співробітник з безпеки, Власник).
- Identity Protection: для захисту облікових записів використовується виявлення підозрілої активності, зокрема атипових входів або змін у профілях.
- Just-In-Time (JIT) доступ: дозволяє надавати тимчасовий доступ до

ресурсів через Privileged Identity Management (PIM), що знижує ризик компрометації.

- Access Reviews: регулярний аудит доступів для підтвердження відповідності дозволів поточним вимогам безпеки.

Приклад налаштування для розробників програми:

- Група розробників може отримати роль Співробітника на рівні ресурсної групи, що дозволяє створювати і змінювати ресурси в межах своєї програми.
- Адміністратори можуть надати їм тимчасовий доступ до певних адміністративних функцій через JIT.
- Після завершення розробки доступ обмежується до ролі Читача для перегляду і моніторингу додатка, що захищає продуктивне середовище від змін.

Google Cloud використовує Cloud Identity and Access Management (IAM) для надання детального контролю доступу до своїх сервісів. Цей інструмент підтримує гнучку систему призначення ролей і політик, що дозволяє точно налаштовувати дозволи для користувачів і сервісних облікових записів.

Основні принципи безпеки IAM для Google Cloud:

- Least Privilege Principle: мінімізація доступу для кожного користувача або сервісного облікового запису, що обмежує ризик несанкціонованого доступу.
- Custom Roles: Google Cloud дозволяє створювати власні ролі для задоволення унікальних вимог безпеки, що виходять за рамки стандартних ролей.
- Service Accounts: спеціальні облікові записи для застосунків та сервісів, які автоматично використовують права доступу.
- IAM Recommender: інструмент, який надає рекомендації для оптимізації дозволів, що базується на поведінці користувачів.

Приклад налаштування для операторів системи:

- Група операторів може мати доступ до ролей Editor для певних ресурсів (наприклад, BigQuery), що дозволяє їм створювати, оновлювати і переглядати записи даних.
- Вони можуть мати роль Viewer для ресурсів продуктивного

середовища, що обмежує можливість змін, але дозволяє моніторинг.

- Усі сервісні облікові записи обмежені до мінімальних привілеїв за допомогою рекомендацій від IAM Recommender, що знижує ризик надлишкового доступу.

Другий розділ присвячений огляду хмарних технологій та їх застосуванню для відновлення після інцидентів. Хмарні платформи, такі як AWS, Azure та Google Cloud, надають широкий спектр інструментів для автоматизованого резервного копіювання та відновлення. Використання хмарних сервісів дозволяє значно скоротити час простою систем, мінімізувати втрати даних та забезпечити надійність процесу відновлення. Особлива увага приділена контейнеризації та оркестрації, які забезпечують гнучкість та масштабованість сучасних IT-інфраструктур.

РОЗДІЛ 3 ПОБУДОВА АВТОМАТИЗОВАНОЇ СИСТЕМИ ВІДНОВЛЕННЯ ПІСЛЯ КІБЕРІНЦИДЕНТІВ В РАМКАХ DDOS АТАКИ

3.1 Проектування архітектури автоматизованої системи

Процес побудови автоматизованої системи відновлення після кіберінцидентів, що виникають унаслідок DDoS-атак, передбачає розроблення гнучкої, масштабованої та надійної архітектури, яка зможе ефективно протидіяти загрозам та забезпечувати відновлення з мінімальними втратами. У контексті DDoS-атак центральним завданням є своєчасне виявлення аномального трафіку, налаштування коректної маршрутизації трафіку й оперативне повернення інфраструктури до робочого стану.

У межах побудови автоматизованої системи відновлення після DDoS-інцидентів було обрано гібридний підхід, що поєднує корпоративний дата-центр та інфраструктуру в Amazon Web Services (AWS). Така модель дає змогу оптимізувати як існуючі ресурси дата-центру, так і хмарні сервіси AWS для забезпечення відмовостійкості й безпеки.

Гібридна хмарна модель передбачає, що основні бізнес-сервіси функціонують у власній (он-преміс) інфраструктурі, але критичні вузли або додаткові резервні компоненти розміщено в AWS. Зокрема, у хмарі реалізується автоматизована система відновлення з можливістю масштабування та інтегрованим захистом від DDoS. Передбачено також безперервний обмін даними між локальною мережею та AWS за допомогою приватного підключення (AWS Direct Connect).

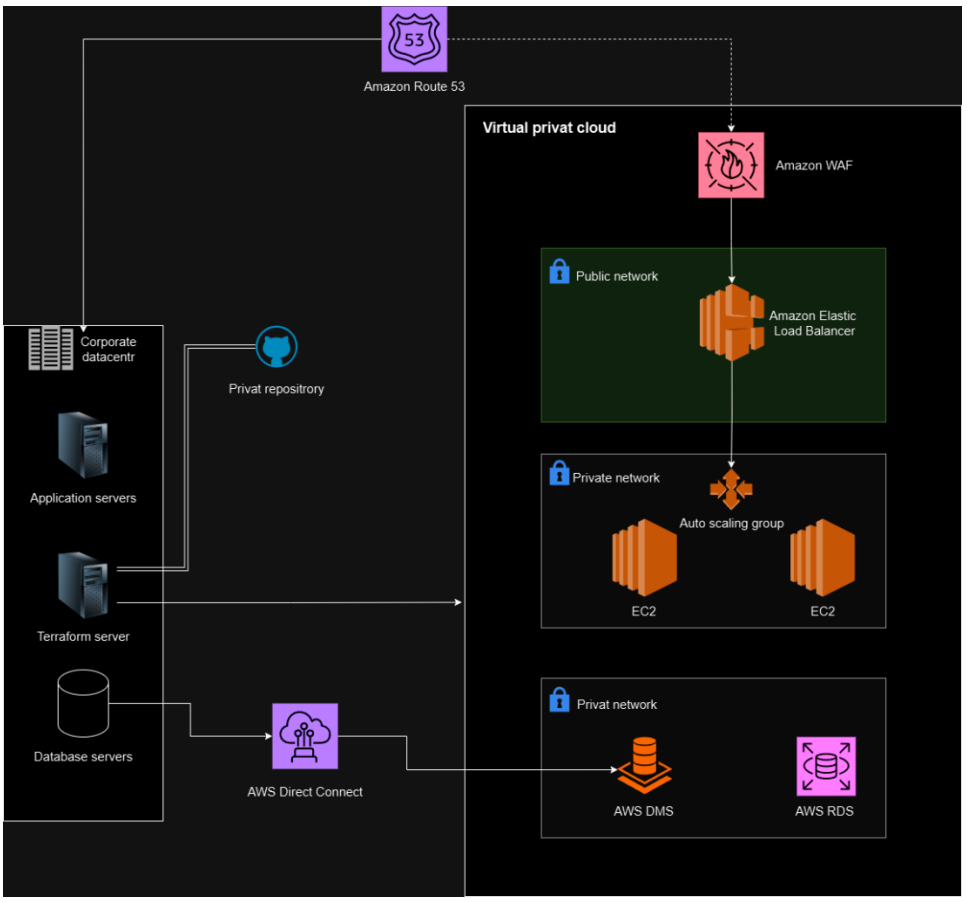

Рисунок 3.1 – Архітектура автоматизованої системи відновлення після кіберінциденту

Зображена на схемі архітектура (рис. 3.1) містить ключові підсистеми й сервіси, необхідні для побудови автоматизованого відновлення після DDoS-інцидентів та для підтримки стабільної роботи застосунків:

1. Корпоративний дата-центр

- Application Servers (сервери додатків): основне середовище запуску бізнес-логіки, вебсервісів чи корпоративних застосунків.
- Terraform Server: призначений для керування інфраструктурою як кодом (IaC). Через нього виконуються сценарії автоматизованого розгортання, оновлення та відновлення як у локальній мережі, так і в AWS.
- Database Servers (локальні сервери баз даних): основне сховище критичних бізнес-даних та журналів системи. Виконують роль майстер-репозиторію для подальшої реплікації в хмару.
- Privat Repository (приватний репозиторій): Git-сховище, де

зберігаються скрипти Terraform, файли конфігурацій та програмний код застосунків.

2. AWS Direct Connect

- Забезпечує виділений канал зв’язку між корпоративним дата-центром та AWS. Цей підхід знижує залежність від публічних інтернет-з’єднань, покращує пропускну здатність і підвищує безпеку передачі даних.

3. Amazon Route 53

- Сервіс DNS, що маршрутизує вхідний вебтрафік з усього світу до потрібних ресурсів AWS. Також може виконувати Health Checks і автоматичне перемикання (failover) на резервні сервіси в разі збою чи DDoS-атаки.

4. Amazon WAF (Web Application Firewall)

- Фаєрвол на 7-му рівні, розміщений перед вебресурсами. Виявляє та блокує типові атаки (SQL-ін’єкції, XSS), а також частково фільтрує HTTP/HTTPS DDoS-трафік.

- Розміщується над балансувальником навантаження (Amazon Elastic Load Balancer), забезпечуючи додатковий рівень захисту безпосередньо перед доступом до EC2-інстансів.

5. Amazon Elastic Load Balancer (ELB)

- Спрямовує вхідні HTTP/HTTPS-запити з WAF на пул EC2-екземплярів у Auto Scaling Group.

- Допомагає розподіляти та балансувати трафік, тим самим знижує ризик відмови окремого вузла під час високого навантаження або DDoS-спроб.

6. Auto Scaling Group

- Набір EC2-екземплярів, де автоматично збільшується чи зменшується кількість інстансів залежно від метрик навантаження. Якщо відбувається DDoS-атака, ASG може динамічно розширитися, зберігаючи стабільність сервісів.

- Тісно інтегрований із Terraform Server: щойно ініціюється подія масштабування, Terraform вносить необхідні зміни до конфігурації інфраструктури (додає EC2, оновлює Security Groups тощо).

7. AWS DMS (Database Migration Service)

- Інструмент для реплікації та міграції даних з локальної бази (Database Servers) до AWS RDS. Дає можливість підтримувати копію даних у хмарі, актуальну для аварійного перемикання (Disaster Recovery).

8. AWS RDS (Relational Database Service)

- Реляційна база даних у хмарі (MySQL, PostgreSQL, Oracle чи інші). Працює у приватній підмережі VPC, не доступній ззовні напряму.

- Може виступати в ролі «Slave»-репліки чи запасного майстер-сервера, щоб у разі виходу з ладу локальної БД забезпечити швидке відновлення працездатності.

9. Virtual Private Cloud (VPC) із зонуванням мереж

- Public network: підмережа для ресурсів, до яких є публічний доступ (Amazon WAF, Elastic Load Balancer).

- Private network: підмережа для EC2 та баз даних (RDS). Вона закрита від зовнішнього доступу та забезпечує більший рівень безпеки.

Надалі будуть наведені рекомендації щодо налаштувань для виконання алгоритму відпрацювання системи у розділі 3.2.

3.1.1 Route 53 для маршрутизації трафіку

Amazon Route 53 — це масштабований і високодоступний DNS-сервіс від AWS, що керує доменними іменами й маршрутизацією мережевого трафіку на глобальному рівні. У контексті гібридної хмарної архітектури він відповідає за:

1. Розподіл навантаження між корпоративним дата-центром та AWS(наприклад, напрям потоку запитів на вебсервери або балансувальники).

2. Failover-механізм — автоматичне переключення DNS-записів на резервну інфраструктуру, якщо основний вузол недоступний.

3. Інтеграцію з IaC (Infrastructure as Code): Terraform може автоматизовано змінювати Route 53 записи (A/CNAME), тим самим перенаправляючи трафік з одного Load Balancer на інший.

Отже, Route 53 виконує центральну роль у рішенні, коли Terraform playbook визначає, на який балансувальник чи інфраструктуру слід спрямовувати запити. Якщо основний вузол виходить з ладу (наприклад, під час DDoS-атаки), Terraform оновлює Route 53, і DNS-записи вказують на резервний Load Balancer в AWS.

Code example

Пояснення основних блоків:

1. Перемінні:

- domain_name та route53_zone_id зберігають назву домену й ідентифікатор (Hosted Zone ID) в Route 53.

2. aws_route53_record

- Запис lb_record створює або оновлює ресурс типу A для заданого домену.

- Використовує alias на Load Balancer: aws_lb.my_lb.dns_name та aws_lb.my_lb.zone_id.

- evaluate_target_health = true дозволяє Route 53 автоматично визначати стан доступності LB.

3. Failover Routing (опціонально)

- Приклад налаштування failover (PRIMARY / SECONDARY), коли один запис вважається основним, а інший – резервним.

- Health Check переконується, що основний LB відповідає, і якщо він стає недоступним, Route 53 автоматично переключить трафік на Secondary Record.

Таке рішення дає змогу Terraform змінювати DNS-записи в Route 53 та прямує трафік безпосередньо на Load Balancer, який був попередньо створений в іншому playbook чи в тому ж проєкті.

3.1.2 AWS Direct Connect для швидкого підключення до хмарної інфраструктури

AWS Direct Connect – це виділений фізичний канал зв’язку між вашим

локальним дата-центром та інфраструктурою Amazon Web Services. Він надає більш стабільне та безпечне з'єднання із нижчою затримкою порівняно зі звичайним виходом в інтернет. Нижче наведено детальніший огляд ролі Direct Connect та його принципів роботи.

1. Призначення та роль у гібридній архітектурі

- Прискорена взаємодія з хмарними ресурсами: Завдяки виділеній лінії, дані передаються напряму до AWS без проходження через загальний інтернет; це особливо важливо для гібридних рішень, коли частина сервісів працює в локальному дата-центрі, а частина – у хмарі.
- Гарантована пропускна здатність: Ви обираєте потрібну швидкість каналу (від 50 Mbps до 100 Gbps, залежно від потреб), що забезпечує стабільну пропускну спроможність для критично важливих навантажень або реплікації баз даних.
- Зниження витрат на трафік: Вихід на AWS через Direct Connect може бути економічно вигіднішим для великих обсягів даних, оскільки внутрішні тарифні плани AWS відрізняються від стандартних "інтернет"-тарифів.
- Безпека та контроль: Завдяки приватному каналу дані не подорожують інтернетом, що зменшує ризик перехоплення чи втрати. Ви також маєте гнучкі можливості контролю трафіку (BGP, власні ACL тощо).
- Інтеграція з Terraform: Terraform playbook може автоматично налаштовувати й оновлювати конфігурації AWS (VPC, маршрути) та вашого локального середовища. Таким чином, коли Terraform розгортає нові EC2-інстанси чи Load Balancer, трафік до них "пройде" через Direct Connect, мінімізуючи затримку та знижуючи ризик перевантаження від DDoS-сплесків в інтернет-каналі.

2. Як функціонує AWS Direct Connect

- Фізичне підключення:
  - Орендується приватна лінія зв'язку від вибраного телеком-провайдера, яка з'єднує ваш дата-центр з одним із Direct Connect

Location (спеціальних точок присутності AWS).

- На кожному кінці лінії налаштовується відповідне мережеве обладнання (маршрутизатори), що підтримує BGP (Border Gateway Protocol).
- Логічна конфігурація
- У AWS створюється Direct Connect Gateway або Virtual Private Gateway (VGW), який буде “дверима” до вашої Virtual Private Cloud (VPC).
- Між локальним маршрутизатором і VGW встановлюється BGP-сесія. Таким чином, приватні підмережі (CIDR-діапазони) вашої VPC та локальної мережі анонсуються одна одній автоматично.
- VLAN/Virtual Interface (VIF)
- Залежно від задач, налаштовуються Public VIF, Private VIF або Transit VIF. Наприклад, Private VIF дає змогу підключатися до приватних ресурсів у VPC (EC2, RDS), а Public VIF дозволяє досягати публічних сервісів AWS (S3, DynamoDB) без “виходу” в інтернет.
- Transit VIF використовується для з’єднання з AWS Transit Gateway, якщо потрібно реалізувати складніші маршрутизаційні сценарії з кількома VPC.
- Маршрутизація та політики безпеки
- У VPC налаштовуються маршрути, вказуючи, що частина CIDR-блоків (наприклад, 10.0.0.0/16) досяжна через Direct Connect Gateway.
- Можна застосовувати Network ACL та Security Groups для регулювання inbound/outbound-трафіку з локального середовища до хмари.
- Інтеграція з Terraform
- Terraform може створювати aws_dx_connection, aws_dx_gateway, aws_dx_gateway_association та налаштовувати BGP-параметри. Коли конфігурація змінюється, Terraform оновлює всі необхідні ресурси автоматично.
- Це особливо корисно для сценаріїв Disaster Recovery, коли

нові маршрути чи підключення потрібно розгорнути в стислі терміни.

3. Переваги використання Direct Connect для перенаправлення трафіку

- Вища пропускна здатність та низькі затримки: Забезпечує швидку синхронізацію даних між локальними базами й AWS (через DMS або інші механізми), а також плавну роботу застосунків.
- Сталий канал при DDoS-атаках: Трафік до AWS не йде “через інтернет”, тому за умови DDoS на загальнодоступний канал ваш приватний лінк залишається доступним.
- Легка автоматизація: Уся конфігурація Direct Connect, VIF та BGP-сесій може бути визначена інфраструктурою як кодом (Terraform), що полегшує зміну маршрутів і швидке розгортання нового середовища.
- Безпека даних: Трафік маршрутизується напряму, без проходження через треті мережі. Можна додатково шифрувати передані дані (наприклад, за допомогою VPN over DX) для посилення захисту.

Таким чином, AWS Direct Connect виступає ключовим “містком” між локальним дата-центром і хмарними ресурсами, надаючи стабільний канал з передбачуваною пропускною здатністю. Завдяки цьому Terraform playbook може оперативно змінювати конфігурації, перенаправляти трафік і виконувати оновлення інфраструктури без затримок і ризиків, пов’язаних з інтернетом.

3.1.3 AWS DMS для міграції даних

AWS DMS (Database Migration Service) – це сервіс для безперервної міграції й реплікації баз даних між локальним середовищем і хмарою AWS. Він дає змогу мінімізувати час простою під час перенесення критичних даних, а також забезпечувати актуальність копії (або резерву) в хмарі для сценаріїв відмовостійкості та Disaster Recovery.

Основна ідея сервісу:

- AWS DMS — це керована (managed) служба від Amazon, яка спрощує перенесення даних між різними базами даних або середовищами (онпрем чи

хмарними).

- Використовується для міграції (переміщення) сховищ: від старих локальних або хмарних БД на RDS або Aurora (чи навпаки), або від одного типу бази даних (наприклад, Oracle) до іншого (наприклад, PostgreSQL).

- Дає змогу мінімізувати простої застосунків, оскільки дозволяє виконувати реплікацію даних у реальному часі, роблячи так звану міграцію з невеликими або нульовими простоями (Minimal/Zero Downtime Migration).

Як працює AWS DMS:

1. Створюється реплікаційна «задача» (migration task) в консолі AWS DMS.

2. Джерело (source): це може бути база даних або інше сховище даних (наприклад, MongoDB, MySQL, SQL Server, PostgreSQL, Oracle та ін.).

3. Ціль (target): зазвичай це один з сервісів AWS типу RDS (Amazon Aurora, MySQL, PostgreSQL), Redshift або DynamoDB. Також може бути просто EC2 із встановленою базою або інше сховище.

4. DMS-проксі (Data Migration Service Replication Instance): служба розгортає екземпляр-реплікатор, який зчитує дані з джерела й копіює їх у ціль. Може здійснювати повне копіювання на старті міграції, а потім відслідковувати й довантажувати зміни (change data capture, CDC) без зупинки основної БД.

5. Перевірка та моніторинг: якщо потрібна додаткова трансформація даних або валідація, налаштовуються відповідні параметри в DMS. Також доступні логи, CloudWatch метрики для моніторингу, Health Checks тощо.

У гібридній архітектурі DMS відіграє роль “містка” між локальними СУБД і хмарними базами даних (AWS RDS):

- Реплікує дані у реальному часі з он-преміс (on-prem) у хмару, підтримуючи синхронізацію двох середовищ.

- Забезпечує аварійне перемикання (failover), коли Terraform може оновлювати конфігурацію DMS або саму інфраструктуру баз даних, переключаючи застосунки на хмарну RDS.

Якщо Terraform playbook ініціює зміну endpoint (наприклад, змінює DNS-

запис чи параметри підключення), то “трафік” запитів до БД автоматично скеровується з локальної бази на хмарний екземпляр, який DMS тримає в актуальному стані. Таким чином DMS, керований Terraform, стає ключовим механізмом підтримки синхронності й готовності хмарної бази даних до повного або часткового переносу робочих навантажень в AWS.

3.1.4 Elastic Load Balancer для забезпечення доступності сервісів

Elastic Load Balancer (ELB) – це керована служба балансування навантаження, що в гібридній архітектурі відповідає за прийом вхідних запитів (зокрема, після їх фільтрації через AWS WAF) та рівномірний розподіл їх між екземплярами, які входять до Auto Scaling Group.

Основні аспекти роботи ELB:

1. Керованість: AWS відповідає за масштабування, оновлення та підтримку інфраструктури балансувальника. Це знімає з вас тягар адміністрування окремих «Load Balancer»-сервісів чи власних балансувальних серверів.

2. Типи балансувальників:

- Application Load Balancer (ALB) — працює на 7-му рівні моделі OSI (L7), що дає змогу маршрутизувати запити за шляхом (path-based routing), заголовками (header-based routing) чи значеннями cookies (host-based routing). Відмінно підходить для мікросервісної архітектури та контейнерів (ECS/EKS).
- Network Load Balancer (NLB) — працює на 4-му рівні моделі OSI (L4), забезпечуючи дуже високу пропускну здатність і низькі затримки під час балансування трафіку TCP або UDP. Підходить для низькорівневих протоколів або високонавантажених систем.
- Classic Load Balancer (CLB) — застаріла версія ELB (попередній тип), який працював як на L4, так і частково на L7, але має обмежений функціонал порівняно з ALB/NLB. Сьогодні зазвичай не рекомендується для нових проєктів.

3. Забезпечення «високої доступності» та відмовостійкості: ELB автоматично розподіляє трафік між Availability Zones (AZs) у регіоні AWS. Якщо один із екземплярів або навіть одна з AZ вийде з ладу, балансувальник автоматично перенаправить запити на здорові інстанси.

4. Інтеграція з Auto Scaling Group

- Коли Auto Scaling Group (ASG) додає або видаляє EC2-екземпляри, вони автоматично реєструються / відреєструються в ELB (при відповідній конфігурації Target Group).
- Завдяки цьому додавання нових серверів або видалення зайвих відбувається безперешкодно для користувачів, що підвищує гнучкість системи й дозволяє автоматично нарощувати або зменшувати потужність залежно від навантаження.

5. Здоров’я та Health Checks: ELB постійно виконує перевірки на здоров’я (health checks) зареєстрованих ресурсів. Якщо інстанс відповідає некоректно або недоступний, балансувальник припиняє направляти до нього трафік, забезпечуючи безперебійне обслуговування клієнтів.

6. Безпека

- Тісна інтеграція з AWS WAF (у випадку ALB) дає змогу блокувати або фільтрувати шкідливий трафік ще до потрапляння в Auto Scaling Group.
- Підтримка SSL/TLS дає змогу встановлювати HTTPS-з'єднання й шифрувати трафік до/від балансувальника.

7. Гнучкість маршрутів і правила: Зокрема, Application Load Balancer дає розширені можливості маршрутизації: можна створювати правила (Listener Rules), що визначають, куди спрямовувати запити — до різних Target Groups (EC2, Lambda, IP-адреси, контейнери, Fargate), залежно від URL, хедерів або типу запиту.

8. Логування та моніторинг

- ELB генерує Access Logs, які можна зберігати в Amazon S3 і аналізувати (наприклад, в Amazon Athena чи Splunk).

- Показники моніторингу (CloudWatch Metrics) дозволяють відстежувати продуктивність — кількість запитів, затримку (latency), помилки (HTTP 4xx/5xx) тощо.

lastic Load Balancer (ELB) є центральною ланкою у гібридній хмарній архітектурі, що забезпечує надійний розподіл навантаження та високу доступність застосунків. Завдяки інтеграції з іншими сервісами AWS (Auto Scaling Group, Route 53) та інструментами інфраструктури як код (Terraform), ELB стає гнучким "маршрутизатором" трафіку, який може автоматично масштабуватися в пікові періоди та ефективно обробляти потенційні збої чи атаки. Нижче наведено ключові сценарії й ролі ELB, які ілюструють його важливість у сучасних гібридних середовищах.

1. Роль у гібридній архітектурі:

- ELB працює у публічній підмережі VPC, передаючи трафік на приватні EC2-екземпляри в Auto Scaling Group.Інтегрується з Route 53 і може бути кінцевою точкою (Alias Target) для DNS-записів.

- Інтегрується з Route 53 і може бути кінцевою точкою (Alias Target) для DNS-записів.

2. Взаємодія з Auto Scaling Group:

- Коли Terraform або Auto Scaling додає нові EC2-екземпляри, вони автоматично реєструються в ELB; якщо примірник стає недоступним, ELB вилучає його з активного пулу.

- Це забезпечує безперервну доступність сервісів: під час пікових навантажень нові інстанси динамічно додаються, а при спаді – виводяться з експлуатації.

3. Перенаправлення трафіку через Terraform:

- Terraform може централізовано конфігурувати ELB, зокрема змінювати Health Check, протоколи (HTTP/HTTPS), SSL-сертифікати й цілі (Target Groups).

- У разі DDoS-атаки або оновлення інфраструктури Terraform

змінює параметри ELB та маршрути, перенаправляючи трафік на нові або резервні EC2-інстанси в ASG.

Завдяки ELB та Auto Scaling Group, система лишається відмовостійкою і гнучко реагує на зміну навантаження, водночас підвищуючи рівень захищеності від мережевих інцидентів.

3.1.5 Auto Scaling Group для динамічного масштабування

Auto Scaling Group (ASG) – це сервіс AWS, що динамічно регулює кількість EC2-екземплярів залежно від заданих метрик (наприклад, завантаження процесора чи мережевого трафіку). У моїй гібридній архітектурі ASG перебуває у приватній підмережі за Elastic Load Balancer та інтегрується з Route 53 і Terraform. Коли попит на сервіс зростає (наприклад, під час DDoS-сплеску), ASG додатково “піднімає” інстанси; при спаді навантаження зайві ресурси автоматично виводяться.

Разом ці механізми формують масштабовану й відмовостійку систему, що підлаштовується до реальних потреб бізнесу. Гнучке масштабування дає змогу швидко додавати чи вилучати екземпляри залежно від поточної метрики навантаження: під час піків (наприклад, коли DDoS спричиняє велике надходження трафіку) ASG збільшує пул EC2, а при зниженні навантаження — навпаки, оптимізує витрати, вивільняючи непотрібні ресурси.

Завдяки тісній інтеграції з Elastic Load Balancer запити завжди перенаправляються лише до здорових (healthy) інстансів, отже користувачі не відчувають перебоїв у разі відмови окремих вузлів системи. При цьому Terraform використовується для опису «інфраструктури як коду» — параметрів ASG (мінімальна та максимальна кількість інстансів, політики масштабування тощо). У сценаріях Disaster Recovery (DR) Terraform здатен оперативно розгорнути ASG у резервному регіоні або оновити конфігурацію в чинному, що пришвидшує відновлення й мінімізує час простою.

Роль ASG в архітектурі:

1. Гнучке масштабування: автоматичне розширення пулу EC2 під час пікових навантажень й оптимізація витрат при зниженні трафіку.

2. Перенаправлення трафіку: разом із ELB, ASG забезпечує, щоб запити спрямовувались виключно на “живі” інстанси.

3. Terraform-інтеграція: визначення параметрів ASG (розмір, політики масштабування) як коду. У сценарії Disaster Recovery (DR) Terraform може розгорнути ASG у резервному регіоні чи змінити конфігурацію в поточному.

3.1.6 Інтеграція WAF для захисту від DDoS атак

AWS WAF (Web Application Firewall) – це хмарне рішення для захисту вебзастосунків та API від поширених загроз, включно з DDoS-навантаженням на рівні HTTP/HTTPS (Layer 7). У вибраній вами гібридній архітектурі WAF розташовано перед Elastic Load Balancer, де відбувається первинна фільтрація шкідливого трафіку.

Роль в архітектурі:

- Фільтрація Layer 7: AWS WAF блокує чи обмежує запити, що відповідають певним правилам (SQL-ін’єкції, XSS, бот-трафік, тощо). При DDoS-атаках на HTTP-рівні він може згладжувати піки навантаження ще до потрапляння запитів на Auto Scaling Group.

- Інтеграція з Terraform: Терраформ-скрипт може визначати й оновлювати правила (AWS WAF WebACL), прикріплювати WAF до ALB/CloudFront, а також у DR-сценарії перевести трафік на резервний Load Balancer, де теж активні подібні чи ідентичні WAF-налаштування.

Перенаправлення трафіку:

- Якщо стається відмова в первинному регіоні, Terraform DR-плейбук оновлює Route 53, ASG, RDS та WAF-політику в DR-регіоні. Відтак клієнти автоматично спрямовуються на резервну інфраструктуру з діючим WAF-налаштуванням.

Модулі які використовуються в роботі WAF модулю:

1. Primary WAF:

- aws_wafv2_web_acl.primary_waf із базовим правилом rate limiting та передбаченим блокуванням “зайвого” трафіку. WebACL прив’язано до aws_lb.primary_lb.

2. DR WAF:

- Аналогічні правила, але в DR-регіоні (aws_wafv2_web_acl.dr_waf).

Прив’язка dr_waf_alb_assoc до aws_lb.dr_lb.

3. DR-активація:

- Коли включається DR (змінна dr_mode = true), Terraform піднімає Auto Scaling Group, Load Balancer і WAF у DR-регіоні.

- Terraform також може оновити Route 53 (Failover Routing чи звичайний CNAME), щоб спрямувати запити на dr_lb.

- Отже, запити фільтруються через DR WAF, забезпечуючи аналогічний захист, як у Primary.

Таким чином, AWS WAF у кожному регіоні захищає додатки на рівні L7 від DDoS та інших типових вебзагроз. Завдяки Terraform ви автоматизуєте розгортання Web ACL, керування правилами та сценарії “Disaster Recovery”, забезпечуючи відмовостійкість і безпеку застосунків.

3.1.7 Використання Terraform для автоматизації процесі

Terraform – це інструмент Infrastructure as Code (IaC), що дає змогу централізовано описувати, розгортати й оновлювати інфраструктурні ресурси як у локальній мережі, так і в AWS. Натомість Git слугує системою керування версіями, яка зберігає Terraform-конфігурації та усі зміни (commit) в приватному репозиторії. Такий підхід дозволяє:

1. Відстежувати зміни в коді конфігурацій та повертатися до стабільних версій.

2. Уникати помилок і “drifty” в інфраструктурі: під час виконання

terraform plan/apply відразу видно, що саме зміниться.

3. Безпечно зберігати секрети й доступи у приватному Git-репозиторії (зокрема, використовувати GitLab чи GitHub Enterprise із шифруванням або Vault).

У гібридній архітектурі, де сервери та бази даних частково розташовані в корпоративному дата-центрі, а частково в AWS, Git + Terraform спрощує керування як локальними ресурсами (через відповідні Terraform-провайдери), так і хмарними сервісами. Для сценарію Disaster Recovery (DR) в Terraform-залежностях зберігаються конфігурації Auto Scaling Group, WAF, RDS, Route 53 та інших компонентів — при потребі система розгортає резервні копії інстансів та змінює DNS-записи.

Припустимо, у приватному репозиторії Git зберігається структура проекту, для цього оглянемо рисунок 3.2.

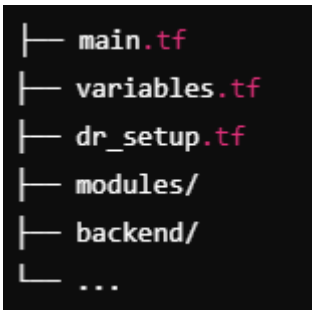

Рисунок 3.2 - Варіант організації terraform конфігураційних файлів

Наступні компоненти містяться в репозиторії:

1. main.tf: Описує провайдери AWS (primary та DR-регіон), налаштовує бекенд (S3 або локальний для state).

2. dr_setup.tf: Містить ресурси, що потрібно розгортати під час перемикання на DR (Auto Scaling Group, WAF, Load Balancer у резервному регіоні).

3. variables.tf: Зберігає вхідні параметри (домен, регіони, dr_mode = true/false).

При dr_mode = true, Terraform застосовує ті чи інші налаштування (наприклад, збільшує desired_capacity у DR-регіоні з 0 до 2).

Щоб виконувати Terraform-скрипти, потрібен Terraform Server

(розміщений у корпоративному дата-центрі):

1. CPU: 4 vCPU (для одночасної обробки декількох планів розгортання)
2. RAM: від 8 ГБ (Terraform може бути досить ресурсомістким при великих інфраструктурах)
3. OS: Linux-сервер (наприклад, Ubuntu 20.04 LTS або Amazon Linux 2). Актуальні оновлення важливі для безпеки.
4. IP Access: дозволити вхід (Inbound) лише на:
   - порт 22 (SSH) з корпоративного VPN/white-list IP,
   - опційно порт 443 (HTTPS) якщо планується власний інтерфейс/CI.
   - додатково – під'єднання до Git-репозиторію (SSH ключі чи HTTPS токен), закрите будь-яким firewall.
5. Disk Storage: SSD на ~50–100 ГБ: цього вистачить для state-файлів, кешу та логів. Якщо передбачено централізований бекенд (наприклад, S3), то локальний диск може бути мінімальним.

Підтримувати Terraform і Git-сервер на відокремлених інстансах з бекапами state-файлів. Це знизить ризики втрати керування інфраструктурою й мінімізує наслідки DDoS-атак або внутрішніх збоїв.

Підсумовуючи: Із використанням Terraform + Git у гібридній моделі, усі конфігурації та DR-сценарії зберігаються централізовано й прозоро, а само розгортання й маршрутизація (Auto Scaling, WAF, Route 53, RDS) виконуються за кілька хвилин завдяки IaC. Сервер під Terraform має бути досить продуктивним, з захищеним SSH-доступом та регулярними бекапами, щоб гарантовано забезпечувати безперебійну роботу інфраструктури.

Як результат, така архітектура поєднує гнучкість і масштабованість хмарних сервісів зі збереженням контролю над власною інфраструктурою, водночас забезпечуючи відмовостійкість і швидке відновлення сервісів після DDoS-атаки. Це відповідає завданням розділу 3, де необхідно було передбачити механізми автоматизованої реакції (автоскейлінг, WAF), відновлення (Terraform, резервний RDS). Отже, наведений дизайн повністю підкріплює стратегію безпеки та мінімізує

ризик простоїв і втрати даних у разі потужних мережевих атак.

3.2 Алгоритм відпрацювання процесу відновлення

Нижче наведено детальний алгоритм відновлення (DR-процес), який враховує всі компоненти гібридної архітектури та їх залежності. Логіка описує, як система автоматично перебудовується та переключає трафік у разі збою або DDoS-атаки.

Алгоритм відновлення:

1. Виявлення інциденту
   - Моніторинг (CloudWatch, Health Checks, SIEM) фіксує аномалію: DDoS-атака або збій основної інфраструктури.
   - Спрацьовують тригери: наприклад, Health Check у Route 53 переходить у “Failed” або Terraform отримує зовнішній сигнал про DR-режим (dr_mode = true).
2. Ініціація сценарію DR
   - Запускаються Terraform playbook чи CI/CD-пайплайн, які переходять у DR-режим. У конфігурації виставляється dr_mode = true.
   - Terraform аналізує поточний state та планує, які ресурси потрібно створити/оновити.
3. Перевірка та створення AWS Load Balancer
   - Першим кроком терраформ (або вже існуючі налаштування) готує Load Balancer у DR-регіоні чи резервній VPC.

   Якщо балансувальник уже існує, Terraform перевіряє його конфігурацію; в разі потреби – оновлює.
4. Розгортання (або активація) Auto Scaling Group
   - Auto Scaling Group (ASG) у DR-регіоні підключається до Load Balancer; лише після того, як LB є доступним, піднімаються EC2-інстанси.
   - Terraform збільшує desired_capacity з 0 до заданого значення (наприклад, 2), тим самим активуючи резервну групу серверів.

5. Створення та прив’язка WAF

- AWS WAF (Web ACL) створюється або оновлюється з аналогічними правилами (rate limiting, bot-фільтри) для DR.
- WAF асоціюється з Load Balancer, щоб почати фільтрувати підозрілий трафік (без діючого LB WAF не має сенсу).

6. Актуалізація баз даних і даних конфігурацій

- Якщо використовується AWS DMS або реплікація (RDS Read Replica), DR-середовище баз даних синхронізується з основною БД.
- У DR-сценарії Terraform (за потреби) змінює налаштування DB-ендпоінтів, аби застосунки в ASG зверталися до актуальної копії даних.

7. Оновлення Route 53

- Failover Policy або DNS-запис оновлюється, вказуючи, що тепер доступна DR-інфраструктура (Load Balancer + ASG + WAF).
- У разі класичного failover, Primary-запис стає “Failed”, і Route 53 починає віддавати Secondary (DR). Якщо використовується manual switch, Terraform змінює CNAME/A-запис.

8. Перенаправлення трафіку

- Клієнтські запити поступово починають надходити на WAF у DR-регіоні, потім – на Load Balancer, і врешті розподіляються поміж активованих інстансів ASG.
- Так система повноцінно функціонує в резервному середовищі, доки основна інфраструктура не відновиться.

9. Контроль та оптимізація

- Моніторинг (CloudWatch, CloudTrail) фіксує, що DR-ресурси вийшли на робочу потужність, WAF активно блокує шкідливий трафік.
- Після стабілізації можна зменшити DR-середовище або повернутися до primary, оновивши dr_mode = false і повернувши запис у Route 53.

10. Аналіз інциденту та закріплення знань

- Збираються логи WAF та ELB (CloudTrail, S3), аналізуються

метрики.

- На основі уроків, отриманих під час атаки чи збою, оновлюються правила WAF, настройки масштабування, політики безпеки.

У цьому візуальному алгоритмі зображено покрокову послідовність дій у разі збою Health Check або виявлення серйозної DDoS-атаки. На діаграмі відображено, як автоматично формується інфраструктура в DR-регіоні: спочатку Terraform перевіряє або створює Load Balancer, потім ініціює Auto Scaling Group, далі підключається WAF для фільтрації шкідливого трафіку. Після цього Route 53 змінює DNS-записи, спрямовуючи потік запитів на резервний Load Balancer, тоді як DMS реплікує дані в DR-базу. Моніторинг та аналіз навантаження дають змогу перевірити ефективність DR-інфраструктури й блокування DDoS. По завершенні інциденту рішення про повернення до основного регіону чи подальше використання DR ухвалюється згідно з внутрішньою політикою. Усе це проілюстровано на рисунку 3.3, щоб чітко продемонструвати ланцюг взаємодії та залежностей між компонентами.

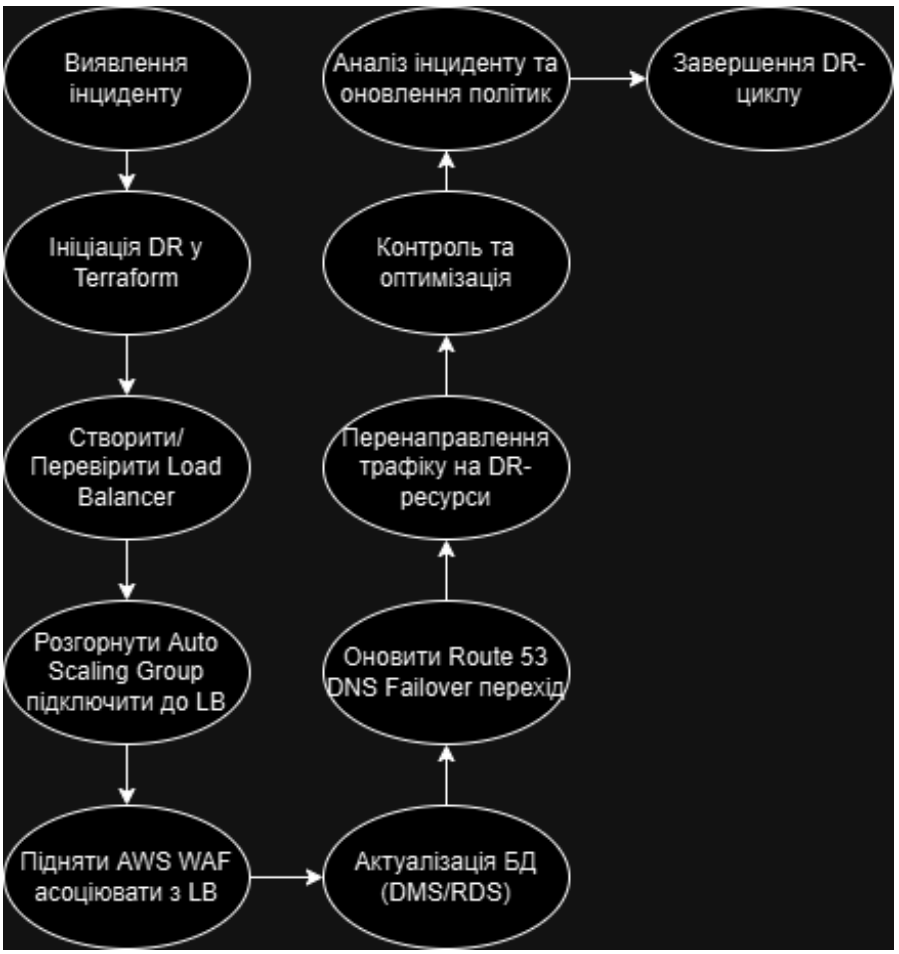

Рисунок 3.3 – Алгоритм відпрацювання системи

Описання кожного кроку:

1. Виявлення інциденту: Health Check або моніторинг визначає збій / DDoS.
2. Ініціація DR: Terraform playbook застосовує dr_mode = true.
3. Створення LB: Без LB система не може прикріпити ASG чи WAF. LB – перша сходинка.
4. Розгортання ASG: LB уже готовий, тому піднімається ASG і реєструє EC2-інстанси в LB.
5. Прив’язка WAF: Web ACL чіпляється до ARN Load Balancer. Запити від клієнтів починають фільтруватися на L7.
6. Актуалізація БД (бази даних): DR RDS або DMS синхронізують дані; застосунки можуть звертатися до нової БД.
7. Оновлення Route 53: DNS або Failover Routing починає перенаправляти

запити на DR LB.

8. Перенаправлення трафіку: Клієнти заходять через DR WAF, трафік баланситься на DR ASG.

9. Контроль: ASG масштабиться, WAF блокує підозрілий трафік, адміністратор стежить через логи і метрики.

10. Аналіз та вдосконалення: Після стабілізації збори логів, навчання, оновлення політик безпеки.

Таким чином, алгоритм дає змогу швидко перебудувати ланцюжок LB-ASG-WAF і переключити клієнтів на робоче DR-середовище. Ключова логіка в тому, що Load Balancer створюється першим, далі його використовує Auto Scaling Group, і лише тоді WAF прив’язується до LB. Terraform впорядковує ці залежності автоматично, коли ви правильно вкажете depends_on і посилання між ресурсами.

РОЗДІЛ 4 ОЦІНКА ЕФЕКТИВНОСТІ ТА ПОДАЛЬШИЙ РОЗВИТОК СИСТЕМ

4.1 Аналіз існуючих ринкових рішень

Сучасний ринок пропонує різноманітні рішення для забезпечення резервного копіювання та відновлення після інцидентів, які можуть ефективно працювати як у локальних середовищах, так і в хмарі. Огляд найбільш популярних платформ дає змогу зрозуміти їхні особливості, переваги та недоліки.

1. Microsoft Azure Site Recovery (ASR): Azure Site Recovery є повністю керованим сервісом для відновлення після інцидентів, що підтримує реплікацію віртуальних машин (VM), фізичних серверів і середовищ на основі Hyper-V і VMware. Цей сервіс спрощує управління, адже користувачам не потрібно займатися інфраструктурою на рівні платформи. ASR підтримує тестування сценаріїв відновлення без впливу на основну інфраструктуру, що особливо важливо для підприємств з обмеженим IT-персоналом. Одним із недоліків ASR є його ціноутворення, яке базується на кількості віртуальних машин, що може бути менш вигідним для великих середовищ з високою кількістю віртуальних машин, порівняно з рішеннями з фіксованою вартістю на сервер. [17]

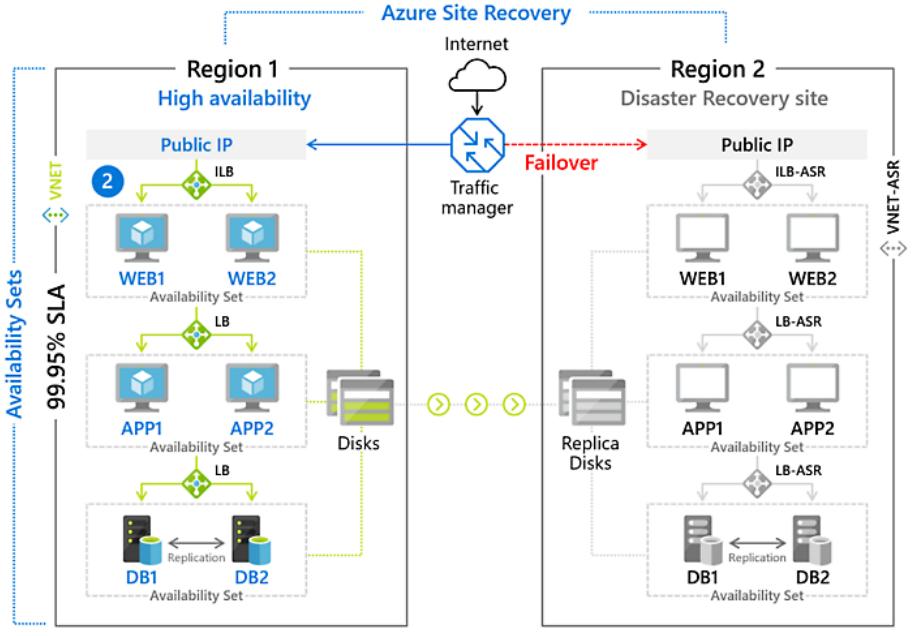

Рисунок 4.1 – Azure Site Recovery архітектура

Рисунок 4.1 показує архітектуру рішення Azure Site Recovery (ASR) для забезпечення високої доступності та аварійного відновлення. Ось ключові елементи на схемі:

Region 1: High Availability (Висока доступність):

- Віртуальна мережа (VNET) з інфраструктурою, що має 99,95% доступності (SLA).
- WEB1 і WEB2: Веб-сервери, розміщені в наборі доступності (Availability Set) і об'єднані з внутрішнім балансувальником навантаження (ILB) для розподілу трафіку.
- APP1 і APP2: Сервери додатків, також в наборі доступності, що розподіляють навантаження через балансувальник.
- DB1 і DB2: Сервери баз даних, налаштовані для реплікації та об'єднані в окремий набір доступності.

Region 2: Disaster Recovery Site (Сайт аварійного відновлення):

- Запасний регіон з репліками серверів, до яких можна переключитися у випадку збою основного сайту.
- WEB1, WEB2, APP1, APP2, DB1, DB2: Реплікація серверів з Region 1, налаштована через Azure Site Recovery для швидкого відновлення.

Traffic Manager:

- Використовується для управління інтернет-трафіком між двома регіонами. У випадку збою, він автоматично перенаправляє трафік на Region 2 для забезпечення безперервної роботи.

Failover:

У випадку збоїв у Region 1 трафік перенаправляється на Region 2 для підтримки роботи додатків та мінімізації простою.

Ця архітектура допомагає підтримувати безперервну роботу критично важливих систем навіть у випадку збоїв, забезпечуючи високу доступність у основному регіоні та готовність до аварійного відновлення в запасному регіоні.

2. Amazon Web Services (AWS) Disaster Recovery: AWS пропонує набір

інструментів для відновлення після інцидентів, включно з AWS Elastic Disaster Recovery та іншими сервісами для реплікації даних. На відміну від ASR, AWS Disaster Recovery вимагає більше налаштувань і управління інфраструктурою, надаючи користувачам більше контролю над підсистемами, такими як EC2 (Elastic Compute Cloud) для розгортання цільових екземплярів. Це рішення підходить компаніям, які потребують більше гнучкості та контролю над інфраструктурою, хоча це може ускладнювати управління, особливо для підприємств із невеликим IT-штатом. [18]

При виборі стратегії відновлення після аварій для робочого навантаження з високою доступністю на AWS, важливо враховувати рівень можливої аварії та вимоги до надійності. Для аварій, що обмежуються втратою одного фізичного дата-центру, може бути достатньо стратегії резервного копіювання та відновлення. Однак, якщо визначення аварії включає втрату цілого регіону або якщо регуляторні вимоги вимагають відновлення на рівні регіону, тоді варто розглянути більш складні стратегії, такі як Pilot Light, Warm Standby або Multi-Site Active/Active.

При виборі стратегії та ресурсів AWS для її реалізації слід враховувати поділ на площину даних (data plane) та контрольну площину (control plane). Площина даних відповідає за надання послуг у реальному часі, тоді як контрольна площина*використовується для конфігурації середовища. Для максимальної надійності варто використовувати лише операції площини даних у процесі перемикання на резервний сайт, оскільки площина даних зазвичай має вищі цілі доступності у порівнянні з контрольними площинами.

Такий підхід дозволяє забезпечити високу стійкість до відмов, а також відповідати вимогам щодо доступності та регуляторним стандартам у разі серйозних аварій.

У підході Pilot Light реплікуються дані з одного регіону в інший і розгортаєте копію основної інфраструктури робочого навантаження. Ресурси, потрібні для реплікації даних і резервного копіювання, такі як бази даних і об'єктне сховище, працюють постійно. Інші елементи, зокрема сервери застосунків із завантаженим кодом і конфігураціями, залишаються “вимкненими” і використовуються лише під

час тестування або при активації аварійного відновлення.

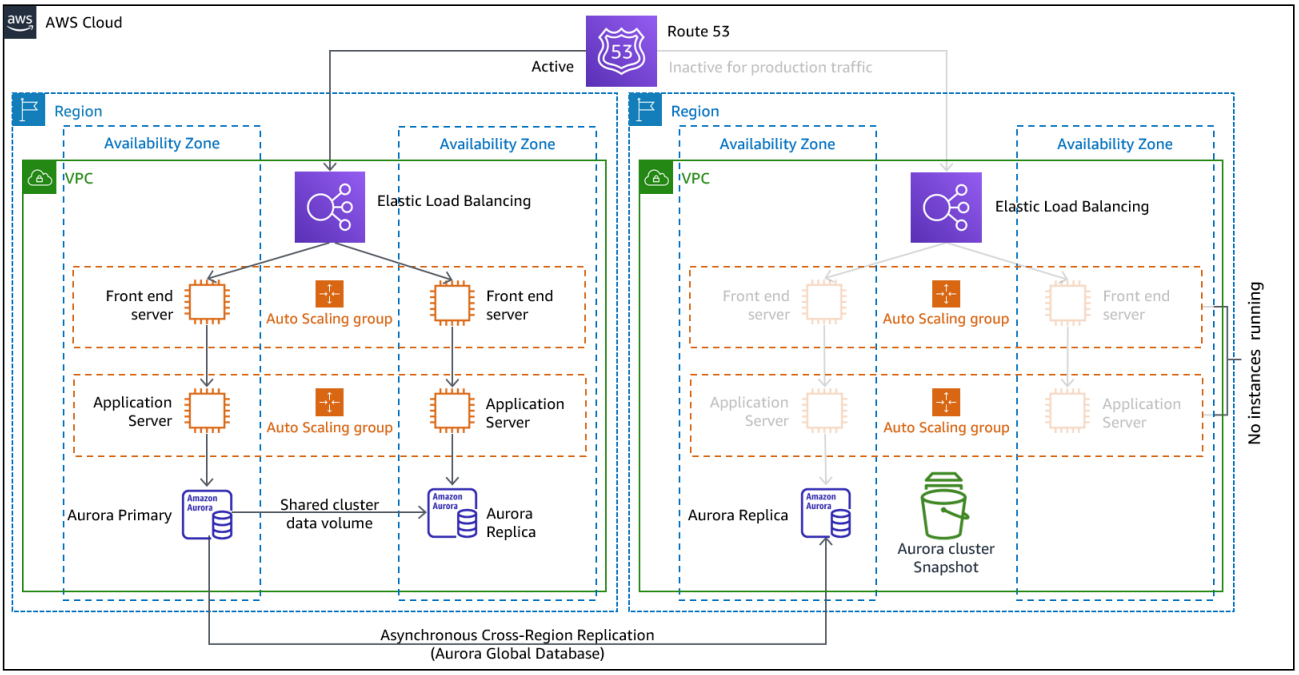

Рисунок 4.2 – Архітектура Pilot Light

Рисунок 4.2 ілюструє архітектуру Pilot Light для аварійного відновлення, що забезпечує готовність до відновлення критичних систем у разі катастрофи. У даній архітектурі основні ресурси, необхідні для підтримки резервної копії та синхронізації даних, постійно активні в головному регіоні (Region), де інфраструктура працює у звичайному режимі.

У головному регіоні розміщено сервери front-end і application, що об'єднані в Auto Scaling Group, що дозволяє автоматично змінювати кількість інстансів на основі навантаження. Також тут розташована база даних Amazon Aurora Primary та її репліка, які забезпечують високу доступність та синхронізацію даних через асинхронне крос-регіонне реплікування. Всі ці компоненти об'єднані через балансувальник навантаження (Elastic Load Balancing), що розподіляє трафік між інстансами.

У другому регіоні інфраструктура налаштована, але інстанси не активні. У разі відмови в головному регіоні, можна швидко активувати ці компоненти, що забезпечує готовність резервної інфраструктури для перемикання виробничого

трафіку.

Завдяки гнучкості хмарних технологій можна масштабувати ресурси за потреби: зберігати конфігурації серверів у вимкненому стані, а потім “увімкнути” їх, коли це необхідно. Рекомендованою практикою є не розгортати ресурс, коли він не потрібен, а лише мати готові налаштування для швидкого розгортання за запитом. На відміну від підходу резервного копіювання та відновлення, у моделі Pilot Light основна інфраструктура завжди доступна, що дозволяє за потреби оперативно масштабувати та запускати повноцінне виробниче середовище, активувавши й масштабуючи сервери застосунків.

3. Druva: Druva спеціалізується на відмовостійкості даних за моделлю “платформа як послуга” (PaaS). Ключовою бізнес-пропозицією компанії є надання рішень для захисту даних, які спрямовані на подолання поширених ризиків. Унікальний підхід Druva до резервного копіювання та відновлення суттєво змінив методи безпеки, захисту та використання даних. Завдяки хмарі відмовостійкості даних Druva відпадає потреба в дорогому обладнанні, програмному забезпеченні та послугах. Ця хмарна архітектура пропонує безпечні, постійно доступні та масштабовані рішення для роботи з даними. Druva пропонує швидке відновлення з мінімальними часами простою (RTO) та відновленням даних за годину (RPO). Така структура робить Druva хорошим вибором для компаній, які хочуть мінімізувати ризики віртуальних атак і втрат даних при низьких витратах на управління. Крім того, Druva забезпечує централізоване управління, автоматизоване резервне копіювання та відновлення, а також інтеграцію з популярними хмарними платформами, такими як AWS та Azure. [19] [20]

4. Carbonite Recover: Орієнтована на малий і середній бізнес, Carbonite Recover надає хмарні та гібридні рішення для резервного копіювання, підтримуючи безперервну реплікацію даних на рівні байтів та швидке переключення в разі відмови системи. Платформа підтримує відновлення для Windows, Hyper-V, Linux та VMware, забезпечуючи високу доступність і відновлення з мінімальними витратами часу та зусиль.

5. Appranix: Це сучасне рішення для відновлення після інцидентів на

основі технології Recovery-as-Code (RaC), яке автоматизує відновлення хмарних середовищ за допомогою моделі, орієнтованої на додатки. Appranix виходить за рамки резервного копіювання та відновлення простих обчислювальних екземплярів, файлових систем і баз даних, пропонуючи наскрізний захист складних хмарних додатків, що складаються. Воно забезпечує автоматизоване відновлення в режимі реального часу повних середовищ додатків, разом з конфігураційними даними та метаданими. Результатом є справжня хмарна кіберстійкість з можливостями швидкого відновлення, що мінімізує час простою, значно скорочує середній час усунення несправностей і дозволяє швидше відновити бізнес-операції. Appranix забезпечує швидке відновлення після кібератак або збоїв у хмарних регіонах завдяки відсутності потреби в ручному втручанні. Платформа також пропонує інтеграцію з основними хмарними сервісами та високий рівень автоматизації, що підходить для компаній, яким необхідне безпечне відновлення хмарних додатків. [21]

4.1.1 Порівняльний аналіз

Сучасний ринок резервного копіювання та відновлення після інцидентів пропонує безліч рішень, які можуть ефективно працювати як у локальних середовищах, так і в хмарі. Кожна платформа має свої особливості, переваги та недоліки, що робить їх більш придатними для різних типів підприємств та інфраструктур. Вибір залежить від багатьох факторів, таких як масштаб бізнесу, вимоги до доступності, бюджет, складність управління, а також потреба в автоматизації та гнучкості.

У додатку 3.1 наведено порівняльний аналіз п'яти популярних платформ для резервного копіювання та відновлення після інцидентів: Microsoft Azure Site Recovery (ASR), Amazon Web Services (AWS) Disaster Recovery, Druva, Carbonite Recover та Appranix. Ця таблиця допоможе отримати загальне уявлення про ключові можливості кожної платформи, а також їх переваги й недоліки.

Результати аналізу:

- Microsoft Azure Site Recovery (ASR) і AWS Disaster Recovery –

найбільш підходять для великих підприємств із високими вимогами до налаштування та контролю, але їхня вартість і складність можуть обмежувати використання для малого бізнесу.

- Druva і Carbonite Recover – це легкі у впровадженні рішення для малих та середніх бізнесів, які потребують мінімальних витрат на управління та хочуть забезпечити простий захист і відновлення.
- Appranix – сучасне автоматизоване рішення, яке підходить для підприємств, орієнтованих на хмарні середовища та Recovery-as-Code, особливо тих, хто прагне мінімізувати вплив кібератак.

Кожне рішення має свої сильні сторони і краще підходить для різних сценаріїв та потреб. Вибір платформи залежить від розміру підприємства, його IT-ресурсів та вимог до доступності й захисту даних.

4.1.2 Економічний аналіз

Для проведення економічного аналізу різних рішень резервного копіювання та відновлення після інцидентів необхідно врахувати ключові фактори, що впливають на витрати та економічну ефективність кожної платформи. Нижче наведені основні елементи такого аналізу для платформ Microsoft Azure Site Recovery (ASR), Amazon Web Services (AWS) Disaster Recovery, Druva, Carbonite Recover та Appranix.

1. Прямі витрати

Прямі витрати включають вартість підписки або ліцензування, витрати на інфраструктуру та зберігання, а також витрати на управління. Кожна платформа має свої особливості ціноутворення:

- Microsoft Azure Site Recovery (ASR): Вартість залежить від кількості віртуальних машин (VM) і фізичних серверів, які потребують резервного копіювання. Це може стати дорогим рішенням для великих середовищ, оскільки ціна зростає зі збільшенням кількості серверів та об’єктів для реплікації.
- AWS Disaster Recovery: AWS пропонує кілька стратегій резервного копіювання (Backup & Restore, Pilot Light, Warm Standby, Multi-Site Active/Active),

кожна з яких має різну вартість, що залежить від необхідного рівня доступності та відновлення. Витрати можуть значно варіюватися залежно від конфігурації інфраструктури та використання додаткових сервісів, таких як EC2, S3 та RDS.

- Druva: Витрати на Druva базуються на моделі "підписка як послуга" (SaaS), що включає всі витрати на резервне копіювання та зберігання даних у хмарі. Це робить Druva вигідним варіантом для організацій, які хочуть уникнути великих капіталовкладень, оскільки вартість включає зберігання даних у хмарі.

- Carbonite Recover: Carbonite використовує фіксовану модель ціноутворення для малого та середнього бізнесу з безперервною реплікацією. Це робить його привабливим варіантом для компаній із меншими масштабами, де складніші рішення, як ASR чи AWS, можуть бути надто дорогими.

- Appranix: Appranix пропонує відновлення як код (Recovery-as-Code), що дозволяє економити на управлінні та підтримці завдяки високій автоматизації. Вартість залежить від кількості додатків, які потребують відновлення, і може бути вигідною для компаній, що використовують лише хмарні середовища.

2. Непрямі витрати

Непрямі витрати можуть включати витрати на навчання персоналу, час, необхідний для налаштування та управління рішенням, а також можливі збої під час активації резервного копіювання.

- Навчання персоналу та управління: ASR і AWS Disaster Recovery можуть вимагати більше часу на навчання та управління через складніші налаштування, тоді як рішення, такі як Druva та Carbonite, є простішими у використанні і, відповідно, менш затратними з точки зору навчання персоналу.

- Можливий час простою: Час простою є критичним фактором при оцінці економічної ефективності. Високий час простою може призвести до фінансових втрат, особливо для підприємств, що залежать від безперервної роботи. ASR та Appranix, завдяки високій автоматизації, можуть забезпечити швидке відновлення, що допомагає мінімізувати час простою та відповідні фінансові втрати.

3. Потенційна економія витрат

Резервне копіювання та відновлення після інцидентів можуть забезпечити значну економію витрат у випадках, коли збої трапляються рідко, але потребують швидкого відновлення. Вибір підходу до резервного копіювання може вплинути на загальну економічну ефективність:

- Microsoft Azure Site Recovery (ASR): Завдяки повному керуванню інфраструктурою на рівні платформи, ASR дозволяє скоротити витрати на підтримку та управління, що є вигідним для компаній з обмеженими ресурсами для управління IT-інфраструктурою.
- AWS Disaster Recovery: AWS пропонує масштабованість та адаптивність, що дозволяє компаніям налаштовувати систему під свої потреби. Це може знизити витрати на підтримку та підвищити економічну ефективність за умови правильного вибору стратегії (наприклад, використання Pilot Light для економії витрат у періоди простою).
- Druva: Платформа, побудована на хмарі, дозволяє значно зменшити витрати на інфраструктуру, оскільки користувачі платять лише за підписку, що робить її економічно вигідною для компаній, які прагнуть знизити витрати на підтримку фізичної інфраструктури.
- Carbonite Recover та Appranix: Ці рішення особливо підходять для малих і середніх компаній, оскільки дозволяють уникнути великих капіталовкладень завдяки моделям "як послуга" та високому рівню автоматизації, що скорочує витрати на управління.

4. Аналіз окупності інвестицій (ROI)

Для оцінки ефективності кожного рішення можна також розглянути показник окупності інвестицій (ROI), який враховує потенційні витрати на простої, капіталовкладення та експлуатаційні витрати.

- ASR та AWS: Обидва рішення забезпечують високий рівень доступності та готовності до відновлення, що є важливим для організацій, які не можуть дозволити довготривалий простій. Їх висока початкова вартість може окупитися завдяки зниженню витрат на простої та підтримку, особливо у великих компаніях із складною інфраструктурою.

- Druva: За рахунок моделі SaaS та відсутності капітальних витрат, Druva має швидку окупність для компаній, які потребують простого, але надійного рішення для резервного копіювання в хмару.
- Carbonite та Appranix: Завдяки нижчим початковим витратам, ці рішення можуть забезпечити швидшу окупність для невеликих компаній, особливо тих, які не мають складної інфраструктури та високих вимог до доступності.

Вибір оптимального рішення для резервного копіювання та відновлення після інцидентів залежить від розміру компанії, складності інфраструктури та вимог до доступності. Для великих компаній із складною IT-інфраструктурою оптимальними варіантами є Microsoft Azure Site Recovery (ASR) та AWS Disaster Recovery, незважаючи на їхню відносно високу вартість. Вони пропонують широкий функціонал та адаптивність, які забезпечують окупність інвестицій у довгостроковій перспективі.

Для середніх і малих підприємств рішення Druva, Carbonite Recover та Appranix є економічно вигідними завдяки меншому обсягу капіталовкладень і можливості швидкого налаштування та управління, що забезпечує швидку окупність вкладень.

Таблиці 3.1 – Резюме економічного аналізу

Платформа	**Прямі витрати**	**Непрямі витрати**	**Потенційна економія витрат**	**ROI**
Microsoft Azure Site Recovery (ASR)	Високі витрати для великих середовищ	Висока вартість навчання та управління	Скорочення витрат на підтримку та простої	Високий для великих компаній
AWS Disaster Recovery	Залежить від обраної стратегії	Висока вартість налаштування	Економія за рахунок вибору оптимальної стратегії	Високий за умов правильної конфігураці

Продовження таблиці 3.1 – Резюме економічного аналізу

Платформа	**Прямі витрати**	**Непрямі витрати**	**Потенційна економія витрат**	**ROI**
Druva	Низькі капіталовкладення, SaaS-модель	Мінімальні витрати на управління	Економія завдяки відсутності фізичної інфраструктури	Швидкий для мал

Таблиця 3.1 резюме економічного аналізу надає порівняльну характеристику найбільш популярних рішень для резервного копіювання та відновлення після інцидентів. У таблиці наведено ключові аспекти кожного рішення, включно з вартістю впровадження, загальними експлуатаційними витратами, рівнем автоматизації, вимогами до управління та рівнем адаптивності до різних типів інфраструктур.

Цей аналіз допомагає зрозуміти, яке з рішень є найбільш економічно вигідним для різних типів бізнесів, зважаючи на їхні потреби в надійності, бюджеті та доступності технічного персоналу.

4.2 Можливості подальшого вдосконалення системи

4.2.1 Інтеграція зі штучним інтелектом

Інтеграція зі штучним інтелектом (ШІ) може значно підвищити ефективність автоматизованих систем відновлення після кіберінцидентів. Використання ШІ дозволяє:

- Автоматичне виявлення аномалій: ШІ може аналізувати великі обсяги даних у реальному часі, виявляючи аномалії та потенційні загрози, які можуть бути пропущені традиційними методами.
- Прогнозування інцидентів: На основі історичних даних ШІ може прогнозувати можливі кіберінциденти та пропонувати превентивні заходи для їх

запобігання.

- Оптимізація процесів відновлення: ШІ може автоматично вибирати найефективніші стратегії відновлення, зменшуючи час простою та втрати даних.
- Покращення безпеки: Використання ШІ для аналізу поведінки користувачів та систем дозволяє виявляти підозрілу активність та запобігати внутрішнім загрозам.

Рисунок 4.3 ілюструє інтеграцію штучного інтелекту (ШІ) у процеси кібербезпеки. На схемі показано, як ШІ може бути використаний на різних етапах відновлення після кіберінцидентів: виявлення загроз, прогнозування інцидентів, оптимізація процесів відновлення, покращення безпеки, автономні системи відновлення. Ця схема демонструє, як ШІ може бути інтегрований у різні аспекти кібербезпеки, забезпечуючи більш ефективне та швидке реагування на загрози та інциденти.

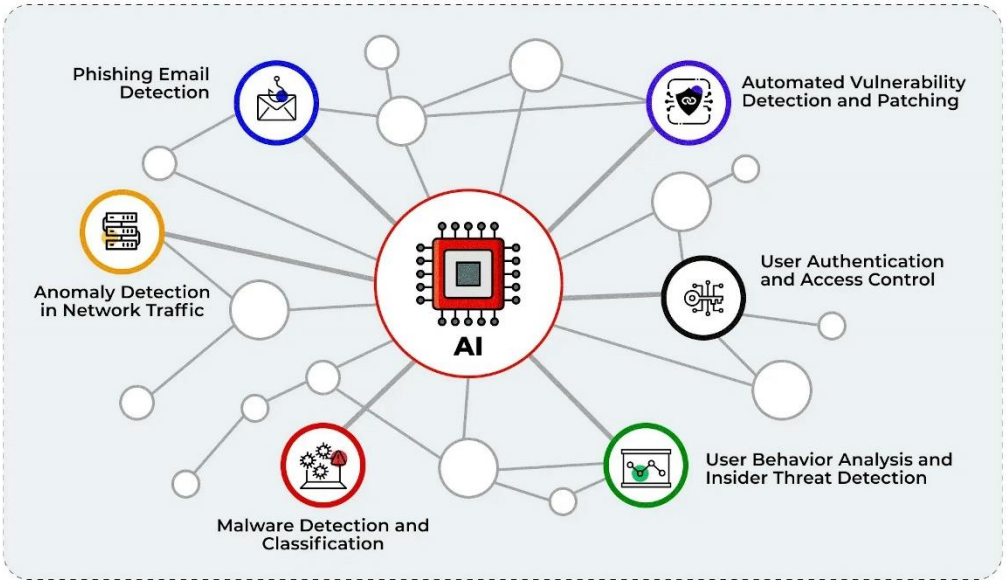

Рисунок 4.3 – Приклад використання штучного інтелекту у кібербезпеці

4.2.2 Розширення функціональності

Розширення функціональності автоматизованих систем відновлення може включати:

- Підтримка нових хмарних платформ: Додавання підтримки для нових хмарних провайдерів, таких як IBM Cloud або Oracle Cloud, дозволить розширити

можливості системи та забезпечити більшу гнучкість для користувачів.

- Інтеграція з DevOps інструментами: Інтеграція з популярними DevOps інструментами, такими як Jenkins, GitLab CI/CD або Ansible, дозволить автоматизувати процеси відновлення в рамках CI/CD пайплайнів.
- Підтримка контейнерних оркестраторів: Додавання підтримки для інших контейнерних оркестраторів, таких як Docker Swarm або Apache Mesos, дозволить забезпечити більшу гнучкість та масштабованість для користувачів, які використовують різні технології контейнеризації.
- Розширені можливості моніторингу: Додавання нових функцій для моніторингу та аналізу стану системи, таких як детальний аналіз логів, візуалізація метрик та інтеграція з системами моніторингу, такими як Prometheus, Grafana або Enterprise-рішеннями, такими як Dynatrace.

4.3 Майбутні тренди в автоматизації відновлення після кіберінцидентів

Автоматизація відновлення після кіберінцидентів є динамічною сферою, яка постійно розвивається. Нижче наведено кілька ключових трендів, які, ймовірно, визначатимуть майбутнє цієї галузі:

4.3.1 Використання штучного інтелекту та машинного навчання

Штучний інтелект (ШІ) та машинне навчання (МН) стають невід'ємною частиною сучасних систем кібербезпеки. Вони дозволяють:

- Автоматичне виявлення загроз: ШІ може аналізувати великі обсяги даних у реальному часі, виявляючи аномалії та потенційні загрози.
- Прогнозування інцидентів: На основі історичних даних ШІ може прогнозувати можливі кіберінциденти та пропонувати превентивні заходи.
- Оптимізація процесів відновлення: ШІ може автоматично вибирати найефективніші стратегії відновлення, зменшуючи час простою та втрати даних. [22]

4.3.2 Інтеграція з хмарними сервісами

Хмарні технології продовжують розвиватися, і їх інтеграція з системами відновлення після інцидентів стає все більш важливою. Основні напрямки включають:

- Гібридні та мультихмарні середовища: Підтримка гібридних та мультихмарних середовищ дозволяє організаціям використовувати переваги різних хмарних провайдерів.
- Автоматизація резервного копіювання та відновлення: Використання хмарних сервісів для автоматизації процесів резервного копіювання та відновлення даних.
- Підвищення безпеки: Хмарні платформи надають інструменти для шифрування даних, управління доступом та моніторингу, що підвищує загальний рівень безпеки.

4.3.3 Розвиток контейнеризації та оркестрації

Контейнеризація та оркестрація додатків стають стандартом для сучасних IT-інфраструктур. Основні тренди включають:

- Масштабованість та гнучкість: Контейнери дозволяють швидко масштабувати додатки та адаптуватися до змін у навантаженні.
- Автоматизація відновлення: Інструменти оркестрації, такі як Kubernetes, дозволяють автоматизувати процеси відновлення та забезпечити безперервність роботи додатків.
- Інтеграція з DevOps: Контейнеризація сприяє інтеграції з DevOps-практиками, що дозволяє швидше впроваджувати зміни та покращувати процеси відновлення.

4.3.4 Підвищення рівня автоматизації та автономності

Системи відновлення після інцидентів стають все більш автономними, що дозволяє зменшити втручання людини та підвищити ефективність:

- Автономні системи відновлення: Використання автономних систем, які можуть самостійно виявляти та усувати проблеми без втручання людини.
- Інтелектуальні сценарії відновлення: Розробка сценаріїв відновлення, які автоматично адаптуються до конкретних умов та типів інцидентів.
- Зменшення людського фактору: Автоматизація процесів відновлення знижує ризик помилок, пов'язаних з людським фактором, та підвищує надійність систем.

4.3.5 Підвищення уваги до кіберстійкості

Кіберстійкість стає ключовим аспектом для організацій, які прагнуть забезпечити безперервність бізнесу:

- Інтеграція з бізнес-процесами: Відновлення після інцидентів стає невід'ємною частиною загальної стратегії кіберстійкості організації.
- Підвищення обізнаності та навчання: Організації інвестують у навчання персоналу та підвищення обізнаності щодо кіберзагроз та методів відновлення.
- Регулярне тестування та вдосконалення: Постійне тестування та вдосконалення планів відновлення дозволяє забезпечити їхню ефективність у разі реальних інцидентів.

У третьому розділі проведено порівняння існуючих ринкових рішень для резервного копіювання та відновлення після інцидентів. Виявлено, що рішення, такі як Microsoft Azure Site Recovery та AWS Disaster Recovery, забезпечують високий рівень функціональності та автоматизації. Інтеграція зі штучним інтелектом та машинним навчанням може значно підвищити ефективність автоматизованих систем відновлення, дозволяючи автоматично виявляти загрози, прогнозувати інциденти та оптимізувати процеси відновлення. Майбутні тренди включають подальший розвиток хмарних технологій, контейнеризації та оркестрації, а також підвищення рівня автоматизації та автономності систем відновлення.

ВИСНОВКИ

У цій роботі було розглянуто важливість побудови надійної стратегії резервного копіювання та аварійного відновлення для захисту даних і забезпечення безперервності бізнес-процесів в умовах сучасних хмарних обчислень.

Мета роботи — розробити метод побудови автоматизованої системи відновлення після кіберінцидентів з використанням хмарних технологій для забезпечення відмовостійкості та мінімізації втрат даних — була досягнута. У межах дослідження було створено структурну схему рішення, яка дозволяє оптимізувати процес відновлення, а також впроваджено підхід автоматизованого керування інфраструктурою з використанням сучасних інструментів.

У першому розділі проведено аналіз сучасного стану кібербезпеки та кібератак, зокрема розглянуто найбільш поширені загрози та їхній вплив на критичну інфраструктуру. Було визначено, що зростання кількості кібератак вимагає впровадження нових підходів до резервного копіювання та відновлення даних. Особлива увага приділена аналізу інцидентів у хмарних середовищах та розгляду інструментів, які забезпечують підвищення рівня кіберстійкості.

Другий розділ присвячений аналізу існуючих технологій автоматизації процесів відновлення. Досліджено можливості використання хмарних платформ, таких як AWS та Microsoft Azure, для забезпечення високого рівня готовності до аварійного відновлення. Було проведено порівняльний аналіз п'яти популярних платформ, що дозволило визначити їхні сильні сторони та обмеження залежно від бізнес-потреб організацій різного масштабу. Особливу увагу приділено функціоналу резервного копіювання, реплікації даних та автоматизованого масштабування.

У третьому розділі розроблено структурну схему автоматизованої системи відновлення. Запропонована архітектура включає ключові компоненти, такі як AWS Route 53, Auto Scaling Group, AWS DMS, Terraform та AWS WAF, що забезпечують автоматизацію процесів, їхню гнучкість та масштабованість. Було створено алгоритм відпрацювання відновлення, що враховує сценарії різних типів

інцидентів, включно з DDoS-атаками. Проведено теоретичний аналіз ефективності моделі, що дозволяє швидко переключатися на резервну інфраструктуру з мінімальними втратами даних і часу.

У четвертому розділі проведено аналіз результатів існуючих автоматизованої системи відновлення на основі хмарних технологій. Було виконано теоретичний аналіз результативності захисту запропонованих рішень для забезпечення захисту від DDoS атак. Результати показали, що запропонована система забезпечує високий рівень готовності до аварійного відновлення та має високий рівень захисту завдяки Web application firewall.

На основі проведених досліджень було зроблено висновок, що використання хмарних сервісів для резервного копіювання та аварійного відновлення має значні переваги, зокрема гнучкість, надійність, безпеку та ефективне використання ресурсів. Впровадження таких рішень є критично важливим для сучасних організацій, які прагнуть захистити свої ресурси та забезпечити стабільну роботу в умовах постійно зростаючих кіберзагроз.

ПЕРЕЛІК ПОСИЛАНЬ

1 Кількість кібератак на Україну зросла майже втричі. [Електронний ресурс]. – Електронні дані. – Економічна правда. – Режим доступу: https://epravda.com.ua/news/2024/01/31/709355/

2 В Україні з початку року сталося 14 млн кіберінцидентів. [Електронний ресурс]. – Електронні дані. – РБК-Україна. – Режим доступу: https://www.rbc.ua/ukr/news/ukraine-nachala-goda-proizoshlo-14-mln-kiberintsidentov-1611938414.html

3 Кількість зареєстрованих кіберінцидентів у 2022 році зросла майже втричі. [Електронний ресурс]. – Електронні дані. – CIP.gov.ua. – Режим доступу: https://cip.gov.ua/ua/news/u-2022-roci-kilkist-zareyestrovanikh-kiberincidentiv-virosla-maizhe-vtrichi-zvit

4 IT Security Economics 2020 [Електронний ресурс]. – Електронні дані. – Kaspersky Blog. – Режим доступу: https://www.kaspersky.com/blog/it-security-economics-2020-main/37205/

5 Computer & Security (стаття в ScienceDirect, фрагмент #br0250) [Електронний ресурс]. – Електронні дані. – ScienceDirect. – Режим доступу: https://www.sciencedirect.com/science/article/pii/S0167404824000385#br0250

6 BBC News: Technology (новина 2021 року) [Електронний ресурс]. – Електронні дані. – BBC. – Режим доступу: https://www.bbc.com/news/technology-59176826

7 Computer & Security (стаття в ScienceDirect, фрагмент #br0280) [Електронний ресурс]. – Електронні дані. – ScienceDirect. – Режим доступу: https://www.sciencedirect.com/science/article/pii/S0167404824000385#br0280

8 GDPR Enforcement Tracker [Електронний ресурс]. – Електронні дані. – Enforcement Tracker. – Режим доступу:https://www.enforcementtracker.com/

9 Cloud Security: Google Cloud Security Best Practices [Електронний ресурс]. – Електронні дані. – SentinelOne. – Режим доступу: https://www.sentinelone.com/cybersecurity-101/cloud-security/google-cloud-security-best-practices/

10 Top Security Best Practices for Google Cloud Platform (GCP) [Електронний ресурс]. – Електронні дані. – Cado Security Blog. – Режим доступу: https://www.cadosecurity.com/blog/top-security-best-practices-for-google-cloud-platform-gcp

11 Кіберполіція виявила понад 3600 кіберзлочинів у 2023 році [Електронний ресурс]. – Електронні дані. – Суспільне. – Режим доступу: https://suspilne.media/673484-za-2023-rik-kiberpolicia-viavila-ponad-3600-kiberzlociniv/

12 Як працює Ansible. [Електронний ресурс]. – Електронні дані. – Ansible.com. – Режим доступу: https://www.ansible.com/how-ansible-works

13 Вступ до Terraform. [Електронний ресурс]. – Електронні дані. – HashiCorp. – Режим доступу: https://developer.hashicorp.com/terraform/intro

14 Чому контейнери - це майбутнє віртуалізації. [Електронний ресурс]. – Електронні дані. – GigaCloud. – Режим доступу: https://gigacloud.ua/blog/navchannja/chomu-kontejneri-ce-majbutne-virtualizacii
15 Що таке Kubernetes. [Електронний ресурс]. – Електронні дані. – Colobridge. – Режим доступу: https://blog.colobridge.net/uk/2023/12/kubernetes-what-is-it-ua
16 Identity and Access Management (IAM). [Електронний ресурс]. – Електронні дані. – IBM. – Режим доступу: https://www.ibm.com/topics/identity-access-management
17 Про Azure Site Recovery. [Електронний ресурс]. – Електронні дані. – Microsoft Learn. – Режим доступу: https://learn.microsoft.com/en-us/azure/site-recovery/site-recovery-overview
18 AWS Disaster Recovery. [Електронний ресурс]. – Електронні дані. – AWS. – Режим доступу: https://aws.amazon.com/disaster-recovery/
19 Druva Data Security Cloud. [Електронний ресурс]. – Електронні дані. – Gartner. – Режим доступу: https://www.gartner.com/reviews/market/it-resilience-orchestration/vendor/druva/product/druva-data-security-cloud
20 Druva Azure VM Backup. [Електронний ресурс]. – Електронні дані. – Help Net Security. – Режим доступу: https://www.helpnetsecurity.com/2023/11/08/druva-azure-vm-backup/
21 Druva на AWS Marketplace. [Електронний ресурс]. – Електронні дані. – AWS Marketplace. – Режим доступу: https://aws.amazon.com/marketplace/pp/prodview-f3h4sc6ehdvhu
22 AI in Incident Response. [Електронний ресурс]. – Електронні дані. – LeewayHertz. – Режим доступу: https://www.leewayhertz.com/ai-in-incident-response/#What-is-AI-powered-incident-response

Додаток А

Таблиця А.1 – Види витрат, пов’язаних з кіберінцидентами

Платформа	Ключові можливості	Відсоток відбиття	Переваги	Недоліки
Microsoft Azure Site Recovery (ASR)	- Реплікація віртуальних машин, фізичних серверів та середовищ на основі Hyper-V та VMware. - Підтримка тестування сценаріїв відновлення без впливу на основну інфраструктуру. - Повністю керований сервіс.	~70–75%	- Спрощене управління, не потрібно займатися інфраструктурою на рівні платформи. - Підтримка тестування сценаріїв для перевірки готовності до відновлення.	- Висока вартість для великих середовищ через цінову політику на основі кількості VM. - Обмежені можливості для налаштування порівняно з AWS.
Amazon Web Services (AWS) Disaster Recovery	- Підтримка реплікації даних та відновлення на рівні інфраструктури з використанням EC2, S3, RDS тощо. - Гнучкі стратегії відновлення: Backup & Restore, Pilot Light, Warm Standby, Multi-Site Active/Active.	~85–90%	- Більше контролю над підсистемами та налаштуваннями. - Підтримка різних стратегій, що дозволяє адаптувати DR до різних рівнів вимог.	- Вимагає більше налаштувань та управління, що може бути складно для компаній із малим IT-персоналом. - Складність у реалізації для невеликих команд.
Druva	- SaaS-рішення для резервного копіювання та відновлення. Підтримка RTO та RPO на рівні години.	~70–80%	- Повністю хмарне рішення, не потребує додаткового ПЗ чи апаратного забезпечення. - Безпечне збереження даних у різних хмарних регіонах. Мінімальний час відновлення.	- Менші можливості для налаштування та контролю, ніж у рішень, які потребують більше інфраструктурних ресурсів.
Carbonite Recover	- Гібридне рішення з безперервною реплікацією даних на рівні байтів. - Підтримка відновлення для Windows, Hyper-V, Linux та VMware.	~60–70%	- Орієнтоване на малий та середній бізнес, просте в налаштуванні та управлінні. Швидке переключення	- Менш придатне для великих підприємств або компаній з високими вимогами до доступності та

			у разі відмови з мінімальними витратами часу.	масштабу.
Appranix	- Автоматизація відновлення хмарних середовищ за допомогою моделі Recovery-as-Code (RaC). - Інтеграція з основними хмарними сервісами. Відновлення після кібератак та регіональних збоїв.	~80–85%	- Високий рівень автоматизації, відсутність потреби в ручному втручанні. Підходить для підприємств з високими вимогами до безпеки та швидкого відновлення.	- Може не підходити для традиційних середовищ, оскільки орієнтоване на додатки у хмарі. -Обмежена адаптивність для гібридних середовищ.
Запропонована система	- Гібридна архітектура з Terraform, Auto Scaling, WAF, Route 53 та RDS (реплікація через DMS). - Автоматизоване відновлення після DDoS-інцидентів за сценарієм DR. - Динамічна реакція на навантаження (ASG + ELB), фільтрація Layer 7 у WAF.	~90–95%	- Гнучка інтеграція з локальним дата-центром (AWS Direct Connect), низька затримка. - Повне керування інфраструктурою як кодом (IaC), простий DR-перехід (Terraform). - Масштабованість і контроль налаштувань (WAF-правила, Auto Scaling політики).	- Вимагає значної початкової конфігурації та компетенції в Terraform / AWS. - Більш трудомістке налаштування для швидкого старту, ніж у «керованих» рішень.

Додаток Б

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

Магістерська робота

«Автоматизована система відновлення після кіберінцидентів з використанням хмарних технологій»

Виконав: студент групи КСДМ-61 Сарафанюк Руслан Олександрович

Керівник: к.т.н., доц. Антоненко Артем Васильович

Київ - 2025

Актуальність

З розвитком кіберзлочинності та збільшенням кількості кіберінцидентів, автоматизовані системи відновлення стають критично важливими для сучасних організацій. Оскільки значно посилюється з урахуванням стрімкого зростання кількості кіберінцидентів в Україні, особливо у 2022 році. За даними, кількість кібератак на Україну зросла майже втричі порівняно з попередніми роками, досягнувши понад 4500 інцидентів. Більшість атак організовані хакерськими групами, що мають прямі зв’язки з силовими структурами Росії. Ці угруповання націлюються на критичну інфраструктуру, урядові та комерційні установи, а також енергетичний сектор.

Мета, Об’єкт та Предмет

Мета роботи: оптимізувати процеси відновлення після кіберінцидентів з використанням хмарних технологій для забезпечення мінімального часу простою та втрат даних.

Об’єкт дослідження: процес автоматизованого відновлення систем.

Предмет використання: автоматизація процесів відновлення за допомогою хмарних технологій.

Статистика кіберінцидентів

- Цільові сектори: Державний сектор (75% атак), фінансовий сектор (5%), комерційні організації.
- Типи атак: Несанкціонований доступ (70%), сканування ресурсів (10%), атаки типу brute-force (7%).
- Збільшення кількості DDoS атак: Кількість кібератак у 2021 році становила приблизно 1400, а в 2022 та 2023 роках ця цифра перевищила 4500 інцидентів.

План відновлення після аварій у хмарі

1. Розуміння інфраструктури та визначення ризиків
2. Проведення аналізу впливу на бізнес (Business Impact Analysis)
3. Створення плану відновлення на основі RPO та RTO
4. Документування плану відновлення
5. Побудова хмарної інфраструктури для відновлення
6. Пошук надійного партнера для хмарного DR
7. Регулярне тестування DR-плану

Рисунок 2.1 – Етапи створення Disaster Recovery Plan

Архітектура хмарної інфраструктури AWS для відновлення

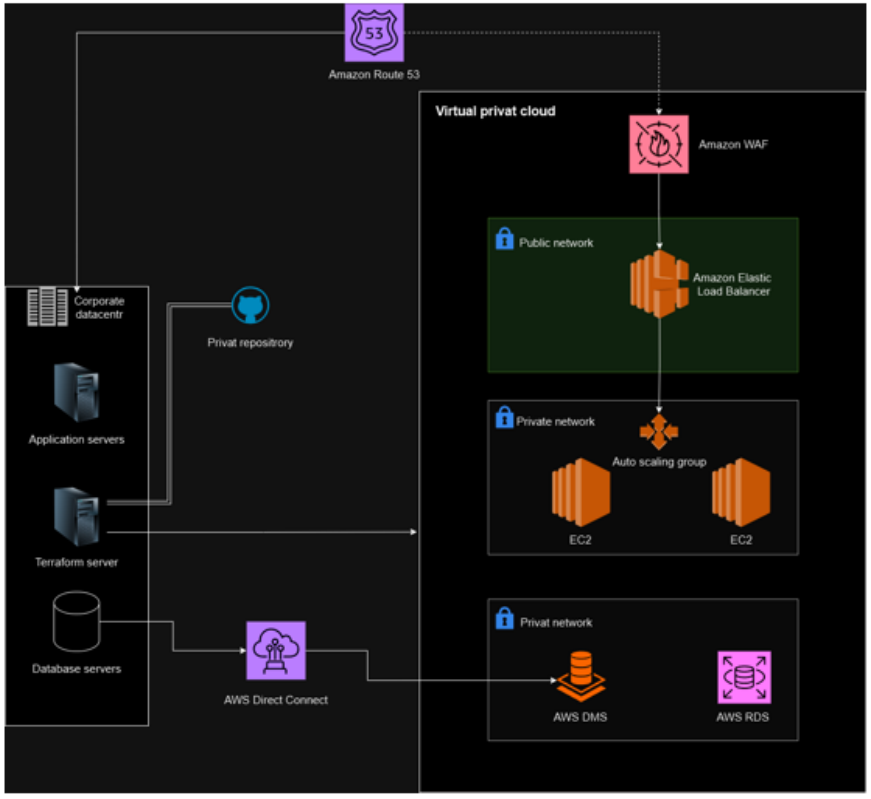

Рисунок 3.1 – Архітектура автоматизованої системи відновлення після кіберінциденту

1. Route 53
2. AWS Direct Connect
3. AWS Web Application Firewall (WAF)
4. AWS DMS
5. VPC з Elastic Load Balancer
6. Auto scaling group
7. Git Repository та Terraform

Алгоритм відновлення

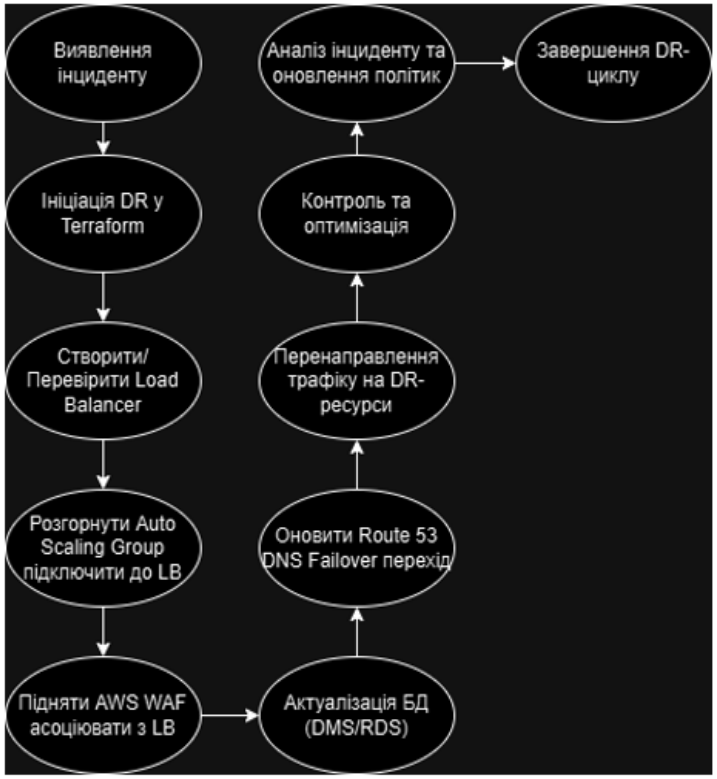

Рисунок 3.3 – Алгоритм відпрацювання системи

1. Виявлення інциденту: Health Check або моніторинг визначає збій / DDoS.
2. Ініціація Disaster Recovery (DR): Terraform playbook застосовує dr_mode = true.
3. Створення Load Balancer (LB): Без LB система не може прикріпити ASG чи WAF. LB – перша сходинка.
4. Розгортання Auto Scaling Group (ASG): LB уже готовий, тому піднімається ASG і реєструє EC2-інстанси в LB.
5. Прив'язка Web Application Firewall (WAF): WAF чіпляється до ARN Load Balancer. Запити від клієнтів починають фільтруватися на L7.
6. Актуалізація БД (бази даних): DR RDS або DMS синхронізують дані; застосунки можуть звертатися до нової БД.
7. Оновлення Route 53: DNS або Failover Routing починає перенаправляти запити на DR LB.
8. Перенаправлення трафіку: Клієнти заходять через DR WAF, трафік баланситься на DR ASG.
9. Контроль: ASG масштабиться, WAF блокує підозрілий трафік, адміністратор стежить через логи і метрики.
10. Аналіз інциденту та оновлення політик: Після стабілізації збори логів, навчання, оновлення політик безпеки.

Порівняльний аналіз п'яти популярних платформ

Платформа	Ключові можливості	Відсоток відбиття	Переваги	Недоліки
Microsoft Azure Site Recovery (ASR)	Реплікація віртуальних машин, фізичних серверів та середовищ на основі Hyper-V та VMware. Підтримка тестування сценаріїв відновлення. Повністю керований сервіс.	~70–75%	Спрощене управління, не потрібно займатися інфраструктурою на рівні платформи. Підтримка тестування сценаріїв для перевірки готовності до відновлення.	Висока вартість для великих середовищ через цінову політику на основі кількості VM. Відсутність автоматичного підняття WAF компоненту.
Amazon Web Services (AWS) Disaster Recovery	Підтримка реплікації даних та відновлення на рівні інфраструктури з використанням EC2, S3, RDS тощо. Гнучкі стратегії відновлення: Backup & Restore, Pilot Light, Warm Standby, Multi-Site Active/Active.	~85–90%	Більше контролю над підсистемами та налаштуваннями. Підтримка різних стратегій, що дозволяє адаптувати DR до різних рівнів вимог.	Вимагає більше налаштувань та управління. Відсутність автоматичного підняття WAF компоненту.
Druva	SaaS-рішення для резервного копіювання та відновлення. Підтримка RTO та RPO на рівні години.	~70–80%	Повністю хмарне рішення, не потребує додаткового ПЗ чи апаратного забезпечення. Безпечне збереження даних у різних хмарних регіонах. Мінімальний час відновлення.	Менші можливості для налаштування та контролю. Відсутність автоматичного підняття WAF компоненту.
Carbonite Recover	Гібридне рішення з безперервною реплікацією даних на рівні байтів. Підтримка відновлення для Windows, Hyper-V, Linux та VMware.	~60–70%	Орієнтоване на малий та середній бізнес, просте в налаштуванні та управлінні. Швидке переключення у разі відмови з мінімальними витратами часу.	Менш придатне для великих підприємств або компаній з високими вимогами до доступності та масштабу. Відсутність автоматичного підняття WAF компоненту.
Запропонована система	Гібридна архітектура з Terraform, Auto Scaling, WAF, Route 53 та RDS (реплікація через DMS). Автоматизоване відновлення після DDoS-інцидентів за сценарієм DR. Динамічна реакція на навантаження (ASG + ELB), фільтрація Layer 7 у WAF.	~90–95%	Гнучка інтеграція з локальним дата-центром (AWS Direct Connect), низька затримка. Повне керування інфраструктурою як кодом (IaC), простий DR-перехід (Terraform). Масштабованість і контроль налаштувань (WAF-правила, Auto Scaling політики).	Вимагає значної початкової конфігурації та компетенції в Terraform / AWS. Більш трудомістке налаштування для швидкого старту, ніж у «керованих» рішень.

Висновки

У першому розділі проаналізовано сучасний стан кібербезпеки та кібератак, визначено необхідність впровадження нових підходів до резервного копіювання та відновлення даних. Розглянуто кіберзагрози у хмарних середовищах та інструменти для підвищення кіберстійкості.

Другий розділ присвячений аналізу існуючих технологій автоматизації відновлення, порівняно можливості п'яти популярних хмарних платформ для забезпечення готовності до аварійного відновлення.

У третьому розділі розроблено структурну схему автоматизованої системи відновлення з використанням таких компонентів, як AWS Route 53, Auto Scaling Group та Terraform. Запропоновано алгоритм, що враховує різні типи інцидентів, включно з DDoS-атаками.

У четвертому розділі проведено теоретичний аналіз запропонованих рішень. Показано, що автоматизована система забезпечує високий рівень готовності до відновлення та ефективний захист завдяки використанню Web Application Firewall.

На основі проведених досліджень було зроблено висновок, що використання хмарних сервісів для резервного копіювання та аварійного відновлення має значні переваги, зокрема гнучкість, надійність, безпеку та ефективне використання ресурсів. Впровадження таких рішень є критично важливим для сучасних організацій, які прагнуть захистити свої ресурси та забезпечити стабільну роботу в умовах постійно зростаючих кіберзагроз.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

Магістерська робота

«Дослідження автоматизованої системи відновлення після кіберінцидентів з використанням хмарних технологій»

Виконав: студент групи КСДМ-61 Сарафанюк Руслан Олександрович

Керівник: к.т.н., доц. Антоненко Артем Васильович

Київ - 2025