Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«___» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітній ступень бакалавр

освітня програма Кібербезпека
(назва освітньо-професійної програми)

на тему: «Засоби захисту інформації в інформаційних системах»

Виконавець: студент IV курсу, групи КБ-42

_____ Данило ІЗОТОВ
(підпис) (ім’я, прізвище)

	Підпис	**Ім’я ПРІЗВИЩЕ**
Керівник		Лариса МИРУТЕНКО
Нормоконтроль		Сергій ДАКОВ

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-42** (група) **Ізотову Данилу Антоновичу** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Засоби захисту інформації в хмарних середовищах

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ
Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ
Модель хмарних обчислень, інструменти безпеки AWS (IAM, CloudTrail, KMS), стандарти ISO/IEC 27017 та NIST SP 800-144.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ
Аналіз загроз у хмарному середовищі, огляд стандартів безпеки, налаштування базових сервісів захисту в AWS, розробка рекомендацій із безпечного використання хмарних платформ.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ
Практична цінність Надано практичні рекомендації щодо вибору хмарних сервісів та реалізації захисту даних у середовищі AWS.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видав _____ Лариса МИРУТЕНКО
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Данило ІЗОТОВ
(підпис) (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Найменування етапів робіт	Строки виконання (початок – кінець)	Відмітка про виконання
1	Уточнення теми, постановка мети, об'єкта, предмета та завдань дослідження	29.11.2024 – 22.01.2025	виконано
2	Аналіз джерел, нормативної бази, класифікації загроз	29.01.2025 – 11.02.2025	виконано
3	Обґрунтування вибору платформи для реалізації (AWS) та архітектури захисту	12.02.2025 – 15.02.2025	виконано
4	Концепція хмарних обчислень та їх безпекові особливості	16.02.2025 – 04.03.2025	виконано
5	Аналіз сучасних проблем безпеки в хмарних середовищах	05.03.2025 – 21.03.2025	виконано
6	Практична реалізація засобів захисту в AWS (IAM, KMS, CloudTrail, Config тощо)	22.03.2025 – 08.04.2025	виконано
7	Розробка рекомендацій та моделі безпеки на основі отриманих результатів	09.04.2025 – 10.05.2025	виконано
8	Оформлення пояснювальної записки, списку джерел, висновків, додатків	11.05.2025 – 01.06.2025	виконано
9	Підготовка до захисту кваліфікаційної роботи (презентація, реферат, коригування згідно зауважень)	02.06.2025 – 13.06.2025	виконано

Завдання видав _____ Лариса МИРУТЕНКО
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Данило ІЗОТОВ
(підпис) (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 66 сторінок, включає в себе зміст, вступ, три розділи кваліфікаційної роботи, висновки та список джерел. У пояснювальній записці представлено 25 рисунків,4 таблиці.

Метою роботи є впровадження засобів захисту хмарних сервісів з урахуванням актуальних загроз, нормативних вимог та особливостей архітектури обчислювальних платформ.

Для досягнення поставленої мети були визначені та вирішені такі основні завдання:

- здійснено огляд сучасного стану хмарних технологій і класифікацію загроз безпеці в цих середовищах;
- проаналізовано принципи організації захисту інформації та нормативно-правове забезпечення у сфері хмарних обчислень;
- побудовано модель безпеки з урахуванням особливостей хмарного середовища та використаних сервісів AWS;
- реалізовано практичні заходи з налаштування доступу, моніторингу, шифрування та журналювання в середовищі Amazon Web Services;

Об'єктом дослідження є процес забезпечення інформаційної безпеки у хмарних інфраструктурах.

Предметом дослідження є засоби та сервіси, що реалізують механізми захисту інформації в хмарних обчисленнях.

Практична цінність отриманих результатів полягає у можливості впровадження побудованої моделі безпеки в корпоративних або державних інформаційних системах, що використовують публічні або гібридні хмарні сервіси.

Ключові слова: хмарні обчислення, інформаційна безпека, шифрування, конфіденційність, ризики, доступ, журналювання.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AWS	—	Amazon Web Services
API	—	Application Programming Interface (інтерфейс прикладного програмування)
AD	—	Active Directory
BYOK	—	Bring Your Own Key (власний ключ шифрування)
CI/CD	—	Continuous Integration / Continuous Deployment (безперервна інтеграція / розгортання)
CSA	—	Cloud Security Alliance
DLP	—	Data Loss Prevention (запобігання витоку даних)
EKS	—	Elastic Kubernetes Service
ENISA	—	European Union Agency for Cybersecurity
GCP	—	Google Cloud Platform
GDPR	—	General Data Protection Regulation (Загальний регламент захисту даних)
HIPAA	—	Health Insurance Portability and Accountability Act
IAM	—	Identity and Access Management (керування ідентифікацією та доступом)
IaaS	—	Infrastructure as a Service
KMS	—	Key Management Service (сервіс керування ключами)
MFA	—	Multi-Factor Authentication (багатофакторна автентифікація)
NIST	—	National Institute of Standards and Technology
OWASP	—	Open Web Application Security Project
RBAC	—	Role-Based Access Control (контроль доступу на основі ролей)

S3	—	Simple Storage Service (сервіс зберігання об'єктів AWS)
SaaS	—	Software as a Service
SIEM	—	Security Information and Event Management
SSE-KMS	—	Server-Side Encryption with AWS KMS
STS	—	Security Token Service
VPN	—	Virtual Private Network (віртуальна приватна мережа)
Zero Trust	—	Модель нульової довіри (Zero Trust Architecture)

ВСТУП

Актуальність теми зумовлена стрімким зростанням обсягів конфіденційної інформації, що обробляється у хмарних середовищах, а також необхідністю дотримання вимог інформаційної безпеки відповідно до міжнародних стандартів і нормативно-правової бази. Із переходом державних організацій, комерційних структур і приватних користувачів до хмарної інфраструктури критично зростає потреба в інтегрованому захисті даних, що передбачає застосування технічних і організаційних засобів безпеки. Сучасні тенденції цифрової трансформації вимагають впровадження гнучких, масштабованих і водночас захищених рішень, серед яких хмарні сервіси відіграють провідну роль.

Стрімкий розвиток хмарних обчислень докорінно змінив підходи до зберігання, обробки та управління інформацією в інформаційно-телекомунікаційних системах. Такі платформи, як Amazon Web Services, Google Cloud Platform та Microsoft Azure, стали основою цифрової інфраструктури для підприємств різних масштабів у всьому світі. Проте водночас із перевагами масштабованості, економічної ефективності та доступності, хмарні середовища породжують низку специфічних ризиків — зокрема через мультиорендарність, динамічність доступу до ресурсів, а також обмежений контроль користувача над інфраструктурою.

Забезпечення конфіденційності, цілісності та доступності інформації у хмарному середовищі стає все більш актуальним завданням. Зростає інтерес до класифікації загроз, організації контролю доступу, використання засобів шифрування, журналювання дій, аудиту конфігурацій та забезпечення відповідності стандартам. Водночас практична реалізація таких заходів у межах конкретних хмарних платформ залишається недостатньо опрацьованою, що ускладнює їх застосування в реальних умовах.Таким чином, дослідження питань забезпечення інформаційної безпеки у хмарних обчисленнях є своєчасним і практично значущим, оскільки дозволяє поєднати теоретичні

аспекти кібербезпеки з практикою впровадження ефективних засобів захисту даних у публічному хмарному середовищі.

Актуальність теми підтверджується динамікою зростання ринку хмарних обчислень (див. додаток А).

Метою даної кваліфікаційної *роботи* є реалізація засобів забезпечення інформаційної безпеки в хмарному середовищі шляхом впровадження сучасних інструментів контролю доступу, шифрування, моніторингу та аудиту, зокрема на базі AWS.

Завдання дослідження:

- здійснити аналіз типових загроз та ризиків у хмарному середовищі;
- розглянути принципи організації систем захисту інформації;
- дослідити нормативно-правову базу у сфері хмарної безпеки;
- побудувати артхітектуру захисту з урахуванням особливостей хмарного середовища;
- реалізувати практичні засоби захисту в середовищі AWS.

Практична цінність отриманих результатів полягає в розробці моделі безпеки для хмарного середовища на базі сервісів Amazon Web Services, яка може бути використана як основа для впровадження захисних заходів у реальних проєктах — як у державних установах, так і в приватних компаніях. Реалізовані в роботі рішення охоплюють основні компоненти сучасної системи захисту — контроль доступу, шифрування, журналювання, перевірку конфігурацій, — що дозволяє не лише знизити ризики витоку даних, а й відповідати вимогам міжнародних стандартів.

Об'єкт дослідження – процес забезпечення інформаційної безпеки в хмарних обчислювальних середовищах.

Предмет дослідження – засоби та технології захисту даних у хмарній інфраструктурі, зокрема механізми контролю доступу, шифрування, аудит конфігурацій та моніторинг подій.

Галузь застосування – результати кваліфікаційної роботи можуть бути використані для побудови захищених хмарних архітектур в установах, що

використовують або планують використовувати публічні хмарні сервіси, а також як основа для подальших досліджень у галузі кібербезпеки.

Апробація результатів дослідження здійснювалася у процесі реалізації практичних сценаріїв на базі платформи AWS, з використанням її вбудованих сервісів безпеки, а також при підготовці презентаційних матеріалів у межах захисту кваліфікаційної роботи.

РОЗДІЛ 1. АНАЛІЗ ПІДХОДІВ ДО ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ХМАРНОМУ СЕРЕДОВИЩІ

1.1. Класифікація моделей надання і розгортання послуг

У сучасних інформаційно-телекомунікаційних системах хмарні обчислення стали ключовим елементом трансформації цифрової інфраструктури, що дозволяє забезпечити гнучкість, масштабованість та економічну ефективність оброблення й зберігання інформаційних ресурсів. Згідно з визначенням Національного інституту стандартів і технологій США, хмарні обчислення — це підхід до організації IT-інфраструктури, що передбачає доступ користувачів до обчислювальних ресурсів (включно з мережами, серверами, сховищами та програмними сервісами) через мережу за запитом, із можливістю динамічного масштабування та мінімального втручання в адміністрування.

Ця модель значною мірою змінює підхід до побудови IT-інфраструктур, оскільки забезпечує динамічне управління обчислювальними потужностями та орієнтована на використання сервісно-орієнтованої архітектури. Водночас виникає низка викликів, зокрема у сфері забезпечення інформаційної безпеки, з огляду на особливості передачі, обробки та зберігання даних у спільному, часто публічному середовищі.

Моделі надання хмарних послуг класифікуються за рівнем абстракції доступних користувачу ресурсів та ступенем розподілу відповідальності між споживачем і провайдером. Виділяють три базові моделі:

- IaaS (Infrastructure as a Service) — модель, за якої користувач отримує у своє розпорядження віртуалізовані інфраструктурні ресурси: обчислювальні потужності, сховища даних, мережеві сервіси. У цьому випадку саме користувач несе відповідальність за безпеку операційної системи, прикладного програмного забезпечення та даних, тоді як провайдер забезпечує захист на рівні фізичної інфраструктури та віртуалізації.

- PaaS (Platform as a Service) — надає користувачу повноцінне середовище для розробки, тестування, розгортання та супроводу програмних рішень. Провайдер повністю управляє інфраструктурою, операційною системою та проміжним програмним забезпеченням, що знижує ризики, пов’язані з адмініструванням, проте створює залежність від внутрішніх політик безпеки платформи.

SaaS (Software as a Service) — модель, у межах якої споживач користується функціоналом прикладного програмного забезпечення, доступного через веб-інтерфейс або API. Повна відповідальність за інфраструктуру, платформу та програму лежить на провайдері, тоді як користувач зосереджується лише на введенні, обробці й захисті власних даних.

Оскільки архітектура хмарних обчислень охоплює не лише програмну або сервісну логіку, але й спосіб організації самої інфраструктури, важливим аспектом класифікації виступає модель розгортання хмарного середовища. Вона визначає рівень контролю над даними, ступінь ізоляції користувача, а також способи реалізації політик доступу, шифрування та відповідальності сторін.

Щоб краще зрозуміти особливості хмарних сервісів з погляду рівня контролю, відповідальності сторін і характеру безпекових ризиків, доцільно розглянути відмінності між базовими моделями надання послуг — IaaS, PaaS та SaaS. Узагальнені характеристики цих моделей подано в таблиці 1.1.

Як видно з таблиці, найбільшу гнучкість у налаштуванні та контролі користувач отримує в моделі IaaS, але саме вона й вимагає найбільшої відповідальності у забезпеченні безпеки — зокрема, налаштування ОС, політик доступу, моніторингу та шифрування. У PaaS і SaaS ризики зміщуються в бік постачальника, що знижує навантаження на користувача, але також обмежує його контроль над захистом. Це потрібно враховувати при виборі моделі для розгортання критичних інформаційних систем.

Таблиця 1.1

Порівняння моделей надання хмарних послуг

Критерій порівняння	IaaS	PaaS	SaaS
Контроль користувача	ОС, ПЗ, дані	Додатки, дані	Лише дані
Контроль провайдера	Фізична інфраструктура, гіпервізор	ОС, проміжне ПЗ, середовище виконання	Уся інфраструктура, ПЗ, безпека доступу
Рівень гнучкості	Користувач повністю контролює конфігурацію ОС, мережеві правила, типи сховищ, балансувальники навантаження тощо	Користувач може налаштовувати логіку застосунку, середовище виконання та API, але не має доступу до інфраструктури	Користувач має лише доступ до функціональності додатку. Він не може змінювати логіку його роботи або архітектуру, однак отримує готовий продукт без потреби технічного супроводу
Типові приклади	AWS EC2, Microsoft Azure VM	Google App Engine, Heroku	Google Workspace, Microsoft 365
Основні ризики безпеки	Неправильне конфігурування, уразливості ОС	Компрометація середовища, помилки API	Несанкціонований доступ, витік даних

Моделі розгортання класифікуються відповідно до того, хто є власником, адміністратором та користувачем ресурсів, а також за ступенем спільного використання інфраструктури між клієнтами. Згідно з ISO/IEC 17788:2014 [1] та рекомендаціями NIST SP 800-145 [2], виділяють чотири основні моделі: публічна, приватна, гібридна та громадська хмара. Кожна з них має відмінності у способах управління, технічному забезпеченні безпеки та варіантах практичного застосування.

- Публічна хмара (Public Cloud) — модель, за якої обчислювальні ресурси фізично належать третій стороні (хмарному провайдеру) та спільно використовуються багатьма клієнтами. Забезпечення конфіденційності та ізоляції даних досягається засобами логічного розділення, шифрування та автентифікації.
- Приватна хмара (Private Cloud) — інфраструктура експлуатується однією організацією або розміщується у власному центрі обробки даних, або управляється стороннім провайдером. Забезпечує вищий рівень контролю та гнучкості у впровадженні політик безпеки.
- Гібридна хмара (Hybrid Cloud) — поєднує переваги приватної та публічної хмари, дозволяючи здійснювати переміщення навантажень між середовищами. Вимагає узгодженості засобів захисту та ефективного управління каналами інтеграції.
- Громадська хмара (Community Cloud) — спільно використовуване середовище, що належить кільком організаціям із подібними вимогами до безпеки. Часто використовується у сфері охорони здоров’я, державному секторі та наукових установах.

Окрім відмінностей у моделі надання послуг, критично важливим чинником при проєктуванні хмарної інфраструктури є спосіб її розгортання.У таблиці 1.2 узагальнено основні характеристики публічної, приватної, гібридної та громадської моделей у розрізі ключових параметрів управління й захисту.

Таблиця 1.2

Порівняльна характеристика моделей розгортання хмарних середовищ

Критерій	Публічна хмара	Приватна хмара	Гібридна хмара	Громадська хмара
Власник інфраструктури	Зовнішній провайдер (AWS, Azure, GCP)	Організація або обраний провайдер	Змішаний: організація + провайдер	Кілька організацій, що мають спільні вимоги
Рівень контролю користувача	Обмежений, залежить від наданого сервісу	Повний контроль над усією інфраструктурою	Контроль розподілений між компонентами (локальними та хмарними)	Контроль розподілений за регламентом між учасниками
Ізоляція даних	Логічна (через віртуалізацію, tenancy)	Фізична або логічна, залежно від реалізації	Частково фізична (локально), частково логічна (у хмарі)	Спільна політика логічної ізоляції
Безпека даних	Високий ризик, потребує довіри до провайдера; критично важливе шифрування, MFA, IAM	Висока безпека завдяки повному контролю політик доступу, мережі та шифрування	Залежить від ефективної інтеграції безпеки між приватним та публічним сегментами	Визначається міжорганізаційною угодою; часто адаптується до нормативів державного рівня
Відповідність стандартам	Залежить від провайдера: більшість сертифіковані за ISO/IEC 27001, GDPR	Повністю адаптується під внутрішні або галузеві стандарти	Ускладнена координацією, часто потребує мультисертифікацій	Може регламентуватись окремими міжвідомчими протоколами
Масштабованість	Максимальна, автоматизована	Обмежена потужностями локальної інфраструктури	Висока у публічній частині, обмежена – у приватній	Середня, залежить від домовленостей між учасниками

Продовження до таблиці 1.2

Порівняльна характеристика моделей розгортання хмарних середовищ

Керованість	Мінімальні зусилля користувача	Високі вимоги до IT-персоналу	Складна координація між компонентами	Координація здійснюється спільно, часто – через централізовані органи
Приклади застосування	Стартапи, мобільні застосунки, масштабовані вебсервіси, резервне копіювання	Банки, держустанови, критичні об’єкти інфраструктури	Великі корпорації з розгалуженою мережею, які поєднують власні ЦОД з публічними сервісами	Галузеві об’єднання (медичні, наукові, держсектор), що мають спільні IT-ресурси

Зіставлення показує, що жодна з моделей не є універсальною — кожна має свої сильні сторони та обмеження. Приватні хмари забезпечують найвищий рівень контролю, проте поступаються в гнучкості. Публічні, навпаки, максимально масштабовані, але вимагають ретельного підходу до конфіденційності. Гібридні рішення намагаються поєднати переваги обох, хоча їх реалізація зазвичай складніша. Громадські хмари залишаються рідкісним, але перспективним підходом для галузевих об’єднань з уніфікованими вимогами.

1.2. Класифікація загроз безпеці в хмарному середовищі

Однією з ключових проблем, яка супроводжує стрімкий розвиток хмарних технологій, є забезпечення належного рівня інформаційної безпеки. Відмова від централізованої архітектури на користь розподілених обчислень, широке використання віртуалізації, багатокористувацьке середовище, а також передача контролю над значною частиною інфраструктури сторонньому провайдеру — усе це створює нові вектори атак і знижує ефективність класичних моделей захисту.

З огляду на це, у науковій та прикладній літературі сформовано низку підходів до класифікації загроз, що виникають під час експлуатації хмарних сервісів. Основну увагу дослідники приділяють трьом найбільш авторитетним джерелам систематизації: Європейському агентству з мережевої та інформаційної безпеки (ENISA) [3], альянсу Cloud Security Alliance (CSA) [4] та проєкту OWASP [5], орієнтованому на безпеку вебзастосунків, зокрема в хмарному середовищі.

1.2.1. **Загрози за класифікацією ENISA**

ENISA ще у 2009–2012 роках здійснила одне з перших комплексних досліджень ризиків у хмарних обчисленнях, виокремивши понад 30 типів загроз. Цей перелік надалі був оновлений і уточнений відповідно до змін у технологіях. Агентство класифікує загрози за походженням (внутрішні/зовнішні), типом впливу (на конфіденційність, цілісність, доступність), а також за рівнем реалізації (інфраструктура, платформа, застосунок).

До критичних загроз, які найбільш часто згадуються у звітах ENISA, належать:

- порушення політик доступу та автентифікації;
- атаки через неналежно захищені API;
- міжорендні атаки в середовищі гіпервізора;
- експлуатація вразливостей у компонентах віртуалізації;
- недостатня сегментація мережевої інфраструктури;
- людський фактор і помилки конфігурації.

Особливу увагу приділено також таким загрозам, як втрата керованості над даними після їх передачі у хмару, залежність від одного постачальника послуг (vendor lock-in) та порушення вимог до географічного розміщення даних, які мають правовий характер.

1.2.2. **Загрози за класифікацією Cloud Security Alliance**

Cloud Security Alliance регулярно публікує аналітичні звіти під назвою Top Threats to Cloud Computing [4], в яких наводяться найактуальніші загрози, виявлені під час аналізу інцидентів у провідних хмарних платформах. На відміну від більш широкої та академічної моделі ENISA [3], CSA концентрується на практичних кейсах і реальних помилках конфігурації.

У звіті 2022 року CSA виокремила такі ключові загрози:

- неправильне керування ідентифікацією та доступом;
- витоки облікових даних (через фішинг, слабкі паролі, недостатню автентифікацію);
- невірно налаштовані об’єкти зберігання (наприклад, відкритий доступ до AWS S3);
- небезпека з боку зловмисників, які вже мають доступ до інфраструктури (insider threats);
- відсутність шифрування при зберіганні або передаванні даних;
- уразливості в логіці багатокористувацького середовища;
- атаки через відмову в обслуговуванні (DDoS).

На відміну від загальних технічних описів, CSA надає практичні рекомендації щодо усунення вразливостей, враховуючи специфіку IaaS, PaaS та SaaS-сервісів. Такий підхід дозволяє не лише ідентифікувати загрозу, а й безпосередньо інтегрувати захисні механізми у відповідну архітектуру.

1.2.3. **Загрози за класифікацією OWASP Cloud Top 10**

Організація OWASP [5], що спеціалізується на питаннях безпеки вебзастосунків, у своїй класифікації Cloud-Native Application Security Top 10 зосереджується на специфіці сучасних застосунків, розгорнутих у хмарі. Ці загрози переважно стосуються DevOps-процесів, CI/CD-пайплайнів, контейнеризації (Docker, Kubernetes), а також відкритих API.

До ключових загроз, виділених OWASP, належать:

- погано захищені облікові записи з широкими правами доступу;
- незахищені або неправильно реалізовані інтерфейси API;
- відсутність журналювання подій і моніторингу;
- залежність від вразливих бібліотек або сторонніх компонентів;
- відсутність сегментації середовища виконання (наприклад, у Kubernetes).

Ці загрози особливо актуальні в умовах автоматизованого масштабування та коротких життєвих циклів хмарних сервісів, де класичні механізми безпеки, такі як фаєрвол або антивірус, не забезпечують належного захисту.

1.2.4. Приклади атак

Історично найгучнішими прикладами зловживання уразливостями хмарної інфраструктури стали:

- витік даних користувачів Capital One (2019), що був результатом атаки через вразливість у web application firewall на Amazon Web Services;
- інциденти, пов’язані з відкритими хмарами, які призводили до масових витоків персональних даних;
- компрометація внутрішніх компонентів у SolarWinds (2020), що хоч і не була виключно хмарною, продемонструвала вразливість до ланцюгових атак у комплексних цифрових середовищах [6].

Ці приклади ілюструють, що на практиці загрози реалізуються не лише через технічні недоліки, але й через відсутність політик безпеки, неефективне навчання персоналу та недотримання принципів модульності систем.

1.3. Організація захисту інформації у хмарних обчисленнях

З огляду на динамічну природу хмарних сервісів та специфіку взаємодії між постачальниками послуг і кінцевими користувачами, побудова ефективної системи інформаційної безпеки в хмарному середовищі повинна ґрунтуватися на низці фундаментальних принципів. Ці принципи визначають базові підходи

до організації, управління та контролю за обробкою інформації та ресурсів у розподіленому цифровому середовищі.

1.3.1. Ізоляція середовищ

Одним із ключових завдань при організації захисту є забезпечення ізоляції користувацьких середовищ у межах спільної хмарної інфраструктури. В умовах багатокористувацької архітектури (multi-tenancy) існує постійний ризик несанкціонованого доступу до даних або сервісів іншого клієнта внаслідок помилок конфігурації, вразливостей гіпервізора або порушень політик доступу. Забезпечення ізоляції може реалізовуватися на декількох рівнях: логічному (віртуальні мережі, політики firewall), фізичному (виділені ресурси), та програмному (контейнери, sandbox-середовища).

Цей принцип є фундаментальним для моделей IaaS та PaaS, де ресурси фактично розділяються між численними споживачами. Практичними механізмами реалізації виступають VLAN-сегментація, технології віртуалізації з підтримкою hardware-assisted isolation, а також системи контролю контексту доступу.

1.3.2. Багаторівневий контроль доступу

Організація доступу до хмарних ресурсів вимагає застосування багаторівневих моделей контролю, які поєднують аутентифікацію, авторизацію та аудит дій користувачів. Одним із ключових напрямів розвитку систем доступу у хмарі є впровадження багатофакторної автентифікації, що дозволяє зменшити ризики, пов’язані з компрометацією облікових даних.

Окрім того, все ширше застосовуються моделі на основі ролей (RBAC — Role-Based Access Control), атрибутів (ABAC — Attribute-Based Access Control) та контексту (Context-Aware Access). Ці моделі дозволяють налаштувати динамічний доступ з урахуванням не лише прав користувача, але й умов, за яких здійснюється запит (наприклад, з якого географічного розташування чи за допомогою якого пристрою).

У межах системного підходу багаторівневий контроль доступу охоплює не лише кінцеву автентифікацію, але й ізоляцію внутрішніх сегментів інфраструктури, контроль API-запитів, моніторинг дій у реальному часі та подальше логування для виявлення аномальної активності.

1.3.3. **Принцип мінімальних привілеїв**

Цей принцип передбачає, що кожен користувач, процес або сервіс має отримувати лише ті права доступу, які необхідні йому для виконання конкретних функцій, і не більше. Його дотримання дозволяє знизити ризики вертикального та горизонтального ескалування привілеїв у разі компрометації елементів системи.

У хмарній інфраструктурі дотримання принципу мінімальних привілеїв зазвичай досягається шляхом застосування політик управління доступом (IAM), точного визначення прав доступу до окремих об’єктів зберігання (наприклад, через ACL для бакетів Amazon S3), а також автоматизованого обмеження строку дії токенів і ключів автентифікації.

Особливої актуальності цей принцип набуває в умовах DevOps-процесів, де розгортання сервісів здійснюється автоматично і з великою частотою. Неналежне обмеження прав у таких умовах може призвести до масштабних порушень безпеки, які складно локалізувати у реальному часі.

1.3.4. **Принцип довіри за замовчуванням відсутній**

У сучасних хмарних архітектурах все частіше впроваджується модель «нульової довіри» (Zero Trust Architecture, ZTA), відповідно до якої не існує автоматично довірених зон, користувачів чи пристроїв — кожен запит повинен бути перевірений незалежно від того, з якої частини системи він надходить. Такий підхід передбачає постійне оцінювання контексту доступу, використання мікросегментації мережі, автоматизованих засобів верифікації та поведінкової аналітики.

Zero Trust є не стільки технічним механізмом, скільки філософією побудови інформаційної безпеки, яка особливо релевантна для гібридних та мультихмарних середовищ, де межі традиційних периметрів фактично розмиті.

Таким чином, ефективна побудова захисту у хмарному середовищі передбачає впровадження взаємопов’язаних принципів, кожен з яких спрямований на зниження конкретної категорії ризику. Їх практична реалізація дає змогу створити стійку архітектуру безпеки, здатну адаптуватися до динамічних змін конфігурацій, сервісів та моделей використання. У подальших підпунктах ці принципи розглядатимуться у контексті нормативного регулювання та управління ризиками.

1.4. Нормативно-правова база регулювання безпеки хмарних технологій

У контексті стрімкого впровадження хмарних технологій усе більшого значення набуває відповідність цифрової інфраструктури чинним нормативно-правовим та галузевим вимогам. Забезпечення інформаційної безпеки у хмарному середовищі не обмежується суто технічними аспектами — воно повинно базуватися на узгоджених стандартах, регламентованих процедурах та механізмах контролю, затверджених на міжнародному й національному рівнях. Саме нормативне підґрунтя визначає рамки допустимих підходів до захисту даних, розмежування відповідальності між учасниками взаємодії, а також вимоги до управління ризиками.

1.4.1. Міжнародні стандарти з безпеки хмарних середовищ

На глобальному рівні фундаментом для впровадження політик безпеки у хмарі є стандарти родини ISO/IEC, насамперед:

- ISO/IEC 27001:2013 — стандарт управління інформаційною безпекою, який задає загальні вимоги до побудови, впровадження, підтримки та вдосконалення системи управління інформаційною безпекою (СУІБ).

- ISO/IEC 27017:2015 — спеціалізоване розширення до ISO/IEC 27002, що встановлює рекомендації щодо засобів управління безпекою в хмарних сервісах, як для постачальників, так і для споживачів.
- ISO/IEC 27018:2019 — надає рекомендації щодо захисту персональних даних у публічному хмарному середовищі відповідно до принципів приватності.

Ці стандарти визначають мінімально необхідний рівень безпеки для організацій, які розміщують, обробляють або передають інформацію в хмарній інфраструктурі, а також орієнтовані на аудит і сертифікацію.

1.4.2. **Рекомендації NIST щодо безпеки хмарних обчислень**

На рівні США одним із найбільш авторитетних джерел є документи Національного інституту стандартів і технологій**.** Зокрема, NIST SP 800-144: Guidelines on Security and Privacy in Public Cloud Computing [2], містить огляд типових ризиків публічних хмар, рекомендації з побудови безпечної архітектури та орієнтири для замовників щодо вибору провайдера.

Документ охоплює такі аспекти:

- безпека віртуалізації та гіпервізорів;
- ізоляція клієнтських середовищ;
- вимоги до логування, моніторингу та відповідальності провайдерів;
- захист метаданих і конфіденційної інформації у багатокористувацькому середовищі.

NIST SP 800-145, який містить офіційне визначення хмарних обчислень, також використовується як базовий норматив при стандартизації термінів та моделей впровадження сервісів.

1.4.3. **Вимоги Європейського Союзу до захисту даних**

У межах Європейського Союзу нормативне регулювання захисту даних у хмарі здійснюється, перш за все, через Загальний регламент про захист даних [7]**.** Його положення мають екстериторіальний характер і поширюються на будь-які компанії, що обробляють персональні дані громадян ЄС, незалежно від географічного розташування хмарного сервісу.

Вимоги GDPR стосуються:

- явної згоди на обробку персональних даних;
- права на доступ, виправлення, видалення та перенесення даних;
- призначення відповідального за захист даних (DPO);
- сповіщення про витік протягом 72 годин;
- мінімізації обсягу даних і часу зберігання.

Хмарні провайдери, що не дотримуються вимог GDPR, можуть бути притягнуті до значних штрафних санкцій, що надає цьому регламенту практично-орієнтоване значення для компаній, що використовують хмарну інфраструктуру в межах або на ринку ЄС.

1.4.4. **Українське законодавство захисту інформації у хмарних середовищах**

Національна нормативна база в Україні також містить ряд документів, що регулюють сферу інформаційної безпеки, які мають застосування до хмарних обчислень:

- Закон України «Про інформацію» (редакція 2011 року) окреслює ключові положення, що регулюють інформаційні відносини в державі, зокрема право громадян на доступ до відомостей та механізми відповідальності у разі порушення цього права [8].
- Закон «Про захист персональних даних» встановлює засади для обробки і збереження персональних відомостей, включаючи правила їх передачі, заходи безпеки та розмежування обов’язків між володільцем і розпорядником даних [9].

- Вимоги до створення комплексних систем захисту інформації формуються на основі національних нормативів, таких як НД ТЗІ 2.5-010-03 та НД ТЗІ 2.6-001-11, які визначають критерії побудови, атестації та підтримки ІКС, зокрема в умовах використання хмарних технологій.

Як показано в попередніх підпунктах, нормативно-правове забезпечення інформаційної безпеки у хмарному середовищі має багаторівневу структуру. На міжнародному рівні діють стандарти ISO/IEC, які задають вимоги до організаційної та технічної побудови систем захисту. У США такі вимоги деталізуються через рекомендації NIST, що мають прикладний характер для провайдерів і споживачів хмарних послуг. Європейське законодавство акцентує увагу на захисті персональних даних через жорсткі вимоги GDPR, тоді як в Україні застосовуються положення законів «Про захист персональних даних», «Про інформацію», а також нормативи НД ТЗІ щодо створення КСЗІ.

Особливої уваги заслуговує документ НД ТЗІ 2.6-001-11, який встановлює вимоги до побудови комплексної системи захисту інформації в інформаційно-телекомунікаційних системах, зокрема і при використанні хмарних технологій. Цей норматив вимагає проведення класифікації інформації, визначення зон доступу, організації контролю повноважень користувачів та впровадження засобів технічного захисту інформації, що підтверджується сертифікацією відповідних рішень. Виконання положень КСЗІ є обов’язковим для державних органів та підприємств, які працюють з конфіденційними або персональними даними.

Для зручності аналізу та порівняння основних положень цих документів доцільно представити їх в узагальненій формі. У таблиці 1.3 наведено порівняльну характеристику нормативно-правових актів і стандартів, що регулюють безпеку даних у хмарних середовищах, із зазначенням їх юрисдикції, сфери застосування та ключових вимог.

Таблиця 1.3

Нормативно-правові акти та стандарти у сфері захисту інформації в хмарному середовищі

Документ / Стандарт	Юрисдикція	Сфера охоплення	Основні положення
ISO/IEC 27017:2015	Міжнародна	Безпека хмарних сервісів	Рекомендації для провайдерів і користувачів хмар щодо контролю доступу, шифрування, логування
ISO/IEC 27018:2019	Міжнародна	Приватність у публічній хмарі	Визначає правила обробки персональних даних у хмарі
NIST SP 800-144	США	Публічні хмарні обчислення	Визначає типові ризики, принципи захисту, критерії вибору безпечного провайдера
GDPR	ЄС	Персональні дані	Право на забуття, згода на обробку, обов’язкове сповіщення про витік, мінімізація даних
Закон України «Про захист ПД»	Україна	Персональні дані, бази даних	Вимоги до згоди, обмеження передачі, відповідальність за витік
НД ТЗІ 2.6-001-11	Україна	Комплексна система захисту інформації (КСЗІ)	Визначає вимоги до систем з технічним захистом інформації, зокрема в ІКС, у тому числі хмарного типу

1.5. Управління ризиками в хмарних середовищах

У хмарних обчисленнях традиційне уявлення про управління ризиками потребує переосмислення, оскільки архітектура таких середовищ істотно відрізняється від класичних IT-систем. В умовах, коли IT-інфраструктура не повністю контролюється замовником, а багато компонентів делеговано сторонньому постачальнику послуг, розподіл відповідальності між сторонами та визначення зон ризику стає принципово важливим.

Першим кроком у формуванні ефективної моделі управління ризиками є чітка ідентифікація активів, які підлягають захисту. У хмарі до активів належать не лише очевидні елементи — дані, облікові записи чи сервіси — а й менш очевидні ресурси: ключі шифрування, метадані, журнали подій, API-точки інтеграції, об’єкти зберігання, контейнеризовані модулі. У хмарних моделях IaaS ці активи здебільшого розміщені у віртуальних середовищах, що потребують ізоляції, шифрування та жорсткого контролю привілеїв. У SaaS же активами стають не самі інфраструктурні ресурси, а логіка доступу до інформації та налаштування користувацького профілю.

Після ідентифікації активів необхідно здійснити типізацію загроз, які мають потенціал їх скомпрометувати. Найбільш поширеними є:

- загрози, пов’язані з людським фактором: помилки конфігурації, публічний доступ до приватних ресурсів, слабкі паролі, відсутність MFA;
- інфраструктурні загрози: атаки через міжорендні канали, експлуатація вразливостей гіпервізора, DoS-атаки на загальнодоступні елементи;
- загрози, пов’язані з API: незахищені інтерфейси, відсутність rate-limiting, небезпечні інтеграції;
- логічні загрози: неправильне розмежування прав, відсутність принципу мінімальних привілеїв, занадто широкі політики доступу;

- загрози приватності: недотримання принципів GDPR, зберігання даних у невизначеній юрисдикції, несанкціоноване використання даних провайдером.

Оцінка ризику відбувається на стику трьох ключових величин: імовірності, вразливості та потенційного впливу. Наприклад, якщо внутрішній адміністратор має повний доступ до всіх ресурсів через один обліковий запис — це точка високого ризику. Імовірність помилки чи зловживання висока, вразливість очевидна, наслідки — масштабні. Такий випадок має бути класифікований як критичний незалежно від контексту використання хмари.

У хмарних середовищах управління ризиками ускладнене тим, що значна частина інфраструктури — поза прямим контролем користувача. Саме тому сучасні моделі управління ризиками включають елемент розмежування відповідальності. Наприклад, у моделі IaaS провайдер забезпечує захист фізичної інфраструктури, віртуалізації та гіпервізора, тоді як клієнт відповідає за налаштування ОС, політики доступу, шифрування даних, оновлення ПЗ. У SaaS більшість ризиків передано провайдеру, але саме клієнт контролює автентифікацію, рівні доступу, життєвий цикл даних.

Укладання договорів SLA із деталізованим описом зон відповідальності дозволяє мінімізувати не лише технічні ризики, а й юридичну невизначеність. У більшості провайдерів (AWS, Azure, Google Cloud) передбачено типові SLA, які можна адаптувати під специфіку організації, зокрема — включити вимоги до географічного розміщення даних, журналювання подій, часу реакції на інциденти.

Ще одним напрямом є інструментальна автоматизація оцінки ризиків. Наприклад:

- AWS Security Hub дозволяє оцінювати налаштування ресурсів за понад 200 критеріями безпеки (CIS, NIST) [10];
- Microsoft Defender for Cloud автоматично формує карту ризиків для кожного ресурсу в Azure і дає рекомендації з усунення [11];

- Google Cloud Security Command Center збирає інформацію про події, оцінює їх критичність, попереджає про потенційні витоки [12].

Усе частіше використовуються інтегровані засоби моніторингу та реагування — SIEM-системи, які збирають дані про події безпеки в режимі реального часу, аналізують поведінкові аномалії, корелюють події між різними джерелами. Наприклад, виявлення входу з нового географічного регіону, одночасно з ініціацією запиту на експорт даних, може бути ознакою атаки, навіть якщо жоден із елементів сам по собі не виглядає небезпечним.

Щодо стратегій управління ризиками, у хмарних середовищах використовуються ті ж самі принципи, що й у традиційних: уникнення, зменшення, передача або прийняття ризику. Проте в умовах хмари вони реалізуються специфічно. Наприклад, зменшення ризику — це не лише встановлення MFA, а й обмеження часу дії токенів, автоматичне скасування прав після завершення доступу, періодична переоцінка ролей. Передача ризику — не тільки делегування задач провайдеру, а й залучення сторонніх сервісів моніторингу, аудитору, кіберстрахової компанії.

Ключова вимога — управління ризиками має бути безперервним процесом. У світі, де хмарні середовища змінюються щоденно, застосування разового аналізу (наприклад, при впровадженні) є недостатнім. Повна картина формується лише за умови постійного аудиту, автоматичної перевірки конфігурацій, відстеження змін у політиках доступу, оновлення ПЗ і аналізу журналів активності.

На рисунку 1.1 наведено загальну логіку циклу управління ризиками в хмарному середовищі, яка включає ключові етапи — від ідентифікації активів до постійного моніторингу та перегляду заходів захисту.

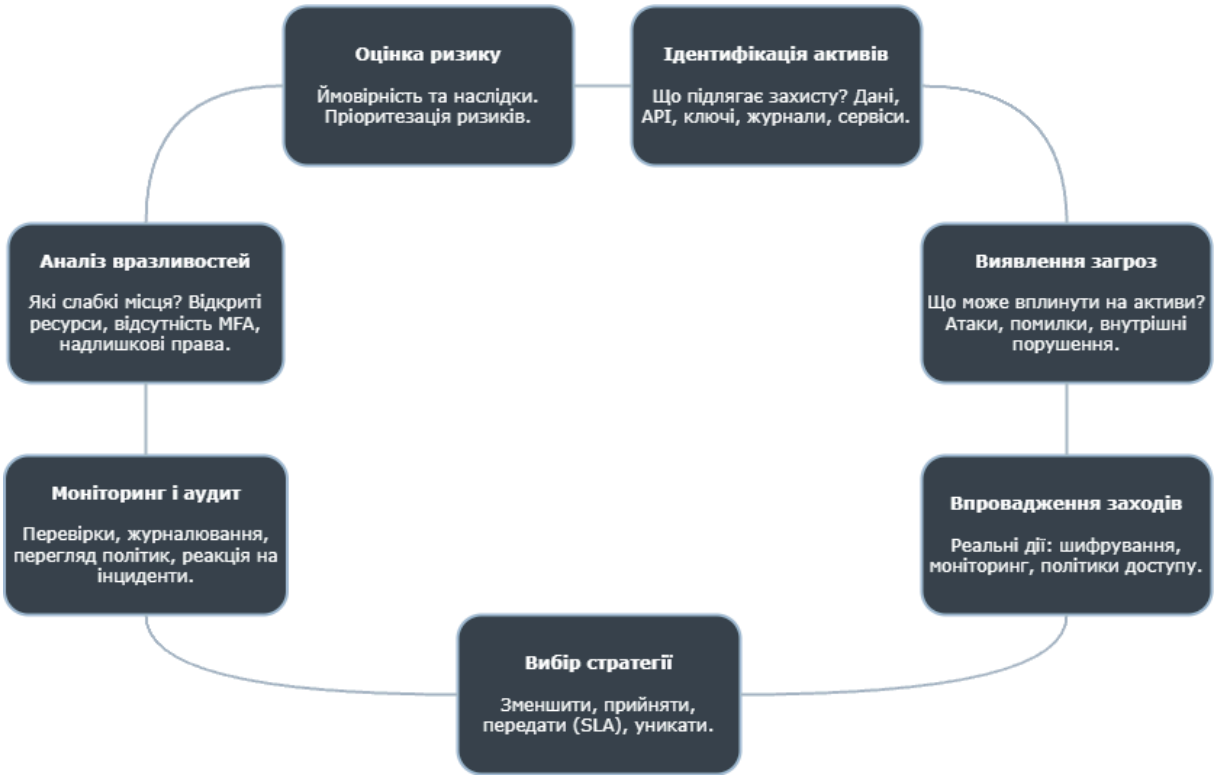

Рисунок 1.1 – Цикл управління ризиками в хмарних обчислювальних середовищах

Представлений рисунок демонструє безперервність процесу управління ризиками, акцентуючи увагу на необхідності регулярної переоцінки загроз та адаптації політик безпеки відповідно до змін у конфігурації хмарної інфраструктури.

Висновок за розділом 1

У процесі дослідження було здійснено огляд сучасних підходів до організації інформаційної безпеки в хмарних обчисленнях, що охопив класифікацію типових загроз, принципи захисту, нормативно-правову базу та методи управління ризиками. Особливу увагу приділено аналізу загроз відповідно до авторитетних джерел — ENISA, CSA та OWASP [3, 4, 5], що дозволило виокремити ключові вектори атак, характерні для хмарної інфраструктури, серед яких: компрометація облікових даних, помилки

конфігурацій, вразливості інтерфейсів API та порушення безпеки на рівні постачальника.

Сформульовано базові принципи побудови ефективної системи захисту в хмарному середовищі, зокрема реалізацію ізоляції ресурсів, обмеження привілеїв користувачів, багаторівневу автентифікацію та централізоване управління доступом. Водночас було опрацьовано нормативну основу, що регламентує безпечне функціонування хмарних сервісів, включаючи міжнародні стандарти та українське законодавство, зокрема Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» і «Про захист персональних даних» [8, 9].

Окрему увагу приділено практичним аспектам управління ризиками: від виявлення активів і оцінки вразливостей до планування реагування на загрози. Це дозволило закласти чітку основу для подальшого вибору захисних механізмів у хмарній інфраструктурі.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЇ У ХМАРНОМУ СЕРЕДОВИЩІ

1.1. Огляд платформ і засобів захисту інформації у хмарних середовищах

У сучасній практиці хмарних обчислень платформи типу IaaS, PaaS і SaaS реалізуються як на базі глобальних інфраструктур великих провайдерів (AWS, Microsoft Azure, Google Cloud Platform) [10, 11, 12], так і у вигляді приватних або гібридних хмар, побудованих на open-source рішеннях (OpenStack, Proxmox, Eucalyptus). Незалежно від моделі розгортання, забезпечення інформаційної безпеки у хмарі передбачає наявність багаторівневих механізмів контролю, адаптованих до розподіленого характеру інфраструктури, мультиорендності, віртуалізації та обмеженої видимості з боку користувача.

Типові засоби захисту у хмарних середовищах поділяються на кілька категорій:

- Ідентифікація та контроль доступу (IAM): дозволяє керувати обліковими записами, ролями, правами доступу до окремих ресурсів та API. Ці системи підтримують автентифікацію за допомогою паролів, ключів, токенів, а також багатофакторну автентифікацію (MFA).
- Захист даних (Data Protection): включає в себе механізми шифрування даних при зберіганні (encryption at rest) та при передачі (encryption in transit), управління ключами, політики контролю життєвого циклу об'єктів (data retention), а також запобігання втраті даних (DLP).
- Моніторинг і аудит (Monitoring, Logging, SIEM): системи для безперервного спостереження за активністю користувачів та сервісів, виявлення аномалій, фіксації інцидентів безпеки. До таких належать CloudTrail (AWS), Azure Monitor, Google Cloud Operations, а також інтеграції з SIEM-рішеннями (Splunk, ELK Stack).
- Управління вразливостями (Vulnerability Management): автоматизовані сервіси для перевірки конфігурацій, виявлення відхилень від

політик безпеки, оцінювання ризиків. Провайдери надають інструменти на кшталт AWS Security Hub, Azure Defender, Google Security Command Center.

- Мережевий захист: включає в себе системи віртуальних файрволів, мережевих ACL, ізоляцію VPC, VPN-тунелі, засоби мікросегментації, а також сервісну взаємодію через HTTPS, TLS та інші захищені протоколи.

Найбільш комплексні та масштабовані системи безпеки реалізуються у хмарних платформах глобального рівня. Наприклад:

- AWS пропонує широкий спектр сервісів безпеки: Identity and Access Management (IAM), Key Management Service (KMS), GuardDuty (виявлення загроз), CloudTrail (аудит дій), Security Hub (агрегація ризиків), WAF та Shield (мережевий захист) [13].
- Microsoft Azure забезпечує інтегровану модель Zero Trust, включаючи Azure Active Directory, Microsoft Defender for Cloud, Azure Information Protection, а також модулі безпечної інтеграції з локальними середовищами.
- Google Cloud Platform (GCP) робить акцент на автоматичній класифікації даних, контроль доступу через Cloud IAM, а також об’єднану систему спостереження за ресурсами через Security Command Center.

На підставі узагальненого аналізу хмарних платформ можна констатувати, що реалізація захисту інформації у такому середовищі вимагає поєднання кількох механізмів, орієнтованих на різні рівні: від керування ідентифікацією та шифруванням даних — до автоматизованого моніторингу і проактивного виявлення ризиків. Особливої ваги набуває здатність адаптувати засоби безпеки до змінного навантаження, гнучкості сервісної моделі та специфіки внутрішніх політик організацій.

1.2. Технології захисту інформації в хмарному середовищі

Забезпечення інформаційної безпеки у хмарній інфраструктурі передбачає впровадження низки взаємопов'язаних механізмів, що охоплюють ідентифікацію користувачів, шифрування даних, контроль доступу, моніторинг подій та виявлення аномальної активності. Особливість таких рішень полягає у необхідності адаптації до розподіленого середовища, де багато компонентів не перебувають під повним контролем користувача, а загрози можуть реалізовуватись як ззовні, так і зсередини хмарної платформи.

1.2.1. Керування ідентифікацією та доступом

Одним з ключових аспектів інформаційної безпеки в хмарних середовищах є контроль над тим, хто і до яких ресурсів має доступ. У середовищах з динамічною інфраструктурою, великою кількістю облікових записів та розподіленими сервісами використання централізованої системи керування ідентифікацією та доступом є обов'язковим.

Основне завдання IAM-сервісів — гарантувати, що доступ до хмарних ресурсів отримують лише ті суб'єкти, які мають відповідні повноваження, і лише на той обсяг дій, який дійсно необхідний. Це дозволяє реалізовувати принцип найменших привілеїв, знижуючи ризики несанкціонованого доступу або ескалації прав.

Усі провідні хмарні платформи мають вбудовані сервіси IAM. Наприклад:

- AWS IAM дає змогу створювати політики доступу, ролі та групи користувачів, а також підтримує MFA, тимчасові токени та умовний доступ [14];
- Azure Active Directory (Entra ID) забезпечує єдиний центр автентифікації з підтримкою SSO, RBAC і умовного доступу;
- Google Cloud IAM застосовує концепцію identity-based permissions та дозволяє гнучко управляти правами доступу на всіх рівнях — від організації до окремого ресурсу.

Щоб показати, як побудовано логіку доступу в хмарному середовищі, розглянемо ієрархічну модель управління правами у Google Cloud Platform. На рисунку 2.1 наведено приклад такої моделі, де політики безпеки застосовуються каскадно — починаючи з рівня організації й закінчуючи окремими сервісами.

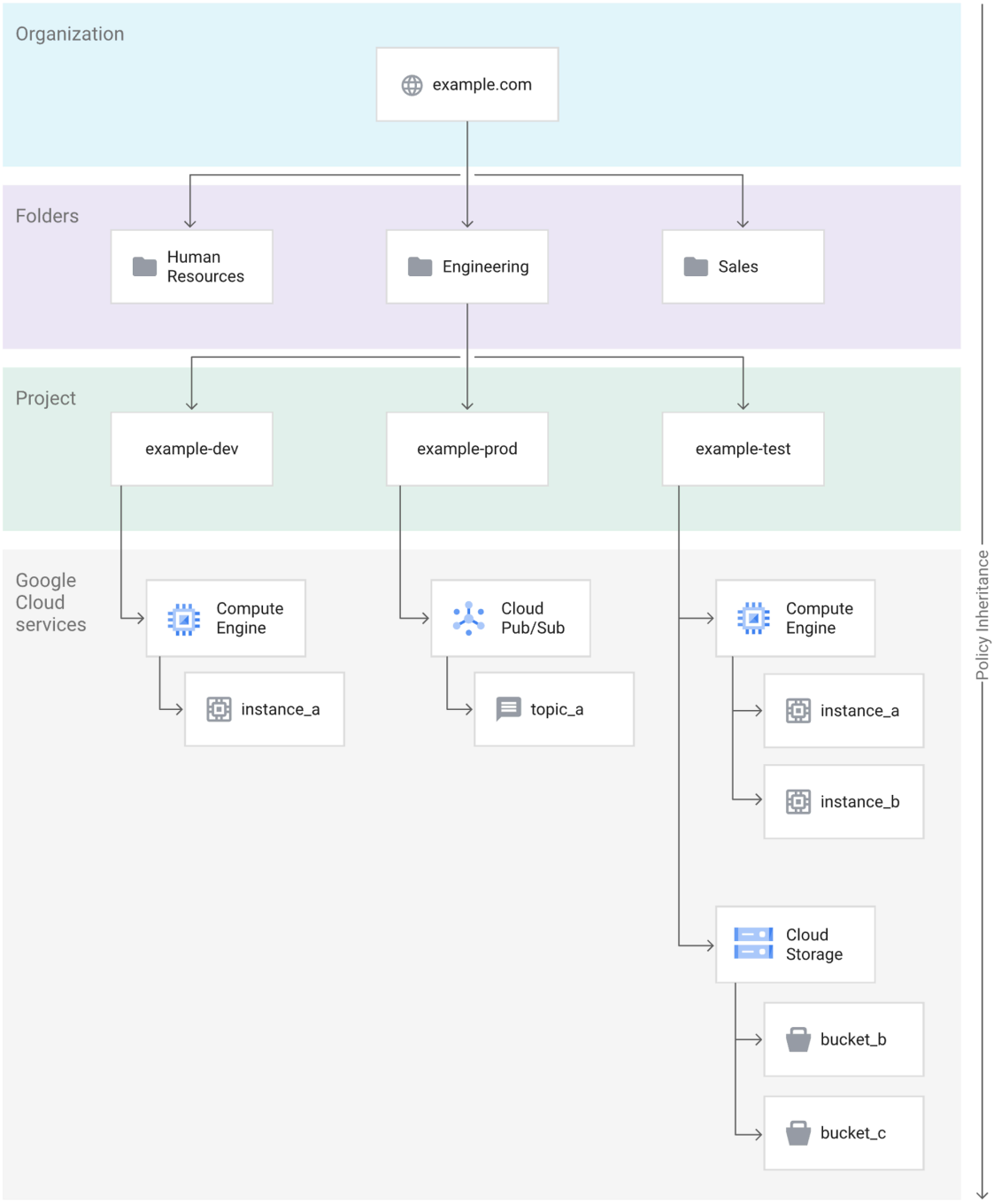

Рисунок 2.1 – Ієрархічна модель управління доступом у GCP

Як видно з рисунку, модель доступу в Google Cloud Platform має ієрархічну структуру: на найвищому рівні розміщується організація, далі — папки, проєкти й конкретні ресурси. Політики доступу (permissions) можуть задаватися на будь-якому рівні, і всі нижчі рівні автоматично їх успадковують. Наприклад, якщо користувачу надано права на рівні папки, він автоматично отримає доступ до всіх проєктів і ресурсів у цій папці, якщо не задано винятків. Це дозволяє централізовано керувати правами й уникати надмірного дублювання політик.

1.2.2. Шифрування та керування ключами

Для гарантування конфіденційності критично важливих даних у хмарі обов'язковим є використання шифрування — як під час зберігання (encryption at rest), так і під час передавання (encryption in transit). Реалізація шифрування, як правило, супроводжується сервісами для управління ключами (Key Management Service, KMS), які дозволяють:

- генерувати ключі автоматично або вручну;
- розмежовувати доступ до ключів (ACL, RBAC);
- автоматично обертати ключі (key rotation);
- логувати всі операції над ключами.

У AWS — це AWS KMS і CloudHSM**,** в Azure **—** Azure Key Vault, у Google **—** Cloud KMS**.**

1.2.3. Моніторинг, журналювання, виявлення загроз

Моніторинг хмарної інфраструктури дозволяє виявляти відхилення від нормальної поведінки, вести аудит операцій, запобігати несанкціонованому доступу. Хмарні провайдери надають для цього цілу низку сервісів:

- AWS CloudTrail — сервіс журналювання дій користувачів, API-запитів, змін конфігурацій [16];
- AWS GuardDuty — сервіс виявлення загроз з використанням ML;

- Azure Monitor + Sentinel — централізоване логування, аналітика і SIEM-функціонал;
- Google Cloud Logging + Security Command Center — аудит, інцидент-менеджмент, поведінкова аналітика.

Перелічені загрози вкотре підтверджують необхідність уважного ставлення до конфігурацій, управління доступом та контролю на всіх рівнях хмарної інфраструктури.

1.2.4. Захист на рівні конфігурацій і політик

Одна з найчастіших причин інцидентів у хмарі — помилки конфігурацій (наприклад, публічний доступ до приватних об’єктів). Для виявлення та запобігання цьому провайдери інтегрують політики перевірки конфігурацій:

- AWS Config**:** служба перевірки відповідності ресурсів політикам безпеки [17].
- Azure Policy**:** дозволяє створювати, тестувати й застосовувати правила відповідності.
- GCP Policy Intelligence**:** виявляє надмірні дозволи, пропонує оптимізації.

Такі інструменти забезпечують постійний контроль за станом конфігурації хмарної інфраструктури, дають змогу виявляти відхилення від стандартів безпеки, автоматизувати перевірку відповідності та зменшувати вплив людського фактора. Їх інтеграція в процеси DevOps і CI/CD сприяє формуванню безпечного середовища на всіх етапах життєвого циклу хмарних ресурсів.

1.2.5. Інтегровані системи безпеки

З метою агрегації результатів з кількох джерел і автоматизації реагування на інциденти застосовуються централізовані сервіси:

- AWS Security Hub — єдиний дашборд для перевірки політик, виявлення ризиків, агрегування даних з GuardDuty, Inspector, IAM Access Analyzer [10];
- Microsoft Defender for Cloud — мультисервісний інструмент для безпеки ресурсів в Azure, AWS, GCP;
- Google Security Command Center — інструмент аналізу стану безпеки, сповіщень та вразливостей.

На рисунку 2.2 зображено архітектуру взаємодії сервісів безпеки AWS . Дані від GuardDuty, Macie, Inspector та інших модулів надходять у центральну систему Security Hub, де аналізуються, агрегуються та передаються на подальшу обробку у CloudWatch, SIEM чи сервіси автоматизації дій, такі як AWS Lambda.

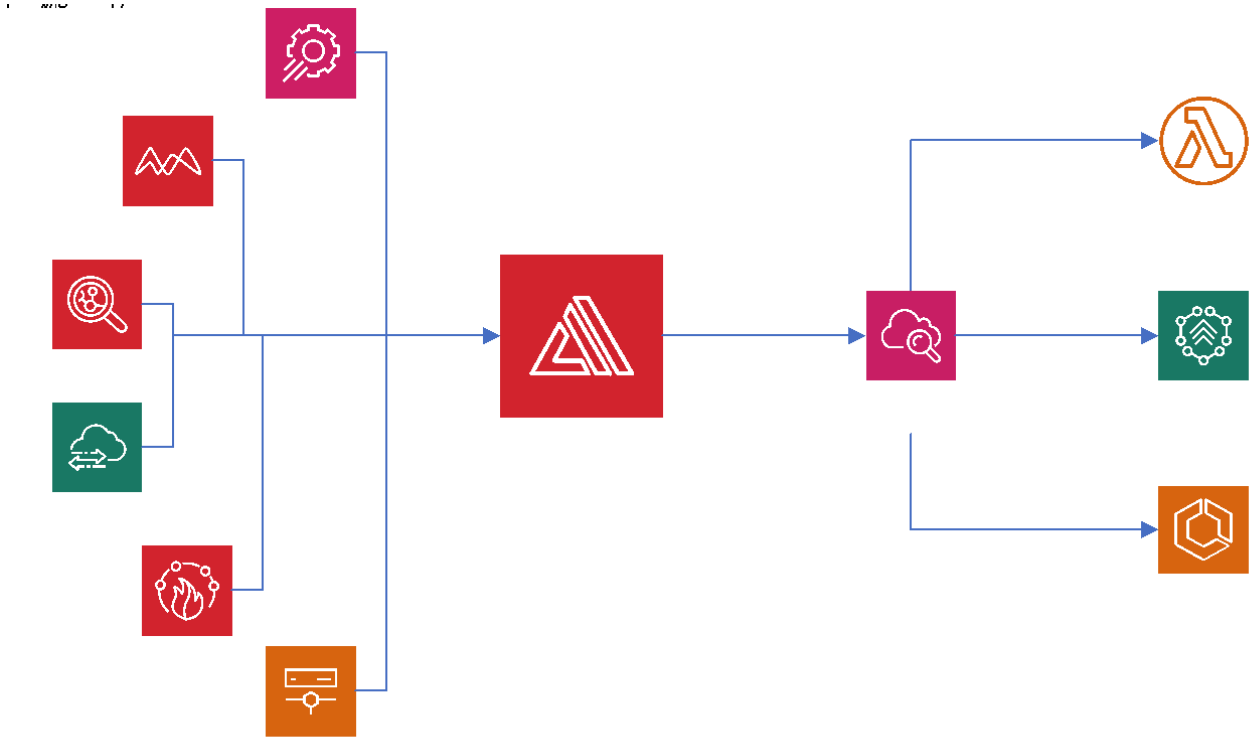

Рисунок 2.2 – Архітектура взаємодії безпекових сервісів AWS через Security Hub

Представлена схема демонструє логіку функціонування інтегрованої системи безпеки в хмарному середовищі AWS, побудованої навколо центрального модуля AWS Security Hub. До нього надходять дані з вбудованих

сервісів виявлення загроз і аналізу безпеки, таких як Amazon GuardDuty**,** Macie**,** Inspector**,** Firewall Manager**,** IAM Access Analyzer, а також із зовнішніх інструментів. Security Hub виконує агрегацію та уніфікацію подій безпеки, передаючи їх до Amazon CloudWatch для подальшого моніторингу та автоматичного реагування. У відповідь на виявлені інциденти можуть запускатися функції AWS Lambda, передаватися дані до SIEM-систем або надсилатися повідомлення адміністраторам.

1.3. Огляд засобів безпеки у хмарному середовищі

Ефективне управління інформаційною безпекою в хмарних обчисленнях не обмежується лише впровадженням спеціалізованих технологій. Важливо враховувати реальні обставини експлуатації, специфіку оброблюваних даних, модель надання хмарних послуг (IaaS, PaaS, SaaS), а також ризик-профіль і відповідність нормативним вимогам. Технології захисту повинні не просто існувати в інфраструктурі, а бути інтегрованими у її архітектуру, логіку доступу та автоматизовані процеси. У цьому підпункті представлено функціональний аналіз та порівняльну оцінку провідних технологій забезпечення безпеки, які впроваджуються у середовищах Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP).

1.3.1. Системи керування ідентифікацією та доступом

Технології IAM (Identity and Access Management) забезпечують фундаментальну безпеку, відповідаючи за контроль над доступом користувачів, сервісів і процесів до хмарних ресурсів. У сучасних хмарних системах IAM виконує роль центрального механізму реалізації політик безпеки, забезпечуючи дотримання принципу найменших привілеїв (Least Privilege) та розмежування повноважень.

AWS IAM дозволяє створювати користувачів, групи, ролі та політики, що регулюють доступ на рівні API, сервісів або окремих об'єктів. Сервіс підтримує багатофакторну автентифікацію (MFA), тимчасові сесії доступу (STS), а також

інтеграцію з зовнішніми постачальниками ідентичностей. У Microsoft Azure рольову модель реалізовано через Entra ID, яка включає підтримку RBAC, Federation Services, умовного доступу (Conditional Access) і гібридної автентифікації. У GCP модель побудована на granular permissions із гнучкою ієрархією ресурсів (організація, папки, проєкти, ресурси).

Інтеграція IAM у хмарну інфраструктуру значно знижує ризик ескалації привілеїв, забезпечує контроль доступу на рівні сервісів і дозволяє забезпечити повний аудит активності користувачів. Ці засоби є особливо ефективними в середовищах з великою кількістю акаунтів, мікросервісною архітектурою, а також при дотриманні політик Zero Trust.

1.3.2. Шифрування даних та управління ключами

Технології шифрування критично важливі в контексті конфіденційності даних. Вони дозволяють захистити інформацію як у стані зберігання (encryption at rest), так і при передачі (encryption in transit). AWS KMS підтримує генерацію, зберігання, обертання ключів шифрування, логування запитів до ключів і можливість використання зовнішніх HSM (CloudHSM). Azure Key Vault та Google Cloud KMS мають схожі функціональні можливості, включаючи підтримку BYOK, CYOK, автоматичну ротацію, а також тісну інтеграцію з іншими сервісами.

Шифрування виконує не лише технічну, а й юридичну функцію, дозволяючи організаціям виконувати вимоги регуляторів (GDPR, PCI DSS, HIPAA). Водночас керування ключами є складною задачею в мультихмарних середовищах. Саме тому особливої уваги набуває централізоване управління ключами з підтримкою аудитів, розмежуванням доступу та валідацією конфігурацій.

1.3.3. Журналювання, моніторинг та аудит подій

Механізми журналювання та моніторингу дозволяють забезпечити прозорість усіх операцій у хмарній інфраструктурі. AWS CloudTrail, Azure

Monitor та Google Cloud Logging автоматично фіксують дії користувачів, зміну політик, доступ до об’єктів та API-запити. Ці дані зберігаються у відповідних сервісах, можуть передаватися у SIEM або використовуватись для тригерів безпеки (наприклад, у AWS Lambda) [18].

У складних середовищах наявність системи журналювання є обов’язковою умовою відповідності стандартам безпеки. Вона дозволяє виявити спроби несанкціонованого доступу, відстежити причини збоїв та отримати доказову базу під час розслідування інцидентів. Журналювання також є основою автоматизації: за певними подіями можуть запускатися політики, сканування або блокування доступу.

1.3.4. Централізовані платформи аналізу безпеки

У великих інфраструктурах ручний аналіз подій безпеки стає малоефективним. Саме тому використовуються інтегровані сервіси аналітики, що агрегують дані з кількох джерел (IAM, SIEM, Config, журнали подій), автоматично оцінюють ризики та надають узагальнені висновки. Security Hub (AWS), Defender for Cloud (Azure), Security Command Center (GCP) об’єднують у собі інструменти агрегації, нормалізації та візуалізації інформації про події.

Ці сервіси дозволяють значно скоротити час виявлення інцидентів, знизити навантаження на аналітиків SOC, автоматизувати реакцію на загрози. Вони також інтегруються з зовнішніми CMDB, SIEM, XDR-рішеннями та можуть бути частиною end-to-end процесів безпеки.

1.3.5. Перевірка конфігурацій та політик відповідності

Більшість вразливостей у хмарних середовищах виникають не через складність загроз, а через некоректну або ненадійну конфігурацію ресурсів. AWS Config, Azure Policy та Google Policy Intelligence дозволяють створювати шаблони налаштувань, контролювати їх відповідність, запускати коригувальні дії або генерувати сповіщення у разі виявлення порушень.

Доцільність впровадження таких систем зростає в умовах високої динаміки змін, наприклад, у DevOps-середовищах або при CI/CD-розгортаннях. Ці сервіси дозволяють автоматизувати контроль відповідності, формувати звіти для аудитів та знижувати ризик людських помилок.

Оскільки реалізація захисту в хмарному середовищі охоплює цілий комплекс сервісів — від управління доступом до контролю конфігурацій і аудитів — доцільно провести порівняльний аналіз основних інструментів безпеки. У таблиці 2.1 наведено ключові характеристики найбільш поширених засобів захисту.

Таблиця 2.1.

Порівняння засобів безпеки у хмарних середовищах

Засіб безпеки	Функціональне призначення	Ключові переваги	Недоліки або обмеження	Рекомендоване застосування
IAM (AWS, Azure, GCP)	Управління ідентифікацією, ролями, правами доступу, реалізація MFA, політик доступу	Гнучкість налаштувань, глибока інтеграція з іншими сервісами, контроль доступу	Висока складність у великих середовищах, ризик помилок конфігурації політик	Обов’язкове використання у будь-якому хмарному середовищі
KMS / Key Vault / Cloud KMS	Шифрування даних у стані спокою та при передачі, управління ключами, аудит доступу	Відповідність GDPR, PCI DSS; підтримка BYOK; прозоре управління ключами	Складність інтеграції в мультихмарних середовищах, залежність від типу ключа	Обробка чутливих даних, дотримання нормативних вимог
CloudTrail/ Monitor/ Logging	Фіксація всіх дій у середовищі, журналювання API-запитів, аудит і трасування	Масштабованість, простота інтеграції з SIEM, аудит активностей користувачів	Необхідність зберігання великого обсягу логів, обмежена аналітика без SIEM	Критично важливий для середовищ з високими вимогами до аудиту
Security Hub /Defender/ SCC	Централізований аналіз ризиків, агрегація подій безпеки, оцінка відповідності	Швидке виявлення відхилень, автоматизація аналізу, інтеграція з іншими сервісами	Висока вартість і потреба в налаштуванні, складність для невеликих проєктів	Великі організації, SOC, мультихмарні структури
AWS Config / Azure	Автоматична перевірка	Зниження ризику людської помилки,	Потреба в супроводі та	DevOps, CI/CD,

Policy / GCP PI	налаштувань ресурсів на відповідність політикам безпеки	формалізація стандартів безпеки, масштабованість	періодичному оновленні правил, залежність від архітектури	середовища з частими змінами конфігурацій

Формування ефективної стратегії захисту у хмарному середовищі вимагає використання широкого спектра засобів безпеки, що працюють на різних рівнях — від автентифікації до автоматизованого реагування. Практика показує, що найбільш ефективними є саме комбіновані підходи, які охоплюють усі аспекти інфраструктури: контроль доступу, захист даних, аудит активностей, автоматизований аналіз подій і перевірку відповідності політикам. Вибір конкретних інструментів має ґрунтуватися на оцінці ризиків, масштабі середовища, рівні критичності оброблюваної інформації та вимогах до відповідності стандартам.

1.4. Побудова моделі безпеки на базі сервісів AWS

У контексті забезпечення інформаційної безпеки в хмарних обчислювальних середовищах постає потреба у створенні уніфікованих підходів, які б дозволяли не лише локалізовано впроваджувати окремі сервіси захисту, а й інтегрувати їх у цілісну функціональну архітектуру. З огляду на високий ступінь розподіленості хмарних платформ, відсутність класичних периметрів і швидкоплинність конфігурацій, ефективне управління безпекою вимагає моделювання внутрішніх взаємозв’язків між компонентами захисту, орієнтованих на динамічну зміну контексту.

З урахуванням особливостей екосистеми Amazon Web Services — однієї з найпоширеніших хмарних платформ — доцільним є формування моделі, яка б відображала як структурну організацію засобів безпеки, так і логіку їхньої взаємодії. Така модель має виконувати роль опорної конструкції при впровадженні базових принципів захисту (ізоляції, контролю доступу, спостереження, відповідності політикам), забезпечуючи водночас адаптивність до зміни архітектурного оточення.

Запропонована нижче модель безпеки не є жорстко прив’язаною до конкретної конфігурації або типу сервісу, а розглядається як логічна абстракція, на основі якої може бути реалізовано повноцінну систему захисту в межах будь-якої хмарної інфраструктури, що функціонує на базі AWS. Її побудова базується на функціональному розподілі компонентів за рівнями відповідальності та на визначенні типових сценаріїв взаємодії між сервісами безпеки.

1.4.1. Принципи побудови моделі безпеки у хмарному середовищі

Побудова ефективної моделі безпеки у хмарній інфраструктурі передбачає не лише впровадження окремих технічних засобів, а й формування системної логіки, що охоплює всі рівні управління доступом, контролю над даними та відповідності вимогам інформаційної безпеки. У зв’язку з цим доцільним є виділення ключових принципів, які визначають засадничі підходи до організації захисту у середовищах з розподіленою відповідальністю.

Серед основних принципів, які було покладено в основу побудови моделі безпеки на базі сервісів AWS, можна виокремити наступні:

- Принцип мінімальних привілеїв**:** кожному суб’єкту надаються виключно ті права доступу, які є необхідними для виконання функціональних обов’язків у межах конкретного сценарію використання ресурсів. Реалізація цього принципу мінімізує ймовірність ескалації привілеїв у разі компрометації облікового запису.
- Zero Trust Architecture: в основі даного підходу лежить передумова відсутності автоматично довірених зон чи користувачів. Кожен запит до ресурсу має перевірятися незалежно від походження, що знижує ризики, пов’язані з внутрішніми загрозами або міжорендними атаками.
- Модульність і масштабованість**:** модель має бути такою, що допускає розширення без порушення загальної логіки безпеки. Нові сервіси, облікові записи або конфігурації повинні безперешкодно інтегруватися у вже існуючу архітектуру захисту.

- Безперервне моніторинг та адаптація: середовище, в якому функціонують хмарні сервіси, є динамічним за своєю природою, внаслідок чого безпекові механізми мають не лише фіксувати стан на певний момент часу, а й реагувати на зміни в реальному режимі. Особливу роль у цьому відіграє системне журналювання дій користувачів та аналіз конфігурацій.
- Відповідність нормативним вимогам**:** модель повинна дозволяти перевірку та демонстрацію відповідності міжнародним стандартам безпеки інформації (ISO/IEC 27001, 27017), галузевим регламентам (GDPR, HIPAA) та національному законодавству.

Ці принципи, покладені в основу побудови моделі безпеки, не лише забезпечують методологічну цілісність підходу, а й створюють передумови для її масштабованої реалізації в межах різнорівневих хмарних архітектур. Їх практичне втілення на платформі AWS дозволяє перейти від фрагментарного застосування сервісів до впровадження логічно завершеної системи інформаційної безпеки з підтримкою автоматичного контролю, відповідності та реагування.

1.4.2. Архітектура моделі безпеки в AWS

Після визначення базових принципів, на яких ґрунтується модель забезпечення інформаційної безпеки в хмарному середовищі, доцільно перейти до розгляду її архітектурної реалізації. У межах платформи Amazon Web Services можна побудувати логічно зв’язану систему захисту, яка поєднує різні типи сервісів — від контролю доступу до автоматизованого реагування на події — у межах єдиної структурованої моделі.

На рисунку 2.3 зображено узагальнену архітектуру моделі безпеки, в якій кожен блок виконує чітко визначену функцію. Структура побудована таким чином, щоб забезпечити наскрізний контроль: від ідентифікації користувачів і реєстрації їхніх дій — до аналізу ризиків та ініціювання відповідних захисних заходів.

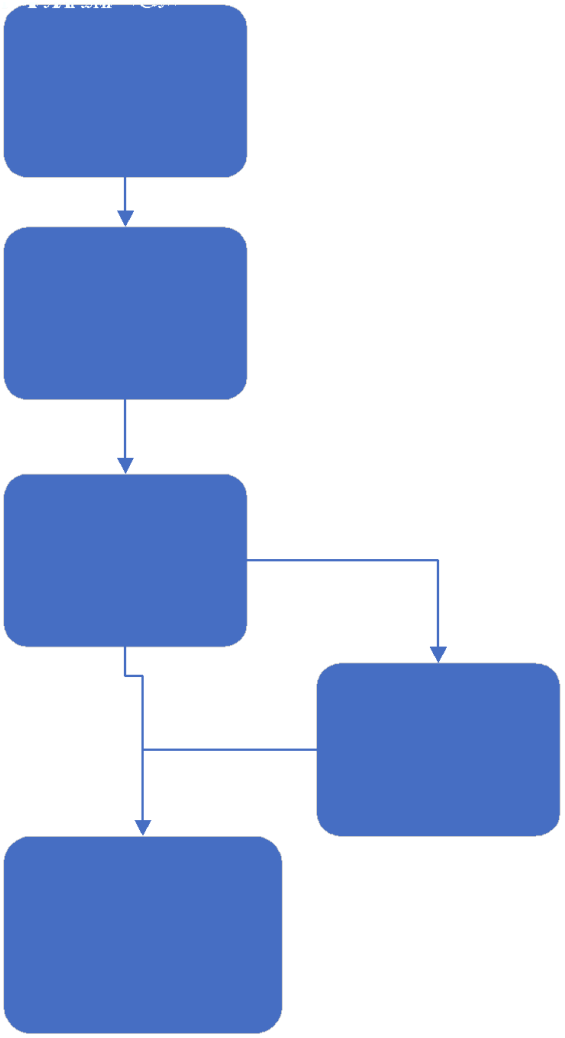

Рисунок 2.3 – Архітектура моделі безпеки в AWS

У структурному відношенні модель складається з п’яти взаємодоповнюваних блоків:

1. Блок автентифікації та авторизації — реалізований за допомогою механізмів AWS IAM (Identity and Access Management) [14], що забезпечують контроль над суб’єктами доступу та їхніми правами. Включає політики на рівні ролей, груп і сервісів, а також підтримку багатофакторної автентифікації.

2. Блок реєстрації дій користувачів і сервісів — відповідає за логування подій на рівні API-викликів та змін інфраструктури. Реалізується через AWS CloudTrail, що забезпечує незалежність записів та можливість наступного аудиту.

3. Блок контролю конфігураційної відповідності — охоплює засоби AWS Config, які дозволяють порівнювати поточний стан ресурсів із політиками, визначеними адміністратором, і виявляти порушення у режимі наближеному до реального часу.

4. Блок шифрування та керування ключами — представлений AWS Key Management Service (KMS), що відповідає за створення, зберігання, ротацію та контроль доступу до криптографічних ключів, необхідних для захисту даних у S3, RDS, EBS та інших сервісах.

5. Блок моніторингу, кореляції подій та реагування — об’єднує Security Hub (централізований збирач подій безпеки), CloudWatch (систему метрик та тригерів), а також сервісні механізми реагування, реалізовані через SNS, Lambda або EventBridge, через які реалізується автоматизована відповідь на інциденти.

Ця модель не є жорстко закріпленою під конкретну архітектуру або сервіс AWS, однак завдяки модульності та узгодженості компонентів може бути використана як універсальний шаблон побудови захисту у хмарних середовищах із різним рівнем складності.

1.4.3. Логіка взаємодії сервісів у моделі

Побудова функціональної моделі безпеки в хмарному середовищі має сенс лише за умови, коли всі її складові не лише існують паралельно, а й працюють у тісному взаємозв’язку. В Amazon Web Services така взаємодія реалізується через стандартизовані канали обміну подіями між сервісами, що дозволяє досягати цілісності процесу контролю — від автентифікації до автоматизованої реакції на порушення.

На рівні логіки модель передбачає послідовну обробку подій безпеки за такою схемою:

1. Користувач або сервіс ініціює дію (наприклад, створення ресурсу, зміну конфігурації або доступ до об’єкта зберігання).

2. IAM здійснює перевірку прав доступу та застосовує відповідну політику авторизації.

3. CloudTrail фіксує подію у журналі, створюючи запис для подальшого аналізу.

4. AWS Config негайно перевіряє, чи відповідає новий стан ресурсу встановленим правилам, та фіксує відхилення у разі порушення.

5. Security Hub збирає дані з CloudTrail, Config, IAM Access Analyzer, GuardDuty та інших джерел, здійснюючи кореляцію та класифікацію подій за критеріями критичності [10, 15].

6. У разі перевищення визначених порогів CloudWatch створює алерт, який може передаватися у SNS, Lambda або EventBridge, ініціюючи сповіщення чи запуск автоматичної дії у відповідь (наприклад, блокування доступу або переведення ресурсу у захищений режим) [18, 19].

Узгоджена взаємодія цих сервісів створює замкнутий цикл реагування на інциденти без необхідності постійного ручного втручання. Такий підхід дозволяє знизити час виявлення та усунення порушень, а також забезпечує відповідність вимогам стандартів щодо ретроспективного аудиту та звітності.

Зокрема, варто звернути увагу на можливість інтеграції Security Hub з іншими інструментами оцінки відповідності, що відкриває додаткові можливості для аналізу політик безпеки, автоматизованої верифікації конфігурацій, а також імплементації процедур безперервного вдосконалення безпекової моделі.

Таким чином, взаємодія між сервісами у межах моделі відбувається не ізольовано, а за принципом логічного спадкування та перевірки подій на кожному етапі, що дозволяє забезпечити комплексний підхід до управління інформаційною безпекою у хмарному середовищі.

Висновок за розділом 2

У другому розділі проведено ґрунтовне дослідження сучасних технологій та інструментів забезпечення інформаційної безпеки у хмарному середовищі.

Описано ключові платформи, що домінують на ринку хмарних послуг — зокрема AWS, Microsoft Azure та Google Cloud Platform — та здійснено порівняльний аналіз їхніх вбудованих засобів захисту. Розглянуто найважливіші напрямки побудови системи безпеки, такі як управління ідентифікацією та доступом (IAM), шифрування даних та керування ключами (KMS), аудит подій, моніторинг (CloudTrail, CloudWatch), а також контроль конфігурацій і дотримання політик відповідності.

На основі проведеного дослідження сформовано бачення того, як сучасні сервіси безпеки реалізуються на практиці в умовах хмарної інфраструктури. Особливу увагу зосереджено на архітектурі захисту в Amazon Web Services — описано принципи побудови моделі безпеки, її логічну структуру та взаємодію окремих компонентів у межах обраної архітектури [13]. Представлений підхід дозволяє враховувати особливості масштабованості, розподіленості та багаторівневого доступу, що є характерними для хмарних середовищ.

Отримані результати дають змогу не лише сформувати модель безпеки, але й перейти до її практичної реалізації. Це створює методичну і технічну основу для подальшого впровадження комплексу захисних заходів.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЗАСОБІВ ЗАХИСТУ ХМАРНОГО СЕРЕДОВИЩА НА БАЗІ AMAZON WEB SERVICES

2.1. Створення захищеного середовища на базі AWS

Першим етапом у реалізації безпечної хмарної інфраструктури стало налаштування облікового середовища в AWS. Йдеться не просто про створення акаунта, а про впровадження таких початкових дій, які дозволяють мінімізувати ризики вже на стартовому рівні: захист від несанкціонованого доступу, унеможливлення випадкового витоку даних, ізоляція критичних ресурсів.

Загальна логіка цього етапу полягає в тому, щоб створити захищене середовище, яке потім використовуватиметься як основа для побудови ширшої інфраструктури: з EC2-ресурсами, моніторингом, аудитом і контрольованим доступом.

2.1.1. Реєстрація облікового запису в AWS

Реєстрація акаунта в AWS — процедура доволі проста. Я створив обліковий запис на aws.amazon.com, використавши особисту електронну пошту, зазначивши себе як індивідуального користувача, й підтвердивши номер телефону. На цьому етапі також потрібно ввести дані банківської картки, але гроші не списуються — якщо дотримуватися меж безкоштовного плану (Free Tier).

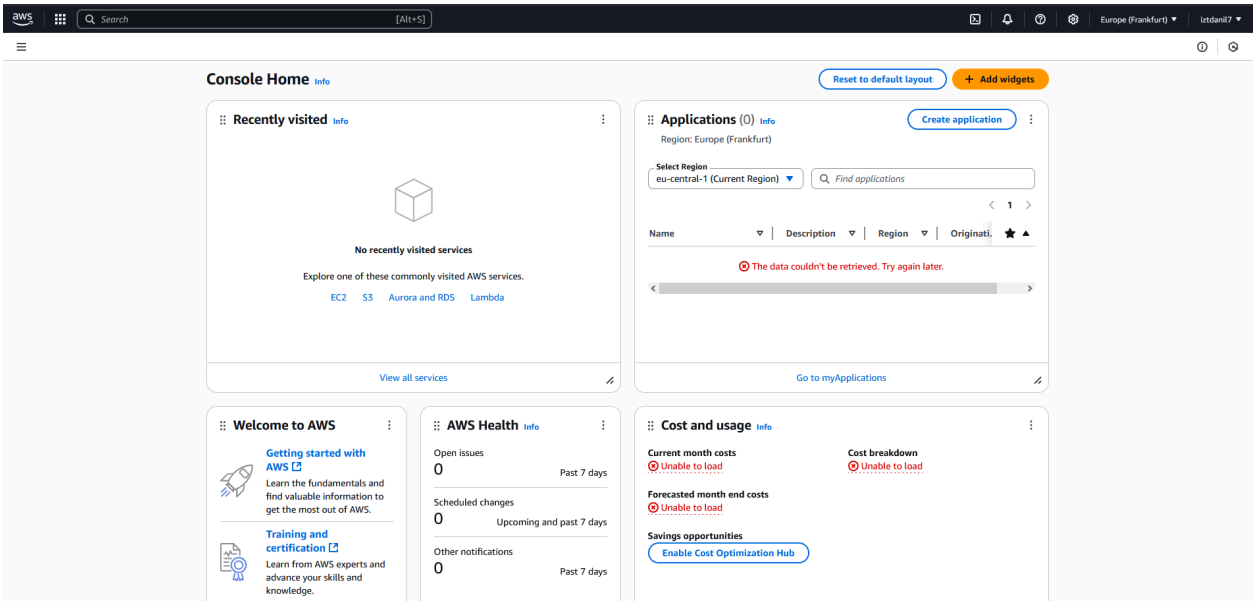

Рисунок 3.1 – Головна панель керування AWS Management Console

Одразу після реєстрації я виконав рекомендацію AWS — увімкнув багатофакторну автентифікацію (MFA) для root-користувача. Це захищає акаунт у разі, якщо хтось дізнається пароль. MFA реалізується за допомогою мобільного застосунку (я використав Google Authenticator). Процедура доволі проста: у розділі IAM → Security Recommendations потрібно обрати Activate MFA, просканувати QR-код і підтвердити два коди з мобільного додатка.

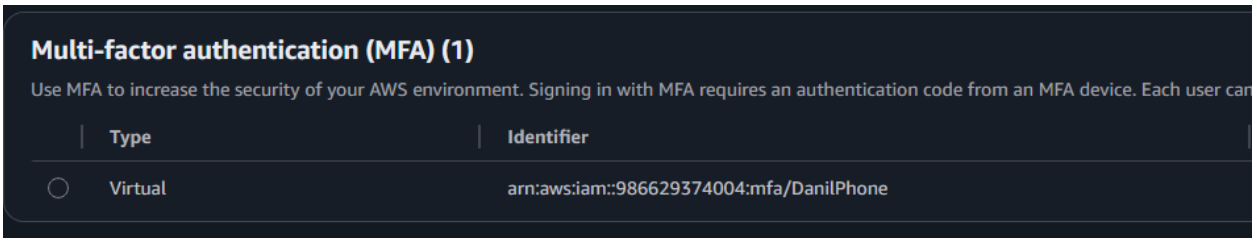

Рисунок 3.2 – Пристрій для MFA автентифікації

У результаті налаштування MFA обидва облікові записи — root та адміністратор — отримали додатковий рівень захисту. Кожна спроба входу тепер вимагає не лише пароля, а й одноразового коду з мобільного пристрою, що суттєво зменшує ризик несанкціонованого доступу.

2.1.2. Створення адміністративного користувача IAM

Після активації облікового запису AWS та налаштування базового захисту для root-користувача, було прийнято рішення не використовувати цей обліковий запис для щоденних операцій. Це відповідає загальноприйнятим практикам у сфері кібербезпеки: обліковий запис із максимальними правами доступу має бути зарезервований лише для аварійних або критичних ситуацій. Натомість повсякденне керування інфраструктурою доцільно здійснювати через окремого адміністративного користувача з контрольованим рівнем доступу.

Для цього було використано сервіс IAM (Identity and Access Management) — централізований інструмент для створення та управління користувачами, групами, ролями і політиками доступу в хмарному середовищі AWS. Основною перевагою IAM є гнучкість у налаштуванні прав: можна чітко визначити, до яких ресурсів матиме доступ кожен користувач, і які дії він зможе виконувати.

Процедура створення адміністратора охоплює кілька етапів:

1. У розділі IAM → Users було ініційовано створення нового користувача, якому надано ім’я admin-secure.
2. При створенні активовано доступ до AWS Management Console [22], а також автоматично згенеровано пароль.
3. У процесі призначення дозволів користувача було додано до системної групи з політикою AdministratorAccess. Це дозволяє управляти всіма ресурсами AWS без обмежень — подібно до root, але з можливістю гнучкого журналювання й подальшого обмеження при потребі.

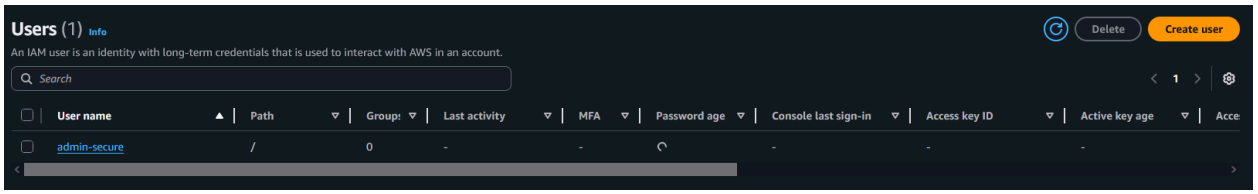

Рисунок 3.3 – Список користувачів у AWS IAM

На цьому етапі в хмарному середовищі було створено незалежного адміністратора з персоніфікованим доступом, захищеним двофакторною

автентифікацією. Усі подальші дії в межах практичної частини виконуватимуться саме від імені цього користувача, що відповідає принципам побудови надійного середовища адміністрування в AWS.

2.1.3. Налаштування об’єктного сховища Amazon S3

Одним із ключових компонентів інфраструктури AWS є Amazon S3 (Simple Storage Service) [20] — об’єктне сховище, призначене для зберігання файлів, журналів, резервних копій, конфігурацій тощо. В умовах побудови безпечного хмарного середовища важливо не просто створити сховище, а забезпечити його ізоляцію від публічного доступу, а також можливість контролювати права читання і запису**.**

У реальній практиці саме неправильна конфігурація S3-бакетів є частою причиною витоку конфіденційної інформації. Тому цей етап реалізації зосереджено на грамотному налаштуванні доступу й активуванні механізмів відстеження змін.

На рисунку 3.4 показано, що бакет secure-storage-lab7 успішно створено та зареєстровано в регіоні eu-central-1. Цей бакет було сконфігуровано як приватний, із повним блокуванням публічного доступу, увімкненим версіонуванням і шифруванням об’єктів. Надалі дане сховище використовуватиметься для зберігання конфігураційних файлів, логів та інших службових об’єктів у рамках хмарної інфраструктури.

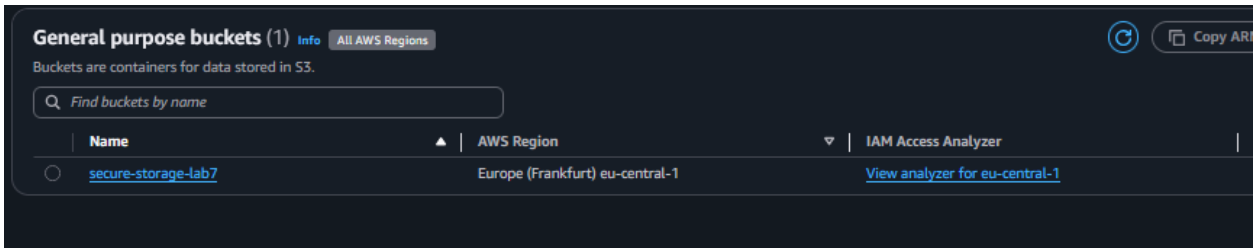

Рисунок 3.4 – Створений бакет Amazon S3

Щоб надати користувачу admin-secure доступ саме до мого бакету, я створив окрему політику з нуля, у режимі JSON. Це дало змогу максимально точно вказати, які саме дії дозволені, і до якого ресурсу.

Як видно на рисунку 3.5, у першій частині політики я прописав дозвіл на перегляд списку об’єктів (s3:ListBucket), а в другій — дозволив читати, записувати й видаляти файли (s3:GetObject, s3:PutObject, s3:DeleteObject) лише в межах одного бакету secure-storage-lab7.

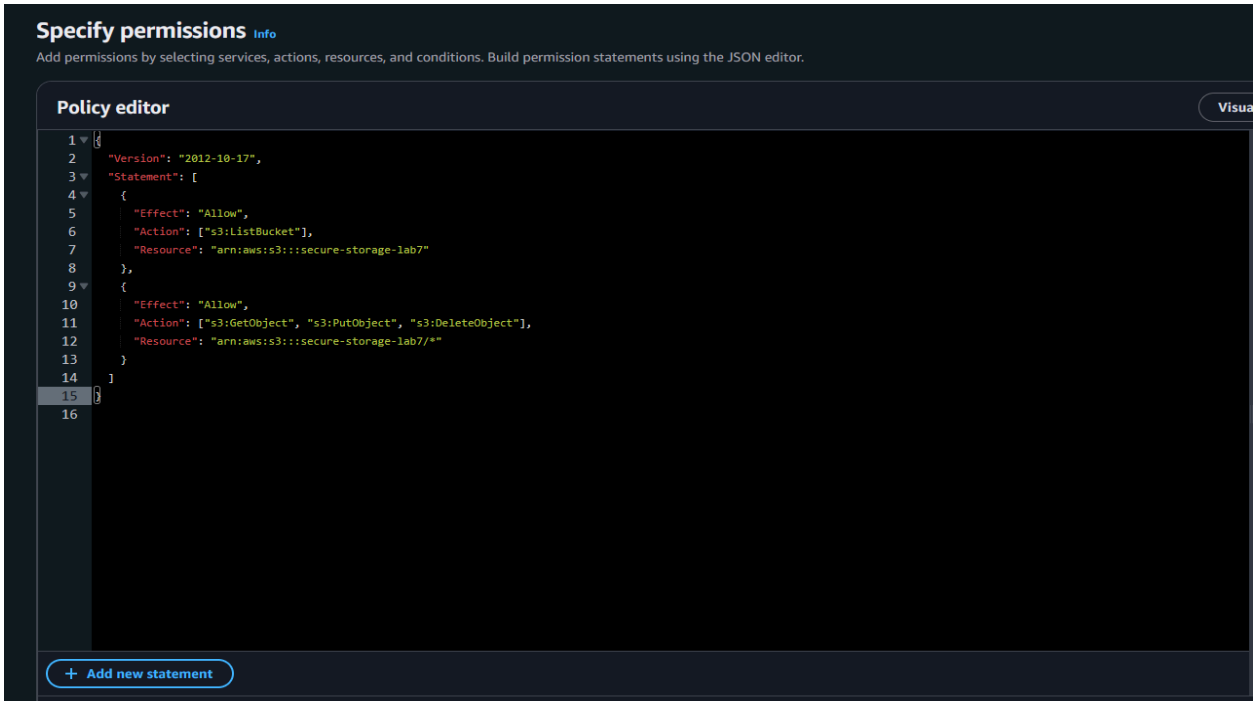

Рисунок 3.5 – Створення IAM-політики з доступом до бакету

Політика чітко вказує ARN ресурсу, що важливо — інакше політика може бути надто загальною. Таким чином, я обмежив доступ лише до одного конкретного бакету, не надаючи прав до всіх інших S3 у моєму акаунті.

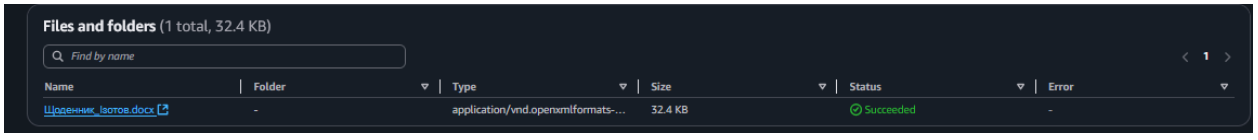

Рисунок 3.6 – Завантажений тестовий файл у сховище

Після застосування цієї політики я виконав тест: увійшов під admin-secure і зміг завантажити тестовий файл у сховище. Це підтвердило, що політика працює як треба.

2.2. Побудова ізольованої хмарної інфраструктури

Після налаштування користувача, MFA та доступу до сховища, настав момент перейти до створення базової інфраструктури, в якій усе це буде працювати. Я вирішив побудувати просту, але логічно ізольовану мережу, в межах якої можна буде запускати сервери, надавати їм доступ до Інтернету і водночас мати контроль над тим, хто й куди має доступ.

2.2.1. Створення VPC — власної віртуальної мережі

Розгортання будь-якої хмарної інфраструктури в AWS логічно починати зі створення окремої віртуальної мережі, відомої як VPC (Virtual Private Cloud) [21]. Це дозволяє повністю контролювати адресний простір, маршрути, шлюзи доступу та правила безпеки в межах хмарного середовища.

Хоча AWS автоматично створює одну дефолтну VPC у кожному регіоні, я вирішив не користуватись нею. Для кваліфікаційної роботи важливо було показати повне самостійне налаштування ізольованої мережі з нуля, з чітким розумінням її архітектури та функціонального призначення.

На цьому етапі я створив власну віртуальну хмару з назвою secure-lab-vpc, яка є повністю ізольованою від інших мереж. Як видно з рисунка 3.7, для цієї VPC я задав CIDR-блок 10.0.0.0/16, що дозволяє гнучко формувати підмережі в межах цієї адресної області.

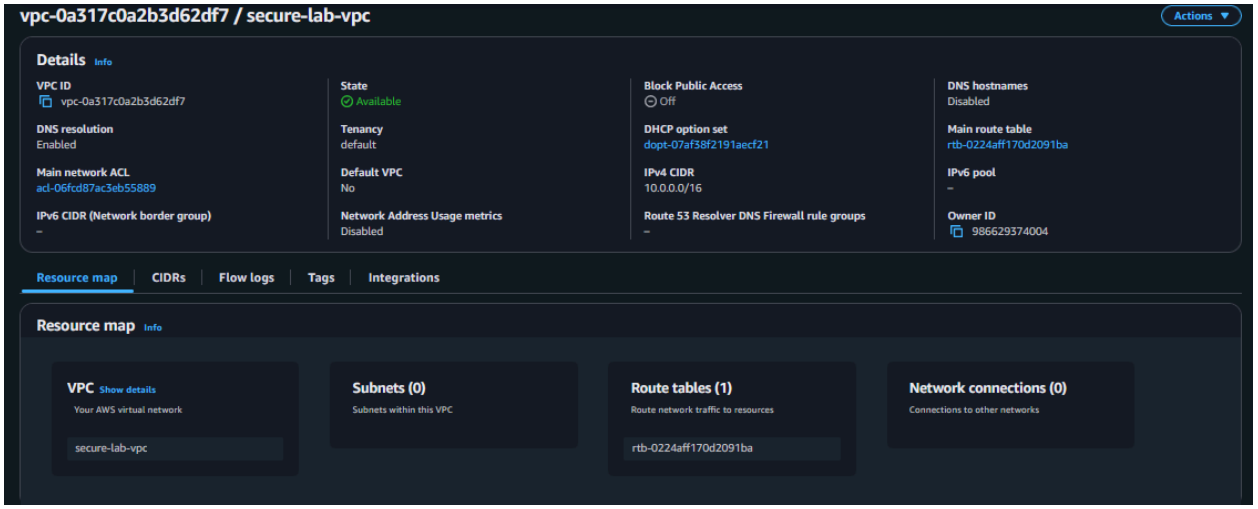

Рисунок 3.7 – Створена віртуальна мережа з параметрами доступу та мережею в AWS VPC

Поки що в цій VPC не налаштовано жодної підмережі, а також відсутні підключення до інтернету. Я спеціально обрав опцію "VPC only", щоб усе налаштувати вручну — від маршрутизації до шлюзів.

DNS-підтримка для цієї мережі також поки що вимкнена, як і автоматичне призначення DNS-імен — я це зроблю пізніше, коли знадобиться доступ до EC2.

У Resource Map можна побачити, що наразі в цій мережі немає підмереж, підключень або спеціальних правил маршрутизації. Це зручно — починаєш з “чистого аркуша”, і далі сам вирішуєш, що й куди має доступ.

2.2.2. Додавання публічної підмережі та інтернет-шлюзу

Після створення власної віртуальної хмари (VPC), наступним кроком було формування публічної інфраструктури — тобто створення підмережі, яка дозволяє хостам (наприклад, EC2-інстансам) виходити в Інтернет. Це критично важливо для оновлень, завантаження пакетів, підключення через SSH та реалізації базової взаємодії з зовнішнім світом.

У межах створеної VPC (secure-lab-vpc) я створив нову підмережу з назвою public-subnet-1. Її адресний простір було визначено як 10.0.1.0/24 — це частина CIDR-блоку моєї мережі 10.0.0.0/16. Така підмережа дає можливість

розмістити до 256 адрес (фактично трохи менше через службові IP), чого більш ніж достатньо для мого навчального середовища.

Після створення підмережі public-subnet-1 у межах моєї VPC secure-lab-vpc, я увімкнув параметр "Enable auto-assign public IPv4 address". Це дозволяє EC2-інстансам, які будуть запускатись у цій підмережі, автоматично отримувати публічну IP-адресу без додаткових ручних налаштувань.

На рисунку 3.8 видно, що після відкриття налаштувань підмережі й редагування IP-параметрів, було активовано цю опцію. Це важливий крок, оскільки без нього навіть за наявності Internet Gateway EC2-інстанси не зможуть виходити в Інтернет або приймати з’єднання, якщо не матимуть зовнішньої адреси.

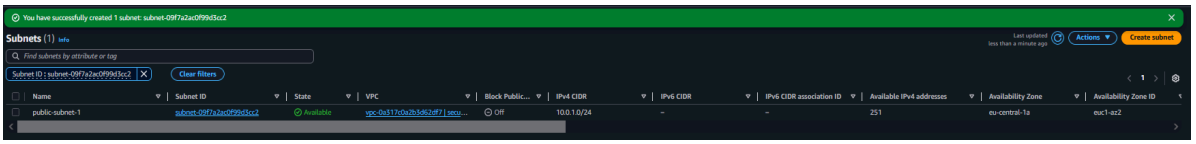

Рисунок 3.8 – Створена підмережа secure-lab-vpc

Інші параметри, зокрема DNS64 або RBN (Resource-based Name), залишено вимкненими, оскільки вони не є обов’язковими для базового лабораторного середовища.

На рисунку 3.9 представлено створений інтернет-шлюз (Internet Gateway) з ідентифікатором igw-0a540a..., який було названо secure-igw. Цей шлюз було успішно підключено до раніше створеної віртуальної мережі secure-lab-vpc, про що свідчить стан Attached.

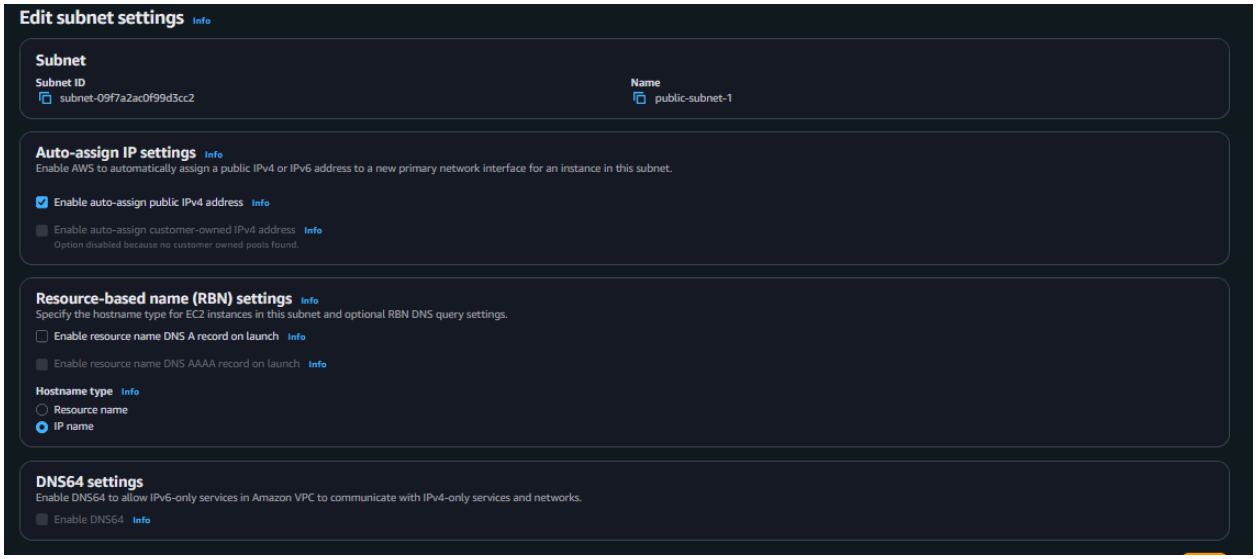

Рисунок 3.9 – Увімкнення автоматичної видачі публічної IPv4-адреси для підмережі

Internet Gateway виступає критичним компонентом інфраструктури, оскільки саме він забезпечує зовнішній інтернет-доступ для ресурсів усередині VPC. Після його створення та приєднання до мережі я зміг використовувати його як цільовий маршрут для виходу за межі приватного діапазону IP-адрес (0.0.0.0/0), що налаштовується в таблиці маршрутизації.

Цей крок став основою для забезпечення повноцінного функціонування EC2-інстансів з доступом до зовнішніх ресурсів — наприклад, для оновлення системи або встановлення програм через менеджер пакетів.

На наступному етапі я створив окрему маршрутну таблицю з назвою public-rtb, призначену для використання у публічних підмережах. Як видно з рисунка 3.10, у таблицю було додано маршрут типу 0.0.0.0/0, що відповідає усім зовнішнім адресам. Цей маршрут спрямовується через інтернет-шлюз secure-igw, який я створив та приєднав до VPC раніше.

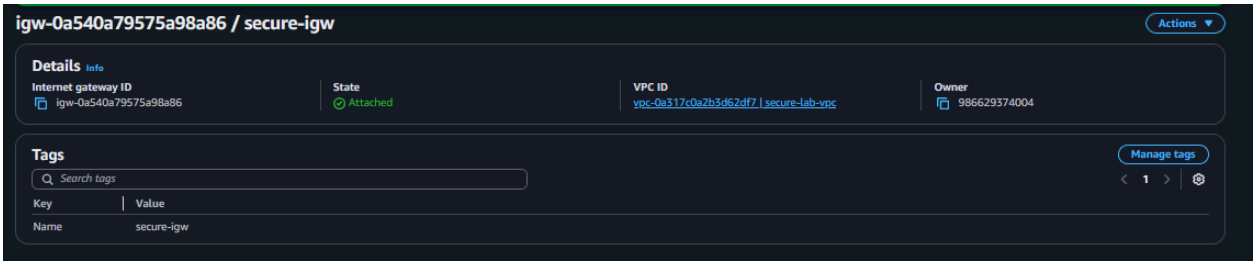

Рисунок 3.10 – Створений та підключений інтернет-шлюз

Такий запис дає змогу інстансам, розміщеним у підмережі, яка прив'язана до цієї таблиці, виходити в Інтернет. Крім того, таблиця містить стандартний маршрут 10.0.0.0/16 → local**,** що забезпечує зв'язок між усіма ресурсами в межах VPC.

Після створення таблиці я також виконав її асоціацію з публічною підмережею public-subnet-1, щоб маршрути були активні саме для тих ресурсів, які будуть у ній розміщуватись. Без цього кроку навіть за наявності IGW інстанси залишалися б ізольованими.

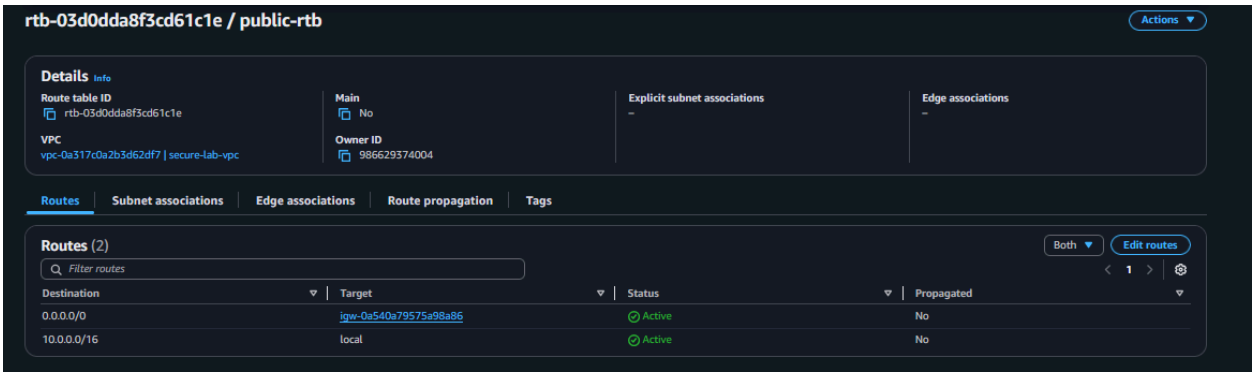

Рисунок 3.11 – Таблиця маршрутизації public-rtb з маршрутом до Інтернету через шлюз

Після створення маршрутної таблиці я перейшов до вкладки Subnet associations та виконав прив'язку підмережі public-subnet-1. Це гарантує, що всі ресурси в цій підмережі зможуть використовувати маршрути, зокрема вихід в Інтернет через IGW. Без цього маршрути залишаються неактивними для обраної зони.

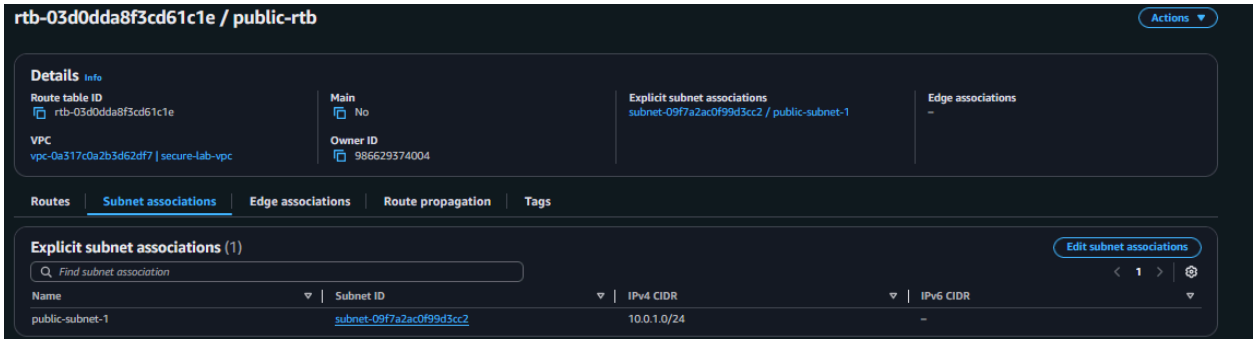

Рисунок 3.12 – Прив’язка підмережі public-subnet-1 до таблиці маршрутизації public-rtb

Цей етап завершив побудову мережевої частини інфраструктури. Після нього я можу з впевненістю переходити до створення EC2-інстансу — оскільки знаю, що він отримає публічну IP-адресу і матиме вихід в Інтернет згідно з заданими правилами.

2.2.3. Створення EC2-інстансу з публічною IP-адресою

Після завершення налаштування віртуальної мережі, підмережі та інтернет-шлюзу, наступним кроком стало розгортання першого віртуального сервера в середовищі AWS. У ролі такого сервера виступає EC2-інстанс (Elastic Compute Cloud) — базова обчислювальна одиниця в AWS, що дозволяє створити віртуальну машину з гнучкими параметрами.

На рисунку 3.13 зображено фінальний етап створення EC2-інстансу у хмарному середовищі AWS. Після завершення налаштувань, що включали вибір типу інстансу, образу операційної системи, мережевих параметрів та ключа доступу, система вивела повідомлення "Successfully initiated launch of instance", що свідчить про коректну ініціалізацію процесу запуску віртуальної машини.

У нижній частині інтерфейсу відображено блок "Next Steps", який пропонує рекомендовані дії після запуску інстансу. Серед них: підключення до інстансу через SSH, створення політик для знімків (EBS snapshot), увімкнення детального моніторингу (CloudWatch), керування буджетами AWS та навіть налаштування сповіщень про підозрілу активність.

Я скористався кнопкою "Connect to instance", яка дозволяє отримати точну команду для підключення через термінал (SSH) до EC2-інстансу, з урахуванням шляху до згенерованого ключа (.pem). Після цього перевірив доступність ресурсу, а також підтвердив, що він отримав публічну IP-адресу, як і було заплановано під час конфігурації мережевих налаштувань.

Цей етап завершив базове розгортання серверної частини в межах VPC та підтвердив, що вся попередньо створена інфраструктура (VPC, підмережа, IGW, маршрутна таблиця) працює коректно.

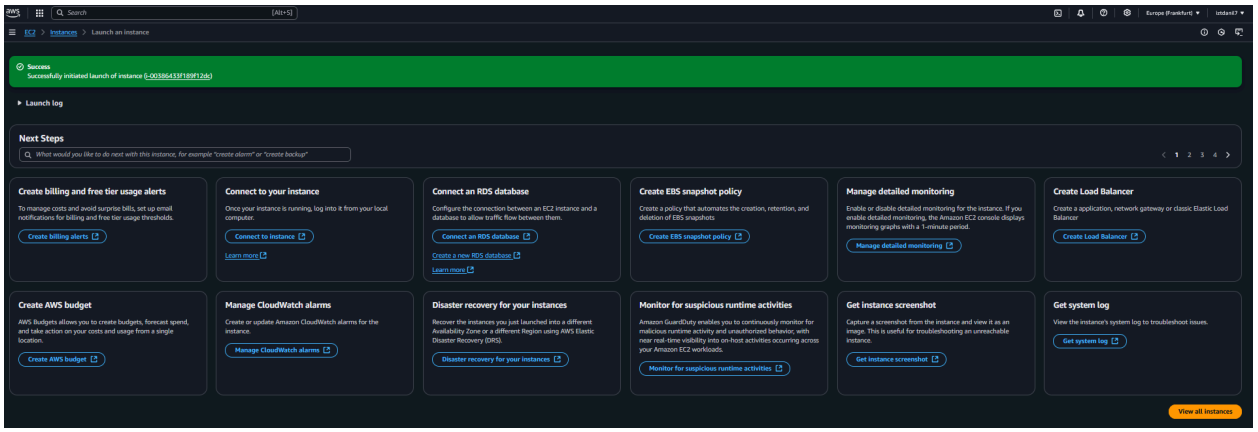

Рисунок 3.13 – Повідомлення про успішний запуск EC2-інстансу в AWS та перелік рекомендованих наступних дій

На рисунку 3.14 відображено результат встановлення віддаленого з’єднання з розгорнутим EC2-інстансом через захищений протокол SSH. Підключення здійснювалося з локального середовища PowerShell за допомогою команди:

ssh -i "izotov.pem" ec2-user@35.159.172.148

де izotov.pem — приватний ключ, згенерований при створенні інстансу, а IP-адреса була автоматично призначена службою AWS відповідно до увімкненого параметра Auto-assign public IP.

Успішне SSH-підключення вказує на те, що всі компоненти інфраструктури, налаштовані в межах попередніх етапів — VPC [21], публічна підмережа, маршрутна таблиця та Internet Gateway — функціонують коректно.

Також це свідчить про правильність конфігурації групи безпеки, ключа доступу та прив’язки IP-адреси.

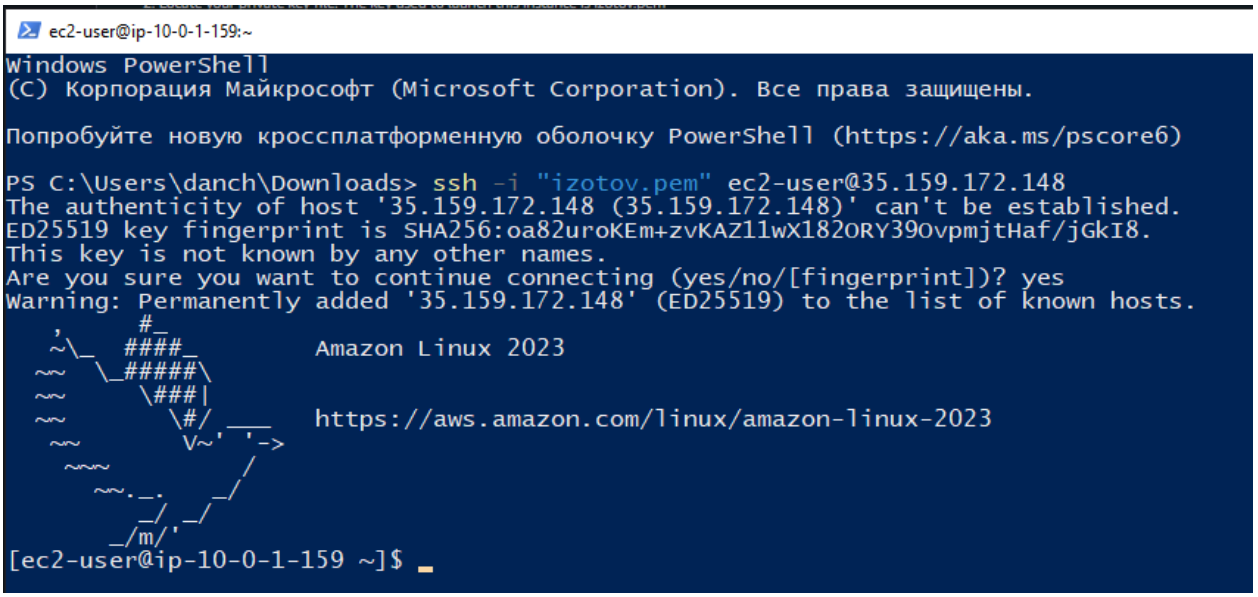

Рисунок 3.14 – Підключення до інстансу через SSH

Перший етап побудови хмарної інфраструктури було завершено. Розгорнутий віртуальний сервер успішно функціонує у межах ізольованого середовища, має зовнішню IP-адресу, захищений доступ через ключову автентифікацію та забезпечений мінімально необхідними мережевими маршрутами.

Наступним кроком у контексті забезпечення безпеки середовища є організація централізованого журналювання дій користувачів та сервісів у межах облікового запису AWS. Це дозволить не лише фіксувати критичні події, пов’язані з управлінням ресурсами, але й у перспективі — реалізувати автоматизоване сповіщення та виявлення інцидентів.

2.3. Підлючення журналу дій користувачів за допомогою CloudTrail

Забезпечення ефективного аудиту активності в хмарній інфраструктурі є ключовим елементом комплексної безпеки. Саме тому наступним етапом реалізації стало впровадження механізму журналювання подій**,** який дозволяє відстежувати всі дії, виконані як вручну через консоль, так і програмно за

допомогою API-запитів. У межах сервісів AWS цю функціональність забезпечує CloudTrail — вбудований інструмент для автоматичного збору та зберігання логів користувацької та сервісної активності.

На цьому етапі я переконався, що створений мною журнал подій secure-lab-trail працює коректно — про це свідчить статус Logging**:** Enabled, який відображається у верхній частині вікна (Рисунок 3.15). Також зазначено, що останній файл логів було доставлено 4 червня 2025 року, тобто журналювання дій відбувається в реальному часі.

В якості сховища для логів я вказав окремий S3-бакет, а для оперативного перегляду активності додатково налаштував інтеграцію з CloudWatch Logs — створив лог-групу secure-cloudtrail-group. Для цього система автоматично згенерувала IAM-роль із обмеженим доступом, яка дозволяє CloudTrail передавати туди події [15].

Журнал охоплює два основні типи подій: administrative (Management events) — наприклад, запуск або зупинка інстансу, зміна налаштувань безпеки, і data events — усі дії, пов’язані з доступом до об'єктів у S3. Це дозволяє не лише бачити, що саме відбулося, але й контролювати хто, коли і з якого IP ініціював ту чи іншу дію.Я поки що не вмикав Insights events, оскільки для них потрібна окрема конфігурація та бюджет, але при потребі їх можна активувати додатково.

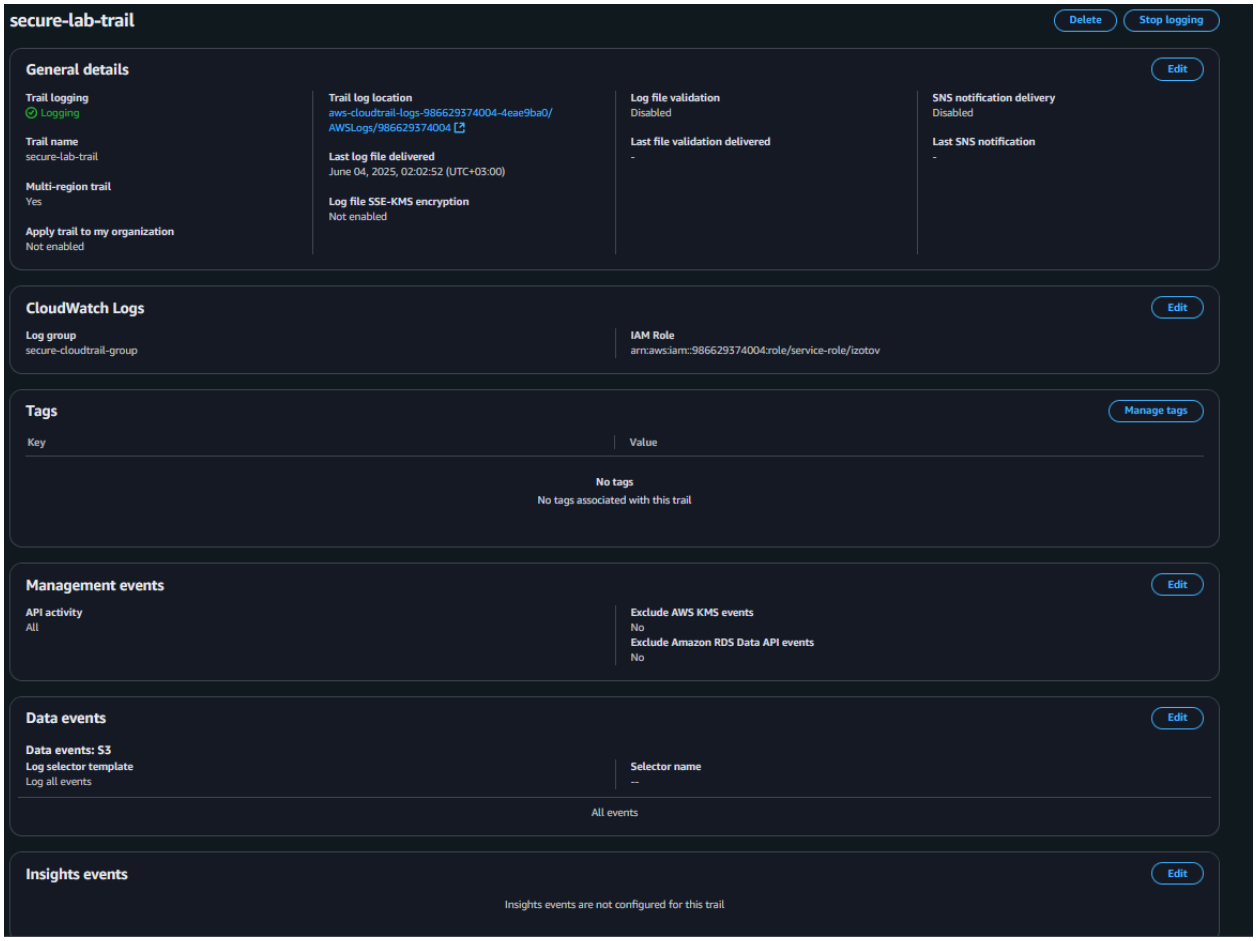

Рисунок 3.15 – Активне журналювання дій у secure-lab-trail та його параметри

Після завершення налаштуіання я вирішив перевірити, чи дійсно події фіксуються — для цього відкрив розділ Event history у CloudTrail. Як видно на рисунку 3.16, тут уже зберігається повний список усіх адміністративних дій, які я виконував у рамках розгортання середовища.

У списку можна побачити події на кшталт CreateTrail, StartLogging, CreateLogGroup, PutBucketPolicy, UpdateTrail — це дії, які я сам виконував у рамках попередніх етапів. Вони фіксуються разом із інформацією про джерело (сервіс), час виконання, користувача (root або CloudTrail), а також пов’язані ресурси.

Важливо, що всі події мають статус успішного виконання, а також зберігають точні параметри: наприклад, до якого S3-бакету було звернення або які права були змінені. Такий журнал дозволяє не лише бачити факт події, але й перевірити її безпеку, відновити логіку змін і при потребі провести повноцінний аудит.

Як на мене, цей інструмент значно полегшує контроль за тим, що відбувається всередині акаунту — не потрібно щоразу перевіряти кожен сервіс окремо, бо всі дії збираються централізовано. Особливо зручно те, що можна швидко фільтрувати події за сервісом, користувачем чи датою.

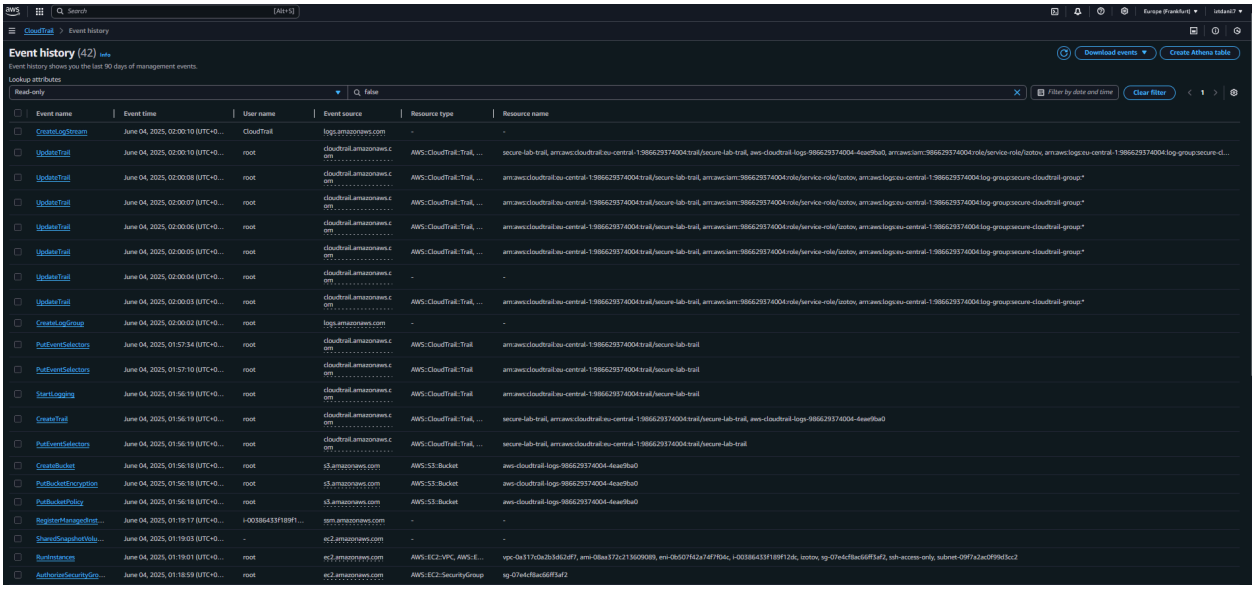

Рисунок 3.16 – Журнал зафіксованих подій у CloudTrail

Реалізація журналювання подій за допомогою AWS CloudTrail дозволила забезпечити повну історію дій у межах хмарного середовища, включно з фіксацією адміністративних запитів і доступу до ресурсів. Проте для досягнення комплексної картини стану інформаційної безпеки цього недостатньо: крім подій управління, необхідно контролювати поточні технічні параметри роботи сервісів, зокрема навантаження на віртуальні машини, використання ресурсів і аномальні відхилення.

2.3.1. Моніторинг стану EC2-інстансу за допомогою Amazon CloudWatch

Amazon CloudWatch — це сервіс спостереження та телеметрії, що дозволяє відстежувати системні метрики, логи та події від різних ресурсів AWS, включаючи EC2, RDS, Lambda, S3, а також сторонні сервіси через API. У межах цієї практичної частини було зосереджено увагу на базовому моніторингу EC2-інстансу, зокрема:

- рівень завантаження процесора (CPU utilization),
- мережевий трафік (Network In/Out),
- збереження логів життєвого циклу інстансу,
- побудова простої тривоги (alarm) при перевищенні порогового значення.

На рисунку 3.17 представлено інтерактивну панель моніторингу, створену за допомогою сервісу Amazon CloudWatch [19], яка відображає основні показники активності інстансу secure-lab-server. Графіки згенеровано автоматично на основі базового моніторингу, увімкненого за замовчуванням для EC2-ресурсів.

У верхньому лівому куті розташовано графік CPUUtilization, який демонструє рівень завантаження процесора протягом останніх трьох годин. Як видно, на момент спостереження навантаження не перевищувало 3%, що свідчить про відсутність інтенсивних обчислювальних операцій.

У середній частині екрана розміщено графіки, що відповідають за мережеву активність інстансу:

- NetworkIn, NetworkOut — кількість байтів, отриманих та відправлених;
- NetworkPacketsIn/Out — кількість пакетів, що пройшли через мережевий інтерфейс.

Усі ці метрики дозволяють виявляти сплески трафіку, спроби сканування, а також нетипове використання інстансу.

Також доступні дискові метрики: DiskReadBytes, DiskWriteBytes, DiskReadOps, DiskWriteOps, які відображають операції зчитування та запису. У даному випадку значення залишаються на нульовому рівні, що характерно для інстансу, який не має активного навантаження на зберігання.

Особливу увагу слід звернути на статуси перевірок системи (StatusCheckFailed, StatusCheckFailed_Instance, StatusCheckFailed_System). Всі вони мають значення 0, що свідчить про відсутність апаратних або логічних збоїв на рівні хосту чи гіпервізора.

Внизу екрана відображено деталі інстансу: назва, тип (t2.micro), статус (running), зона доступності (eu-central-1a) та статус моніторингу (disabled для деталізованого режиму).

Рисунок 3.17 – Візуалізація основних метрик EC2-інстансу у сервісі Amazon CloudWatch

Після запуску інстансу я спеціально спостерігав за поведінкою метрик у CloudWatch, щоб переконатися, що система реагує на активність. Навіть базового моніторингу достатньо, щоб оперативно виявляти проблеми. Наприклад, якщо CPU стрімко зростає — це може бути ознакою DoS-атаки, некоректного скрипта або підвищеного навантаження. Аналогічно, аналіз мережевого трафіку може вказати на спробу зламати SSH або витік даних.

На рисунку 3.18 зображено етап конфігурації дій тривоги в сервісі Amazon CloudWatch**,** яка активується за умови перевищення заданого порогового значення системної метрики. У полі Alarm state trigger вказано умову In alarm, що означає: дія буде виконана тоді, коли метрика (у даному випадку CPUUtilization) перевищить заданий поріг.

Як метод сповіщення було обрано використання сервісу SNS (Simple Notification Service)**.** Я створив окремий SNS-топік з назвою cpu-alarm-topic та вказав власну email-адресу izt.danil7@knu.ua як отримувача. Після створення

системою було надіслано підтвердження підписки на електронну пошту, яке потрібно було схвалити, щоб увімкнути доставку сповіщень.

У разі активації тривоги система автоматично надсилає лист на вказану адресу з деталями події. Такий підхід дозволяє оперативно реагувати на аномалії без постійного моніторингу вручну.

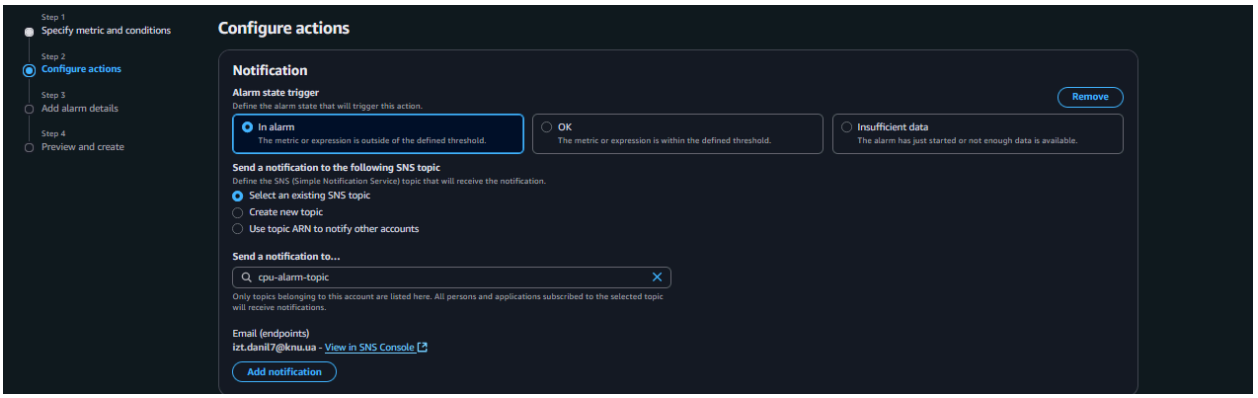

Рисунок 3.18 – Налаштування дії тривоги CloudWatch з використанням SNS-сповіщення

Після налаштування метрики CPU та її тривоги, я переконався, що AWS CloudWatch дозволяє досить швидко реалізувати базове оповіщення про перевищення навантаження. У рамках невеликого навчального середовища цього більш ніж достатньо — наприклад, при тестуванні або ручному навантаженні система відразу реагує. Але в промислових умовах цю логіку можна масштабувати: запускати Lambda-функції, збільшувати кількість інстансів через Auto Scaling або автоматично відправляти події в SIEM-систему.

2.4. Шифрування об'єктів за допомогою AWS Key Management Service

У сучасних умовах цифрової обробки даних питання збереження конфіденційності та цілісності інформації, що зберігається у хмарних сховищах, набуває особливої актуальності. Одним із найбільш поширених сценаріїв використання хмарної інфраструктури є зберігання файлів, логів, резервних копій або інших критичних даних у сервісі Amazon Simple Storage Service (S3) [20]. Саме тому на цьому етапі практичного дослідження було реалізовано

механізм шифрування об'єктів у S3-бакеті з використанням AWS Key Management Service (KMS).

Після завершення попередніх етапів налаштування S3-бакету, я перейшов до створення власного ключа шифрування, який буде використовуватись для автоматичного захисту даних при збереженні в хмарі. Для цього я скористався сервісом AWS Key Management Service (KMS).

На етапі конфігурації ключа я обрав тип — симетричний (Symmetric), оскільки саме цей тип підтримується серверним шифруванням в Amazon S3 (режим SSE-KMS). Ключ створено як одно-регіональний, тобто його можна використовувати тільки в обраному регіоні eu-central-1, що відповідає всій інфраструктурі, яку я розгортаю.

Для зручності в подальшому використанні я задав alias s3-storage-key — це умовна назва, під якою ключ буде відображатись у консолі AWS та в політиках доступу. Опис залишив пустим, але при потребі його легко можна змінити.

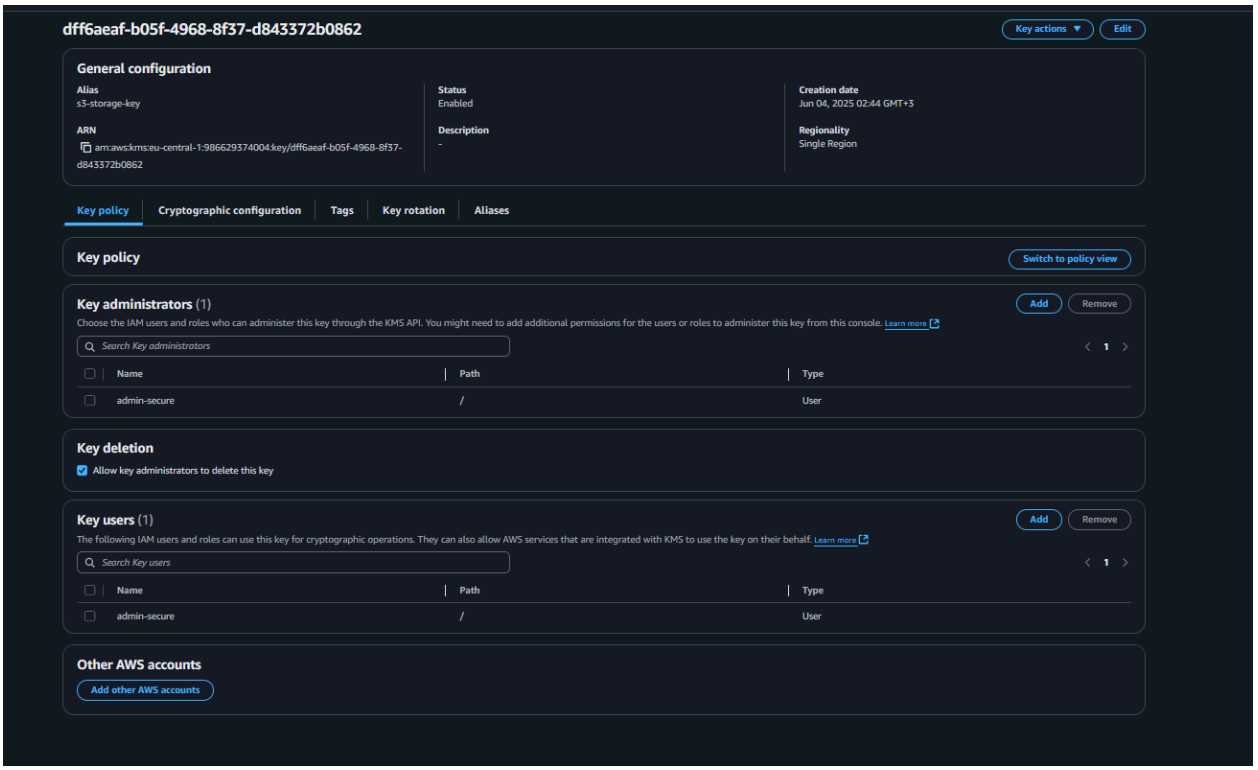

Рисунок 3.19 – Створений симетричний ключ

Далі я додав адміністратора ключа — користувача admin-secure, який має повний набір прав: змінювати політику, включати або вимикати ротацію, додавати нові дозволи тощо. Цей самий користувач був також призначений **як** користувач ключа**,** що дозволяє йому безпосередньо виконувати операції Encrypt, Decrypt, GenerateDataKey тощо. Таким чином, я самостійно матиму можливість використовувати ключ як для завантаження зашифрованих об'єктів у S3, так і для читання їх у майбутньому.

У завершальному огляді я перевірив усі параметри — специфікацію (SYMMETRIC_DEFAULT), регіональність, alias, політики доступу — та створив ключ. Згодом саме цей ключ буде прив’язаний до мого S3-бакету як дефолтний для шифрування (SSE-KMS)**.**

На підтвердження коректності налаштування шифрування в цілому, я відкрив властивості самого бакету secure-storage-lab7 (Рисунок 3.20) і переглянув розділ Default encryption. Тут чітко зазначено:

- Encryption type: Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Encryption key ARN: посилання на той самий ключ KMS, який я створив вручну (s3-storage-key)
- Bucket Key: Enabled (тобто додаткова оптимізація увімкнена)

Ці налаштування підтверджують, що всі нові файли, які завантажуються в бакет, автоматично шифруються на стороні сервера. Мені не потрібно кожного разу вручну вказувати ключ або змінювати політики — все працює за принципом "налаштував один раз — і працює завжди".

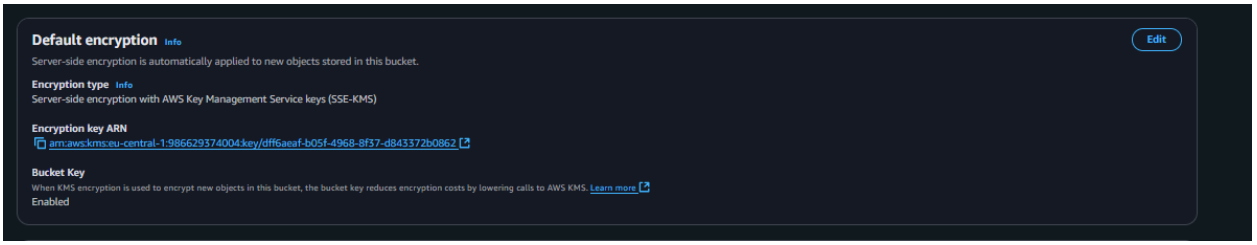

Рисунок 3.20 – Увімкнене шифрування за замовчуванням у властивостях бакету S3

Після активації серверного шифрування для мого бакету secure-storage-lab7 я вирішив перевірити, чи шифруються нові об’єкти автоматично. Для цього я завантажив файл hello_world.txt (Рисунок 3.21) і відкрив його властивості.

У секції Server-side encryption settings було чітко вказано, що об’єкт зашифровано з використанням AWS Key Management Service (KMS) [16].

Тип шифрування: SSE-KMS, а саме — серверне шифрування з ключем, який я створив раніше. Це підтверджується рядком Encryption key ARN, у якому видно, що застосовується саме мій власний ключ із KMS (...key/d1ff6eaf...).

Також зазначено, що увімкнено Bucket Key — це додатковий механізм, який допомагає знизити кількість звернень до KMS і зменшити витрати при масовому шифруванні об’єктів.

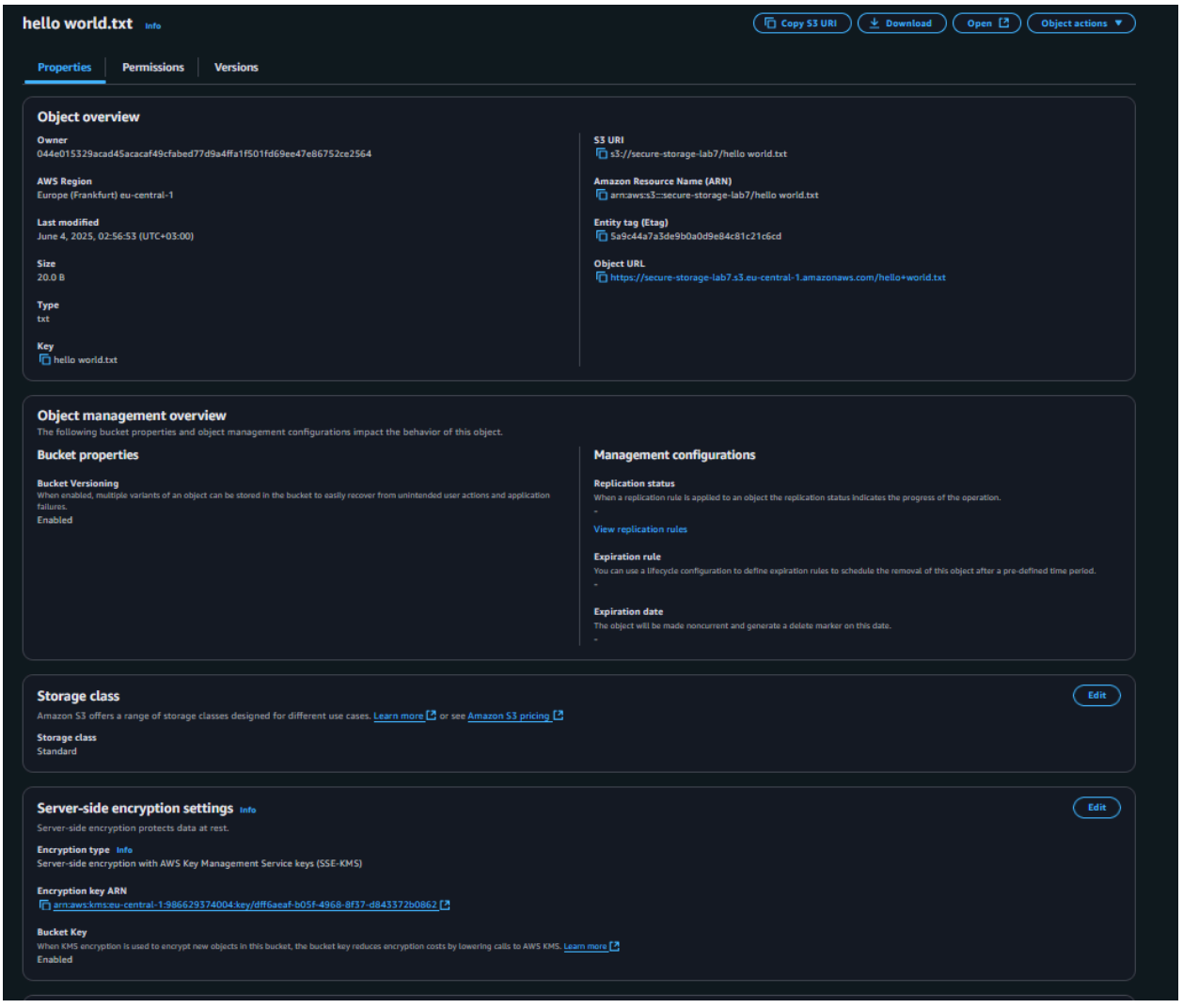

Рисунок 3.21 – Властивості об’єкта з підтвердженням шифрування через KMS

Після реалізації низки захисних механізмів, таких як контроль доступу через IAM, налаштування моніторингу з використанням CloudWatch, журналювання подій у CloudTrail, а також шифрування даних у S3 за допомогою KMS, постає питання: наскільки ці заходи є ефективними і чи забезпечують вони очікуваний рівень безпеки**?**

Зважаючи на те, що хмарне середовище є динамічним і постійно змінюється, важливо не лише впровадити засоби захисту, а й мати інструменти та критерії для оцінювання їхньої актуальності, повноти та стійкості до потенційних загроз**.**

Висновок за розділом 3

У межах третього розділу було реалізовано комплекс практичних заходів, спрямованих на впровадження засобів забезпечення інформаційної безпеки в хмарному середовищі AWS. Проведена робота охопила базові компоненти системи безпеки, що відповідають вимогам моделі CIA (Confidentiality, Integrity, Availability), і дозволила перевірити їхню функціональність у контексті реального застосування.

Сервіс AWS IAM було використано для налаштування політик контролю доступу, зокрема з дотриманням принципу найменших привілеїв. Це забезпечило розмежування повноважень між користувачами, ізоляцію прав доступу та підвищення рівня автентифікації за рахунок MFA.

Застосування AWS KMS дало змогу впровадити шифрування даних на рівні S3-бакетів, забезпечивши конфіденційність критичних об'єктів без порушення доступності або продуктивності. Було створено індивідуальну політику керування ключами, що дозволила диференціювати адміністративні й операційні права.

Інструмент AWS CloudTrail виконав функцію повного аудиту дій у хмарній інфраструктурі. Логування змін конфігурацій, запитів до API та

операцій доступу надало змогу не лише фіксувати інциденти, а й забезпечити доказову базу для подальшого аналізу.

За допомогою AWS Config було реалізовано безперервний моніторинг стану ресурсів на предмет відповідності політикам безпеки. Сервіс автоматично виявляв відхилення від заданих параметрів, що сприяло оперативному виявленню потенційних загроз.

У підсумку, реалізовані заходи показали не лише відповідність концепції CIA, а й здатність адаптувати запропоновану модель до конкретних сценаріїв хмарної інфраструктури. Комбінація сервісів AWS дала змогу побудувати взаємопов’язану систему захисту, яка функціонує автономно та дозволяє мінімізувати людський фактор у процесі контролю безпеки. Результати цієї реалізації демонструють практичну цінність обраного підходу та створюють основу для подальшого розгортання масштабованих безпекових рішень у хмарному середовищі.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досліджено комплекс питань, пов’язаних із забезпеченням інформаційної безпеки в умовах хмарних обчислень. На основі аналізу сучасних загроз, нормативно-правових вимог і практичних можливостей хмарних платформ сформовано цілісне уявлення про побудову ефективної системи захисту даних у середовищі публічної хмари.

У першому розділі здійснено систематизацію типових загроз безпеці в хмарних середовищах згідно з підходами ENISA, Cloud Security Alliance та OWASP. Розглянуто принципи побудови захисту інформації на різних рівнях — від ізоляції та контролю доступу до впровадження концепції Zero Trust. Окрему увагу приділено аналізу нормативної бази, що регламентує безпеку хмарних сервісів на міжнародному та національному рівнях — ISO/IEC 27017, NIST SP 800-144, GDPR, а також українським стандартам у сфері КСЗІ.

У другому розділі розроблено архітектуру інформаційної безпеки у хмарному середовищі на основі сервісів Amazon Web Services. Виокремлено п’ять функціональних блоків захисту: керування доступом (IAM), аудит подій (CloudTrail), перевірка відповідності (AWS Config), шифрування даних (KMS) та моніторинг інцидентів (Security Hub, CloudWatch, SNS). Визначено архітектурні принципи побудови взаємозв’язків між цими компонентами для забезпечення безперервного контролю й масштабованості.

У третьому розділі здійснено практичну реалізацію моделі в тестовому середовищі AWS. Реалізовані налаштування сервісів дозволили побудувати автоматизовану систему захисту, яка забезпечує реєстрацію подій, контроль за конфігураціями, реагування на інциденти та підтримку безпеки без потреби в постійному ручному втручанні. Така реалізація підтвердила життєздатність і прикладну цінність розробленої архітектури.

Загалом виконана робота засвідчує, що ефективна система захисту у хмарному середовищі може бути створена шляхом інтеграції існуючих сервісів AWS за умови правильного проектування логіки їх взаємодії. Запропонована

модель відповідає основним принципам інформаційної безпеки — конфіденційності, цілісності та доступності — і має високий потенціал до масштабування, адаптації та впровадження в реальні проєкти. Її структура дозволяє подальший розвиток через інтеграцію із зовнішніми SIEM-рішеннями, розширення на мультихмарні середовища та впровадження інтелектуальних засобів аналізу поведінки.

Отримані результати можуть бути використані як методологічна та практична основа для впровадження комплексних систем інформаційної безпеки у державних та комерційних структурах, що використовують хмарну інфраструктуру для зберігання та обробки вразливої інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISO/IEC 17788:2014. Information technology – Cloud computing – Overview and vocabulary. – Geneva: ISO, 2014. – 22 p.

2. NIST SP 800-144: Guidelines on Security and Privacy in Public Cloud Computing [Електронний ресурс]. – Gaithersburg: NIST, 2011. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf

3. ENISA. Cloud Computing Risk Assessment [Електронний ресурс]. – Режим доступу: https://www.enisa.europa.eu/publications/cloud-computing-risk-assessment

4. Cloud Security Alliance. Security Guidance for Critical Areas of Focus in Cloud Computing v4.0 [Електронний ресурс]. – Режим доступу: https://cloudsecurityalliance.org/artifacts/security-guidance-v4/

5. OWASP Foundation. OWASP Top 10 for Cloud [Електронний ресурс]. – Режим доступу: https://owasp.org/www-project-top-ten/

6. Cybersecurity and Infrastructure Security Agency (CISA). *CISA* AA20-352A: Advanced Persistent Threat Compromise of Government Agencies, Critical Infrastructure, and Private Sector Organizations via Supply Chain Compromise of SolarWinds Orion Software. – Washington, D.C.: CISA, Dec. 2020. – [Електронний ресурс]. – Режим доступу: https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a

7. General Data Protection Regulation (GDPR). Regulation (EU) 2016/679. – Official Journal of the European Union, 2016.

8. Закон України «Про інформацію» від 02.10.1992 № 2657-XII [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2657-12

9. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2297-17

10. AWS Security Hub documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/securityhub/

11. Microsoft Azure. Security documentation [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/security/

12. Google Cloud Security Overview [Електронний ресурс]. – Режим доступу: https://cloud.google.com/security

13. Amazon Web Services. Documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/

14. AWS Identity and Access Management documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

15. AWS CloudTrail documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/cloudtrail/

16. AWS Key Management Service (KMS) documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/kms/

17. AWS Config documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/config/

18. Amazon Web Services. AWS Lambda – Developer Guide [Електронний ресурс]. – URL: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

19. Amazon Web Services. Amazon CloudWatch – *User Guide* [Електронний ресурс]. – URL: docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/

20. Amazon Web Services. Amazon Simple Storage Service (S3) – User Guide [Електронний ресурс]. – URL: docs.aws.amazon.com/AmazonS3/latest/userguide/

21. Amazon Web Services. Amazon Virtual Private Cloud (VPC) – User Guide [Електронний ресурс]. – URL: docs.aws.amazon.com/vpc/latest/userguide/

22. Amazon Web Services. AWS Management Console – Getting Started Guide [Електронний ресурс]. – URL: docs.aws.amazon.com/awsconsolehelpdocs/latest/gsg/

23. Stallings W. Cryptography and Network Security: Principles and Practice. – 7th ed. – Pearson, 2016. – 752 p.

24. Bishop M. Computer Security: Art and Science. – 2nd ed. – Addison-Wesley, 2018. – 1344 p.

25. Гнатенко І. В. Інформаційна безпека в комп’ютерних системах і мережах: навч. посіб. – К.: Ліра-К, 2021. – 368 с.

26. Колесник В. О. Хмарні обчислення: архітектура, сервіси, безпека. – К.: ДП НВЦ «Пріоритети», 2020. – 312 с.

27. Гуменюк С. В., Храмцов О. С. Управління інформаційною безпекою підприємства в умовах хмарних технологій // Інформаційні технології і засоби навчання. – 2020. – №4(78). – С. 109–121.

ДОДАТКИ

Додаток А

Динаміка зростання світового ринку хмарних обчислень у 2018–2025 роках