Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Методи захисту даних і контролю активності вузлів у хмарній інфраструктурі

Виконала: студентка VI курсу, групи СБмз-61
спеціальності 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

(підпис) Кіт С.Б. (прізвище та ініціали)

Керівник (підпис) Скарга-Бандурова І.С. (прізвище та ініціали)

Нормоконтроль (підпис) Стадник М.А. (прізвище та ініціали)

Завідувач кафедри (підпис) Загородна Н.В. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Кіт Софії Богданівні
(прізвище, ім'я, по батькові)

1. Тема роботи Методи захисту даних і контролю активності вузлів у хмарній інфраструктурі

Керівник роботи Скарга-Бандурова Інна Сергіївна, д.т.н., професор кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «_11_» грудня_ 2025 року № _4/7-1066

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Наукові публікації про загрози хмарної безпеки та проблем безпеки хмарних середовищ

4. Зміст роботи (перелік питань, які потрібно розробити):
Вступ, Розділ 1. Теоретичні основи захисту даних та контролю активності вузлів у хмарній інфраструктурі, 1.1 Поняття та моделі хмарних обчислень: IaaS, PaaS, SaaS; архітектури хмарної інфраструктури, 1.2 Загрози інформаційній безпеці у хмарі: ризики конфіденційності, цілісності та доступності, 1.3 Моделі довіри та відповідальності: Shared Responsibility Model, SLA та вимоги комплаєнсу, 1.4 Основні підходи до захисту даних: шифрування, керування ключами, контроль доступу, резервування, 1.5 Контроль активності вузлів і телеметрія: журнали подій, метрики, трасування, інцидент-менеджмент, 2 Розділ Методи та моделі реалізації захисту даних і моніторингу активності вузлів, 2.1 Формалізація задачі: об'єкти захисту, загрози, вимоги та критерії ефективності, 2.2 Методи захисту даних «у русі», «у спокої» та «у використанні» (TLS/mTLS, KMS/HSM, секрети, конфіденційні) Розділ 3. Практична реалізація та оцінювання ефективності запропонованих рішень, 3.1 Вибір хмарної платформи та інструментарію Azure: обґрунтування та обмеження, 3.2 Реалізація захисту даних в Azure: політики шифрування, керування ключами, секрети 4 Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Титульна сторінка 2. Актуальність теми і наукова новизна. 3. Мета роботи і завдання. 4. Об'єкт/предмет дослідження та підхід. 5. Контекст: моделі хмари та відповідальність. 6. Модель загроз і оцінювання ризику. 7. Вимоги до системи. 8. Запропонована архітектура. 9. Захист даних: шифрування і керування ключами. 10. Контроль активності вузлів і телеметрія. 11. Практична реалізація (Azure). 12. Детекція та реагування. 13. Оцінювання ефективності. 14. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н. доцент		
Безпека в надзвичайних ситуаціях	Теслюк В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 19.09 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	21.09.2025-29.09.2025	*Виконано*
2.	Підбір наукових джерел про хмарні платформи	30.09.2025-03.10.2025	*Виконано*
3.	Переклад та опрацювання наукових джерел про		
	дослідження методів захисту відомих хмарних		
	платформ	04.10.2025-10.10.2025	*Виконано*
4.	Виконання дослідження щодо аналіз інструментів для		
	організації інфраструктури та безпеки публічної		
	хмари	11.10.2025-17.10.2025	*Виконано*
5.	Оформлення розділу «Теоретичні основи захисту		
	даних та контролю активності вузлів у хмарній		
	інфраструктурі»	18.10.2025-24.10.2025	*Виконано*
6.	Оформлення розділу «Методи та моделі реалізації		
	захисту даних і моніторингу активності вузлів»	25.10.2025-31.10.2025	*Виконано*
7.	Оформлення розділу «Практична реалізація та	01.11.2025-07.11.2025	*Виконано*
	оцінювання ефективності запропонованих рішень»		
8.	Виконання завдання до підрозділу «Охорона праці»	10.12.2025-12.12.2025	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в надзви-		
	чай них ситуаціях»	10.12.2025-12.12.2025	*Виконано*
10.	Оформлення кваліфікаційної роботи	13.12.2025-14.12.2025	*Виконано*
11.	Нормоконтроль	15.12.2025-17.12.2025	*Виконано*
12.	Перевірка на плагіат	17.12.2025-19.12.2025	*Виконано*
13.	Попередній захист кваліфікаційної роботи	22.12.2025-23.12.2025	*Виконано*
14.	Захист кваліфікаційної роботи	24.12.2025	

Студент _____ Кіт С.Б.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Скарга-Бандурова І.С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Методи захисту даних і контролю активності вузлів у хмарній інфраструктурі // ОР «Магістр» // Кіт Софія Богданівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБмз-61 // Тернопіль, 2025 // С. 160, рис. – 21, табл. – 45, кресл. – 12, додат. – 2

Ключові слова: хмарні обчислення, інформаційна безпека, загрози та вразливості, автентифікація, авторизація, контроль доступу, шифрування даних

Кваліфікаційна робота присвячена дослідженню методів захисту даних і контролю активності вузлів у хмарній інфраструктурі.

У першому розділі розглянуто предметну область і обґрунтовано актуальність теми, пов'язаної із забезпеченням безпеки даних та надійності роботи систем у хмарному середовищі. Наведено базові поняття, терміни та типову архітектуру хмарних рішень, які використовуються як основа для подальшого проєктування. Проаналізовано основні загрози, вразливості та типові сценарії атак, характерні для хмарних сервісів і веб-застосунків. Описано принципи побудови системи захисту (CIA-тріада, контроль доступу, шифрування, журналювання) та вимоги до рішення. Сформульовано мету, завдання роботи, об'єкт і предмет дослідження, а також окреслено структуру подальших розділів.

У другому розділі виконано проєктування запропонованого рішення та визначено функціональні й нефункціональні вимоги до системи. Обґрунтовано вибір технологій, сервісів і засобів реалізації, а також описано логічну та фізичну структуру компонентів. Розроблено модель даних і схеми взаємодії між модулями (користувачі, доступи, журнали подій, політики

безпеки, обмін повідомленнями). Детально наведено алгоритми основних процесів: автентифікація/авторизація, керування ролями, шифрування даних, збір логів і первинна обробка подій безпеки. Підготовлено основу для практичного впровадження, що реалізується та перевіряється в третьому розділі.

У третьому розділі подано практичну реалізацію запропонованої системи захисту даних і моніторингу активності вузлів у хмарному середовищі. Розглянуто архітектуру рішення, порядок розгортання компонентів та налаштування контролів безпеки (керування доступом, шифрування, робота з ключами/секретами, журналювання). Описано організацію збору телеметрії (логи, метрики, події) та механізми виявлення підозрілої активності з формуванням сповіщень і сценаріїв реагування. Наведено методику експериментальної перевірки, тестові сценарії та критерії оцінювання ефективності, зокрема показники якості детекцій і швидкості реагування. Зроблено висновки щодо результативності впроваджених рішень і визначено напрями подальшого удосконалення системи.

ANNOTATION

Methods for data protection and node activity control in cloud infrastructure // Thesis of educational level «Master» // Sofiya Kit // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cyber Security, group СБмз-61 // Ternopil, 2025 // P. 160, figs. - 21, tables - 45, tbls. 1, drws. 12, apps. 2.

Key words: cloud computing, information security, threats and vulnerabilities, authentication, authorization, access control, data encryption

The qualification work is devoted to the study of methods for data protection and control of node activity in the cloud infrastructure.

The first section considers the subject area and justifies the relevance of the topic related to ensuring data security and reliability of systems in the cloud environment. The basic concepts, terms and typical architecture of cloud solutions are presented, which are used as the basis for further design. The main threats, vulnerabilities and typical attack scenarios characteristic of cloud services and web applications are analyzed. The principles of building a protection system (CIA triad, access control, encryption, logging) and the requirements for the solution are described. The goal, objectives of the work, object and subject of the study are formulated, and the structure of subsequent sections is outlined.

In the second section, the design of the proposed solution is performed and functional and non-functional requirements for the system are determined. The choice of technologies, services and means of implementation is justified, and the logical and physical structure of the components is described. A data model and interaction schemes between modules (users, access, event logs, security policies, messaging) are developed. The algorithms of the main processes are presented in detail: authentication/authorization, role management, data encryption, log

collection and primary processing of security events. A basis for practical implementation is prepared, which is implemented and verified in the third section.

The third section presents a practical implementation of the proposed data protection and node activity monitoring system in a cloud environment. The solution architecture, the procedure for deploying components and configuring security controls (access control, encryption, working with keys/secrets, logging) are considered. The organization of telemetry collection (logs, metrics, events) and mechanisms for detecting suspicious activity with the formation of notifications and response scenarios are described. The experimental verification methodology, test scenarios and performance evaluation criteria are presented, in particular indicators of detection quality and response speed. Conclusions were drawn regarding the effectiveness of the implemented solutions and directions for further improvement of the system were identified.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CISO – Chief Information Security Officer (керівник відділу IT-безпеки, директор з IT-безпеки).

CIO – Chief Information Officer (головний менеджер з інформатизації, директор з інформаційних технологій).

CSO – Chief Security Officer (директор із комп'ютерної (інформаційної) безпеки, начальник служби інформаційної безпеки; особа, відповідальна за інформаційну безпеку організації).

CRM – Customer relationship management (Управління відносинами з клієнтами).

ZTNA – Zero Trust Network Access (Мережевий доступ з нульовою довірою).

SOA – Service-oriented architecture (Сервіс-орієнтована архітектура).

VPN – Virtual Private Network (Віртуальне приватне мережеве підключення).

VPC – Virtual Private Cloud (Віртуальна приватна хмара).

IRM – Integrated Risk Management (Інтегроване управління ризиками).

IT – Information Technology (інформаційні технології).

OT – Operational Technology (операційні технології).

IOT – Internet of Things (Інтернет речей).

ESG – Environmental, social, and governance (Екологічне, соціальне та корпоративне управління).

VM – Virtual Machine (Віртуальна машина).

LVM – Local Virtual Machine (Локальна віртуальна машина).

AWS – Amazon Web Services (Веб-сервіси Амазон).

ПЗ – програмне забезпечення.

БД – база даних.

ХТ – хмарні технології.

ЗМІСТ

ВСТУП

Актуальність теми. Активне впровадження хмарних сервісів у бізнесі та державному секторі підвищує вимоги до конфіденційності, цілісності й доступності даних. Хмарні середовища характеризуються спільною відповідальністю провайдера та замовника, складною моделлю доступу, великою кількістю інтеграцій і високою динамікою змін, що збільшує ризики помилок конфігурації та кіберінцидентів.

Мета і задачі дослідження. Метою кваліфікаційної роботи є розроблення та обґрунтування комплексу технічних і організаційних заходів (моделі/підходу) забезпечення інформаційної безпеки в хмарному середовищі для зниження ризиків витоку даних і несанкціонованого доступу.

Для досягнення поставленої мети було потрібно виконати наступні завдання:

- проаналізувати сучасні загрози та типові вразливості хмарних систем;
- дослідити нормативно-методичну базу та кращі практики (підходи до IAM, шифрування, журналювання, моніторингу, резервування);
- сформувати вимоги безпеки та модель ризиків для хмарної інфраструктури;
- розробити архітектуру захисту (контроль доступу, автентифікація/авторизація, шифрування, керування ключами, сегментація, логування та реагування);
- реалізувати/налаштувати прототип або приклад конфігурації для типового сценарію використання хмари;
- оцінити ефективність запропонованих рішень (ризик-аналіз, тестування, порівняння показників до/після) та сформувати рекомендації впровадження.

Об’єкт дослідження. Процеси забезпечення інформаційної безпеки інформаційних систем та даних у хмарному середовищі.

Предмет дослідження. Методи, моделі та засоби захисту хмарної інфраструктури: автентифікація й авторизація, контроль доступу, шифрування та керування ключами, журналювання й моніторинг, управління ризиками та інцидентами.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у вдосконаленні підходу до вибору та узгодження контролів безпеки для хмарного середовища на основі поєднання класифікації даних, моделі загроз і ризик-орієнтованої пріоритизації; у формалізації структури багаторівневого захисту (IAM + шифрування + моніторинг) з визначенням взаємозв'язків між політиками доступу, подіями безпеки та процедурами реагування.

Практичне значення одержаних результатів. Результати роботи можуть бути використані під час проєктування та експлуатації хмарних рішень в організаціях для підвищення рівня захисту даних. Запропоновані рекомендації, архітектурні схеми та приклади налаштувань/політик можуть слугувати основою для впровадження контролів доступу, шифрування, журналювання та моніторингу, а також для підготовки внутрішніх регламентів і чек-листів безпеки.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на: XIV Міжнародній науково-технічній конференції молодих учених та студентів Актуальні задачі сучасних технологій. (м.Тернопіль).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (див. Додаток А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури із 86 найменувань та 1 додаток. Загальний обсяг кваліфікаційної роботи складає 159 сторінок, з них 152 сторінок основного тексту, який містить 21 рисунки та 45 таблиць.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗАХИСТУ ДАНИХ ТА КОНТРОЛЮ АКТИВНОСТІ ВУЗЛІВ У ХМАРНІЙ ІНФРАСТРУКТУРІ

1.1 Поняття та моделі хмарних обчислень: IaaS, PaaS, SaaS; архітектури хмарної інфраструктури

Хмарні обчислення (cloud computing) – це модель надання обчислювальних ресурсів і цифрових сервісів через мережу з можливістю швидкого масштабування, керування за потребою та оплатою за фактом використання. На відміну від традиційної IT-інфраструктури, де організація закуповує й утримує власні сервери, системи зберігання та мережеве обладнання, у хмарі споживач отримує доступ до стандартизованих ресурсів (обчислення, зберігання, мережа, платформи, прикладні сервіси) як до послуги. Ключові властивості хмарних обчислень включають: самообслуговування “на вимогу” (програмне замовлення ресурсів), широкодоступність через мережу, об’єднання ресурсів провайдера, еластичність (швидке збільшення/зменшення потужностей) та вимірюваність/прозоре білінг-облікування. [1]

Найбільш уживаною класифікацією є поділ хмарних сервісів на три базові моделі: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) та Software as a Service (SaaS). Вони відрізняються рівнем абстракції та розподілом відповідальності між провайдером і споживачем.

IaaS надає “віртуалізовану інфраструктуру” – віртуальні машини/контейнери, мережі, сховища, балансувальники. Користувач керує операційними системами, проміжним ПЗ, прикладними компонентами та налаштуваннями безпеки на рівні ОС і застосунку. IaaS підходить для міграції спадкових систем (lift-and-shift), побудови приватних сегментів у публічній хмарі, а також для сценаріїв, де потрібен контроль над мережевою топологією та ОС.

PaaS надає “платформу виконання” – керовані середовища для розгортання вебзастосунків, API, фонових обробників, керовані СУБД, черги, кеші тощо. Споживач зосереджується на коді та конфігурації застосунку, тоді як провайдер бере на себе ОС, середовище виконання, масштабування, частину моніторингу й патчинг. PaaS зменшує операційне навантаження та стандартизує розгортання, але накладає обмеження на глибокі зміни середовища[2].

SaaS надає готовий прикладний продукт (пошта, документообіг, CRM/ERP, аналітика, сервіс-деск тощо). Користувач керує переважно даними, ролями доступу та налаштуваннями на рівні бізнес-логіки, а провайдер забезпечує всю технічну складову. SaaS максимально швидко впроваджується, проте має найменшу гнучкість і вимагає ретельного контролю доступів, інтеграцій та політик збереження/експорту даних, наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння IaaS, PaaS, SaaS за рівнем відповідальності та контролю

Критерій	IaaS	PaaS	SaaS
Що надається	Віртуальна інфраструктура (VM/мережа/сховище)	Платформа і керовані сервіси (runtime, БД, черги)	Готовий застосунок
Хто керує ОС/патчингом	Споживач	Провайдер	Провайдер
Хто керує застосунком/кодом	Споживач	Споживач	Провайдер (споживач – налаштування)
Гнучкість конфігурацій	Висока	Середня	Обмежена
Операційне навантаження	Високе	Середнє/низьке	Низьке
Типовий фокус безпеки	ОС, мережа, IAM, hardening	IAM, секрети, конфігурації сервісів	IAM, політики доступу, DLP, аудит

Для задач цієї роботи важливо, що зі зміною моделі (від IaaS до SaaS) зменшується контроль над “нижніми” шарами інфраструктури, але зростає значущість коректної організації ідентифікації/автентифікації, політик доступу, журналювання та контролю активності на рівні сервісів і користувачів. [3]

Хмарна інфраструктура зазвичай описується як багатошарова система, у якій фізичні ресурси абстрагуються до керованих віртуальних сутностей, а керування та безпека реалізуються через централізовані контрольні площини (control plane).

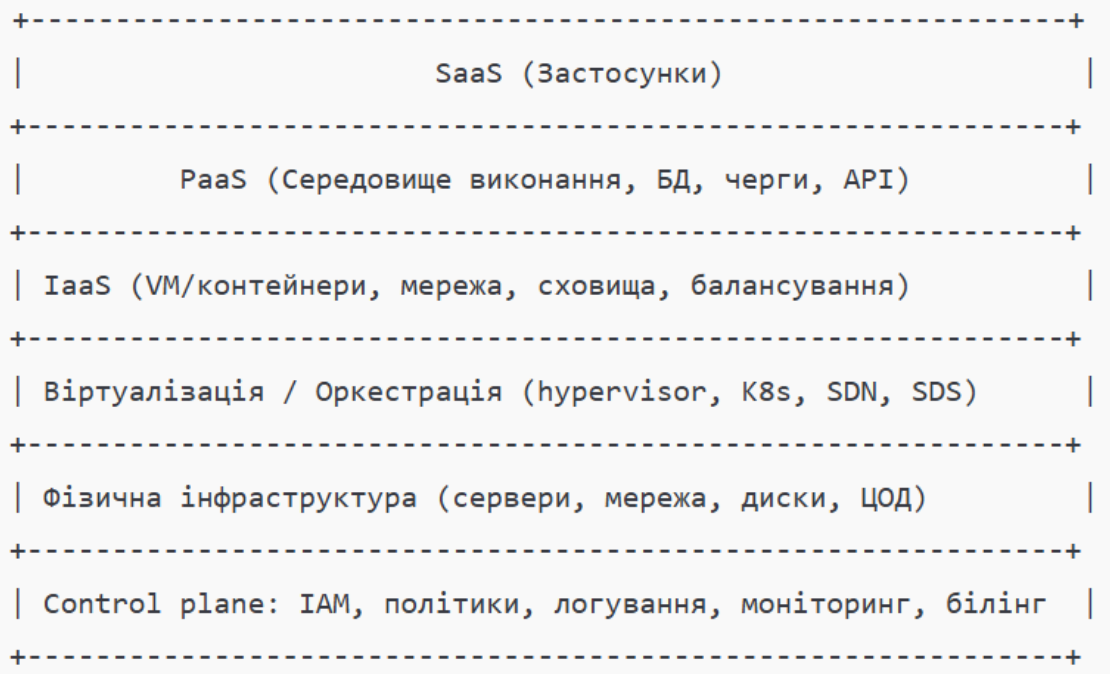

Рисунок 1.1 – Узагальнена шарова модель хмарних обчислень

На практиці архітектура хмари включає щонайменше такі групи компонентів:

- Compute-рівень – віртуальні машини, контейнери, кластерні середовища, безсерверні функції. Для контролю активності вузлів саме compute-рівень є джерелом телеметрії про процеси, мережеві з’єднання, завантаження, події доступу та зміни конфігурації.

- Storage-рівень – об’єктні сховища, блокові томи, файлові сховища, керовані СУБД. Захист даних тут включає шифрування, керування ключами, політики життєвого циклу, резервне копіювання, незмінюваність (immutability) та контроль доступу до об’єктів/таблиць.
- Network-рівень – віртуальні мережі, підмережі, маршрутизація, міжмережеві екрани, приватні канали до хмарних сервісів, балансувальники. У хмарі мережа стає “програмованою” (SDN), що дає гнучкість, але потребує дисципліни в сегментації та застосуванні політик.
- Management & Security-рівень (керування і безпека) – ідентифікація та доступ (IAM), керування секретами, централізоване журналювання, моніторинг, інвентаризація ресурсів, політики відповідності, керування конфігураціями (Infrastructure as Code), а також механізми реагування на інциденти.

Типові архітектурні патерни розгортання

Хмарні застосунки часто будуються як мікросервіси або модульні компоненти, що працюють у контейнерах, керованих оркестратором. Альтернативою є serverless-підхід, де логіка виконується у вигляді функцій/керованих компонентів. Для корпоративних систем поширені гібридні архітектури, коли частина даних або сервісів залишається у приватному контурі, а публічна хмара використовується для пікових навантажень, аналітики чи резервування. Додатково зростає роль multi-cloud як способу підвищення відмовостійкості та зменшення залежності від одного провайдера, але ціною більш складного моніторингу, уніфікації політик і централізації аудит-логів. [4]

Отже, моделі IaaS/PaaS/SaaS задають різний рівень контролю над інфраструктурою, а багатошарова архітектура хмари визначає, де саме потрібно реалізовувати механізми захисту даних і які джерела телеметрії використовувати для контролю активності вузлів, наведено в таблиці 1.2.

Таблиця 1.2 – Приклади прив’язки задач безпеки до шарів архітектури

Шар	Типові об’єкти	Типові події для контролю активності	Основні ризики
Compute	VM/контейнери/функції	старт/зупинка, доступ по SSH/RDP, зміни образів, процеси, мережеві сесії	компрометація вузла, виконання шкідливого коду
Storage	БД/сховища/бекапи	читання/запис/видалення, зміни політик, експорт даних	витік, несанкціонований доступ, руйнування цілісності
Network	VPC/VNet, FW, LB	зміна правил FW, відкриття портів, аномальні потоки	бокове переміщення, DDoS, перехоплення/сканування
Control plane	IAM, політики, журнали	зміни ролей, створення ключів, вимкнення логування	ескалація привілеїв, приховування слідів

У подальших підрозділах детально розглянемо, які методи (шифрування, керування ключами, політики доступу, централізоване логування, кореляція подій і виявлення аномалій) забезпечують необхідний рівень конфіденційності та керованості саме в динамічному хмарному середовищі. [5]

1.2 Загрози інформаційній безпеці у хмарі: ризики конфіденційності, цілісності та доступності

Хмарна інфраструктура істотно змінює профіль ризиків інформаційної безпеки порівняно з класичними on-premise середовищами. Причина полягає в тому, що ресурси хмари керуються програмно через API, мають високу динаміку (масштабування, авто-відновлення, короткоживучі екземпляри), використовують багатотенантність (спільне фізичне середовище для різних клієнтів) та значною мірою залежать від керованих сервісів провайдера. У такій моделі класична тріада CIA (Confidentiality, Integrity, Availability – конфіденційність, цілісність, доступність) залишається базою оцінки безпеки, але механізми реалізації й типові сценарії атак набувають специфічних “хмарних” проявів: помилки конфігурацій, компрометація облікових даних

IAM, зловживання привілеями, ризики ланцюга постачання, атаки на контрольну площину (control plane), а також інциденти на стороні провайдера. [6]

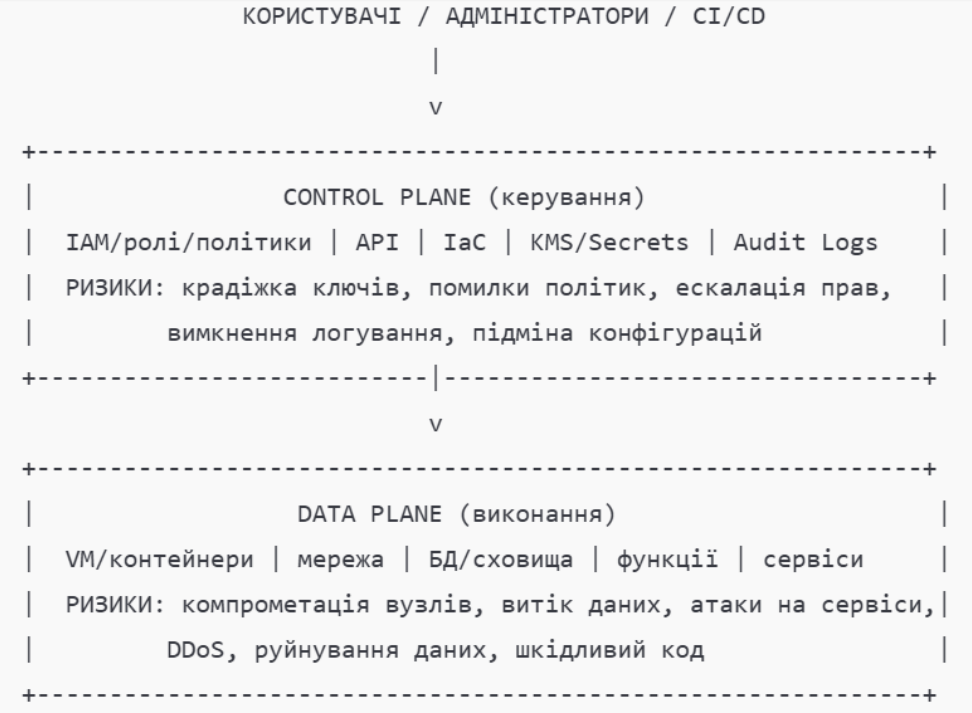

Рисунок 1.2 – Поверхня атак у хмарі (Data plane vs Control plane)

У загальному випадку ризики у хмарі формуються не лише зовнішніми атаками, а й неправильними налаштуваннями та організаційними недоліками: відсутність сегментації, надмірні привілеї, неконтрольований випуск доступів, слабкий аудит, неузгодженість політик між середовищами (dev/test/prod). Далі розглянемо загрози за кожним компонентом тріади CIA.

1.2.1 Ризики конфіденційності (Confidentiality)

Конфіденційність означає захист даних від несанкціонованого доступу та розкриття. У хмарі типові сценарії порушення конфіденційності часто не потребують “злому” в класичному сенсі – достатньо експлуатувати помилки доступу або вкрадені облікові дані.

Ключові загрози конфіденційності: [7]

- Помилки конфігурації зберігання та публічні доступи. Найпоширеніший сценарій – випадкове відкриття об’єктного сховища, публічний доступ до резервних копій, снапшотів дисків або тестових БД. У хмарі “публічність” інколи задається одним параметром політики, що створює високий ризик людської помилки.
- Компрометація IAM-облікових даних і токенів. Облікові дані хмарного провайдера (API-ключі, токени, ключі доступу сервісних акаунтів) є “універсальними ключами” до ресурсів. Їх можуть викрасти через фішинг, витік із репозиторіїв, уразливості CI/CD, шкідливе ПЗ на робочій станції або перехоплення токенів у середовищі виконання.
- Надмірні привілеї та некоректні політики доступу. Політики “admin-подібного” рівня для сервісних акаунтів, відсутність принципу найменших привілеїв (Least Privilege), спільні облікові записи та слабкі процеси ревізії ролей призводять до того, що навіть часткова компрометація однієї сутності відкриває широкий доступ до даних.
- Небезпечні API та помилки інтеграцій. Оскільки керування ресурсами відбувається через API, недостатній контроль доступу до API, відсутність обмежень за джерелом запиту, неправильна конфігурація OAuth/SSO, помилки у gateway/WAF можуть розкрити дані або дозволити їх ексфільтрацію.
- Багатотенантність та ризики ізоляції. Хоча провайдери реалізують жорстку ізоляцію, теоретичні та практичні сценарії “cross-tenant” (наприклад, через вразливості гіпервізора або side-channel атаки) розглядаються як висококритичні, особливо для систем із чутливими даними.
- Витік через журнали, телеметрію та резервні копії. Логи застосунків можуть містити персональні дані, токени, ключі, “секрети” та фрагменти payload. Аналогічно, резервні копії часто мають ширші права доступу й довший життєвий цикл, що робить їх привабливою ціллю.

1.2.2 Ризики цілісності (Integrity)

Цілісність означає захист даних і конфігурацій від несанкціонованої модифікації, підміни або руйнування структури. Для хмари характерно, що атака на цілісність може бути спрямована не лише на дані, а й на конфігурацію інфраструктури як коду (IaC), політики доступу, образи контейнерів або параметри керованих сервісів.

Основні загрози цілісності

Несанкціоновані зміни у сховищах і БД. Зловмисник, який отримав права запису, може непомітно модифікувати записи, змінити історію транзакцій, підмінити файли або знищити версії. Ризик зростає, якщо не використовується версіонування об’єктів, контроль цілісності, журнали незмінюваних подій та розділення обов’язків (SoD).

Підміна артефактів постачання – образів, пакетів, шаблонів. У контейнерних середовищах атакують supply chain: підміна образу в registry, компрометація залежностей, вставка бекдорів у бібліотеки. Якщо CI/CD має надмірні привілеї в хмарі, підміна pipeline може призвести до масової зміни конфігурацій або розгортання шкідливих компонентів.

Маніпуляції з контрольними політиками та конфігурацією. Зміна правил мережевих фільтрів, security groups, ролей IAM, налаштувань KMS або вимкнення аудит-логів є атаками на цілісність “середовища”. Наслідком стає спрощення подальшого витоку даних або саботаж, що часто складніше виявити, ніж прямий доступ до файлів.

Конфігураційний дрейф і помилки автоматизації. Автоматизовані системи (autoscaling, оркестратори, IaC) можуть помилково застосувати неправильні параметри або розгорнути застарілу конфігурацію. Такі інциденти інколи зовні виглядають як атака, але є наслідком слабкого контролю змін.

1.2.3 Ризики доступності (Availability)

Доступність означає здатність сервісів і даних бути доступними у потрібний час із визначеними показниками SLA/SLO. У хмарі доступність порушується як через атаки (DDoS, саботаж), так і через технічні збої у провайдера, помилки конфігурації та ефект каскадної залежності керованих сервісів.

Ключові загрози доступності: [8]

- DDoS-атаки та перевантаження ресурсів. Публічні endpoints (веб-додатки, API) є типовою ціллю. У хмарі додатковим ризиком є “економічний DDoS” – примусове масштабування, що генерує витрати або вичерпує квоти.
- Вичерпання квот і лімітів. Багато сервісів мають обмеження на кількість запитів, інстансів, дисків, IP-адрес. Атакувальник або помилка автоматизації може “вибити” квоту, після чого система перестає масштабуватися чи створювати нові компоненти.
- Руйнування або шифрування даних (ransomware), знищення резервних копій. Якщо зловмисник отримує привілеї на рівні control plane, він може видалити снапшоти, знищити бекапи, вимкнути реплікацію. Це перетворює інцидент доступності на катастрофічний сценарій відновлення.
- Збої провайдера та регіональні інциденти. Хоча великі провайдери мають високу надійність, інциденти в конкретній зоні/регіоні або у ключовому керованому сервісі можуть призвести до недоступності бізнес-критичних систем, особливо якщо архітектура не передбачає відмовостійкість між зонами/регіонами, наведено в таблиці 1.3-1.4.

Таблиця 1.3 – Приклади загроз у хмарі за тріадою CIA

Компонент CIA	Типові загрози	Наслідки	Типові контрзаходи (узагальнено)
Конфіденційність	відкриті сховища/бекапи; крадіжка IAM-ключів; надмірні права; витік через логи	витік персональних/комерційних даних, компрометація таємниць	Least Privilege, MFA/SSO, KMS/Secrets, DLP, приватні endpoints, аудит доступів
Цілісність	підміна образів/артефактів; несанкціоновані зміни політик; зміни в БД/сховищах	фальсифікація даних, бекдори, приховані зміни, саботаж	підпис артефактів, контроль змін (IaC), immutable-логи, версіонування, SoD, політики незмінюваності
Доступність	DDoS; квотне виснаження; ransomware; регіональні інциденти	простій сервісів, втрата виручки, зупинка процесів	WAF/DDoS-захист, rate limiting, multi-AZ/DR, резервні копії з immutability, runbooks відновлення

Таблиця 1.4 – Де виникають загрози у хмарній архітектурі

Рівень	Приклади “хмарних” загроз	Чому це критично
Control plane (IAM/API/IaC)	компрометація токенів, ескалація ролей, вимкнення логів, підміна конфігурацій	один інцидент дає керування багатьма ресурсами одразу
Compute (VM/контейнери)	експлуатація уразливостей ОС/рантайму, бокове переміщення, шкідливі образи	компрометований вузол стає точкою ексфільтрації та саботажу
Storage/DB	відкритий доступ, видалення версій, підміна даних	прямий вплив на конфіденційність і цілісність
Network	неправильні правила доступу, відкриті порти, відсутність сегментації	збільшується площа атак і можливість lateral movement

Таким чином, у хмарі загрози тріади CIA посилюються через програмно-керовану природу інфраструктури та високу концентрацію повноважень у control plane. Порушення конфіденційності найчастіше пов’язане з помилками доступу та компрометацією IAM, порушення цілісності – з підміною

конфігурацій/артефактів і зловживанням привілеями, а доступності – з DDoS, квотним виснаженням, ransomware та регіональними інцидентами. Для подальших розділів важливо, що ефективна протидія цим ризикам потребує одночасно технічних механізмів (шифрування, IAM, журналювання, сегментація, резервування) і процесних практик (контроль змін, ревізія доступів, реагування на інциденти), що в комплексі забезпечує керованість і надійність хмарного середовища.

1.3 Моделі довіри та відповідальності: Shared Responsibility Model, SLA та вимоги комплаєнсу

Перехід до хмарної інфраструктури змінює традиційні підходи до управління ризиками: частина контролів “спускається” на рівень провайдера, натомість у замовника з’являється потреба точніше формалізувати довіру, розподіл відповідальності та договірні гарантії. На практиці це реалізується через три взаємопов’язані механізми: (1) Shared Responsibility Model (модель спільної відповідальності), (2) SLA (Service Level Agreement – угода про рівень сервісу) як частина контрактних зобов’язань, та (3) комплаєнс-вимоги (законодавчі й галузеві стандарти), які визначають мінімально прийнятний рівень контролів та доказів їх виконання. [9-10]

1.3.1 Shared Responsibility Model як основа “операційної довіри”

Shared Responsibility Model описує, які безпекові й операційні функції виконує провайдер, а які – клієнт. У класичному формулюванні провайдер відповідає за “security of the cloud” (фізичні ЦОД, мережі, гіпервізор/віртуалізація, базова платформа), тоді як клієнт відповідає за “security in the cloud” (дані, ідентичності, конфігурації, робочі навантаження).

Сам принцип та його залежність від моделі сервісу (IaaS/PaaS/SaaS) прямо підкреслюється в документації провайдерів.

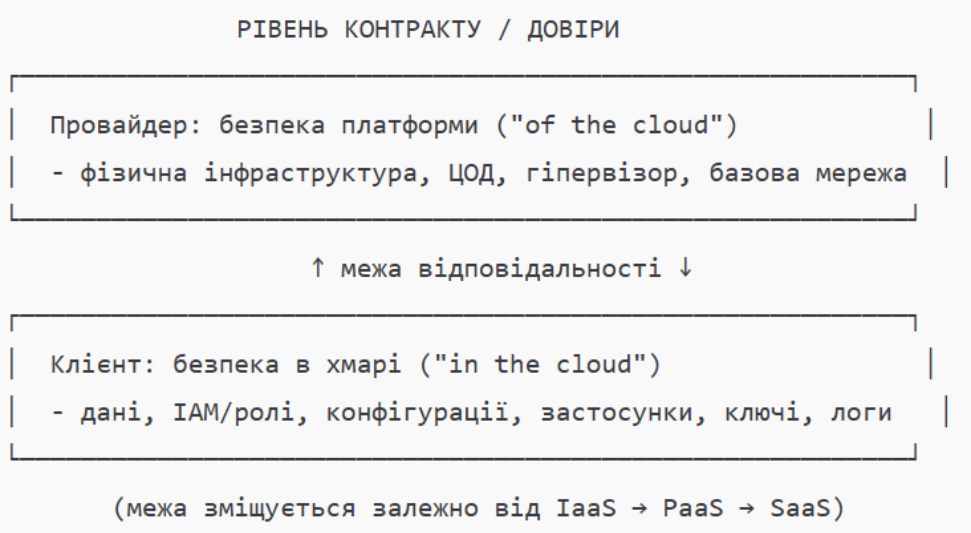

Рисунок 1.3 – Логіка спільної відповідальності (концептуальна схема)

Критично важливо, що в SaaS більшість технічних контролів забезпечує провайдер, але клієнт несе підвищену відповідальність за керування доступами, класифікацію даних, політики зберігання/експорту, а також за правильне використання функцій безпеки сервісу (MFA, DLP, журнали доступу). В IaaS навпаки: клієнт отримує більше контролю, але й більше операційної відповідальності (hardening ОС, патч-менеджмент, безпека мережевих правил). Ця залежність прямо описується в методичних матеріалах Azure щодо “shared responsibility” між on-prem / IaaS / PaaS / SaaS.

Таблиця 1.5 – Приклад розподілу відповідальності за моделями сервісу

Контроль / область	IaaS	PaaS	SaaS
Фізична безпека, ЦОД	Провайдер	Провайдер	Провайдер
Віртуалізація/платформа	Провайдер	Провайдер	Провайдер
ОС/патчинг ОС	Клієнт	Провайдер	Провайдер
Конфігурація сервісів	Клієнт	Клієнт (в межах сервісу)	Клієнт (налаштування)
IAM/ролі/MFA	Клієнт	Клієнт	Клієнт
Захист даних (класифікація, доступ)	Клієнт	Клієнт	Клієнт
Логування та моніторинг	Спільно	Спільно	Спільно

Для формування “довіри” ця модель означає: навіть якщо провайдер має сертифікації та гарантує безпеку платформи, інциденти часто виникають у зоні відповідальності клієнта (надмірні права, помилки конфігурації, витік ключів). Тому в роботі доцільно трактувати Shared Responsibility Model як базу для побудови матриці контролів та подальшого розмежування методів захисту даних і контролю активності вузлів. [11]

1.3.2 SLA як договірна “межа” надійності та якості

SLA (Service Level Agreement) – це контрактно закріплені параметри якості послуги: насамперед доступність (availability), інколи – продуктивність/латентність, підтримка, порядок компенсацій (service credits) та винятки. Концептуально SLA в хмарі стандартизується підходами на кшталт ISO/IEC 19086-1, що описує терміни, контексти та “будівельні блоки” хмарних SLA, наведено в таблиці 1.6.

Таблиця 1.6 – Типові показники SLA та як їх інтерпретувати

Показник SLA	Що означає	Навіщо замовнику
Uptime / Availability (%)	Частка часу, коли сервіс доступний	Оцінка ризику простою, планування DR/BCP
Визначення “downtime” і винятки	Які простої не рахуються (форс-мажор, дії клієнта тощо)	Реалістична оцінка відповідальності провайдера
Service credits	Компенсації при порушенні SLA	Фінансовий важіль, але не замінює технічної відмовостійкості
Підтримка (response time)	Час реакції залежно від критичності	Організація інцидент-менеджменту
Обмеження/квоти	Ліміти, що можуть вплинути на доступність	Запобігання “квотним” відмовам і DoS-by-quota

На рівні практики провайдери публікують SLA для конкретних сервісів. Наприклад, для Amazon EC2 наводиться зобов’язання щодо “Monthly Uptime Percentage” (за певних умов архітектури, зокрема розміщення в кількох зонах доступності).

Microsoft, зі свого боку, публікує актуальні та архівні редакції SLA для Online Services (включно з Azure).

SLA не гарантує безперервність бізнес-процесів, а лише встановлює юридичні зобов'язання щодо рівня сервісу. Тому для критичних систем SLA потрібно доповнювати архітектурними рішеннями (multi-AZ/region, резервні копії, RTO/RPO, механізми graceful degradation) та операційними процедурами (runbooks, тестування відновлення).

1.3.3 Комплаєнс: стандарти, регуляторика та докази виконання

Комплаєнс у хмарі охоплює дві площини: [12]

- Регуляторні вимоги (наприклад, персональні дані);
- Галузеві стандарти/аудитні практики (ISO/IEC, SOC 2, PCI DSS тощо).

Для персональних даних актуальними є вимоги GDPR (якщо обробляються дані резидентів ЄС) та національне законодавство (для України – Закон "Про захист персональних даних").

У GDPR важливою є концепція ролей controller/processor: контролер має залучати лише процесорів, які надають "достатні гарантії" щодо технічних та організаційних заходів – це безпосередньо впливає на вибір хмарного провайдера та вимоги до договорів (DPA, підпроцесори).

Серед стандартів і підтверджень практики найчастіше використовують:

- ISO/IEC 27017 (контролі безпеки для хмарних сервісів, орієнтовані і на провайдера, і на споживача).
- ISO/IEC 27018 (захист PII у публічній хмарі, коли провайдер виступає PII processor; актуальні редакції стандарту публікуються ISO).
- SOC 2 (аудиторський звіт щодо контролів сервісної організації за критеріями Trust Services: security, availability тощо).

- PCI DSS (базові технічні й операційні вимоги для захисту платіжних даних).

Практично комплаєнс у хмарі зводиться до зіставлення контролів: що покриває провайдер (через сертифікації/атестації/whitepapers), а що має реалізувати клієнт (політики доступу, шифрування, журнали, класифікація даних, реагування на інциденти), наведено в таблиці 1.7.

Таблиця 1.7 – “Карта доказів” комплаєнсу: що дає провайдер і що має довести клієнт

Вимога	Докази/артефакти провайдера	Докази/артефакти клієнта
Захист платформи	сертифікації/звітність (ISO/SOC), опис shared responsibility	конфігурації ресурсів, журнали змін, hardening, IAM-ревізії
Захист персональних даних	DPA, прозорість підпроцесорів, засоби шифрування/ключів	класифікація PII, мінімізація доступів, DLP, retention, реагування
Доступність сервісів	SLA конкретного сервісу, статус-сторінки	архітектура HA/DR, тестування відновлення, RTO/RPO

Підсумовуючи, модель довіри у хмарі будується не “на віру”, а на поєднанні: (а) формального розподілу відповідальності (Shared Responsibility), (б) вимірюваних контрактних зобов’язань (SLA), (в) підтверджених стандартами та аудитом контролів (комплаєнс). Саме ця тріада визначає, які методи захисту даних і контролю активності вузлів є обов’язковими у межах кваліфікаційної роботи. [14]

1.4 Основні підходи до захисту даних: шифрування, керування ключами, контроль доступу, резервування

Захист даних у хмарній інфраструктурі доцільно розглядати як комплекс взаємопов’язаних контролів, що працюють за принципом defense-in-depth: навіть якщо один бар’єр буде обійдено (наприклад, помилка конфігурації або компрометація облікових даних), інші механізми мають зменшити ймовірність витоку, підміни або втрати даних. У практиці управління

інформаційною безпекою базовий набір підходів зазвичай включає: шифрування (encryption), керування криптографічними ключами (key management), контроль доступу (access control / IAM) та резервування й відновлення (backup & recovery). Така комбінація напряму відповідає потребам тріади CIA (Confidentiality–Integrity–Availability) і узгоджується з підходами міжнародних стандартів управління контролями безпеки (наприклад, ISO/IEC 27001 та ISO/IEC 27002).

1.4.1 Шифрування даних: «у русі», «у спокої», «у використанні»

У хмарі дані зазвичай перебувають у трьох станах: дані в русі (data in transit), дані у спокої (data at rest) та дані у використанні (data in use). Для кожного стану застосовуються різні технічні підходи, а їх правильне поєднання суттєво знижує ризики несанкціонованого доступу та підміни.

Шифрування “у русі” реалізується переважно через TLS (або mTLS у сервісних взаємодіях). NIST надає практичні рекомендації щодо вибору й конфігурування TLS, підкреслюючи необхідність використання надійних версій протоколу та криптографічних наборів, наведено в таблиці 1.8. [14]

Таблиця 1.8 – Практичне застосування шифрування у хмарі

Стан даних	Ціль	Типові механізми	Типові помилки
In transit	Захист каналу, автентичність вузлів	TLS 1.2+/1.3, mTLS, VPN/Private Link	слабкі набори шифрів, відсутність перевірки сертифікатів, plaintext всередині мережі
At rest	Захист носіїв, снапшотів, бекупів	Disk/Object/DB encryption, application-level encryption	«за замовчуванням увімкнено» без контролю ключів, незашифровані резервні копії
In use	Захист під час обробки	TEE/конфіденційні VM (за потреби)	завищені очікування без моделі загроз і контролів доступу

Шифрування “у спокої” охоплює диски VM, об’єктні сховища, резервні копії, бази даних. Важливо, що шифрування на носії не замінює контроль доступу: воно зменшує шкоду при компрометації носія/снапшота/бекупу та при неправомірному доступі до фізичного середовища. Загальні принципи й типові архітектури storage encryption систематизовані в рекомендаціях NIST щодо технологій шифрування сховищ.

Шифрування “у використанні” (наприклад, конфіденційні обчислення) застосовується за підвищених вимог до захисту даних у пам’яті та під час обробки, але для більшості інформаційних систем базовими є перші два стани, доповнені коректним керуванням ключами та IAM.

1.4.2 Керування ключами: життєвий цикл, KMS/HSM, CMK/BYOK

Шифрування ефективне рівно настільки, наскільки захищені криптографічні ключі. NIST у рекомендаціях з key management описує ключові принципи: визначення ролей ключів, політики генерації/зберігання, ротацію, компрометацію та знищення ключового матеріалу.

Типова хмарна практика базується на envelope encryption: дані шифруються DEK (data encryption key), а DEK «обгортається» KEK (key encryption key), що зберігається в KMS/HSM. Це дає контрольовану ротацію KEK без необхідності масового перешифрування даних. [14]

```
(1) KMS генерує/зберігає KEK (root/CMK)
(2) Сервіс отримує DEK (GenerateDataKey) і шифрує дані
(3) DEK зберігається лише в RAM, а в БД/сховищі — лише "обгорнутий" DEK
Дані = Encrypt(DEK, plaintext)
Wrapped_DEK = Encrypt(KEK, DEK)
```

Рисунок 1.4 – Envelope encryption із використанням KMS

Таблиця 1.9 – Ключові практики керування ключами

Практика	Суть	Який ризик знижує
Централізація в KMS/HSM	Ключі не «розкидані» по застосунках/VM	витоки ключів із конфігів, репозиторіїв, образів
Принцип мінімальних привілеїв для ключів	окремі ролі на Encrypt/Decrypt/Rotate	зловживання доступом, ескалація привілеїв
Ротація й контроль версій	планова та позапланова (при інцидентах)	довготривала компрометація ключів
Розділення обов'язків (SoD)	адміністратор ключів ≠ адміністратор даних	внутрішні загрози, помилки персоналу
Аудит і журналювання операцій KMS	хто/коли/що шифрував або розшифровував	ускладнення приховування слідів, краща форензіка

Для реалізації такого підходу провайдери надають керовані сервіси KMS:

- AWS KMS: клієнтські (customer managed) ключі створюються і керуються у власному акаунті; доступ контролюється політиками та журналюється.
- Azure Key Vault / Key management: описуються підходи до централізованого керування ключами та використання customer-managed keys для шифрування даних у сервісах (наприклад, Azure Storage).
- Google Cloud KMS (CMEK): механізм customer-managed encryption keys надає клієнту контроль над ключами, що захищають дані at rest у сервісах Google Cloud. [15]

1.4.3 Контроль доступу: IAM, RBAC/ABAC, найменші привілеї

Контроль доступу – це «точка управління» тим, хто і за яких умов може читати/змінювати дані або конфігурувати ресурси. У хмарі контроль доступу реалізується через IAM, політики сервісів та механізми автентифікації. Каталог контролів NIST SP 800-53 містить детальні вимоги й практики щодо доступу, зокрема принцип least privilege.

На практиці виділяють:

- RBAC (рольова модель) – права призначаються ролям, ролі – суб’єктам (користувачам/сервісним акаунтам).
- ABAC/Policy-based – доступ визначається атрибутами (контекст, теги ресурсів, час, мережеве походження, рівень ризику тощо).

Таблиця 1.10 – Порівняння RBAC і ABAC у хмарній інфраструктурі

Критерій	RBAC	ABAC (policy-based)
Гнучкість	Середня	Висока (контекстні умови)
Зрозумілість адміністрування	Висока	Складніша через множинні умови
Типові сценарії	стандартні ролі (admin/devops/read-only)	доступ за тегами середовища (prod/dev), географією, ризиком, часом
Ризики	рольова «інфляція», надмірні права	помилки політик, складність тестування

Критично важливими є: MFA, контроль ідентичностей (особливо сервісних), регулярна ревізія ролей, а також ізоляція середовищ (dev/test/prod) через окремі облікові простори або жорсткі політики. [16]

1.4.4 Резервування та відновлення: RPO/RTO, незмінюваність, тестування

Шифрування та IAM знижують ризики витоку, але не гарантують відновлення після руйнування даних, помилок адміністратора або ransomware. Тому резервування та планування відновлення – обов’язковий елемент захисту, що напряму підтримує доступність і частково цілісність. NIST SP 800-34 формалізує підходи до contingency planning і відновлення сервісів після збоїв.

У хмарній практиці резервування варто проєктувати через показники:

- RPO (Recovery Point Objective) – допустима втрата даних у часі.
- RTO (Recovery Time Objective) – допустимий час простою.

Додатково рекомендовано застосовувати контролі на кшталт захисту даних відновлення на рівні політик і доступів (бекупи мають бути захищені не гірше за «бойові» дані), що відображено в підходах CIS Controls щодо відновлення даних, наведено в таблиці 1.11.

Таблиця 1.11 – Мінімальний набір вимог до резервування у хмарі

Вимога	Практичний зміст	Коментар
Розділення середовищ і доступів	окремі ролі на видалення/зміни бекупів	зменшує ризик саботажу та ransomware
Наявність незмінюваних копій	immutable/WORM, retention lock	унеможливлює масове видалення резервів
Географічна/зональна диверсифікація	multi-AZ/region копії для критичних даних	зменшує вплив регіональних інцидентів
Регулярне тестування відновлення	планові DR-тести, перевірка цілісності	без тестів «бекуп існує» ≠ «бекуп придатний»

Шифрування, керування ключами, контроль доступу та резервування формують «кістяк» захисту даних у хмарі. Шифрування без сильного key management створює ілюзію безпеки; IAM без журналювання та ревізій призводить до накопичення надмірних привілеїв; резервування без immutability та тестів може виявитися непридатним у критичний момент. Тому в наступних розділах доцільно розглядати ці підходи як єдину систему контролів із вимірюваними політиками й метриками.

1.5 Контроль активності вузлів і телеметрія: журнали подій, метрики, трасування, інцидент-менеджмент

Контроль активності вузлів у хмарній інфраструктурі (віртуальні машини, вузли Kubernetes, керовані сервіси, мережеві компоненти) базується на безперервному моніторингу та збиранні телеметрії, яка відображає фактичний стан системи й поведінку користувачів/процесів. З позиції управління ризиками ключовою є ідея continuous monitoring: організація має

постійно отримувати видимість щодо активів, загроз/уразливостей та ефективності застосованих контролів.

1.5.1 «Три сигнали» телеметрії та їх роль у контролі активності

OpenTelemetry визначає телеметричні сигнали так: traces відображають шлях запиту через систему, metrics – вимірювання, зафіксовані під час виконання, logs – записи про події.

Для задач контролю активності вузлів це означає: [17]

- Журнали подій (logs) дають детальну «слідову» інформацію: хто, коли, з якого джерела й що саме зробив (аутентифікація, зміни прав, запуск процесів, модифікації конфігурацій, помилки).
- Метрики (metrics) описують стан і ресурсний профіль (CPU, пам’ять, диски, мережа), а також сервісні показники (latency, error rate, throughput).
- Трасування (traces) дозволяє пов’язати події та затримки між компонентами в розподілених системах, що критично для хмарних застосунків із мікросервісною взаємодією.

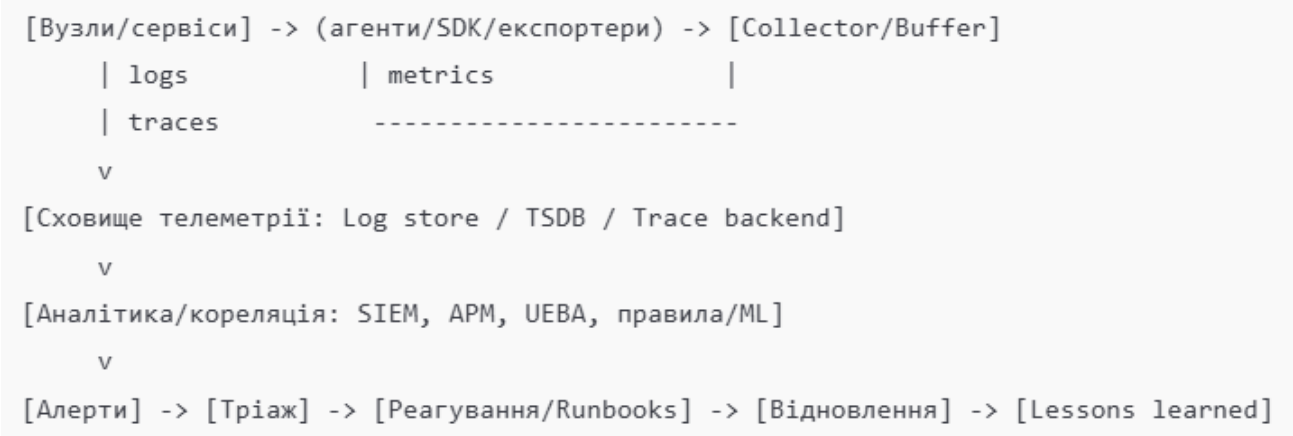

Рисунок 1.5 – Референсна схема потоку телеметрії та управління інцидентами

1.5.2 Журнали подій: джерела, вимоги до якості та безпеки логів

Журнали подій – основа доказовості та форензіки. NIST підкреслює, що ефективне лог-менеджмент середовище має охоплювати політики збирання, зберігання, аналізу, а також процеси підтримки й життєвого циклу логів.

В умовах хмари доцільно виділяти щонайменше такі категорії джерел:

- Control plane (аудитні журнали хмари) – події IAM, створення/видалення ресурсів, зміни політик, операції з ключами/секретами. Саме control plane є «єдиною точкою керування», тому журнали змін тут мають першочергову цінність для виявлення ескалації привілеїв або спроб приховати сліди (наприклад, вимкнення аудит-логів).
- Data plane журнали вузлів – системні (OS/kernel), сервісні (daemon/service), мережеві (flow logs, firewall/WAF), події контейнерного середовища (kube-audit, kubelet, runtime).
- Журнали застосунків – події бізнес-логіки, помилки, спроби доступу до чутливих операцій, security events на рівні застосунку (login, password reset, admin actions).

Таблиця 1.12 – Рекомендований мінімум категорій подій для контролю активності вузлів

Категорія подій	Приклади	Навіщо потрібно
Аутентифікація/сесії	login/logout, MFA, невдалі входи	виявлення brute force, компрометації акаунтів
Привілеї та ролі	зміни ролей, підвищення прав, використання admin-дій	контроль ескалації привілеїв
Зміни конфігурації	зміна політик, security groups, маршрутизації, CI/CD налаштувань	виявлення «тихих» змін, що відкривають периметр
Доступ до даних	читання/експорт/видалення, масові вибірки	контроль витоків та саботажу
Події вузла	запуск/зупинка сервісів, нові процеси, підозрілі бінарники	виявлення шкідливої активності на хості

Критичною є визначеність подій, що підлягають журналюванню. У каталозі контролів NIST SP 800-53 (контроль AU-2) як приклади подій наведено зміни паролів, невдалі входи, невдалі доступи, використання адміністративних привілеїв тощо – тобто події, що напряму відображають ризики компрометації та зловживань.

Окремий аспект – захист самих логів: вони містять чутливі дані (ідентифікатори, IP, інколи токени), тому мають передаватися захищеними каналами, зберігатися з контролем доступу та політиками ретенції. NIST SP 800-92 акцентує на необхідності організаційних політик і стандартизованих операційних процесів для збору/утримання/аналізу логів.

1.5.3 Метрики: моніторинг стану вузлів, SLI/SLO та якість алертингу

Метрики забезпечують «панель приладів» здоров'я системи й дозволяють будувати керовані пороги та SLO-орієнтований моніторинг. У практиках SRE (Google) підкреслюється, що моніторинг і алертинг мають фокусуватися на корисних сигналів, які варті «пейджу», а також на чітких сервісних цілях (SLO), що спираються на вимірювані індикатори (SLI).

Для контролю активності вузлів типово збирають: [17]

- ресурсні метрики (CPU, RAM, disk IO, filesystem usage, network throughput/packets);
- метрики платформи (кількість pod'ів, restarts, evictions, node conditions);
- метрики сервісу (latency p95/p99, error rate, request rate, saturation).

Важлива практична вимога – уникати «шумного алертингу» (too many alerts) і забезпечувати алерти з діагностичною цінністю: алерт має містити контекст (service, instance, region, версія релізу) і посилання на відповідні логи/трейси.

1.5.4 Трасування: наскрізна кореляція подій у розподілених системах

У хмарі типова транзакція проходить кілька сервісів (API gateway → auth → бізнес-сервіс → БД → черга → worker). Distributed tracing моделює це як набір span'ів у межах trace'у й дає можливість відповісти на питання «де саме виникла затримка/помилка» без ручного зіставлення журналів. OpenTelemetry підкреслює, що логи, метрики й трейси можуть корелюватися щонайменше за часом виконання, а також шляхом перенесення контексту (trace/span) між сигналами.

Практично це реалізується через:

- вставлення trace_id/span_id у структуровані логи (щоб із рядка логу перейти до відповідного trace);
- екземпляри (exemplars) у метриках, які пов'язують конкретний вимір із конкретним trace/span;
- уніфіковані атрибути ресурсу (service.name, host.name, cloud.region тощо) для всіх сигналів, наведено в таблиці 1.13.

Таблиця 1.13 – Порівняння логів, метрик і трасування для завдань безпеки та експлуатації

Сигнал	Сильні сторони	Обмеження	Типові задачі
Logs	деталізація «що сталося»; форензіка	високий обсяг; складність нормалізації	аудит доступів, розслідування інцидентів
Metrics	швидке виявлення відхилень; SLO-контроль	мало контексту про причину	алертинг, capacity, "health" вузлів
Traces	причинно-наслідковий ланцюг у мікросервісах	потребує інструментування	RCA, пошук «вузьких місць», кореляція з логами

1.5.5 Інцидент-менеджмент: від детекції до відновлення та покращень

Телеметрія має цінність лише тоді, коли вбудована в керований процес реагування. Актуальна редакція NIST SP 800-61r3 розглядає реагування на інциденти як частину кіберризик-менеджменту та узгоджує рекомендації з NIST CSF 2.0, акцентуючи на підготовці, детекції, реагуванні, відновленні та безперервному вдосконаленні.

У практичній реалізації інцидент-менеджмент зазвичай включає: [18]

- Підготовка (Prepare) – політики, ролі (SOC/чергові), інструменти (SIEM/APM), runbooks, навчання, тестування сценаріїв.
- Виявлення та аналіз (Detect/Analyze) – кореляція подій (SIEM), збагачення контекстом (asset inventory, threat intel), первинна класифікація за критичністю.
- Локалізація та стримування (Contain) – ізоляція вузла/облікового запису, блокування ключів, тимчасові мережеві політики.
- Усунення та відновлення (Eradicate/Recover) – видалення шкідливих артефактів, відкат конфігурацій (IaC), відновлення з резервних копій, перевірка цілісності.
- Післяінцидентні дії (Lessons learned) – RCA, корекція правил детекції, оновлення SLO/алертів, hardening.

Для побудови покриття телеметрією корисним є підхід MITRE ATT&CK, який систематизує джерела даних (data sources) для детекції технік супротивника й допомагає зіставити «які події треба бачити» з конкретними сценаріями зловмисної активності, наведено в таблиці 1.14.

Таблиця 1.14 – Які сигнали критичні на різних етапах інциденту

Етап	Пріоритетні сигнали	Приклад результату
Detect	метрики + security-логи + control plane аудит	швидке виявлення аномалії/компрометації
Analyze	логи + трейси (trace_id) + контекст активів	встановлення причини й масштабу
Contain	control plane логи + зміни політик + EDR події	підтвердження ізоляції/блокування
Recover	метрики SLO/SLI + логи відновлення	контроль стабілізації й відсутності рецидиву

Контроль активності вузлів у хмарі є по суті задачою керованої спостережуваності: систематичне збирання логів, метрик і трасування з кореляцією між сигналами та інтеграцією в процес інцидент-менеджменту. Поєднання рекомендацій NIST щодо log management і continuous monitoring із сучасними стандартами телеметрії (OpenTelemetry) та практиками SRE забезпечує як експлуатаційну надійність, так і безпекову керованість хмарної інфраструктури.

1.6 Висновки до першого розділу

У розділі 1 розглянуто базові теоретичні положення, які визначають підхід до захисту даних і контролю активності вузлів у хмарній інфраструктурі. Було пояснено, що хмарні обчислення надають ресурси як сервіс і дають еластичність, швидке масштабування та оплату за фактом використання, але при цьому ускладнюють безпеку через динамічність і багатошаровість середовища. Окремо описано моделі IaaS, PaaS і SaaS та показано, що зі зростанням рівня сервісу зменшується контроль над нижніми шарами інфраструктури, а роль керування доступами, конфігураціями та аудитом стає більш критичною. Також наведено уявлення про шарову архітектуру хмари, де контрольні механізми та безпека реалізуються через централізовану контрольну площину, а активність вузлів проявляється на compute-, storage- і network-рівнях.

Далі було розкрито, як у хмарі змінюється профіль загроз, і чому тріада конфіденційності, цілісності та доступності залишається основою оцінки ризиків. Показано, що суттєва частина інцидентів пов’язана з помилками конфігурації, компрометацією облікових даних і зловживанням привілеями, а також з атаками на контрольну площину, де зосереджено керування ресурсами. Окремо підкреслено, що порушення конфіденційності часто виникає через відкриті сховища та витік ключів або токенів, порушення цілісності пов’язане з несанкціонованими змінами даних і конфігурацій та атаками на ланцюг постачання, а порушення доступності може бути наслідком DDoS, виснаження квот, ransomware або відмов провайдера на рівні регіону чи сервісу.

У розділі також сформовано уявлення про модель довіри у хмарі. Було пояснено, що спільна відповідальність означає чіткий розподіл ролей між провайдером і клієнтом, і що реальна безпека залежить від того, як клієнт виконує свою частину контролів. Додатково показано, що SLA визначає юридично закріплені показники якості сервісу, але не замінює технічних рішень відмовостійкості та планів відновлення. Окремо розглянуто комплаєнс як набір регуляторних і стандартних вимог, які потребують не лише наявності контролів, а й доказів їх виконання у вигляді політик, журналів, аудитних артефактів і процедур.

Після цього узагальнено основні практичні підходи до захисту даних. Було показано, що шифрування має застосовуватися для даних у русі та у спокої, але його ефективність залежить від керування ключами, включно з контролем доступу до операцій шифрування і ротації ключів. Окремо обґрунтовано значення IAM як механізму, що визначає, хто і за яких умов має доступ до ресурсів і даних, та зазначено, що принцип найменших привілеїв і регулярна ревізія ролей є критично важливими. Також підкреслено, що резервування і відновлення є обов’язковими для забезпечення доступності й стійкості до помилок та атак, а ефективність резервного копіювання залежить

від незмінюваності копій, розділення доступів і регулярного тестування відновлення.

Завершальною частиною розділу стало обґрунтування контролю активності вузлів через телеметрію. Було пояснено, що спостережуваність у хмарі будується на поєднанні журналів подій, метрик і трасування, які разом дають повну картину стану системи та дій користувачів і сервісів. Показано, що журнали є основою аудиту та форензіки, метрики забезпечують оперативне виявлення відхилень і контроль SLO, а трасування дозволяє пов’язати події між компонентами розподіленої системи. Також наголошено, що телеметрія має практичну цінність лише тоді, коли вона інтегрована в інцидент-менеджмент із чіткими процедурами виявлення, аналізу, стримування, відновлення та подальшого вдосконалення контролів.

2 РОЗДІЛ МЕТОДИ ТА МОДЕЛІ РЕАЛІЗАЦІЇ ЗАХИСТУ ДАНИХ І МОНІТОРИНГУ АКТИВНОСТІ ВУЗЛІВ

2.1 Формалізація задачі: об’єкти захисту, загрози, вимоги та критерії ефективності

Формалізація задачі захисту даних і контролю активності вузлів у хмарній інфраструктурі потрібна для того, щоб перейти від загальних принципів безпеки до вимірюваної моделі: що саме ми захищаємо, від яких сценаріїв, якими засобами та як перевіряємо, що ці засоби справді працюють. У практиках ризик-менеджменту інформаційної безпеки базовою точкою відліку є оцінювання ризиків як комбінації ймовірності реалізації загрози та масштабу її наслідків (impact). Це узгоджується з підходом NIST до ризик-оцінювання та з рекомендаціями ISO/IEC щодо управління ризиками інформаційної безпеки. [19]

2.1.1 Межі системи та модель середовища

Розглянемо хмарну інфраструктуру як множину ресурсів і сервісів, керованих через контрольну площину (control plane), що впливає на площину даних (data plane). Для формалізації введемо три логічні компоненти:

- Середовище виконання (compute/storage/network), де обробляються дані та працюють вузли (VM, контейнерні ноди, керовані сервіси).
- Контрольна площина (IAM, API, політики, IaC, KMS, журнали керування), через яку створюються/змінюються ресурси.
- Підсистема спостережуваності та реагування (логування, метрики, трасування, SIEM/SOAR, процеси IR), що забезпечує видимість і кероване

реагування. Концепція безперервного моніторингу як основи підтримки рішень з управління ризиками підкреслюється в NIST SP 800-137.

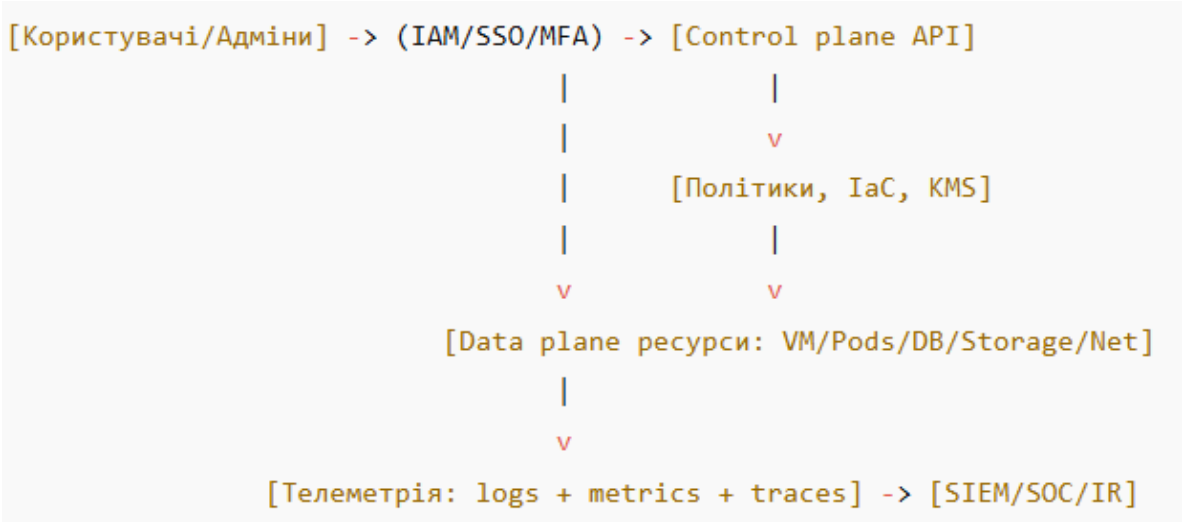

Рисунок 2.1 – Контекстна схема задачі захисту та контролю активності

На рисунку 2.1 наведено узагальнену контекстну схему задачі.

2.1.2 Об’єкти захисту

Об’єкти захисту в хмарі доцільно групувати не лише як “дані”, а як сукупність активів, компрометація яких призводить до порушення конфіденційності, цілісності або доступності (CIA). Каталог контролів безпеки NIST SP 800-53 прямо орієнтує на захист організаційних операцій і активів від широкого спектра загроз, що корисно для структурування об’єктів і відповідних контролів, наведено в таблиці 2.1. [20]

Таблиця 2.1 – Основні об’єкти захисту в хмарній інфраструктурі

Група об’єктів	Приклади	Типові наслідки компрометації
1	2	3
Дані (Data assets)	PII, комерційні дані, журнали, конфігурації, резервні копії	витік, підміна, втрата даних, штрафи/репутаційні втрати
Ідентичності та секрети	акаунти IAM, сервісні акаунти, токени, ключі API, секрети застосунків	ескалація привілеїв, захоплення контрольної площини

Продовження таблиці 2.1

1	2	3
Робочі навантаження (Workloads)	VM, контейнерні ноди, функції, образи контейнерів	запуск шкідливого коду, lateral movement, саботаж
Мережеві компоненти	security groups, FW/WAF, маршрути, VPN/Private Link	відкриття периметра, перехоплення/сканування, DDoS-ефект
Контрольна площина та політики	IaC, політики доступу, KMS-політики, налаштування аудит-логів	масова зміна ресурсів, “вимкнення видимості”, руйнування HA/DR

У межах кваліфікаційної роботи “об’єкт захисту” вважаємо активом, для якого встановлюються вимоги до конфіденційності C, цілісності I та доступності A, а також вимога аудитованості/спостережуваності O (можливість довести, що відбувалося в системі).

2.1.3 Загрози та модель порушника

Модель загроз будуємо як множину сценаріїв $T = \{t_1, \ldots, t_n\}$ що реалізуються через вразливості або помилки конфігурації та призводять до небажаних подій. Для практичної деталізації сценаріїв доцільно спиратися на бази знань про техніки супротивника, зокрема MITRE ATT&CK, де окремо виділяються джерела даних (data sources), необхідні для детекції технік. [21]

Таблиця 2.2 – Типові класи загроз у хмарі та їх фокус

Клас загроз	Приклади сценаріїв	Домінуючий вплив
Компрометація облікових даних	фішинг, витік ключів API, викрадення токенів сервісних акаунтів	C/I (через захоплення control plane), далі A
Помилки конфігурації	публічні сховища, відкриті порти, надмірні ролі	C (витік), I (підміна), A (саботаж)
Supply chain / CI/CD	підміна образів, шкідливі залежності, компрометація pipeline	I (бекдор), C (ексфільтрація), A (руйнування)
Зловживання привілеями / insider	несанкціоновані дії адміністратора, видалення логів/бекапів	I/A, погіршення форензіки
Атаки на доступність	DDoS, виснаження квот, ransomware	A (простій), далі I/C

Формально ризик для загрози t_i оцінюємо як:

$$R_i = P(t_i) \times Impact(t_i), \tag{2.1}$$

де $P(t_i)$ – ймовірність реалізації, а $Impact(t_i)$ – очікуваний збиток (фінансовий, операційний, юридичний). Такий підхід відповідає логіці оцінювання ризиків і визначення контрзаходів у методиках NIST та ISO/IEC.

2.1.4 Вимоги: безпека, експлуатаційність, комплаєнс

Вимоги сформуємо як множину $Req = \{req_1, \ldots, req_m\}$, поділену на функціональні (що система має робити) та нефункціональні (якими показниками якості/обмеженнями володіти). У хмарі вимоги мають покривати не лише технічні механізми, а й доказовість та керованість (auditability). Це узгоджується з ідеєю каталогів контролів (NIST SP 800-53) та з хмарно-специфічними настановами ISO/IEC 27017 для провайдерів і споживачів хмарних сервісів.

Таблиця 2.3 – Узагальнена матриця вимог (що має бути забезпечено)

Група вимог	Формулювання	Типові механізми реалізації
Конфіденційність даних	Дані не розкриваються неавторизованим суб'єктам	шифрування in transit/at rest, KMS, сегментація, DLP
Цілісність даних і конфігурацій	Дані/політики не можуть бути непомітно змінені	контроль змін (IaC), підпис артефактів, версіонування/immutability
Доступність сервісів	Критичні сервіси працюють у межах SLO/SLA	multi-AZ, DR, WAF/DDoS, квотний контроль, бекапи
Керування доступом	Доступ надається за Least Privilege і керовано	RBAC/ABAC, MFA/SSO, ревізії ролей, JIT-доступ
Спостережуваність та аудит	Події та дії фіксуються і доступні для аналізу	centralized logging, SIEM, кореляція, захист логів
Реагування на інциденти	Є процеси та засоби стримування і відновлення	runbooks, SOAR, IR-процедури (NIST SP 800-61r3)
Комплаєнс і доказовість	Є артефакти відповідності (політики/логи/аудит)	контроль ретенції, звіти, карти відповідальності

У частині аудиту доцільно конкретизувати “які події мають логуватися”. Наприклад, контроль AU-2 у NIST SP 800-53 описує необхідність визначення типів подій для журналювання (невдалі входи, використання адмінпривілеїв, зміни атрибутів безпеки тощо), що прямо підтримує формалізацію вимог до телеметрії.

Окремо виділяємо вимоги до SLA-показників і взаємозв’язку з SLO. ISO/IEC 19086-1 описує базові “будівельні блоки” для хмарних SLA (поняття, терміни, контексти), що корисно для формальної фіксації доступності, підтримки та винятків. Якщо система обробляє персональні дані, додаткові вимоги до обробки та захисту PII у публічних хмарах можуть формалізуватися через контрольні цілі ISO/IEC 27018. [2]

2.1.5 Критерії ефективності та метрики

Ефективність підсистеми захисту й контролю активності має вимірюватися не деклараціями (“впроваджено шифрування”), а показниками покриття, якості детекції та здатності відновлюватися. З огляду на continuous monitoring (NIST SP 800-137) і сучасну телеметрію, доцільно визначати критерії для кожного рівня: захист даних, керування доступом, спостережуваність, реагування.

Таблиця 2.4 – Критерії ефективності (приклади вимірюваних показників)

Напрям	Критерій	Приклад метрики/перевірки
1	2	3
Конфіденційність	покриття шифрування	частка сховищ/БД/бекапів із увімкненим шифруванням; частка сервісів із TLS 1.2+/1.3
Least Privilege	мінімізація привілеїв	% ролей, що пройшли ревізію; кількість “admin-like” ролей; % JIT-доступів
Аудитованість	повнота логів control plane	покриття подій створення/зміни ресурсів, IAM, KMS; незмінюваність і ретенція логів
Детекція	швидкість і точність виявлення	MTTD (mean time to detect), FP/FN rate для правил/алертів

Продовження таблиці 2.4

1	2	3
Реагування	керованість інцидентів	MTTR (mean time to respond/recover), частка інцидентів із runbook
Доступність	виконання SLO/SLA	uptime %, помилки p95/p99, відповідність RTO/RPO за тестами DR
Стійкість до саботажу	захищеність бекапів	наявність immutable копій; ізоляція доступу до видалення бекапів

Узагальнено задачу можна сформулювати як оптимізаційну: знайти набір контролів C (шифрування, IAM-політики, логування, резервування, процедури IR), який мінімізує сумарний очікуваний ризик $\sum R_i$ за обмежень на вартість, продуктивність і складність експлуатації:

$$\min_{C} \sum_{i=1}^{n} R_i(C) \quad \text{за умов} \quad Cost(C) \leq B, \;\; Performance(C) \geq P_{min} \tag{2.2}$$

Практично це означає, що в межах магістерської роботи доцільно визначити мінімальний, але достатній набір контролів та показників, які забезпечують прийнятний рівень ризику й одночасно дають вимірювані докази (логи, метрики, результати тестів відновлення) у разі аудиту або розслідування інциденту. При цьому рекомендації щодо інцидент-реагування як частини кіберризик-менеджменту доцільно узгоджувати з NIST SP 800-61r3. [23]

2.2 Методи захисту даних «у русі», «у спокої» та «у використанні» (TLS/mTLS, KMS/HSM, секрети, конфіденційні обчислення)

Захист даних у хмарній інфраструктурі доцільно будувати за принципом «повного життєвого циклу даних», коли контролі охоплюють три стани: дані у русі (передаються мережею), у спокої (зберігаються на дисках/у сховищах/БД) та у використанні (обробляються в пам’яті та на CPU). Такий

поділ важливий, бо кожен стан має свою поверхню атак і потребує різних технічних механізмів: для «у русі» ключовим є TLS/mTLS, для «у спокої» – шифрування із керуванням ключами (KMS/HSM), для «у використанні» – апаратно ізольовані середовища (TEE) та механізми віддаленої атестації.

```
Data in transit  -> TLS/mTLS, VPN/Private Link, cert management
Data at rest     -> Storage/DB encryption + KMS/HSM + rotation + audit
Data in use      -> TEE (confidential VMs/enclaves) + attestation + key release policy
```

Рисунок 2.2 – «Три стани» даних та відповідні класи контролів

У рекомендаціях NIST щодо TLS акцентується, що TLS є базовим механізмом захисту даних під час електронного передавання, а правильна конфігурація протоколу та криптографічних наборів критично впливає на стійкість до перехоплення й підміни.

2.2.1 Захист даних «у русі»: TLS і mTLS як стандартний транспортний контроль

TLS (Transport Layer Security) забезпечує конфіденційність і цілісність трафіку між клієнтом і сервером, а також автентичність сторін через сертифікати (PKI). У специфікації TLS 1.3 наголошується, що протокол призначений для запобігання підслуховуванню та підміні повідомлень і вводить сучасні механізми узгодження ключів та захисту рукопотискань.

У хмарній інфраструктурі TLS має бути обов’язковим для: [24]

- зовнішніх API та веб-додатків (периметр: LB/WAF/API Gateway);
- сервіс-сервіс взаємодій (east-west трафік у VPC/VNet, Kubernetes, mesh);
- адміністративних каналів (панелі керування, доступ до керованих сервісів).

mTLS (mutual TLS) доповнює TLS двосторонньою автентифікацією: не лише клієнт перевіряє сертифікат сервера, а й сервер перевіряє сертифікат клієнта. На практиці це дає сильну ідентифікацію сервісів у моделі Zero Trust, коли мережа не вважається «довіреною» навіть всередині приватного сегмента. Підхід із вимогами до вибору та конфігурації TLS (версії протоколу, криптонабори, перевірка сертифікатів) систематизовано в NIST SP 800-52 Rev.2.

Критично важливі практики для TLS/mTLS у хмарі:

- мінімізація «plaintext-вікон» (TLS termination має бути контрольованим: де завершується TLS і хто бачить незашифровані дані);
- сувора перевірка сертифікатів (chain validation, hostname verification, контроль довіри до CA);
- централізоване керування сертифікатами (видача, ротація, відкликання), щоб уникнути прострочених або «ручних» сертифікатів;
- журналювання подій TLS (версії, помилки handshake, відхилені сертифікати клієнта) як частина контролю активності.

Таблиця 2.5 – TLS vs mTLS у типовій хмарній взаємодії

Критерій	TLS	mTLS
Автентифікація	Сервер автентичний для клієнта	Взаємна автентифікація клієнт↔сервер
Основний сценарій	публічні веб/API, доступ користувачів	сервіс-сервіс, Zero Trust, service mesh
Переваги	простіше впровадження, стандарт де-факто	сильна ідентичність сервісів, менше залежності від мережевих ACL
Ризики/виклики	TLS termination, помилки конфігурації	PKI-операції, ротація клієнтських сертифікатів, масштабованість

2.2.2 Захист даних «у спокої»: шифрування сховищ і баз даних через KMS/HSM

Шифрування «у спокої» застосовується до дисків VM, об’єктних сховищ, керованих БД, снапшотів і резервних копій. Його мета – зменшити

наслідки компрометації носіїв, витоку резервів або несанкціонованого доступу до фізичного рівня. Проте практична ефективність такого шифрування прямо залежить від того, як керуються ключі: хто має право на операції decrypt, як виконується ротація, як ведеться аудит.

NIST у SP 800-57 надає загальні принципи керування ключовим матеріалом: визначення ролей, вимоги до захисту ключів, життєвий цикл (генерація, розповсюдження, зберігання, ротація, компрометація, знищення).

У хмарних KMS-підходах типовим є envelope encryption: дані шифруються одноразовим ключем даних (DEK), а DEK шифрується (wrap) ключем вищого рівня (KEK), який зберігається в KMS/HSM. AWS прямо описує envelope encryption як практику, що спрощує захист ключів і контроль доступу до операцій розшифрування.

HSM (Hardware Security Module) у цій архітектурі використовується як апаратний корінь довіри: ключі не покидають захищений модуль, а криптооперації виконуються всередині нього. Для регульованих середовищ важливим критерієм є відповідність криптомодулів стандартам на кшталт FIPS 140-3 та наявність валідації в межах CMVP. [25]

Практичні варіанти керування ключами в хмарі:

- Provider-managed keys: ключами керує провайдер (мінімальні операційні витрати, але менше контролю).
- Customer-managed keys (CMK/CMEK): ключ у KMS належить клієнту, а доступи і ротація контролюються клієнтом (кращий контроль і аудит).
- BYOK/HYOK (у певних сервісах): ключі імпортуються або зберігаються у власних HSM/поза хмарою (вищі комплаєнс-вимоги, але більша складність).
- Microsoft описує, що Azure Key Vault є захищеним сховищем секретів/ключів/сертифікатів і підтримує сценарії моніторингу доступів (логування), а також опирається на HSM-підкріплені механізми захисту.

Таблиця 2.6 – Мінімальні вимоги до KMS/HSM-контурів

Вимога	Зміст у хмарі	Навіщо потрібно
Централізація ключів	ключі та політики доступу керуються в KMS/Key Vault	зменшення витоків ключів із VM/репозиторіїв
Розмежування ролей	адміністратор ключів ≠ адміністратор даних	зниження insider-ризиків, контроль SoD
Ротація й версійність	планова ротація, аварійна при інцидентах	обмеження часу компрометації
Аудит операцій	логування encrypt/decrypt/rotate/disable	форензіка та комплаєнс-доказовість
Політики “key release”	видача доступу до decrypt тільки за умов (контекст/атестація)	зв’язок із data-in-use (confidential computing)

2.2.3 Секрети: зберігання, доступ і ротація (Secrets Manager/Key Vault/Secret Manager)

Окремим класом чутливих даних є секрети: паролі, токени, API-ключі, рядки підключення, сертифікати, приватні ключі підпису, ключі шифрування застосунку. Проблема секретів у хмарі зазвичай виникає не через «слабку криптографію», а через операційні помилки: секрети потрапляють у репозиторії, виводяться в логи, «вшиваються» в образи контейнерів або живуть роками без ротації.

Практика полягає в тому, щоб:

- зберігати секрети в керованому секрет-сховищі;
- видавати їх застосункам за принципом найменших привілеїв;
- ротуувати за графіком та/або подіями ризику.

AWS описує ротацію секретів як процес періодичного оновлення значення секрету з одночасною синхронізацією облікових даних у цільовому сервісі; підтримується автоматична ротація.

Google Secret Manager прямо визначає ротацію як заміну чутливої інформації (паролі, ключі, токени) для зниження ризику компрометації та підтримки вимог комплаєнсу, а також підтримує планування ротацій.

Azure Key Vault окремо виділяє об’єкти «keys, secrets, certificates» і надає керування ними як керований сервіс.

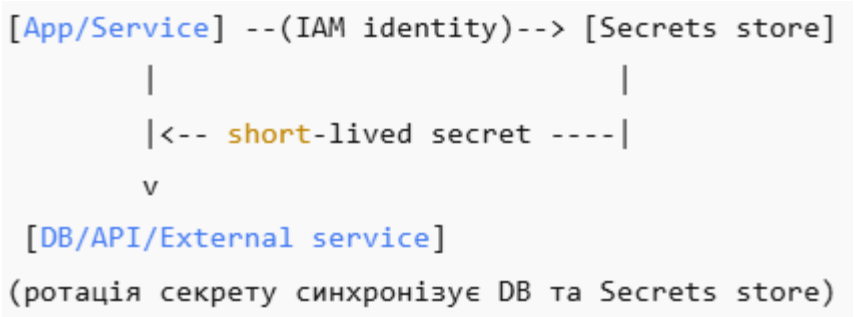

Рисунок 2.3 – Типовий патерн використання секретів у застосунку

Окремо варто виділяти динамічні секрети (JIT-облікові дані з TTL), коли секрет створюється «під запит» і автоматично відкликається після завершення терміну дії. Цей підхід суттєво зменшує вікно компрометації й підтримується у практиках сучасних секрет-менеджерів. [26]

2.2.4 Захист даних «у використанні»: конфіденційні обчислення (TEE, атестація, ізоляція)

Навіть при наявності TLS і шифрування «у спокої» залишається складна зона: під час обробки дані традиційно розшифровуються в пам’яті процесу і стають потенційно доступними для гіпервізора, адміністраторів хоста або шкідливого ПЗ з підвищеними привілеями. Саме для цього застосовуються конфіденційні обчислення (confidential computing) – захист даних in-use за допомогою апаратно ізольованих середовищ виконання (TEE). Google визначає confidential computing як захист даних «у використанні» за рахунок апаратного TEE, що запобігає несанкціонованому доступу або модифікації застосунків і даних під час виконання.

Microsoft також описує TEE як сегреговану область пам’яті та CPU, захищену від решти середовища шифруванням, де сторонній код не може читати або змінювати дані всередині TEE.

Поширені реалізації у хмарі:

- Confidential VMs на базі технологій на кшталт AMD SEV-SNP або Intel TDX/SGX (ізоляція пам’яті, захист від гіпервізора).
- Enclaves (наприклад, Nitro Enclaves) як ізольовані середовища всередині хост-інстансу для обробки найбільш чутливих даних (токенізація, підпис, розшифрування). AWS позиціонує Nitro Enclaves як засіб створення ізольованих compute-середовищ для захисту та безпечної обробки високочутливих даних.

Таблиця 2.7 – Порівняння підходів захисту in-use

Підхід	Що захищає	Типові сценарії	Обмеження
Звичайні VM/контейнери + hardening	дані в пам’яті не ізольовані від привілейованого рівня	стандартні бізнес-застосунки	не закриває «host/hypervisor-level» загрози
Confidential VM (TEE)	пам’ять/виконання ізольовані апаратно	обробка PII, secure analytics, ML на чутливих даних	сумісність типів VM, вимоги до конфігурації, можливий overhead
Enclave (ізольований домен)	вузький контур для криптоопераций	підпис, токенізація, дешифрування, key custody	складніша інтеграція застосунку, окремий дизайн потоку даних

Ключовим елементом є атестація: можливість криптографічно перевірити, що код запущений у справжньому TEE з очікуваною конфігурацією. На практиці це дозволяє побудувати політику «видачі ключа лише після атестації»: KMS або секрет-сховище розкриває ключ/секрет тільки тоді, коли отримує доказ, що запит походить із довіреного TEE. Це особливо важливо для сценаріїв, де загроза включає адміністраторів інфраструктури або компрометацію гіпервізора.

На практиці confidential computing не замінює TLS і KMS, а «закриває прогалину» між ними: TLS захищає канал, KMS – зберігання, а TEE – безпечну обробку даних і контрольоване розкриття ключів/секретів у момент виконання. Саме тому у сучасних хмарних підходах логічно будувати зв’язаний ланцюг: mTLS для сервісної ідентичності, KMS/HSM для керування

ключами та аудитованих криптоопераций, secrets manager для секретів із ротацією, TEE для in-use захисту і політик «key release on attestation».

2.3. Механізми контролю доступу: RBAC/ABAC, MFA, Zero Trust, політики IAM

Контроль доступу в хмарній інфраструктурі є центральним механізмом зменшення ризиків, пов’язаних із компрометацією облікових даних, помилками конфігурації та зловживанням привілеями. На відміну від класичного периметрового підходу, у хмарі більшість адміністративних дій виконується через API та “control plane”, тому правильне проєктування IAM-політик і моделей авторизації прямо визначає рівень безпеки всіх інших контролів. Базовими принципами є “deny by default”, мінімальні привілеї (least privilege) і розділення обов’язків (separation of duties), що закріплено в каталозі контролів NIST SP 800-53 (зокрема, концептуально в контролі AC-6). [27]

2.3.1 RBAC і ABAC як базові моделі авторизації в хмарі

RBAC (Role-Based Access Control) задає доступ через ролі: користувачу/сервісному акаунту призначають роль, а роль містить набір дозволених дій над ресурсами. Перевага RBAC – керованість і зрозумілість у великих організаціях, де типові функції (адміністратор, оператор, аудитор, розробник) повторюються. Недолік – “інфляція ролей”: з часом з’являється багато винятків, ролі розростаються, а права стають надмірними.

ABAC (Attribute-Based Access Control) визначає доступ на основі атрибутів суб’єкта, об’єкта, операції й контексту середовища. NIST SP 800-162 формалізує ABAC як методологію, де рішення “дозволити/заборонити” приймається шляхом оцінювання атрибутів і правил політики.

У хмарі ABAC найчастіше реалізується через теги ресурсів і умовні вирази (conditions): наприклад, “дозволити читання об’єктів лише з тегом Environment=Dev” або “дозволити доступ лише з керованого пристрою чи з корпоративної мережі”.

Таблиця 2.8 – Порівняння RBAC та ABAC для хмарного IAM

Критерій	RBAC	ABAC
Одиниця керування	роль (набір дозволів)	політика + атрибути + умови
Масштабованість	висока для типових посад	висока для різноманітних ресурсів і середовищ
Гнучкість	середня (винятки ускладнюють)	висока (контекст, теги, час, місце, ризик)
Основні ризики	надмірні ролі, “role sprawl”	складність тестування політик, помилки умов
Типовий best practice	мінімальні ролі + SoD + ревізії	стандартні шаблони умов + централізовані атрибути

Провайдери реалізують ABAC по-різному, але ідея спільна:

- AWS трактує атрибути як tags і описує ABAC-політики, де операція дозволяється, якщо тег принципала збігається з тегом ресурсу.
- Google Cloud надає IAM Conditions як механізм умовної (attribute-based) авторизації.
- Azure розширює RBAC умовами призначення ролей (role assignment conditions) як елемент Azure ABAC, наприклад перевіряючи теги об’єкта.

Практично оптимальним для хмари є гібрид: RBAC як “каркас” (хто в принципі може працювати з класом ресурсів) + ABAC як “фільтр” (за яких умов і лише з якими ресурсами).

2.3.2 Політики IAM і логіка прийняття рішення “Allow/Deny”

IAM у хмарі функціонує як централізований механізм авторизації для control plane та (частково) data plane.

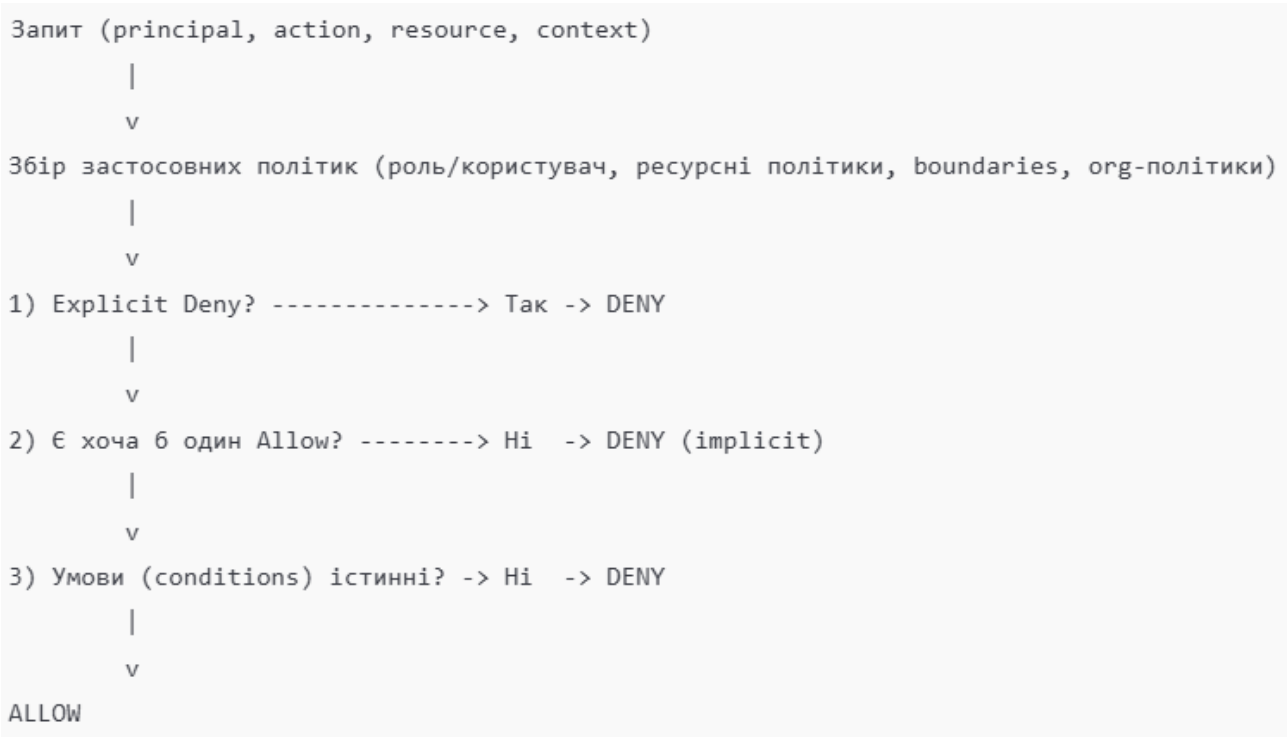

Рисунок 2.4 – Узагальнений потік оцінювання запиту доступу в IAM

Ключовий момент – алгоритм оцінювання політик: у більшості платформ діє принцип “implicit deny” (якщо явно не дозволено – заборонено) та пріоритет “explicit deny” (явна заборона перекриває дозволи). На прикладі AWS детально описано, як обробляється контекст запиту, які політики застосовуються й як формується фінальне рішення дозволу або відмови.

Для зниження ризиків у хмарі політики IAM варто будувати не лише навколо “хто що може”, а й навколо обмежень контексту: дозволяти адміністративні дії тільки з керованих пристроїв, із визначених мереж, у межах часових вікон, або лише після проходження MFA/посиленої автентифікації. Це логічно узгоджується з ідентифікаційно-керованим підходом у Zero Trust (див. нижче) і технологічно підтримується умовами (conditions) у хмарних IAM. [28]

2.3.3 MFA як контроль проти компрометації облікових даних

Компрометація облікових даних є одним із найчастіших шляхів захоплення хмарних ресурсів, оскільки IAM-ключі, токени та SSO-сесії відкривають доступ до control plane. MFA (Multi-Factor Authentication) зменшує ймовірність несанкціонованого входу за рахунок вимоги щонайменше двох різних факторів (знання/володіння/біометрія). NIST SP 800-63B-4 визначає технічні вимоги до автентифікації та рівнів AAL (Authenticator Assurance Level), включно з вимогами до багатофакторної автентифікації й керування автентифікаторами.

У сучасних хмарних середовищах MFA має охоплювати:

- інтерактивні входи адміністраторів і операторів;
- SSO до консолі та керованих сервісів;
- привілейовані операції (step-up authentication), коли для "небезпечних" дій потрібна сильніша автентифікація.

Як приклад реалізації "policy-driven MFA", Microsoft Entra Conditional Access позиціонується як рушій Zero Trust-політик, що враховує сигнали (ризик входу, стан пристрою, локацію) і може примусово вимагати MFA для доступу.

Додатково, Google Cloud документує поетапне впровадження вимоги MFA для клієнтів, підкреслюючи роль MFA як базового захисту ідентичності.

Таблиця 2.9 – Практична оцінка методів MFA в хмарі

Метод MFA	Стійкість до фішингу	Операційні особливості	Типове застосування
1	2	3	4
OTP (додаток-генератор)	середня	залежить від захисту пристрою	базовий MFA для персоналу
Push-підтвердження	середня	ризик "push fatigue", потрібні політики	масовий корпоративний сценарій

1	2	3	4
FIDO2/WebAuthn/Passkeys	висока	потребує розгортання та підтримки	адміністратори, критичні системи
SMS-код	нижча	вразливості SS7/SIM-swap	як тимчасовий/резервний варіант

2.3.4 Zero Trust як модель “безперервної перевірки” доступу

Zero Trust переносить акцент із мережевого периметра на користувача, пристрій, застосунок і ресурс. NIST SP 800-207 визначає Zero Trust як набір парадигм, що зміщують оборону від статичних периметрів до фокусу на користувачах, активах і ресурсах, а також описує логічні компоненти Zero Trust Architecture (ZTA).

У практичній інтерпретації це означає: “ніколи не довіряй за замовчуванням, завжди перевіряй”, мінімальні привілеї, мікросегментація та постійний моніторинг/оцінка ризику.

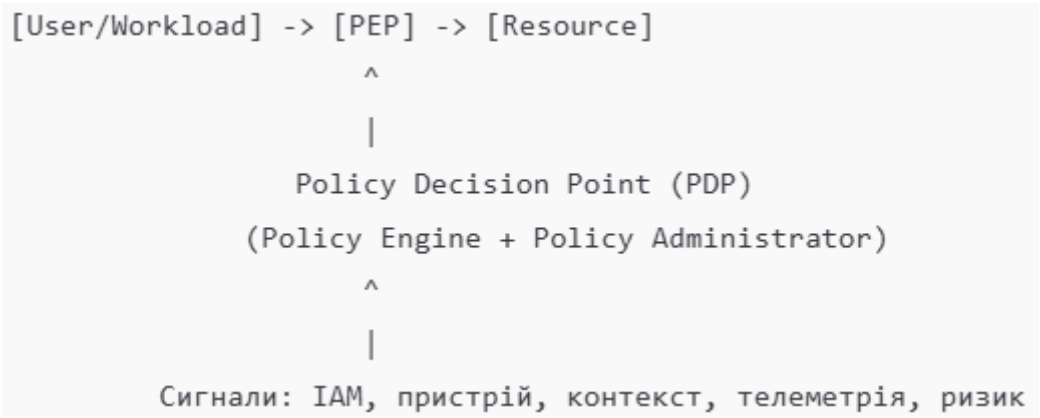

Рисунок 2.5 – Логічні компоненти ZTA (узагальнено за NIST SP 800-207)

У хмарі Zero Trust реалізується через комбінацію IAM-політик (RBAC/ABAC), сильну автентифікацію (MFA/step-up), контекстні рішення доступу (Conditional Access), а також технічні механізми на рівні сервісів (mTLS для сервіс-ідентичності, приватні endpoints, мікросегментація). Критично, що Zero Trust не заперечує RBAC чи ABAC, а задає режим їх

застосування: доступ надається динамічно, на найкоротший потрібний час і з урахуванням поточного ризику. [29]

2.3.5 Типові політики IAM для хмарної інфраструктури

Щоб IAM був не декларативним документом, а працюючим механізмом, політики варто структурувати як набір повторюваних “шаблонів” із вбудованими умовами та аудитом. Практично корисними є такі підходи:

- Розділення адміністративних доменів: окремі ролі для керування мережею, даними, ключами, CI/CD. Це зменшує blast radius при компрометації одного акаунта (узгоджується з AC-6 least privilege).
- ABAC-умови для обмеження доступу до ресурсів за тегами середовища/проєкту/власника (AWS tags-ABAC, GCP IAM Conditions, Azure role assignment conditions).

Політики оцінювання та пріоритету deny: явні заборони для небезпечних операцій (наприклад, вимкнення аудит-логів), щоб знизити ризик приховування слідів. Логіка оцінювання політик і пріоритети описані на прикладі AWS.

MFA-gating для привілейованих дій, що узгоджується з вимогами NIST до багатофакторної автентифікації та рівнів AAL.

Таблиця 2.10 – Приклади IAM-політик і контрольних подій аудиту

Політика/контроль	Приклад практичного правила	Які події журналювати
Least privilege	лише потрібні actions для конкретної ролі	усі deny/allow на критичні API
ABAC за тегами	доступ тільки до ресурсів з Project=X	зміни тегів і політик
MFA/step-up	“адмін-операції лише після MFA”	MFA-події, невдалі входи
Заборона вимкнення аудиту	explicit deny на “disable logging”	спроби змінити/вимкнути логування
Обмеження контексту	доступ лише з trusted location/device	зміни умов, ризикові входи

Механізми RBAC/ABAC, MFA, Zero Trust і IAM-політики потрібно розглядати як єдину систему. RBAC забезпечує структурованість і керованість прав, ABAC додає контекст і точність, MFA знижує ризик компрометації ідентичностей, а Zero Trust задає принципи безперервної перевірки та мінімізації довіри. Для хмарної інфраструктури це означає, що ефективний контроль доступу – це не одноразове “налаштувати ролі”, а постійний цикл: нормалізація атрибутів, ревізія ролей, посилення автентифікації, тестування політик і аудит control plane.

2.4 Методи контролю активності вузлів: агентний/безагентний моніторинг, збір логів, EDR/XDR, SIEM/SOAR

Контроль активності вузлів у хмарній інфраструктурі слід розуміти як безперервне отримання та аналіз телеметрії про стан і поведінку обчислювальних вузлів (VM, ноди Kubernetes, керовані середовища виконання), мережевих компонентів і сервісів з метою своєчасного виявлення відхилень, компрометацій та помилок конфігурації. Такий підхід відповідає концепції continuous monitoring, де організація підтримує “постійну обізнаність” щодо активів, загроз і ефективності контролів.

Особливість хмари полягає в тому, що значна частина критичних подій відбувається в control plane (операції через API/консоль), а вузли можуть бути короткоживучими та автоматично замінюваними, тому контроль активності має поєднувати безагентні “платформні” джерела, агентну телеметрію з хостів і спеціалізовані засоби детекції/реагування. [30]

2.4.1 Агентний та безагентний моніторинг: принципи і сфери застосування

Агентний підхід передбачає встановлення на вузол програмного агента (monitoring/log/EDR agent), який збирає системні події, процеси, мережеві з'єднання, журнали ОС/контейнерного рантайму, метрики продуктивності й передає їх у централізоване сховище або систему аналітики. Перевагою є глибина видимості (process-level та host-level), можливість активних дій (ізоляція, карантин, блокування) і краща придатність до форензіки. Недоліки – операційні витрати на розгортання/оновлення, ризики сумісності, вплив на продуктивність, а також необхідність захищати самого агента як компонент довіри.

Безагентний підхід базується на телеметрії платформи (audit/platform logs, API events, flow logs), інтеграціях із гіпервізором/провайдером або скануванні "ззовні", без інсталяції ПЗ на вузол. У практиці cloud security цей підхід широко використовується для інвентаризації, оцінки конфігурацій, вразливостей і базової видимості. Наприклад, Microsoft описує agentless machine scanning як механізм, що не потребує встановлення агентів і не впливає на продуктивність машини.

Обмеження безагентного підходу – менша деталізація подій усередині хоста (процеси/команди), залежність від якості й повноти платформних логів та відмінності реалізацій у різних провайдерів.

Таблиця 2.11 – Порівняння агентного та безагентного контролю активності

Критерій	Агентний моніторинг	Безагентний моніторинг
1	2	3
Глибина видимості	висока (процеси, системні події, деталі ОС)	середня (control plane, конфігурації, частина мережевих подій)
Вплив на вузол	можливий (CPU/RAM/IO)	мінімальний або відсутній

Продовження таблиці 2.11

1	2	3
Операційні витрати	вищі (розгортання, оновлення, політики)	нижчі (інтеграції/налаштування сервісів)
Стійкість при “коротких” інстансах	потребує автоматизації (golden images, DaemonSet)	природно підходить (платформні логи зберігаються незалежно)
Типові сценарії	EDR, форензіка, контроль процесів, глибокий runtime-моніторинг	аудит API, комплаєнс, виявлення misconfiguration, базова активність

На практиці доцільно застосовувати комбіновану стратегію: безагентні джерела забезпечують обов’язковий контроль control plane і конфігурацій, а агентні – високодетальні сигнали з критичних вузлів і середовищ виконання.

2.4.2 Збір логів і спостережуваність: “три сигнали” та якість журналювання

Ефективний контроль активності неможливий без централізованого log management. NIST SP 800-92 підкреслює, що журнали потрібні для моніторингу активності, розслідування інцидентів і підтримки безпекових програм, а організація має визначити політики збору, зберігання та аналізу логів.

У сучасних системах доцільно будувати телеметрію навколо “трьох сигналів”: logs, metrics, traces. OpenTelemetry визначає ці сигнали як журнали подій, вимірювання під час виконання та шлях запиту через систему, а також описує кореляцію між ними через спільний контекст (resource/trace/baggage).

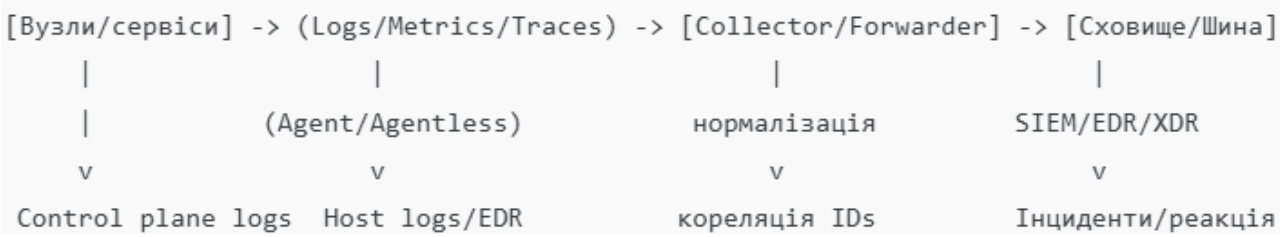

Рисунок 2.6 – Узагальнений конвеєр телеметрії для контролю активності

Джерела “control plane” є критичними, оскільки відображають адміністративні дії та зміни конфігурацій. Для прикладу, Azure Activity Log описується як платформний лог подій control plane (створення/модифікація ресурсів, помилки деплою).

Аналогічно, Google Cloud Audit Logs розділяє журнали на Admin Activity, Data Access, Policy Denied та System Event, що дає основу для аудиту “хто/що/де/коли”.

У AWS роль системи аудиту API виконує CloudTrail, який фіксує дії користувачів/ролей/сервісів як події, забезпечуючи доказовість керування ресурсами.

Таблиця 2.12 – Мінімальний набір джерел логів для контролю активності вузлів у хмарі

Категорія	Приклади джерел	Які інциденти виявляє
Control plane audit	Activity Log / Audit Logs / CloudTrail	ескалація прав, зміни політик, створення ключів, вимкнення журналювання
Хостові журнали	OS (auth, syslog), процеси, служби	підозрілі запуски, persistence, lateral movement на хості
Контейнер/K8s	kube-audit, kubelet, runtime events	зловживання RBAC у кластері, запуск привілейованих контейнерів
Мережеві журнали	flow logs, WAF/FW, DNS	сканування, ексфільтрація, C2, аномальні потоки
Застосункові логи	auth, адмін-дії, помилки бізнес-операцій	атаки на акаунти, зловживання функціями, підозрілі транзакції

При проєктуванні збору логів важливо забезпечити:

- повноту (щоб уникати “сліпих зон”, зокрема в control plane);
- цілісність (неможливість непомітного підчищення/підміни);
- часову синхронізацію (коректна кореляція подій);
- контроль доступу до логів (логи самі по собі є чутливими даними).

Ці вимоги узгоджуються з практичними рекомендаціями NIST щодо log management.

2.4.3 EDR та XDR як засоби детекції і реагування на рівні вузлів та “поверхонь атаки”

EDR (Endpoint Detection and Response) фокусується на кінцевих точках (сервери, робочі станції, VM, інколи контейнери) та забезпечує безперервний збір подій, детекцію підозрілої поведінки і дії реагування. Узагальнено EDR описується як підхід, що моніторить endpoints на ознаки компрометації та надає інструменти для розслідування й реагування.

У хмарі EDR особливо цінний там, де потрібні хостові деталі: дерево процесів, командні рядки, модулі, зв’язки процес-мережа, події доступу до файлів і пам’яті.

XDR (Extended Detection and Response) розширює детекцію/реагування за межі endpoint, поєднуючи сигнали з кількох доменів (ідентичності, пошта, застосунки, хмара) і забезпечуючи корельовану картину атаки. Microsoft визначає Defender XDR як платформу, що координує detection, investigation та response “через endpoints, identities, email, applications”.

Практично це знижує час на тріаж інцидентів і покращує якість кореляції, особливо коли атака починається з фішингу або компрометації ідентичності, а далі переходить у хмарні ресурси. [31]

2.4.4 SIEM та SOAR: централізована кореляція подій і автоматизація реагування

SIEM (Security Information and Event Management) використовується для централізованого збору, нормалізації та кореляції подій з різних джерел (control plane, хости, мережа, застосунки). SIEM є “точкою збірки” доказів і контексту: інвентар активів, правила кореляції, аналітика, пошук, звітність, комплаєнс. У хмарі SIEM має пріоритетно індексувати audit-події провайдера, адже атаки часто спрямовані на зміну конфігурацій і вимкнення видимості.

SOAR (Security Orchestration, Automation and Response) додає оркестрацію та автоматизацію дій реагування: playbooks/runbooks, тригери, інтеграції з IAM, EDR, ticketing, messaging. Microsoft Sentinel прямо описує SOAR-можливості через automation rules та playbooks, які запускаються у відповідь на alerts/incidents для підвищення ефективності SOC.

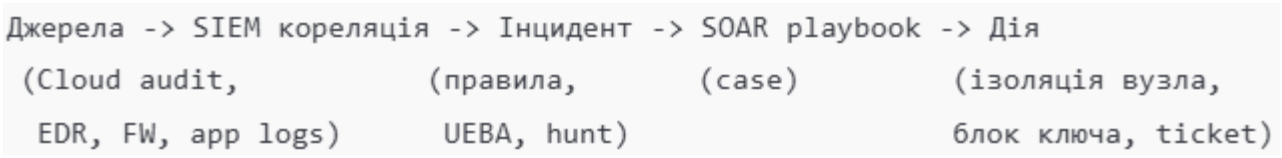

Рисунок 2.7 – Ланцюг “Detections → Incident → Orchestration” у SIEM/SOAR

Важливо, щоб SIEM/SOAR процесно підтримували життєвий цикл інциденту. Оновлена NIST SP 800-61r3 акцентує на фазах Detect/Respond/Recover, включно з пріоритизацією, containment, eradication та recovery, що добре узгоджується з автоматизованими playbooks (наприклад, блокування ключа доступу, ізоляція вузла, примусова ротація секретів).

2.4.5 Виявлення спроб “осліплення” моніторингу та роль MITRE ATT&CK

Окремий клас подій для контролю активності – спроби зловмисника зменшити видимість: вимкнути аудит, змінити експорт логів, видалити “trail/sink”, підчистити журнали. MITRE ATT&CK систематизує джерела даних (data sources) як типи інформації, що можуть збиратися сенсорами/логами для детекції технік.

Для cloud-середовищ MITRE також наводить detection strategy щодо Disable or Modify Cloud Logs, де фокусом є cloud API events, які зупиняють або змінюють логування.

Практичний висновок для архітектури контролю активності: події зміни налаштувань логування та експортів мають бути “high-priority detections” у

SIEM із негайним реагуванням (наприклад, SOAR playbook: відкотити налаштування, заблокувати сесію, створити інцидент критичного рівня).

2.4.6 Критерії ефективності контролю активності

Ефективність системи контролю активності вузлів доцільно оцінювати за сукупністю технічних і процесних показників:

- покриття телеметрії (частка активів, що генерують необхідні логи/метрики/трейси; повнота control plane);
- час виявлення і реагування (MTTD/MTTR) та зменшення “dwell time”;
- якість детекцій (false positives/false negatives, можливість відтворення ланцюга подій);
- стійкість до саботажу (неможливість непомітно вимкнути логування; незмінюваність/контроль доступу до логів, відповідно до практик log management).

Контроль активності вузлів у хмарі є багаторівневою системою: безагентна платформа забезпечує аудит control plane і базову видимість, агентний рівень (у т.ч. EDR) дає глибокі runtime-сигнали, XDR підсилює кореляцію між доменами, а SIEM/SOAR перетворює телеметрію на керовані інциденти та автоматизовані дії реагування. [32]

2.5 Виявлення аномалій і підозрілої активності: правила кореляції, поведінкова аналітика, базові ML-підходи

Виявлення аномалій у хмарній інфраструктурі є логічним продовженням контролю активності вузлів: після збирання логів, метрик і трасування організація має перетворити великий масив подій на обмежену кількість високодостовірних сигналів небезпеки. У практиці безпеки це завдання реалізується як комбінація трьох підходів. Перший – детекції на основі правил і кореляції, де відомі шаблони інцидентів описуються у вигляді умов, часових вікон і послідовностей подій. Другий – поведінкова аналітика (UEBA), що будує профілі «нормальної» активності користувачів і сутностей (вузлів, сервісів, акаунтів) та сигналізує про відхилення. Третій – базові ML-підходи (переважно без учителя або напівконтрольовані), які допомагають знаходити рідкісні або нетипові патерни у телеметрії, коли правила недостатньо гнучкі. Такі підходи узгоджуються з ідеєю безперервного моніторингу (ISCM), де однією з цілей прямо визначається виявлення аномалій і змін у середовищі.

2.5.1 Правила кореляції: перетворення подій на детекції

Правила кореляції – це формалізовані умови, які пов’язують кілька «атомарних» подій у змістовний сценарій загрози: наприклад, «невдалі входи → успішний вхід → створення нового ключа доступу → масовий експорт даних». Якість таких правил критично залежить від організованого log management: події мають бути зібрані, нормалізовані, синхронізовані в часі та захищені від підміни/втрати, і саме це NIST розглядає як основу ефективного аналізу та кореляції журналів.

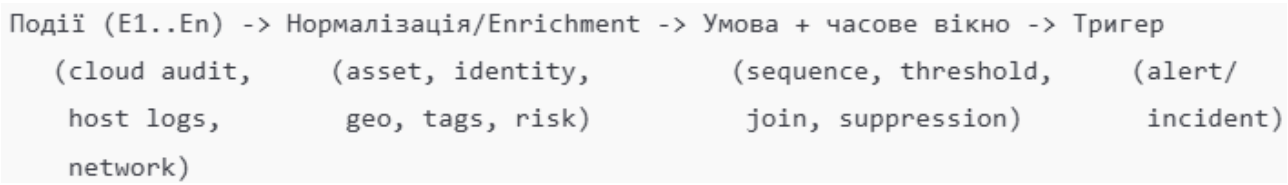

Рисунок 2.8 – Узагальнена логіка кореляційного правила

У SIEM-практиці правила кореляції реалізуються як “correlation searches / analytics rules”, що запускаються за розкладом або потоково, формують алерти та можуть підживлювати ризик-скоринг. Для прикладу, Splunk Enterprise Security описує налаштування correlation searches як окремий механізм детекції та генерації notables/alerts.

Поширеною еволюцією кореляційних правил є risk-based alerting (RBA): система накопичує «ризикові події» в індексі ризику та піднімає інцидент лише коли виконано узгоджені критерії, що зменшує шум і підвищує пріоритизацію.

Таблиця 2.13 – Типові класи кореляційних правил у хмарі

Клас правила	Приклад умови	Дані, що потрібні	Основний плюс	Типовий ризик/мінус
Порогові (threshold)	>N невдалих входів за 5 хв	IAM/SSO логи	простота, швидка детекція brute force	багато false positives без контексту
Послідовні (sequence)	“login” → “role change” → “disable logging”	audit/control plane	добре ловить сценарії ескалації	потребує точних подій і часу
Кореляція по сутності (entity join)	одна IP/акаунт фігурує в різних джерелах	IAM + network + app	пов’язує розрізнені сигнали	складність нормалізації
Rare/новизна за правилом	«перший раз» доступ до критичного сховища	доступи до даних + CMDB	корисно проти інсайдів/зловживань	треба якісний baseline/ретенція
Anti-tamper	вимкнення/зміна експорту логів	audit logs	захищає видимість	висока критичність → потрібна автоматизація

Практично правила кореляції бажано «прив’язувати» до технік супротивника (наприклад, за MITRE ATT&CK), щоб системно керувати покриттям детекцій та уникати хаотичного набору правил. MITRE формалізує підходи до детекції через Detection Strategies як контейнер для аналітик, що поєднують телеметрію з методологією виявлення техніки.

2.5.2 Поведінкова аналітика (UEBA): профілі нормальної активності та відхилення

Поведінкова аналітика UEBA (User and Entity Behavior Analytics) орієнтована на виявлення відхилень у поведінці конкретних користувачів і сутностей. На відміну від правил, які описують «що саме є підозрілим», UEBA вивчає «що є нормою» для даного суб’єкта, а підозру формує як статистичне/поведінкове відхилення. На практиці UEBA спирається на (1) збагачення подій контекстом (роль користувача, критичність ресурсу, тип пристрою, регіон), (2) побудову baseline у часовому вікні, (3) детекцію аномалій і формування risk score. [33]

Наприклад, Microsoft Sentinel прямо має сервіс UEBA, який працює на основі вхідних джерел даних і додає enrichments до сутностей для надання контексту алертам та інцидентам.

Microsoft Learn

У промислових платформах аналогічний підхід часто називають entity analytics або advanced entity analytics і поєднують з ML-аналітикою. Так, Elastic Security описує наявність вбудованих ML jobs для автоматичного виявлення хостових і мережевих аномалій, що є близьким до задач UEBA в частині «нетипової активності сутностей».

Таблиця 2.14 – Типові ознаки (features) UEBA для хмари

Сутність	Приклади ознак поведінки	Основні джерела
Користувач/акаунт	геолокація/ASN входів, частота MFA-відмов, час активності, рідкісні дії	IAM/SSO, audit logs
Вузол/VM/нода	рідкісні процеси, нетипові вихідні з'єднання, скачки ресурсів	EDR/host logs, metrics, flow logs
Сховище/БД	нетипові обсяги читання, масові експорт-операції, доступ поза "звичними" ролями	data access logs, audit
CI/CD сутності	незвичні деплої, зміни секретів, підміна артефактів	pipeline logs, KMS/secret logs

Ключова перевага UEBA в тому, що вона здатна виявляти інциденти, які не мають чіткого «підпису» (signature) або не вкладаються у статичне правило. Водночас UEBA чутлива до якості даних і «дрейфу» поведінки: зміни ролі працівника, релізи системи, міграції в хмарі можуть тимчасово підвищити кількість хибних спрацювань. Тому UEBA зазвичай ефективна у зв'язці з кореляційними правилами та ризик-скорингом (щоб «підозріла, але слабка» аномалія не створювала інцидент сама по собі, а підсилювала інші сигнали).

2.5.3 Базові ML-підходи для аномалій: що застосовно в SOC-практиці

У задачах безпеки маркованих даних часто мало, а клас «атака» є дуже рідкісним. Тому найбільш практичними є підходи без учителя (unsupervised) та новизна/один клас (semi-supervised novelty detection), які будують модель «нормального» і шукають відхилення. У класичній теорії аномалій це відповідає підходу, де outliers трактуються як спостереження, що суттєво відрізняються від решти сукупності. [34]

```
Детекції
 ├ Правила/підписи (correlation, signatures)
 ├ Статистичні baseline (пороги, сезонність, EWMA)
 ├ ML без учителя (Isolation Forest, LOF, clustering, PCA)
 └ ML новизни (One-Class SVM та подібні)
```

Рисунок 2.9 – Таксономія практичних методів детекції

NIST у керівництві щодо IDPS підкреслює, що реальні системи детекції зазвичай комбінують signature-based detection, anomaly-based detection та stateful protocol analysis, оскільки комбінація підвищує точність.

Це важлива практична теза, ML не «замінює» правила, а доповнює їх там, де потрібна адаптивність.

До базових ML-методів, які часто застосовуються в SOC/CloudSec-аналітиці, належать:

- Isolation Forest – підхід, який «ізолює» аномалії, будуючи випадкові дерева розділення; інтуїція полягає в тому, що аномальні точки ізолюються коротшими шляхами. Метод добре масштабується і підходить для багатовимірних ознак активності користувачів/вузлів.
- LOF (Local Outlier Factor) – щільнісний підхід, який оцінює локальну «відокремленість» об’єкта від сусідів, що корисно при неоднорідних режимах роботи (різні сервіси/групи вузлів).
- One-Class SVM – клас моделей новизни, де навчання відбувається на даних «нормального класу», а потім відхилення трактуються як аномалії; метод корисний, коли є стабільний набір «нормальних» періодів і потрібні формальні межі новизни.

Критичною частиною ML-аналітики є підготовка даних і ознак. Для security-аналітики ознаки зазвичай будуються як агрегати у часових вікнах: кількість невдалих входів за 10 хв, частка доступів до критичних ресурсів, середній обсяг експортованих даних, кількість нових ключів/токенів, ентропія DNS-запитів, частота запуску рідкісних процесів тощо. Далі важливими стають нормалізація (однакові шкали), контроль пропусків, усунення дублювання, а також «збагачення» ознак контекстом активу (критичність, середовище prod/dev, роль користувача). Без цього ML-методи перетворюються на генератор шуму.

Таблиця 2.15 – Короткий вибір методу під тип даних безпеки

Тип сигналу	Приклад задачі	Рекомендований базовий метод	Коментар
Часові ряди метрик	різкі стрибки latency/error rate	статистичний baseline / EWMA	простіший за ML, менше “чорної скриньки”
Багатовимірні профілі	“дивна” поведінка вузла/акаунта	Isolation Forest	стійкий до високої розмірності, швидкий
Різні «класи нормальності»	кілька режимів навантаження	LOF / clustering	краще ловить локальні аномалії
Новизна на «нормі»	пошук рідкісних комбінацій дій	One-Class SVM	чутливий до масштабування ознак

Оцінюючи ефективність детекцій, у безпеці недостатньо дивитися лише на стандартні метрики точності. Практично важливі показники: зменшення кількості алертів при збереженні виявлення інцидентів, час до виявлення (MTTD), час до локалізації (MTTR), частка хибних спрацювань у черзі SOC, а також відтворюваність пояснення «чому це аномалія». Тому в SIEM/SOAR-ландшафті ML-аномалії часто використовують як сигнал для кореляції (додають вагу у risk score), а не як самодостатній тригер інциденту. Саме такий підхід підтримують практики risk-based alerting, де рішення про інцидент приймається за сукупністю сигналів і критеріїв. [35]

Виявлення аномалій у хмарі доцільно будувати як багатошарову систему. Кореляційні правила забезпечують контроль відомих сценаріїв і критичних подій control plane. UEBA дає адаптивність до поведінкових зловживань та інсайдерських ризиків, якщо є достатня якість даних і контекст. Базові ML-методи (Isolation Forest, LOF, One-Class SVM) підсилюють виявлення нетипових патернів, але мають працювати у зв’язці з процесом пріоритизації та кореляції, щоб зберігати керованість алертів і практичну цінність для SOC.

2.6 Проєктування комплексної системи захисту та моніторингу: архітектура, компоненти, потоки даних, сценарії реагування

Проєктування комплексної системи захисту та моніторингу для хмарної інфраструктури доцільно виконувати як побудову цілісної «ланки керування ризиками», де технічні контролі (шифрування, IAM, сегментація, EDR, журнали) поєднані з безперервним моніторингом і стандартизованим процесом реагування на інциденти. У термінах NIST ISCM це означає створення програми, яка забезпечує постійну обізнаність про активи, загрози/уразливості та ефективність застосованих контролів. Водночас лог-менеджмент і контроль подій мають бути формалізовані як керований життєвий цикл збору, зберігання та аналізу журналів.

2.6.1 Референсна архітектура: шари, контрольні площини та «landing zone»

Архітектуру комплексної системи зручно описувати як набір шарів: (1) керування ідентичностями та політиками (IAM, Zero Trust), (2) захист даних (KMS/Secrets, класифікація, DLP/retention), (3) захист робочих навантажень (hardening, EDR/XDR, контроль конфігурацій), (4) мережевий захист (сегментація, WAF, приватні endpoints), (5) телеметрія та аналітика (logs/metrics/traces → SIEM), (6) оркестрація реагування (SOAR, playbooks, workflow automation). Такий поділ узгоджується з практиками «security foundations» у хмарних референсах, де акцент робиться на ідентичностях, трасованості (traceability), захисті інфраструктури та реагуванні.

Важливою передумовою є підготовка «безпечної посадкової зони» (secured landing zone), тобто початкового стандартного контуру: ієрархія акаунтів/підписок, централізовані політики, базові журнали, виділені

середовища dev/test/prod, обов’язкові теги ресурсів, стандартні мережеві сегменти й точки інтеграції телеметрії.

2.6.2 Компоненти системи та їхня роль

Компонентний склад має покривати як «запобігання», так і «виявлення/реагування». Для структурування доцільно опиратися на контрольні каталоги та матриці, наприклад CSA Cloud Controls Matrix (CCM) v4 як хмарно-орієнтовану рамку контролів, що допомагає пов’язати вимоги безпеки й приватності з реалізацією у хмарі.

Таблиця 2.16 – Компоненти комплексної системи (мінімальний практичний склад)

Група компонентів	Приклади	Функція в системі	Ключові артефакти/виходи
IAM і Zero Trust	SSO, MFA, RBAC/ABAC, conditional access	контроль суб’єктів доступу та контексту	журнали входів, зміни ролей, risk signals
Захист ключів і секретів	KMS/HSM, Secrets Manager/Vault	шифрування та безпечне зберігання секретів	аудит операцій Encrypt/Decrypt, ротації
Захист навантажень	hardening, EDR/XDR, image signing	контроль стану вузлів, виявлення шкідливої активності	алерти EDR, інвентар ПЗ, IOC/IOA
Мережевий контур	сегментація, FW/SG, WAF, private endpoints	зменшення площі атак, контроль трафіку	flow logs, WAF-алерти, політики мережі
Централізована телеметрія	log sinks, OTel Collector/OTLP	уніфікований збір logs/metrics/traces	нормалізовані події, кореляційні поля
Аналітика/детекції	SIEM, UEBA, rules/ML	кореляція, виявлення аномалій і загроз	alerts/incidents, risk scoring
Оркестрація реагування	SOAR, playbooks, workflow automation	стандартизовані дії при інцидентах	виконані плейбуки, звіти, тикети

Роль журналів у цій архітектурі є фундаментальною: CIS Control 8 прямо визначає аудит-лог-менеджмент як критичний контроль для виявлення, розуміння та відновлення після атак.

2.6.3 Потоки даних: що збираємо, куди відправляємо, як корелюємо

Ключова ідея – розділити телеметрію на потоки control plane і data plane, але звести їх у єдине середовище аналітики. Контрольна площина дає події IAM, створення/зміни ресурсів, операції з ключами та налаштування логування; площина даних дає події вузлів, мережі та застосунків. Для сучасних систем спостережуваності доцільно уніфікувати logs/metrics/traces через OpenTelemetry, який визначає сигнали та типову архітектуру (API/SDK, Collector, протокол OTLP) і орієнтується на кореляцію сигналів через спільний контекст.

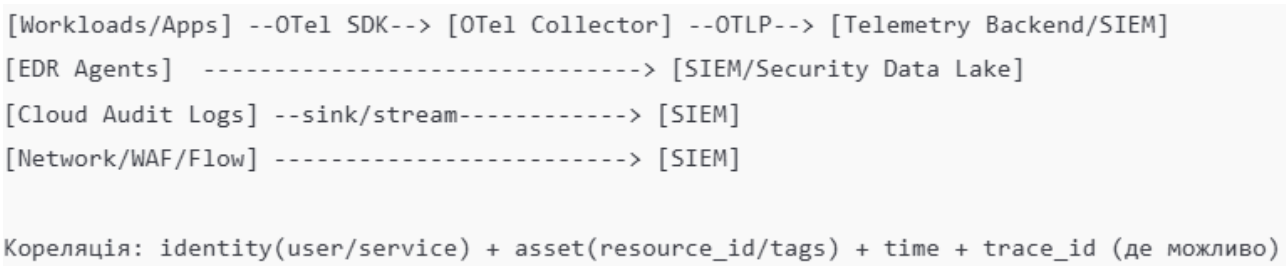

Рисунок 2.11 – Потік телеметрії та кореляції

2.6.4 Сценарії реагування: від детекції до автоматизації дій

Процес реагування доцільно узгоджувати з NIST SP 800-61r3, де реагування розглядається як частина кіберризик-менеджменту і включає підготовку, детекцію/аналіз, стримування, усунення та відновлення з подальшим удосконаленням. [36]

Таблиця 2.18 – Приклади сценаріїв реагування (runbooks/playbooks) у хмарі

Сценарій	Тригери (детекції)	Автоматизовані дії	Ручні дії та докази
Компрометація IAM-ключа	“неможлива географія”,	блок/відкликання токенів, ротація	аналіз журналів, підтвердження

	створення ключа, масові API-дії	ключів, вимога MFA	власника, RCA (NIST Publications)
Витік/експфільтрація даних	масове читання/експорт, нетипові sink-и, аномальні потоки	обмеження доступу, блок egress, заморозка політик	перевірка DLP/класифікації, інформування DPO/юристів
Підозріла активність на вузлі	EDR: suspicious process, lateral movement	ізоляція хоста, зупинка pod/VM, форензичний snapshot	аналіз IOC/IOA, відновлення з «чистого» образу
Спроба приховати сліди	вимкнення/зміна логування, зміни audit sinks	блок змін політик, підняття пріоритету інциденту	перевірка доступів, SoD, аудит конфігурацій (NIST Publications)

На практиці ефективність суттєво підвищується, коли типові кроки автоматизуються через SOAR/плейбуки: наприклад, Microsoft Sentinel описує playbooks і їх прив’язку до правил аналітики/автоматизації, а Microsoft Defender for Cloud підтримує workflow automation для типових процедур реагування (сповіщення, запуск ремедіації, інтеграція з ITSM).

Для стабільної роботи комплексної системи важливо формалізувати «інженерію детекцій»: визначити мінімально необхідні події, схеми нормалізації, політики ретенції, вимоги до незмінності та контроль доступу до телеметрії (оскільки логи самі по собі є чутливими даними). Це прямо відповідає рекомендаціям NIST щодо log management і забезпечує доказовість розслідувань.

Проєктування комплексної системи захисту та моніторингу в хмарі має завершуватися узгодженою архітектурою (landing zone + шари контролів), визначеним переліком компонентів (IAM/KMS/EDR/SIEM/SOAR/CSPM), описаними потоками даних (control plane/data plane → централізована аналітика) і бібліотекою сценаріїв реагування (runbooks/playbooks) з максимальною автоматизацією рутинних дій. Така система забезпечує не лише конфіденційність і контроль доступу, а й практичну керованість інцидентів за рахунок швидкого виявлення, кореляції та стандартизованого відновлення.

2.7 Висновки до розділу 2

У розділі сформовано цілісну постановку задачі захисту даних і контролю активності вузлів у хмарній інфраструктурі: визначено межі системи, складові (control plane, data plane, спостережуваність/реагування), об’єкти захисту та модель загроз із прив’язкою до практик ризик-орієнтованого підходу.

Обґрунтовано, що в хмарі “об’єкт захисту” слід трактувати як актив (дані, ідентичності/секрети, робочі навантаження, мережеві компоненти, контрольна площина), а вимоги мають охоплювати не лише CIA (конфіденційність, цілісність, доступність), але й аудитованість/спостережуваність як окрему критичну властивість.

Систематизовано класи типових загроз (компрометація облікових даних, помилки конфігурацій, supply chain/CI-CD, зловживання привілеями, атаки на доступність) і показано, що ризик доцільно оцінювати як функцію ймовірності реалізації та очікуваних наслідків, що дозволяє формально порівнювати альтернативні контрзаходи.

Визначено узагальнену матрицю вимог (функціональних і нефункціональних) для хмарного середовища, де ключовими є: керування доступом за least privilege, контроль змін (IaC/версійність/immutability), керованість інцидентів (runbooks/playbooks) і комплаєнс-доказовість через журнали, метрики та результати DR-тестів.

Запропоновано систему критеріїв ефективності, яка переводить безпеку з декларативного рівня у вимірюваний: покриття шифрування, повнота control plane логів, MTTD/MTTR, якість детекцій (FP/FN), виконання SLO/SLA, стійкість бекапів (immutable/ізоляція) тощо.

Розглянуто методи захисту даних у трьох станах життєвого циклу:

- in transit — TLS/mTLS як базовий транспортний контроль із акцентом на правильну конфігурацію та керування сертифікатами;
- at rest — шифрування сховищ/БД з опорою на KMS/HSM та керованість ключового життєвого циклу;
- in use — конфіденційні обчислення (TEE/enclaves) як механізм зменшення ризиків привілейованого рівня та основа політик “key release після атестації”.

Показано, що контроль доступу в хмарі є “центром тяжіння” безпеки: RBAC забезпечує керовану структуру прав, ABAC додає контекстні обмеження, MFA знижує ризики компрометації ідентичностей, а Zero Trust задає режим безперервної перевірки та мінімізації довіри. Обґрунтовано комбінований підхід до контролю активності вузлів: безагентні джерела забезпечують аудит control plane та конфігурацій, агентні (включно з EDR) — глибокі runtime-сигнали, XDR підсилює міждоменну кореляцію, а SIEM/SOAR перетворює телеметрію на керовані інциденти та автоматизовані дії реагування. Розкрито підхід до виявлення підозрілої активності як багатошарову систему: кореляційні правила для відомих сценаріїв, UEBA для відхилень поведінки сутностей, та базові ML-методи (наприклад, Isolation Forest/LOF/One-Class SVM) як підсилювачі сигналів у зв’язці з ризик-скорингом і пріоритизацією SOC.

Запропоновано референсну архітектуру комплексної системи захисту та моніторингу (landing zone + шари контролів), визначено ключові компоненти та потоки телеметрії (control plane/data plane → централізована аналітика), а також окреслено бібліотеку типових сценаріїв реагування з фокусом на автоматизації критичних дій (стримування, блокування, ротація, відновлення).

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНИХ РІШЕНЬ

3.1 Вибір хмарної платформи та інструментарію Azure: обґрунтування та обмеження

Вибір хмарної платформи для реалізації методів захисту даних і контролю активності вузлів має бути формалізованим рішенням, яке спирається на вимоги до конфіденційності, цілісності та доступності, а також на потреби спостережуваності, реагування на інциденти й комплаєнсу. У межах цієї кваліфікаційної роботи доцільно обрати Microsoft Azure як цільову платформу через наявність узгоджених “еталонних” практик архітектування (Well-Architected Framework), готових патернів підготовки корпоративного середовища (Landing Zone), зрілої екосистеми безпеки (Defender for Cloud, Entra ID, Key Vault/Managed HSM) та нативного стеку моніторингу й керування журналами (Azure Monitor, Activity Log, Diagnostic settings, Log Analytics). [37]

3.1.1 Критерії вибору платформи в контексті теми роботи

Для теми “Методи захисту даних і контролю активності вузлів у хмарній інфраструктурі” практичний вибір платформи варто оцінювати за такими групами критеріїв:

Безпека та керування ідентичностями: централізований IAM, MFA, умовний доступ, принцип Zero Trust, контроль привілеїв. В Azure ключовим елементом є Microsoft Entra Conditional Access як “policy engine” Zero Trust для примусового виконання політик доступу.

Захист даних: підтримка шифрування, керування ключами (KMS/HSM), customer-managed keys, керування секретами та аудит операцій з ключами.

Azure надає Key Vault Managed HSM як керований, стандартно-орієнтований сервіс з використанням HSM.

Спостережуваність і контроль активності: можливість збирати control plane та data plane журнали, метрики, корелювати події, виконувати пошук/аналітику, налаштовувати автоматизацію реагування. В Azure це реалізується через Azure Monitor, Activity Log, Diagnostic settings і Log Analytics workspace.

Керованість, стандартизація й масштабованість середовища: наявність референсної архітектури (landing zone), політик комплаєнсу, контроль конфігурацій і дрейфу. Azure Landing Zones та Azure Policy забезпечують типову основу для governance та compliance.

Надійність і прозорість SLA: доступність сервісів та договірні гарантії. Microsoft публікує SLA для Online Services (включно з Azure) у поточних та архівних редакціях.

3.1.2 Обґрунтування вибору Azure з позиції архітектури та безпеки

Архітектурна “рамка” для якості та безпеки. Azure Well-Architected Framework формалізує підхід до побудови навантажень за п’ятьма стовпами, а для безпеки надає окремий набір практик і посилань (“Security quick links”) та інструменти оцінювання. Це важливо, бо дозволяє прив’язати рішення з шифрування, IAM, журналювання та реагування до узгоджених design principles і чек-листів, а не будувати систему “інтуїтивно”.

“Landing Zone” як базова модель побудови корпоративного хмарного середовища. Для задач контролю активності вузлів критично мати стандартизовану структуру підписок/ресурсів, єдині правила доступу, логування та політик. Azure Cloud Adoption Framework описує Azure landing zone як цільову архітектуру платформи з урахуванням ідентичностей, мережі,

безпеки, управління та операцій. Окремо визначено design areas (governance, operations тощо) і практики реалізації через IaC. [38]

Нативний стек управління безпекою та “постурою”. Microsoft Defender for Cloud позиціонується як рішення для покращення security posture та захисту робочих навантажень, включно з можливостями agentless/agent-based підходів у хмарному захисті. Це напряму відповідає підрозділам роботи щодо контрольованості вузлів і виявлення підозрілої активності.

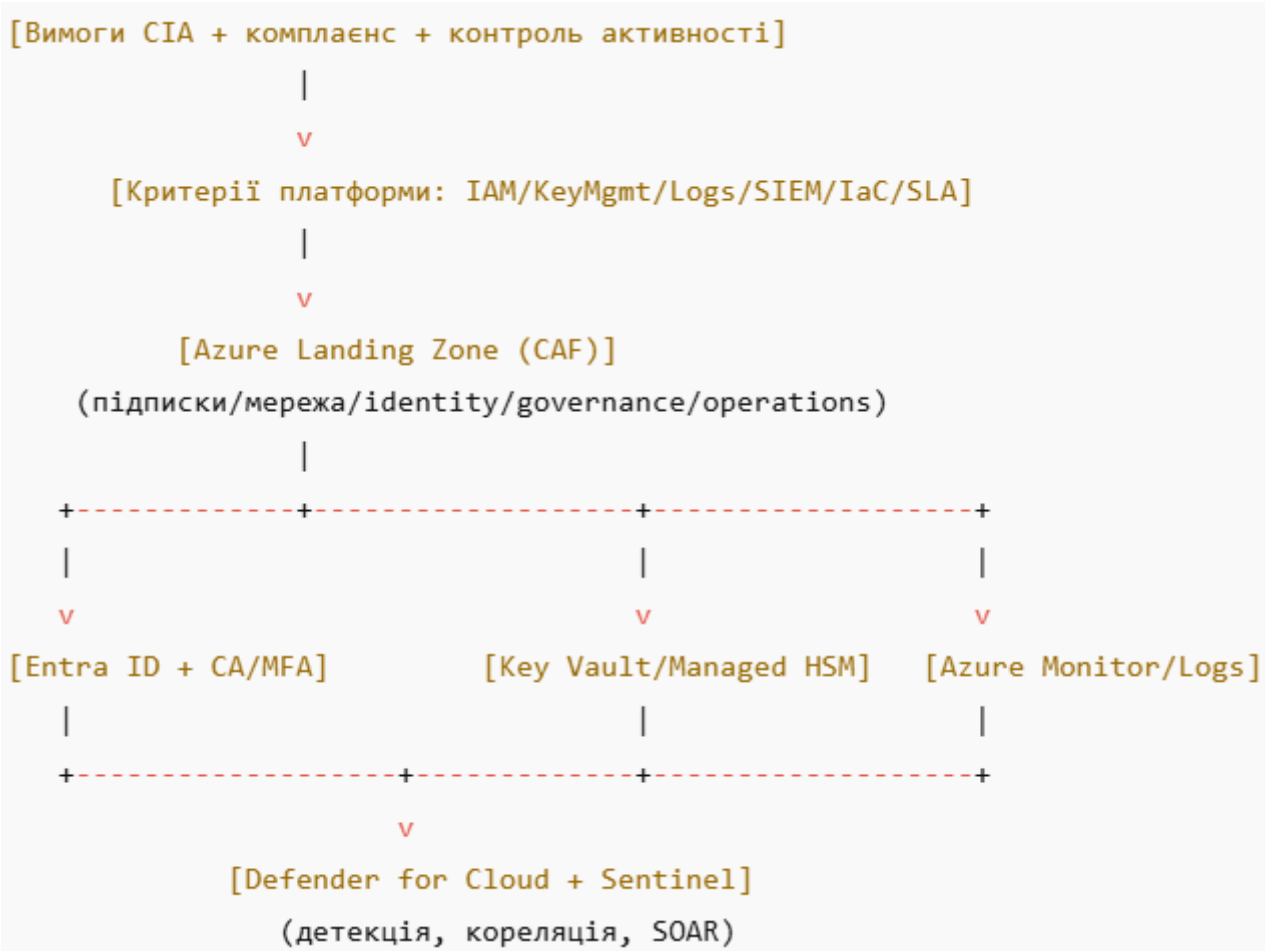

Рисунок 3.1 – Узагальнена логіка вибору Azure та прив’язка до компонентів системи

Зріла модель централізованого моніторингу та лог-менеджменту. Azure Monitor описує збір і аналіз metrics/logs/traces, а Log Analytics workspace виступає як централізоване сховище для кореляції даних з Azure і не-Azure

ресурсів. Окремо Activity Log забезпечує видимість subscription-level подій (control plane), а Diagnostic settings дозволяють маршрутизувати журнали/метрики/Activity Log у потрібні "приймачі" (Log Analytics, Event Hubs, Storage тощо).

Автоматизація реагування на інциденти на рівні SIEM/SOAR. Для практичного інцидент-менеджменту корисна інтеграція аналітики та автоматизованих playbooks. У Microsoft Sentinel playbooks базуються на Azure Logic Apps і можуть прив'язуватися до правил аналітики або automation rules, що зручно для типових сценаріїв реагування (ізоляція, блокування доступу, створення задач, нотифікації).

Таблиця 3.1 – Відповідність задач теми роботи інструментам Azure

Задача в межах роботи	Рекомендовані сервіси/інструменти Azure	Очікуваний результат
Контроль ідентичностей, MFA, Zero Trust доступ	Microsoft Entra ID, Conditional Access, MFA	Зниження ризику компрометації облікових даних і надмірних доступів (Microsoft Learn)
Шифрування та керування ключами, HSM	Azure Key Vault, Key Vault Managed HSM, CMK	Централізоване управління ключами, аудит операцій шифрування (Microsoft Learn)
Логування control plane та ресурсів	Activity Log, Diagnostic settings, Resource logs	Доказовість змін і подій, експорт у сховища/аналітику (Microsoft Learn)
Централізована аналітика логів	Log Analytics workspace, Azure Monitor Logs	Кореляція подій, пошук, побудова запитів і алертів (Microsoft Learn)
Захист постури та детекція загроз	Microsoft Defender for Cloud	Виявлення уразливостей/ризиків конфігурацій, захист навантажень (Microsoft Learn)
SOAR-реагування та автоматизація	Microsoft Sentinel playbooks (Logic Apps), automation rules	Автоматизоване реагування на інциденти за сценаріями (Microsoft Learn)
Governance/комплаєнс-політики	Azure Policy, Regulatory Compliance initiatives	Масштабне оцінювання відповідності та ремедіація (Microsoft Learn)

3.1.3 Інструментарій реалізації: IaC, контроль змін, оцінка архітектури

Для практичної реалізації системи захисту й моніторингу важливо, щоб інфраструктура створювалася відтворювано та контрольовано. У CAF для landing zone прямо акцентується перевага Infrastructure as Code та наявні інструменти-акселератори (Bicep або Terraform на основі Azure Verified Modules) з підтримкою типових VCS/CI-CD (Azure DevOps або GitHub). Це дозволяє мінімізувати конфігураційний дрейф і забезпечити аудит змін через репозиторій.

Додатково, для регулярної “перевірки якості” архітектурних рішень доцільно використовувати механізми оцінювання Well-Architected та Advisor score, які агрегують рекомендації щодо безпеки, надійності й вартості.

3.1.4 Обмеження та ризики вибору Azure (і як їх врахувати в роботі)

Vendor lock-in та специфіка сервісів. Нативні сервіси (Defender/Sentinel/Policy/Key Vault) дають високу інтегрованість, але ускладнюють міграцію до іншого провайдера без зміни процесів та конфігурацій. Це слід враховувати через абстракції (IaC, стандартизовані формати телеметрії, контрактні інтерфейси інтеграцій) та описати як обмеження дослідження.

Квоти, ліміти та експлуатаційні обмеження. Azure має системні subscription/service limits і квоти, які можуть спричиняти відмови розгортання або масштабування, якщо їх не планувати. У роботі доцільно відобразити потребу у квотному плануванні й моніторингу використання.

Вартість спостережуваності та зберігання журналів. Централізація логів у Log Analytics, експорт через Diagnostic settings та використання SOAR-плейбуків можуть створювати додаткові витрати (збір, зберігання, запити,

автоматизація). Практично це означає необхідність політик ретенції, фільтрації, нормалізації та пріоритизації подій, а також визначення мінімально достатнього набору сигналів. [39]

SLA як “межа” відповідальності, а не гарантія безперервності бізнесу. Microsoft публікує SLA для Online Services та механізми service credits, однак SLA не замінює архітектуру відмовостійкості, резервування й DR-процедури. Тому в роботі варто трактувати SLA як вхідний параметр для проєктування (RTO/RPO, multi-zone/region), а не як достатню умову надійності.

Регіональна доступність і комплаєнс-обмеження. Деякі сервіси можуть відрізнятися за доступністю між регіонами; також важливі вимоги до резидентності даних і доказів відповідності. Для Azure це частково знімається використанням Landing Zone і Azure Policy Regulatory Compliance (ініціативи й “ownership” контролів), але потребує правильного проєктування середовища.

Таблиця 3.2 – Узагальнена матриця “перевага/обмеження” вибору Azure

Аспект	Перевага Azure	Потенційне обмеження
IAM/Zero Trust	Conditional Access як політичний рушій, MFA	Складність політик, потреба в дисципліні ролей (Microsoft Learn)
Key management	Managed HSM, CMK-підтримка сервісів	Додаткова складність життєвого циклу ключів (Microsoft Learn)
Спостережуваність	Azure Monitor + Log Analytics, експорт діагностики	Вартість зберігання/запитів, шум подій (Microsoft Learn)
Security posture	Defender for Cloud (постура/захист навантажень)	Потреба налаштування планів/скоупів, процесів ремедіації (Microsoft Learn)
SIEM/SOAR	Sentinel playbooks/automation rules	Додаткові витрати/складність автоматизацій (Microsoft Learn)
Governance	Azure Policy, regulatory compliance initiatives	Ризик “надмірного” контролю, якщо політики не узгоджені з процесами (Microsoft Learn)
Масштабування	Landing zone + IaC підхід	Квоти/ліміти, потреба у плануванні потужностей (Microsoft Learn)

3.1.5 Інструментарій реалізації: IaC, контроль змін, оцінка архітектури

Для практичної реалізації системи захисту й моніторингу важливо, щоб інфраструктура створювалася відтворювано та контрольовано. У CAF для landing zone прямо акцентується перевага Infrastructure as Code та наявні інструменти-акселератори (Bicep або Terraform на основі Azure Verified Modules) з підтримкою типових VCS/CI-CD (Azure DevOps або GitHub). Це дозволяє мінімізувати конфігураційний дрейф і забезпечити аудит змін через репозиторій. [40]

Додатково, для регулярної “перевірки якості” архітектурних рішень доцільно використовувати механізми оцінювання Well-Architected та Advisor score, які агрегують рекомендації щодо безпеки, надійності й вартості.

3.1.6 Обмеження та ризики вибору Azure (і як їх врахувати в роботі)

Vendor lock-in та специфіка сервісів. Нативні сервіси (Defender/Sentinel/Policy/Key Vault) дають високу інтегрованість, але ускладнюють міграцію до іншого провайдера без зміни процесів та конфігурацій. Це слід враховувати через абстракції (IaC, стандартизовані формати телеметрії, контрактні інтерфейси інтеграцій) та описати як обмеження дослідження.

Квоти, ліміти та експлуатаційні обмеження. Azure має системні subscription/service limits і квоти, які можуть спричиняти відмови розгортання або масштабування, якщо їх не планувати. У роботі доцільно відобразити потребу у квотному плануванні й моніторингу використання.

Вартість спостережуваності та зберігання журналів. Централізація логів у Log Analytics, експорт через Diagnostic settings та використання SOAR-плейбуків можуть створювати додаткові витрати (збір, зберігання, запити,

автоматизація). Практично це означає необхідність політик ретенції, фільтрації, нормалізації та пріоритизації подій, а також визначення мінімально достатнього набору сигналів. [41]

SLA як “межа” відповідальності, а не гарантія безперервності бізнесу. Microsoft публікує SLA для Online Services та механізми service credits, однак SLA не замінює архітектуру відмовостійкості, резервування й DR-процедури. Тому в роботі варто трактувати SLA як вхідний параметр для проєктування (RTO/RPO, multi-zone/region), а не як достатню умову надійності.

Регіональна доступність і комплаєнс-обмеження. Деякі сервіси можуть відрізнятися за доступністю між регіонами; також важливі вимоги до резидентності даних і доказів відповідності. Для Azure це частково знімається використанням Landing Zone і Azure Policy Regulatory Compliance (ініціативи й “ownership” контролів), але потребує правильного проєктування середовища.

Таблиця 3.2 – Узагальнена матриця “перевага/обмеження” вибору Azure

Аспект	Перевага Azure	Потенційне обмеження
IAM/Zero Trust	Conditional Access як політичний рушій, MFA	Складність політик, потреба в дисципліні ролей (Microsoft Learn)
Key management	Managed HSM, CMK-підтримка сервісів	Додаткова складність життєвого циклу ключів (Microsoft Learn)
Спостережуваність	Azure Monitor + Log Analytics, експорт діагностики	Вартість зберігання/запитів, шум подій (Microsoft Learn)
Security posture	Defender for Cloud (постура/захист навантажень)	Потреба налаштування планів/скоупів, процесів ремедіації (Microsoft Learn)
SIEM/SOAR	Sentinel playbooks/automation rules	Додаткові витрати/складність автоматизацій (Microsoft Learn)
Governance	Azure Policy, regulatory compliance initiatives	Ризик “надмірного” контролю, якщо політики не узгоджені з процесами (Microsoft Learn)
Масштабування	Landing zone + IaC підхід	Квоти/ліміти, потреба у плануванні потужностей (Microsoft Learn)

Azure є обґрунтованим вибором для практичної частини цієї роботи, оскільки забезпечує цілісну зв’язку “ідентичність → ключі/шифрування →

журнали/телеметрія → детекція → реагування", а також має референсні підходи до побудови керованого середовища (landing zone) і оцінки архітектурної якості (Well-Architected/Advisor). Водночас платформа накладає обмеження, пов'язані з квотами, вартістю спостережуваності та потенційною залежністю від провайдера, що має бути явно зафіксовано як припущення та межі дослідження.

3.2 Реалізація захисту даних в Azure: політики шифрування, керування ключами, секрети, сегментація мережі, резервне копіювання

Реалізацію захисту даних у Microsoft Azure доцільно будувати як керований набір технічних і організаційних контролів, які застосовуються узгоджено на рівнях даних, ідентичностей, мережі та процесів відновлення. Практика показує, що в хмарі критично важливо не лише «увімкнути» окремі опції (наприклад, шифрування сховища), а й формалізувати політики, що гарантують незворотність і повторюваність налаштувань у всіх середовищах (dev/test/prod). Такий підхід узгоджується з концепцією encryption in cloud як поєднання шифрування, керування ключами та контролю доступу до криптографічних операцій, яку описує документація Azure. [42]

3.2.1 Політики шифрування: «у русі», «у спокої», «у використанні»

Дані "у русі" (in transit). Базова вимога – використання TLS 1.2+/TLS 1.3 для зовнішніх і внутрішніх з'єднань, а для сервіс-до-сервіс взаємодії в мікросервісних архітектурах – mTLS (взаємна автентифікація). Вимоги до коректного вибору TLS-параметрів, криптографічних наборів і заборони застарілих конфігурацій описані в рекомендаціях NIST SP 800-52r2.

На рівні PaaS-сховищ типова практична політика – заборонити незахищений HTTP. Для Azure Storage це підтримується параметром Secure

transfer required (примус HTTPS для вхідних запитів), який рекомендовано залишати увімкненим як налаштування «за замовчуванням».

Дані “у спокої” (at rest). Azure декларує шифрування «на носіях» як базову властивість платформи, а для ряду сервісів підтримує Customer-Managed Keys (CMK), щоб замовник контролював ключовий матеріал і політики доступу до операцій шифрування/розшифрування. Це особливо важливо для комплаєнсу та сценаріїв, де потрібна підконтрольність ключів (BYOK/CMK).

Для IaaS-дисків застосовується Disk Encryption Set (DES) як механізм прив’язки CMK до дисків та снапшотів у керованих дисках.

Дані “у використанні” (in use). У межах цього підрозділу достатньо зафіксувати принцип: якщо модель загроз вимагає захисту даних у пам’яті під час обробки, в Azure розглядають конфіденційні обчислення (наприклад, Confidential VMs/TEE). Деталізація зазвичай виноситься в окремий пункт про confidential computing. [43]

Таблиця 3.1 – Відображення політик шифрування на механізми Azure

Стан даних	Ціль контролю	Типові механізми в Azure	Артефакт політики/контролю
In transit	Конфіденційність і цілісність каналу	TLS/mTLS, HTTPS-only endpoints, Private Link	Стандарти TLS, «Secure transfer required» для Storage
At rest	Захист носіїв/бекупів/снапшотів	SSE/TDE, CMK через Key Vault, Disk Encryption Set	Політики CMK, ротація ключів, контроль доступу до KMS
In use	Зменшення ризику доступу до даних у RAM	Confidential computing/TEE (за потреби)	Окрема політика для критичних workload-ів

3.2.2 Керування ключами: Key Vault / Managed HSM, життєвий цикл і незворотність захисту

Шифрування «працює» лише тоді, коли захищені ключі та регламентований їх життєвий цикл (створення, використання, ротація, компрометація, знищення). NIST SP 800-57 Part 1 формалізує принципи керування ключами, включно з ролями, політиками ротації та реакцією на компрометацію.

У Azure роль сховища ключів і секретів виконує Azure Key Vault, а для сценаріїв із підвищеними вимогами до апаратного захисту ключів – Managed HSM. Офіційна документація підкреслює, що Key Vault забезпечує централізоване зберігання ключів/сертифікатів/секретів і контроль доступу до криптографічних операцій.

Ключові практики реалізації:

- Перехід на CMK там, де це обґрунтовано вимогами (комплаєнс, критичні дані, потрібен контроль відкликання/ротації). Для Storage/DB/дисків це дає можливість прив’язати шифрування до ключів у Key Vault і контролювати доступ через IAM.
- Розділення обов’язків (SoD): адміністратор ключів не має бути тим самим суб’єктом, що адмініструє дані/сховища.
- Незворотність захисту від видалення ключів: увімкнути soft delete і purge protection для Key Vault, щоб унеможливити «остаточне» знищення ключів/секретів у межах періоду ретенції. Документація Azure прямо рекомендує soft delete/purge protection як частину data protection для Key Vault і описує обмеження purge protection.
- Мережева ізоляція Key Vault: доступ до сховища ключів і секретів бажано обмежувати приватними мережевими механізмами (Private Endpoint або service endpoints для окремих сценаріїв).

Рисунок 3.1 – Узагальнений потік “envelope encryption” з CMK у Key Vault (концептуально)

3.2.3 Секрети: централізація, мінімізація “статичних” ключів, керований доступ

Секрети (паролі, токени, ключі підключення) – один із найчастіших каналів компрометації, особливо через витоки з репозиторіїв або CI/CD. Azure Key Vault підтримує зберігання secrets як керований тип об’єктів.

Практичний «еталон» для Azure-архітектури – не зберігати секрети у конфігах і на дисках workload-ів, а надавати доступ до Key Vault через контрольовані ідентичності. [44]

Ключовий механізм мінімізації секретів у runtime – Managed Identities (керовані ідентичності), які дозволяють workload-ам отримувати токени доступу без зберігання облікових даних у коді.

Доступ до секретів варто обмежувати принципом найменших привілеїв (read конкретних secrets), доповнювати журналюванням звернень до Key Vault та правилами виявлення аномалій (наприклад, нетипові джерела запитів або масові читання секретів).

3.2.4 Сегментація мережі: ізоляція зон довіри, приватні кінцеві точки, централізований контроль

Мережева сегментація в Azure вирішує дві задачі: (1) зменшення площі атак (мінімум публічних endpoint-ів), (2) локалізація інцидентів і ускладнення lateral movement. Базовими елементами є Virtual Network, subnets, Network Security Groups (NSG) і керовані мережеві сервіси безпеки. NSG забезпечують фільтрацію трафіку правилами allow/deny для inbound/outbound.

Для корпоративних систем типовим шаблоном є hub-spoke топологія, де hub містить спільні мережеві сервіси (DNS, VPN/ExpressRoute, firewall), а spokes ізолюють робочі навантаження за середовищами або доменами. Архітектурний центр Azure описує hub-spoke як референсну модель ізоляції та централізації спільних сервісів.

Таблиця 3.2 – Сегментація мережі та які ризики вона знижує

Контроль сегментації	Реалізація в Azure	Основні ризики, що знижуються
Ізоляція середовищ	Окремі VNet/споки (dev/test/prod)	Помилки доступу між середовищами, lateral movement
Мікросегментація	Subnets + NSG/ASG	Компрометація одного вузла → поширення по мережі
Централізований egress/ingress	Hub + Azure Firewall + UDR	Неконтрольований вихід назовні, C2-канали, витоки
Приватний доступ до PaaS	Private Endpoint/Private Link	Експозиція сервісів у публічній мережі, перехоплення/сканування

У hub часто розміщують Azure Firewall як централізований stateful-контроль для egress/ingress та міжсегментного трафіку; Well-Architected guidance також прямо прив’язує Firewall до типових топологій (hub-spoke, vWAN). [45]

Окремий критичний компонент сучасної сегментації в Azure – Private Link / Private Endpoint, що дозволяє підключати PaaS-сервіси (Storage, SQL тощо) приватно через IP у VNet і прибирати експозицію з публічного Інтернету.

3.2.5 Резервне копіювання: RPO/RTO, захист від видалення, “immutable” та розділення доступів

Резервне копіювання в хмарі – це не лише «копії даних», а керований процес відновлення, який має відповідати заданим RPO/RTO та бути стійким до зловмисних дій (ransomware, insider). NIST SP 800-34r1 підкреслює, що планування відновлення та регулярні вправи (tests) є обов’язковими компонентами contingency planning.

В Azure базовим сервісом є Azure Backup (через Recovery Services vault / Backup vault), який надає функції захисту, зберігання точок відновлення та керування політиками ретенції.

Для підвищення стійкості до руйнівних операцій Azure Backup пропонує:

- Secure by Default / soft delete: затримка остаточного видалення, що дозволяє відновити дані після випадкового або зловмисного видалення в межах періоду ретенції.
- Immutable vault: блокування операцій, які можуть призвести до втрати recovery points, з можливістю «lock» (незворотне увімкнення) та моделлю WORM.
- Multi-User Authorization (MUA) через Resource Guard: додатковий бар’єр для критичних дій (наприклад, вимкнення захистів або руйнівні операції) шляхом вимоги окремої авторизації.

Таблиця 3.3 – Мінімальний “ransomware-resilient” профіль для Azure Backup

Вимога	Реалізація	Навіщо
Відновлюваність після видалення	Secure by Default / soft delete	Захист від випадкового/зловмисного delete
Незмінюваність точок відновлення	Immutable vault (+ lock)	Захист від “очищення” recovery points
Захист критичних операцій	MUA (Resource Guard)	Бар’єр проти rogue admin / компрометації ролей
Перевірка працездатності	DR/Restore tests	Доказ, що RPO/RTO досяжні

Практично, політика резервування має включати: класифікацію даних за критичністю, відповідні RPO/RTO, вибір схеми надмірності (zone/region), окремі ролі на керування backup-ами, а також періодичні тести відновлення, інакше наявність бекупу не гарантує відновлюваності. Як орієнтир мінімальних практик з «data recovery» доречно посилатися на CIS Controls v8, які акцентують на наявності резервних копій і відпрацьованих процедурах відновлення.

3.3 Реалізація контролю активності вузлів: метрики/логи/трейси, централізований збір, правила детекції, оповіщення

Контроль активності вузлів у хмарній інфраструктурі (віртуальні машини, вузли AKS/Kubernetes, вузли інтеграційних сервісів та прикладні компоненти) доцільно будувати як безперервне спостереження (continuous monitoring), у якому телеметрія є основою для виявлення відхилень, форензіки та реагування. Такий підхід узгоджується з практиками організації ефективного лог-менеджменту на рівні підприємства (політики збору, зберігання та аналізу) та з логікою оцінювання програм ISCM (continuous monitoring), де ключову роль відіграють стратегії, процедури та аналіз даних спостереження.

У контексті Azure доцільно використовувати Azure Monitor як базову платформу, оскільки він позиціонується як комплексне рішення для збору, аналізу та реакції на дані моніторингу з хмарних і локальних середовищ. [46]

3.3.1 Телеметрія як основа контролю: метрики, логи, трейси

Практична реалізація контролю активності вузлів спирається на “три сигнали” спостережуваності, які стандартно описуються в OpenTelemetry: traces (шлях запиту через систему), metrics (вимірювання під час виконання) та logs (запис подій)

Це важливо не лише для експлуатаційної надійності, а й для безпеки: одна і та сама подія (наприклад, компрометація вузла) проявляється як аномальні метрики, підозрілі записи в логах і нетипові ланцюжки запитів у трасуванні.

У практичній архітектурі Azure зазвичай виділяють два “плани” спостереження:

- Platform/control-plane (керування ресурсами через ARM/Entra ID, зміни конфігурацій, події сервісів).
- Workload/data-plane (власне активність ОС/контейнерів/процесів і прикладна телеметрія).

Розділення потрібне тому, що інциденти часто починаються з control-plane (компрометація ідентичності, ключів, ролей), а реалізуються в data-plane (запуск шкідливого процесу, ексфільтрація даних).

3.3.2 Централізований збір у Azure: DCR, агентність і маршрутизація

Ключовий принцип реалізації – централізувати правила збору даних, мінімізувати “ручні” відхилення між середовищами та мати контроль вартості

інжесту. У Azure це досягається через Data Collection Rules (DCRs): вони зберігаються як ресурси Azure і дають централізований спосіб визначити, що збирати, як трансформувати (фільтрація/агрегація/форматування) та куди надсилати.

На рівні клієнтських ОС збір реалізується через Azure Monitor Agent (AMA), який прямо вказує на DCR як на механізм визначення типів даних, трансформацій і призначення

Для VM додатково підкреслюється, що платформа автоматично збирає частину хост-метрик і журналів, а от дані з ОС і workload потребують DCR із визначенням “що і куди” збирати.

```
[VM / AKS nodes / Apps]
   |   (metrics, logs, traces)
   v
[Agents & exporters]
  - AMA (guest logs/metrics) -> DCR
  - OTel exporter / App Insights SDK/OTel distro (traces/metrics/logs)
  - Prometheus scraping (K8s/VM exporters)
   v
[Ingestion endpoints & routing]
  - Data Collection Endpoint (DCE) / Azure Monitor ingestion
   v
[Central stores]
  - Azure Monitor Metrics store / Managed Prometheus
  - Log Analytics Workspace (Logs)
  - Application Insights (workspace-based)
   v
[Analytics & response]
  - Azure Monitor Alerts (metric/log/activity/Prometheus)
  - Microsoft Sentinel Analytics (KQL rules) -> Incidents
  - Action Groups / Automation rules / Playbooks
```

Рисунок 3.3 – Узагальнений потік телеметрії (Azure-референс)

Data Collection Endpoint використовується як частина архітектури маршрутизації даних до Azure Monitor (зокрема у сценаріях custom logs/AMA/DCR). [47]

3.3.3 Реалізація збору сигналів: метрики, логи, трейси

Метрики. Для інфраструктурного рівня частина метрик є “platform metrics” і доступна без інструментування, однак для cloud-native середовищ критично додати Prometheus-метрики з Kubernetes/експортерів. Azure Monitor надає керовану службу для Prometheus: можна збирати, зберігати й аналізувати Prometheus-метрики без підтримки власного Prometheus-сервера.

Це дає уніфікований шлях до метрик вузлів (CPU/RAM/FS), метрик kubelet/kube-state, а також метрик прикладних сервісів. Надалі метрики стають основою SLI/SLO і “швидких” алертів (перевантаження, деградація, аномальна кількість рестартів pod тощо).

Логи – основний доказовий шар для контролю активності. В інженерній реалізації доцільно зібрати щонайменше:

- журнали ОС (Syslog/Windows Event Logs);
- логи безпеки (SecurityEvent/аудитні події);
- логи контейнерного середовища (kube-audit, контейнерні stdout/stderr, події вузла);
- платформні/ресурсні журнали (події керування ресурсами, зміни політик, мережеві журнали тощо).

У Azure збір з ОС/вузлів стандартизується через AMA+DCR (типи даних, фільтрація, призначення).

Фільтрація на рівні DCR важлива для двох причин: (1) зниження “шуму” й підвищення якості детекції; (2) оптимізація витрат на інжест і зберігання. Цей підхід узгоджується з NIST-логікою: ефективний log management має включати інфраструктуру та процеси, а не лише “факт наявності логів”.

Трейси. Для розподілених застосунків (мікросервіси, API, асинхронні пайплайни) трасування дозволяє корелювати події між сервісами, а не аналізувати їх ізольовано. В Azure доцільно застосовувати Application Insights з підтримкою OpenTelemetry, що дає стандартизований збір телеметрії та підтримує distributed tracing, метрики й аналіз логів.

Таблиця 3.3 – Відповідність сигналів, інструментів і мов запитів у Azure

Сигнал	Основне сховище	Інструментування/збір	Типова аналітика
Metrics (platform)	Azure Monitor Metrics	“zero-config” для багатьох ресурсів	Порогові/динамічні алерти
Metrics (Prometheus)	Azure Monitor Managed Prometheus	Scrape + керований сервіс	PromQL алерти, SLO
Logs	Log Analytics Workspace	AMA + DCR; ресурсні логи	KQL кореляція/детекція
Traces	Application Insights (workspace-based)	OTel distro/SDK/exporter	APM, кореляція з логами

Важливо практично забезпечити кореляцію: trace_id має потрапляти в логи (структуровані), а ключові метрики можуть містити посилання (exemplars) на конкретні трейси – це прискорює RCA та форензіку. [48]

3.3.4 Правила детекції: базові (baseline), кореляційні та SIEM-рівень

Реалізація детекції зазвичай двошарова:

(A) Azure Monitor Alerts – для швидких інфраструктурних сигналів і операційних інцидентів. Azure Monitor підтримує типи алертів: metric alerts, log search alerts, activity log alerts, Prometheus alerts, а також механізми масштабування алертингу і рекомендовані правила для типових ресурсів (зокрема VM, AKS).

На цьому рівні логічно налаштовувати:

- “health” алерти (CPU saturation, memory pressure, disk full);
- алерти доступності (endpoint down, node not ready);

- алерти конфігураційних змін (критичні події activity log);
- Prometheus-алерти для Kubernetes (crashloop, high restart rate, apiserver latency).

(B) Microsoft Sentinel Analytics – для безпекових сценаріїв, кореляції та побудови інцидентів SOC-рівня. Sentinel описує scheduled analytics rules як основний механізм, а також дає майстер створення правил і шаблони (templates).

Ключова перевага – кореляція між джерелами (control-plane + workload) і виведення результату у вигляді інцидентів, сутностей та хантинг-запитів.

Приклади практичних сценаріїв детекції для контролю активності вузлів:

- Ескалація привілеїв у control-plane: нетипове призначення ролі “Owner/Contributor”, створення нових ключів/секретів;
- Ознаки компрометації вузла: запуск нетипових процесів, підозрілі мережеві з’єднання, масове читання/архівація даних;
- Lateral movement: зростання внутрішніх сканувань, RDP/SSH “стрибки” між сегментами;
- Ексфільтрація: аномальні обсяги вихідного трафіку, нетипові експорти зі сховищ/БД.

3.3.5 Оповіщення та автоматизація реагування: Action Groups і SOAR

Оповіщення в Azure Monitor будується навколо дій після спрацювання алерта (наприклад, повідомлення, webhook, ITSM-інтеграції). В Azure Monitor також підтримуються alert processing rules для додавання/приглушення дій, фільтрації та розкладів реакції.

На SIEM-рівні Sentinel надає SOAR-функції: automation rules і playbooks для підвищення ефективності SOC і скорочення часу реакції.

Практично це дає типовий “ланцюжок”: детекція → інцидент → авто-тріаж → ізоляція вузла/блокування облікового запису/ротація ключів → створення задачі → збір артефактів для розслідування.

Наскрізний вимір ефективності оповіщення повинен включати не лише кількість алертів, а й операційні метрики: MTTD/MTTR, частку “false positives”, частоту повторних інцидентів і якість заповнення контексту (ресурс, середовище, власник, критичність). Це напряму пов’язує телеметрію з керованістю безпеки, що відповідає ідеї ISCM як програми з політиками, процедурами та аналізом даних. [49]

3.4 Налаштування контрольних перевірок безпеки: аудит, сканування конфігурацій, оцінка відповідності базовим бенчмаркам

У хмарній інфраструктурі Azure фактичний рівень безпеки визначається не лише наявністю захисних сервісів, а передусім коректністю конфігурацій, повнотою аудит-трейлів та здатністю середовища постійно підтверджувати відповідність вимогам (compliance). Це зумовлює потребу в системі контрольних перевірок, яка працює безперервно: фіксує події, оцінює ресурси відносно політик і бенчмарків, формує рекомендації/попередження та запускає процеси ремедіації. У Microsoft-екосистемі базову методичну основу для таких перевірок задає Microsoft cloud security benchmark (MCSB), який є наступником Azure Security Benchmark (ASB) після ребрендингу у жовтні 2025 року.

3.4.1 Базові бенчмарки та «цільовий профіль» відповідності

Як «опорні» бенчмарки доцільно використати:

- Microsoft cloud security benchmark (MCSB) – набір прескриптивних рекомендацій і контролів для захисту навантажень, даних і сервісів в Azure та мультихмарних середовищах.
- CIS Benchmarks для Microsoft Azure – спільнотно узгоджені настанови щодо безпечних конфігурацій, які часто застосовують як «мінімальний» рівень hardening.

Таблиця 3.4 – Роль бенчмарків у контрольних перевірках

Бенчмарк/рамка	Для чого застосовується	Як інструментується в Azure
MCSB	Єдина «мовна модель» контролів безпеки для Azure/мультихмари	Вбудовані ініціативи Regulatory Compliance та відображення у Defender for Cloud (learn.microsoft.com)
CIS Azure Foundations	Базовий hardening конфігурацій і сервісів	Мапінг контролів у Azure Policy (CIS initiative) (learn.microsoft.com)
NIST SP 800-53	Каталог контролів для формалізації вимог та аудит-оцінювання	Використовується як референс для зіставлення контролів/вимог (csrc.nist.gov)

У практичній реалізації в Azure ці бенчмарки найзручніше «приземляти» через Azure Policy Regulatory Compliance built-ins (вбудовані ініціативи відповідності), які дозволяють бачити контролі та домени відповідності з урахуванням відповідальності сторін (Customer/Microsoft/Shared).

3.4.2 Аудит як «доказовий шар»: що саме журналювати і де зберігати

Контрольні перевірки неможливі без надійного аудиту, який дає відповіді «хто/що/коли/звідки/яку дію виконав». В Azure ключовими джерелами є:

- Azure Monitor Activity Log – платформний журнал подій control plane (створення/модифікації ресурсів, помилки розгортання тощо), який

використовується для перегляду й аудиту операцій та для створення проактивних сповіщень.

- Resource logs конкретних сервісів (data plane/операційні журнали), які збираються через Diagnostic settings і можуть надсилатися до Log Analytics workspace, Event Hubs або Storage.

Практично рекомендовано централізувати аудит у Log Analytics (або сумісному сховищі) для кореляції подій і побудови звітів, а також визначити політики ретенції. Наявність повної «аудит-лінії» є критичною і для комплаєнсу, і для технічного розслідування інцидентів, оскільки порушення в хмарі часто починаються з операцій у control plane (зміни ролей, політик, вимкнення журналювання). [50]

3.4.3 Сканування конфігурацій через Azure Policy: політики, ініціативи, оцінювання та ремедіація

Azure Policy забезпечує масштабне оцінювання відповідності ресурсів організаційним стандартам і дозволяє переходити від «виявлення» до «виправлення» через механізми ремедіації.

Типова схема така: визначення (policy definitions) групуються в ініціативи (initiative definitions), після чого призначаються на рівні management group/subscription/resource group. Для відповідності бенчмаркам використовують Regulatory Compliance built-ins, зокрема ініціативи для MCSB та CIS, де контролі вже згруповані за доменами й відповідальністю сторін.

Важлива перевага Azure Policy – кероване виправлення відхилень. Для політик типу deployIfNotExists або modify можна створювати remediation tasks, які застосовують шаблон розгортання або модифікацію до ресурсів, що є non-compliant.

Процедурно ремедіація виконується через відповідний механізм Azure Policy, що описує кроки створення задачі виправлення для невідповідних ресурсів.

Інвентаризація ресурсів → Оцінювання Azure Policy (policies/initiatives) → Статус compliance (compliant/non-compliant) →(а) Звіт/алерт → (б) Remediation task (modify/deployIfNotExists) → Повторна оцінка → Підтвердження відповідності

Окремо доцільно підкреслити, що для MCSB і CIS існують офіційні сторінки зіставлення контролів у Regulatory Compliance, які показують, як ініціатива Azure Policy відповідає доменам/контролям бенчмарка. [51]

3.4.4 Перевірки «всередині» віртуальних машин: Azure Machine Configuration

Перевірки рівня Azure Policy добре покривають конфігурації ресурсів (NSG, Storage, Key Vault, шифрування тощо), але частина вимог стосується налаштувань ОС і ПЗ на VM/Arc-вузлах (локальні політики, стан служб, параметри безпеки). Для цього в Azure застосовується Azure Machine Configuration (раніше Guest Configuration) – можливість «аудитити або конфігурувати налаштування ОС як код», вбудовано й керовано через Azure Policy.

Це дозволяє, наприклад, виявляти відхилення від мінімальних вимог hardening на рівні ОС та формувати доказову базу для аудитів, де потрібні контролі саме «host-level».

3.4.5 Оцінка уразливостей як частина контрольних перевірок

Конфігураційна відповідність не гарантує відсутності відомих уразливостей у VM або встановлених пакетах. Тому контрольні перевірки мають включати vulnerability scanning для машин. У Defender for Cloud передбачено вмикання сканування вразливостей на рівні підписки й налаштування рішення оцінювання (зокрема через інтеграції з Defender Vulnerability Management).

Таким чином, у єдиному контурі контролю поєднуються: (1) відповідність конфігурацій (policy compliance) і (2) технічний стан оновлень/уразливостей (vulnerability posture). [52]

3.4.6 Зведені показники й комплаєнс-дашборди: Secure Score та Regulatory Compliance

Для управлінського контролю важливо мати інтегральні метрики. Secure score у Microsoft Defender for Cloud агрегує знахідки в єдиний показник, що дає швидку оцінку поточного рівня захищеності та пріоритизацію робіт.

Окремо Regulatory compliance dashboard у Defender for Cloud показує підключені стандарти, їхні контролі та пов’язані оцінювання, що відображають рівень відповідності.

При цьому за замовчуванням у дашборді відображається саме Microsoft cloud security benchmark, а інші стандарти додаються явно.

Таблиця 3.5 – Рекомендований набір контрольних перевірок і артефактів

Напрям перевірок	Інструмент Azure	Типовий артефакт результату
Аудит подій control plane	Activity Log (Azure Monitor) (learn.microsoft.com)	Трейл змін ресурсів/деплойментів, алерти на критичні операції
Логи сервісів і компонентів	Diagnostic settings + Resource logs (learn.microsoft.com)	Централізовані журнали для кореляції та звітності
Конфігураційна відповідність	Azure Policy + initiatives (learn.microsoft.com)	Compliance state, список non-compliant ресурсів
Автовиправлення	Remediation tasks (learn.microsoft.com)	Журнал виконання remediations, повернення до compliant
Hardening на рівні ОС	Azure Machine Configuration (learn.microsoft.com)	Звіт про відповідність конфігурацій усередині VM
Сканування уразливостей	Defender for Cloud VA (learn.microsoft.com)	Findings по CVE/пакетах, план ремедіації
Комплаєнс-репортинг	Regulatory compliance dashboard (learn.microsoft.com)	Статус контролів стандарту, прогрес у часі

Нналаштування контрольних перевірок у Azure доцільно розглядати як «замкнений цикл»: бенчмарк → політики/ініціативи → аудит і збір доказів → автоматизоване оцінювання → пріоритизація (secure score) → ремедіація → повторна валідація. Такий підхід зменшує залежність від ручних аудитів, скорочує час виявлення помилок конфігурації та підвищує відтворюваність безпеки при масштабуванні інфраструктури.

3.5 Експериментальна перевірка: тестові сценарії атак/інцидентів, навантажувальне тестування, аналіз спрацювань

Експериментальна перевірка в межах кваліфікаційної роботи має підтвердити, що спроєктована система захисту та моніторингу в Azure не є «декларативним набором сервісів», а демонструє відтворювану здатність: (1) виявляти типові інциденти та аномальні дії в control plane і data plane, (2) забезпечувати керований процес реагування, (3) зберігати прийнятний вплив на продуктивність і витрати. Методично доцільно будувати експерименти як набір сценаріїв із чітко сформульованими умовами, очікуваною телеметрією, правилами детекції та критеріями успішності, що узгоджується з підходами

NIST до організації реагування та управління ризиками інцидентів (NIST SP 800-61r3).

3.5.1 Тестовий стенд і спостережуваність як передумова експерименту

Для коректності експериментів важливо забезпечити «замкнутий контур спостережуваності» (telemetry pipeline): події мають виникати у ресурсах, збиратися через налаштування діагностики та доходити до централізованого сховища/аналітики, де працюють правила виявлення та автоматизація реагування.

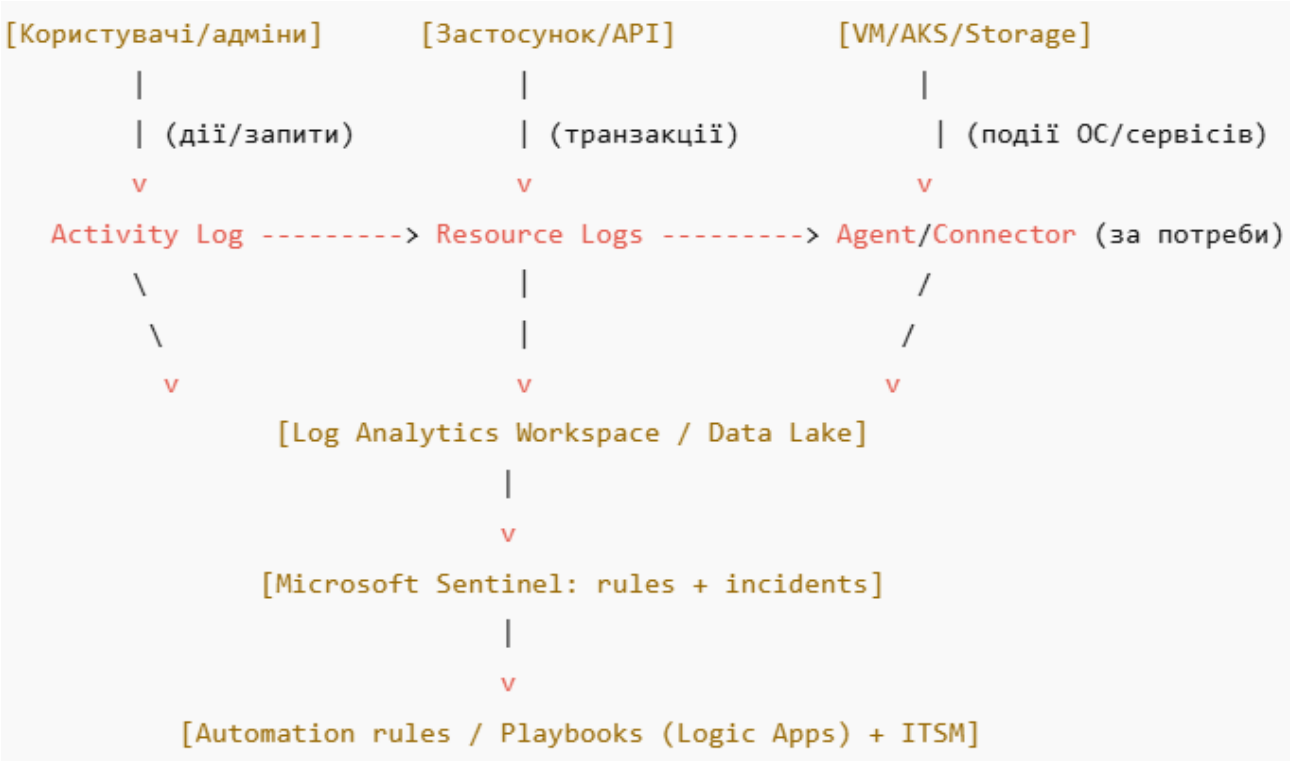

Рисунок 3.4 – Логічна схема тестового стенда та каналів телеметрії (Azure)

У контексті Azure базовими джерелами є:

- Azure Monitor Activity Log як платформний журнал подій control plane (зміни ресурсів, помилки розгортання тощо), який можна переглядати та використовувати для проактивних сповіщень.
- Resource logs сервісів (data plane / операційні журнали), які не збираються за замовчуванням і потребують створення Diagnostic settings для кожного ресурсу.

Для валідації «готовності стенда» доцільно виконати контрольні дії (створити/змінити ресурс, змінити роль, виконати операцію зі сховищем) і перевірити, що відповідні записи з’явилися у централізованому сховищі та доступні для запитів KQL (Kusto Query Language), який застосовується в Azure Monitor / Microsoft Sentinel.

3.5.2 Методика побудови сценаріїв атак/інцидентів

Сценарії варто класифікувати щонайменше у дві групи:

- Control plane інциденти (IAM, політики, вимкнення логування, зміни мережевих правил) – небезпечні тим, що масштабуються на всю інфраструктуру, оскільки контрольна площина керує багатьма ресурсами одночасно.
- Data plane інциденти (підозріла активність на VM/AKS, аномальна робота застосунку, ексфільтрація з Storage/DB) – проявляються у телеметрії хостів/сервісів і часто потребують кореляції сигналів.

Щоб сценарії були системними, зручно опиратися на MITRE ATT&CK як таксономію поведінки противника та на data sources MITRE як підказку, які саме джерела даних потрібні для детекції технік.

Таблиця 3.6 – Шаблон опису тестового сценарію

Поле	Зміст
Мета сценарію	Який ризик/контроль перевіряємо (CIA/контрольна вимога)
Контекст	Які ресурси залучені (RG/VNet/VM/AKS/Storage/Key Vault)
Стимул (інцидент)	Опис події на рівні дії (без експлойт-інструкцій)
Очікувана телеметрія	Які логи/метрики/трейси мають спрацювати
Правило детекції	Sentinel scheduled rule / Azure Monitor alert / Defender сигнал
Критерій успіху	Інцидент створено, кореляція подій, алерт/оповіщення
Порогові значення	SLO для детекції (наприклад, ≤ 5 хв)
Післядія	Ремедіація/ізоляція/відкат (playbook/runbook)

3.5.3 Набір тестових сценаріїв (безпечна емуляція)

Нижче подано типовий набір сценаріїв, який не містить інструкцій з експлуатації уразливостей, але є достатнім для перевірки моніторингу та реагування.

Таблиця 3.7 – Прикладний перелік сценаріїв атак/інцидентів і джерел телеметрії

№	Сценарій	Тип	Очікувані джерела	Очікуваний результат
1	2	3	4	5
S1	Підозріла спроба призначення привілейованої ролі (RBAC)	Control plane	Activity Log / журнали IAM	Створений інцидент; оповіщення SOC; фіксація суб’єкта і scope
S2	Створення нового ключа/секрету або нетиповий доступ до Key Vault	Control plane + Data	Resource logs Key Vault + Activity Log	Алерт; кореляція з акаунтом/джерелом; рекомендація блокування
S3	Вимкнення/зміна Diagnostic settings на критичному ресурсі	Control plane	Activity Log + конфіг-події	Алерт «tampering», пріоритет високий
S4	Різке розширення мережевого периметра (відкриття правила доступу)	Control plane	Activity Log	Алерт + вимога RFC/approval; можливий авто-відкат
S5	Аномальний обсяг читання/експорту даних зі сховища	Data plane	Resource logs Storage + метрики	Інцидент «data exfil suspicion», зв’язка з IP/ідентичністю

Продовження таблиці 3.7

1	2	3	4	5
S6	Підозрілий процес/скрипт на VM (зовнішній агент/EDR)	Data plane	VM Security events/Syslog + EDR телеметрія	Інцидент; рекомендація ізоляції хоста/облікового запису
S7	Підозрілі невдалі входи/MFA fatigue (ідентичності)	Control plane	журнали автентифікації (Entra ID)	Алерт, блокування/вимога reset, підвищення ризику
S8	Деградація сервісу під навантаженням + сплеск помилок	Availability	метрики/трейси/лог-помилки	Алерт SLO; RCA через traces; рішення про масштабування

Для практичної реалізації детекції в Microsoft Sentinel застосовуються scheduled analytics rules, які будуються на KQL-запиті та мають параметри розкладу, lookback, пороги та обробку сутностей.

Для частини control plane подій доцільно використовувати Azure Monitor activity log alerts, які реагують на появу конкретної події в Activity Log (наприклад, видалення ресурсів, призначення ролей, створення/видалення).

3.5.4 Фіксація та аналіз спрацювань: які поля є ключовими

Аналіз спрацювань слід проводити не лише за фактом «алерт є/немає», а й за якістю контексту: чи містить інцидент достатні сутності (account, IP, resource, operation), чи можлива швидка реконструкція ланцюга подій. Практично це означає:

- перевірку повноти журналювання (чи не пропущені resource logs через відсутні diagnostic settings);
- перевірку коректності часової кореляції (time generated, ingestion time, затримки доставки);
- перевірку точності фільтрів правил (чи не породжують вони «шумний» потік інцидентів);
- перевірку працездатності автоматизації.

```
Подія -> Потрапляє у Log Analytics -> KQL-запит scheduled rule ->
   якщо умова істинна -> Alert -> Incident ->
      Automation rule (маршрутизація/пріоритет/умови) ->
         Playbook (enrichment/containment/ITSM ticket) ->
            Запис у аудиті та часові мітки MTTD/MTTR
```

Рисунок 3.5 – Логіка оцінювання спрацювання правила в Sentinel

Для автоматизації реагування в Sentinel застосовуються automation rules і playbooks, які будуються на Azure Logic Apps та можуть запускатися на інциденти/алерти.

3.5.5 Навантажувальне тестування як перевірка «непомітності» моніторингу

Окремий блок експерименту – навантажувальне тестування, яке має показати:

- як система поводиться при пікових навантаженнях;
- чи зберігається якість телеметрії (втрати логів, затримки інжесту);
- який вплив мають агенти/логування/трасування на latency та ресурсні показники.

Для Azure практичною базою є сервіс Azure Load Testing, який має власні можливості моніторингу та інтеграцію з Azure Monitor (діагностичні налаштування, логи й метрики).

Рекомендована схема:

- Етап A (baseline): навантаження без «посиленого» логування/трейсингу (мінімальні необхідні логи).
- Етап B (observability on): навантаження з увімкненими ресурсними логами, трасуванням застосунку (за потреби), підвищеним рівнем audit.
- Етап C (security on): додатково увімкнені security-сигнали (EDR/Defender, правила Sentinel, автоматизація).

Таблиця 3.8 – Метрики для порівняння під час load-тестів

Група	Метрика	Як фіксується
Продуктивність	p95/p99 latency, throughput, error rate	метрики застосунку + серверні метрики
Ресурси	CPU/RAM/IO, network	Azure Monitor metrics (Microsoft Learn)
Телеметрія	обсяг інжесту, затримка інжесту, пропуски	workspace ingestion/таблиці логів
Детекція	кількість алертів, час до інциденту	Sentinel incidents timeline
Стабільність	rate limiting, квоти, деградація	алерти Azure Monitor/Sentinel

У ході тестів важливо контролювати, чи не створюють правила детекції надмірного навантаження на сховище логів і кореляційний механізм (зокрема, важкі KQL-запити, широкий lookback). На практиці це вирішується оптимізацією запитів і вибором правильного scope з урахуванням обмежень та накладних витрат, особливо якщо дані рознесені між регіонами або робочими просторами.

3.6 Оцінювання результатів: точність/повнота детекції, час реагування, вплив на продуктивність, економічна доцільність

Оцінювання результатів експериментів має бути формалізованим, щоб забезпечити порівнюваність (між сценаріями, між конфігураціями «baseline vs security», між середовищами). Доцільно застосувати чотири групи показників: (1) ефективність детекції, (2) швидкість реагування, (3) вплив на продуктивність, (4) економічна доцільність.

3.6.1 Точність і повнота детекції (Precision/Recall) та матриця помилок

Для кожного тестового сценарію формується «еталон подій» (ground truth): ми знаємо, що сценарій був ініційований у певний момент часу та мав породити певні сигнали. Далі для кожного правила оцінюємо:

- TP (true positive): правило спрацювало, коли сценарій реально відбувся.
- FP (false positive): правило спрацювало без сценарію.
- FN (false negative): сценарій відбувся, але правило не спрацювало.
- TN (true negative): відсутність спрацювання за відсутності сценарію (у практиці безпеки часто оцінюється опосередковано).

	Факт: інцидент	Факт: немає інциденту
Спрацювало	TP	FP
Не спрацювало	FN	TN

Рисунок 3.6 – Матриця помилок для оцінювання детекції

Тоді:

Precision = TP / (TP + FP) – «наскільки спрацювання точні».

Recall = TP / (TP + FN) – «наскільки повно ми виявляємо інциденти».

За потреби використовується F1 = 2 (PrecisionRecall)/(Precision+Recall)** як компромісний показник.

Практичний зміст: для SOC критично уникати FP-шуму (інакше виникає alert fatigue), але FN можуть бути значно небезпечнішими для бізнесу. Тому цілі Precision/Recall варто встановлювати диференційовано: для high-impact сценаріїв допускається менший Precision (більше перевірок), але прагнуть до вищого Recall; для «масових» правил навпаки – високий Precision, інакше перевантажиться чергова зміна.

3.6.2 Час реагування: MTTD/MTTR та контрольні точки процесу

NIST SP 800-61r3 підкреслює інтеграцію реагування на інциденти з функціями NIST CSF 2.0 та орієнтацію на керованість і ефективність процесу.

У практичній метриці це трансформується в часові показники:

- MTTD (Mean Time To Detect) – середній час від початку інциденту до моменту детекції (поява інциденту в Sentinel/алерту).
- MTTA (Mean Time To Acknowledge) – час до підтвердження черговим аналітиком.
- MTTC (Mean Time To Contain) – час до стримування (блок акаунта, ізоляція хоста, відкат правила FW).
- MTTR (Mean Time To Recover/Respond) – час до відновлення/завершення реагування.

Щоб метрики були точними, потрібно уніфікувати «timestamp-події»:

t0 – час ініціації сценарію;

t1 – час появи даних у Log Analytics;

t2 – час спрацювання правила / створення інциденту;

t3 – час запуску автоматизації;

t4 – час підтвердження/закриття.

Microsoft Sentinel надає механізми scheduled rules та автоматизації, що дозволяє прямо вимірювати t2–t4 (інцидентний таймлайн), а Azure Monitor alerts – t2 для control plane подій.

3.6.3 Вплив на продуктивність: технічні та операційні показники

Вплив на продуктивність у хмарі проявляється у трьох площинах:

- вплив на застосунок (latency/error rate через трасування/логування/агенти);
- вплив на інфраструктуру (CPU/RAM/IO на VM/нодах);
- вплив на аналітичний контур (затримки інжесту, важкі запити, перевищення лімітів).

Azure Monitor Metrics збирає числові метрики у time-series форматі і є базою для порівняння ресурсного профілю до/після увімкнення моніторингу та security-компонентів.

Окрема практична пастка – «розподілений» лог-ландшафт: якщо запити або кореляції охоплюють багато робочих просторів/регіонів, можуть виникати додаткові накладні витрати, а для деяких режимів діють обмеження.

Таблиця 3.9 – Показники продуктивності та пороги інтерпретації

Показник	Як вимірюється	Небажаний симптом
p95/p99 latency	load-test + метрики застосунку	зсув p95/p99 > X% після ввімкнення телеметрії
CPU/RAM на VM	Azure Monitor metrics	сталі піки після інсталяції агентів
Інджест-затримка	t1–t0, t2–t1	затримки, що порушують SLO детекції
Обсяг логів	GB/day у workspace	непропорційне зростання витрат/шуму
Важкість KQL	час виконання запитів	уповільнення кореляції/інцидентів

3.6.4 Економічна доцільність: модель витрат і ефекту

Економічна оцінка повинна спиратися на прозору структуру витрат і на очікуваний «збиток, якого вдалося уникнути» (risk reduction). Для Azure моніторингу ключова частина витрат пов’язана з інжестом і зберіганням логів у Log Analytics; Microsoft прямо зазначає, що для більшості впроваджень саме інжест і ретенція є найбільш значущими складниками вартості.

Компоненти витрат доцільно формалізувати як:

- C_logs – витрати на інжест/ретенцію/експорт логів (плани інжесту, тривалість зберігання, довготривала ретенція), що описано в документації Azure Monitor Logs щодо розрахунку вартості.
- C_security – витрати на security-функції (наприклад, Defender for Cloud відповідно до обраних планів; орієнтиром є офіційна сторінка pricing).
- C_ops – операційні витрати (час SOC/DevSecOps на супровід правил, triage FP, розвиток playbooks).

Ефект (benefit) можна оцінювати як:

- B_risk = зменшення очікуваного збитку: Δ(Probability * Impact) для ключових сценаріїв (витік даних, простій, компрометація облікових).
- B_efficiency = економія часу реагування та зменшення простою завдяки кращому MTTD/MTTR.

Таблиця 3.10 – Спрощена фінансова модель доцільності

Елемент	Позначення	Як оцінити в межах роботи
Витрати на логи	C_logs	обсяг GB/day × план інжесту × ретенція (Microsoft Learn)
Витрати на security	C_security	ввімкнені плани Defender/інші рішення (Microsoft Azure)
Операційні витрати	C_ops	людино-години на тиждень × ставка (умовно)
Зменшення ризику	B_risk	сценарії × (зміна ймовірності/шкоди)
Економія часу	B_efficiency	ΔMTTD/ΔMTTR → Δdowntime → економічний ефект

Ключовий практичний висновок: економічна доцільність у хмарному моніторингу майже завжди залежить від дисципліни керування даними (які таблиці збираємо, який рівень деталізації, які retention-політики, чи є фільтрація «шумних» логів). Саме тому в рекомендаціях до впровадження (п. 3.7) лог-економіка повинна бути вбудована в архітектуру, а не додана постфактум.

3.7 Рекомендації щодо впровадження та масштабування в реальній інфраструктурі

Пілотна реалізація в лабораторному стенді часто має відмінності від «бойового» середовища: більша кількість підписок, різні команди, неоднорідні системи, регіональна рознесеність, вимоги комплаєнсу, обмеження по бюджету. Тому рекомендації слід формувати як набір практик, що забезпечують масштабованість, керованість і контрольовані витрати.

3.7.1 Поетапне впровадження: від пілоту до промислового контуру

Рекомендована послідовність:

- MVP-контур безпеки: мінімальний набір логів і правил для найбільш критичних сценаріїв (IAM tampering, зміни мережевого периметра, доступ до секретів, видалення ресурсів).
- Розширення покриття: додавання data plane логів (Storage/DB), EDR-сигналів для VM/вузлів, SLO-орієнтований моніторинг.
- Автоматизація: маршрутизація інцидентів, enrichment, playbooks для containment.
- Оптимізація витрат і якості: зменшення FP, скорочення обсягу логів, оптимізація KQL.

На практиці основу «вмикання телеметрії» становлять Diagnostic settings, які дозволяють спрямовувати resource logs/metrics/activity log у потрібні призначення; їх слід стандартизувати як політику конфігурації для всіх типових ресурсів.

3.7.2 Централізація та архітектура робочих просторів: баланс між єдністю та регіональністю

Для великої організації важливо вирішити, як організувати Log Analytics workspaces і доступ до них:

- централізований workspace (спрощує кореляцію, єдині правила й дашборди);
- рознесені workspaces за середовищами/регіонами (краще відповідають ізоляції та комплаєнсу).

При цьому треба враховувати, що запити/кореляція через багато регіонів можуть створювати накладні витрати, і документація Azure Monitor прямо попереджає про можливий overhead для запитів між регіонами.

Якщо потрібні cross-workspace запити, слід пам’ятати про обмеження та коректно вибирати механізм виконання (implicit/explicit), оскільки існують ліміти на кількість ресурсів у таких запитах.

Практична рекомендація: для SOC-кореляції з низькими затримками тримати критичні security-дані в одному «операційному» контурі (або в мінімальній кількості узгоджених workspaces), а довготривале зберігання/архівування – відокремлювати.

3.7.3 Побудова детекцій як «продукту»: життєвий цикл правил і вимірювання якості

Microsoft Sentinel scheduled analytics rules мають параметри розкладу, lookback і логіку сутностей, що дозволяє будувати стандартизований каталог детекцій.

Рекомендований життєвий цикл правила:

- розробка (KQL + логіка порогів);
- тестування на історичних даних;

- контрольована експлуатація (пілот на частині середовища);
- вимірювання Precision/Recall (за інцидентами/еталонними подіями);
- оптимізація (зменшення FP, уточнення контексту);
- регулярний review (зміни середовища, нові сервіси, нові тактики атак).

Для планування покриття та візуалізації детекцій доцільно мапити правила на MITRE ATT&CK (техніки/тактики) та використовувати інструменти на кшталт ATT&CK Navigator для аналізу «прогалин».

3.7.4 Автоматизація реагування: від «сповіщення» до «керованого containment»

Зріла система реагування мінімізує ручні дії там, де це безпечно та відтворювано:

- automation rules у Sentinel застосовуються для маршрутизації/пріоритизації інцидентів та запуску дій за умовами.
- playbooks на базі Logic Apps реалізують enrichment, взаємодію з ITSM, блокування акаунтів/ключів, ізоляцію ресурсів (за політиками організації) та документування.

Практична рекомендація: автоматизацію впроваджувати поетапно (спочатку enrichment і створення тикета, далі – напівавтоматичні containment-дії з approval, і лише потім – повністю автоматичні дії для чітко визначених сценаріїв).

3.7.5 Керування витратами на логи як обов'язкова частина масштабування

Оскільки інжест і ретенція логів є головними драйверами вартості, потрібні політики:

- мінімально достатній перелік категорій логів;
- фільтрація «шумних» подій на рівні збору (де це дозволено) або на рівні правил;
- розумна ретенція (коротша для high-volume, довша для audit-критичних);
- регулярні ревізії обсягу даних і оптимізація запитів.

Документація Azure Monitor Logs описує, як саме обчислюється вартість даних у Log Analytics workspaces і які опції впливають на витрати.

3.7.6 Валідація у виробництві: регулярні вправи та «контрольні інциденти»

Рекомендації щодо підтримки ефективності після впровадження:

- регулярні table-top exercises (імітаційні розбори) для SOC/DevOps;
- квартальні/піврічні контрольні тести ключових сценаріїв (IAM tampering, data access anomalies, availability incidents);
- перегляд runbooks/playbooks за результатами «lessons learned», що узгоджується з рекомендаціями NIST щодо безперервного вдосконалення процесів реагування.

3.8 Висновки до розділу 3

У розділі 3 виконано практичну реалізацію запропонованих підходів до захисту даних і контролю активності вузлів у хмарному середовищі та сформовано формалізовану методику оцінювання їх ефективності. Як цільову платформу обрано Microsoft Azure, оскільки вона забезпечує цілісну інтеграцію ланцюга «ідентичність → криптографічні сервіси та ключі → журналювання/телеметрія → детекція → реагування», а також надає референсні рамки архітектування і керованості (Well-Architected, Cloud Adoption Framework, Landing Zone) та розвинену екосистему безпеки й спостережуваності (Entra ID, Key Vault/Managed HSM, Azure Monitor/Log Analytics, Defender for Cloud, Sentinel). Одночасно зафіксовано ключові обмеження вибору Azure: ризики vendor lock-in, квоти/ліміти, вартість спостережуваності та регіональні/комплаєнс-аспекти, що визначають межі дослідження й вимоги до проєктних абстракцій (IaC, стандартизована телеметрія, політики ретенції).

Показано, що захист даних у хмарі має будуватися як керований набір політик і контролів, узгоджених для даних «у русі», «у спокої» та, за потреби, «у використанні». Практична реалізація орієнтується на примусовий TLS/HTTPS для каналів, на використання керування ключами через Azure Key Vault / Managed HSM (з підтримкою CMK там, де це виправдано), а також на централізоване керування секретами із мінімізацією статичних облікових даних (керовані ідентичності). Мережева сегментація (ізоляція зон довіри, hub-spoke, приватні кінцеві точки) розглянута як базовий механізм зменшення площі атаки й обмеження lateral movement. Окремо обґрунтовано, що резервне копіювання повинно гарантувати відновлюваність у межах RPO/RTO і містити анти-руйнівні механізми (soft delete, immutable, MUA), оскільки саме стійкість до видалення/шифрування recovery points є критичною в сценаріях ransomware та insider.

Реалізовано підхід до контролю активності вузлів як безперервне спостереження на основі «трьох сигналів» (метрики, логи, трейси) з розділенням control plane та data plane. Запропоновано централізований збір і маршрутизацію телеметрії (через правила збору та єдині сховища/аналітику), що забезпечує кореляцію подій, керований алертинг і можливість переходу від операційного моніторингу до SOC-рівня детекції. Показано доцільність двошарового підходу до виявлення: швидкі інфраструктурні спрацювання (Azure Monitor) та безпекова кореляція/інцидент-менеджмент (Sentinel), а також сформовано логіку оповіщення й автоматизації реагування (action groups, playbooks/automation rules), що зменшує час до containment і підвищує відтворюваність дій.

Контрольні перевірки безпеки сформовано як «замкнений цикл» комплаєнсу: бенчмарк → політики/ініціативи → збір доказів → автоматизоване оцінювання → ремедіація → повторна валідація. Підкреслено, що поєднання аудит-трейлу (Activity Log, ресурсні логи), конфігураційного контролю (Azure Policy), гостевих перевірок (Machine Configuration) і сканування уразливостей (Defender for Cloud) створює практично придатний механізм підтримки безпечного стану середовища при масштабуванні.

Експериментальна перевірка побудована як набір відтворюваних сценаріїв інцидентів для control plane і data plane з визначеними очікуваними джерелами телеметрії, правилами детекції та критеріями успішності. Окремо передбачено навантажувальне тестування у режимах «baseline → observability on → security on», що дозволяє оцінити не лише здатність до виявлення, але й «непомітність» моніторингу з позиції затримок, ресурсного профілю та стабільності інжесту.

Оцінювання результатів формалізовано через метрики точності/повноти (Precision/Recall/F1), часові показники процесу реагування

(MTTD/MTTA/MTTC/MTTR), вплив на продуктивність (p95/p99, ресурси, затримка інжесту, важкість запитів) та економічну модель доцільності, де головним фактором витрат виступають інжест і ретенція логів, а головним ефектом — зниження ризику та скорочення простоїв за рахунок швидшого виявлення і containment.

Сформульовані рекомендації визначають шлях перенесення лабораторного рішення в реальну інфраструктуру: поетапне впровадження від MVP до промислового контуру, продумана архітектура робочих просторів і доступів, управління життєвим циклом детекцій як «продукту», контрольована автоматизація реагування та обов’язкове керування лог-витратами через політики збору/фільтрації/ретенції. Запропоновано також регулярні вправи та контрольні інциденти як механізм підтримки ефективності після впровадження.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Метою кваліфікаційної роботи магістра є дослідження методів захисту відомих хмарних платформ. Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної та безпечної роботи колективу працівників, у тому числі фахівців, які виконують роботи з підвищення ефективності контролю доступу до приміщень, необхідно організувати належні умови праці. Роботодавець (керівник організації та уповноважені посадові особи) несе відповідальність за створення безпечних і здорових умов праці, організацію системи охорони праці та дотримання вимог законодавства з охорони праці.

На робочих місцях працівників, що використовують комп'ютерну техніку, обов'язково забезпечується виконання Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями (НПАОП 0.00-7.15-18), затверджених наказом Мінсоцполітики України від 14.02.2018 № 207 [73]. Зазначені вимоги охоплюють організацію робочого місця та робочого середовища, ергономіку, режими праці й відпочинку, а також вимоги до обладнання та його розміщення.

Відстані до екрана, розміщення обладнання, організація робочого простору та інші гігієнічні параметри слід обґрунтовувати чинними вимогами НПАОП 0.00-7.15-18 [73] з урахуванням того, що ДСанПіН 3.3.2.007-98 втратив чинність у 2025 році внаслідок дерегуляційних змін [74], що також відображено у картці відповідного документа в офіційній базі нормативних актів [75].

Параметри виробничого середовища (рівні шуму та мікроклімат) мають відповідати чинним санітарним нормам: рівні шуму — ДСН 3.3.6.037-99 [76], а показники мікроклімату в межах робочої зони — ДСН 3.3.6.042-99 [77]. Для виконання робіт із застосуванням дисплейних пристроїв важливим з точки зору охорони праці є також забезпечення достатньої величини природного та штучного освітлення, яке слід проєктувати/оцінювати відповідно до ДБН В.2.5-28:2018 «Природне і штучне освітлення» [78].

Приміщення, у яких розміщені робочі місця операторів та/або серверне обладнання, повинні відповідати вимогам пожежної безпеки та бути забезпечені необхідними системами протипожежного захисту. Проєктні та технічні вимоги до систем протипожежного захисту (пожежна сигналізація, оповіщення, а за необхідності — автоматичне пожежогасіння) визначаються ДБН В.2.5-56:2014 [79], а загальні вимоги пожежної безпеки об’єктів будівництва — ДБН В.1.1-7:2016 [80]. Організаційні та експлуатаційні вимоги (утримання шляхів евакуації, заборона захаращення проходів, порядок утримання приміщень, загальні обов’язки щодо пожежної безпеки тощо) регламентуються Правилами пожежної безпеки в Україні [81]. Проходи до засобів пожежогасіння та елементів протипожежного захисту мають бути вільними та доступними.

Приміщення повинні бути забезпечені вогнегасниками згідно з Правилами експлуатації та типовими нормами належності вогнегасників (наказ МВС України від 15.01.2018 № 25) [82]. Технічне обслуговування вогнегасників виконується відповідно до ДСТУ 4297:2004 [83].

Електроживлення комп’ютерної техніки та периферійних пристроїв необхідно виконувати окремими груповими лініями із застосуванням трипровідної мережі (L, N, PE) із обов’язковим захисним провідником PE; забороняється використання N як захисного провідника PE та будь-які рішення, що порушують розділення робочого і захисного провідників. Загальні вимоги до улаштування електроустановок, вибору провідників,

захисного заземлення та апаратів захисту визначаються Правилами улаштування електроустановок (ПУЕ), затвердженими наказом Міненерговугілля України від 21.07.2017 № 476 [84], а вимоги безпечної експлуатації електроустановок споживачів — відповідними правилами, затвердженими наказом Мінпраці України від 09.01.1998 № 4 [85]. У приміщеннях із значною кількістю комп’ютерів доцільно передбачати аварійне (резервне) вимикання електроживлення приміщення (крім освітлення) відповідно до проєктних рішень та вимог електробезпеки [84–85]. Комп’ютери повинні підключатися лише через справні штепсельні з’єднання та розетки заводського виготовлення; застосування тимчасових/нештатних перехідників і підключення до двопровідної мережі не допускається [84–85].

Організація робочого місця фахівця має забезпечувати відповідність ергономічним вимогам: параметри робочої поверхні, крісла, зони досяжності, розміщення пристроїв введення-виведення та монітора мають відповідати ДСТУ 8604:2015 [86] і вимогам НПАОП 0.00-7.15-18 [73]. Розміщення принтера та іншого периферійного обладнання має забезпечувати добру видимість екрана, зручність керування та відсутність вимушених поз у межах моторного поля працівника [73], [86].

4.2 Безпека в надзвичайних ситуаціях

4.2.1 Міжнародний тероризм

Терор (лат. terror – страх, жах) – має ознаку «усувати», «закривати». Ця обставина і визначає терор як особливу форму політичного насильства, що характеризується жорстокістю, цілеспрямованістю й уявленою ефективністю. Ці особливості визначили широке використання терору упродовж людської історії як засобу політичної боротьби в інтересах держави, організацій чи окремих угруповань. Безпосередньо сам факт привселюдної страти кримінальних чи політичних злодіїв, чи процес «аутодафе» в період

середньовікової інквізиції, є класичною формою терору в інтересах держави чи католицької церкви.

Правовою основою боротьби з міжнародним тероризмом є «Декларація про заходи для ліквідації міжнародного тероризму», що затверджена на 49-й сесії Генеральної асамблеї ООН (резолюція 49/60 від 9 грудня 1994 р.)

Цей документ встановлює принципи відносин світової спільноти і програму заходів з метою ліквідації такого огидного суспільного явища, як міжнародний тероризм, а також встановлює подальше співробітництво між державами для невідкла-деної ліквідації будь-яких форм і проявів терористичної діяльності. Характерним для розвитку світової спільноти є те, що наявність лідера (провідної країни чи провідної сили) народжує відповідну реакцію – формування нижчого за рангом (рівнем) іншого лідера (іншої країни чи іншої провідної сили). Має місце формування біполярності, виникають реалії антагонізму на різних рівнях світового суспільства, в т.ч. суперечності на рівні «держава»↔«держава», «держава»↔«внутрішня організація» (організація зовнішня), «держава»↔«партія» та ін. Крім того, у світовій практиці мають місце комбіновані види із вищезгаданих «пар», з яких формуються інші групи (сили), в т.ч. політичні, злочинні та ін. відповідні сили чи угруповання. На другому етапі формування ці сили (групи) шукають собі відповідні «ніші» існування; економічну, політичну, наукову та інші види підтримок; формують свої озброєні сили, відповідні професійні кадри, джерела озброєння, територію знаходження тощо. При цьому використовуються всі «блага» цивілізації особистого розвитку і поширення впливу на світову спільноту.

Міжнародний тероризм, створюючи свій плацдарм, може викликати кризи (системні) в світовій, моральній, політичній, економічній системі відносин і зруйнувати та усунути всі передумови розвитку світової спільноти.

В Україні, за даними служби безпеки, за останні два роки скоєно понад 560 злочинів терористичного характеру, внаслідок цього 90 осіб (із них 15

представників владних структур) загинуло. В Україні зростає активність міжнародних терористичних організацій, насамперед із країн Ближнього Сходу («Хезболах», «Абу Ніджаль», «Хамас», «Брати мусульмани»), які прагнуть використати територію України для транзиту своїх бойовиків до країн західної Європи, підготовки терористичних акцій.

Головними принципами попередження та боротьби з міжнародним тероризмом має стати постійне удосконалення відповідної законодавчої бази, співробітництво з правоохоронними організаціями, консолідація з іншими країнами й організація напрямів запобігань поширенню будь-яких терористичних організацій і угруповань.

Терористичний акт не має безпосередніх можливостей досягнення оголошеної кінцевої мети і звичайно складається з таких елементів: насильницька дія у різноманітних її формах, політичний мотив в основі здійснення самого терористичного акту; сам акт спрямовано проти осіб, організацій, націй, національностей і меншин, державних інститутів чи їх представників з метою їх залякування чи виконання окремих вимог. Терор щодо націй, етнічної, расової чи релігійної групи, що здійснюється для її повного чи часткового усунення, розглядається світовою спільнотою вже як акт геноциду.

Варіанти комбінацій за спрямованістю суб'єкт—об'єкт здійснення терористичного акту багатоспрямовані, тому важко дати універсальне визначення «терору». Проте деякі критерії певної класифікації можна встановити:

- індивідуальний, організований терор і терор як політика держави;
- терор як метод внутрішньополітичної боротьби і терористичні акти міжнародного характеру.

4.2.2 Структура системи БЖД

Поняття «життєдіяльність» стосується тільки людини. Людина живе і працює в безпосередньому зв’язку з навколишнім середовищем.

Життєдіяльність (ЖД) – це складна фізіологічна система, яка має назву «система ЖД».

Системою називають сукупність взаємозв’язаних елементів, функціонування яких спрямоване на досягнення певної загальної мети.

Система ЖД складається із взаємопов’язаних елементів: життя, діяльності людини, навколишнього середовища, – і має підтримувати комфортне та безпечне існування людини, забезпечити сталий розвиток людства.

Розглянемо характеристики елементів системи ЖД.

Життя – це форма існування матерії, яка характеризується обміном речовин, здатністю до розмноження і розвитку, вмінням пристосовуватись до навколишнього середовища.

Людина – вища форма розвитку живої матерії, і її існування – дуже складний процес, що не тільки підтримує її фізіологічний стан, але й задовольняє духовні потреби. Крім того, на життя людини суттєво впливають умови проживання та праці, медичний догляд і багато інших факторів, що виникають завдяки діяльності самих людей.

Діяльність – це специфічна форма ставлення людей до навколишнього середовища та одне до одного, яка має задовольняти потреби та інтереси людини. Це соціальна категорія, нерозривно зв’язана із суспільством. Тільки завдяки діяльності людини створено всі блага, які має людство.

Основні види діяльності такі:

- виробнича;
- наукова;
- мистецька;
- освітня.

Однією із специфічних форм діяльності людини є праця – перша й основна умова існування людини (людства).

Праця – цілеспрямована діяльність людини, у процесі якої вона впливає на природу і використовує її з метою виробництва матеріальних та інших благ, необхідних для задоволення своїх потреб.

Потреби – це необхідність для людини того, що забезпечує її існування і самозабезпечення (фізіологічне, матеріальне, соціальне, духовне та ін.).

Навколишнє середовище (довкілля) або середовище існування – це все, що оточує людину впродовж її життя. Навколишнє середовище, у свою чергу, поділяють на такі види:

- природне середовище;
- штучне середовище.

Природне середовище (біосфера) – це частина Землі і простору навколо неї, де зосереджено все живе. Біосфера включає:

- атмосферу (газоподібна частина);
- гідросферу (рідка водна частина);
- літосферу (тверда частина).

На ЖД людей найбільше впливає частина біосфери від поверхні Землі вглиб на 15–20 км і до висоти 20–22 км, де починається озоновий шар. Природне середовище є джерелом природних ресурсів для існування людини: повітря, води, деревини, корисних копалин, ґрунту та ін.

Штучне середовище – це складова довкілля, створена людством за тривалий час його існування. Штучне середовище умовно можна поділити на два види:

- виробниче середовище;
- побутове середовище.

Виробничим називають середовище, в якому людина реалізує свою трудову діяльність (підприємства, установи, навчальні заклади тощо).

Побутовим є середовище, де люди мешкають або проводять вільний час. Воно охоплює сукупність житлових будинків, комунально-побутових об’єктів, місця відпочинку та ін.

Організм людини може нормально функціонувати тільки тоді, коли умови (параметри) зовнішнього середовища відповідають оптимальним. Якщо умови середовища змінюються, стають несприятливими, то на протидію їм організм людини включає спеціальні механізми, які зберігають постійність параметрів внутрішнього середовища (всередині організму) чи змінюють їх у межах допустимого.

Можливість функціонування організму в середовищі, параметри якого постійно змінюються, забезпечується завдяки механізму, який називають адаптацією.

Адаптація (лат. adapto – пристосування) – динамічний процес пристосування організму до мінливих умов зовнішнього середовища, який спостерігається в будь-якому виді діяльності щоразу, коли виникають значні зміни в системі «людина – середовище». Адаптація може бути фізіологічною, психологічною, соціальною.

Отже, для функціонування системи ЖД середовище має обов’язково відповідати природним параметрам. Відхилення можливі в межах допустимого, коли організм людини здатний адаптуватися, захистити себе, підтримувати існування. Усе, що існує за цими межами, становить загрозу життю, тому виникає потреба захисту ЖД людей. Отже, безпека – важлива складова системи ЖД.

Розглядаючи систему ЖД як взаємодію людей з навколишнім середовищем, слід зауважити, що вона завжди підпорядкована певним принципам, правилам, умовам життя, природним умовам, традиціям тощо.

Система ЖД має такі характерні ознаки:

- її функціонування підпорядковане об’єктивним законам природи;

- це динамічна система, яка розвивається, удосконалюється, пристосовується до змін умов існування;
- тяжіє до сталого розвитку, вживаючи заходів захисту від упливу негативних факторів.

Основні принципи забезпечення ЖД такі:

- своєчасність, достатність, якість забезпечення людей необхідними для життя засобами високої якості і заходами в потрібний час у належній кількості;
- безпека ЖД (захист ЖД від упливу негативних факторів, що виникають унаслідок як природних явищ, так і діяльності людей).

Рівень реалізації цих принципів значною мірою залежить від способів забезпечення ЖД. Виходячи із сказаного, можна визначити такі головні способи забезпечення ЖД:

1. Організація ефективної трудової діяльності людей в суспільстві з максимальним залученням усіх ресурсів (створення робочих місць, упровадження високопродуктивного виробництва і технологій, нормування праці тощо).

2. Організація та удосконалення освіти і підготовка кадрів, розвиток науки відповідно до вимог часу.

3. Розвиток сфери послуг (комунальних, транспортних, торговельних, побутових і т. ін.).

4. Розширення мережі культурних, спортивних, розважальних установ.

5. Проведення заходів щодо збереження здоров’я людей (диспансеризація, оздоровлення, кваліфіковане медичне обслуговування і лікування, санітарно-епідеміологічний стан).

6. Розроблення законодавчих і нормативно-правових актів із забезпечення прав, свобод і захисту людей і суспільства в цілому.

Залежно від того, якою мірою реалізуються принципи та способи забезпечення ЖД, визначається рівень життя людей окремих країн і загальний розвиток людства.

4.2.3 Елементи теорії, що відповідають моделі безпеки життєдіяльності

Модель у широкому розумінні – це предмет, явище, система (опис, схема, знак, графік, план, макет та ін.), які за певних умов відіграють роль замінника або представника будь-якого іншого предмета, явища чи системи.

З точки зору науки модель – це матеріальна чи уявна система, що відображає чи імітує принципи внутрішньої організації, функціонування, певні властивості чи характеристики об'єкта дослідження, безпосереднє вивчення якого неможливе. Модель може замінити цей об'єкт у пізнавальному процесі з метою отримання нових знань про нього. Таким чином, відношення «модель—оригінал» не природне, а зумовлене процесом пізнання, і питання про їх співвідношення, ступінь їх подібності, адекватності – одне з найважливіших і найскладніших у процесі використання моделей у науковому пізнанні.

Сам процес моделювання – це непрямий, опосередкований метод наукового дослідження об'єктів пізнання на їх моделях, коли з певних причин безпосереднє їх вивчення неможливе.

Моделі в дисципліні «Безпека життєдіяльності» можна систематизувати за об'єктом зв'язків. Усі моделі можна умовно поділити на дві множини залежно від обсягу зв'язків, які вони демонструють.

Перша множина об'єднує моделі, що характеризуються структурою зв'язків.

Друга множина об'єднує моделі парних зв'язків. Певна умовність щодо цієї множини пов'язана з тим, що запровадження глибокого аналізу дозволяє уявити механізми реалізації цих зв'язків діючих великих систем.

Для характеристики довкілля на глобальному, державному і регіональному рівні використовують поняття структури зв'язків (на світовому рівні – навіть загальної). Відповідно до визначеної послідовності рівнів (за територією, від світового до регіонального) зменшується кількість таких зв'язків – з одного боку, а з іншого – збільшується рівень їх деталізації.

Під державним рівнем у цьому випадку розуміють сукупність діючих галузей виробництва як джерел забруднення і географічні чинники території, що одержує це забруднення. Відповідно до двох визначених рівнів подано моделі, що формують уявлення про стан світового довкілля і держави (на прикладі сільськогосподарської галузі). На регіональному рівні модель, що формує стан довкілля, може бути представлена у вигляді взаємодій комплексу діючих (діючого) підприємств із середовищем виробництва.

Для визначення умов роботи підприємства найбільшу увагу для застосування привертають моделі, що відображають зв'язки:

- «регіональний природно-виробничий комплекс – середовище виробництва»;
- «виробниче підприємство – довкілля»;
- «виробниче середовище виробничого підприємства (середовище робочого місця) – людина».

Здобуття найбільш деталізованої інформації за взаємодії можливе на рівні парних (взаємодій) у вигляді: забруднювач середовища (джерелом є підприємство) – елемент довкілля. Таким чином, необхідно розробити відповідні моделі парної взаємодії.

До таких моделей (як зразок) належать:

- модель розповсюдження елемента забруднення в середовищі (елементи довкілля – атмосфера, гідросфера, літосфера);

- моделі обігу елемента забруднення в елементах довкілля;
- моделі обігу елементів середовища;
- моделі взаємних впливів на елементи довкілля;
- моделі взаємодій екологічних компонентів і організації екосистем;
- моделі впливів небезпечних і шкідливих чинників;
- моделі ієрархії екосистем та ін.

У рамках пари «виробниче середовище – людина» певний зміст взаємодій реалізується на базі спрощення уявлення «виробниче середовище» і представлення його як «технологічний процес, обладнання, види господарських робіт тощо».

В період виконання «технологічного процесу...» виникають небезпеки. Це може бути ініційовано як з боку «технологічного процесу, обладнання, видів господарських робіт», так і з боку – «людини». Виходячи з цього, у схемі розгляду нещасного випадку необхідно йти двома шляхами відносно:

- технологічного процесу, обладнання, видів господарських робіт та ін.;
- «людини» як джерела небезпек.

Розвиток подій вивчають за допомогою ступеневих логіко-імітаційних моделей. Характер ступеневої суті моделі визначає перехід від події до події. Події і переходи за змістом формуються трьома складовими: 1) технологічний процес, його операції й елементи; 2) конструкція обладнання; 3) стан охорони праці при їх взаємодії.

За наявності небезпечних обставин під час виконання будь-яких робіт людина сприяє, усвідомлює, приймає і реалізує відповідні рішення в послідовності.

Обидві моделі в межах поєднання свого змісту дають змогу усвідомити комплексний розвиток подій, причини аварій та ін., сприяють створенню безпечних умов праці і запобіганню травматизму.

4.3 Висновки до 4 розділу

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп’ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи.

Також розглянуто питання міжнародного тероризму, структури системи БЖД, елементів теорії, що відповідають моделі безпеки життєдіяльності.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розглянуто проблему захисту даних і контролю активності вузлів у хмарній інфраструктурі в умовах високої динамічності та багатошаровості середовища.

Актуальність теми зумовлена зростанням частки критичних сервісів у хмарі та підвищенням ризиків, пов’язаних із помилками конфігурації, компрометацією IAM та атаками на контрольну площину. Метою роботи було розробити і обґрунтувати комплекс методів, які одночасно підвищують конфіденційність, цілісність і доступність даних та забезпечують керований моніторинг активності вузлів. Для досягнення мети проаналізовано базові моделі надання хмарних сервісів IaaS, PaaS і SaaS та їх вплив на розподіл відповідальності і рівень контролю. Показано, що зі зміщенням від IaaS до SaaS зменшується контроль над нижніми шарами інфраструктури, а зростає роль IAM, журналювання та політик доступу на рівні сервісів. Узагальнено шарову архітектуру хмарних обчислень і визначено ключові поверхні атак у площинах data plane та control plane.

Виконано систематизацію загроз відповідно до тріади CIA та описано типові сценарії порушення конфіденційності внаслідок відкритих сховищ і витоків облікових даних. Для складової цілісності охарактеризовано ризики несанкціонованих змін даних, підміни артефактів постачання та маніпуляцій політиками й конфігураціями. Для складової доступності розглянуто DDoS, виснаження квот, ransomware та регіональні інциденти провайдерів як джерела простою. У роботі підтверджено, що ефективний захист у хмарі неможливий без врахування моделі спільної відповідальності, договірних SLA та комплаєнс-вимог. На основі цієї тріади сформовано підхід до побудови матриці контролів і доказовості виконання вимог.

Формалізовано межі системи як взаємодію середовища виконання, контрольної площини та підсистеми спостережуваності й реагування.

Визначено об’єкти захисту, до яких віднесено дані, ідентичності та секрети, робочі навантаження, мережеві компоненти і політики керування. Сформовано модель загроз і порушника та запропоновано оцінювання ризику як добуток імовірності реалізації загрози на очікуваний збиток.

Сформульовано набір вимог, що охоплює CIA та додаткову властивість аудитованості, необхідну для розслідувань та відповідності стандартам. Запропоновано критерії ефективності, які переводять безпеку у вимірюваний формат через покриття шифрування, повноту audit-логів, MTTD, MTTR та виконання SLO. Обґрунтовано застосування шифрування даних у русі через TLS і mTLS як базового транспортного контролю для публічних і внутрішніх взаємодій.

Визначено, що критичною умовою якості TLS є централізоване керування сертифікатами, правильний вибір версій протоколу та журналювання помилок рукопотискань. Для даних у спокої обґрунтовано використання шифрування сховищ і баз даних із керуванням ключами через KMS та HSM. Показано, що envelope encryption у поєднанні з розмежуванням ролей, ротацією і аудитом операцій KMS зменшує ризик довготривалої компрометації ключів. Для секретів визначено практики зберігання в керованих сховищах, мінімізації доступів та регулярної або подієвої ротації. Розглянуто конфіденційні обчислення як засіб захисту даних у використанні та як основу політик видачі ключів після атестації довіреного середовища. Центральним елементом запропонованої системи визначено контроль доступу, який поєднує RBAC як структурний каркас і ABAC як контекстний фільтр доступу. Підтверджено важливість MFA та підходів step-up authentication для привілейованих операцій і зниження ризиків фішингу та викрадення сесій.

Показано, що модель Zero Trust забезпечує безперервну перевірку доступу, мікросегментацію та мінімізацію довіри до мережевих меж. Для контролю активності вузлів обґрунтовано використання телеметрії у вигляді

логів, метрик і трасування з кореляцією через спільний контекст. Визначено мінімальний набір подій для журналювання, включно з аутентифікацією, змінами ролей, конфігураційними діями та операціями доступу до даних. Доведено, що захист самих логів, їх ретенція та незмінюваність є критичними для форензіки і протидії спробам «осліплення» моніторингу. Обґрунтовано комбіновану стратегію агентного та безагентного моніторингу, де платформа забезпечує аудит control plane, а агенти дають глибину на рівні процесів і хоста. Розглянуто роль EDR і XDR для детекції поведінкових ознак компрометації та кореляції інцидентів між доменами.

Показано, що SIEM забезпечує централізоване збирання, нормалізацію та кореляцію подій, а SOAR підвищує швидкість реагування через автоматизовані плейбуки. Запропоновано багатошаровий підхід до виявлення аномалій, який поєднує кореляційні правила, UEBA та базові ML-методи як підсилювачі сигналів ризику. Розроблено референсну архітектуру комплексної системи, яка охоплює IAM, захист даних, захист навантажень, мережевий контур, телеметрію та оркестрацію реагування.

Для практичної реалізації обрано платформу Microsoft Azure, що обґрунтовано наявністю зрілих сервісів ідентичності, керування ключами та нативного стека моніторингу. У межах практичної частини описано використання Azure Monitor, Activity Log, Diagnostic settings і Log Analytics як основи централізованого збору журналів і метрик. Для захисту ключів і секретів застосовано підходи на базі Azure Key Vault та HSM-орієнтованих механізмів, що підсилюють контроль доступів і аудит криптоопераций. Визначено сценарії реагування на компрометацію ідентичностей, ексфільтрацію даних, підозрілу активність вузлів та спроби вимкнення логування, з орієнтацією на автоматизацію стримування.

Оцінювання ефективності запропонованих рішень виконано через метрики покриття контролів, показники MTTD і MTTR, а також перевірку готовності до відновлення за RPO та RTO. Практичні результати

підтверджують, що поєднання шифрування, керованого IAM, централізованої телеметрії та стандартизованого інцидент-менеджменту суттєво зменшує ризик і масштаб інцидентів у хмарі. Подальший розвиток роботи доцільно спрямувати на поглиблення автоматизованої ремедіації, розширення поведінкової аналітики та уніфікацію політик у multi-cloud або гібридних середовищах.Охарактеризовано та описано схеми організації перевірки безпеки, запобіганню вірусним загрозам та методи запобіжних заходів при проникненні шкідливого ПЗ в систему.

Також були розглянуті питання з охорони праці і безпеки в надзвичайних ситуаціях.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Amazon Web Services. (n.d.). Shared responsibility model. Retrieved November 17, 2025, from https://aws.amazon.com/compliance/shared-responsibility-model/

2. Microsoft. (n.d.). Shared responsibility in the cloud (Azure). Retrieved November 17, 2025, from https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility

3. Amazon Web Services. (n.d.). Shared responsibility model. In AWS risk and compliance whitepaper. Retrieved November 17, 2025, from https://docs.aws.amazon.com/whitepapers/latest/aws-risk-and-compliance/shared-responsibility-model.html

4. International Organization for Standardization. (2016). ISO/IEC 19086-1:2016 Cloud computing—Service level agreement (SLA) framework—Part 1: Overview and concepts. Retrieved November 17, 2025, from https://www.iso.org/standard/67545.html

5. Amazon Web Services. (n.d.). Amazon compute service level agreement (SLA) (Amazon EC2 SLA). Retrieved November 17, 2025, from https://aws.amazon.com/compute/sla/

6. Microsoft. (n.d.). Service level agreements (SLA) for online services. Retrieved November 17, 2025, from https://www.microsoft.com/licensing/docs/view/Service-Level-Agreements-SLA-for-Online-Services?lang=1

7. International Organization for Standardization. (2015). ISO/IEC 27017:2015 Security techniques—Code of practice for information security controls based on ISO/IEC 27002 for cloud services. Retrieved November 17, 2025, from https://www.iso.org/standard/43757.html

8. International Organization for Standardization. (n.d.). ISO/IEC 27018 (PII in public clouds). Retrieved November 17, 2025, from https://www.iso.org/standard/27018

9. European Union. (2016). Regulation (EU) 2016/679 (General Data Protection Regulation). Retrieved November 17, 2025, from https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

10. AICPA. (n.d.). SOC 2®—SOC for service organizations. Retrieved November 17, 2025, from https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-2

11. PCI Security Standards Council. (n.d.). PCI data security standard (PCI DSS). Retrieved November 17, 2025, from https://www.pcisecuritystandards.org/standards/pci-dss/

12. National Institute of Standards and Technology. (n.d.). SP 800-52 Rev. 2: Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/52/r2/final

13. National Institute of Standards and Technology. (n.d.). SP 800-111: Guide to storage encryption technologies for end user devices. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/111/final

14. National Institute of Standards and Technology. (n.d.). SP 800-57 Part 1 Rev. 5: Recommendation for key management—Part 1: General. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final

15. National Institute of Standards and Technology. (n.d.). SP 800-53 Rev. 5: Security and privacy controls for information systems and organizations. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

16. National Institute of Standards and Technology. (n.d.). SP 800-34 Rev. 1: Contingency planning guide for federal information systems. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/34/r1/upd1/final

17. Center for Internet Security. (n.d.). CIS critical security control 3: Data protection. Retrieved November 17, 2025, from https://www.cisecurity.org/controls/data-protection

18. Center for Internet Security. (n.d.). CIS Controls v8 (guide). Retrieved November 17, 2025, from https://www.cisecurity.org/controls/v8

19. International Organization for Standardization. (2025). ISO/IEC 27001:2025 Information security management systems—Requirements. Retrieved November 17, 2025, from https://www.iso.org/standard/27001

20. International Organization for Standardization. (2025). ISO/IEC 27002:2025 Information security controls. Retrieved November 17, 2025, from https://www.iso.org/standard/75652.html

21. 2Amazon Web Services. (n.d.). AWS Key Management Service (AWS KMS): Developer guide—Concepts. Retrieved November 17, 2025, from https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html

22. Microsoft. (n.d.). Overview of key management in Azure. Retrieved November 17, 2025, from https://learn.microsoft.com/en-us/azure/security/fundamentals/key-management

23. Google Cloud. (n.d.). Customer-managed encryption keys (CMEK)—Cloud KMS. Retrieved November 17, 2025, from https://docs.cloud.google.com/kms/docs/cmek

24. National Institute of Standards and Technology. (n.d.). SP 800-92: Guide to computer security log management. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/92/final

25. National Institute of Standards and Technology. (n.d.). SP 800-137: Information Security Continuous Monitoring (ISCM) for federal information systems and organizations. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/137/final

26. National Institute of Standards and Technology. (n.d.). SP 800-61 Rev. 3: Incident response recommendations and considerations for cybersecurity risk management (A CSF 2.0 community profile). Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/61/r3/final

27. National Institute of Standards and Technology. (2025). NIST revises SP 800-61 (news). Retrieved November 17, 2025, from https://csrc.nist.gov/news/2025/nist-revises-sp-800-61

28. National Institute of Standards and Technology. (n.d.). SP 800-53 Rev. 5: Security and privacy controls for information systems and organizations. Retrieved November 17, 2025, from https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

29. OpenTelemetry. (n.d.). Signals (traces, metrics, logs). Retrieved November 17, 2025, from https://opentelemetry.io/docs/concepts/signals/

30. OpenTelemetry. (n.d.). Logs specification (log correlation). Retrieved November 17, 2025, from https://opentelemetry.io/docs/specs/otel/logs/

31. OpenTelemetry. (n.d.). Metrics specification (correlation goals, exemplars). Retrieved November 17, 2025, from https://opentelemetry.io/docs/specs/otel/metrics/

32. Google. (n.d.). Monitoring distributed systems. In Site reliability engineering. Retrieved November 17, 2025, from https://sre.google/sre-book/monitoring-distributed-systems/

33. Google. (n.d.). Service level objectives. In Site reliability engineering. Retrieved November 17, 2025, from https://sre.google/sre-book/service-level-objectives/

34. MITRE. (n.d.). ATT&CK® data sources. Retrieved November 17, 2025, from https://attack.mitre.org/datasources/

35. MITRE. (n.d.). ATT&CK® analytics. Retrieved November 17, 2025, from https://attack.mitre.org/analytics/

36. National Institute of Standards and Technology. (n.d.). SP 800-30 Rev. 1: Guide for conducting risk assessments. Retrieved November 18, 2025, from https://csrc.nist.gov/pubs/sp/800/30/r1/final

37. International Organization for Standardization. (n.d.). ISO/IEC 27005: Guidance on managing information security risks. Retrieved November 18, 2025, from https://www.iso.org/standard/80585.html

38. National Institute of Standards and Technology. (n.d.). SP 800-59 Rev. 5: Security and privacy controls for information systems and organizations. Retrieved November 18, 2025, from https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

39. International Organization for Standardization. (2015). ISO/IEC 27017:2015 Security techniques—Code of practice for information security controls for cloud services. Retrieved November 18, 2025, from https://www.iso.org/standard/43757.html

40. International Organization for Standardization. (2025). ISO/IEC 27001:2025 Information security management systems—Requirements. Retrieved November 18, 2025, from https://www.iso.org/standard/27001

41. MITRE. (n.d.). ATT&CK® data sources. Retrieved November 18, 2025, from https://attack.mitre.org/datasources/

42. National Institute of Standards and Technology. (n.d.). SP 800-137: Information Security Continuous Monitoring (ISCM) for federal information systems and organizations. Retrieved November 18, 2025, from https://csrc.nist.gov/pubs/sp/800/137/final

43. National Institute of Standards and Technology. (n.d.). SP 800-61 Rev. 3: Incident response recommendations and considerations for cybersecurity risk management. Retrieved November 18, 2025, from https://csrc.nist.gov/pubs/sp/800/61/r3/final

44. International Organization for Standardization. (2016). ISO/IEC 19086-1:2016 Cloud computing—Service level agreement (SLA) framework—Part 1: Overview and concepts. Retrieved November 18, 2025, from https://www.iso.org/standard/67545.html

45. International Organization for Standardization. (2019). ISO/IEC 27018:2019 Protection of personally identifiable information (PII) in public clouds acting as PII processors. Retrieved November 18, 2025, from https://www.iso.org/standard/76559.html

46. OpenTelemetry. (n.d.). Signals (traces, metrics, logs). Retrieved November 18, 2025, from https://opentelemetry.io/docs/concepts/signals/

47. CSF Tools. (n.d.). NIST SP 800-53 AU-2: Event logging. Retrieved November 18, 2025, from https://csf.tools/reference/nist-sp-800-53/r5/au/au-2/

48. National Institute of Standards and Technology. (n.d.). SP 800-52 Rev. 2: Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations. Retrieved November 18, 2025, from https://csrc.nist.gov/pubs/sp/800/52/r2/final

49. Internet Engineering Task Force. (n.d.). RFC 8446: The Transport Layer Security (TLS) protocol version 1.3. Retrieved November 18, 2025, from https://datatracker.ietf.org/doc/html/rfc8446

50. National Institute of Standards and Technology. (n.d.). SP 800-57 Part 1 Rev. 5: Recommendation for key management—Part 1: General. Retrieved November 18, 2025, from https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final

51. Amazon Web Services. (n.d.). AWS Key Management Service (AWS KMS)—Concepts. Retrieved November 18, 2025, from https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html

52. Amazon Web Services. (n.d.). AWS KMS cryptography essentials (envelope encryption). Retrieved November 18, 2025, from https://docs.aws.amazon.com/kms/latest/developerguide/kms-cryptography.html

53. Microsoft Learn. (n.d.). Azure Key Vault overview. Retrieved November 18, 2025, from https://learn.microsoft.com/en-us/azure/key-vault/general/overview

54. Microsoft Learn. (n.d.). Azure Key Vault: About keys, secrets, and certificates. Retrieved November 18, 2025, from https://learn.microsoft.com/en-us/azure/key-vault/general/about-keys-secrets-certificates

55. Amazon Web Services. (n.d.). Rotate AWS Secrets Manager secrets. Retrieved November 18, 2025, from https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html

56. Google Cloud. (n.d.). Secret Manager overview. Retrieved November 18, 2025, from https://docs.cloud.google.com/secret-manager/docs/overview

57. Google Cloud. (n.d.). About secret rotation. Retrieved November 18, 2025, from https://docs.cloud.google.com/secret-manager/regional-secrets/about-rotation-schedules-rs

58. Leshchyshyn, Y., Scherbak, L., Nazarevych, O., Gotovych, V., Tymkiv, P., & Shymchuk, G. (2019, May). Multicomponent Model of the Heart Rate Variability Change-point. In 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH) (pp. 110-113). IEEE.

59. Lytvynenko, I., Lupenko, S., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021, September). Mathematical model of gas consumption process in the form of cyclic random process. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 1, pp. 232-235). IEEE.

60. Kozlovskyi, V., Balanyuk, Y., Martyniuk, H., Nazarevych, O., Scherbak, L., & Shymchuk, G. (2025, April). Information Technology for Estimating City Gas Consumption During the Year. In 2025 International Conference on Smart Information Systems and Technologies (SIST) (pp. 1-4). IEEE.

61. Lytvynenko, I., Lupenko, S., Kunanets, N., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021). Simulation of gas consumption process based on the mathematical model in the form of cyclic random process considering the scale factors. In 1st International Workshop on Information Technologies: Theoretical and Applied Problems, ITTAP (Vol. 2021).

62. Lupenko, S., Lytvynenko, I., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021, December). Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process. In Proceedings of the International Conference „Advanced applied energy and

information technologies 2021", 2021 (pp. 213-219). TNTU, Zhytomyr «Publishing house „Book-Druk "» LLC.

63. Kunanets, N., Pasichnyk, V., Bodnarchuk, I., Martsenko, S., Matsiuk, O., Matsiuk, A., ... & Shymchuk, H. (2019). Information system for visual analyzer disease diagnostics. In CEUR Workshop Proceedings (pp. 43-56).

64. Lytvynenko, I., Lupenko, S., Nazarevych, O., Shymchuk, H., & Hotovych, V. (2025). Additive mathematical model of gas consumption process. Вісник Тернопільського національного технічного університету, 104(4), 87-97.

65. Leschyshyn, Y. Z., Nazarevych, O. B., Shymchuk, G. V., Revutskyi, E. A., & Shcherbak, L. M. (2016, September). The Methods of Change Point Detection and Statistical Estimating of Dynamic of the Noise Stochastic Signals Characteristics. In THE SEVENTH WORLD CONGRESS "AVIATION IN THE XXI-st CENTURY" Safety in Aviation and Space Technologies September 19-21, NATIONAL AVIATION UNIVERSITY. Kyiv: NAU.

66. Nazarevych, O., Leshchyshyn, Y., Lupenko, S., Hotovych, V., Shymchuk, G., & Shabliy, N. (2020, September). Method of Gas Consumption Change-point Detection Based on Seasonally Multicomponent Model. In 2020 10th International Conference on Advanced Computer Information Technologies (ACIT) (pp. 152-155). IEEE.

67. ШИМЧУК, Г., ШЕВЧЕНКО, Н., ШВИРЛО, К., & ГАРМАТЮК, Н. (2025). СИСТЕМА ВІДНОВЛЕННЯ ДАНИХ У БЕЗДРОТОВИХ СЕНСОРНИХ МЕРЕЖАХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical sciences, 353(3.2), 246-250.

68. Shymchuk, G., Lytvynenko, I., Hromyak, R., Lytvynenko, S., & Hotovych, V. (2023). Gas Consumption Forecasting Using Machine Learning Methods and Taking Into Account Climatic Indicators. In CITI (pp. 156-163).

69. Шимчук, Г. В., Маєвський, О. В., & Назаревич, О. Б. (2016). Конспект лекцій з дисципліни «Розподілені системи моніторингу та керування».

70. ШИМЧУК, Г., ШЕВЧЕНКО, Н., ШВИРЛО, К., & ГАРМАТЮК, Н. (2025). СИСТЕМА ВІДНОВЛЕННЯ ДАНИХ У БЕЗДРОТОВИХ СЕНСОРНИХ МЕРЕЖАХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical sciences, 353(3.2), 246-250.

71. Шимчук, Г. В., Назаревич, О. Б., Литвиненко, Я. В., Готович, В. А., Никитюк, В. В., & Боднарчук, І. О. (2025). Грід-системи та технології хмарних обчислень. Навчальний посібник для здобувачів освітнього рівня «магістр» спеціальностей: F3 «Комп'ютерні науки», F6 «Інформаційні системи та технології».

72. Malyuta, Y., Derkach, M., & Lobur, T. (2025). Modeling a Fog Computing Network Architecture for Secure IoT Data Processing. *Security of Infocommunication Systems and Internet of Things*, *3*(2), 02017-02017.

73. Stanko, A., Wieczorek, W., Mykytyshyn, A., Holotenko, O., & Lechachenko, T. (2024). Realtime air quality management: Integrating IoT and Fog computing for effective urban monitoring. CITI, 2024, 2nd.

74. Babakov, R. M., et al. "Internet of Things for Industry and Human Application. Vol. 3." (2019): 1-917.

75. Sachenko A.O., Kochan V.V., Bykovyy P.Ye., Zahorodnia D.I., Osolinskyy O.R., Skarga-Bandurova I.S., Derkach M.V., Orekhov O.O., Stadnik A.O., Kharchenko V.S., Fesenko H.V. Internet of Things for intelligent transport systems: Practicum / A.O. Sachenko (Eds.) – Ministry of Education and Science of Ukraine, Ternopil National Economic University, Volodymyr Dahl East Ukrainian National University, National Aerospace University "Kharkiv Aviation Institute", 2019. – 135 p. ISBN 978-617-7361-92-2. https://aliot.eu.org/wp-content/uploads/2019/10/ALIOT_ITM3_IoT-for-Int-TransSys_web.pdf

76. Skarga-Bandurova, I., Biloborodova, T., Kjelstrup-Johnson, K. R., Scheper, T. V. O., & Derkach, M. (2026). Securing Tomorrow's Cities: Smart Infrastructure for Emergency Response, Crisis Management, and Defence. In *Sustainable, Innovative, and Intelligent Industries and Societies* (pp. 69-111). Cham: Springer Nature Switzerland.

77. Міністерство соціальної політики України. (2018, 14 лютого). Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями (Наказ № 207; НПАОП 0.00-7.15-18). База даних «Законодавство України». https://zakon.rada.gov.ua/go/z0508-18

78. Кабінет Міністрів України. (2025, 8 квітня). Про скасування деяких наказів міністерств та інших центральних органів виконавчої влади (Розпорядження № 317-р). База даних «Законодавство України». https://zakon.rada.gov.ua/go/317-2025-%D1%80

79. Міністерство охорони здоров’я України. (1998, 10 грудня). Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин (ДСанПіН 3.3.2.007-98) (Наказ № 7; документ втратив чинність). База даних «Законодавство України». https://zakon.rada.gov.ua/go/v0007282-98

80. Міністерство охорони здоров’я України. (1999, 1 грудня). ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. База даних «Законодавство України». https://zakon.rada.gov.ua/go/va037282-99

81. Міністерство охорони здоров’я України. (1999, 1 грудня). ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. База даних «Законодавство України». https://zakon.rada.gov.ua/go/va042282-99

82. Єдина державна електронна система у сфері будівництва (e-Construction). (n.d.). ДБН В.2.5-28:2018. Природне і штучне освітлення. Retrieved December 21, 2025, from https://e-construction.gov.ua/laws_detail/3074958732556240833?doc_type=2

83. Єдина державна електронна система у сфері будівництва (e-Construction). (n.d.). ДБН В.2.5-56:2014. Системи протипожежного захисту. Retrieved December 21, 2025, from https://e-construction.gov.ua/laws_detail/3200383488549193714?doc_type=2

84. Єдина державна електронна система у сфері будівництва (e-Construction). (n.d.). ДБН В.1.1-7:2016. Пожежна безпека об’єктів будівництва.

Загальні вимоги. Retrieved December 21, 2025, from https://e-construction.gov.ua/laws_detail/3080743763845318619

85. Міністерство внутрішніх справ України. (2014, 30 грудня). Про затвердження Правил пожежної безпеки в Україні (Наказ № 1417). База даних «Законодавство України». https://zakon.rada.gov.ua/go/z0252-15

86. Міністерство внутрішніх справ України. (2018, 15 січня). Про затвердження Правил експлуатації та типових норм належності вогнегасників (Наказ № 25). База даних «Законодавство України». https://zakon.rada.gov.ua/go/z0225-18

87. Український національний стандарт. (2004). ДСТУ 4297:2004. Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги. Retrieved December 21, 2025, from https://zakon.isu.net.ua/node/51525/printable/pdf

88. Міністерство енергетики та вугільної промисловості України. (2017, 21 липня). Про затвердження Правил улаштування електроустановок (Наказ № 476). База даних «Законодавство України». https://zakon.rada.gov.ua/go/v0476732-17

89. Міністерство праці та соціальної політики України. (1998, 9 січня). Про затвердження Правил безпечної експлуатації електроустановок споживачів (ДНАОП 0.00-1.21-98) (Наказ № 4). База даних «Законодавство України». https://zakon.rada.gov.ua/go/z0093-98

90. ДП «Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості» (ДП «УкрНДНЦ»). (2015, 21 грудня). ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги (Наказ № 204). База даних «Законодавство України». https://zakon.rada.gov.ua/go/v0204774-15

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя
Маріборский університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Краківський економічний університет (Польща)
Вроцлавський економічний університет (Польща)
Університет «Опольська Політехніка» (Польща)
Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Вінницький національний аграрний університет
Львівський національний університет ім. І. Франка
Головне управління Пенсійного фонду в Тернопільській області
Наукове товариство ім. Шевченка
Тернопільський обласний комунальний інститут післядипломної педагогічної освіти
Сумський державний педагогічний університет
Запорізький національний університет

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

XIV Міжнародної науково-технічної конференції молодих учених та студентів

11-12 грудня 2025 року

УКРАЇНА
ТЕРНОПІЛЬ – 2025

практичні дії (провітрювання, контроль вологості, оптимізація опалення). Подальший розвиток доцільно спрямувати на прогнозування параметрів (короткострокові прогнози CO_2/вологості), автоматичне керування виконавчими пристроями та розширення кіберзахисту (керування ключами, політики доступу, виявлення підозрілої активності).

Література

1. OASIS. MQTT Version 5.0 : OASIS Standard. – 2019. – URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf (дата звернення: 02.12.2025).

2. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3 : RFC 8446. – IETF, 2018. – URL: https://www.rfc-editor.org/rfc/rfc8446.html (дата звернення: 02.12.2025).

3. Fielding R., Nottingham M., Reschke J. HTTP Semantics : RFC 9110. – IETF, 2022. – URL: https://www.rfc-editor.org/rfc/rfc9110.html (дата звернення: 02.12.2025).

4. Lodderstedt T., Bradley J., Labunets A., Fett D. Best Current Practice for OAuth 2.0 Security : RFC 9700 (BCP 240). – IETF, 2025. – URL: https://datatracker.ietf.org/doc/rfc9700/ (дата звернення: 02.12.2025).

5. Voas J. Networks of "Things" : NIST Special Publication 800-183. – Gaithersburg, MD : NIST, 2016. – URL: https://csrc.nist.gov/pubs/sp/800/183/final (дата звернення: 02.12.2025).

6. Fagan M., Megas K., Scarfone K., Smith M. IoT Device Cybersecurity Capability Core Baseline : NISTIR 8259A. – Gaithersburg, MD : NIST, 2020. – URL: https://doi.org/10.6028/NIST.IR.8259A (дата звернення: 02.12.2025).

7. Fagan M., Marron J., Brady K., Cuthill B. та ін. IoT Device Cybersecurity Guidance for the Federal Government : NIST SP 800-213. – Gaithersburg, MD : NIST, 2021. – URL: https://doi.org/10.6028/NIST.SP.800-213 (дата звернення: 02.12.2025).

8. ETSI EN 303 645 V2.1.1. Cyber Security for Consumer Internet of Things: Baseline Requirements. – ETSI, 2020. – URL: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf (дата звернення: 02.12.2025).

9. Sensirion. SGP40. VOC Sensor : Datasheet. – 2021. – URL: https://sensirion.com/media/documents/296373BB/618ED821/Sensirion_Gas_Sensors_SGP40_Datasheet.pdf (дата звернення: 02.12.2025).

10. Microsoft. Azure IoT Hub documentation (device connectivity / telemetry ingestion). –URL: https://learn.microsoft.com/azure/iot-hub/ (дата звернення: 02.12.2025).

користувачу практичної користі, тоді як хмарний підхід дозволяє забезпечити централізоване зберігання, доступ із будь-якого пристрою, масштабування та розширення функцій (алерти, аналітика, інтеграції з іншими сервісами).

Мета роботи – розробити концепцію та прототип хмарної платформи для збору, зберігання та візуалізації телеметрії мікроклімату smart-квартири з підтримкою подієвих сповіщень і базових механізмів безпеки.

Завдання:

– визначити вимоги до телеметрії (частота вимірювань, затримка, надійність доставки, збереження історії);
– спроєктувати хмарну архітектуру приймання даних і API-доступу;
– обґрунтувати вибір сховища часових рядів та підходу до агрегації;
– реалізувати веб-дашборди для багатозонної візуалізації;
– додати правила алертингу та аудит стану сенсорних вузлів;
– передбачити механізми автентифікації, шифрування й розмежування доступу.

Запропоновано багаторівневу модель "сенсори → шлюз → хмара → веб-інтерфейс". На рівні квартири сенсорні вузли (кімната/зона) вимірюють параметри мікроклімату та передають дані на домашній шлюз (наприклад, ESP32/Raspberry Pi/роутер). Шлюз виконує буферизацію при нестабільному інтернет-з'єднанні, нормалізацію форматів і публікацію телеметрії у хмару через MQTT/HTTPS.

У хмарі платформа містить такі компоненти:

– Ingress-рівень: брокер повідомлень або endpoint приймання телеметрії (MQTT/Webhook), контроль авторизації пристроїв.
– Stream Processing: сервіс обробки подій (валідація, дедуплікація, перерахунок одиниць, формування агрегатів 1/5/15 хв).
– Time-series Storage: база часових рядів (для швидких графіків і довгої історії) з політиками зберігання (retention) та downsampling.
– Alerting Service: правила з гістерезисом і часовими вікнами підтвердження (наприклад, CO_2 > порога 10 хв; ризик конденсації за комбінацією температури й вологості).
– Web/API Layer: REST/GraphQL API, веб-кабінет користувача, управління зонами/пристроями та ролями доступу.

Веб-інтерфейс орієнтований на швидке розуміння стану квартири:

– "карта кімнат" з кольоровою індикацією (норма/попередження/критично);
– графіки трендів по зонах, порівняння кімнат, добові профілі;
– обчислювані індикатори (наприклад, комфортність, індекс вентиляції за CO_2, ризик плісняви);
– журнали подій: перевищення порогів, відпадання вузла ("last seen"), зміна налаштувань;
– експорт даних (CSV/JSON) для подальшого аналізу.

Для забезпечення стійкості платформи передбачено асинхронний прийом даних через брокер, повторні відправки з боку шлюзу та ідемпотентність обробки. Захист телеметрії базується на TLS, токенах/сертифікатах для пристроїв, розмежуванні доступу (ролі: користувач/адміністратор), а також аудиті операцій у веб-кабінеті. Для зменшення вартості підтримується гнучка частота вимірювань і багаторівневі retention-політики (детальні дані – коротше, агрегати – довше).

Хмарна платформа візуалізації телеметрії мікроклімату для smart-квартири забезпечує централізований збір та зберігання часових рядів, зручну багатозонну візуалізацію і покрокове сповіщення, що перетворює "сирі" показники сенсорів на

– Організовано сховище часових рядів і дашборди для аналізу трендів мікроклімату в кожній кімнаті та порівняння зон квартири.

– Додано правила сповіщення: перевищення CO_2 у спальні, ризик конденсації при високій вологості та низькій температурі, різкі зміни VOC.

– Показано, що БСМ-підхід з енергоефективними протоколами дає змогу розміщувати вузли без прокладання кабелів та з мінімальним впливом на інтер'єр.

Новизна полягає в поєднанні БСМ з інформаційною підсистемою часових рядів та подієвого сповіщення, адаптованих до умов квартири (багатозонність, перешкоди, короткі відстані, потреба в простій експлуатації).

Практична цінність – можливість оперативного виявлення проблем (погана вентиляція, перезволоження, забруднення повітря) та формування рекомендацій мешканцям, а також основа для подальшої автоматизації (керування вентиляцією/зволожувачем/опаленням).

Розроблена концепція інформаційної системи моніторингу мікроклімату квартири на базі бездротової сенсорної мережі забезпечує багаточочковий збір даних, централізоване зберігання часових рядів, зручну візуалізацію та своєчасне сповіщення про критичні відхилення. Подальші напрями розвитку: прогнозування мікрокліматичних параметрів (короткострокові прогнози CO_2/вологості), інтеграція з системами "розумного дому" (Home Assistant), підвищення кібербезпеки (шифрування, керування ключами, контроль доступу).

Література

1. Sensor Systems for Greenhouse Microclimate Monitoring and Control: a Review / [J. K. Basak, H. T. Kim, A. Bhujel та ін.] // Sensors. – 2021. URL: https://link.springer.com/content/pdf/10.1007/s42853-020-00075-6.pdf.

2. Cureau R. J. New Wearable System for Sensing Outdoor Environmental Conditions for Monitoring Hyper-Microclimate / R. J. Cureau, I. Pigliautile, A. L. Pisello // Sensors. – 2022. URL: https://mdpires.com/d_attachment/sensors/sensors-22-00502/article_deploy/sensors-22- 00502.pdf?version=1641819298.

3. Modular and Cost-Effective Microclimate Monitoring System for Ecological Research // Sensors. – 2021. URL: https://mdpi-res.com/d_attachment/sensors/sensors-21-04615/article_deploy/sensors21-04615-v2.pdf?version=1625565877.

УДК 004.738.5:681.518

С.Б. Кіт, В.Р. Перун, С.І. Перун, Г.В. Шимчук
(Тернопільський національний технічний університет імені Івана Пулюя)

ХМАРНО-ОРІЄНТОВАНА ПЛАТФОРМА СПОСТЕРЕЖЕННЯ ЗА МІКРОКЛІМАТОМ КВАРТИРИ НА ОСНОВІ IOT

S.B. Kit, V.R. Perun, S.I. Perun, G.V. Shymchuk

CLOUD-BASED PLATFORM FOR MONITORING APARTMENT MICROCLIMATE BASED ON IOT

Зростання популярності smart-home рішень формує потребу у надійному моніторингу мікроклімату житлових приміщень у реальному часі. Температура, вологість, рівень CO_2, TVOC та інші показники змінюються нерівномірно в різних кімнатах і залежать від вентиляції, кількості людей, режиму опалення та побутових процесів. Локальний збір даних без аналітики та зручної візуалізації часто не дає

системи, здатної безперервно збирати показники з кількох кімнат, оперативно виявляти відхилення та формувати рекомендації/сповіщення для мешканців.

Мета роботи – розробити інформаційну систему моніторингу мікроклімату квартири (на прикладі 2–3-кімнатного планування) на базі бездротової сенсорної мережі (БСМ), що забезпечує масштабованість, енергоефективність та надійність збору даних.

Завдання:

– визначити набір параметрів мікроклімату та вимоги до періодичності вимірювання;

– спроєктувати архітектуру БСМ (сенсорні вузли – шлюз – сервер – клієнтські сервіси);

– реалізувати протокол обміну та сховище часових рядів;

– забезпечити візуалізацію, порогове сповіщення та базову аналітику;

– оцінити якість зв'язку, автономність вузлів і стабільність системи в умовах квартири.

Об'єкт дослідження – процес моніторингу параметрів мікроклімату в житловому приміщенні.

Предмет дослідження – методи та програмно-апаратні засоби побудови бездротової сенсорної мережі та інформаційної системи збору, зберігання й аналізу мікрокліматичних даних.

Система реалізується у вигляді багаторівневої архітектури:

– Сенсорні вузли (по одному на кімнату): мікроконтролер класу ESP32/NRF52; сенсори температури/вологості, атмосферного тиску, CO_2, TVOC/якості повітря (за потреби – освітленість та шум). Вузол виконує опитування сенсорів, первинну фільтрацію (усереднення/медіанний фільтр), часову синхронізацію та передачу пакетів.

– Бездротова мережа: для квартири доцільні Zigbee або BLE Mesh (низьке енергоспоживання, стійкість у приміщеннях), альтернативно Wi-Fi для спрощення інтеграції, якщо автономність не критична. Топологія – зірка (через шлюз) або mesh (для покриття складних зон).

– Шлюз (Raspberry Pi/домашній роутер/міні-ПК): прийом телеметрії та маршрутизація повідомлень до брокера MQTT; кешування при тимчасовій втраті інтернету.

– Серверний рівень: брокер MQTT, сервіс обробки (нормалізація, валідація, детекція аномалій), сховище часових рядів (InfluxDB/TimescaleDB), модуль сповіщень (Telegram/e-mail/push).

– Користувацький рівень: веб-панель і дашборди (Grafana/власний UI) з картами кімнат, трендами, статистикою та історією подій.

Для підвищення якості даних застосовуються:

– калібрування сенсорів за еталонними вимірюваннями (особливо CO_2/VOC);

– контроль достовірності (відсікання нереалістичних стрибків, перевірка діапазонів);

– адаптивні пороги сповіщень (наприклад, різні межі для дня/ночі у спальні).

– Основні результати:

– Запропонована архітектура системи, що дозволяє масштабувати кількість вузлів без суттєвого ускладнення серверної частини.

– Реалізована передача телеметрії через MQTT з буферизацією на шлюзі, що знижує втрати при нестабільному каналі.

дубльованих Content-Length виявлено ситуації, коли один компонент бере перше значення, інший - останнє, що призводить до розділення потоку на валідний та некоректний запити. Для змішаних CRLF/LF-варіантів зафіксовано випадки, коли Nginx приймає LF як допустимий роздільник, тоді як Apache повертає 400 Bad Request через некоректне завершення секції заголовків, що також відображає критичну асиметрію трактувань.

Диференціальний фазинг у поєднанні з HTTP Request Smuggler є ефективним підходом до систематичного виявлення HRS-вразливостей у багаторівневих вебінфраструктурах. Запропонована методика дозволяє не лише виявляти конкретні експлуатаційні сценарії CL.TE, TE.CL і CRLF-інжекцій у типових зв'язках Nginx–Apache, але й формалізувати критерії парсингових розбіжностей для подальшого впровадження превентивних заходів, зокрема жорсткішої валідації заголовків, уніфікації політики обробки Content-Length/Transfer-Encoding та суворого дотримання стандартів HTTP/1.1 у проміжних вузлах.

Література

1. Exploiting and Preventing HTTP Request Smuggling. VAADATA - Ethical Hacking Services. URL: https://www.vaadata.com/blog/what-is-http-request-smuggling-exploitations-and-security-best-practices/ (date of access: 30.11.2025).

2. What is HTTP request smuggling? Tutorial & Examples | Web Security Academy. Web Application Security, Testing, & Scanning - PortSwigger. URL: https://portswigger.net/web-security/request-smuggling (date of access: 30.11.2025).

3. NGINX Reverse Proxy | NGINX Documentation. NGINX Documentation. URL: https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/ (date of access: 30.11.2025).

4. Documentation: Apache HTTP Server - The Apache HTTP Server Project. Welcome! - The Apache HTTP Server Project. URL: https://httpd.apache.org/docs/ (date of access: 30.11.2025).

5. Interactive cybersecurity training system based on simulation environments / D. Tymoshchuk et al. Measuring and computing devices in technological processes. 2024. No. 4. P. 215–220. URL: https://doi.org/10.31891/2219-9365-2024-80-26 (date of access: 30.11.2025).

УДК 004.7:681.518:004.89

С.Б. Кіт, В.Р. Перун, С.І. Перун, Г.В. Шимчук
(Тернопільський національний технічний університет імені Івана Пулюя)

ІНТЕЛЕКТУАЛЬНА СИСТЕМА КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ В ЖИТЛОВОМУ ПРИМІЩЕННІ З ВИКОРИСТАННЯМ БСМ

S.B. Kit, V.R. Perun, S.I. Perun, G.V. Shymchuk

INTELLIGENT SYSTEM FOR CONTROLLING MICROCLIMATE PARAMETERS IN A RESIDENTIAL PREMISES USING BSM

Мікроклімат житлового приміщення безпосередньо впливає на самопочуття людини, працездатність і ризики респіраторних захворювань. У квартирах типові проблеми пов'язані з нерівномірним прогрівом кімнат, локальними зонами підвищеної вологості (кухня/ванна), накопиченням CO_2 у спальнях та появою летких органічних сполук (VOC) від меблів і побутової хімії. Тому актуальним є створення інформаційної