МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ С.В. Казмірчук

«_____»_____ 2020 р.

На правах рукопису
УДК 004.

МАГІСТЕРСЬКА АТЕСТАЦІЙНА РОБОТА

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Удосконалений модуля автоматизованої розгортки хмарного постачальника AWS

Автор: Я. В. Ковальчук
Науковий керівник: к.т.н. Н. К. Гулак
Нормоконтролер: к.т.н. Н. К. Гулак

Київ 2020

ВСТУП

Актуальність. В сучасних умовах велика кількість компаній, що діють в різних сферах, відходять від використання приватних серверів та вдаються до співпраці з платформами хмарних обчислень.

Хмарна платформа Amazon, яка створена в 2006 році стала першовідкривачем в даній області, завдяки чому отримала значну долю ринку. З постійними нововведеннями та покращеннями протягом багатьох років, AWS ввела більше 70 послуг з широким відсотком покриття по всьому світу. Серверни доступне в 14 географічних регіонах. Відсоток компанії на ринку нестримно зростає, у другому кварталі 2020 року хмарні технології компанії Amazon займали 33% ринку.

В сучасних реаліях постає необхідність безпечно розгортати інфраструктуру в Amazon Web Services через використання програмного коду. Це стосується налаштування та керування обчислювальними та мережевими ресурсами за допомогою програмних продуктів, який набагато швидший, зручніший та безпечніший на відміну від внесення необхідних налаштувань обладнання власноруч чи з допомогою додаткових інструментів.

Відомі підходи до вирішення поставленої задачі.

Існують різні рішення, які дозволяють керування обчислювальними та мережевими ресурсами через програмний код AWS CloudFormation та Terraform.

AWS CloudFormation надає можливість використовувати мову програмування YAML або звичайний текстовий файл для автоматичного безпечного моделювання та редагування ресурсів у всіх регіонах і на всіх облікових записах користувачів без необхідності виконувати ручні дії. CloudFormation самостійно визначає які операції слід виконати при керуванні стеком, впорядковує їх найбільш вдалим чином і автоматично відміняє зміни, якщо знаходить помилки.

Terraform – це популярний інструмент з відкритим кодом від компанії HashiCorp. Ця утиліта дозволяє керувати хмарною інфраструктурою в парадигмі

Infrastructure as a Code на мові прлограмування Hashicorp Configuration Language (HCL). Цей інструмент підтримує всі сучасні хмарні платформи та дозволяє безпечно та передбачувано змінити інфраструктуру.

Під час проведення дослідження було надано перевагу Terraform. Цей вибір ґрунтується на тому, шо при роботі з ним немає потреби переходити на WEB сайт постачальника, який налаштовується, та виконувати додаткові дії. Від користувача вимагається лише встановити Terraform згідно з інструкцією на офіційному сайті, вибрати зручний текстовий редактор та почати написання інфраструктури. Надалі цей вибір економить певний відрізок часу.

Метою цієї роботи вдосконалення автоматизованої розгортки інфраструктури в Amazon Web Services через розробку програмного коду для підвищення рівня безпеки інформації в хмарній мережі.

Для досягнення поставленої мети вирішуються такі задачі**:**

- аналіз загроз інформаційної безпеки та методів захисту інформації в хмарі
- вибір типу безпечного розгортання хмарної мережі в Amazon Web Services
- розробка код для модулю автоматизованої розгортки для покращення захисту інформації від несанкціонованого доступу до хмари та тестування коду.

Галузь застосування. Розроблений метод та програмне забезпечення можуть бути використані для безпечного розгортання серверів, віртуальних мереж, WEB-сайтів та інших ресурсів на базі платформи Amazon Web Services.

Об’єктом дослідження є процес захисту інформації в хмарних мережах.

Предметом дослідження є методи захисту інформації від несанкціонованого доступу в хмарних мережах.

Методи дослідження базуються на теорії інформаційної безпеки, моделі безпеки Amazon Web Services та методі попереднього аналізу небезпек.

Новизна одержаних результатів полягає в наступному:

Удосконалення модуля автоматизованої розгортки завдяки створеному коду для Terraform яке дозволило підвищити рівень безпеки інформації за рахунок зменшення часу розгортання хмарної мережі, що було протестовано на розробленій віртуальній мережі.

Практична цінність отриманих результатів: Створено код для Terraform, який розгортає віртуальну приватну хмару в AWS 2 хвилини та 13 секунд. Цей показник в 53 рази швидший, порівняно з аналогічним розгортанням вручну, відповідно і більш безпечний.

РОЗДІЛ 1
НОРМАТИВНО-ПРАВОВА БАЗА

1.1 Нормативно-правова база України

Дотримання інформаційної безпеки на території України, захищеності її інтересів в інформаційній сфері передбачає пріоритетний розвиток в сфері систем нормативно-правового регулювання у напрямку протидії загрозам цих інтересів та упорядкування відповідного правотворчого процесу.

По-перше, це зумовлено тим, що в умовах існування правової держави та громадського суспільства діяльність державних органів влади, які несуть головну відповідальність за національну безпеку, повинна керуватись визначеними правовими нормами, які існують для забезпечення конституційних права та свободи громадян. Правотворчий процес у цій сфері спрямований на нормативне закріплення мети протидії загрозам національній безпеці України, засобів та методів їх досягнення, забезпечення погодження політики органів державної влади. По-друге, вхід України в міжнародне товариство значним чином посилює можливість закріплення інформаційної безпеки держави завдяки участі в розвитку норм міжнародного права, створення міжнародної системи забезпечення безпеки сфери інформації як світу в цілому, так і кожної окремої держави. По-третє, реалізація гарантій прав та свобод громадян, захисту національних інтересів України передбачає суттєве підвищення ролі держави в регулюванні суспільних відносин, присутність відкритої та зрозумілої державної політики у цій сфері.

Нормативно-правове керування інформаційною безпекою у сфері прав та свобод здійснюється через Конституцію України і такими законами України, як: «Про інформацію», «Про науково-технічну інформацію», «Про Національну програму інформатизації», «Про Концепцію Національної програми інформатизації», «Про поштовий зв'язок» та ін. Вказані нормативно-правові акти відповідають за регуляцію аспектів забезпечення інформаційної безпеки, питання захисту інформації, охорони державної таємниці, забезпечення захисту

конфіденційної інформації, інформаційних ресурсів, покликані реалізацувати положення Доктрини безпеки особистості, держави і суспільства та ін. Можна простежити кількісний пріоритет нормативно-правових актів, як спрямовані на регулювання інформаційно-технічної безпеки, яка стосується інформаційно-психологічної та інформаційної безпеки в області прав та свобод, що пов'язано, із значним інтенсивним розвитком інформаційних технологій, а як наслідок, з потребою в оперативному реагуванні на зміни певних догматів у цій сфері. Значним недоліком в нормативно-правовому регулювання інформаційної безпеки України є розповсюдження його у значній кількості нормативно-правових актах з різною юридичною силою. Хоча важливі аспекти нормативно закріплюються в підзаконних нормативно-правових актах. Не менш необхідним аспектом для забезпечення ефективної інформаційної безпеки в Україні є неузгодженість між різними нормативно-правових актами як між собою, так і з чинною Конституцією. Головною особливістю в національному інформаційному законодавстві є існування значного масиву норм без указівок на можливі шляхи їх впровадження, внаслідок чого присутній низький рівень реалізації норм права, які відповідають за регулювання суспільних відносини у сфері забезпечення інформаційної безпеки. Крім того, існування численних відсильних норм права, багатьох нечітких, суб'єктивних понять, що потребують офіційного пояснення чи чіткого визначення, а також відсутність закріплення фундаментальних, базових термінів (наприклад, інформаційна безпека) є загрозами інформаційній безпеці України. Аналіз нормативно-правових актів у сфері забезпечення інформаційної безпеки України дозволяє зрозуміти необхідність удосконалення законодавства в сфері інформаційної безпеки. В.А. Ліпкан зазначає, що інформаційна безпека в України є частиною національної, отже, її розгляд необхідний для формування фундаментальних знань та уявлень про національну безпеку. Нормальне існування суспільства визначається розвитком, якістю функціонування і забезпеченням безпеки інформаційного середовища, а також рівнем і станом нормативного та правового забезпечення зазначених процесів. Законодавство в сфері інформаційної безпеки спрямоване на укорінення

державної інформаційної політики, яка базується на забезпеченні гарантованого рівня національної безпеки в сфері інформації, наявності значного розвитку інформаційних технологій і засобів захисту інформації, запобігання монополізму в даній області, виключення розробки деструктивних технологій впливу на людську популяцію, захист авторських і суміжних прав тощо [3, с. 194]. Інформаційна безпека це складова частина загальної проблеми в інформаційному забезпеченні людини, держави і суспільства. Вона орієнтується на захист важливих об'єктів інформаційних ресурсів та законних інтересів. Зміст поняття «інформаційна безпека» тлумачиться у діях відповідних установ, наукових дослідженнях, а також нормативно-правовій базі[3, с. 198]. Запитання дотримання національної безпеки в сфері інформаційної безпеки, на даний час, перебуває на стадії розробки. Забезпечення інформаційної безпеки відбувається шляхом впровадження єдиної державної політики безпеки в галузі інформації, через систему заходів в економічній, політичній й організаційній сфері, можливими загрозами та небезпеками національним інтересам особи, суспільства та держави в сфері інформації. Задля створення і впровадження високого рівня національної безпеки в сфері інформації необхідна розробка системи правових норм, які здійснюють регуляцію відносини в інформаційній сфері, вказують на ключові напрями в діяльності органів державного управління, створюють або змінюють органи чи шляхи забезпечення безпеки інформації і методи контролю та моніторингу за їх діяльністю. Важливим є вислів В.А. Ліпкана, який говорить, що функціонування системи, яка забезпечує інформаційну безпеку не обмежено лише значною кількістю нормативно-правових документів. Неможливо говорити про повне створення основних складових системи, яка забезпечує інформаційну безпеку. Цей фактор і не повна сформованість системи, яка забезпечує національну безпеку, і невизначеність національної політики, а отже, і політики щодо інформації. Отже, недостатня досконалість нормативно-правової бази даних процесів негативно позначається на державному управлінні у цій сфері [3, с. 220-221]. Нормативно-правова база в Україні у сфері забезпечення інформаційної безпеки є недосконалою. 11 січня

2011 р. постановою Верховної Ради України прийнято проект Закону України «Про Концепцію державної інформаційної політики України», подання Кабінетом Міністрів України (реєстр. № 7251) [4]. В преамбулі цього документа зазначено: «Ця Концепція визначає мету, принципи, пріоритетні завдання та основні напрями діяльності держави з розвитку інформаційної сфери, що включає систему виробництва, використання ресурсів і регулювання суспільних відносин, пов'язаних з одержанням, використанням, поширенням та зберіганням інформації». Це є лише окремим напрямом державної політики в сфері інформаційної безпеки, який творці документа розглядають відокремлено від загального інформаційного забезпечення життєдіяльності особи, суспільства, держави, державної внутрішньої і зовнішньої політики. Головні тези державної політики в сфері забезпечення інформаційної безпеки і безпеки інформації в сфері її обігу, в тому числі у процесах покращення співробітництва між державами, створені опираючись на методології, яка включає такі головні тези.

1. Інформаційна безпека та її складові аналізуються з точки зору функціональності в якості об'єкта управління. Існуючі підходи до видів безпеки, в тому числі в сфері інформаційної безпеки, мають певне нормативно-правове місце в межах тезису «безпека є захистом від загроз». Ця парадигма знайшла відображення у Концепції (основах державної політики) національної безпеки України, схваленої Верховною Радою України у січні 1997 р. [5], та у Законі України «Про основи національної безпеки України», прийнятому у липні 2003 р. [6]. Сьогодні очевидно, що без змін парадигми концепції щодо видів безпеки і без впровадження функціонального підходу забезпечення інформаційної безпеки є проблемними. Створення моделі загроз національній безпеці в сфері інформаційної безпеки та розробка заходів захисту здійснюють окремі державні органи. Інформаційні складові складаються з усіх напрямів дотримання національної безпеки, тобто інформаційна безпека – це спільна відповідальність усіх міністерств, відомств та інших суб'єктів України. Лише в межах функціонального підходу в інтересах інформаційної безпеки задіяти існуючий інтелектуальний, організаційний, матеріально-технічний потенціал, забезпечити

взаємодію міністерств і відомств, координацію їх діяльності. Мета полягає у тому, що функціональний підхід дозволяє оцінювати і прогнозувати стан системи управління та ефективність управлінських дій щодо попередження або нейтралізації загроз національним інтересам. Необхідність зміни парадигми концепції інформаційної безпеки випливає й із правової норми Основного Закону України, відповідно до якої інформаційна безпека віднесена до найважливішої функції держави, справи всього українського народу.

2. Захист інформаційних ресурсів України в процесі розширення співробітництва між державами представлено як складова (підсистема) загальної системи захисту інформації та важлива складова інформаційної безпеки. Головну суть інформаційної безпеки в узагальненому вигляді можна викласти як збірник превентивних дій, які спрямовані на забезпечення права на інформацію і свободи в інформаційній діяльності, на захист інформації і права власності на інформацію, на захист від інформації та від інформаційних впливів. Методи формування системи забезпечення безпеки інформації та практичне вирішення поставлених проблем говорять про те, що ефективність будь-якої підсистеми безпосередньо залежить від ефективного функціонування системи, в яку ця підсистема інкапсульована. Іншими словами, базою для покращення системи забезпечення інформаційної безпеки у процесі розширення співробітництва між державами має бути ефективно діюча загальна система забезпечення безпеки інформації як важлива складова національної безпеки України.

3. Інформація, інформаційна сфера розглядаються як системоутворюючі факторів в усіх сферах життя і діяльності соціальної системи, а інформаційна безпека – як все більш важливий фактор у стані політичної, економічної, соціальної, оборонної, власне інформаційної сфер та інших складових національної безпеки. Слід зазначити, що вимоги до інформаційної безпеки повинні бути органічно включені до всіх рівнів законодавства, в тому числі і в конституційне законодавство, основні загальні закони, закони щодо організації державної системи управління, спеціальні закони, відомчі правові акти тощо. Приведемо таку структуру правових актів, які орієнтованих на забезпечення

інформаційної безпеки держави [7, с. 36-37]. Перший пункт – конституційне законодавство. Норми, що стосуються питань інформатизації, безпеки інформації тощо, входять у нього як складові елементи. Другий пункт – загальні закони, кодекси (про власність, про надра, про землю, про права громадян, про громадянство, про податки, про антимонопольну діяльність тощо), які містять норми з питань інформаційної безпеки. Третій пункт – закони про організацію керування, які стосуються окремих структур господарства, економіки, системи державних органів та визначають їх статус. Вони містять окремі норми з забезпечення інформаційної безпеки. Поряд із загальними аспектами інформаційного забезпечення та інформаційної безпеки конкретного органу ці норми мають встановлювати його обов'язки з формування, актуалізації інформаційної безпеки, що представляють загальнодержавний інтерес. Четвертий пункт – спеціальні закони, які розповсюджуються на конкретні сфери відносин, галузі господарства, процеси. До них входить і Закон України «Про інформацію» та інші. Саме склад і зміст цього блоку законів і створює спеціальне законодавство як основу правового забезпечення інформаційної безпеки. П'ятий пункт – підзаконні нормативні акти із забезпечення інформаційної безпеки. Шостий пункт – законодавство України, що містить норми про відповідальність за правопорушення у сфері безпеки інформації. Правову основу забезпечення інформаційної безпеки України мають складати Конституція України, Концепція інформаційної безпеки України, закони України, міжнародні договори, згода на обов'язковість яких надана Верховною Радою України, а також підзаконні нормативно-правові акти, видані на їх впровадження. Слід звернути увагу, що ця сфера суспільних відносин врегульована більш ніж 30 Законами. До третього пунтку нормативно-правового забезпечення безпеки інформації необхідно віднести: загальнодержавні, міжвідомчі напрями державної політики інформаційної безпеки, що реалізуються суб'єктами у сферах їх відання; - захист прав, свобод і законних інтересів особи, суспільства, держави; забезпечення інформаційного суверенітету; формування державної інформаційної політики та підвищення її ролі в забезпеченні інформаційної політики; забезпечення безпеки

функціонування всіх елементів національного інформаційного простору та його інтегрування у світовий інформаційний простір; створення єдиної системи охорони та технічного захисту інформації обмеженого доступу, що підлягає охороні з боку держави; забезпечення безпеки інформаційно-телекомунікаційних систем, мереж зв'язку та використання Інтернету; захист національних інтересів у процесі міжнародного співробітництва; прогнозування: ризиків державної внутрішньої і зовнішньої політики, соціально-економічного розвитку, державного будівництва; потенційних і реальних інформаційних загроз, викликів і небезпек; адекватне реагування на негативні інформаційні безпекогенні чинники, виявлення, попередження і нейтралізація джерел внутрішніх і зовнішніх інформаційних загроз; участь у двосторонніх і багатосторонніх системах забезпечення міжнародної інформаційної безпеки; забезпечення загальнодержавного керівництва, координації і контролю у сфері реалізації державної політики з питань інформаційної безпеки та оцінки її результативності. Спеціальне законодавство у сфері безпеки інформаційної діяльності (інформаційної безпеки) може бути представлено сукупністю законів. В їх структурі особливе місце відведене базовому Закону «Про інформацію», який закладає основу правового визначення всіх найважливіших компонентів інформаційної діяльності: – цей закон закріплює право громадян України на інформацію, закладає правові основи інформаційної діяльності. – спеціальні закони «Про інформацію», «Про інформаційну безпеку», Концепція безпеки інформації України, закони, які забезпечують інформаційну безпеку в певних сферах діяльності осіб, суспільства, держави за напрямом державної політики забезпечення безпеки інформаційної: політичній, економічній, оборонній, державної безпеки і правопорядку, соціально-гуманітарній, науково-технологічній, екологічній, інформаційній. підзаконні нормативні акти, які врегульовують правовідносини у сфері забезпечення безпеки інформації що стосуються взаємодії органів державної влади, керування діяльністю з цих питань тощо. створення бази для наукового, матеріально-технічного, фінансового і кадрового забезпечення безпеки інформації. здійснення контролю

і нагляду за виконанням чинного законодавства з питань забезпечення безпеки інформації. Встановлення адміністративної, цивільно-правової та кримінальної відповідальності за порушення чинного законодавства про інформаційну безпеку. Слід також зауважити, що Закон України «Про основи національної безпеки України» визначає дев'ять сфер національної безпеки: зовнішньополітичну, державну безпеку, економічну, соціальну та гуманітарну, воєнну та сферу безпеки державного кордону України, внутрішньополітичну, екологічну, науково-технологічну та інформаційну. Отже, інформаційна безпека є складовою національної безпеки. У законі досить системно формулюються основні функції, які має виконувати система забезпечення національної безпеки в усіх зазначених сферах. Їх конкретизація (з урахуванням особливостей інформаційної сфери) дає можливість визначити основні функції системи забезпечення інформаційної безпеки України. Але при загальному зростанні кількості законів, що приймаються, поки що залишається багато таких сфер життя суспільства і громадян, які ще не врегульовані на законодавчому рівні, що дає простір відомчій правотворчості. Національний банк, Державна податкова адміністрація, Державна митна служба, Фонд держмайна, інші центральні та місцеві органи виконавчої влади буквально «наплоджують» нормативні правові акти, до яких самі ж з не меншою швидкістю вносять зміни та доповнення. Недостатність правового врегулювання правової бази щодо інформаційних правовідносин значно ускладнює настання якісних змін у цій сфері суспільних відносин. На сьогодні у зв'язку з відсутністю взаємопов'язаних, чітко розроблених заходів та теоретичних розробок із забезпечення інформаційної безпеки держави маємо цілу низку перешкод на шляху повноцінної реалізації державою свого обов'язку щодо забезпечення інформаційної безпеки, яка, у свою чергу, є невід'ємною частиною національної безпеки. Лише реалізація науково обґрунтованої державної інформаційної політики може створити ефективну систему протидії правопорушенням у цій сфері. Постала нагальна потреба в розробленні єдиного комплексного системоутворюючого законодавчого акта, який би забезпечив: - створення єдиної стратегії реалізації

державної політики у сфері інформаційної безпеки; - розроблення організаційно-правових механізмів забезпечення інформаційної безпеки; - визначення правового статусу суб'єктів інформаційних відносин, встановлення їх відповідальності за дотримання національного законодавства у цій сфері; - створення системи підготовки кадрів, які використовуються в галузі забезпечення інформаційної безпеки [8, с. 46].

1.2 Світова нормативно-правова база

На сьогодні головними організаціями, що займаються питаннями безпеки в хмарі, є Альянс безпека в хмарі (Cloud Security Alliance, CSA), що складається з представників IT-індустрії, а також дві державні організації Європи та США: Європейське агентство мережної та інформаційної безпеки (ENISA) і Національний інститут стандартів і технологій (NIST).

Кожна з організацій створила відповідний нормативний документ, в якому подано класифікацію всіх існуючих проблем інформаційної безпеки(ІБ) в хмарі. Розглянемо їх та проведемо порівняння[8, с. 65]..

1.2.1 CSA підхід

CSA є некомерційною організацією, яку було створено наприкінці 2008 року. Її засновниками стали великі IT-компанії, зацікавлені у впровадженні хмарних технологій: Google, Microsoft, IBM, Salesforce.com, VMware та інші.

Головним документом, який розглядає проблеми безпеки в хмарі, є «Керівництво з безпеки критичних областей для хмарних обчислень». Перша версія його була опублікована в 2009 році. Актуальною є остання, 3 версія цього документу, яка доступна на офіційному сайті організації.

Основними складовими вимог ІБ у хмарах, що рекомендуються до розгляду та аналізу, є такі [4]:

1 Організаційні та правові питання ІБ

1.1 Управління ризиками.

1.1.1 Корпоративне управління ризиками.

1.1.2 Управління ризиками підприємства та постачальника послуг.

1.1.3 Управління інформаційними ризиками

1.2 Правові питання та законодавство.

1.2.1 Стандартизація, міжнародне законодавство, узгодження законодавства держав.

1.2.2 Договір між постачальником та клієнтом.

1.2.3 Право власності на інформацію, що обробляється в хмарі.

1.3 Відповідність вимогам та аудит.

1.3.1 Проведення експертизи; моніторинг, тестування та оновлення програмного та апаратного забезпечення постачальника.

1.3.2 Дотримання існуючих міжнародних та державних законів.

1.3.3 Аудит безпеки постачальника послуг.

1.4 Управління інформацією та безпекою даних.

1.4.1 Безпека даних (запобігання витоку, несанкціонованого доступу, втраті тощо).

1.4.2 Управління життєвим циклом даних (створення, зберігання, використання, обмін, архівування, відновлення, знищення).

1.4.3 Керування розташуванням даних.

1.4.4 Управління авторським правом (DRM).

1.5 Переносимість та інтероперабельність.

1.5.1 Сумісність апаратного, програмного забезпечення, архітектурних рішень постачальників.

1.5.2 Стандартизований інтерфейс взаємодії з постачальником хмарних послуг.

2 Технічні питання ІБ

2.1 Традиційна безпека забезпечення безперервності бізнесу та аварійного відновлення.

2.1.1 Захист від стихійних лих, техногенних

катастроф та ін.

2.1.2 Захист від людського фактору та обслуговуючого персоналу.

2.2 Операції центру обробки даних.

2.2.1 Взаємодія між центрами постачальника.

2.2.2 Взаємодія між різними постачальниками.

2.3 Реагування на інциденти, повідомлення про них та відновлення.

2.3.1 Попередження виникнення інцидентів ІБ.

2.3.2 Визначення та аналіз інцидентів ІБ.

2.3.3 Відновлення після інцидентів безпеки.

2.4 Безпеки додатків та програм.

2.4.1 Контроль якості, тестування на відмову, безпечність програмного забезпечення.

2.4.2 Розмежування доступу користувачів до додатків.

2.4.3 Моніторинг активності додатків.

2.4.4 Виявлення небезпечних програм та забезпечення безпеки існуючих.

2.5 Шифрування та управління ключами.

2.5.1 Альтернативні підходи до шифрування даних у хмарі.

2.5.2 Криптографія в хмарі.

2.5.3 Шифрування баз даних.

2.5.4 Управління ключами в хмарі (генерація, використання, зберігання, знищення, відновлення).

2.6 Ідентифікація і управляння доступом.

2.6.1 Моделі ідентифікації в хмарі.

2.6.2 Керування профілями користувачів.

2.6.3 Надання послуг ідентифікації, автентифікації, спільного доступу до інформації в хмарі або ресурсів.

2.6.4 Реалізація ідентифікації користувачів у програмному забезпеченні.

2.6.5 Доступність, цілісність даних, доступ до даних авторизованих користувачів.

2.7 Віртуалізація.

2.7.1 Забезпечення захисту гістьової віртуальної машини від атак.

2.7.2 Механізми захисту від неправомірних дій адміністраторів.

2.7.3 Питання швидкодії, пікового збільшення навантаження, збільшення числа вузлів.

2.7.4 Забезпечення безпеки даних на рівні віртуальної машини.

2.7.5 Забезпечення цілісності образів віртуальних машин.

2.8 Безпека як сервіс.

2.8.1 Продаж послуг безпеки.

2.8.2 Проблеми при реалізації послуги безпеки.

2.8.3 Класифікація послуг безпеки.

Документ окрім питань ІБ містить також концепції архітектури побудови хмари та надає рекомендації та шляхи вирішення цих проблем. У цілому питання ІБ у хмарі поділяються на дві великі групи: питання управління ІБ у хмарі (організаційні питання ІБ) та ІБ у хмарі під час її використання (технічні питання ІБ). Кожна з груп розбита на більш малі, що називаються доменами. Домени, що відносяться до організаційних, у першу чергу розглядаються з метою вироблення рішень правових питань, питань політики ІБ, управління ризиками та стандартизація. В рамках технічних питань розглядаються питання реалізації та впровадження захисту в хмарі.

1.2.2 ENISA підхід

Європейське агентство з мережної та інформаційної безпеки (ENISA) є організацією, діяльність якої спрямована на «підвищення здатності Європейського Союзу, держав-членів ЄС та бізнес-спільноти на попередження, ліквідацію і реагування на проблеми мережної та інформаційної безпеки» [9, с. 42]..

Організацією ENISA було підготовлено і опубліковано документ “Безпека хмарних обчислень та оцінка ризиків” [10, с. 86]., в якому були розглянуті питання інформаційної безпеки в хмарі, їх переваги та недоліки, існуючі ризики, аналіз та шляхи їх зменшення, існуючі загрози в середовищі хмарних обчислень. Згідно з цим документом можна виділити такі ризики ІБ, які існують в хмарі :

1 Організаційні питання ІБ

1.1 Втрата можливості керування користувачем деякими налаштуваннями безпеки в хмарі.

1.2 Замкнутість користувача на одному постачальнику послуг у зв’язку з відсутністю переносимості та інтероперабельності розгорнутої інфраструктури.

1.3 Дотримання вимог стандартів.

1.4 Втрати ділової репутації постачальника.

1.5 Припинення роботи сервісом хмари.

1.6 Відмова в роботі одного з постачальників.

1.7 Придбання провайдера хмарних послуг.

2 Правові питання ІБ

2.1 Судовий розгляд та законодавство в сфері електронних даних.

2.2 Ризик зміни юрисдикції.

2.3 Ризики щодо захисту даних.

2.4 Ризики ліцензування.

2.5 Право власності електронних даних.

3 Технічні питання ІБ

3.1 Порушення ізоляції даних користувачів.

3.2 Часткове або неповне знищення даних користувача.

3.3 Вичерпання ресурсів.

3.4 Загроза інсайдерів в інфраструктурі постачальника послуг.

3.5 Ризики інтерфейсу управління.

3.6 Перехоплення даних зловмисником при передачі.

3.7 Витік даних при завантаженні та скачуванні.

3.8 Розподілена відмова в обслуговуванні(DDoS-атака).

3.9 Економічна відмова в обслуговуванні(EDoS-атака).

3.10 Втрата ключів шифрування.

3.11 Проведення сканування та тестування з метою виявлення вразливостей.

1.2.3 NIST підхід

Задля регуляції хмарних обчислень уряд США в організації NIST замовив розробку стандарту з забезпечення безпеки та конфіденційності в громадських хмарах. Тому починаючи з 2011 року NIST створив ряд нормативних документів,

які давали визначення хмарним обчисленням, розглядали питання ІБ у хмарі, пропонували архітектуру безпеки в хмарі, давали рекомендації з оцінки та усунення існуючих ризиків ІБ у хмарі.

Класифікація питання ІБ у хмарі розглядається в таких документах NIST: «Посібник з безпеки та конфіденційності в громадських хмарних обчисленнях» [7] та «Короткий огляд хмарних обчислень та рекомендації» [11, с. 21].. На відміну від розглянутих питань ІБ у хмарі CSA та ENISA, в документації NIST питання ІБ чітко не поділяють на такі рівні як організаційні питання, правові питання та технічні питання ІБ. Розглянемо їх в натуральному вигляді.

1 Керування

1.1 Контроль і нагляд урядової організації за політикою, процедурами та стандартами в ході розробки додатків та інформаційних технологій, одержання послуг, а також проектування, впровадження, тестування, використання та моніторинг розгорнутих хмар.

2 Дотримання законів, правил, стандартів та специфікацій

2.1 Дотримання міжнародних та державних стандартів, законів і правил.

2.2 Дотримання законів та правил держав, та їх застосування до даних хмари, що фізично розташовуються в межах цієї держави.

2.3 Законодавство в сфері електронних даних.

2.4 Підтримка в проведенні експертизи.

3 Довіра до постачальника послуг

3.1 Доступ до конфіденційної інформації осіб (інсайдерів), завдяки своєму службовому становищу.

3.2 Право власності електронних даних.

3.3 Складені сервіси та сервіси, які використовують сервіси хмари, надані третьою стороною.

3.4 Аудит постачальника, моніторинг, тестування та оновлення програмного та апаратного забезпечення постачальника.

3.5 Захист персональних даних користувача.

3.6 Управління ризиками.

4 Архітектура програмного і апаратного забезпечення

4.1 Зовнішні атаки на інфраструктуру.

4.2 Захист віртуальної мережі.

4.3 Захист образів віртуальних машин.

4.4 Клієнтський захист.

4.5 Захист серверів постачальника.

4.6 Моніторинг захисту.

5 Ідентифікація і управління доступом

5.1 Автентифікація.

5.2 Контроль доступу.

5.3 Спільний доступ до інформації в хмарі.

5.4 Управління ключовими даними.

6 Ізоляція програмного забезпечення

6.1 Складність ОС, програмного чи апаратного забезпечення, призначеного для розміщення та роботи віртуальної машини.

6.2 Загрози, пов’язані з іншими користувачами віртуальних машин.

7 Захист даних

7.1 Концентрація даних.

7.2 Ізоляція даних.

7.3 Безпечне зберігання, відновлення, архівування, видалення даних.

8 Доступність ресурсів та даних

8.1 Відключення хмарних сервісів (тимчасове, тривале, постійне).

8.2 Атаки DDoS.

8.3 Загрози, пов’язані з розташуванням даних.

9 Реагування на інциденти

9.1 Моніторинг наявності та доступності даних.

9.2 Аналіз інцидентів та їх розв’язання.

1.2.4 Порівняння підходів CSA, ENISA, NIST

Порівняння відбувається за трьома основними складовими групами: правові, організаційні та технічні питання ІБ у хмарі. Базуючись на розглянутих

класифікаціях, було виділено питання ІБ до кожної з груп, що наведені в таблицях 1, 2 та 3. Якщо питання ІБ було розглянуто в класифікації повністю, воно відмічено як «+», якщо частково – «+/-», в разі відсутності – «-».

Аналіз даних таблиць показує, що в основному складові ІБ збігаються в усіх класифікаціях. Найбільш повна та структурована класифікація була надана організацією CSA, але її недоліком є об'єднання правових та організаційних проблем ІБ. Головною перевагою класифікації ENISA є оцінка ймовірності виникнення ризиків, пов'язаних з ІБ, причинами їх виникнення, взаємозв'язки з іншими ризиками, та їх вплив на систему та її елементи.

Таб. 1.1

Порівняння правових складових ІБ

№	Правові питання ІБ	Класифікація		
		CSA	ENISA	NIST
1	Дотримання міжнародних та державних стандартів, законів і правил	+	+	+
2	Договір між постачальником та клієнтом	+	+	+
3	Право власності на електронні дані	+	+	+
4	Невідповідність законодавств різних держав у сфері електронних даних	+	+	+
5	Захист авторського права (DRM)	+	-	-
6	Дотримання законів та правил держав до даних у хмарі	+	+	+
7	Зміна постачальника послуг, або його купівля іншим постачальником	+	+	+

Таб. 1.2

Порівняння організаційних складових ІБ

№	Організаційні питання ІБ	Класифікація		
		CSA	ENISA	NIST
1	Управління ризиками (корпоративними, підприємства, інформаційними, постачальника послуг)	+	+	+
2	Управління безпекою інформації користувача	+	+	+
3	Довіра до постачальника послуг (проведення аудиту, тестування, оновлення забезпечення, підтримка в проведенні експертизи)	+	+	+
4	Захист від інсайдерів	+	+	+
5	Реагування на інциденти ІБ, їх моніторинг, вирішення	+	+	+
6	Захист персональних даних користувача	-	-	+
7	Управління авторськими правами	+	+	+
8	Відмова сервісів хмари по причині стихійного лиха, збоїв у роботі сервісів хмари, що підтримуються третьою стороною	+	+	+

Таб. 1.3

Порівняння технічних складових ІБ

№	Технічні питання ІБ		Класифікація		
			CSA	ENISA	NIST
1	Доступність даних та ресурсів	1.1 Відключення хмарних сервісів	+	+	+
		1.2 Атаки DDoS	-	+	+
		1.3 EDoS-атака	-	+	-
		1.4 Розташування даних	+	+	+
		1.5 Вичерпання ресурсів	+	+	-
2	Переносимість та інтероперабельність забезпечення	2.1 Сумісність забезпечення	+	+	+
		2.2 Стандартизований інтерфейс	+	-	-
3	Безпеки додатків та програм	3.1 Безпечність ПЗ	+	+	+
		3.2 Розмежування доступу	+	+	+
		3.3 Моніторинг активності додатків	+	+	+
		3.4 Виявлення небезпечних програм	+	+	+
		3.5 Захист образів віртуальних машин від модифікації	+	-	+
4	Управління даними та захист	4.1 Ізоляція даних	+	+	+
		4.2 Безпечне зберігання та оброблення даних	+	+	+
		4.3 Шифрування даних	+	+	+
		4.4 Управління ключами	+	+	+
5	Ідентифікація, автентифікація та управляння доступом	5.1 Моделі ідентифікації та автентифікація в хмарі	+	-	-
		5.2 Керування профілями користувачів у хмарі	+	+	+
		5.3 Надання послуг ідентифікації, автентифікації, спільного доступу до інформації в хмарі або ресурсів	+	+	+
		5.4 Реалізація ідентифікації користувачів	+	+/-	+
		5.5 Доступ до даних авторизованих користувачів	+	+	+
6	Віртуалізація	6.1 Забезпечення захисту гістьової віртуальної машини від атак	+	-	+
		6.2 Механізми захисту від неправомірних дій адміністраторів	+	+	+
		6.3 Питання швидкодії, пікового збільшення навантаження, збільшення числа вузлів	+	+	+
		6.4 Забезпечення безпеки даних на рівні віртуальної машини	+	-	+

1.2.5 Аналіз проблемних питань захисту інформації в хмарі

Більшість з проблем захисту інформації користувача в хмарі може бути вирішено з використанням існуючих методів криптографічного захисту інформації, адміністративних мір з боку як постачальника хмарних послуг, так і користувача, укладання договорів на надання послуг, які б враховували індивідуальні потреби клієнтів, прийняття міжнародних стандартів у галузі, введення контролю з боку держави та створення незалежних експертів у цій галузі. Так, наприклад, для забезпечення конфіденційності та цілісності даних, що зберігаються в хмарі, необхідно використовувати алгоритми цифрового підпису та шифрування, які засновані на міжнародних стандартах. Для запобігання несанкціонованого використання профілю користувача можна використовувати існуючі методи двофакторної автентифікації користувача. Сьогодні більшість постачальників мають свій власний, іноді навіть добре документований інтерфейс для програмування, але це призводить до неможливості переходу користувачів від одного постачальника послуг до іншого. Практика в таких питаннях показує, що лише розробка відкритого

єдиного міжнародного стандарту може вирішити це питання. Головними проблемами, які потребують подальшого детального аналізу та вирішення, є такі:
а) Проблема привілегійованих користувачів. Найбільшу загрозу для безпеки інформації в хмарі становлять користувачі, які мають привілегійований доступ до функцій системи або адміністратори хмарних сервісів, тому для зменшення ризику можливих деструктивних дій з їх боку, доцільно вести незалежний нагляд та контроль за їх діями в хмарі. Як показує статистика саме на внутрішніх користувачів припадає найбільша кількість порушень безпеки. [12, с. 127].

б) Однією з головних проблем, що гальмує поширення хмарних обчислень, є невідповідність законів у сфері обробки, передачі, збереження та захисту інформації різних держав. Вирішення цієї проблеми є ключовим фактором для можливості фізичного розміщення серверів постачальника хмарних сервісів у різних країнах та регіонах, а також використанням користувачами з різних країн одного постачальника послуг.

Ця проблема найбільш істотно торкатиметься транснаціональних корпорацій.

в) Питання довіри до постачальника послуг можуть бути вирішені лише за рахунок проведення аудиту безпеки постачальника хмарних послуг та перевірки відповідності його системи безпеки міжнародним вимогам до захисту інформації, що сформульовані в міжнародних стандартах. Формулювання та обґрунтування вимог є одним з важливих питань.

г) Питання загальних вразливостей у хмарі практично нічим не відрізняються від аналогічних у традиційних системах, за винятком того, що знайдена одна вразливість може бути використана для всієї хмари, але водночас її можна легше виправити за допомогою централізованого оновлення, на відміну від традиційних систем. І в цей час її критичність набагато більша, бо вона може з легкістю уразити всіх користувачів даного постачальника послуг, тому потребує превентивних мір та засобів захисту.

д) Проблеми доступності до сервісів та даних

користувачами, відновлення їх роботи після збоїв, чи втрати даних повинні вирішуватися на адміністративному та правовому рівнях. При укладанні договорів з користувачем мають бути чітко визначені обов'язки сторін та міра їх відповідальності в залежності від обставин події, що призвела до цих наслідків, а розслідування повинна проводити третя незалежна сторона. Аналогічна проблема існує і в традиційних системах, але користувач має можливість безпосередньо впливати на рівень резервування в системі, що дає можливість більш гнучко її налаштовувати під конкретні вимоги користувача та його фінансові можливості.

є) Проблема надання доступу, спільного доступу та блокування доступу до ресурсів і даних у хмарі користувачам.

е) Проблема захисту інтелектуальної власності в хмарі, зокрема програмного забезпечення та даних.

1.3 Концепції інформаційної безпеки в хмарі

1.3.1 Вимоги до безпеки даних в хмарі

Хмарні обчислення – це модель, яка забезпечує повсюдний та зручний доступу через мережу до спільного набору обчислювальних ресурсів, що підлягають налаштуванню, наприклад, до комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів тощо. Вони можуть бути швидко надані та звільнені з мінімальними експлуатаційними затратами або зверненням до провайдера. Основними моделями розгортання хмарних сервісів є такі: приватна, громадська, публічна та гібридна хмари. Ознакою такої класифікації є категорії користувачів, що мають доступ та можуть використовувати ресурс та дані хмари. При цьому постачальники хмарних ресурсів можуть надавати, а користувачі отримувати такі послуги, як: програмне забезпечення як послуга (SaaS), платформа як послуга (PaaS) та інфраструктура як послуга (IaaS). Аналіз показує, що незалежно від моделі розгортання та обслуговування хмари, всі ключі, що використовуються в середовищі хмарних обчислень, можна поділити за призначенням та власником на такі два класи :

– ключі, що використовуються провайдером хмарних послуг та є його власністю;

– ключі, що використовуються клієнтами провайдера хмарних послуг та є їх власністю.

Наприклад, якщо хмара розгорнута як публічна та надає послуги PaaS, то користувач хмари на основі сервісу, що надається провайдером, реалізує свої рішення, послугами якого користуються клієнти користувача. За цих умов користувач для своїх клієнтів буде виступати в якості провайдера хмарних послуг, а отже в цьому випадку будуть існувати також два класи ключів:

– клас ключів провайдера хмарних послуг, до якого відносяться ключі провайдера хмарних послуг публічної хмари, що надає послуги PaaS та ключі користувача хмари, по відношенню до клієнтів користувача;

– клас ключів користувача хмарних послуг, до якого відносяться по відношенню до провайдера хмарних послуг, ключі користувача хмарних послуг та ключі клієнтів користувача хмарних послуг.

Аналогічним чином можна виділити та показати існування лише двох класів ключів для інших моделей розгортання та надання послуг в хмарі. Така модель хмари, у якій існує тільки дві ролі – користувач та провайдер, на відміну від моделі NIST, дозволяє зменшити складність аналізу безпеки управління ключами, включаючи крипто живучість. Це досягається за рахунок виключення ролі аудитора хмари, яка виступає в якості пасивного елементу хмари, та має доступ до хмари лише під час проведення аудиту з використанням строгого переліку доступних можливостей. Вказане є справедливим і до посередника (брокера) хмарних послуг, який для провайдера хмарних послуг розглядається з точки зору клієнта, а для клієнта брокер є провайдером хмарних послуг. Теж саме можна прийняти і до транспортувальника хмарних послуг – в моделі безпеки його головною задачею повинно бути забезпечення доступності сервісів, забезпечення конфіденційність та цілісність даних, що передаються забезпечується тільки користувачем та провайдером хмарних послуг.

Ґрунтуючись на наведеному, розглянемо модель порушника хмарних обчислень, на основі якої побудуємо модель загроз відносно управління ключовими даними.

1.3.2 Модель порушника хмарних обчислень

При побудові моделі порушника застосовується методика, що ґрунтується спочатку на побудові моделі порушника, виявленні усіх можливих загроз та визначенні способів їх реалізації, а на останок – побудові моделі загроз. Побудовані моделі порушника та загроз дозволяють сформулювати вимоги до системи захисту інформації в середовищі хмарних обчислень. Під моделлю порушника будемо розуміти абстрактний формалізований чи неформалізований опис дій порушника, який відображає його практичні та теоретичні можливості, апріорні знання, час та місце дії тощо. Складність побудови моделі порушника для інформаційно-телекомунікаційної системи (ІТС) хмарних обчислень полягає в необхідності врахування моделі розгортання хмари, моделі надання послуг, власника та рівень контролю інформації, а також рівень контролю провайдера та користувачів над інфраструктурою хмари. Також така модель має бути ще адекватною реальному порушнику. Аналіз моделі хмари NIST показує, що по відношенню до ІТС хмари порушники можуть бути внутрішніми (з числа співробітників, користувачів системи) або зовнішніми (сторонні особи). Особливу небезпеку в середовищі хмарних обчислень становлять внутрішні порушники. Навіть при наданні послуг на рівні IaaS, внутрішня інфраструктура хмари, в тому числі і середовище передачі даних, контролюється провайдером хмарних послуг.

Рис 1.1 Модель хмарних обчислень

Наведена на рис. 1.1 модель визначає:

– категорії осіб, з числа яких може бути порушник;

– припущення про рівень можливостей, кваліфікацію, рівень ознайомлення з системою, характер дій порушника;

– методи та засоби, що може використовувати порушник;

– мета, яку перед собою ставить порушник;

– елементи системи, які порушник буде атакувати.

За категорією осіб порушниками можуть бути:

– внутрішні порушники: працівники провайдера хмарних послуг, працівники користувача, сторонні особи, що отримують доступ до ресурсів ІТС хмари. Для їх визначення слід детально розглянути можливості несанкціонованого доступу до ресурсів ІТС кожного із працівників провайдера та користувача, а також можливості сторонніх осіб щодо несанкціонованого доступу до ресурсів ІТС з урахуванням наявної системи організаційного обмеження їх доступу;

– зовнішні порушники: інші особи, що не мають безпосереднього доступу до ресурсів ІТС хмари. Для їх визначення слід детально розглянути можливі канали витоку інформації та вразливі місця системи. За рівнем можливостей порушників розділимо на чотири класи:

– перший рівень припускає можливість запуску фіксованого набору завдань (програм), що реалізують заздалегідь передбачені функції обробки інформації;

– другий рівень визначається можливістю створення і запуску власних програм з новими функціями обробки інформації;

– третій рівень визначається можливістю управління функціонуванням ІТС хмари, тобто впливом на базове програмне забезпечення системи, на склад і конфігурацію її устаткування;

– четвертий рівень визначається всім обсягом можливостей осіб, що здійснюють проектування, реалізацію і ремонт апаратних компонентів ІТС хмари, з можливістю включення до складу ІТС хмари власних засобів з новими функціями обробки інформації.

Таб. 1.4

Профіль порушника відносно хмарних сервісів

№	Категорія	Значення (Позначення)
1	Категорія осіб	внутрішній (В)
		зовнішній (З)
2	Характер дій	випадковий (В)
		пасивний (П)
		активний (А)
3	Рівень доступу та можливостей	перший рівень (1)
		другий рівень (2)
		третій рівень (3)
		четвертий рівень (4)
4	Рівень ознайомленості	користувач (К)
		спеціаліст (С)
		адміністратор системи (А)
		спеціаліст з найвищим рівнем знань (Н)
		адміністратор безпеки (Б)
5	Методи та засоби	агентурні (Р)
		штатні (Ш)
		пасивні (П)
		активні (А)
6	Мета дій порушника	отримання атрибутів доступу персоналу чи користувачів АС (О)
		отримання несанкціонованого доступу до обчислювальних ресурсів хмари (НСД)
		проникнення на територію хмарного ЦОД з метою впливу на фізичне обладнання (ВФ)
		зміна режимів функціонування чи виводу з ладу фізичних ресурсів АС (М)
		встановлення засобів технічної розвідки (ТР)
		встановлення технічних засобів нав'язування (ТЗН)
		встановлення програмних закладок розвідки (ПЗР)
		встановлення програмних засобів нав'язування (ПЗН)

За рівнем ознайомлення з системою, порушників будемо класифікувати за такими рівнями, як:

– що не володіють спеціальними знаннями з обчислювальної техніки та програмування, проектування та експлуатації хмарної ІТС, а є лише користувачами сервісу;

– що володіють базовим або високим рівнем знань у галузі обчислювальної техніки та програмування, проектування та експлуатації хмарних ІТС, а також базовим або високим рівнем знать про системи захисту хмарних ІТС;

– що володіють інформацією про функціональні особливості визначеної ІТС хмари, основні закономірності формування в ній масивів даних та потоків запитів до них, вміють користуватися штатними засобами;

– що володіють високим рівнем знань та досвідом роботи з технічними засобами системи хмари та їхнього обслуговування;

– що володіють інформацією про функції та механізм дії засобів захисту в визначеній ІТС хмари.

Аналіз показує, що незалежно від моделі розгортання хмари та моделі надання послуг найнебезпечнішими порушниками в ІТС хмари є адміністратори хмари та адміністратори безпеки хмари. Порушення з боку цих осіб можуть бути як ненавмисними, так і зловмисними. При цьому якщо «випадковий порушник» здійснює загрози ІТС під час виконання своїх функціональних обов’язків внаслідок помилкових дій, за рахунок неуважності чи недбалості, то порушник, що здійснює зловмисне порушення, чітко розуміє свої дії та має змогу проаналізувати їх вплив. Також необхідно брати до уваги, що порушник в середовищі хмари може здійснювати активні чи пасивні загрози ресурсам ІТС чи АС. Під активною загрозою слід розуміти навмисні та не навмисні несанкціоновані дії порушника, що призвели до зміни стану ІТС (АС), а під пасивною загрозою – дії, що призвели до несанкціонованого проникнення в систему без зміни її стану.

За характером дій порушників можна класифікувати на :

– «випадковий порушник» – це користувачі, обслуговуючий персонал, які не навмисно подолали засоби управління (адміністрування) доступом до об’єкту

захисту, виконали непередбачені дії відносно цього об’єкту, чим спричинили порушення політики безпеки до об’єкта захисту;

– «пасивний порушник» – авторизований користувач, або обслуговуючий персонал хмари, який навмисно порушив політику безпеки послуги, але не вживає рішучих дій. З метою прихованого подолання засобів управління (адміністрування), використовує атрибути доступу інших користувачів;

– «активний порушник» – порушник, який не приховує своїх дій та може використовувати всі доступні методи та засоби для порушення властивості захищеної інформації. До таких дій відносяться дії, що спрямовані на подолання організаційного обмеження доступу, охоронної сигналізації, управління доступом до фізичних ресурсів, фізичний доступ до засобів оброблення, зберігання чи передавання інформаційних об’єктів з метою виведення їх з ладу, зміни режимів функціонування, крадіжки чи пошкодження носіїв тощо;

– «віддалений порушник» – порушник, що використовує засоби віддаленого доступу до інформаційних об’єктів: виток інформації технічними каналами, спеціальний вплив на інформацію по технічним каналам, мережне обладнання локальних чи розподілених мереж, в тому числі і засоби телекомунікаційних мереж.

За методами та засобами, що використовує порушники, їх можна класифікувати як таких, що використовують:

– виключно агентурні методи одержання відомостей;

– пасивні технічні засоби перехоплення інформаційних сигналів;

– для реалізації спроб НСД виключно штатні засоби АС або недоліки проектування КСЗІ;

– способи і засоби активного впливу на АС, що змінюють конфігурацію системи (підключення додаткових або модифікація штатних технічних засобів, підключення до каналів передачі даних, впровадження і використання спеціального ПЗ тощо). Також при подальших дослідженнях будемо вважати, що метою зловмисника, при здійсненні порушення, може бути наступне:

– авторизація та отриманні атрибутів доступу з найбільшими правами до ресурсів ІТС з метою їх використання для ознайомлення з конфіденційною інформацією, її модифікації чи знищення, використання обчислювальних ресурсів хмари в своїх власних цілях;

– пошук та/або здобуття атрибутів доступу особи з персоналу чи користувачів АС, які мають атрибути доступу з найбільшими правами, з метою заволодіння їх атрибутами доступу. Здобуття атрибутів особи може бути реалізовано шляхом використання технічних засобів, крадіжок, купівлі, чи отримання іншим шляхом;

– проникнення на місця розміщення тих чи інших компонентів, елементів чи ресурсів АС (обчислювальних ресурсів, інформаційних ресурсів, базового, прикладного програмного забезпечення та програмного забезпечення системи ТЗІ, включаючи носії резервних копії, ресурсів вводу/виводу, телекомунікаційного обладнання, включаючи мережу передачі даних) шляхом подолання перешкод (огорожі, елементів будівельних конструкцій, охорони чи охоронної сигналізації та ін.) та нанесення збитків шляхом знищення матеріальних та інформаційних цінностей; – зміна режимів функціонування чи виводу з ладу ресурсів АС;

– установка фізичних засобів (апаратурних закладок) чи інших засобів технічної розвідки в місцях розміщення елементів АС (в тому числі і віддалених, наприклад в елементах комунікаційної мережі зв’язку) для знімання інформації; – установка фізичних чи інших засобів (апаратурних закладок) в місцях розміщення елементів АС (в тому числі і віддалених, наприклад, в елементах комунікаційної мережі зв’язку) для генерації несправжніх сигналів, інформаційних символів чи повідомлень; – установка програмних засобів (програмних закладок) знімання інформації з метою її того чи іншого використання; – установка програмних засобів (програмних закладок чи вірусів) для модифікації як програмних засобів, так і інформації АС, шляхом генерації (впровадження) програмних вірусів, несправжніх сигналів, інформаційних

символів чи повідомлень з метою перевантаження систем АС і порушення, таким чином, доступності компонентів АС чи АС в цілому;

– здійснення спроб несанкціонованого доступу до обчислювальних ресурсів, інформаційних ресурсів, базового та прикладного програмного забезпечення та програмного забезпечення системи ТЗІ як власне АС, так і її телекомунікаційної підсистеми шляхом подолання системи управління доступом.

Важливим також є визначення та використання в моделі того, яким чином загрози можуть бути реалізовані при хмарних обчисленнях. Ґрунтуючись на, будемо вважати, що вони реалізуються наступними способами:

– технічними каналами, що включають канали побічних електромагнітних випромінювань і наводок, акустичні, віброакустичні, акустоелектричні, оптичні, радіо- та радіотехнічні, хімічні та інші канали;

– каналами спеціального впливу шляхом формування полів і сигналів з метою руйнування системи захисту або порушення цілісності інформації;

– несанкціонованим доступом шляхом підключення до апаратури та ліній зв’язку, маскування під зареєстрованого користувача, подолання заходів захисту з метою використання інформації або нав’язування хибної інформації, застосування закладних пристроїв чи програм та вкорінення комп’ютерних вірусів, як на стороні користувача так і провайдера хмарних послуг.

1.3.3 Модель загроз хмарним сервісам

Наведені вище в п. 1.3.2 потенційні можливості порушника дозволили розробити модель загроз відносно хмарних сервісів (обчислень). Детально опис такої моделі загроз наведена в таб. 1.5. В ній вказується об’єкт, для якого реалізується загроза, мета порушника, ймовірність загрози та мета здійснення захисту.

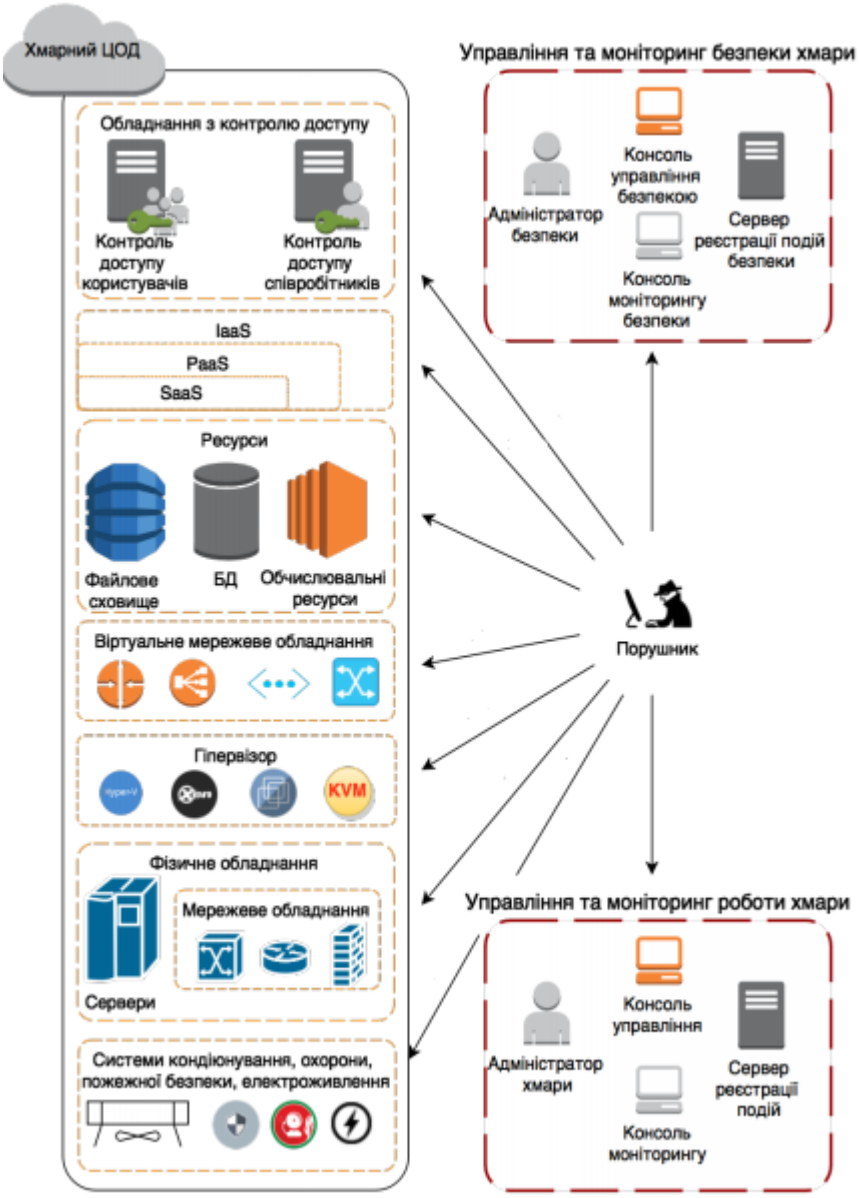


Рис 1.2 Модель загроз для хмарних обчислень

Класифікація загроз за ймовірністю була проведена з урахуванням рекомендацій [8, с. 23-56]. Згідно цих рекомендацій, найбільшу ймовірність мають загрози, що здійснюються на компоненти хмарної інфраструктури, які мають інтерфейси доступу з зовні та/або знаходяться в віртуалізованому середовищі. Аналіз моделі загроз, зображеної на рис. 1.2, показав, що найбільшу небезпеку становлять загрози управління хмарою та її безпекою , а також загрози гіпервізору.

Таб. 1.5

Загрози в середовищі хмари

Загроза	Об'єкт для якого реалізується загроза	Мета	Ймовірність загрози	Методи захисту
З.1	Управління та моніторинг безпеки хмари	Отримання несанкціонованого доступу до хмари	низька	Використання систем з контролю доступу, політики безпеки, атестація персоналу
З.2	Обладнання з контролю доступу	Отримання несанкціонованого доступу до хмарних ресурсів чи управління хмарою	висока	Використання захищених носіїв ключа для автентифікації
З.3	Середовище хмари: сервіси, додатки та інфраструктура	Порушення сервісами, додатками та об'єктами інфраструктури прав доступу. Несанкціонований доступ до функцій управління інфраструктурою, даних сервісів та додатків користувача хмари. Зараження шпигунським ПЗ та вірусами	висока	Використання технологій розмежування та обмеження доступу, контроль над цілісністю об'єктів та їх моніторинг
З.4	Ресурси хмари	Отримання несанкціонованого доступу до файлів, записів БД або використання обчислювальних ресурсів	висока	Впровадження механізмів обмеження доступу, шифрування даних, моніторинг роботи
З.5	Віртуальні мережі в межах хмарної інфраструктури	Прослуховування трафіку, порушення цілісності, доступності, організація атак DDoS, несанкціоноване підключення до мережі	висока	Використання засобів захисту даних в мережі, систем виявлення та протидії мережевим атакам, моніторинг роботи
З.6	Гіпервізор	Повний контроль над розгорнутим віртуальним середовищем	низька	Захист гіпервізора та контроль доступу до його налаштувань
З.7	Фізичне обладнання	Встановлення засобів несанкціонованого доступу, модифікації та знищення інформації. Порушення доступності ЦОД	низька	Використання систем контролю доступу, політики безпеки, атестація персоналу. Встановлення контрольованої зони
З.8	Допоміжні системи (живлення, охорони, безпеки, охолодження)		низька	
З.9	Управління та моніторинг роботи хмари	Отримання несанкціонованого доступу до налаштувань хмари	середня	Використання технологій розмежування та обмеження доступу, моніторинг дій адміністраторів
З.10	Зв'язки між хмарними ЦОД	Порушення доступності ЦОД, отримання несанкціонованого доступу до інформації, що передається мережею	низька	Використання надійних протоколів з стійкими криптограф. алгоритмами

1.3.4 Методи захисту від загроз в хмарі

При дослідженнях та аналізі загроз рекомендується використовувати метод попереднього аналізу небезпек (PHA). Він дозволяє провести аналіз на попередніх стадіях проектування, з недостатньою кількістю інформації, зробити ранжування небезпек та ризику.

Запропоновані на основі аналізу сучасного стану стандартизації та застосування хмарних сервісів моделі хмарних обчислень, порушника та загроз хмарних сервісів дозволили встановити, що найбільш проблемними та такими, що вимагають вирішення в частині надання послуг конфіденційності, цілісності, справжності та доступності тощо, є задачі захисту ключів та ключової інформації. Для цього на основі аналізу стану встановлено, що в середовищі хмари відносно ключових даних існують та можуть бути реалізованими такі загрози як компрометація, несанкціоноване знищення, перехоплення та

запам’ятовування, нав’язування слабких та несанкціоноване використання тощо ключів. При цьому встановлено, що найбільшу небезпеку в середовищі хмарних обчислень для ключових даних користувача представляють адміністратори хмарних сервісів, які мають доступ до середовища, в якому розгорнуто хмарні додатки користувача.

Також на основі детального аналізу стану та вимог відносно безпечності управління інформаційною безпекою зі сторони нормативно-правових документі та стандартів. Вони зводяться до використання для забезпечення високого рівня безпеки, тобто високого рівня ймовірностей реалізації загроз в середовищі хмарних обчислень, комплексу технічних, організаційних та організаційно-технічних заходів та засобів, в тому числі до використання:

– на рівні користувача захищених з необхідним рівнем безпеки ключових носіїв;

– на рівні каналів зв’язку між користувачем та хмарою захищених каналів зв’язку з взаємною автентифікацією сторін та стійкістю вищою за стійкість ключів, що передаються;

– на рівні сервісів ідентифікації, автентифікації, авторизації та керування правами доступом надійних протоколів автентифікації з стійкими криптографічними алгоритмами.

Відштовхуючись від вищезазначених пунктів визначимо, що найоптимальнішим варіантом буде використання IAC (інраструктура як код) інструментів. Подібні інструменти використовують ключі шифрування та дозволяють керувати хмарною інфраструктурою через запити у вигляді коду. Такий підхід мінімізує кількість осіб, які мають адміністративний доступ до хмари та час, за який відбувається побудова та редагування інфраструктури. Зі свого боку, зменшення часу розгортання інфраструктури зменшує імовірність перехоплення даних, порівняно з стандартним розгортанням ресурсів через веб-інтерфейс AWS.

1.4 ЗАГАЛЬНА ХАРАКТЕРИСТИКА AWS.

1.4.1 Характеристика послуг безпеки AWS

AWS забезпечує безліч засобів контролю, щоб захистити робочі навантаження клієнтів, і досить часто клієнти не знають про свою частку відповідальності за безпеку та необхідний контроль за безпекою користування та розміщення своїх ресурсів в хмарі AWS.

AWS надає безліч послуг, інструментів і методів, таких як контроль доступу, брандмауер, шифрування, реєстрація, моніторинг, відповідність тощо, щоб забезпечити безпеку у хмарі. Ці служби AWS підтримують безліч випадків та сценаріїв, щоб дотриматись усіх вимог щодо безпеки, реєстрації користувачів, аудиту та дотримання вимог у хмарному середовищі. Існує послуга AWS Identity and Access Management (IAM), яка дозволяє вам контролювати безпечний доступ та дії для ваших користувачів AWS, Virtual Private Cloud (VPC) дозволяє захистити свою інфраструктуру в хмарі AWS шляхом створення віртуальної мережі до вашої приватної мережі у локальному центрі обробки даних. Більше того, існують веб-сервіси, такі як Служби управління ключами (KMS), що використовуються для управління ключами шифрування для додатуового захисту ваших даних. Існує AWS Shield та веб-брандмауер веб-додатків AWS (WAF) для захисту ваших ресурсів AWS та програм від загальних загроз безпеці, таких як розподілена відмова в обслуговуванні (DDoS) шляхом налаштування брандмауерів на різних рівнях. AWS Config разом із AWS CloudTrail та AWS CloudWatch підтримує аудит та управління конфігурацією для всіх ваших ресурсів AWS. Артефакт AWS - це керований сервіс самообслуговування, який надає вам документи про відповідність на вимогу вашого аудитора. [15, с. 22].

AWS та хмара загалом значно змінилися з того часу, коли безпека в хмарі розглядалась як перешкода для переміщення ваших даних, програм та робочого навантаження в хмару сьогодні, коли безпека в хмарі є однією з основних причин, з яких організації переносять центри обробки даних у хмару. Все більше і більше керівників та відповідальних осіб помічають, що безпека в хмарі випереджає час, а також надійніша та економічніша, ніж безпека в локальних

центрах обробки даних. Ці дані зібрані з кількох географічних регіонів та різних галузей з жорсткими вимогами безпеки та відповідності нормативним актам, таким як Міністерство оборони, банківської справи, охорони здоров'я, тощо, і належать до всіх рівнів, таких як CIO, CTO, CEO, CISO, системні адміністратори, менеджери проектів, розробники, аналітики безпеки тощо.

Як результат, рівень впровадження хмарних технологій швидко зростає протягом останніх кількох років в різних галузях. Ця тенденція очолюється великими підприємствами, де безпека відіграє ключову роль у вирішенні питання про те, чи слід переходити підприємству до хмари чи ні. AWS забезпечує повністю інтегровані та уніфіковані рішення безпеки для своїх хмарних служб, що дозволяє їх клієнтам перенести свої робочі навантаження в хмару. Розглянемо деякі прогнози щодо експоненціальне зростання хмарних обчислень лідерами галузі:

- Gartner зазначає, що до 2020 року корпоративна політика без хмар буде такою ж рідкістю, як підприємство без доступу до Інтернету.
- Global Cloud Index (GCI) прогнозує, що хмара становитиме 92% даних до 2021 року, тобто 92% усіх даних та обчислювальних ресурсів будуть знаходитись в хмарі до 2021року.
- International Data Corporation (IDC) заявляє, що сьогодні найперспективнішою для розвитку є хмара.

AWS створено як одне з найбільш гнучких та безпечних хмарних середовищ. Це прибирає більшість навантажень на безпеку, які традиційно пов'язані з IT-інфраструктурою. AWS забезпечує повну конфіденційність клієнтів та має безліч вбудованих функції безпеки. Більше того, кожен клієнт виграє від процесів безпеки, глобальна інфраструктура та архітектура мережі, запроваджені AWS для дотримання суворих вимог безпеки. [16, с. 58].

У міру того, як все більше і більше організацій рухаються до хмар, безпека в хмарі стає все більш доступною для багатьох. Незважаючи на те, що здебільшого це безпека, яка забезпечується у хмарі, вона має більшість тих самих

функціональних можливостей, що і безпека в традиційних IT інфраструктурах, таких як захист інформація від крадіжки, витоку даних та їх видалення.

1.3.2 Модель спільної відповідальності

Безпека в хмарі дещо відрізняється від безпеки в локальних даних центру. Коли ви переміщуєте сервери, дані та робоче навантаження до хмари AWS, обов'язки захисту ваших даних та робочого навантаження покладаються на вас та AWS. AWS відповідає за захист базової інфраструктури, яка підтримує хмару, через глобальну мережу регіони, зони доступності, розташування ресурсів, кінцеві точки тощо, а клієнт несе відповідальність за все, що він розміщує у хмарі, наприклад, дані, застосунки або все, що підключається до хмари, наприклад, сервери в їх центрах обробки даних. Клієнт також відповідає за забезпечення доступу до своєї віртуальної мережі та ресурсів у хмарі, це модель відома як модель спільної відповідальності за безпеку AWS. [17, с. 39].

На рисунку 1 показана ця модель:

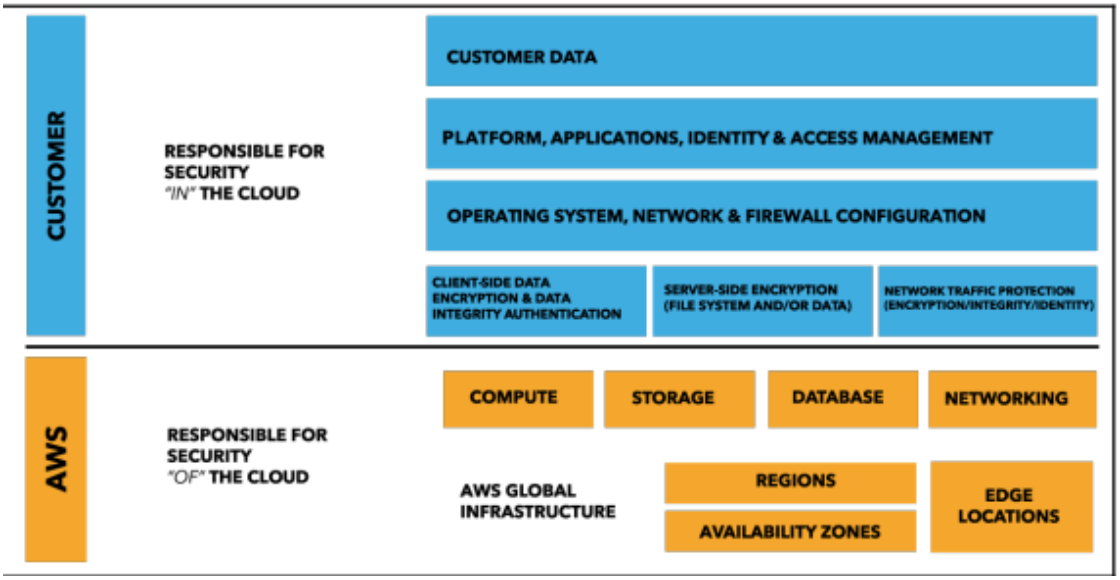


Рис 1.3 Модель спільної відповідальності за безпеку AWS.

Для того, щоб освоїти безпеку AWS, нам важливо визначити та зрозуміти що таке часткова відповідальність AWS за безпеку, а також знати нашу частку відповідальності за безпеку. AWS пропонує безліч різноманітних послуг, які можна поділити на три широкі категорії: інфраструктура, контейнерні та абстрактні послуги. Кожен з ці категорії мають власну модель власності на

безпеку, засновану на взаємодії кінцевих користувачів з ним і як доступ до функціоналу:

- Інфраструктурні послуги: Ця категорія включає обчислювальні послуги, такі як Amazon Elastic Cloud Compute (EC2) та супутні послуги, такі як Amazon Elastic Block Store (EBS), Elastic Load Balancing та Amazon Virtual Private Cloud (VPC). Ці послуги дозволяють розробляти та будувати власні захищені та приватна мережі в хмарі з інфраструктурою, подібною до локальних рішень. Ця мережа в хмарі AWS також сумісна і може бути інтегрована з локальною мережею. Ви керуєте операційною системою, налаштовуєте правила брандмауера та функціонування будь-якої системи управління ідентифікацією, яка забезпечує доступ на рівень користувача стека віртуалізації.
- Контейнерні послуги: Є певні служби AWS, які працюють на відокремлених серверах. AWS пропонує ці послуги в керованих контейнерах для. Ви відповідаєте за налаштування правила брандмауера, що дозволяють отримати доступ до ваших користувачів та систем для використання цих служб AWS Identity and Access Management (IAM), серед іншого. Ці послуги включають AWS Elastic Beanstalk, Amazon Elastic Map Reduce (EMR) та Служби реляційних баз даних Amazon (RDS).
- Абстраговані послуги: це служби AWS, які абстрагують платформу або рівень управління. Це обмін повідомленнями, електронна пошта, база даних NoSQL та сховище служби, на яких ви можете створювати та експлуатувати хмарні програми. Ці послуги доступ до них здійснюється через кінцеві точки за допомогою API AWS. AWS керує базовими компоненти служби або операційної системи, в якій вони перебувають. Ви отримуєте базову інфраструктуру, яку AWS забезпечує для цих абстрагованих послуг. Ці послуги забезпечують платформу для кількох орендарів, яка ізолює ваші дані від інших

користувачів. Ці послуги інтегровані з AWS IAM для безпечного доступу та використання. Ці послуги включають службу простої черги, Amazon DynamoDB, SNS, Amazon Simple Storage Service (S3) тощо.

1.3.3 Модель спільної відповідальності за послуги інфраструктури

Глобальна інфраструктура AWS надає послуги інфраструктури, такі як Amazon EC2, Amazon VPC та Amazon Elastic Block Storage (EBS). Це регіональні служби; вони діють в межах регіону, де їх було запущено. У них різні рівні довговічності та доступності. AWS забезпечує декілька варіантів використання різних еластичних компонентів у декількох зонах доступності всередині певного регіону. [18, с. 20].

На наступному малюнку показана ця модель:

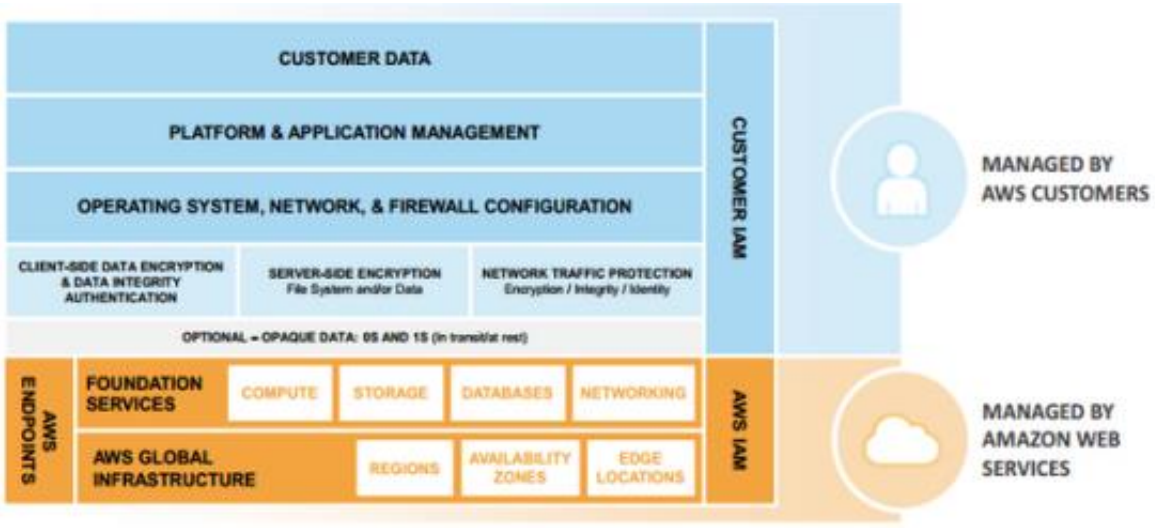


Рис 1.4 Модель спільної відповідальності за послуги інфраструктури.

Спираючись на безпечну глобальну інфраструктуру AWS, подібну до ваших локальних центрів обробки даних, Ви будете встановлювати, налаштовувати та керувати своїми операційними системами та платформами в AWS. Отримавши свою платформу, ви будете використовувати її для встановлення своїх додатків, а потім ви будете зберігати свої дані на цій платформі. Ви налаштуєте безпеку своїх даних як шифрування при передачі та в стані спокою. Якщо для вашого бізнесу потрібно більше шарів захисту через

відповідність нормативним вимогам, ви завжди можете додати один із запропонованих рівнів глобальної інфраструктури AWS.

Ці рівні захисту можуть включати захист даних у стані спокою за допомогою шифрування або захист даних під час передачі або шляхом введення додаткового рівня непрозорості між службами AWS і вашою платформою. Цей рівень може включати безпечну мітку часу, безпеку облікових дані, шифрування даних, програмне забезпечення та передача цифрового підпису у ваших запитах API і так далі. [19, с.54].

Коли ви запускаєте новий екземпляр Amazon Elastic Cloud Compute (EC2) зі стандарту Amazon Machine Image (AMI), ви можете отримати до нього доступ за допомогою безпечного віддаленого доступу до системи такими протоколами, як захищена оболонка (SSH) для екземпляра Linux або віддаленого робочого столу Windows Протокол (RDP) для екземпляра Windows. Щоб налаштувати ваш екземпляр EC2 відповідно до вашиг вимог і щоб отримати до нього доступ, вам потрібно пройти автентифікацію на рівні операційної системи. Після автентифікації на рівні операційної системи ви отримаєте безпечний віддалений доступ до екземпляра Amazon EC2. Потім ви можете налаштувати кілька методів аутентифікації операційні системи, такі як Microsoft Active Directory, автентифікація сертифіката X.509 або облікові записи локальної операційної системи.

AWS пропонує пари ключів Amazon EC2, які складаються з двох різних ключів, відкритого та закритого. Ці пари ключів RSA є галузевим стандартом і використовуються для автентифікації отримати доступ до вашого екземпляра EC2. Коли ви запускаєте новий екземпляр EC2, ви отримуєте варіант будь-якого створити нову пару ключів або використовувати існуючу пару ключів. Існує також третій варіант діяти без пари ключів, але це не рекомендується для забезпечення доступу до вашого EC2 екземпляр. На наступному малюнку 3 показана опція пар ключів EC2 під час запуску EC2 екземпляр. Ви можете створити до 5000 пар ключів для своїх екземплярів EC2 у вашому AWS рахунок. Пари ключів EC2 використовуються лише для доступу до ваших екземплярів

EC2 і не можуть бути використані для входу в консоль управління AWS або для використання інших служб AWS. Більше того, користувачі можуть використовувати різні пари ключів для доступу до різних екземплярів EC2:

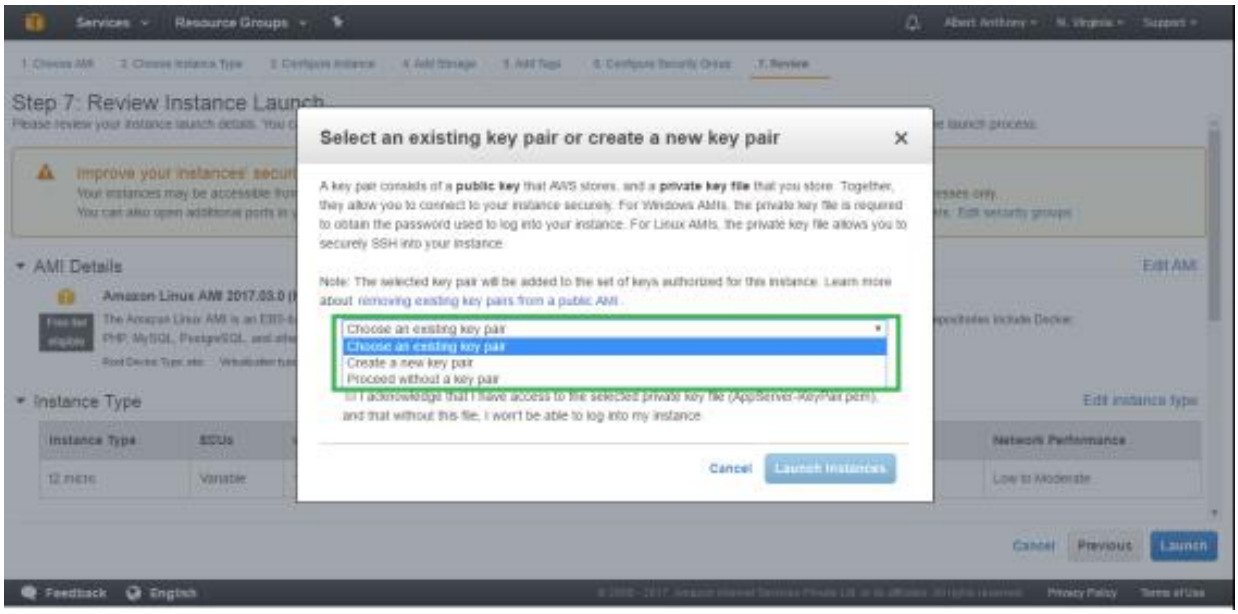


Рис 1.5 Додавання пари ключів при запуску сервера.

Ви можете доручити AWS згенерувати пари ключів EC2 для вас, або ви можете створити свою власну пару ключів Amazon EC2 з використанням стандартних інструментів, таких як OpenSSL. Коли ви вибираєте перший варіант, AWS надає вам як відкритий, так і приватний ключі RSA при запуску екземпляра. Вам потрібно надійно зберігати закритий ключ; якщо він загубився, його неможливо відновити з AWS, і тоді вам доведеться створити нову пару ключів.

Коли ви запускаєте новий екземпляр Linux EC2 за допомогою стандартного AWS AMI, відкритого ключа пара ключів Amazon EC2, яка зберігається в AWS, ключ додається до операційної системи. Для підключення до цього можна використовувати клієнт SSH, який повинен бути налаштований на використання імені користувача EC2, такого як ec2-user та за допомогою закритого ключа для авторизації користувача.

Під час запуску нового екземпляра Windows EC2 за допомогою служби ec2config з стандартного AWS AMI, послуга ec2config встановлює новий випадковий пароль адміністратора для цей примірник Windows і шифрує його,

використовуючи відповідну пару ключів Amazon EC2 відкритий ключ. Ви будете використовувати приватний ключ для розшифровки пароля адміністратора за замовчуванням. Цей пароль буде використовуватися для автентифікації користувачів на екземплярі Windows [20, с. 83].

Хоча AWS пропонує безліч гнучких і практичних інструментів для управління Amazon EC2 ключі та автентифікацію для доступу до екземплярів EC2, якщо вам потрібна вища безпека через вимоги вашого бізнесу або відповідність нормативним актам, ви завжди можете запровадити інші механізми автентифікації, такі як легкий протокол доступу до каталогу (LDAP) та вимкнути автентифікацію пари ключів Amazon EC2.

1.3.4 Модель спільної відповідальності для контейнерних послуг

Модель спільної відповідальності AWS застосовується і до контейнерних служб, таких як Amazon EMR та Amazon RDS. Для цих служб AWS керує операційною системою, базовою інфраструктурою, платформою програм та основними послугами. Наприклад, Amazon RDS для сервера Microsoft SQL - це керована служба баз даних, де AWS управляє усіма шари контейнера, включаючи платформу баз даних сервера Microsoft SQL. Хоча платформа AWS забезпечує інструменти резервного копіювання та відновлення даних для таких служб, як Amazon RDS, це ваша відповідальність за планування, налаштування та використання інструментів для високої доступності (HA), стійкість до несправностей (FT), безперервності бізнесу та аварійного відновлення (BCDR). [21, с. 42].

Ви несете відповідальність за захист своїх даних, за надання доступу до ваших даних та для налаштування правил брандмауера для доступу до цих служб контейнерів. Приклади правил брандмауера включають групи безпеки RDS для Amazon RDS та групи безпеки EC2 для Amazon EMR.

На наступному малюнку показана модель для контейнерних служб:

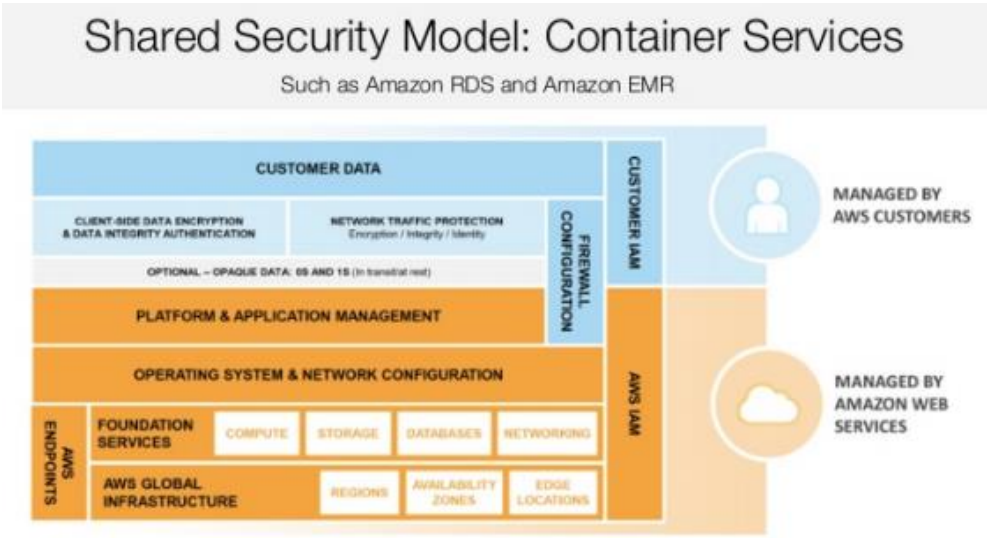


Рис 1.6. Модель спільної відповідальності для контейнерних послуг.

1.3.5 Модель спільної відповідальності для абстрагованих послуг

AWS пропонує абстраговані послуги, такі як Amazon DynamoDB та Amazon Simple Queue Служби, Amazon S3 тощо, де ви можете отримати доступ до кінцевих точок цих служб для зберігання, модифікація та отримання даних. AWS відповідає за управління цими послугами, тобто управління інфраструктурним рівнем, встановлення та оновлення операційної системи та управління платформами. Ці послуги тісно інтегровані з IAM, щоб ви могли вирішити, хто може отримати доступ до ваших даних, що зберігаються в цих послугах.

Ви також несете відповідальність за класифікацію ваших даних та використання спеціальних інструментів для налаштування дозволів на рівні платформи для окремих ресурсів. Використовуючи IAM, ви також може налаштовувати дозволи на основі ролі, ідентифікації користувача або груп користувачів. Amazon S3 надає вам шифрування даних у стані спокою на рівні платформи, а для даних, що передаються – це забезпечує інкапсуляцію HTTPS шляхом. [22, с. 56].

На наступному малюнку показана ця модель для абстрагованих послуг:

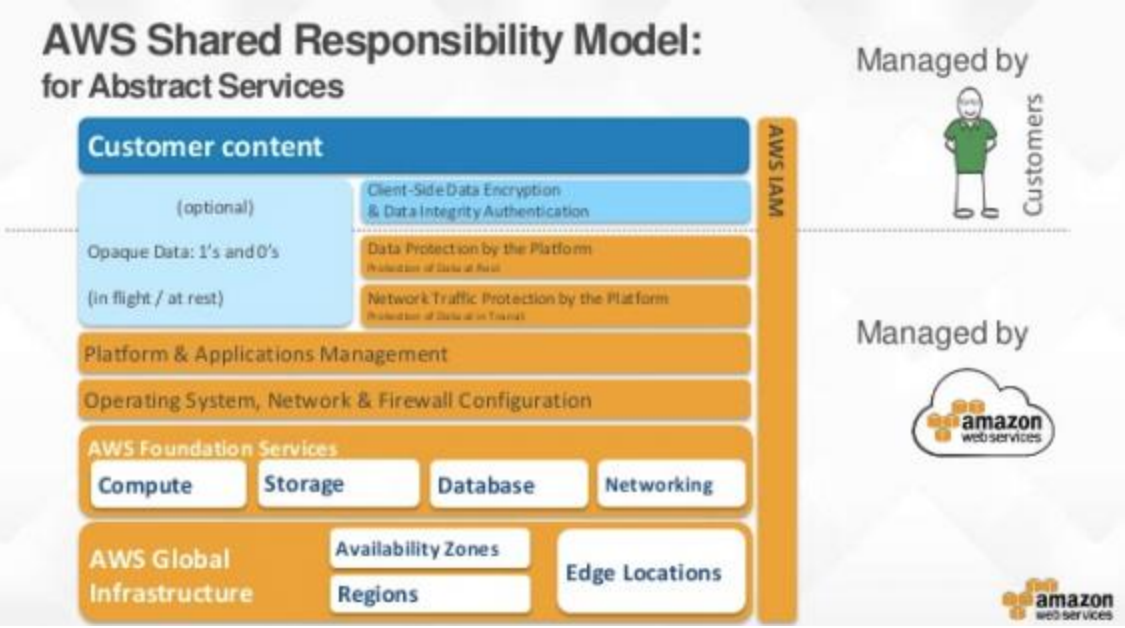


Рис 1.7 Модель спільної відповідальності для абстрагованих послуг.

1.3.6 Обов'язки AWS щодо безпеки

AWS відповідає за забезпечення глобальної інфраструктури, що включає регіони та зони доступносі, що працюють на хмарі AWS. Ці зони доступності містять кілька центрів обробки даних, в яких розміщується обладнання, програмне забезпечення, мережі та інші ресурси, на яких розгорнуті послуги AWS. Забезпечення цієї інфраструктури є першочерговим завданням AWS, і AWS регулярно перевіряється відомими агенціями по всьому світу для забезпечення необхідної безпеки та відповідності стандартам. Ці звіти про аудит доступні для клієнтів AWS, оскільки клієнти не можуть особисто відвідувати центри обробки даних AWS.

Дані клієнтів та робочі навантаження зберігаються в центрах обробки даних AWS, ці центри обробки даних поширюються в географічних регіонах по всьому світу. Ці центри обробки даних належать, експлуатуються та контролюється AWS. Цей контроль включає фізичний доступ та вхід до цих центрів обробки даних і всі мережеві компоненти та обладнання, а також усі інші додаткові центри обробки даних, які є частиною глобальної інфраструктури AWS.

Отже, найперша думка, яка вразить кожного, хто роздумує про перенесення свого навантаження хмара, де фактично зберігаються мої дані? Де

фізичні сервери та жорсткі диски знаходиться, що я створив за допомогою хмари AWS? А як ці апаратні ресурси і хто їх забезпечує? Адже хмара просто віртуалізує всі ресурси, доступні в центрах обробки даних, але ці ресурси є десь фізично. Отже, AWS повністю відповідає за фізичну та логічну безпеку всього обладнання та ресурси, розташовані в центрах обробки даних по всьому світу.

AWS має багаторічний досвід створення, управління та забезпечення великих центрів обробки даних по всьому світу через материнську компанію Amazon. AWS гарантує, що всі його центри обробки даних використовують найкращі технологій та процеси, такі як розміщення їх у непомітному вигляді, дотримання політики найменших привілеїв, відеоспостереження, двофакторна автентифікація для введення дата-центрів та поверхів.

Персонал забороняється розміщувати на поверхах ЦОД, якщо у них немає потреби отримати доступ до фізичних пристроїв зберігання даних особисто. Більше того, AWS твердо впроваджує сегрегацію відповідальності, тому будь-який персонал, який має доступ до фізичного пристрою, не матиме доступ кореневого користувача для цього пристрою, тому він не може отримати доступ до даних на цьому фізичному пристрої.

Це дуже важлива частина моделі спільної відповідальності за безпеку, де AWS робить все щоб турбуватися про фізичну та логічну безпеку центрів обробки даних. Вам не доведеться турбуватися про моніторинг, крадіжки, вторгнення, пожежу, природні лиха, збій живлення тощо для ваших центрів обробки даних. Цими речами опікуються AWS від вашого імені, і вони постійно вдосконалюють свої процедури безпеки, щоб не відставати від зростаючих загроз.

AWS розпочне процес виведення з експлуатації, коли запам'ятовуючий пристрій досягне кінця терміну його корисного використання. Цей процес гарантує, що дані клієнтів не піддаються несанкціонованому доступу інших фізичних осіб. Цей апаратний пристрій буде фізично зруйнований або знешкоджений, якщо він вийде з ладу.

AWS зберігає ваші дані та інші ресурси в центрах обробки даних у різних географічних регіонах по всій земній кулі; ці місця відомі як регіони. У кожному регіоні є дві або більше зон доступності для високої стійкості до несправностей. Ці зони доступності складається з одного або декількох центрів обробки даних. Всі ці центри обробки даних використовуються, і жоден не зберігається офлайн; тобто немає холодних центрів обробки даних. Ці центри обробки даних містять усе фізичні так апаратні ресурси, як сервери, сховища та мережеві пристрої тощо необхідні для підтримання роботи всіх служб AWS відповідно до угоди про рівень обслуговування. Усі основні програми AWS, такі як обчислення, зберігання, бази даних, мережі розгортаються у конфігурації N + 1, так що, у випадку пошкодження центру обробки даних внаслідок стихійного лиха, людської помилки чи будь-якої іншої непередбаченої обставини є достатня потужність для балансування навантаження на решту сайтів.

Кожна зона доступності розроблена як незалежна зона відмов, щоб врахувати можливі збої в роботі. Вони фізично відокремлені в межах географічного розташування і розташовані в верхній частині рівнин.

Залежно від характеру вашого бізнесу, дотримання нормативних вимог, аварійне відновлення, відмовостійкість тощо, ви можете вирішити розробити свої програми, які будуть розподілені по кількох регіонах, щоб вони були доступними, навіть якщо один регіон недоступний.

На наступному малюнку показані типові регіони з їх зонами доступності:

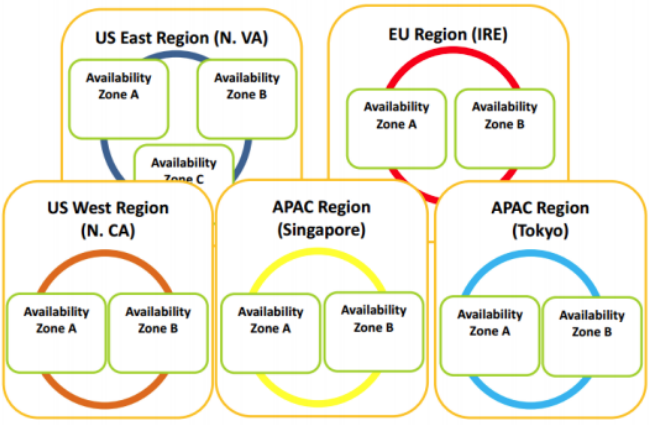


Рис 1.8 Приклад структури регіонів AWS

1.3.7 Служби безпеки AWS

Тепер давайте розглянемо служби безпеки AWS. Це послуги AWS, які в першу чергу надають способи захисту ресурсів в AWS.

Управління ідентифікацією та доступом AWS

AWS IAM дозволяє клієнтам безпечно контролювати доступ до своїх ресурсів AWS та AWS користувачів. У двох словах, IAM забезпечує автентифікацію та авторизацію для доступу до ресурсів AWS. Він підтримує доступ до ресурсів AWS через веб-консоль управління, CLI, або програмно через API та SDK. Він має основні функції для контролю доступу як користувачі, групи, ролі та дозволи, а також розширені функції, такі як ідентифікація для інтеграції із наявною базою даних користувача, якаою може бути Microsoft Active Directory або Facebook, або Google. Ви можете визначити детальні дозволи для усіх ваших ресурсів, а також використовувати тимчасові дані безпеки для надання доступу до зовнішніх користувачів поза вашим обліковим записом AWS.

Віртуальна приватна хмара AWS

AWS VPC - це IaaS, який дозволяє створювати власну VPN у хмарі. Ти можеш надайте свої ресурси в цій логічно ізольованій мережі в AWS. Ця мережа може бути налаштований для безпечного підключення до локального центру обробки даних. Ви можете налаштувати брандмауери для всіх ваших ресурсів у VPC на рівні екземпляру та / або на рівні підмережі для управління трафіком передачі та виходу із вашої VPC. VPC має функцію журналу потоків VPC, яка дозволяє збирати інформація щодо IP-трафіку вашої VPC.

Система управління ключами AWS (KMS)

AWS KMS - це послуга, яка допомагає керувати ключами, що використовуються для шифрування. Є кілька варіанти KMS, які включають принесення власних ключів та управління KMS поряд із тими, що генеруються AWS. Це повністю керована послуга, яка інтегрується з іншими служби AWS, такими як AWS CloudTrail, щоб реєструвати всі дії для ваших служб KMS. Це

Послуга відіграє важливу роль у захисті даних, що зберігаються вашими програмами.

Щит AWS

AWS shield захищає ваші веб-програми, що працюють на AWS, від керованої розподіленої атаки відмови в обслуговуванні (DDoS). Це повністю керована послуга і має два варіанти, стандартний і вдосконалений. Стандарт щита AWS пропонується всім клієнтам безкоштовно та забезпечує захист від найпоширеніших атак, націлених на ваші програми або веб-сайти AWS. Розширений захист AWS забезпечує вищий рівень захисту та інтеграцію з іншими такі послуги, як брандмауери веб-додатків та доступ до команди відповідей AWS DDoS.

Брандмауер веб-додатків AWS (WAF)

AWS WAF - це настроюваний брандмауер для ваших веб-додатків, що дозволяє фільтрувати трафік які ви хочете отримати для своїх веб-програм. Це керована послуга, яка може бути налаштовано з консолі керування або через AWS WAF API, щоб ви могли мати контрольні пункти безпеки на різних рівнях у вашій програмі різними акторами, такими як розробник, інженер DevOps, аналітики безпеки тощо.

AWS CloudTrail

Це послуга ведення журналу, яка реєструє всі запити API і входи до вашого облікового запису AWS. Це допомагає з дотриманням вимог, аудитом та управлінням..Цей журнал можна проаналізувати, використовуючи засоби аналізу журналів для відстеження історії події. Ця послуга відіграє дуже важливу роль в автоматизації та безпеці.

AWS CloudWatch

Це служба моніторингу, яка надає показники, сигнали тривоги та інформаційні панелі для всіх послуг AWS, доступних у вашому обліковому записі. Він інтегрується з іншими службами AWS, такими як AutoScaling, Elastic Load Balancer, AWS SNS, та AWS Lambda для автоматичного реагування для метричного порогу перетину. Він також може збирати та контролювати журнали.

AWS CloudWatch може також використовуватись для збору та моніторингу нестандартних показників для ваших ресурсів або програм AWS.

AWS Config

AWS Config - це послуга, яка дозволяє проводити аудит та оцінювати конфігурацію ваших AWS ресурсів. Ви можете відображати історичну конфігурацію ваших ресурсів AWS, щоб перевірити будь-який інцидент. Це допоможе вам перевірити відповідність, усунути несправності тощо. Ви будете використовувати цю послугу, щоб переконатися, що ваші ресурси AWS залишаються сумісними та налаштованими відповідно до базової конфігурації. Ця послуга забезпечує постійний моніторинг та безперервну оцінка конфігурації ваших ресурсів AWS.

Артефакт AWS

Ця послуга надає вам усі документи, пов’язані із дотриманням вимог. AWS Artificat - це портал на замовлення для самообслуговування, присвячений відповідності та аудиту інформація разом із окремими угодами, такими як додаток до бізнесу та нерозголошення домовленості тощо.

Випробування на проникнення

AWS дозволяє проводити тестування на проникнення для EC2 та реляційної бази даних (RDS); однак спочатку потрібно подати запит до AWS. Якщо AWS схваливши цей запит, ви можете провести тестування на проникнення та сканування вразливості для EC2 та екземпляри RDS у вашому обліковому записі AWS. [5]

1.8 Обов'язки клієнта щодо безпеки

AWS ділиться з клієнтами обов'язками щодо безпеки щодо всіх своїх служб. По суті, Клієнт відповідає за безпеку всього, що він вирішив розмістити в хмарі, наприклад дані, програми, ресурси тощо. Тож захист мережі та захист примірника для Служби IaaS та захист баз даних для контейнерних служб - це сфери, які підпадають під обов'язки клієнта щодо безпеки. Давайте розглянемо відповідальність за безпеку споживачів:

Щодо послуг інфраструктури AWS, замовник відповідає за наступне:

- Дані клієнта
- Застосунки клієнта
- Операційна система
- Конфігурація мережі та брандмауера
- Ідентифікація клієнта та управління доступом
- Управління серверами
- Захист даних (обіг, зберігання та резервне копіювання)
- Забезпечення високої доступності та автоматичного масштабування ресурсів

За контейнерні послуги AWS замовник відповідає за наступне:

- Дані клієнта
- VPC та конфігурація брандмауера
- Керування ідентифікацією клієнта та доступом (користувачі БД і дозволи на таблиці)
- Забезпечення високої доступності
- Захист даних (обіг, зберігання та резервне копіювання)
- Автоматичне масштабування ресурсів

Щодо абстрактних послуг AWS, замовник відповідає за наступне:

- Дані клієнта
- Захист даних у стані спокою за допомогою власного шифрування
- Ідентифікація клієнта та управління доступом

Отже, по суті, коли ми переходимо від інфраструктурних послуг AWS до абстрактних AWS послуг, відповідальність за безпеку клієнтів обмежена конфігурацією та експлуатацією безпекою займається AWS. Більше того, послуги інфраструктури AWS надають вам набагато більше можливостей інтеграції з локальними інструментами безпеки, ніж абстрактні служби AWS.

Усі продукти AWS, які пропонуються як IaaS, такі як Amazon EC2, Amazon S3 та Amazon VPC повністю перебувають під контролем замовника. Ці послуги вимагають від замовника налаштувати параметри безпеки для доступу до цих ресурсів та здійснення управління завдання. Наприклад, для екземплярів EC2

замовник відповідає за управління операційною системою, включаючи оновлення та дотримання безпеки, встановлення та обслуговування будь-якого прикладного програмного забезпечення чи утиліт на екземплярах та групи безпеки (брандмауер на рівні екземпляра, наданий AWS) для кожного екземпляра. Ось такі по суті ті завдання безпеки, які клієнт виконує незалежно від того, де знаходяться його сервери. [5, с.31-40]

Висновки до розділу 1:

Розглянуто нормативно-правову базу України ти світу в сфері забезпечення інформаційною безпеки в хмарі. Розглянуто загрози безпеці в хмарі. Розроблено модель порушника та модель загроз для хмарної інфраструктури. Відштовхуючись від цього визначено, що найбільш безпечним варіантом буде використання IAC (інраструктура як код) інструментів. Подібні інструменти використовують ключі шифрування та дозволяють керувати хмарною інфраструктурою через запити у вигляді коду. Такий підхід мінімізує кількість осіб, які мають адміністративний доступ до хмари та час, за який відбувається побудова та редагування інфраструктури.

Вибрано AWS, оскільки цей постачальник дозволяє автоматизувати завдання забезпечення безпеки, які вирішуються вручну, щоб ви могли сконцентруватися на питаннях розвитку і впровадження інновацій. Перевага AWS для клієнтів полягає в тому, що це єдина хмара з власними пропозиціями щодо сервісів і з перевіреним ланцюжком поставок, яка вважається досить надійною для виконання робочих навантажень високого рівня секретності.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ЗАХИЩЕНОЇ, ВІДМОВОСТІЙКОЇ ТА ВИСОКОДОСТУПНОЇ МЕРЕЖІ

2.1 Реєстрація в AWS та налаштування облікового запису

Для того, щоб приступити до виконання роботи нам необхідно:

Зареєструвати обліковий запис на сайті https://aws.amazon.com/.

Встановити Terraform на робочий комп’ютер.

Приступимо до реєстрації облікового запису. Переходимо на сайт https://aws.amazon.com та натискаємо Create an AWS account. Нас зустрічає вікно вводу реєстраційних даних(рис 2.1).

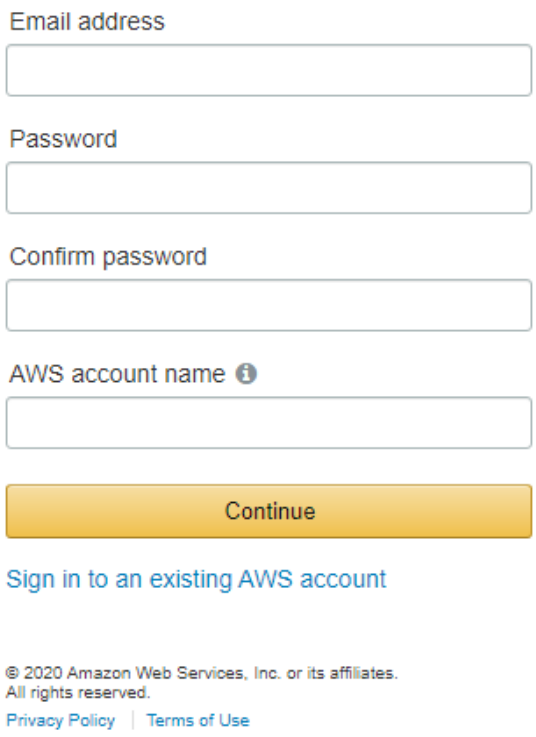


Рис 2.1 Вікно вводу реєстраційних даних

Після вводу необхідної інформації потрапляємо до наступного вікна(рис 2.2) з вводом контактів.

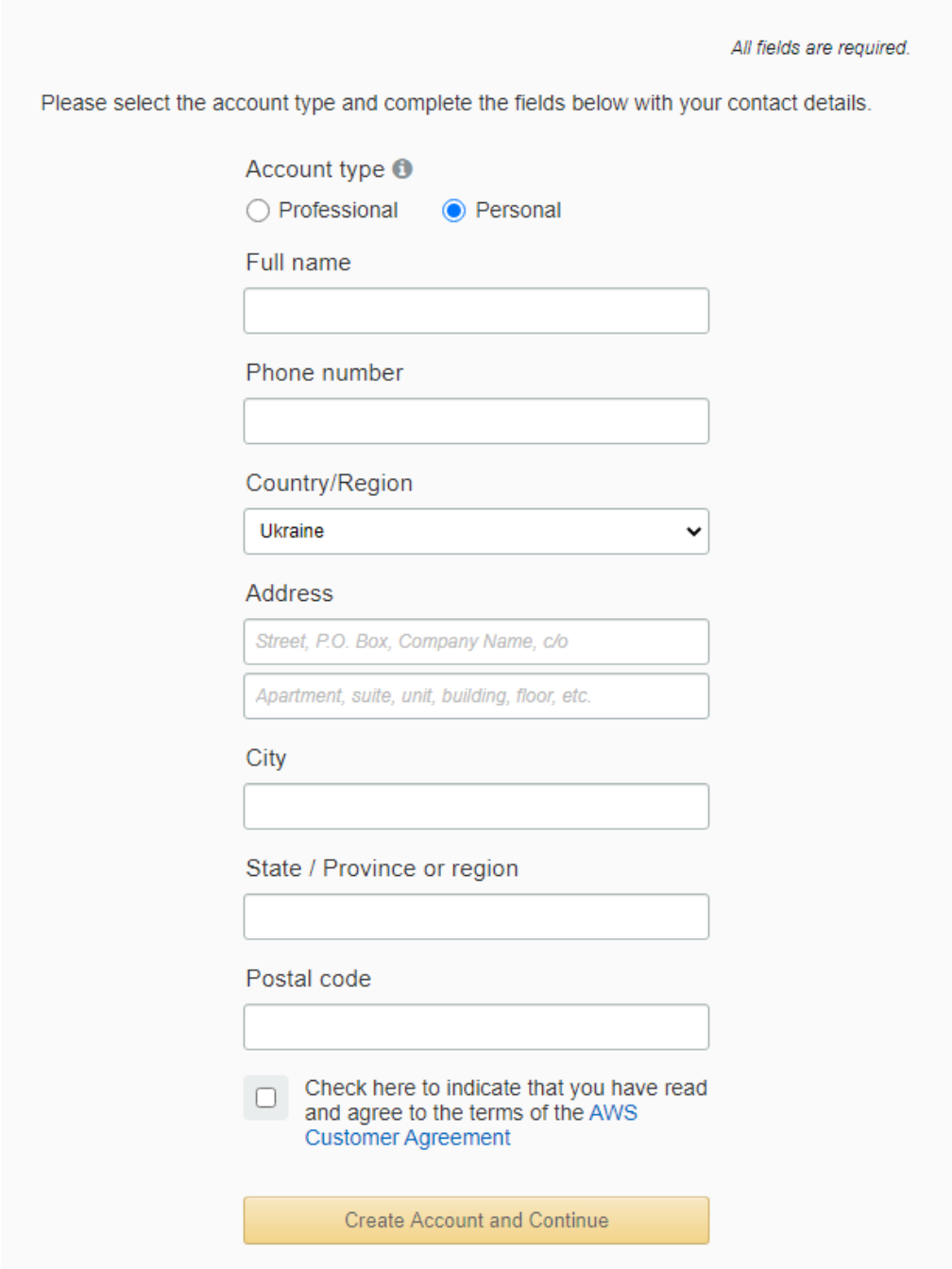


Рис 2.2 вікно вводу контактів

Після успішної реєстрації потрапляємо на головну сторінку Amazon Web Services. Тут необхідно створити користувача для Terraform з адміністративним доступом до VPC та EC2. Для цього переходимо на вкладку Services та знаходимо сервіс IAM. Через інтерфейс IAM відкриваємо розділ Users та натискаємо Add user(рис 2.3). В наступному вікні вводимо ім’я користувача та ставимо галку біля розділу Programmatic access.

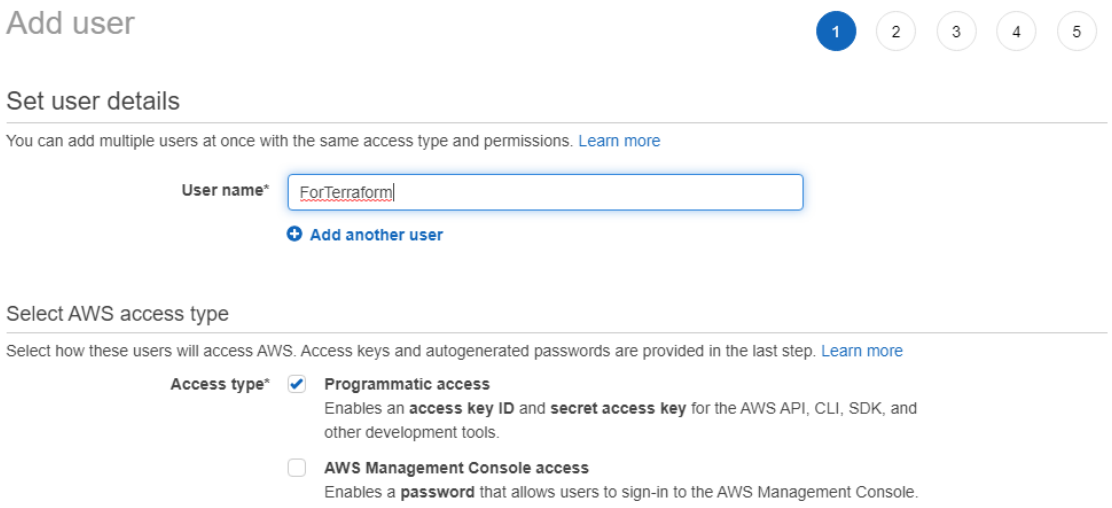


Рис 2.3 Створення користувача через IAM

Наступним кроком додаємо користувачу адміністративні права. Для цього переходимо до Attach existing policies directly(рис 2.4) і знаходимо в списку AmazonEC2FullAccess та AmazonVPCFullAccess. Вибираємо ці пункти та переходимо до наступної сторінки.

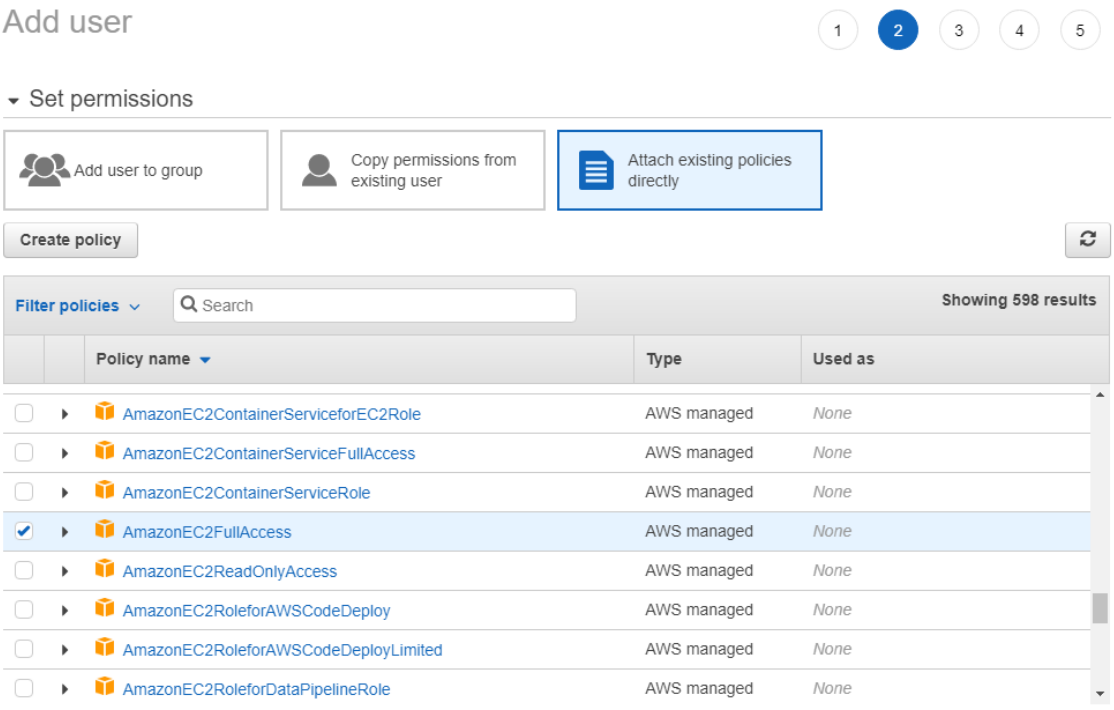


Рис 2.4 Додавання політики доступу

На наступній сторінці додаємо теги для зручності(рис 2.5).

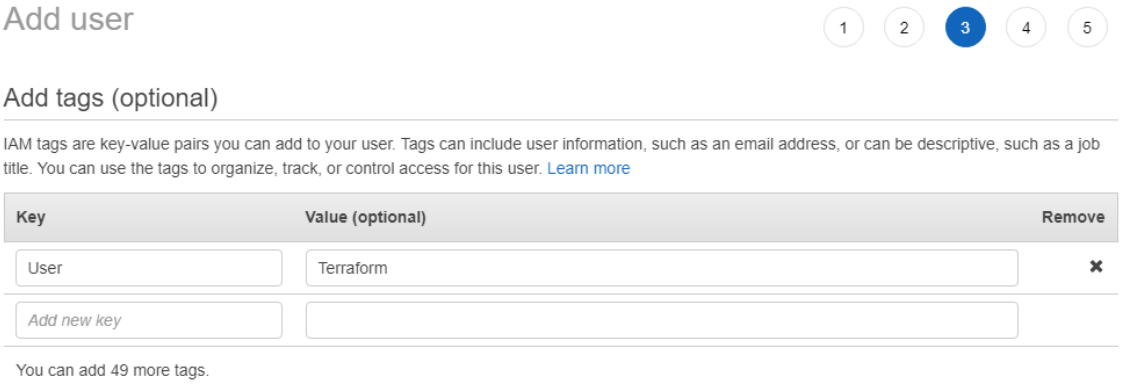


Рис 2.5 Додавання тегів

Наступним кроком AWS дозволяє нам переглянути всі деталі та права нашого користувача(рис 2.6). Перевіряємо інформацію та рухаємось далі.

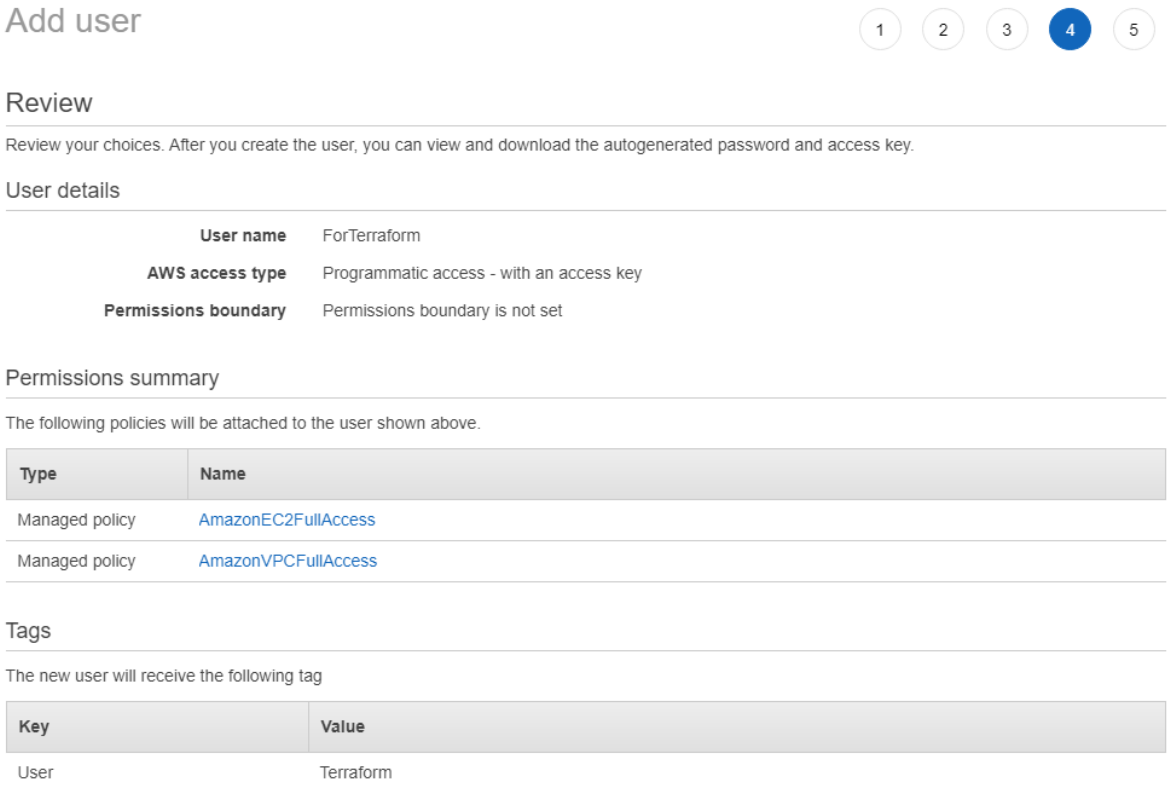


Рис 2.6 Попередній перегляд створеного користувача

Після цього натискаємо Create user та спостерігаємо вікно, яке повідомляє про успішне створення нового користувача. Обов'язково зберігаємо Access key ID та Secret access key(рис 2.7), оскільки вони нам знадобляться в майбутньому для керування нашим обліковим записом.

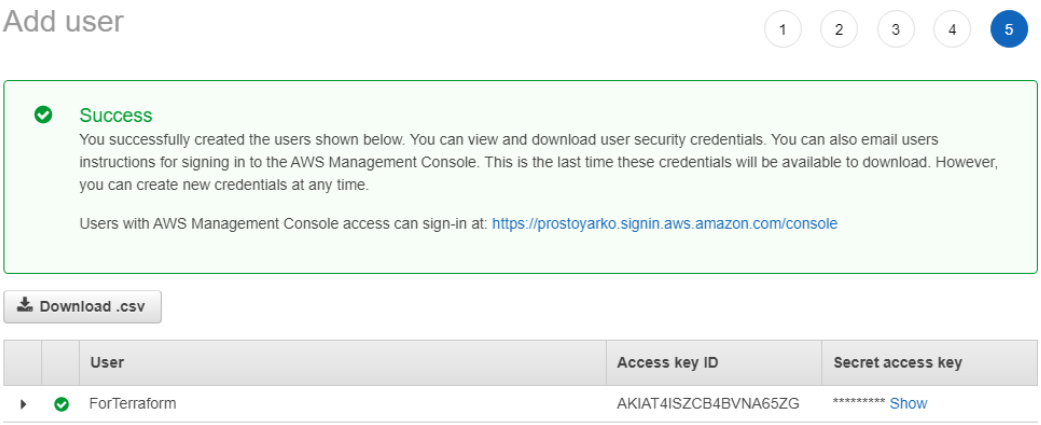


Рис 2.7 Облікові дані користувача

2.2 Налаштування Terraform на робочому комп'ютері

Переходимо на сайт https://www.terraform.io/ та відкриваємо розділ Downloads, вибираємо відповідну операційну систему(рис 2.8) – в нашому випадку це Windows 64-bit та завантажуємо файл.

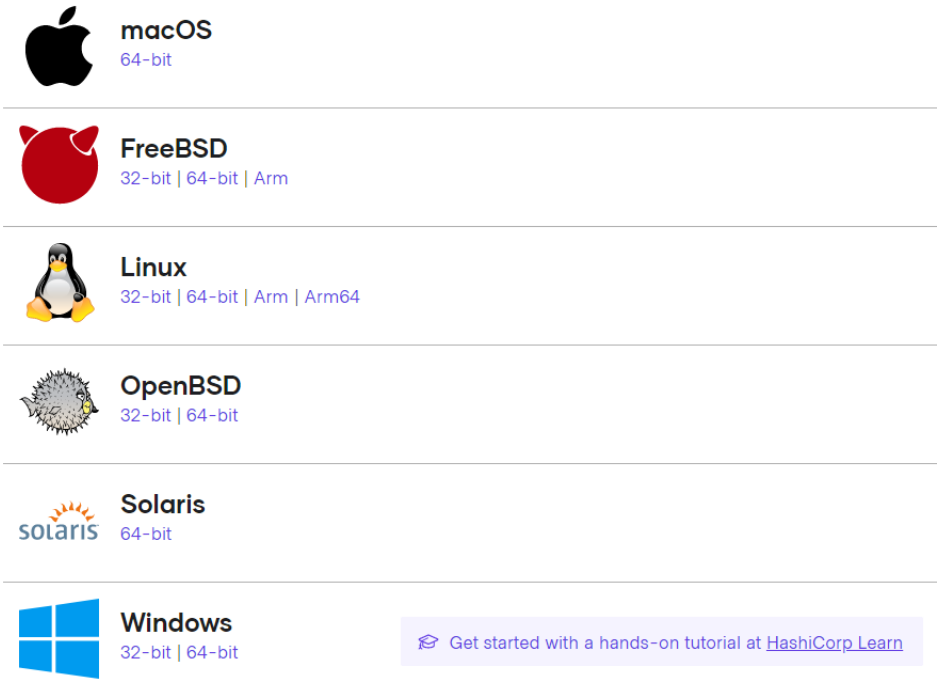


Рис 2.8 Операційні системи, на яких працює Terraform

Створюємо папку з назвою "Terraform" на диску "C" та розпаковуємо завантажений ахрів в неї. Через пошук на робочій станції знаходимо Edit the system environment variables та переходимо в розділ Environment Variables, знаходимо та редагуємо Path та додаємо шлях до папки C:\Terraform(рис 2.9).

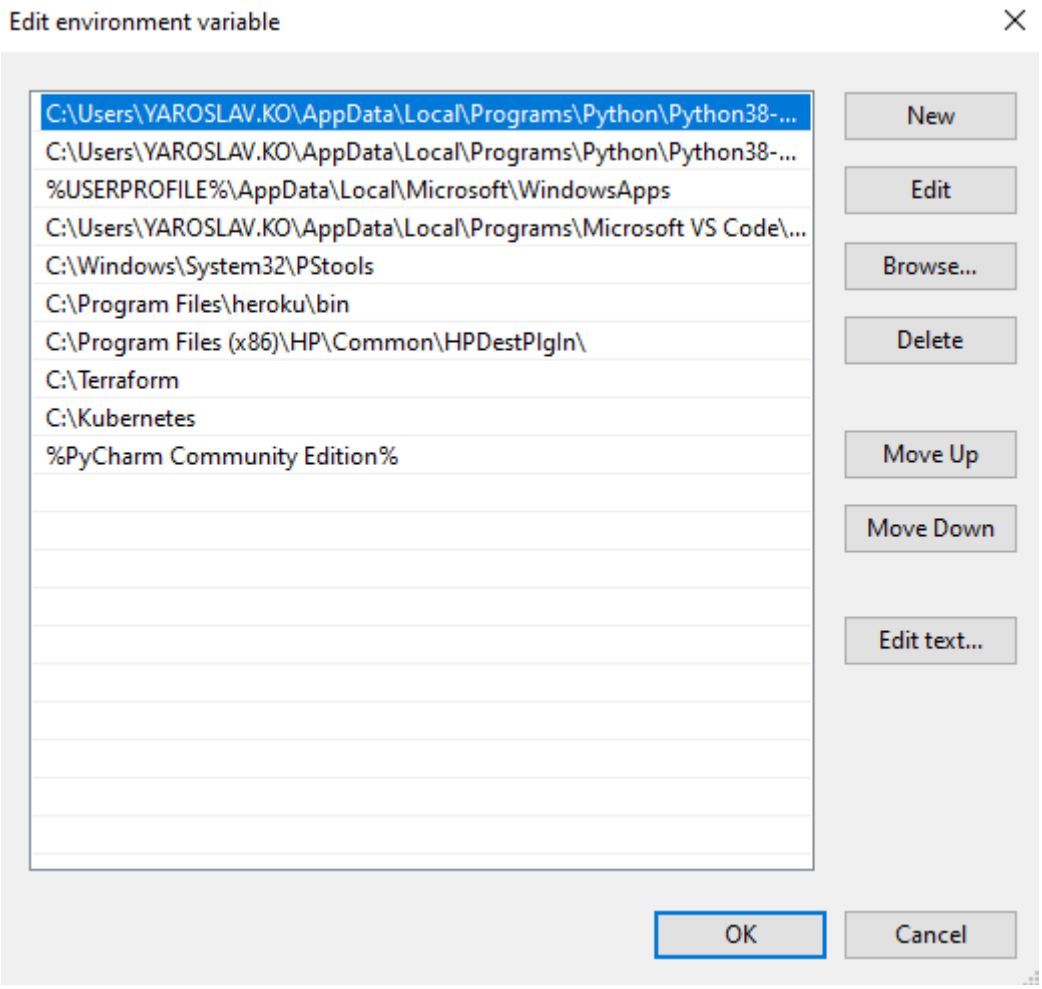


Рис 2.9 Додавання Terraform до system environment variables

Для перевірки правильності встановлення відкриваємо CMD та виконуємо команду(рис 2.10):

terraform --version

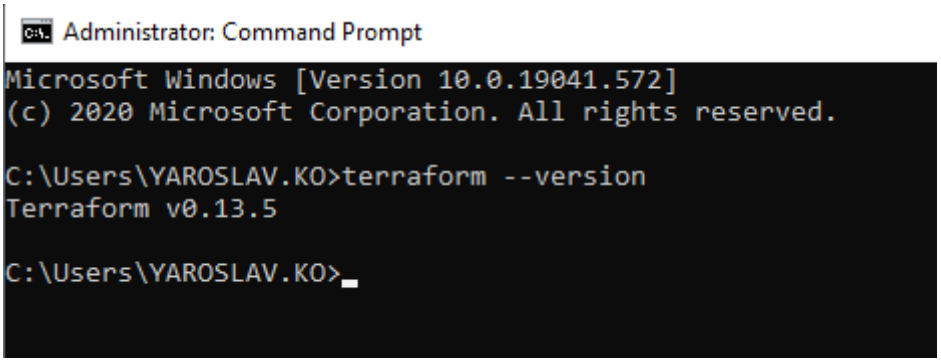


Рис 2.10 Вивід версії Terraform

2.3 Проектування віртуальної мережі

2.3.1 Проектування CIDR блоків VPC та підмереж

Для початку визначимо діапазон IP-адрес для нашої мережі. AWS надає можливість використовувати всі три загальноприйняті діапазони внутрішніх адрес(рис 2.11).

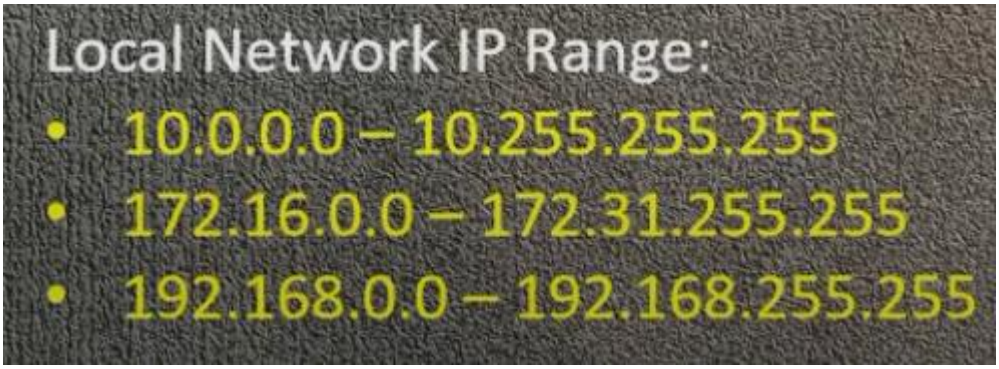


Рис 2.11 Загальноприйняті діапазони внутрішніх адрес

Для проектування віртуальної мережі використаємо CIDR block з маскою підмережі /16. Таке рішення дасть нам 65536 IP-адрес (мінус 5 зарезервованих Амазоном) для майбутніх серверів. AWS резервує їх(рис 2.12) тому, що XXX.XXX.XXX.0 – адреса мережі, XXX.XXX.XXX.255 - широкомовна адреса, XXX.XXX.XXX.1 - XXX.XXX.XXX.3 – можуть бути використані для службових ресурсів.

Рис 2.12 Приклад резервації IP-адрес Амазоном

Використаємо VPC CIDR block 10.0.0.0/16. Такий діапазон дозволить розгорнути достатню кількість внутрішніх підмереж. Попередньо зарезервуємо

6 підмереж з наступними CIDR блоками(рис 2.13).

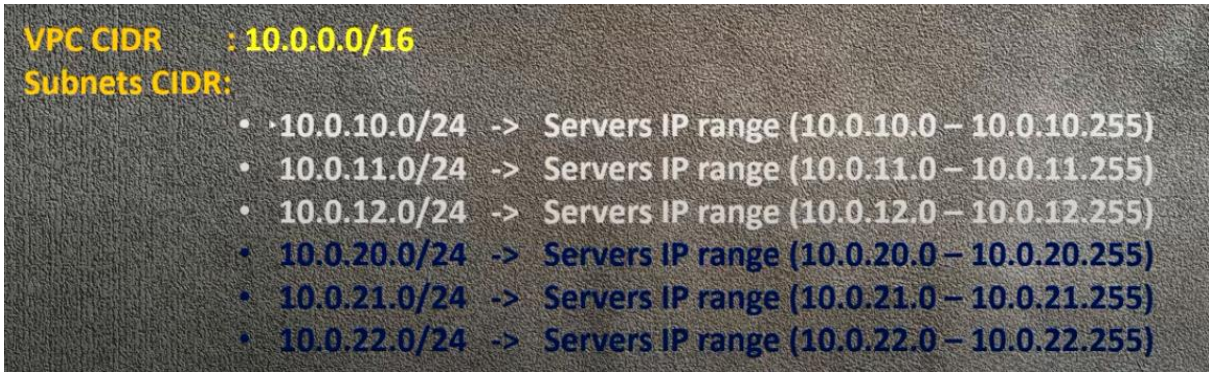
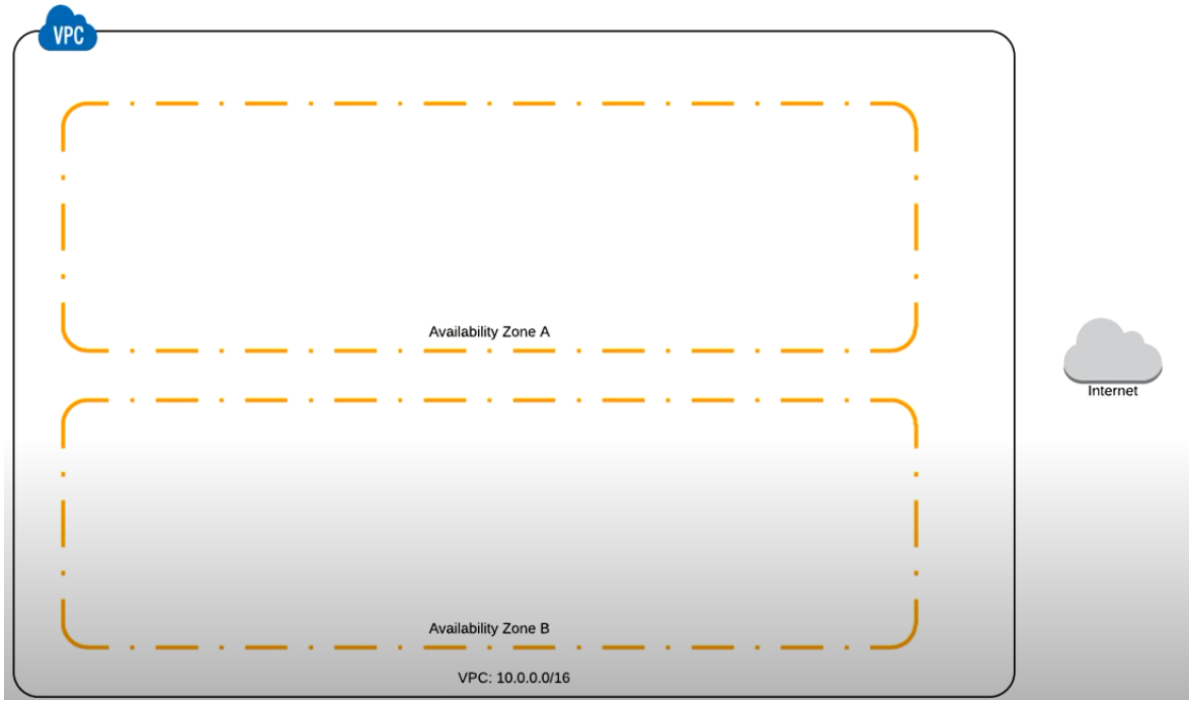


Рис 2.13 Проектування підмереж та CIDR блоків

2.3.2 Проектування підмереж в межах VPC

Створимо віртуальну мережу в регіоні Europe (Frankfurt)eu-central-1(рис 2.14). Такий вибір обумовлений найменшою віддаленістю обчислювального центру AWS від України. Це забезпечить мінімальну латентність – затримку між запитом до серверів та моментом повернення запитуваних даних. Розгорнемо сервери в двох зонах доступності та налаштуємо автоматичне балансування між ними. Такий підхід підвищить відмовостійкість, тобто дозволить системі правильно діяти у випадку виходу з ладу одного з серверів, або ж усієї зони доступності AWS. Також, подібне проектування забезпечує високу доступність ресурсів – безвідмовну та безупинну їх роботу.



2.14 Віртуальна мережа в регіоні Europe (Frankfurt)eu-central-1

За замовчуванням підмережі не мають доступу до інтернету. Для виходу в мережу Інтернет необхідно створити Інтернет шлюз. Також створимо дві публічні підмережі (в різних зонах доступності для підвищення відмовостійкості та високодоступності) та створимо правила маршрутизації, за якими всі пакети з серверів всередині підмереж рухатимуться на шлюз(рис 2.15). Така архітектура дозволь отримувати доступ до ресурсів в цих підмережах з усіх куточків світу.

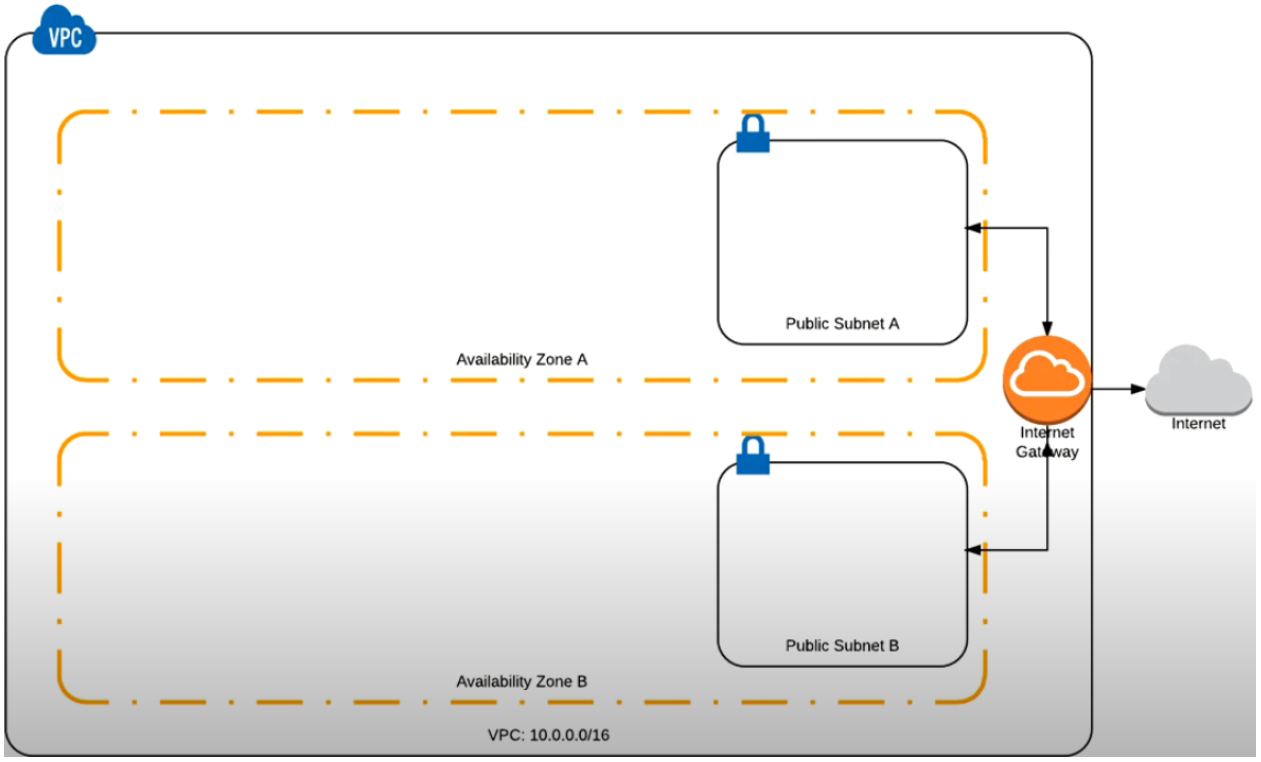


Рис 2.15 Публічні підмережі в VPC

Наступним кроком додаємо дві приватні підмережі та дві підмережі баз даних. За аналогією з публічними підмережами розгорнемо їх в двох зонах доступності. Через свою захищеність та відокремленість вони не матимуть маршрутизації на Інтернет шлюз і, відповідно не будуть доступні ззовні. Але, водночас сервери в цих підмережах можуть бути маршрутизовані на NAT шлюз для доступу до ресурсів мережі Інтернет (рис 2.16).

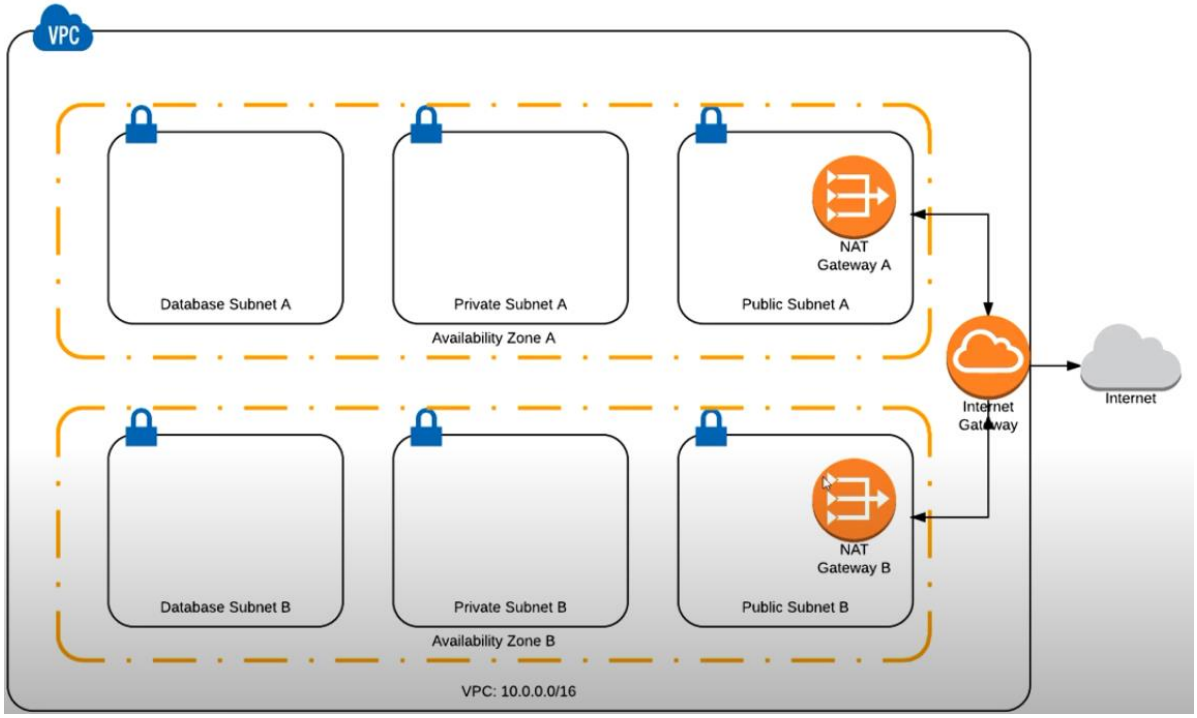


Рис 2.16 Підмережі та NAT шлюзи

Для підвищення відмовостійкості та високодоступності системи створимо вузол-бастіон в межах двох публічних підмереж. Він дасть можливість підключатись до будь-якого з серверів в VPC. Для його коректної роботи необхідно додати Auto Scaling з мінімальною та максимальною кількістю вузлів – 1(Рис 2.17). У такий спосіб, ми завжди матимемо один живий вузол-бастіон.

Таким чином VPC матиме вигляд як на рисунку (рис 2.18).

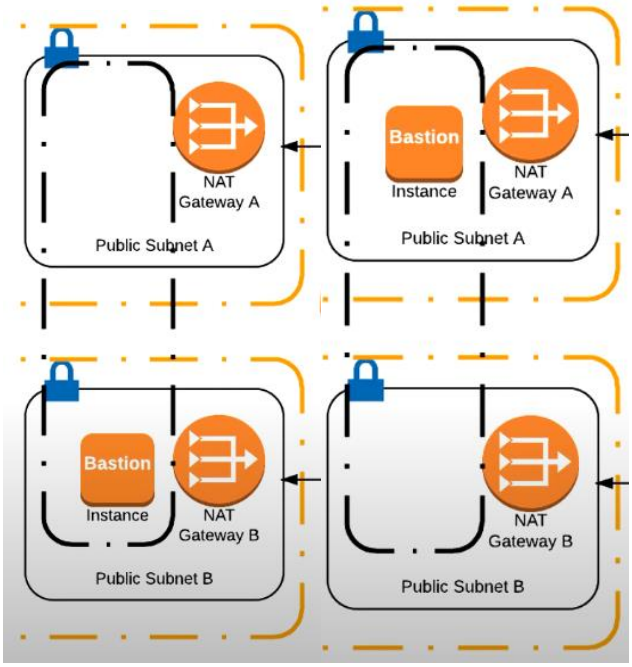


Рис 2.17 Auto Scaling серверів

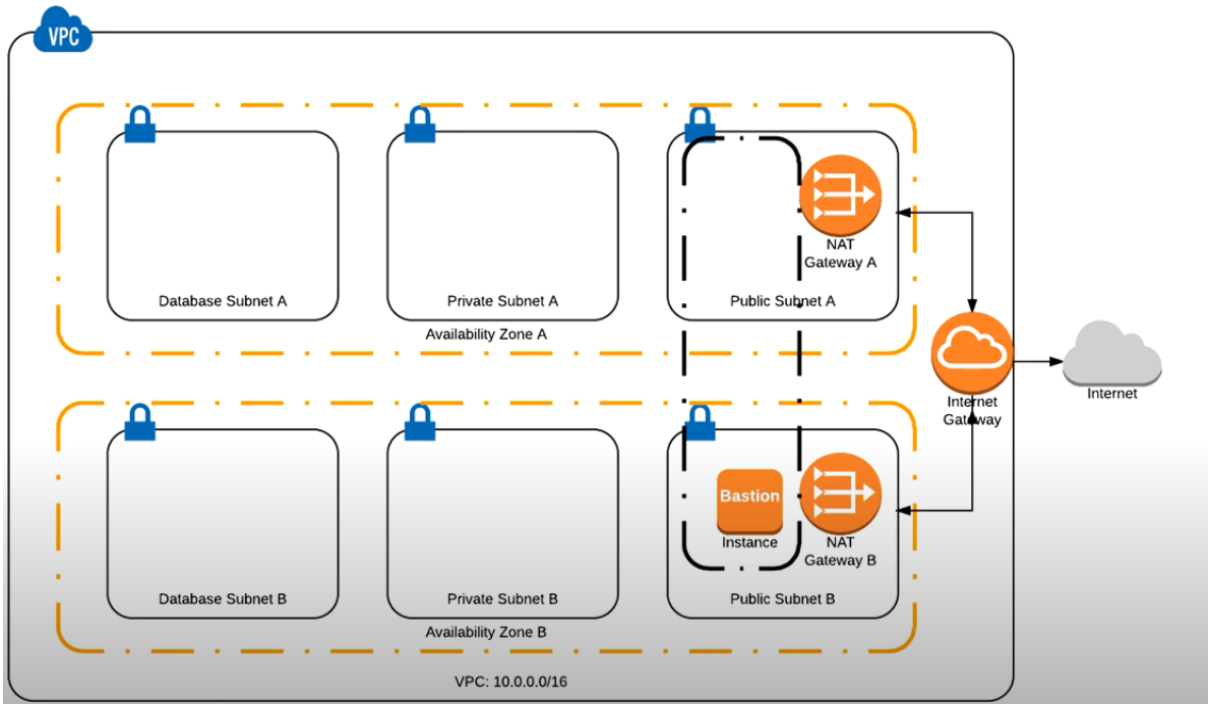
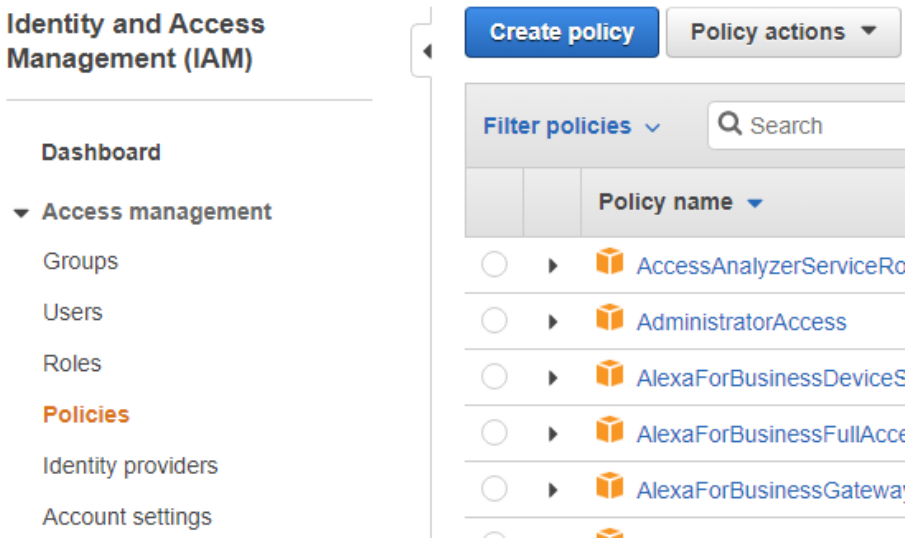


Рис 2.18 Кінцева модель VPC

2.4 Додаткові можливості налаштування VPC

2.4.1 VPC Flow Logs та AWS CloudWatch

Для дослідження інцидентів інформаційної безпеки AWS дозволяє збирати статистику за відправленими та отриманими пакетами на рівні підмережі або певного мережевого адаптера. Для налаштування функції збору статистики необхідно створити IAM Role і надати дозвіл на запис подій до CloudWatch Logs(рис 2.19). Переходимо в розділ IAM > Policies > Create Policy.



2.19 Роль доступу для CloudWatch Logs

Для створення політики використаємо скрипт в форматі JSON (рис 2.20). Він дозволяє наступні дії: CreateLogGroup, CreateLogStream, PutLogEvents, DescribeLogGroups, DescribeLogStreams.

Рис 2.20 Скрипт в форматі JSON

Назвемо політику VPCFlowLogs, перевіримо правильність введених даних та натиснемо Create policy. За аналогією створимо роль доступу. Переходимо в IAM > Roles > Create role. Із запропонованого списку вибираємо EC2, оскільки збираємось збирати дані з серверів. Далі переходимо в Permissions та додаємо політику, яка була створена в попередньому кроці. Називаємо роль VPCFlowLogs та створюємо її. Після цього повертаємось до цієї ролі, переходимо на вкладку Trust Relationship та додаємо скрипт для коректної взаємодії ролі з політикою(рис 2.21).

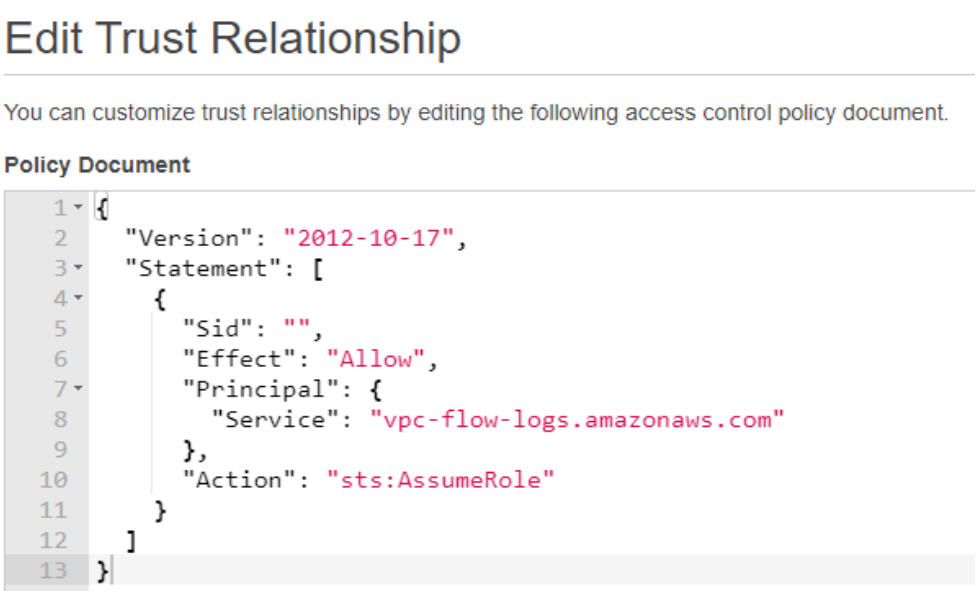


Рис 2.21 Trust Relationship

Створимо групу в яку записуватимуться лог-файли. Для цього переходимо в CloudWatch > Log groups > Create log group(рис 2.22). Тут вписуємо ім'я групи, залишаємо Retention setting в статусі Never expire та створюємо групу.

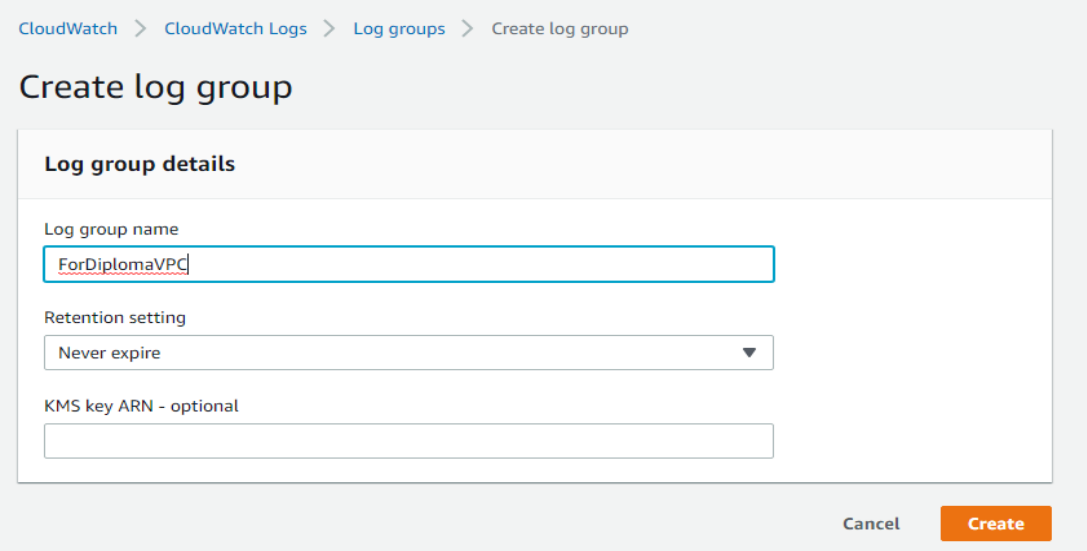


Рис 2.22 Група для лог-файлів

Переходимо до сервісу VPC та відкриваємо створену мережу. Переходимо на вкладку Flow logs та натискаємо Create flow log. Вводимо назву для записів, встановлюємо фільтрацію за всіма пакетами з максимальним інтервалом в 10 хвилин та вказуємо групу для запису даних і роль доступу, які були створені раніше (рис 2.23).

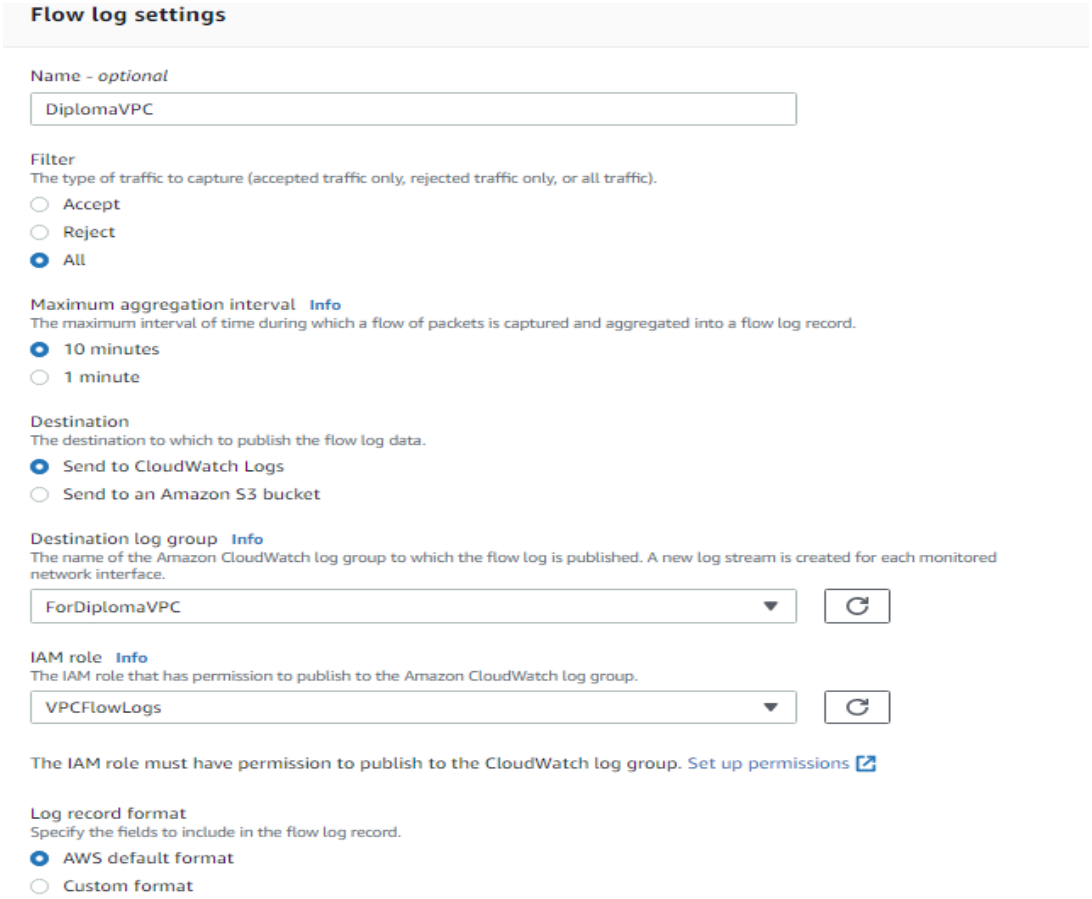


Рис 2.23 Налаштування групи логів

Повертаємось до сервісу CloudWatch, переходимо до відповідної групи та спостерігаємо за відправленими та отриманими пакетами(рис 2.24).

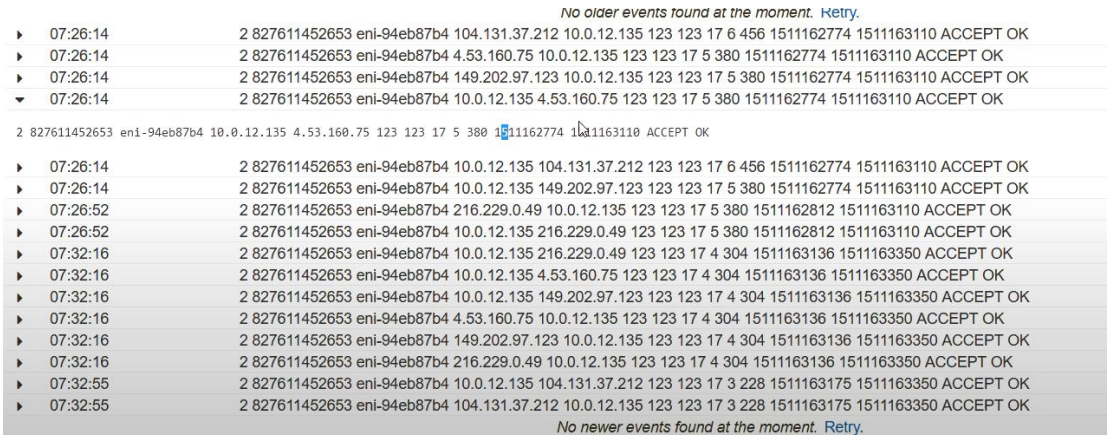


Рис 2.24 Перегляд логів

2.4.2 Запис подій в обліковому записі через CloudTrail

Для коректного налаштування запису подій до CloudTrail необхідно перейти до однойменного сервісу в AWS та натиснути Create trail. Вказуємо

відповідне ім’я та бачимо що разом з записом в CloudTrail буде створено S3 bucket до якого записуватимуться події в хмарі(рис 2.25).

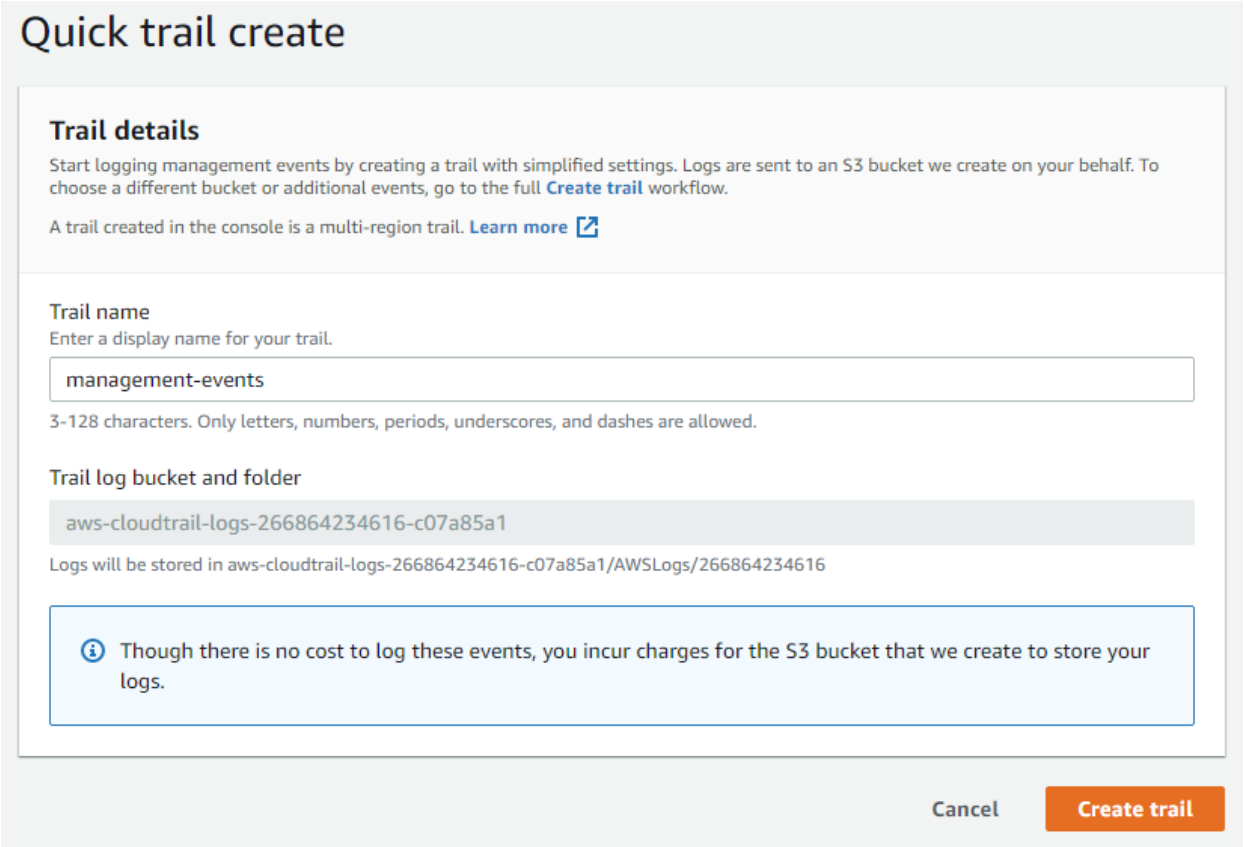


Рис 2.25 Запис подій до CloudTrail

Тепер можемо переглянути лог-файли для певного регіону. Для цього переходимо в S3 та знаходимо відповідний bucket. В ньому знаходимо файл в форматі .json та завантажуємо його. Вміст файлу виглядає наступним чином(рис 2.26):


```
{"Records":[{"eventVersion":"1.08","userIdentity":
{"type":"AWSService","invokedBy":"cloudtrail.amazonaws.com"},
"eventTime":"2020-11-24T12:24:30Z",
"eventSource":"s3.amazonaws.com","eventName":"GetBucketAcl",
"awsRegion":"eu-central-1","sourceIPAddress":"cloudtrail.amazonaws.com",
"userAgent":"cloudtrail.amazonaws.com",
"requestParameters":{"bucketName":"aws-cloudtrail-logs-266864234616-c07a85a1",
"Host":"aws-cloudtrail-logs-266864234616-c07a85a1.s3.eu-central-1.amazonaws.com","acl":""},
"responseElements":null,"additionalEventData":{"SignatureVersion":"SigV4","CipherSuite":"ECDHE-RSA-AES128-SHA",
"bytesTransferredIn":0,"AuthenticationMethod":"AuthHeader",
"x-amz-id-2":"25LT+gVhk60jFZ/hpNudICY9KxyWsTAiwu4qBgxRwPXGJbN21Xuj6LJ0H44VLQP3h8y4nanXZ4U\u003d",
"bytesTransferredOut":480},"requestID":"695F48645C571D0F",
"eventID":"1bbfd443-f51f-45ab-8cfc-282c33c64ec7","readOnly":true,
"resources":[{"accountId":"266864234616","type":"AWS::S3::Bucket",
"ARN":"arn:aws:s3:::aws-cloudtrail-logs-266864234616-c07a85a1"}],
"eventType":"AwsApiCall","managementEvent":true,"eventCategory":"Management",
"recipientAccountId":"266864234616","sharedEventID":"d47245f1-71e8-44d3-9506-8383b84741a0"}]}
```


Рис 2.26 Перегляд логів

Висновки до розділу 2:

В цьому розділі було проведено підготовчі роботи для подальшого написання коду на Terraform: створено та налаштовано обліковий запис в AWS, встановлено Terraform на робочий комп’ютер на створено проект майбутньої віртуальної приватної мережі.

Подібна багаторівнева структура дозволяє суворо контролювати дотримання прав доступу і обмеження безпеки між WEB – серверами, серверами застосунків і баз даних. Доступ між серверами і підмережами можна контролювати використовуючи фільтрацію вхідних та вихідних пакетів. VPC надає розширені можливості забезпечення безпеки. До них належать: групи безпеки і мережеві списки контролю доступу.

Сукупність вищезазначених факторів визначає високий рівень надійності та захищеності інфраструктури. Оптимізований та професійно складений код Terraform дозволяє швидко розгорнути подібну мережу.

РОЗДІЛ 3

НАПИСАННЯ ТА ЗАПУСК КОДУ В TERRAFORM

3.1 Створення коду в Terraform

Перед початком написання коду необхідно авторизувати облікові дані користувача для Terraform, який був створений раніше. Для цього необхідно відкрити CMD та ввести команди (рис 3.1) «SET AWS_ACCESS_KEY_ID=» та «SET AWS_SECRET_ACCESS_KEY=». Такий формат вводу, на відміну від команди SETX, зберігає змінні середовища до кінця поточної сесії CMD, що підвищує рівень безпеки при роботі з AWS.

```
C:\Users\PC>set AWS_ACCESS_KEY_ID=AKIAT4ISZCB4DE2YMY6P

C:\Users\PC>
C:\Users\PC>set AWS_SECRET_ACCESS_KEY=8pmfpo2vwfxtrNOJzq2GWZRnDktRxoESitv++Jy9
```

Рис 3.1 Введення облікових даних в CMD

Наступним кроком необхідно створити папку, в якій зберігатиметься код Terraform. Створюємо цю папку за шляхом: C:\UniverVPC. Також створюємо текстовий файл, в якому писатимемо код та називаємо його VPC.tf.

Приступаємо до написання коду. За загальними правилами роботи з Terraform вказуємо провайдера (рис 3.2) з яким працюємо та регіон, в якому розгортатимемо ресурси.

```
provider "aws" {
  region = "eu-central-1"
}
```

Рис 3.2 Зазначення провайдера для Terraform

Для перевірки правильності роботи вводимо команду «terraform init» в CMD. У виводі бачимо успішний перший запуск Terraform (рис 3.3).

```
C:\UniverVPC>terraform init

Initializing the backend...

Initializing provider plugins...
- Using previously-installed hashicorp/aws v3.18.0
- Using previously-installed -/aws v3.18.0

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, we recommend adding version constraints in a required_providers block
in your configuration, with the constraint strings suggested below.

* -/aws: version = "~> 3.18.0"
* hashicorp/aws: version = "~> 3.18.0"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Рис 3.3 Успішний перший запуск Terraform

Задамо головний ресурс, в межах якого розгортатимуться підмережі та сервери – VPC (рис 3.4).

```
resource "aws_vpc" "Univer_VPC" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "Univer_VPC"
  }
}
```

Рис 3.4 Створення VPC в Terraform

Створюємо першу публічну підмережу використовуючи команду «resource "aws_subnet" "Public1"». Параметр «cidr_block = "10.0.10.0/24"» вказує на діапазон адрес від 10.0.10.0 до 10.0.10.254 для серверів в підмережі. Частина коду: «availability_zone = "eu-central-1a"» говорити про те, що ця підмережа існуватиме лише в межах зони доступності eu-central-1a. Найважливішим параметром на цьому етапі є «map_public_ip_on_launch = true» - він визначає автоматичне присвоєння публічних IP адрес серверам при запуску та робить їх

доступними з мережі Інтернет (рис 3.5). Саме ця ознака відрізняє публічну підмережу від приватної.

```
resource "aws_subnet" "Public1" {
  vpc_id                  = aws_vpc.Univer_VPC.id
  cidr_block              = "10.0.10.0/24"
  availability_zone       = "eu-central-1a"
  map_public_ip_on_launch = true
  tags = {
    Name = "Public1"
  }
}
```

Рис 3.5 Створення першої публічної підмережі

За аналогією створимо ще одну публічну підмережу. Вона матиме cidr_block - "10.0.20.0/24", тобто наступний діапазон IP адрес 10.0.20.0 – 10.0.20.254. Також вона існуватиме лише в межах зони доступності eu-central-1b (рис 3.6).

```
resource "aws_subnet" "Public2" {
  vpc_id                  = aws_vpc.Univer_VPC.id
  cidr_block              = "10.0.20.0/24"
  availability_zone       = "eu-central-1b"
  map_public_ip_on_launch = true
  tags = {
    Name = "Public2"
  }
}
```

Рис 3.6 Створення другої публічної підмережі

Згідно з планом напишемо код для розгортання приватних підмереж. Cidr_block цієї підмережі 10.0.11.0/24, тобто сервери в ній отримуватимуть IP адреси від 10.0.11.0 до 10.0.11.254 та розгортатимуться в межах зони доступності eu-central-1a (рис 3.7). Найважливішою ознакою приватних підмереж є

відсутність публічної IP адреси. За це відповідає параметр «map_public_ip_on_launch = false»

```
resource "aws_subnet" "Private1" {
  vpc_id                  = aws_vpc.Univer_VPC.id
  cidr_block              = "10.0.11.0/24"
  availability_zone       = "eu-central-1a"
  map_public_ip_on_launch = false
  tags = {
    Name = "Private1"
  }
}
```

Рис 3.7 Створення першої приватної підмережі

За аналогією пишемо код для ще однієї приватної підмережі (рис 3.8). Вона матиме cidr_block - 10.0.21.0/24, тобто міститиме IP адреси від 10.0.21.0 до 10.0.21.254 та існуватиме в межах зони доступності eu-central-1b.

```
resource "aws_subnet" "Private2" {
  vpc_id                  = aws_vpc.Univer_VPC.id
  cidr_block              = "10.0.21.0/24"
  availability_zone       = "eu-central-1b"
  map_public_ip_on_launch = false
  tags = {
    Name = "Private2"
  }
}
```

Рис 3.8 Створення другої приватної підмережі

Створюємо підмережу для Баз даних. Перша підмережа матиме cidr_block - 10.0.12.0/24 – міститиме IP адреси від 10.0.12.0 до 10.0.12.254 та існуватиме в зоні доступності eu-central-1a (рис 3.9).

```
resource "aws_subnet" "Database1" {
  vpc_id                  = aws_vpc.Univer_VPC.id
  cidr_block              = "10.0.12.0/24"
  availability_zone       = "eu-central-1a"
  map_public_ip_on_launch = false
  tags = {
    Name = "Database1"
  }
}
```

Рис 3.9 Створення першої підмережі баз даних

Друга підмережа баз даних матиме cidr_block - 10.0.22.0/24 та буде розміщена в зоні доступності eu-central-1b (рис 3.10). Публічних IP адрес для ресурсів в цих підмережах не передбачено, оскільки параметр map_public_ip_on_launch = false.

```
resource "aws_subnet" "Database2" {
  vpc_id                  = aws_vpc.Univer_VPC.id
  cidr_block              = "10.0.22.0/24"
  availability_zone       = "eu-central-1b"
  map_public_ip_on_launch = false
  tags = {
    Name = "Database2"
  }
}
```

Рис 3.10 Створення другої підмережі баз даних

Для отримання доступу до майбутніх серверів необхідно вказати порти, які будуть відкритими. Це можливо реалізувати через налаштування фаєрволу. В Terraform ця змінна звучить як "aws_security_group". Головні необхідні порти це: 22 – для встановлення SSH підключення, 80 – для HTTP протоколу, та 443 – для HTTPS протоколу. В коді це можливо реалізувати через цикл наступним чином (рис 3.11):

```
 dynamic "ingress" {
   for_each = ["22", "443", "80"]
   content {
```

Рис 3.11 Цикл відкритих портів для security group

Також необхідно вказати вхідні та вихідні величини для безкласової маршрутизації з діапазоном IP адрес 0.0.0.0/0. Повна частина коду має наступний вигляд (рис 3.12):

```
resource "aws_security_group" "Univer_VPC" {
  name   = "Univer_VPC"
  vpc_id = aws_vpc.Univer_VPC.id

  dynamic "ingress" {
    for_each = ["22", "443", "80"]
    content {
      description = "TLS from VPC"
      from_port   = ingress.value
      to_port     = ingress.value
      protocol    = "tcp"
      cidr_blocks = ["0.0.0.0/0"]
    }
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Рис 3.12 Повний блок коду для security group

Для доступу до мережі Інтернет з публічних підмереж необхідно створити інтернет шлюз (рис 3.13). Це можливо здійснити командою "aws_internet_gateway":

```
resource "aws_internet_gateway" "Univer_VPC_GW" {
  vpc_id = aws_vpc.Univer_VPC.id

  tags = {
    Name = "Univer_VPC_GW"
  }
}
```

Рис 3.13 Створення інтернет шлюзу

Для коректної маршрутизації пакетів з серверів в публічних підмережах створимо таблицю маршрутизації, в якій зазначимо, що увесь трафік необхідно маршрутизувати на інтернет шлюз (рис 3.14).

```
resource "aws_default_route_table" "Univer_VPC_RT" {
  default_route_table_id = aws_vpc.Univer_VPC.default_route_table_id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.Univer_VPC_GW.id
  }
  tags = {
    Name = "Univer_VPC_RT"
  }
}
```

Рис 3.14 Створення таблиці маршрутизації

Також необхідно створити асоціацію таблиці маршрутизації з приватною підмережою №1 та №2 (рис 3.15).

```
resource "aws_route_table_association" "Public1" {
  subnet_id      = aws_subnet.Public1.id
  route_table_id = aws_default_route_table.Univer_VPC_RT.id

}
resource "aws_route_table_association" "Public2" {
  subnet_id      = aws_subnet.Public2.id
  route_table_id = aws_default_route_table.Univer_VPC_RT.id
}
```

Рис 3.15 Створення асоціації таблиці маршрутизації

Для доступу до мережі Інтернет з серверів в приватних підмережах необхідно створити NAT шлюзи для кожної з них (рис 3.16). Кожен NAT шлюз в AWS вимагає окремої публічної еластичної IP адреси (рис 3.17). В Terraform цей ресурс називається "aws_eip".

```
resource "aws_nat_gateway" "NATGW1" {
  allocation_id = aws_eip.EIP1.id
  subnet_id     = aws_subnet.Public1.id

  tags = {
    Name = "NATGW1"
  }
}
```

Рис 3.16 Створення NAT шлюзу

```
resource "aws_eip" "EIP1" {
  vpc      = true
}
```

Рис 3.17 Створення окремої публічної IP адреси

Наступним кроком необхідно створити таблицю маршрутизації для приватних підмереж (рис 3.17), згідно з якою трафік буде маршрутизуватись на відповідні NAT шлюзи, а також вказати для якої саме підмережі створено конкретний шлюз (рис 3.18).

```
resource "aws_route_table" "PrivateRT1" {
  vpc_id = aws_vpc.Univer_VPC.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_nat_gateway.NATGW1.id
  }
  tags = {
     Name = "PrivateRT1"
   }
```

Рис 3.17 Створення таблиці маршрутизації для приватних підмереж

```
resource "aws_route_table_association" "Private1" {
  subnet_id      = aws_subnet.Private1.id
  route_table_id = aws_route_table.PrivateRT1.id

}
```

Рис 3.18 Створення асоціації таблиці маршрутизації для приватних підмереж

Підмережі баз даних не потребують додаткових налаштувань маршрутизації, оскільки вони не мають жодного доступу до глобальної мережі, а спілкуються з ресурсами лише в межах віртуальної приватної хмари.

Додамо можливість автоматичного балансування навантаження на сервери в межах різних зон доступності. Це можливо реалізувати через Elastic Load Balancing. Для цього в коді необхідно вказати ресурс "aws_lb" та зазначити необхідні підмережі, між якими відбуватиметься розподіл навантаження (рис 3.19). За аналогією створюємо Load Balancing для усіх трьох підмереж.

```
resource "aws_lb" "PublicLB" {
   name               = "PublicLB"
   internal           = false
   load_balancer_type = "application"
   security_groups    = [aws_security_group.Univer_VPC.id]
   subnet_mapping {
  subnet_id     = aws_subnet.Public1.id
   }
   subnet_mapping {
 subnet_id     = aws_subnet.Public2.id
  }
}
```

Рис 3.19 Створення Load Balancing

Перейдемо до безпосереднього запуску серверів. В Terraform за це відповідає ресурс "aws_instance" (рис 3.21). Для перевірки коректності коду запустимо сервери в публічній підмережі №1, приватній підмережі №2 та

підмережі баз даних №1. Тип серверів - t2.micro, образ - ami-0c2b1c303a2e4cb49(Ubuntu 20.04), ключ – створений попередньо та завантажений на робочий комп'ютер. Також на кожен сервер через user_data додано скрипт, який запускатиме ВЕБ сторінку з певним вмістом (рис 3.20).

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install apache2 -y
sudo chown -R ubuntu /var/www/html -y
myip=`curl http://169.254.169.254/latest/meta-data/local-ipv4`
echo "<h2>WebServer with IP: $myip</h2><br>Build by Terraform special for NAU!"
> /var/www/html/index.html
sudo service apache2 start
```

Рис 3.20 Створення ВЕБ сторінки на сервері

```
resource "aws_instance" "Univer_VPC_1" {
  ami                    = "ami-0c2b1c303a2e4cb49"
  instance_type          = "t2.micro"
  key_name               = "frankfurt"
  subnet_id              = aws_subnet.Public1.id
  vpc_security_group_ids = [aws_security_group.Univer_VPC.id]
  user_data              = file("userdata.sh")
  tags = {
    Name = "Univer_VPC_1"
  }
}
```

Рис 3.21 Створення коду для ВЕБ сервера

Наприкінці необхідно створити вузол-бастіон, який існуватиме постійно в межах публічних підмереж. Для цього необхідно використати параметр lifecycle, який вказує на заборону знищення цього сервера до створення копії такого ж зразка.

```
lifecycle {
    create_before_destroy = true
  }
```

Рис 3.22 Створення вузла-бастіона

3.2 Перевірка коректності виконання роботи

Після написання коду настав час вперше виконати його. Для початку в CMD перейдемо до папки зі скрипт-файлом та виконаємо команду “terraform init” – вона створить всі необхідні файли для майбутньої коректної роботи Terraform (рис 3.23).

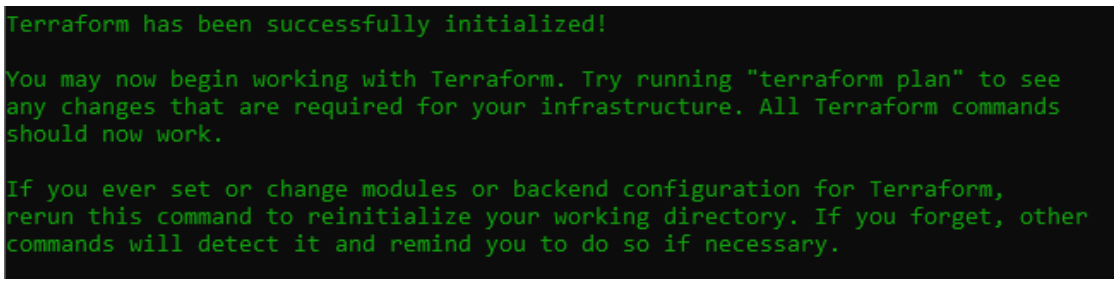


Рис 3.23 Результат першого запуску коду

Виконаємо команду “terraform plan” і отримаємо вивід усіх ресурсів, які будуть створені а AWS. Команда “terraform apply” створить усі заплановані ресурси (рис 3.23). Після підрахунку майбутніх змін Terraform просить ввести “yes” для підтвердження розгортання ресурсів. Вводимо “yes”.

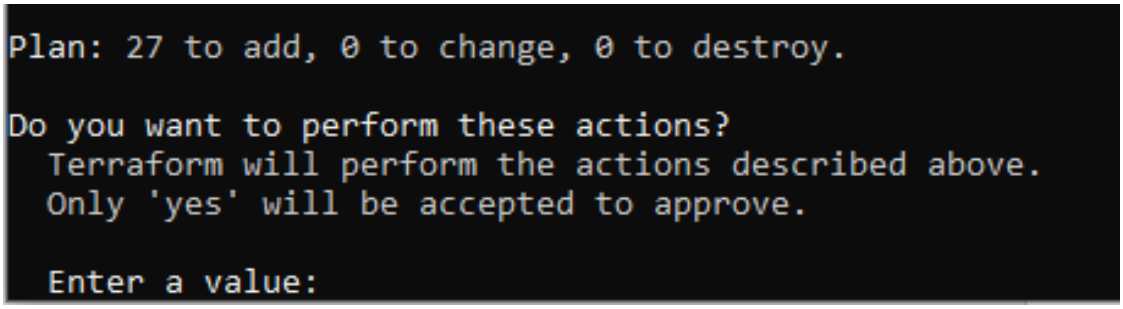


Рис 3.23 Результат команди “terraform apply”

Після застосування змін бачимо, що усі 27 ресурсів успішно додані до AWS, та зауважимо що на це було витрачено 2 хвилини та 13 секунд (рис 3.24).

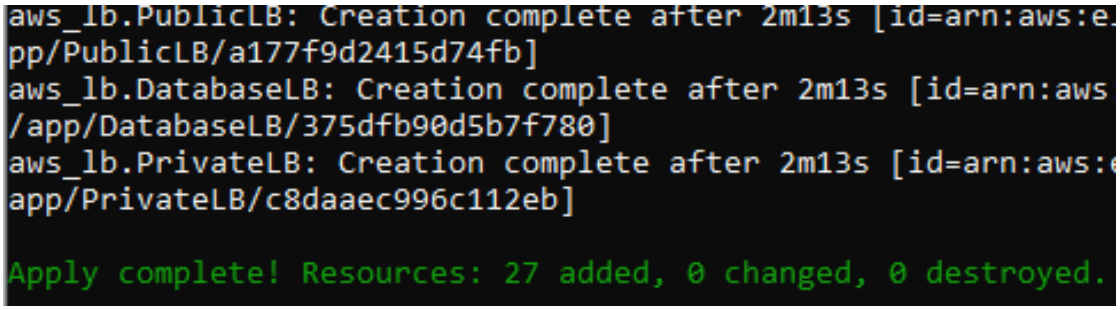


Рис 3.24 Результат команди “terraform apply”

Перейдемо на сайт https://aws.amazon.com/ щоб перевірити правильність розгортання ресурсів. В сервісі EC2 бачимо чотири сервери, кожен з яких знаходиться в потрібній підмережі та лише два з них мають публічні IP адреси (рис 3.25).

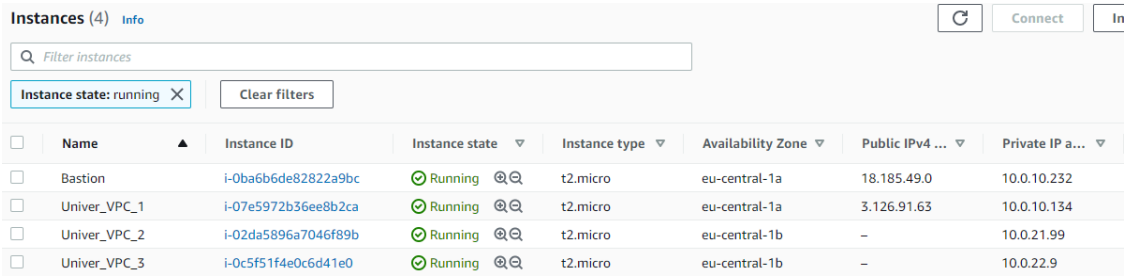


Рис 3.25 Розгорнуті сервери

В розділі Load Balancers присутні три ресурси для кожної з підмереж (рис 3.26).

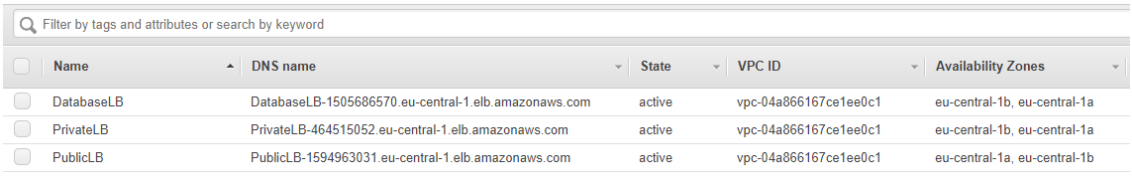


Рис 3.26 Load Balancers

В сервісі VPC бачимо нашу віртуальну мережу та шість підмереж (рис 3.27).

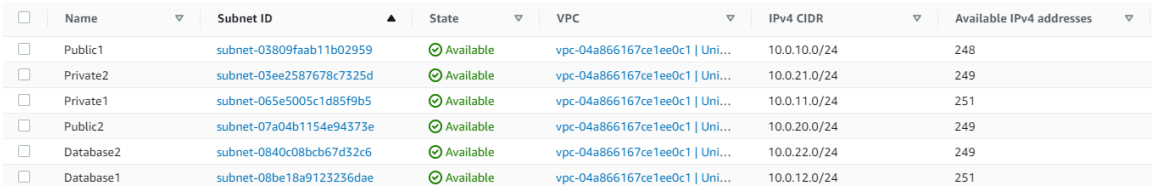


Рис 3.27 Віртуальна мережа та підмережі

В таблицях маршрутизації існують три ресурси, асоційовані з потрібними шлюзами (рис 3.28).

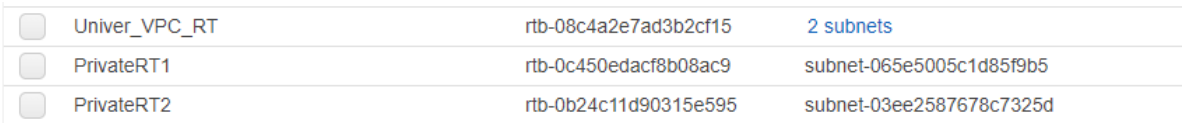


Рис 3.28 Віртуальна мережа та підмережі

Також бачимо інтернет шлюз (рис 3.29) та два NAT – шлюзи (рис 3.30).

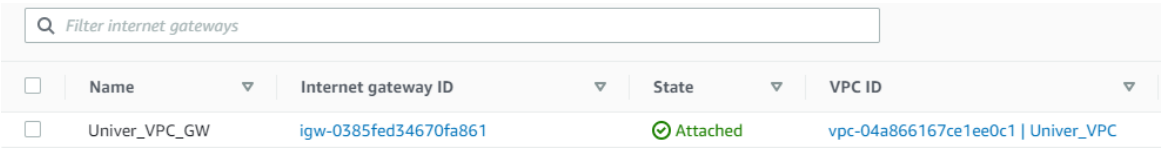


Рис 3.29 Інтернет шлюз

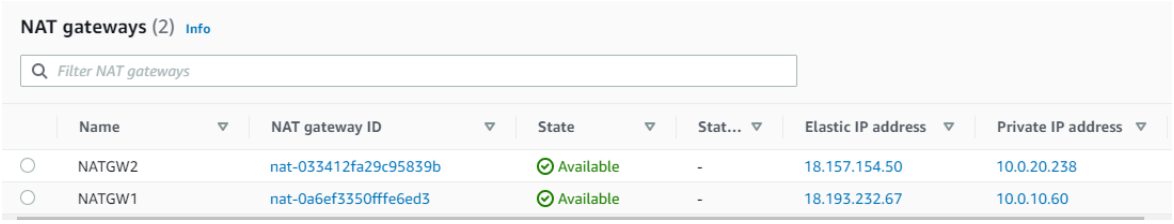


Рис 3.30 NAT – шлюзи

Перевіримо правильність відображення веб-сторінки на сервері, який знаходиться в публічній підмережі. Для цього переходимо за його IP-адресою в браузері (рис 3.31). На сторінці бачимо наступі дані:

WebServer with IP: 10.0.10.92

Build by Terraform special for NAU!

Рис 3.31 Сторінка на серверів в публічній підмережі

3.3 Аналіз отриманих результатів

Час, який необхідно витратити на побудову подібної інфраструктури вручну складає 2 години. На противагу цьому, на написання коду було витрачено 40 хвилин, та 2 хвилини і 13 секунд на запуск сервісів через Terraform.

Опираючись на ці дані можна сказати, що наш підхід зменшив час, який потрібно витратити на побудову інфраструктури втричі. Проте, в нас залишається код, яким можна розгорнути безліч подібних мереж. Тобто, для побудови подібної інфраструктури в майбутньому не доведеться витрачати 40 хвилин на повторне створення коду, лише близько 2 хвилин на повний запуск усіх сервісів.

Таб. 3.1

Витрати часу на ручне розгортання ресурсів

	Сервер	Віртуальна мережа	Підмережа	Інтернет-шлюз	NAT-шлюз	Таблиця маршрутизації	Балансування навантаження
Час розгортання(хв)	8	12	4	14	7	3	5
Кількість ресурсів	4	1	6	1	2	3	3
Загальний час розгортання	32	12	24	14	14	9	15
Сумарний час(хв)	**120**						

Висновки до розділу 3

В даному розділі дипломної роботи було написано та пояснено оптимізований код для Terraform, який розгортає віртуальну приватну хмару в AWS з шістьма підмережами, Інтернет шлюзом, двома NAT шлюзами, трьома серверами з автоматичним балансуванням та вузлом-бастіоном за 2 хвилини та 13 секунд.

Час, який необхідно витратити на побудову подібної інфраструктури вручну складає 2 години. На противагу цьому, на написання коду було витрачено 40 хвилин, та 2 хвилини і 13 секунд на запуск сервісів через Terraform.

Опираючись на ці дані можна сказати, що наш підхід зменшив час, який потрібно витратити на побудову інфраструктури втричі. Проте, в нас залишається код, яким можна розгорнути безліч подібних мереж. Тобто, для побудови подібної інфраструктури в майбутньому не доведеться витрачати 40 хвилин на повторне створення коду, лише близько 2 хвилин на повний запуск усіх сервісів.

ВИСНОВКИ

Головною метою цієї роботи було удосконалення модуля автоматизованої розгортки хмарного постачальника AWS. Для реалізації цього проекту було створено код Terraform, який розгортає захищену, відмовостійку та високодоступну інфраструктуру.

У процесі виконання роботи отримані наступні результати:

Розроблено модель порушника та модель загроз для хмарної інфраструктури. Відштовхуючись від цього визначено, що найбільш безпечним варіантом буде використання IAC (інраструктура як код) інструментів. Подібні інструменти використовують ключі шифрування та дозволяють керувати хмарною інфраструктурою через запити у вигляді коду. Такий підхід мінімізує кількість осіб, які мають адміністративний доступ до хмари та час, за який відбувається побудова та редагування інфраструктури.

Створено проект віртуальної приватної мережі. Його багаторівнева структура дозволяє суворо контролювати дотримання прав доступу і обмеження безпеки між WEB – серверами, серверами застосунків і баз даних. Доступ між серверами і підмережами можна контролювати використовуючи фільтрацію вхідних та вихідних пакетів. VPC надає розширені можливості забезпечення безпеки. До них належать: групи безпеки і мережеві списки контролю доступу.

Написано та пояснено оптимізований код для Terraform, який розгортає віртуальну приватну хмару в AWS з шістьма підмережами, Інтернет шлюзом, двома NAT шлюзами, трьома серверами з автоматичним балансуванням та вузлом-бастіоном за 2 хвилини та 13 секунд. При такому підході отримуємо зменшення загальних витрат часу в 53 рази.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Максименко Ю.Є. Теоретико-правові засади забезпечення інформаційної безпеки України: Дис. канд. юрид. наук. – К., 2007.

2. Ліпкан В.А. Національна безпека України: Навч. посіб. – 2-ге вид. – К., 2009.

3. Про прийняття за основу проекту Закону України «Про Концепцію державної інформаційної

4. Постанова Верховної Ради України від 11.01.2011 № 2897-VI [Електронний ресурс]. – Режим доступу: http://zakon1.rada.gov.ua.

5. Концепція (основи державної політики) національної безпеки України: Схвалено постановою Верховної Ради України від 16.01.1997 // Урядовий кур'єр. – 1997. – 06 лют.

6. Про основи національної безпеки України: Закон України від 19.06.2003 р. [Електронний ресурс]. Режим доступу: http://uadocs.exdat.com/docs/index-208817.html.

7. Ярочкин В.И. Информационная безопасность: Учеб. пособие для студ. непрофильных вузов. – М., 2000. Новицька Н.Б. Правове забезпечення інформаційної безпеки // Інформаційна безпека людини, суспільства, держави. – 2009. – № 1. – С. 44-47.

8. The NIST Definition of Cloud Computing, NIST Special Publication 800-145, 2011.

9. Guidelines on Security and Privacy in Public Cloud Computing , NIST SP800-144, 2011.

10. Cloud Computing Synopsis and Recommendations DRAFT, NIST, 2011.

11. Security Guidance for Critical Areas of Focus in Cloud Computing, Version 3.0. Technical report, Cloud Security Alliance, 2011. Режим доступу http://www.cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

12. D. Catteddu and G. Hogben. Cloud Computing Security Risk Assessment. Technical report, European

13. Network and Information Security Agency, November 2009.

14. D. Catteddu and G. Hogben. Cloud Computing Information Assurance Framework. Technical report, European Network and Information Security Agency, November 2009.

15. Cloud-computing-information-assurance-framework Wayne Jansen, Timothy Grance Guidelines on Security

16. AWS Certified Solutions Architect Official Study Guide: Associate Exam (Aws Certified Solutions Architect Official: Associate Exam) – Joe Baron.

17. AWS Certified Solutions Architect Associate Training Notes 2019 – Neal Davis.

18. Amazon Web Services in Action - Andreas Wittig

19. Learn Amazon Web Services in a Month of Lunches - David Clinton

20. Mastering AWS Security: Create and maintain a secure cloud ecosystem - Albert Anthony

21. AWS Certified Security – Specialty Exam Guide - Stuart Scott

22. Automating Security in the Cloud: Modernizing Governance through Security Design 1st Edition - Tim Sandage