НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри

Комп'ютерних систем, мереж та кібербезпеки

_____ Касаткін Д.Ю., к. пед.н., доц.

підпис ПІБ, вчене звання і ступінь

«___»_____ 2025 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

На тему: «Впровадження засобів віртуалізації для покращення безпеки серверних систем університету»

Спеціальність F7 «Комп’ютерна інженерія»

Гарант освітньої програми к.фіз.-мат.н., доц. Нікітенко Є.В.

Керівник дипломного проекту: _____ / Лахно В.А. /

підпис ПІБ

Виконав: _____ / _____ /

підпис ПІБ

КИЇВ-2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

«ЗАТВЕРДЖУЮ»
завідувач кафедри
комп'ютерних систем, мереж та кібербезпеки
_____/ Касаткін Д.Ю., к.п.н., доц. /
підпис ПІБ, вчене звання і ступінь
«__» _____ 20__ р.

З А В Д А Н Н Я

ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ СТУДЕНТУ

Орищенко Кирило Андрійович
(прізвище, ім'я, по батькові)
Спеціальність (напрям підготовки): комп'ютерна інженерія

Тема кваліфікаційної бакалаврської роботи:
«Впровадження засобів віртуалізації для покращення безпеки серверних систем університету»

затверджена наказом ректора НУБіП України від "___" _____202_р. №____
Термін подання завершеної роботи на кафедру

Вихідні дані до кваліфікаційної бакалаврської роботи

Перелік питань, що підлягають розробці:
1.
2.
3.
Перелік графічного матеріалу (за потреби)

Дата видачі завдання "_____"_____2025 р.

Керівник бакалаврської роботи _____ Лахно В.А.
(підпис) (прізвище та ініціали)
Завдання прийняв до виконання _____ Орищенко К.А.
(підпис) (прізвище та ініціали студента)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області	27.03.2025 р.	Виконано
2	Проектування системи	15.04.2025 р.	Виконано
3	Реалізація системи	28.04.2025 р.	Виконано
4	Тестування системи	01.05.2025 р.	Виконано
5	Оформлення пояснювальної записки	22.05.2025 р.	Виконано
6	Оформлення графічного матеріалу	10.05.2025 р.	Виконано

Студент _____ Орищенко К.А.

(підпис) (ініціали та прізвище)

Керівник проекту (роботи) _____ **ЛАХНО** В.А.

(підпис) (ініціали та прізвище)

РЕФЕРАТ

Пояснювальна записка: 100 сторінок, 15 рисунків, 1 таблиць, 48 джерел.

ВІРТУАЛІЗАЦІЯ, ГІПЕРВІЗОР, БЕЗПЕКА, СЕРВЕР, ВІРТУАЛЬНА МАШИНА.

Об'єкт дослідження — серверна інфраструктура університету.

Метою роботи є впровадження засобів віртуалізації з метою підвищення рівня безпеки серверних систем університету.

Проєкт складається з трьох розділів.

Перший розділ присвячено загальному опису інформаційної інфраструктури університету. Надано характеристику серверного середовища, визначено проблеми безпеки та розглянуто сучасні загрози для IT-систем в освітніх установах.

У другому розділі проаналізовано апаратні й програмні рішення для віртуалізації, зокрема технології VMware, Hyper-V та VirtualBox. Наведено порівняльну характеристику гіпервізорів, визначено доцільність впровадження конкретних рішень у середовищі університету.

Третій розділ присвячено практичному впровадженню віртуального середовища, зокрема налаштуванню серверної віртуальної інфраструктури, створенню ізольованих середовищ для підвищення безпеки, а також моделюванню типових загроз і способів їхнього усунення. Наприкінці розділу сформульовано рекомендації щодо подальшої модернізації та розвитку віртуальної інфраструктури.

У результаті виконання дипломної роботи було проаналізовано сучасні засоби віртуалізації, проведено їх впровадження в тестовому середовищі, досліджено ефективність захисту інформації та сформовано рекомендації щодо підвищення безпеки серверних систем університету на основі віртуалізації.

ЗМІСТ

ВСТУП

Віртуалізація є однією з найважливіших технологій сучасних інформаційних систем, що дозволяє значно підвищити ефективність використання апаратних ресурсів, знизити витрати на інфраструктуру та покращити управління IT-ресурсами. У останні роки віртуалізація стала основою для багатьох великих і малих підприємств, забезпечуючи їм необхідну гнучкість і масштабованість у використанні своїх обчислювальних потужностей. Одним із найпоширеніших рішень для створення віртуалізованих серверів є платформа Proxmox VE, яка є потужним гіпервізором, що підтримує віртуалізацію на основі KVM (для створення віртуальних машин) та контейнеризацію за допомогою LXC (для створення контейнерів).

Метою даної дипломної роботи є розгортання віртуалізованого серверного середовища на базі Proxmox VE з налаштуванням декількох віртуальних машин, що включають різноманітні базові сервіси для забезпечення стабільної та безпечної роботи IT-інфраструктури університету. Проєкт передбачає створення віртуальних машин для таких сервісів, як DNS/DHCP, веб-сервер, сервер баз даних, файловий сервер або Nextcloud. Також буде налаштовано мережеву ізоляцію між віртуальними машинами, механізми резервного копіювання, міжмережевий екран та контролю доступу до ресурсів.

Використання таких технологій, як Proxmox VE, Linux, Apache / Nginx, MySQL / PostgreSQL, інструменти резервного копіювання та firewall, дозволяє створити високонадійне та безпечне середовище для потреб університетської IT-інфраструктури. У результаті реалізації цього проєкту буде продемонстровано ефективність та безпеку віртуалізованого середовища, а також його здатність адаптуватися до змінюваних вимог і забезпечити належний рівень обслуговування.

У дипломній роботі будуть розглянуті основні етапи реалізації проєкту, зокрема, підготовка серверного обладнання, встановлення та налаштування гіпервізора, створення і налаштування віртуальних машин, забезпечення ізоляції між ними, а також коментарі щодо налаштувань, скриптів і конфігурацій. Особлива увага буде приділена питанням оптимізації та моніторингу віртуальних машин, а також заходам з безпеки і резервного копіювання.

Дана робота є актуальною, оскільки допомагає вирішити важливі питання забезпечення безпеки, доступності та масштабованості IT-ресурсів у великих організаціях та університетах, де необхідно ефективно використовувати обмежені апаратні ресурси та одночасно гарантувати високу доступність і надійність сервісів.

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

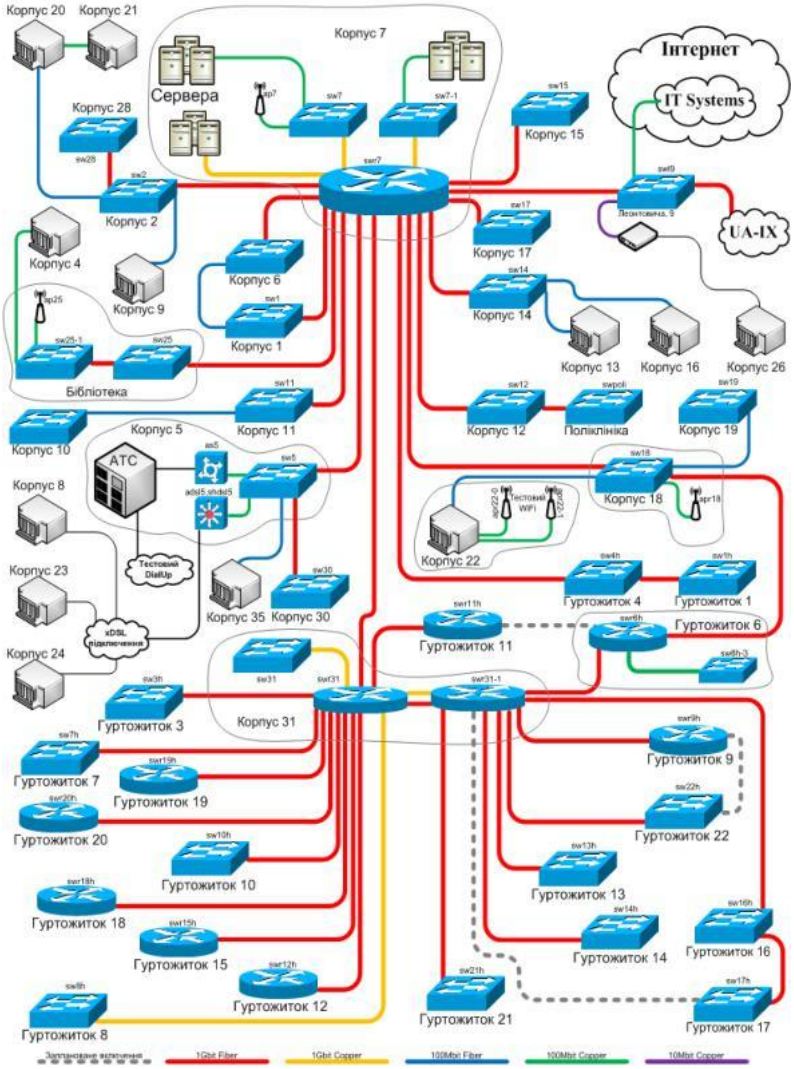
1.1. Загальна характеристика серверної інфраструктури університету

Інформаційно-телекомунікаційна інфраструктура університету є однією з найрозвиненіших серед закладів вищої освіти країни. Це обумовлено не лише масштабами навчального закладу, а й системним підходом до впровадження інноваційних рішень та розширення спектру послуг для користувачів. Основу інфраструктури складає кампусова мережа, яка включає інформаційні, обчислювальні та комунікаційні ресурси, що інтегрують різноманітні сервіси – від внутрішніх баз даних і систем дистанційного навчання до високошвидкісного доступу до глобальної мережі Інтернет.

Експлуатацію та розвиток телекомунікаційної мережі здійснює науково-технічне об’єднання «КПІ-Телеком», створене в 2002 році за ініціативи проректора з наукової роботи з метою централізації управління інформаційними системами та оптимізації витрат на їх обслуговування. Кампусова мережа побудована на базі сучасної оптоволоконної інфраструктури з використанням технології Gigabit Ethernet, що забезпечує пропускну спроможність до 1 Гбіт/с на магістральних каналах. Загалом до інформаційної мережі підключено 29 навчальних корпусів (з них 26 — по оптичним каналам), усі 20 гуртожитків, а також адміністративні та обчислювальні центри [1].

Серверна інфраструктура університету базується на центральному серверному парку, що обслуговується НТО «КПІ-Телеком». Центральний сегмент налічує 17 серверів, які забезпечують функціонування основних інформаційних систем, включно з корпоративними базами даних, електронною поштою, хостингом вебсайтів підрозділів та підтримкою внутрішніх сервісів. Крім того, у структурних підрозділах функціонують ще 18 локальних серверів, що дозволяє ефективно розподіляти обчислювальне

навантаження та підвищувати надійність систем загалом. З метою забезпечення безперервності роботи критичних служб реалізовано системи безперебійного живлення для основних мережевих вузлів, зокрема у 7-му корпусі (ядро навчальної мережі) та 31-му корпусі (ядро мережі студмістечка).



Для задоволення сучасних вимог до надійності та масштабованості інфраструктури, у вузлових точках встановлено високопродуктивні комутатори з підтримкою швидкості комутації до 128 Гбіт/с. Це дозволяє реалізовувати високонавантажені проекти, зокрема сервіси IP-телефонії, передавання потокового відео та мультимедійний контент у режимі реального часу.

Окрему увагу приділено розвитку інфраструктури бездротового доступу. Починаючи з 2016 року, в університеті активно розгортається мережа Wi-Fi стандарту IEEE 802.11g. Станом на кінець 2017 року встановлено 12 точок доступу, розміщених як у корпусах (№1, 6, 7, 18, 22, бібліотека), так і на відкритих територіях. Планується подальше розширення покриття, зокрема вздовж вулиці Політехнічної [2].

Університет також забезпечує віддалений доступ до внутрішніх інформаційних ресурсів для співробітників, які проживають на території студмістечка. Для цього реалізовано проєкт підключення через технології ADSL/ADSL2+ по наявних телефонних лініях, а також запроваджено сервер доступу DIAL-UP. Це дозволяє підключатися до університетської мережі з приватних помешкань, використовуючи звичайне телефонне з’єднання.

Важливою складовою інфраструктури є взаємодія з глобальною мережею Інтернет. З 2001 року університет має власний блок зовнішніх IP-адрес, а з 2005 року, після його вичерпання, отримано новий пул (32768 IP-адрес) у результаті реєстрації в організації RIPE NCC, що надало університету статус локального Інтернет-реєстратора (LIR). Це забезпечує повну

автономність в адмініструванні адресного простору, створенні автономних систем і гнучке масштабування мережевих сервісів.

Таким чином, серверна інфраструктура університету відзначається високим рівнем організації, технічної модернізації та здатністю до масштабування. Завдяки централізованому управлінню, раціональному використанню ресурсів та системному розвитку, вона повною мірою відповідає вимогам сучасного освітнього середовища.

1.2. Аналіз існуючих рішень щодо захисту серверних систем

Сучасні серверні інфраструктури є об’єктами підвищеного ризику внаслідок постійного зростання кількості кіберзагроз, як з боку зовнішніх атакуючих, так і через внутрішні вразливості. Ефективний захист серверних систем передбачає комплексне впровадження організаційних, програмних та апаратних заходів безпеки. У цьому підрозділі проведено аналіз основних напрямів і технологій, що використовуються для захисту серверних середовищ.

Брандмауери залишаються базовим елементом захисту серверів. Вони можуть бути апаратними (наприклад, Cisco ASA, Fortinet) або програмними (iptables для Linux, Windows Defender Firewall). Основна функція брандмауера — контроль вхідного та вихідного трафіку відповідно до заданих правил. Вони дозволяють обмежити доступ до критичних сервісів та виявити спроби несанкціонованого підключення.

Системи IDS/IPS (Intrusion Detection/Prevention Systems) призначені для виявлення аномального трафіку або підозрілих дій. Найпоширенішими відкритими рішеннями є Snort та Suricata. Вони аналізують мережеві пакети в реальному часі та реагують на підозрілі шаблони, блокуючи або фіксуючи потенційні загрози [3].

Захист від шкідливого ПЗ забезпечується шляхом розгортання антивірусних рішень (наприклад, Kaspersky Endpoint Security, ESET Server

Security, Microsoft Defender for Endpoint). Важливим є не лише виявлення, але й моніторинг активності шкідливих процесів, а також можливість централізованого керування політиками безпеки.

Сегментація мережі дозволяє ізолювати сервери з різними рівнями довіри, зменшуючи ризик горизонтального переміщення атакуючого в інфраструктурі. Віртуалізовані середовища (на базі VMware, Hyper-V або KVM) також дозволяють створювати ізольовані віртуальні машини з окремими політиками безпеки, а використання гіпервізора надає додатковий рівень контролю доступу.

Шифрування є критично важливим для захисту конфіденційної інформації. На серверному рівні застосовуються як симетричні (AES), так і асиметричні (RSA, ECC) алгоритми шифрування. Шифрування може бути застосоване до з'єднань (наприклад, TLS/SSL), файлів, баз даних або навіть цілих віртуальних дисків [4].

Системи аудиту та журналювання (auditd у Linux, Windows Event Log) дають змогу фіксувати всі дії користувачів і системних процесів. Це дозволяє виявляти інциденти безпеки, аналізувати їх наслідки та вдосконалювати політики доступу. Журнали повинні зберігатися в захищеному середовищі, бажано на віддаленому сервері.

Наявність політики регулярного резервного копіювання критичних серверів є обов'язковою умовою безпеки. Рішення, такі як Veeam Backup, Acronis або Bacula, дозволяють створювати як повні, так і інкрементальні копії даних. Важливою складовою є тестування відновлення — перевірка працездатності резервних копій у разі інциденту.

Використання багатофакторної автентифікації (2FA) значно знижує ризик компрометації облікових записів. Контроль доступу реалізується через ролі, групи користувачів та ACL (Access Control Lists). Додатково застосовуються технології Single Sign-On (SSO) та протоколи Kerberos, RADIUS, LDAP.

Регулярне оновлення операційних систем і прикладного ПЗ є основним засобом запобігання експлуатації відомих вразливостей. Інструменти, як-от WSUS для Windows Server або автоматизовані системи (наприклад, Ansible, Puppet), дозволяють централізовано розгортати патчі.

1.3. Огляд технологій віртуалізації

Віртуалізація є однією з базових технологій сучасної інформаційної інфраструктури, що дозволяє ефективно використовувати обчислювальні ресурси, спрощувати управління системами та підвищувати їхню надійність і масштабованість. Сутність віртуалізації полягає у створенні віртуального (нефізичного) представлення апаратного чи програмного компонента, що дозволяє запускати на одному фізичному сервері декілька ізольованих середовищ.

Залежно від реалізації, віртуалізація поділяється на кілька основних типів: віртуалізацію серверів, сховищ, мереж та робочих станцій. Віртуалізація серверів дає змогу розміщувати кілька віртуальних серверів на одному фізичному, забезпечуючи гнучке розподілення ресурсів і зниження витрат на обладнання. Віртуалізація сховищ об'єднує фізичні носії в єдині логічні структури, спрощуючи управління даними. Мережева віртуалізація дозволяє створювати окремі віртуальні мережі в межах однієї фізичної інфраструктури, а віртуалізація робочих станцій забезпечує віддалений доступ до віртуальних робочих середовищ [5].

Центральним елементом віртуалізації є гіпервізор — програмне забезпечення, що дозволяє створювати та керувати віртуальними машинами. Існує два основних типи гіпервізорів. Перший тип, або "bare-metal" гіпервізори, встановлюється безпосередньо на фізичне обладнання та забезпечує максимальну продуктивність і безпеку. До таких гіпервізорів належать VMware ESXi, Microsoft Hyper-V та KVM. Другий тип гіпервізорів працює поверх операційної системи хоста — їх часто використовують у

навчальних цілях або для тестування, адже вони простіші у використанні, але менш продуктивні. До цього типу належать VirtualBox, VMware Workstation тощо.

Переваги використання віртуалізації є численними. Насамперед, це ефективніше використання ресурсів, оскільки один сервер може обслуговувати кілька незалежних віртуальних систем. Також важливою перевагою є ізоляція середовищ — помилки або загрози в одній віртуальній машині не впливають на інші. Крім того, віртуалізація спрощує адміністрування, дозволяє швидко створювати резервні копії системи та знімки стану віртуальних машин, що значно полегшує відновлення після збоїв. У разі потреби можна також швидко масштабувати систему або перенести середовище на інше обладнання без переривання роботи [6].

У сучасних умовах окрему нішу займає контейнеризація — більш легка форма віртуалізації, яка не потребує створення повноцінної віртуальної машини. Контейнери працюють поверх однієї операційної системи, але залишаються ізольованими один від одного, що робить їх надзвичайно ефективними у середовищах з високими вимогами до масштабованості та швидкості розгортання. Найбільш відомими прикладами таких технологій є Docker, що забезпечує створення та керування контейнерами, та Kubernetes — система їх оркестрації.

Таким чином, технології віртуалізації та контейнеризації є важливими інструментами в сучасному IT-середовищі, зокрема в контексті побудови захищених та гнучких серверних рішень. Їх впровадження дозволяє оптимізувати інфраструктуру, зменшити витрати та підвищити рівень захисту й надійності інформаційних систем.

1.4. Визначення технічних та функціональних вимог

Перед розробкою будь-якої інформаційної системи одним із ключових етапів є формулювання вимог до неї. Це дозволяє точно окреслити сферу

застосування, визначити очікувану функціональність, рівень надійності, безпеки та інші характеристики, що безпосередньо впливають на кінцеву якість продукту. У контексті розробки серверної інфраструктури з підтримкою віртуалізації та засобів захисту ці вимоги повинні враховувати як функціональні, так і технічні аспекти [7].

Функціональні вимоги описують, які саме функції повинна виконувати система, як вона повинна реагувати на зовнішні впливи та які дії має забезпечувати. У даному проєкті система повинна відповідати таким основним функціональним вимогам:

1. Підтримка віртуалізації – система повинна забезпечувати створення, конфігурування, запуск, зупинку та видалення віртуальних машин або контейнерів.

2. Ізоляція середовищ – кожна віртуальна машина має бути ізольованою від інших, що гарантує безпеку та незалежність роботи служб.

3. Централізоване управління – адміністратор повинен мати змогу керувати віртуальними середовищами з єдиного інтерфейсу (графічного або веб-орієнтованого).

4. Моніторинг стану системи – система повинна забезпечувати візуалізацію навантаження на сервер, використання ресурсів та загальний стан віртуальних машин.

5. Журналювання дій – всі дії в системі повинні бути протоколовані з метою аудиту, відслідковування подій безпеки та відновлення після збоїв.

6. Керування правами доступу – реалізація багаторівневої моделі користувачів з різними ролями (адміністратор, користувач, гість).

7. Автоматичне резервне копіювання – наявність механізмів для регулярного створення резервних копій конфігурацій та даних з можливістю швидкого відновлення.

8. Гнучкість налаштувань – можливість налаштування ресурсів для кожної віртуальної машини (кількість ядер процесора, обсяг оперативної пам’яті, мережеві параметри тощо).

Технічні або нефункціональні вимоги описують якісні характеристики системи, що не пов’язані безпосередньо з її поведінкою, але мають критичне значення для її ефективної роботи. До основних технічних вимог належать:

1. Продуктивність – система повинна забезпечувати стабільну роботу при запуску декількох віртуальних машин одночасно. Очікуване навантаження: 5–10 віртуальних машин, що працюють паралельно без втрати швидкодії.

2. Надійність – у випадку збою або помилки система має гарантувати збереження даних та автоматичне відновлення критичних служб.

3. Безпека – система повинна мати механізми захисту від несанкціонованого доступу, у тому числі шифрування каналів зв’язку, двофакторну автентифікацію, захист від шкідливого ПЗ, фаєрволи, контроль прав доступу.

4. Масштабованість – архітектура повинна передбачати можливість додавання нових серверів або вузлів без необхідності зміни основної логіки системи.

5. Сумісність – система повинна підтримувати різні типи апаратного забезпечення та бути здатною працювати з поширеними операційними системами (Linux, Windows, BSD).

6. Інтерфейс користувача – інтерфейс повинен бути інтуїтивно зрозумілим, бажано із підтримкою багатомовності, з детальною документацією та інструкціями.

7. Автоматизація процесів – можливість використання сценаріїв (скриптів) для автоматичного розгортання віртуальних середовищ, оновлення системного ПЗ, встановлення патчів без ручного втручання.

8. Енергоефективність – система повинна мінімізувати споживання енергоресурсів шляхом оптимізації навантаження на апаратні компоненти.

Визначення перелічених вимог дозволяє перейти до етапів архітектурного проєктування, вибору технологій та подальшої розробки системи, орієнтованої на високу продуктивність, надійність і безпеку.

Дотримання встановлених вимог забезпечить відповідність очікуванням кінцевих користувачів і дозволить ефективно впровадити систему у реальне виробниче середовище.

1.5. Висновки щодо доцільності впровадження віртуалізації

У результаті проведеного аналізу сучасних підходів до побудови серверних систем, а також розгляду існуючих рішень щодо їх захисту, було встановлено, що впровадження віртуалізації є доцільним, обґрунтованим і стратегічно вигідним кроком у напрямі оптимізації обчислювальних ресурсів та підвищення ефективності IT-інфраструктури [8].

По-перше, віртуалізація забезпечує раціональне використання апаратних ресурсів, дозволяючи розміщувати кілька незалежних середовищ на одному фізичному сервері. Це значно знижує витрати на закупівлю додаткового обладнання, а також дозволяє ефективно масштабувати систему без значного збільшення апаратних потужностей.

По-друге, завдяки ізоляції середовищ, віртуалізація підвищує рівень безпеки: у разі компрометації однієї віртуальної машини інші залишаються недоступними для зловмисника. Це дозволяє впроваджувати політики сегментації, зменшувати поверхню атаки та забезпечувати гнучке керування ризиками.

По-третє, віртуалізаційні платформи підтримують високий рівень автоматизації, що спрощує адміністрування, скорочує час на налаштування нових серверів та знижує ризик людських помилок. Адміністратори можуть централізовано керувати всіма середовищами, застосовувати оновлення, моніторити ресурси та керувати резервним копіюванням.

Окрім цього, застосування віртуалізації забезпечує гнучкість та мобільність: віртуальні машини легко мігруються між фізичними серверами, що спрощує обслуговування, балансування навантаження та забезпечення безперервності бізнес-процесів.

З економічної точки зору, впровадження віртуалізації дозволяє зменшити експлуатаційні витрати, пов’язані з електроенергією, охолодженням серверів, фізичним розміщенням обладнання та технічним обслуговуванням [9].

Водночас, незважаючи на всі переваги, слід враховувати, що для ефективного впровадження віртуалізаційного середовища потрібен високий рівень кваліфікації персоналу, початкові інвестиції в ліцензії (у випадку комерційних продуктів) та правильний вибір технологічного стеку.

З огляду на вищезазначене, можна зробити висновок, що впровадження віртуалізації в серверну інфраструктуру є доцільним та актуальним кроком, який дозволяє досягти високої гнучкості, масштабованості, безпеки та економічної ефективності, що відповідає сучасним вимогам до інформаційних систем підприємств, установ та організацій.

РОЗДІЛ 2. ПРИЙНЯТІ ПРОЄКТНІ РІШЕННЯ

2.1. Вибір платформи для віртуалізації

Перед вибором платформи для віртуалізації необхідно здійснити всебічний аналіз вимог, які пред'являються до неї з урахуванням специфіки майбутньої інфраструктури, вимог до продуктивності, масштабованості, безпеки та зручності адміністрування. Ці вимоги умовно поділяються на функціональні, технічні та безпекові.

Функціональні вимоги включають базову та розширену функціональність, яку має підтримувати платформа віртуалізації. Серед основних функцій — підтримка створення та управління віртуальними машинами (ВМ), гнучке керування ресурсами (CPU, RAM, disk), автоматизоване балансування навантаження, інтеграція з системами зберігання даних, можливість кластеризації та високої доступності. Також важливою є підтримка знімків систем (snapshots), швидке розгортання нових ВМ, сумісність з різними ОС (Windows, Linux), а також інструменти для моніторингу та управління [10].

Технічні вимоги визначають апаратні та програмні характеристики, необхідні для стабільної та ефективної роботи віртуалізаційної платформи. Вони включають підтримку апаратної віртуалізації (Intel VT-x, AMD-V), сумісність із наявною серверною інфраструктурою, можливість масштабування як вертикально (додавання ресурсів в існуючу ВМ), так і горизонтально (додавання нових хостів до кластера), підтримку мережевої віртуалізації (віртуальні комутатори, VLAN), інтеграцію з системами резервного копіювання, а також невибагливість до типу операційної системи гіпервізора (bare-metal або host-based) [11].

Безпекові вимоги стосуються гарантування цілісності, конфіденційності та доступності даних у віртуалізованому середовищі. Платформа має підтримувати контроль доступу на основі ролей (RBAC),

журналювання подій, ізоляцію віртуальних машин одна від одної, шифрування з'єднань і зберігання даних, а також інтеграцію з існуючими системами безпеки (наприклад, Active Directory або LDAP). Важливо, щоб система мала сертифікацію відповідно до міжнародних стандартів безпеки (наприклад, ISO/IEC 27001, FIPS) [12].

Загалом, ретельний аналіз вищевказаних вимог є необхідним етапом для прийняття обґрунтованого рішення щодо вибору платформи, яка забезпечить стабільну, масштабовану та безпечну роботу віртуалізованої інфраструктури.

На ринку представлено кілька провідних платформ віртуалізації, кожна з яких має свої особливості, переваги та недоліки. До найпопулярніших рішень належать: VMware vSphere/ESXi, Microsoft Hyper-V, Proxmox VE, KVM (Kernel-based Virtual Machine) та Oracle VirtualBox. Розглянемо короткий огляд кожної з них [13].

VMware vSphere/ESXi — це комерційне рішення, яке давно зарекомендувало себе як одне з найпотужніших і найстабільніших у корпоративному середовищі. Воно підтримує велику кількість інструментів для управління, масштабування, кластеризації та резервного копіювання. Основними перевагами є висока продуктивність, гнучке управління ресурсами, розширена підтримка безпеки та інтеграція з хмарними платформами. Основним недоліком є висока вартість ліцензій [14].

Microsoft Hyper-V — гіпервізор від компанії Microsoft, який інтегрований у серверні ОС Windows Server. Він добре підходить для інфраструктур, що вже побудовані на базі рішень Microsoft. Серед переваг — простота розгортання в середовищі Windows, інтеграція з Active Directory, а також можливість використання у складі Azure Stack. Недоліком є обмежена підтримка ОС сторонніх виробників та дещо складніша настройка в порівнянні з конкурентами [15].

Proxmox VE — безкоштовна платформа з відкритим кодом, яка поєднує в собі KVM для повної віртуалізації та LXC для контейнеризації. Вона має

зручний веб-інтерфейс, можливість створення кластерів, резервного копіювання та відновлення систем. Proxmox VE популярний серед малого та середнього бізнесу завдяки простоті налаштування, активній спільноті та підтримці безкоштовних оновлень [16].

KVM (Kernel-based Virtual Machine) — гіпервізор, інтегрований у ядро Linux, який забезпечує ефективну віртуалізацію на базі відкритого коду. Завдяки своїй гнучкості, KVM широко використовується в дата-центрах та хмарних рішеннях, зокрема, у таких проектах як OpenStack. Його основними перевагами є висока швидкість, масштабованість та повна інтеграція з Linux. Проте налаштування вимагає певного рівня технічної обізнаності. [17]

Oracle VirtualBox — це універсальна та безкоштовна платформа, яка добре підходить для локального використання, тестування ПЗ або навчальних цілей. Вона підтримує різні операційні системи, має простий графічний інтерфейс і легко налаштовується. Однак її продуктивність та функціональність поступаються рішенням рівня підприємств [18].

Таким чином, кожна платформа має свої сильні сторони, і вибір залежить від конкретних потреб, бюджету, технічного рівня адміністраторів та вимог до масштабованості й безпеки. У наступному пункті буде проведено порівняльний аналіз згаданих рішень з урахуванням ключових параметрів.

Для прийняття обґрунтованого рішення щодо вибору платформи віртуалізації проведемо порівняння ключових рішень за основними критеріями: функціональні можливості, технічна продуктивність, вимоги до апаратного забезпечення, рівень безпеки, підтримка, вартість та зручність адміністрування.

Зважаючи на технічні та безпекові вимоги, а також орієнтацію проєкту на економічну доцільність і простоту масштабування, було прийнято рішення на користь використання Proxmox VE. Ця платформа є безкоштовною, базується на перевірених технологіях KVM та LXC, має зручний веб-інтерфейс та дозволяє легко створювати кластери. Крім того, вона має достатній рівень безпеки і активно підтримується спільнотою.

Критерій	VMware ESXi	Microsoft Hyper-V	Proxmox VE	KVM	Oracle VirtualBox
Тип ліцензії	Комерційна (є безкоштовна версія)	Входить до Windows Server	Вільна, з опційною підтримкою	Вільна	Вільна
Продуктивність	Висока	Висока	Висока	Висока	Середня
Масштабованість	Висока	Висока	Середня/висока	Висока	Низька
Підтримка кластеризації	Так	Так	Так	Так (через зовнішні засоби)	Ні
Безпека	Розширена	Висока	Висока	Залежить від налаштувань	Базова
Зручність адміністрування	Висока	Висока в середовищі Windows	Висока	Середня	Висока
Системні вимоги	Помірні	Середні	Помірні	Низькі–середні	Низькі
Підтримка ОС	Багато ОС	Переважно Windows	Багато ОС	Багато ОС	Багато ОС
Вартість	Висока (повна версія)	Включено у Windows Server	Безкоштовно	Безкоштовно	Безкоштовно

Proxmox VE дозволяє гнучко керувати віртуальними машинами, інтегрується з системами резервного копіювання, моніторингу та оновлень, що робить її оптимальним вибором для середовищ із обмеженим бюджетом, де все ж вимагається стабільність і функціональність на рівні підприємства.

Після детального аналізу функціональних, технічних та безпекових вимог до платформи віртуалізації, а також порівняння найпоширеніших рішень, було прийнято рішення обрати Proxmox Virtual Environment (Proxmox VE) як базову платформу для реалізації віртуалізованої системи. Такий вибір зумовлений низкою ключових переваг, що забезпечують ефективну, безпечну та гнучку роботу системи у реальних умовах [19].

По-перше, Proxmox VE є відкритим програмним забезпеченням, що дозволяє використовувати його без ліцензійних витрат. Це надзвичайно важливо для впровадження системи з обмеженим бюджетом або в рамках освітнього чи пілотного проєкту.

По-друге, платформа підтримує сучасні технології віртуалізації на основі KVM (для повної віртуалізації) та LXC (для контейнеризації), що забезпечує високу продуктивність і гнучкість при розгортанні різних типів навантажень. Завдяки цьому можна легко адаптувати систему до змін у структурі сервісів чи ресурсів.

По-третє, Proxmox має зручний веб-інтерфейс управління, який дозволяє адміністраторам ефективно контролювати і моніторити віртуальні машини, сховища, мережі та користувачів. Крім того, система підтримує створення кластерів, резервне копіювання, живу міграцію та інші функції корпоративного рівня.

По-четверте, платформа активно підтримується спільнотою та має регулярні оновлення, що позитивно впливає на безпеку та стабільність. Розгорнута система моніторингу та можливість інтеграції з інструментами сторонніх розробників дозволяє забезпечити дотримання політик безпеки.

Таким чином, вибір Proxmox VE як платформи для реалізації системи є обґрунтованим як з точки зору ефективності та функціональності, так і з погляду безпеки, підтримки масштабування та економічної доцільності.

2.2. Архітектура запропонованої віртуалізованої системи

Загальна схема побудови віртуального середовища базується на концепції централізованого керування обчислювальними, мережевими та сховищними ресурсами через платформу віртуалізації Proxmox VE. Віртуальне середовище реалізується у вигляді кластерної архітектури, що забезпечує масштабованість, відмовостійкість та централізований моніторинг усіх компонентів системи [20].

До складу віртуального середовища входять такі основні елементи:

1. Фізичні сервери (гіпервізори) – це базові апаратні платформи, на яких встановлено Proxmox VE. Кожен сервер може працювати як окремо, так і в складі кластеру. На них розгортаються віртуальні машини (VM) та контейнери (CT), що імітують роботу повноцінних серверів або служб.
2. Кластер управління – об’єднання фізичних вузлів у єдине віртуальне середовище з можливістю спільного керування ресурсами, автоматичного балансування навантаження, міграції ВМ між вузлами без зупинки сервісів (live migration) та централізованого резервного копіювання.
3. Мережна інфраструктура – побудована з урахуванням розділення трафіку на адміністративний, користувацький та реплікаційний. Використовується віртуальні мережі (vLAN) для підвищення безпеки та ізоляції віртуальних середовищ.
4. Системи зберігання даних (Storage) – можуть бути локальними або мережевими (наприклад, NFS, iSCSI, Ceph), що забезпечує гнучкість у розміщенні віртуальних машин, резервних копій та образів. У випадку кластерного розгортання доцільно використовувати розподілене зберігання, наприклад, Ceph або GlusterFS, для досягнення високої доступності.

5. Служби моніторингу та журналювання – інтеграція з такими інструментами, як Zabbix, Grafana або Prometheus, дозволяє здійснювати постійний контроль за станом системи, її навантаженням та виявляти аномальні ситуації.

6. Інтерфейс керування – зручний веб-інтерфейс Proxmox, що надає доступ до основних функцій: створення/видалення ВМ, моніторинг, міграція, резервне копіювання, керування користувачами та правами доступу.

Узагальнюючи, така архітектура забезпечує гнучке масштабування, простоту в адмініструванні, високу доступність і безперервність бізнес-процесів, що є критично важливим для сучасних інформаційних систем.

Інфраструктура віртуалізованого середовища складається з кількох ключових компонентів, кожен із яких виконує окрему роль у забезпеченні функціонування всієї системи. Нижче розглянуто основні компоненти інфраструктури та особливості їх взаємодії [21]:

1. Гіпервізор (Hypervisor)

Це основа віртуалізованого середовища, яка відповідає за створення, запуск і керування віртуальними машинами. У рамках проєкту було обрано Proxmox VE як платформу віртуалізації, що поєднує в собі можливості гіпервізора KVM (Kernel-based Virtual Machine) для повноцінної віртуалізації та LXC (Linux Containers) для контейнеризації.

Основні функції гіпервізора:

- розподіл апаратних ресурсів між віртуальними машинами;
- забезпечення ізоляції процесів;
- підтримка "live migration" та кластеризації;
- керування резервним копіюванням та відновленням.

2. Віртуальні машини (VM)

Віртуальні машини – це логічні екземпляри операційних систем, що функціонують незалежно одна від одної на базі гіпервізора. Кожна VM має власний набір ресурсів: CPU, пам'ять, мережевий інтерфейс та дисковий простір.

Віртуальні машини в середовищі Proxmox використовуються для розгортання серверних служб, прикладного ПЗ або емуляції фізичних серверів у тестових середовищах.

Переваги:

- гнучкість у масштабуванні;
- швидке розгортання;
- можливість клонування;
- легкість у резервуванні.

3. Мережева інфраструктура

Мережевий рівень відповідає за з'єднання між віртуальними машинами, фізичними хостами та зовнішніми мережами.

У структурі використовуються:

- bridge-інтерфейси для підключення VM до фізичної мережі;
- vLAN для логічної сегментації трафіку (наприклад, відокремлення адміністративного трафіку від користувацького);
- Firewall та NAT-рішення для контролю доступу та підвищення безпеки.

4. Системи зберігання даних (Storage)

Сховище відіграє важливу роль у зберіганні образів дисків віртуальних машин, шаблонів, ISO-файлів і резервних копій.

Використовуються:

- локальні SSD або HDD для базових потреб (зберігання критично важливих ВМ);
- мережеві сховища (NFS, iSCSI) – забезпечують централізований доступ до даних;
- розподілені файлові системи (Ceph, GlusterFS) – для побудови високодоступного кластера.

Гіпервізор керує всіма віртуальними машинами та координує доступ до фізичних ресурсів. Мережеві модулі забезпечують комунікацію між ВМ і зовнішнім світом. Сховище працює як централізоване джерело даних для всіх

ВМ. Усе це разом забезпечує гнучкість, масштабованість, безпечність і стійкість віртуалізованої системи.

У віртуалізованому середовищі компоненти інфраструктури працюють як єдина система, забезпечуючи стабільність, ефективність та безпеку всього середовища. Взаємодія між основними компонентами — гіпервізором, віртуальними машинами (VM), мережею та сховищем — є критично важливою для забезпечення безперебійної роботи [22].

Гіпервізор відіграє ключову роль у забезпеченні зв'язку між фізичним апаратним забезпеченням і віртуальними машинами. Він виконує функцію посередника, надаючи віртуальним машинам доступ до фізичних ресурсів, таких як процесор, оперативна пам'ять та дискові простори. Гіпервізор не тільки ізолює віртуальні машини одну від одної, що підвищує рівень безпеки, але й здійснює управління їхнім запуском, зупинкою та міграцією. Це дозволяє, зокрема, ефективно використовувати ресурси на всіх етапах роботи системи.

Крім того, гіпервізор забезпечує тісну взаємодію з мережею. Віртуальні машини підключаються до мережі через віртуальні мережеві інтерфейси, які контролюються саме гіпервізором. Це дозволяє створювати віртуальні комутатори, здійснювати маршрутизацію трафіку, а також налаштовувати політики безпеки, наприклад, через фаєрволи або VPN. Завдяки цьому, кожна віртуальна машина може функціонувати як незалежний елемент мережі, маючи доступ до зовнішніх ресурсів, одночасно підтримуючи внутрішні комунікації серед компонентів системи.

Віртуальні машини, в свою чергу, взаємодіють із сховищем даних через гіпервізор, який керує процесом доступу до зберігання даних, розміщених на фізичних або мережевих пристроях. Віртуальні машини можуть звертатися до своїх віртуальних дисків для зчитування та запису даних, а також для взаємодії з іншими сховищами, що є критичним для ефективного зберігання та резервування даних. Вибір типу сховища, будь то локальне чи мережеве, безпосередньо впливає на продуктивність та швидкість доступу до інформації,

що є важливим для досягнення оптимальних результатів роботи віртуалізованої системи [23].

Мережа та сховище також мають тісні зв’язки, коли мова йде про мережеві сховища, такі як NAS або iSCSI. У таких випадках дані передаються між компонентами через спеціалізовані мережеві інтерфейси, що забезпечує централізоване зберігання та доступ до даних з різних точок системи. Проте для ефективного виконання цієї функції важливо налаштувати окремі канали для передачі даних між сховищем і іншими частинами системи, щоб уникнути перевантаження основних мережевих каналів.

Усі ці взаємозв’язки створюють цілісну інфраструктуру, де кожен компонент має свою роль і взаємодіє з іншими для забезпечення стабільності, безпеки та продуктивності всієї віртуалізованої системи. Чітко налаштовані зв’язки між гіпервізором, віртуальними машинами, мережею та сховищем дозволяють досягти високої ефективності роботи, зокрема, у розподілених та масштабованих середовищах.

У віртуалізованих середовищах для забезпечення безперервної роботи системи та підтримки високої доступності, критично важливими є механізми резервування, масштабування та моніторингу. Ці механізми сприяють надійності, гнучкості та контролю за інфраструктурою, а також забезпечують її стійкість до збоїв і ефективне використання доступних ресурсів.

Резервування є одним з основних способів забезпечення безперебійної роботи. У віртуалізованих системах резервування охоплює кілька напрямків: резервування віртуальних машин, мережі та сховищ. Віртуальні машини можуть бути розподілені між кількома фізичними серверами, що дозволяє автоматично мігрувати віртуальні машини в разі відмови одного з хостів. Така міграція забезпечує збереження доступності сервісів і даних без перерви в роботі. Окрім цього, резервування мережі важливе для збереження зв'язку між компонентами системи. Завдяки дублюванню мережевих каналів та інтерфейсів, у разі збоїв система автоматично переключається на резервний шлях без втрат у доступності. Не менш важливим є резервування даних, де

створюються копії на мережевих сховищах для запобігання втратам інформації під час збоїв у локальних сховищах.

Масштабування є ще одним важливим механізмом для адаптації віртуалізованої інфраструктури до змінних умов навантаження. Масштабування може бути як вертикальним, так і горизонтальним. Вертикальне масштабування передбачає збільшення або зменшення ресурсів (наприклад, оперативної пам’яті або процесорного часу) для окремих віртуальних машин. Це дозволяє ефективно використовувати ресурси та адаптувати систему до поточних потреб без необхідності додавати нові фізичні сервери. Горизонтальне масштабування полягає в додаванні нових віртуальних машин або хостів для розподілу навантаження. Такий підхід сприяє збільшенню продуктивності та знижує ризик виникнення перевантажень [24].

Моніторинг віртуалізованого середовища є необхідним для своєчасного виявлення та усунення проблем. Під час моніторингу важливо відстежувати стан ресурсів, таких як процесор, пам’ять, дисковий простір та мережеві канали, щоб своєчасно виявити перевантаження та забезпечити оптимальний розподіл ресурсів серед віртуальних машин. Крім того, моніторинг стану самих віртуальних машин і хостів дозволяє виявляти апаратні або програмні збої, що можуть вплинути на роботу системи, і приймати необхідні заходи для їх усунення. Окремим аспектом моніторингу є безпека, де здійснюється контроль за наявністю несанкціонованих спроб доступу або атак. Інструменти моніторингу аналізують журнали подій і мережевий трафік для виявлення потенційних загроз і вчасного реагування на них.

Таким чином, механізми резервування, масштабування та моніторингу в комплексі дозволяють забезпечити безперебійну, ефективну та безпечну роботу віртуалізованої інфраструктури. Вони сприяють оптимальному використанню ресурсів, високій доступності та безпеці всієї системи, що є

критичним для забезпечення стабільної роботи в умовах змінних навантажень і можливих збоїв.

2.3. UML-діаграми

Університет КПІ має складну і багатофункціональну серверну інфраструктуру, яка підтримує широкий спектр діяльності: від навчальних процесів до наукових досліджень і адміністративної роботи. Для забезпечення стабільної роботи цієї інфраструктури важливо правильно визначити основні сценарії взаємодії користувачів з її компонентами. Одним з важливих етапів розробки віртуалізованої системи є визначення ролей користувачів і їхніх взаємодій з системою через Use Case-діаграму, яка чітко демонструє основні функціональні сценарії використання серверної інфраструктури.

У загальній характеристиці інфраструктури університету можна виділити кілька ключових користувачів, які взаємодіють із серверною інфраструктурою. Це, в першу чергу, системні адміністратори, користувачі (студенти, викладачі, науковці) та адміністратори безпеки.

Системні адміністратори мають повний доступ до всіх компонентів інфраструктури. Вони відповідають за налаштування серверів, управління віртуальними машинами, адміністрування сховищ та мереж. Вони створюють нові віртуальні машини для різних служб, налаштовують ресурси та забезпечують стабільну роботу усієї інфраструктури [25].

Користувачі (студенти, викладачі та науковці) взаємодіють з серверною інфраструктурою через конкретні сервіси, наприклад, системи для онлайн-навчання, платформи для наукових досліджень, зберігання даних і доступ до спеціалізованого програмного забезпечення. Для них важливо забезпечити стабільний доступ до необхідних ресурсів з можливістю швидкого відновлення в разі збоїв.

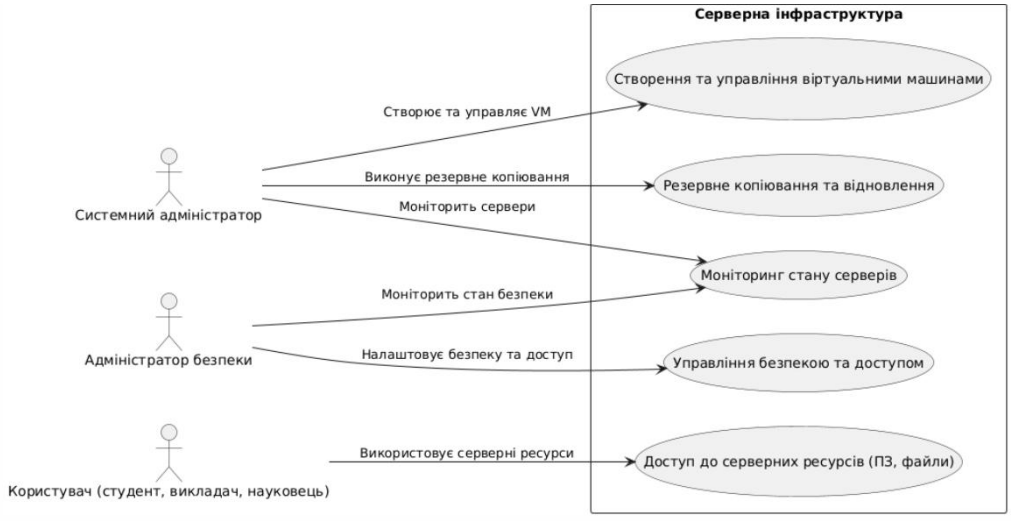
Адміністратори безпеки займаються моніторингом безпеки серверної інфраструктури, контролюючи доступ до ресурсів, впроваджуючи політики

безпеки і відповідаючи за захист даних. Вони здійснюють управління правами доступу, виявлення загроз, а також виконують резервне копіювання і відновлення даних.

Основними сценаріями взаємодії користувачів з серверною інфраструктурою є: створення і управління віртуальними машинами, доступ до серверних ресурсів для виконання навчальних та наукових завдань, моніторинг стану серверів і їхніх компонентів, налаштування доступу та безпеки.

Use Case-діаграма у даному випадку відображає ці взаємодії. Вона показує, які операції можуть виконувати різні типи користувачів, наприклад, адміністратор може створювати і налаштовувати віртуальні машини, науковець або студент може отримувати доступ до програмного забезпечення через віртуальне середовище, а адміністратор безпеки може стежити за станом безпеки та резервним копіюванням [26].

Завдяки Use Case-діаграмі можна зрозуміти, як кожна роль користувача взаємодіє з інфраструктурою, які саме функції системи є критичними для нормальної роботи університету, а також як ці взаємодії можна автоматизувати і оптимізувати в процесі віртуалізації.



У цій діаграмі представлені основні взаємодії користувачів різних ролей із серверною інфраструктурою університету. Взаємодія з

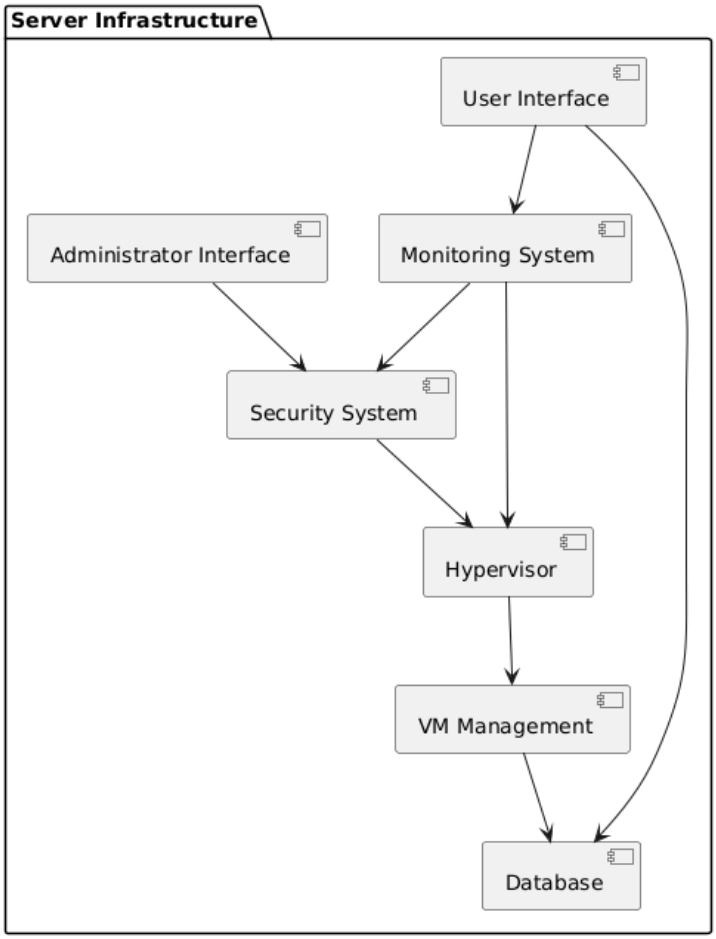
інфраструктурою здійснюється трьома основними типами користувачів: системним адміністратором, користувачем (студентом, викладачем, науковцем) та адміністратором безпеки. Кожен з них виконує певні функції у системі, що забезпечують стабільну та безпечну роботу серверів.

Системний адміністратор має ключову роль у підтримці серверної інфраструктури. Він відповідальний за створення та управління віртуальними машинами (VM). Ця функція є основою для віртуалізації інфраструктури університету, яка дозволяє ефективно використовувати серверні ресурси для різних потреб. Адміністратор також займається моніторингом стану серверів, що дозволяє йому виявляти неполадки та здійснювати заходи для їх усунення, щоб забезпечити безперервну роботу інфраструктури. У разі необхідності він виконує резервне копіювання та відновлення даних, що гарантує захист інформації і відновлення роботи в разі технічних збоїв.

Користувачі системи, такі як студенти, викладачі та науковці, мають доступ до серверних ресурсів. Вони використовують сервери для зберігання файлів, доступу до програмного забезпечення та інших необхідних сервісів для навчальних та наукових цілей. Їхня взаємодія з інфраструктурою обмежується в основному використанням цих ресурсів, без втручання в адміністрування системи [27].

Адміністратор безпеки відповідальний за забезпечення безпеки інфраструктури. Він налаштовує доступ до віртуальних машин та даних, контролюючи, хто може використовувати серверні ресурси, і запобігаючи несанкціонованому доступу. Також адміністратор безпеки здійснює моніторинг стану безпеки системи, виявляючи загрози і вживаючи заходів для їх усунення, що забезпечує належний рівень захисту даних та стабільність інфраструктури.

Загалом, ця Use Case-діаграма надає зрозуміле уявлення про основні функції та взаємодії користувачів із серверною інфраструктурою університету, що дозволяє краще розуміти організацію роботи та забезпечення безпеки та ефективності віртуалізованої інфраструктури.



Основним компонентом віртуалізованої інфраструктури є гіпервізор (Hypervisor). Він відповідає за керування віртуальними машинами (VM) та забезпечує абстракцію апаратного забезпечення, дозволяючи запускати віртуальні машини на фізичних серверах. Гіпервізор дозволяє ефективно використовувати апаратні ресурси, забезпечуючи ізоляцію між віртуальними машинами та їхній незалежний функціонування.

Для управління віртуальними машинами створено окремий компонент — Управління Віртуальними Машинами (VM Management). Цей компонент відповідає за створення, налаштування та управління віртуальними машинами. Він взаємодіє з гіпервізором і дозволяє адміністраторам контролювати ресурси, які виділяються кожній віртуальній машині, такі як процесор, пам'ять і сховище [28].

Всі необхідні дані про віртуальні машини, їхні параметри, статуси та іншу інформацію зберігаються в базі даних (Database). База даних є ключовим

компонентом інфраструктури, який забезпечує збереження та швидкий доступ до всієї інформації, що стосується віртуалізованих систем.

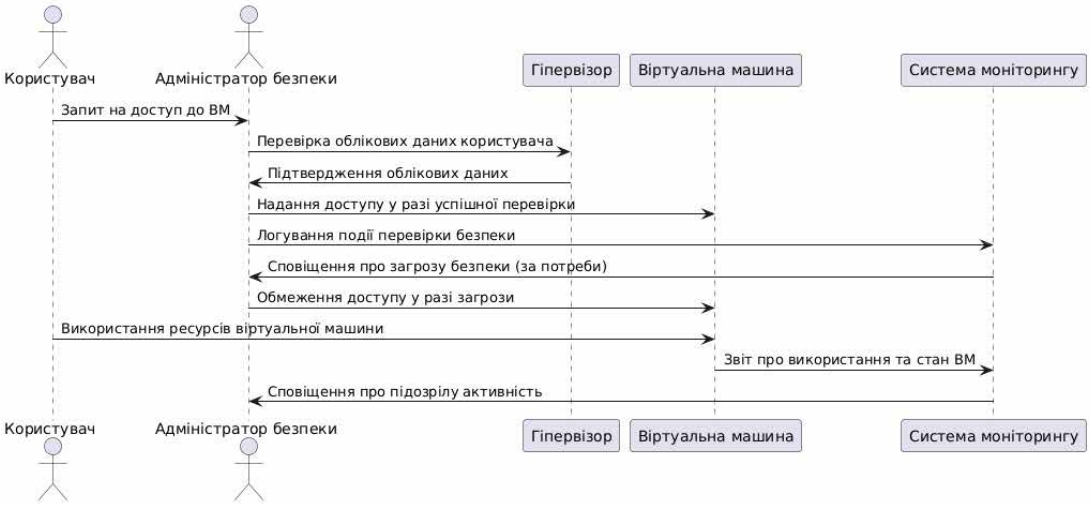
Для забезпечення стабільної роботи всієї інфраструктури важливу роль відіграє система моніторингу (Monitoring System). Цей компонент відповідає за моніторинг стану серверів, віртуальних машин та інших компонентів інфраструктури. Він зібирає дані про стан системи та повідомляє про потенційні збої або несправності, що дозволяє адміністратору оперативно реагувати на проблеми.

Система безпеки (Security System) реалізує всі політики безпеки, включаючи контроль доступу, шифрування даних і виявлення загроз. Вона інтегрується з гіпервізором для забезпечення безпеки віртуальних машин та інших компонентів, що запобігає несанкціонованому доступу до системи.

Для адміністраторів системи створений інтерфейс адміністратора (Administrator Interface), який дозволяє здійснювати управління інфраструктурою, налаштовувати віртуальні машини, перевіряти стан безпеки, моніторити систему та виконувати резервне копіювання даних.

Інтерфейс користувача (User Interface) є важливим компонентом для студентів, викладачів і науковців університету, оскільки він надає доступ до серверних ресурсів, таких як програмне забезпечення та сховища даних. Користувачі можуть виконувати завдання, отримувати доступ до необхідних ресурсів та зберігати свої файли [29].

Взаємодія між усіма компонентами організована таким чином, що адміністративні та безпекові функції інтегруються через відповідні інтерфейси з системами моніторингу, базою даних та гіпервізором. Це дозволяє забезпечити ефективне управління всіма аспектами інфраструктури та надійний контроль над її функціонуванням, що є важливим для забезпечення безперебійної роботи віртуалізованої інфраструктури університету.



Діаграма послідовностей зображує основні етапи взаємодії між користувачем, адміністратором безпеки, гіпервізором, віртуальними машинами та системою моніторингу в контексті безпекових сценаріїв віртуалізованої інфраструктури університету.

Процес починається з того, що Користувач ініціює запит на доступ до віртуальної машини (ВМ). Відповідно, Адміністратор безпеки передає цей запит на перевірку до Гіпервізора, який здійснює перевірку облікових даних користувача.

У разі успішної перевірки, Гіпервізор підтверджує правильність даних і Адміністратор безпеки надає доступ користувачеві до віртуальної машини. Усі ці події записуються в Систему моніторингу, що дозволяє відслідковувати безпекові дії в реальному часі [30].

Паралельно, система моніторингу активно слідкує за станом віртуальних машин і за допомогою аналітики та алгоритмів виявляє можливі загрози. У разі виявлення підозрілої активності, система негайно сповіщає адміністратора про потенційну загрозу.

У разі виявлення серйозної загрози безпеки, Адміністратор безпеки може обмежити доступ до віртуальної машини, щоб запобігти поширенню потенційної загрози. Після цього Користувач може використовувати ресурси віртуальної машини, у той час як Система моніторингу продовжує

здійснювати спостереження за активністю, фіксуючи всі звіти про використання ВМ та повідомляючи адміністратора у разі появи нових підозр.

Ця діаграма дозволяє детально відобразити основні етапи роботи з безпековими сценаріями, де кожен крок взаємодії ретельно контролюється та моніториться для запобігання несанкціонованому доступу та інших загроз.

2.4. Сценарії використання та модель загроз

Типові сценарії використання системи віртуалізації для серверної інфраструктури університету КПІ включають кілька основних процесів, які є критичними для забезпечення її стабільної роботи. Ці процеси включають адміністрування, оновлення та ізоляцію середовищ. Кожен з цих аспектів відіграє ключову роль у підтримці безпеки, ефективності та надійності всієї системи.

Адміністрування є фундаментальним процесом, без якого неможливо ефективно управляти серверною інфраструктурою. Воно передбачає комплексну діяльність, що включає налаштування, моніторинг, управління ресурсами та безпекою.

Перш за все, адміністрування передбачає управління віртуальними машинами. Системний адміністратор відповідає за створення, налаштування та конфігурацію віртуальних машин (VM). Це дозволяє ефективно використовувати фізичні сервери для розміщення численних віртуальних середовищ, які можуть бути використані різними користувачами та під різні потреби. Віртуалізація надає можливість гнучко розподіляти ресурси — процесор, пам'ять, дисковий простір — між різними віртуальними машинами, що дає змогу оптимізувати використання апаратного забезпечення [31].

Крім того, адміністрування включає постійний моніторинг інфраструктури. Адміністратор використовує спеціалізовані системи моніторингу для відстеження стану серверів і віртуальних машин. Це дозволяє оперативно виявляти проблеми, такі як перевантаження ресурсів, збої в роботі

окремих компонентів або навіть загрози безпеці. У разі виявлення несправностей адміністратор може негайно вжити заходів для їх усунення, наприклад, перевірити лог-файли, налаштувати параметри або перезапустити систему.

Окрім цього, адміністратор відповідає за контроль доступу до віртуальних машин. Важливо налаштувати рівні доступу, що дозволяють певним користувачам або групам користувачів взаємодіяти з конкретними ресурсами, в той час як інші ресурси залишаються для них недоступними. Це дозволяє забезпечити безпеку та уникнути несанкціонованого доступу до конфіденційної інформації.

Оновлення є важливим аспектом адміністрування системи, оскільки технології віртуалізації та інші компоненти інфраструктури постійно еволюціонують. Без своєчасних оновлень система може стати вразливою до атак і відставати від сучасних стандартів.

Процес оновлення передбачає оновлення гіпервізора, який є основою віртуалізованої інфраструктури. Гіпервізор керує запуском і розподілом ресурсів між віртуальними машинами, тому його оновлення має вирішальне значення для безпеки та стабільності всієї системи. Оновлення гіпервізора включає виправлення вразливостей, оптимізацію продуктивності та підтримку новітніх версій операційних систем, які працюють на віртуальних машинах.

Окрім цього, необхідно постійно оновлювати віртуальні машини, щоб підтримувати їх сумісність із новими версіями операційних систем і додатків. Оновлення програмного забезпечення на рівні віртуальних машин також важливе для забезпечення їх ефективності та безпеки. Важливо зазначити, що оновлення має бути ретельно спланованим процесом, щоб уникнути простоїв або збоїв у роботі системи [32].

Також необхідно оновлювати системи безпеки, зокрема антивірусні програми, брандмауери та інші інструменти, які забезпечують захист від загроз. Оновлення баз даних загроз і патчів має бути постійним, оскільки нові вразливості постійно з'являються в результаті розвитку технологій.

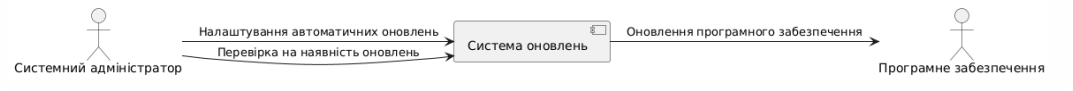
Автоматизація процесу оновлення може значно спростити цей процес і зменшити ймовірність помилок. Це дозволяє забезпечити своєчасне оновлення компонентів без необхідності залучати адміністратора для кожного оновлення.

Ізоляція середовищ є критично важливою для забезпечення безпеки та стабільності роботи системи. Оскільки віртуалізація дозволяє одночасно запускати кілька віртуальних машин на одному фізичному сервері, важливо гарантувати, що одна віртуальна машина не впливає на роботу іншої.

Ізоляція між віртуальними машинами забезпечується через механізми, які дозволяють кожній машині працювати в окремому середовищі, не впливаючи на інші. Це дозволяє уникнути ситуацій, коли збої або атаки на одну віртуальну машину можуть вплинути на інші, що є критично важливим для забезпечення безпеки всієї інфраструктури.

Ізоляція також стосується ізоляції користувачів і проектів. Університет може мати різні групи користувачів, які мають доступ до різних віртуальних середовищ. Наприклад, студенти, викладачі та науковці можуть працювати в різних віртуальних машинах або сегментах мережі, що дозволяє зберегти конфіденційність даних і запобігти несанкціонованому доступу [33].

На додаток до ізоляції на рівні віртуальних машин, важливим аспектом є ізоляція мережі. Університет може створювати віртуальні мережі, які забезпечують додаткову ізоляцію між різними групами користувачів або середовищами. Це дозволяє забезпечити більш високий рівень безпеки, обмежуючи доступ до важливих ресурсів лише для авторизованих користувачів або груп.



Ідентифікація потенційних загроз є важливим етапом у забезпеченні безпеки серверної інфраструктури університету КПІ. Як і в будь-якій іншій системі, віртуалізація та сучасні технології створюють нові можливості для загроз. Визначення та аналіз цих загроз є основою для створення ефективних

стратегій безпеки, які захищають дані і підтримують стабільну роботу інфраструктури.

Загрози можна умовно поділити на дві великі категорії: зовнішні та внутрішні.

Зовнішні загрози походять від атак ззовні системи, тобто з інших мереж або інтернету. Однією з найбільш серйозних зовнішніх загроз є атака типу DDoS (Distributed Denial of Service). Така атака спрямована на перевантаження серверів або віртуальних машин великою кількістю запитів, що може призвести до тимчасової недоступності ресурсів. Відповідно, важливо мати захист від таких атак, який дозволяє розподіляти навантаження або фільтрувати підозрілі запити. Іншою значною зовнішньою загрозою є зломи через вразливості в програмному забезпеченні. У цьому випадку зловмисники можуть скористатися недоліками в гіпервізорі, операційних системах або додатках, щоб отримати несанкціонований доступ до даних або навіть повністю контролювати систему. Крім того, соціальна інженерія та фішинг також є зовнішніми загрозами, що мають на меті обманути співробітників університету для отримання конфіденційних даних, наприклад, паролів або інформації про систему. Такі атаки можуть бути особливо небезпечними, якщо не вживаються заходи для навчання персоналу та підвищення їхньої обізнаності про можливі небезпеки. Шкідливе програмне забезпечення (Malware), яке потрапляє в систему через фішингові листи або вразливості в програмному забезпеченні, може серйозно пошкодити систему або призвести до викрадення даних. І, зрештою, загроза несанкціонованого доступу через мережеві вразливості може призвести до того, що зловмисники отримають доступ до системи через неправильно налаштовані мережеві компоненти, такі як маршрутизатори, брандмауери або інші пристрої [34].

Внутрішні загрози виникають від співробітників або користувачів, які мають легітимний доступ до системи, але можуть зловживати своїм доступом. Одна з основних внутрішніх загроз — це несанкціонований доступ до чутливих даних або систем. Коли співробітники, що працюють з віртуальними

машинами або іншими сервісами інфраструктури, отримують доступ до важливих даних, вони можуть використати цей доступ для особистих цілей або випадково викликати порушення роботи системи. Така ситуація може бути особливо небезпечною, якщо в системі є слабкі місця в управлінні правами доступу або неефективні засоби моніторингу та контролю за діяльністю користувачів.

Неправильне налаштування системи або помилки адміністраторів — ще одна внутрішня загроза. Коли адміністратори не уважно налаштовують системи безпеки, вони можуть створити вразливості, через які зловмисники зможуть отримати доступ до ресурсів або даних. Наприклад, помилки в налаштуванні прав доступу до віртуальних машин або баз даних можуть дозволити несанкціонованим користувачам отримати доступ до важливої інформації. Інколи адміністраторами може бути допущено й помилки, пов'язані з неправильним резервним копіюванням чи відновленням даних, що, в свою чергу, може призвести до втрати важливої інформації.

Внутрішні загрози також можуть включати зловживання правами доступу, коли співробітники системи намагаються використовувати свої повноваження для особистих вигод. Наприклад, зловмисники можуть намагатися отримати доступ до конфіденційних даних, викрадати інтелектуальну власність або змінювати налаштування системи для власних цілей [35].

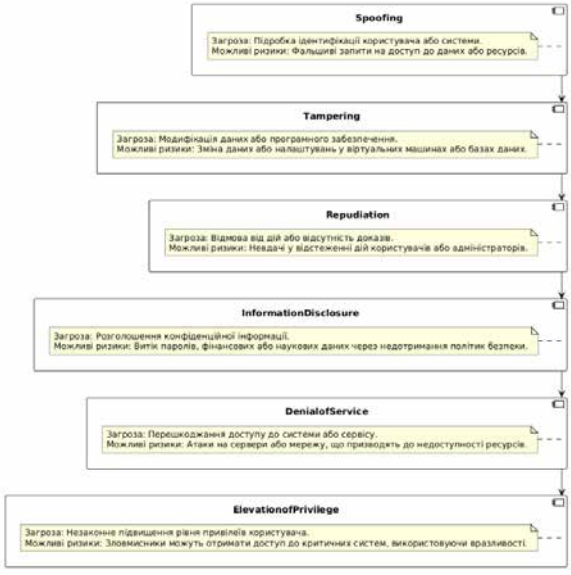
Ще одним аспектом внутрішніх загроз є ненавмисна втрата даних. Порушення правил роботи з даними, недбалість при обробці чи зберіганні важливих файлів може призвести до їх втрати. Проте таку ситуацію можна запобігти, запровадивши ефективні процеси для резервного копіювання та відновлення даних.

Важливим аспектом є внутрішня шпигунська діяльність, коли співробітники або користувачі можуть спробувати викрасти чутливу інформацію для особистих чи комерційних цілей. Це може стосуватися як

внутрішньої інформації університету, так і наукових результатів чи іншої цінної інтелектуальної власності.

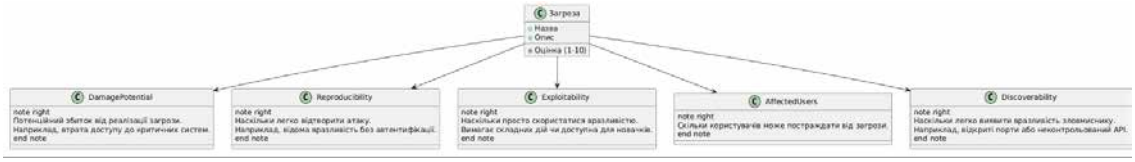
Для ефективного аналізу ризиків, що можуть виникнути в результаті загроз для серверної інфраструктури університету КПІ, необхідно використовувати спеціалізовані моделі для оцінки потенційних небезпек. Одними з таких інструментів є моделі STRIDE та DREAD, які дозволяють систематично виявляти загрози та оцінювати їхній вплив на безпеку та стабільність системи.

Модель STRIDE дає змогу класифікувати загрози за шістьма основними категоріями: підробка (Spoofing), маніпулювання даними (Tampering), заперечення (Repudiation), розкриття інформації (Information Disclosure), відмова в обслуговуванні (Denial of Service) та підвищення привілеїв (Elevation of Privilege). Цей підхід дозволяє системно розглядати можливі загрози в контексті їхнього типу, допомагаючи визначити, які саме аспекти інфраструктури потребують найбільшої уваги. Наприклад, під загрозою підробки може йтися про спроби несанкціонованого доступу до віртуальних машин чи інших важливих даних, а під маніпулюванням — це атаки, що спричиняють зміни в налаштуваннях системи або її даних.



Модель DREAD дозволяє оцінити рівень загрози через п'ять критеріїв: потенціал шкоди (Damage Potential), відтворюваність (Reproducibility),

зловмисність (Exploitability), кількість постраждалих користувачів (Affected Users) та виявленість (Discoverability). Застосування цієї моделі дає змогу оцінити не лише ймовірність виникнення загрози, а й визначити її потенційний вплив на всю систему. Наприклад, висока оцінка за потенціалом шкоди може вказувати на серйозні наслідки для сервісів, якщо загроза реалізується, а висока відтворюваність свідчить про те, що таку атаку буде неважко повторити [36].



Використання цих двох моделей дозволяє виявити найкритичніші загрози для інфраструктури, наприклад, ті, що стосуються конфіденційності даних або безперервності роботи серверів. Це дає можливість зосередитися на запобіганні таких інцидентів і ефективно реагувати на них, забезпечуючи стабільність та безпеку інформаційних систем університету.

Мінімізація ризиків є важливим етапом у забезпеченні безпеки серверної інфраструктури університету КПІ. Оскільки інфраструктура містить чутливі дані, а також забезпечує критично важливі сервіси для користувачів, важливо вжити ефективних заходів для зменшення ймовірності виникнення загроз та зниження їхнього впливу на систему.

1. Зміцнення політик доступу та автентифікації

Один із головних способів мінімізації ризиків — це забезпечення належної автентифікації та контролю доступу до віртуальних машин та серверів. Використання багатофакторної автентифікації (MFA), сильних паролів і обмеження доступу лише до необхідних ресурсів для кожного користувача дозволяє значно знизити ймовірність несанкціонованого доступу. Важливо також регулярно змінювати паролі і застосовувати політики захисту облікових записів для мінімізації можливості підбору паролів.

2. Шифрування даних

Для захисту даних, що зберігаються або передаються між серверними системами, важливо використовувати методи шифрування. Шифрування даних на рівні файлової системи або у вигляді SSL/TLS-шифрування для передаваної інформації дозволяє зменшити ризики їхнього несанкціонованого доступу чи витоку під час атаки. Шифрування також допомагає захистити конфіденційну інформацію, таку як особисті дані студентів та викладачів.

3. Актуалізація та патчинг програмного забезпечення

Регулярне оновлення та патчинг операційних систем, віртуалізаційних платформ і всіх використовуваних програмних продуктів є ключовим аспектом захисту від експлуатації відомих вразливостей. Використання автоматичних оновлень, налаштування регулярних перевірок на наявність вразливих компонентів допомагає підтримувати систему в актуальному стані та знижує можливості для зловмисників використати експлойти.

4. Резервне копіювання та відновлення

Для зниження ризиків втрати даних через збої, атаки або фізичні пошкодження серверів необхідно забезпечити регулярне резервне копіювання всіх критичних даних. Це включає в себе як файли, так і конфігураційні налаштування віртуальних машин. Важливо також перевіряти працездатність системи відновлення даних, щоб у разі необхідності відновлення інформації не виникало затримок чи проблем.

5. Моніторинг та виявлення загроз

Важливим елементом мінімізації ризиків є налаштування систем моніторингу, які відслідковують роботу серверів та віртуальних машин в реальному часі. Система повинна здатна виявляти підозрілі активності, такі як спроби несанкціонованого доступу, незвичні патерни використання ресурсів або аномальні запити. Вчасне виявлення загроз дозволяє оперативно вжити заходів для зупинення атаки.

6. Ізоляція середовищ та віртуальних машин

Ще одним заходом мінімізації ризиків є ізоляція середовищ та віртуальних машин. Це дозволяє обмежити доступ до критичних систем і

запобігти розповсюдженню атаки між віртуальними машинами. Використання віртуальних мереж, брандмауерів та віртуальних сегментів забезпечує більшу безпеку і запобігає можливості поширення інфекцій чи шкідливих програм.

7. Регулярні перевірки та аудит безпеки

Оскільки загрози можуть змінюватися з часом, важливо проводити регулярні перевірки та аудит безпеки на всіх рівнях інфраструктури. Це включає в себе перевірки налаштувань безпеки, аналіз журналів подій, перевірку працездатності механізмів захисту. В результаті таких перевірок можуть бути виявлені потенційно небезпечні вразливості, які потребують негайного усунення [37].

Загалом, ефективні заходи з мінімізації ризиків мають на меті не лише запобігти потенційним загрозам, але й зменшити їхній вплив на систему, забезпечити стабільну роботу інфраструктури і зберегти цілісність і конфіденційність даних.

2.5. Розробка політик безпеки в середовищі віртуалізації

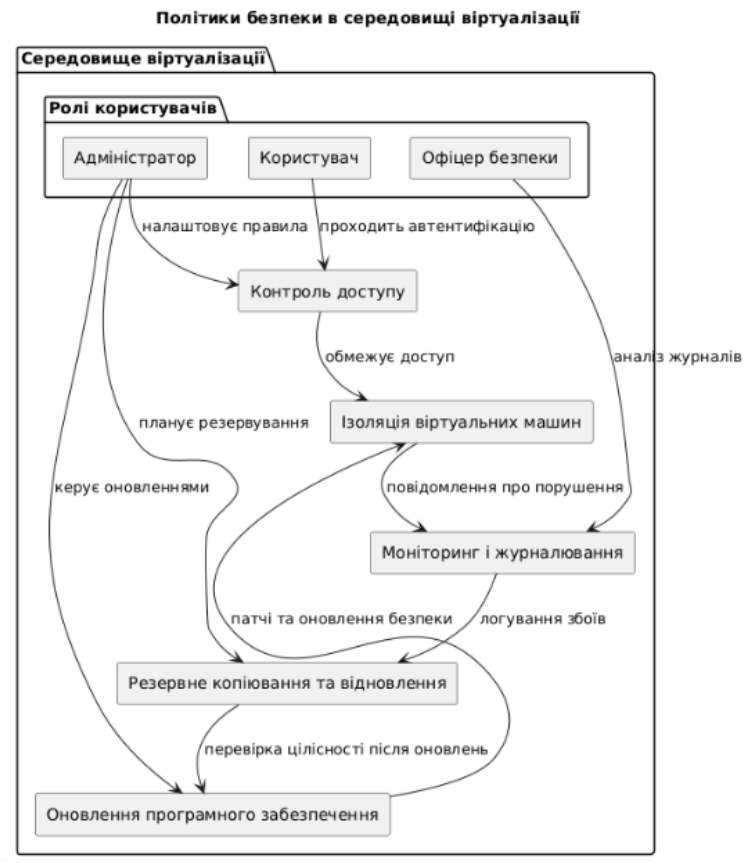
У контексті серверної інфраструктури університету КПІ розробка політик безпеки в середовищі віртуалізації є критичним елементом забезпечення стабільної та захищеної роботи ІТ-систем. Віртуалізація, яка дозволяє розгортати декілька ізольованих середовищ на одному фізичному обладнанні, створює додаткові вектори загроз, що потребують чітко визначених правил взаємодії між компонентами, користувачами та адміністраторськими системами.

Політики безпеки повинні охоплювати кілька ключових напрямів. Насамперед, це контроль доступу — як до фізичних серверів, так і до гіпервізора та віртуальних машин. Доступ до віртуалізованого середовища має надаватися тільки авторизованим користувачам з чітко визначеними правами. Для цього впроваджуються ролі (наприклад, адміністратор системи, користувач, оператор безпеки), які мають різні рівні дозволів. Додатково,

використовуються сучасні методи автентифікації, включаючи багатофакторну автентифікацію (2FA) та регулярну ротацію облікових даних.

Іншим важливим аспектом є ізоляція віртуальних середовищ. У системі повинна бути забезпечена логічна ізоляція між різними віртуальними машинами, щоб запобігти несанкціонованому доступу з однієї VM до іншої. Для цього впроваджуються політики міжмережевих екранів (firewall rules), VLAN-конфігурації та використання механізмів апаратної віртуалізації, що підтримують повну ізоляцію гостьових операційних систем[38].

Важливою складовою політик є моніторинг і аудит. Віртуалізоване середовище повинно бути під постійним наглядом з боку системи моніторингу, яка фіксує події доступу, зміни у конфігурації, підозрілі дії з боку користувачів або програм. Зібрані журнали повинні зберігатися в захищеному середовищі та періодично аналізуватися фахівцями з кібербезпеки.



Також до політик включаються правила оновлення та підтримки гіпервізора, операційних систем віртуальних машин і встановленого ПЗ. Регулярне оновлення програмного забезпечення — це один з основних

заходів, що дозволяє мінімізувати ризики, пов’язані з використанням застарілих компонентів, які можуть містити вразливості.

Резервне копіювання є ще одним напрямком політики безпеки. Має бути визначено порядок створення резервних копій критичних віртуальних машин та даних, періодичність таких копій, а також процедури відновлення після збоїв.

Загалом, політики безпеки повинні бути чітко задокументованими, доступними для відповідального персоналу, регулярно переглядатися й адаптуватися до поточних загроз та змін у IT-інфраструктурі. Їх впровадження дозволяє не лише знизити ризики атак, а й забезпечити високу стабільність, доступність та конфіденційність даних у середовищі віртуалізації університету [39].

Контроль доступу до ресурсів у віртуалізованому середовищі університету КПІ є ключовим елементом забезпечення інформаційної безпеки. У контексті серверної інфраструктури ці політики визначають, які користувачі або системні ролі мають право доступу до конкретних віртуальних машин, мережевих сховищ, служб або програмного забезпечення. Основна мета таких політик — запобігти несанкціонованому доступу, забезпечити розмежування повноважень і гарантувати конфіденційність, цілісність та доступність ресурсів.

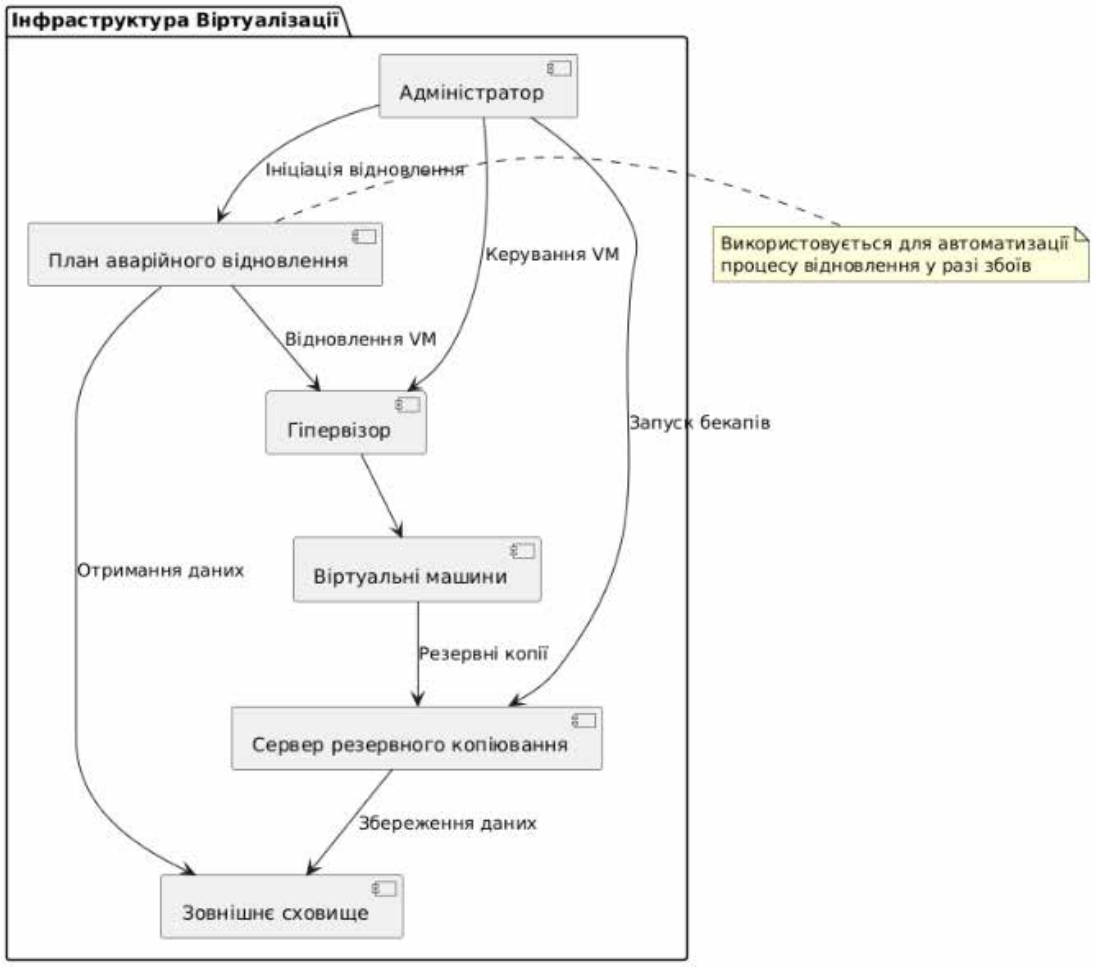
У середовищі КПІ для реалізації контролю доступу використовуються моделі Role-Based Access Control (RBAC) та, в окремих випадках, Attribute-Based Access Control (ABAC). Перша передбачає призначення прав відповідно до ролі користувача (наприклад, студент, викладач, системний адміністратор), що дозволяє централізовано керувати правами доступу без необхідності вручну змінювати кожне призначення. Друга — використовує додаткові атрибути, такі як час доступу, IP-адреса або місце розташування користувача, що забезпечує гнучкішу політику контролю.

Серед найбільш критичних елементів доступу, які підлягають суворій перевірці, є керування адміністративними обліковими записами, доступ до даних користувачів, журналів аудиту та систем резервного копіювання. Будь-яка спроба обійти політику доступу фіксується системою моніторингу та надсилається офіцеру безпеки для перевірки.

Важливою складовою є регулярний перегляд прав доступу — під час зміни посади, завершення навчання або звільнення персоналу. Впроваджені політики також передбачають двофакторну автентифікацію для доступу до критичних систем, а також автоматичне блокування сесій при тривалій неактивності.

Усі ці заходи дозволяють ефективно мінімізувати ризики витоку даних, запобігти внутрішнім загрозам і дотримуватися принципів сучасної кібербезпеки у віртуалізованій інфраструктурі університету [40].

У середовищі віртуалізації резервне копіювання та аварійне відновлення відіграють ключову роль у забезпеченні цілісності, надійності та безперервності роботи серверної інфраструктури університету. Оскільки віртуальні машини об’єднують у собі операційні системи, додатки та дані в єдині образи, процес резервування повинен охоплювати всі рівні функціонування інфраструктури — від гіпервізора до прикладного програмного забезпечення.



Політика резервного копіювання передбачає періодичне збереження повних знімків (snapshot) віртуальних машин на окремих фізичних або віртуальних носіях, що ізольовані від основного середовища. Це дозволяє у разі збою або атаки повернути систему до стабільного стану. Залежно від важливості даних і систем, застосовуються повне, інкрементне або диференційне копіювання. Для критично важливих вузлів резервне копіювання здійснюється щоденно або навіть кілька разів на добу.

Окреме місце займає процедура аварійного відновлення, яка описує покроковий порядок дій персоналу для повернення системи до працездатного стану. Вона включає відновлення гіпервізора, завантаження збережених копій віртуальних машин, перевірку цілісності даних, синхронізацію з базами даних та поновлення доступу до сервісів.

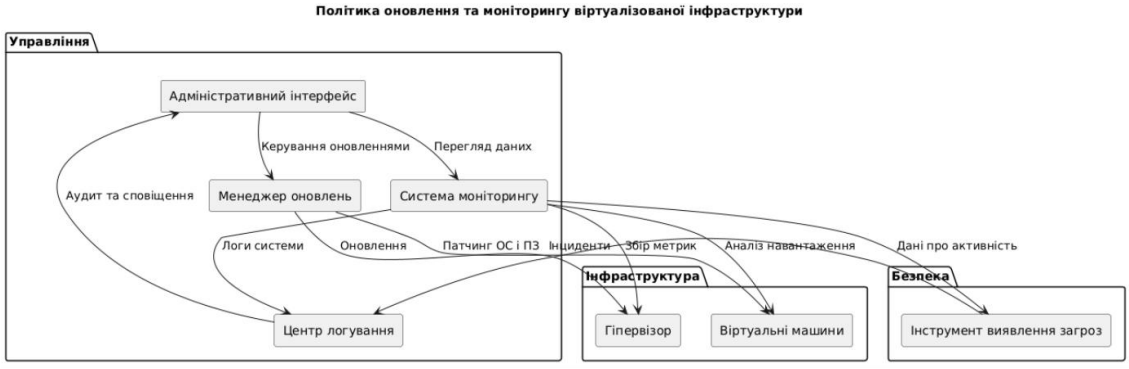
Крім того, політика аварійного відновлення включає тестування сценаріїв катастрофічних збоїв — з метою оцінки швидкості та надійності

системи резервного копіювання в умовах реальних загроз. Це дозволяє адміністраторам оперативно реагувати на інциденти, зменшити час простою та уникнути втрат важливої інформації.

Усі процеси резервного копіювання та відновлення логуються, а доступ до копій обмежується політиками контролю доступу, щоб забезпечити конфіденційність і запобігти несанкціонованому втручанню.

Забезпечення стабільної та безпечної роботи віртуалізованої серверної інфраструктури вимагає чітко визначених політик щодо оновлення та моніторингу системи. У контексті інфраструктури університету КПІ ці політики відіграють критичну роль у підтримці цілісності сервісів, доступності освітніх ресурсів та безперервності наукових обчислень [41].

Політика оновлення передбачає регулярну перевірку актуальності програмного забезпечення, зокрема гіпервізора, операційних систем віртуальних машин, інструментів моніторингу та компонентів безпеки. Впроваджується централізована система управління оновленнями, що дозволяє автоматизувати процес завантаження, перевірки сумісності та встановлення критичних оновлень без порушення поточних сервісів. Перед оновленням виконується резервне копіювання відповідних систем, аби мати змогу швидко повернутися до стабільної версії у разі непередбачених помилок.



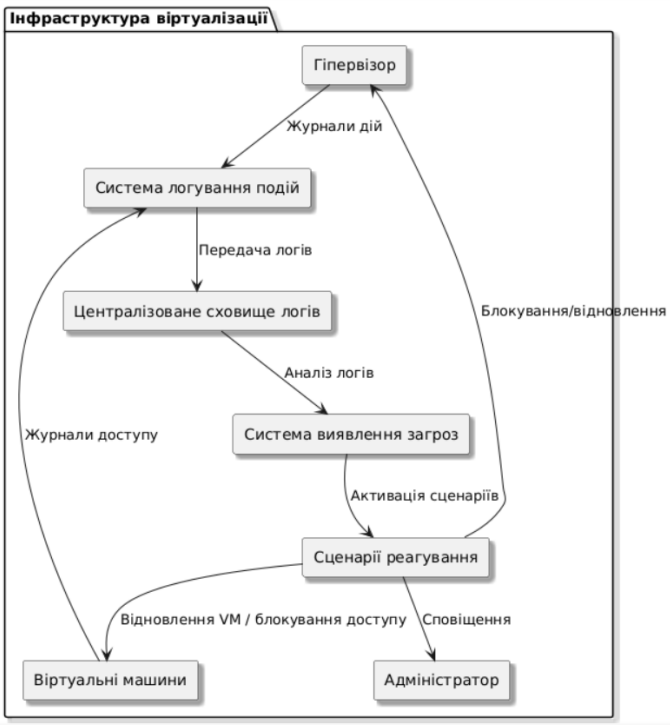
Моніторинг системи здійснюється безперервно з використанням спеціалізованого програмного забезпечення, яке збирає та аналізує метрики продуктивності, навантаження, доступності компонентів та безпекових інцидентів. Усі події реєструються у центральному журналі аудиту, а при

виявленні відхилень від нормативних параметрів автоматично надсилаються сповіщення адміністраторам. Крім того, здійснюється постійний моніторинг систем безпеки для виявлення підозрілої активності, несанкціонованих змін у системі чи спроб експлуатації вразливостей.

Політики оновлення та моніторингу дозволяють підтримувати стабільність роботи IT-інфраструктури, оперативно реагувати на потенційні загрози, а також забезпечувати виконання внутрішніх нормативів безпеки і зовнішніх вимог, зокрема в контексті захисту персональних даних та академічної інформації.

У контексті віртуалізованої серверної інфраструктури університету КПІ логування подій та ефективне реагування на інциденти відіграють ключову роль у підтримці цілісності, доступності та конфіденційності інформаційних ресурсів. В умовах багатокористувацького середовища, де одночасно працюють різноманітні віртуальні машини з доступом до критичних освітніх і наукових даних, систематичне ведення журналів подій дає змогу не лише відслідковувати активність користувачів та систем, але й оперативно виявляти загрози безпеці або відмови в роботі інфраструктури [42].

Система логування подій повинна фіксувати широкий спектр інформації: від входів користувачів і змін у конфігурації віртуальних машин до спроб несанкціонованого доступу, помилок гіпервізора або сповіщень про аномальну активність з боку внутрішніх або зовнішніх агентів. Усі ці дані передаються до централізованого сховища логів, де можуть бути проаналізовані як вручну адміністраторами, так і автоматизованими системами аналізу загроз.

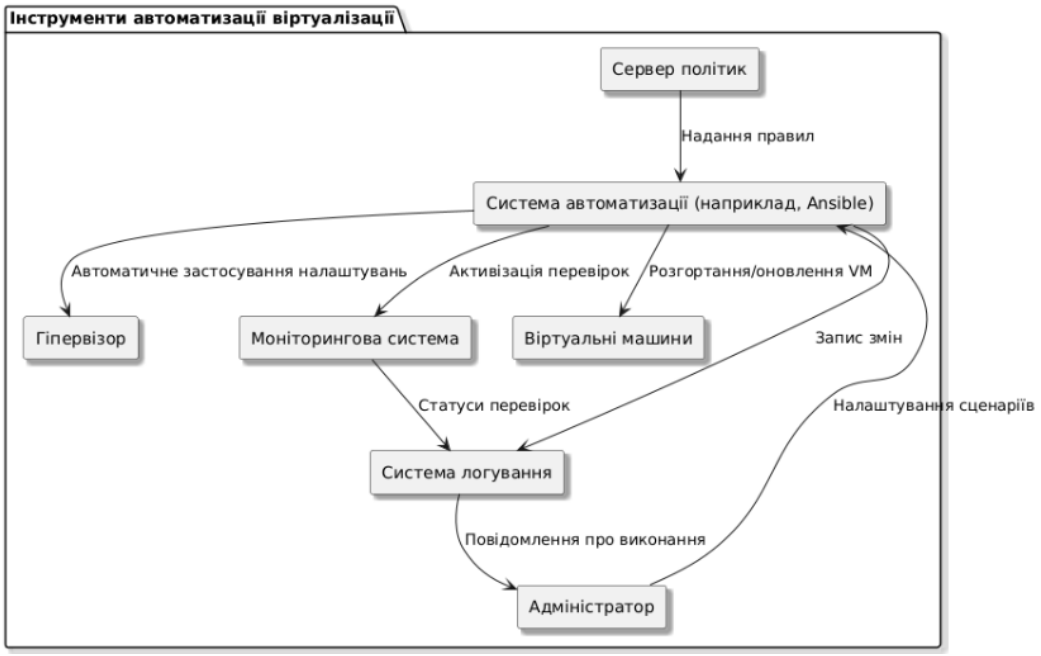


На основі зібраної інформації активується система реагування на інциденти, яка виконує попередньо налаштовані сценарії дій. Наприклад, у випадку виявлення повторюваних спроб входу з неправильним паролем система може автоматично заблокувати обліковий запис, сповістити адміністратора та створити запис для подальшого аудиту. У разі збоїв у роботі віртуальних машин або гіпервізора активуються механізми аварійного відновлення або перемикання на резервні ресурси, що забезпечує безперервність функціонування системи.

Реалізація ефективної системи логування та реагування на інциденти сприяє створенню прозорого та безпечного середовища для навчання і досліджень, дозволяючи адміністраторам оперативно виявляти й усувати загрози, а також дотримуватись внутрішніх політик безпеки й зовнішніх регламентів [43].

У сучасних віртуалізованих середовищах забезпечення безпеки, стабільності та відповідності стандартам є неможливим без ефективної автоматизації. Автоматизація політик безпеки дозволяє централізовано впроваджувати правила, мінімізувати людський фактор та оперативно реагувати на потенційні загрози або зміни в інфраструктурі. У контексті

інфраструктури університету КПІ для цього можуть використовуватися такі інструменти, як Ansible, PowerShell DSC, Terraform або інші платформи керування конфігурацією.



Суть автоматизації полягає в тому, що адміністратор один раз формулює правила або сценарії — наприклад, політики оновлення, бекапів, моніторингу або обмеження доступу — які потім автоматично виконуються у віртуальному середовищі. Сценарії автоматизації зберігаються у спеціальному сховищі (сервер політик), звідки їх завантажує система автоматизації.

Ця система взаємодіє з гіпервізором для конфігурації або створення нових віртуальних машин, із системою моніторингу для налаштування тригерів та перевірок, а також із системою логування для фіксації всіх змін та інцидентів. У результаті, адміністратор отримує звіти про стан виконання сценаріїв, наявні порушення або сповіщення про потенційні проблеми.

Автоматизація політик забезпечує масштабованість, повторюваність дій та дозволяє швидко адаптувати середовище до змін вимог без необхідності ручного втручання, що критично важливо для великої та динамічної IT-інфраструктури університету.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Підготовка серверного обладнання

Під час реалізації віртуалізованого серверного середовища першим кроком є визначення технічних вимог до апаратного забезпечення. Це критично важливий етап, оскільки від обраної конфігурації залежить стабільність, продуктивність і масштабованість усієї інфраструктури. Основна мета цього етапу — підібрати апаратну платформу, що здатна ефективно підтримувати роботу гіпервізора Proxmox VE та всіх віртуальних машин, які планується розгорнути.

Насамперед аналізуються потреби у ресурсах відповідно до кількості й типу сервісів, що будуть розміщені у віртуальному середовищі. Для даного проєкту заплановано встановлення серверів DNS/DHCP, веб-сервера (Apache або Nginx), сервера баз даних (MySQL або PostgreSQL) та хмарного файлового сховища (наприклад, Nextcloud). Кожен із цих компонентів потребує певної кількості оперативної пам’яті, процесорних ресурсів та місця на диску. Окрім того, необхідно врахувати накладні витрати самої віртуалізаційної платформи.

Мінімальні технічні вимоги до системи включають наявність сучасного багатоядерного процесора з підтримкою апаратної віртуалізації (Intel VT-x або AMD-V), не менше 16 ГБ оперативної пам’яті (із можливістю розширення до 32 або більше), твердотільного накопичувача (SSD) для швидкої роботи системи та додаткового HDD для зберігання резервних копій і образів віртуальних машин. Крім того, важливо мати надійний мережевий інтерфейс, бажано з підтримкою VLAN або окремих фізичних інтерфейсів для розмежування зовнішнього і внутрішнього трафіку [44].

Загалом, на цьому етапі закладається апаратна основа, яка дозволить забезпечити гнучку, масштабовану та безпечну роботу віртуалізованого середовища відповідно до поставлених цілей проєкту.

Після визначення технічних вимог наступним кроком є вибір конкретної серверної платформи, яка відповідатиме цим вимогам, а також безпосереднє налаштування апаратної частини. В умовах університетської інфраструктури це може бути як фізичний сервер у серверній кімнаті, так і робоча станція, яка виконує роль сервера для навчального проєкту. Вибір платформи залежить від доступного бюджету, потреб у масштабуванні та надійності, а також планованого навантаження.

У межах даного проєкту було обрано використання стандартного x86-сумісного сервера або потужної робочої станції з підтримкою апаратної віртуалізації. Основу апаратної конфігурації становить багатоядерний процесор Intel Core i7 або AMD Ryzen з підтримкою технології віртуалізації (VT-x / AMD-V), 32 ГБ оперативної пам’яті DDR4 для комфортної роботи кількох віртуальних машин одночасно, SSD-накопичувач обсягом 500 ГБ для системних файлів та операцій гіпервізора, а також додатковий HDD на 1–2 ТБ для зберігання даних і резервних копій [45].

Особливу увагу приділено мережевому забезпеченню. Встановлено два мережеві інтерфейси — один для зовнішнього з’єднання з мережею інтернет, другий для внутрішньої комунікації між віртуальними машинами та управління інфраструктурою. Це дозволяє забезпечити ізоляцію трафіку, що є важливим з міркувань безпеки.

Конфігурація BIOS/UEFI також адаптована для підтримки гіпервізора: активовано параметри віртуалізації (Intel VT-x / AMD-V), вимкнено функції, що можуть заважати стабільній роботі (наприклад, енергозбереження, яке іноді заважає роботі віртуальних машин), а також оновлено прошивку до останньої стабільної версії.

Таким чином, на цьому етапі підготовлено серверну платформу, що відповідає всім потребам проєкту й готова до встановлення Proxmox VE як базового гіпервізора.

Після вибору серверного обладнання необхідно переконатися, що система підтримує апаратну віртуалізацію — це критична умова для

повноцінного функціонування гіпервізора Proxmox VE. Апаратна віртуалізація (технології Intel VT-x або AMD-V) дозволяє гіпервізору напряму взаємодіяти з ресурсами процесора, забезпечуючи високу продуктивність віртуальних машин, повну ізоляцію середовищ та доступ до розширених функцій управління пам’яттю й пристроями.

Перевірка підтримки відбувається на кількох рівнях. Спочатку слід переконатися, що обраний процесор справді має підтримку VT-x (у випадку Intel) або AMD-V (для AMD). Цю інформацію можна знайти в офіційній документації або на сайті виробника процесора. Далі необхідно перевірити, чи не вимкнено підтримку віртуалізації в налаштуваннях BIOS або UEFI. У багатьох системах ця опція вимкнена за замовчуванням, тому користувачеві потрібно вручну активувати параметри Intel Virtualization Technology або SVM Mode (у AMD) [46].

Після цього здійснюється перевірка в операційній системі. Для цього використовуються команди, які виводять інформацію про наявність апаратної віртуалізації. Наприклад, у Linux можна виконати таку команду:

egrep -c '(vmx|svm)' /proc/cpuinfo

Результат, більший за нуль, вказує на те, що процесор підтримує віртуалізацію. Також можна використати утиліту lscpu, яка виводить параметри процесора, зокрема рядок "Virtualization: VT-x" або "Virtualization: AMD-V".

Завершальним етапом є перевірка доступності віртуалізаційних інструкцій для гіпервізора під час інсталяції Proxmox VE. Якщо апаратна віртуалізація не підтримується або вимкнена, система може попередити про обмежену функціональність під час встановлення.

Таким чином, перевірка підтримки VT-x або AMD-V є обов’язковою частиною підготовки середовища і дозволяє гарантувати стабільну роботу гіпервізора та віртуальних машин.

Перш ніж розпочати встановлення гіпервізора Proxmox VE, необхідно підготувати середовище, з якого буде запускатися інсталяція. Найбільш

зручним та поширеним способом є створення завантажувального носія — USB-флешки або ISO-образу для використання в середовищах із підтримкою віртуального завантаження (наприклад, через IPMI, iDRAC або віртуальні носії в BIOS).

Процес починається із завантаження останньої версії образу Proxmox VE з офіційного сайту. Завантажений файл має розширення .iso та включає все необхідне для інсталяції операційної системи та гіпервізора.

Якщо планується встановлення з USB-флешки, слід скористатися утилітами для створення завантажувального диска, такими як Rufus (для Windows), Etcher або dd (для Linux). У Windows за допомогою Rufus достатньо вказати ISO-файл, вибрати флешку та натиснути «Старт», після чого утиліта автоматично створить носій, готовий до використання.

У Linux можна скористатися командою dd, яка копіює образ безпосередньо на пристрій. Наприклад [47]:

sudo dd if=proxmox-ve.iso of=/dev/sdX bs=4M status=progress && sync

де /dev/sdX — це пристрій вашої флешки (важливо бути обережним, щоб не вказати системний диск).

У випадках, коли використовується віддалений доступ до сервера з IPMI або аналогічними технологіями, ISO-образ можна підключити як віртуальний CD/DVD-диск і встановити систему без фізичного носія.

Після завершення підготовки носія необхідно підключити його до сервера, увійти в BIOS або UEFI, і встановити завантаження з USB або віртуального CD. Таким чином розпочнеться процес встановлення Proxmox VE, і сервер буде готовий до розгортання віртуального середовища.

3.2. Встановлення та налаштування гіпервізора

Процес інсталяції Proxmox VE на фізичний сервер починається із завантаження пристрою з попередньо підготовленого носія — USB-флешки або ISO-образу, підключеного через IPMI або схожу систему. Після старту

інсталяційного середовища з’являється графічне меню, де необхідно обрати пункт Install Proxmox VE для початку встановлення.

Далі користувачеві пропонується прийняти ліцензійну угоду, після чого розпочинається налаштування параметрів встановлення. Серед основних налаштувань — вибір цільового жорсткого диска для встановлення системи. За замовчуванням використовується файловий тип LVM-Thin, який добре підходить для гнучкого управління простором у віртуалізованому середовищі. При потребі, можна обрати інший тип файлової системи або налаштувати RAID-масиви.

На наступному кроці вводиться основна інформація: країна, мова та часовий пояс. Потім слід визначити root-пароль для адміністративного доступу та ввести адресу електронної пошти адміністратора, яка буде використовуватись для системних сповіщень.

Далі система запитує мережеві налаштування — IP-адресу (статичну або отриману через DHCP), маску підмережі, шлюз і DNS-сервер. Рекомендується використовувати статичну IP-адресу для стабільного доступу до веб-інтерфейсу управління.

Після підтвердження всіх налаштувань інсталяція починається автоматично й триває декілька хвилин. По завершенні інсталяції система запропонує перезавантажити сервер і витягнути встановлювальний носій.

Після перезавантаження доступ до Proxmox VE здійснюється через веб-інтерфейс, відкривши у браузері вказану IP-адресу за портом 8006 (наприклад, https://192.168.1.100:8006). На цьому етапі система готова до налаштування віртуального середовища, створення VM або LXC-контейнерів, налаштування мережі, сховищ і безпеки.

Налаштування мережевого інтерфейсу в середовищі гіпервізора Proxmox VE є критичним етапом, оскільки забезпечує доступ до веб-інтерфейсу керування, взаємодію між віртуальними машинами та зовнішньою мережею. Після завершення базової інсталяції користувач потрапляє до веб-

інтерфейсу Proxmox, де можна здійснити детальне налаштування мережевих параметрів.

У системі Proxmox VE мережеві інтерфейси найчастіше реалізуються через bridge-інтерфейси (наприклад, vmbr0), які дозволяють об'єднати фізичні мережеві карти з віртуальними мережами. Це дозволяє віртуальним машинам отримувати прямий доступ до зовнішньої мережі, наче вони фізично підключені до маршрутизатора або комутатора.

Початкове налаштування виконується через файл /etc/network/interfaces, однак його можна редагувати й через веб-інтерфейс. У типовій конфігурації створюється міст vmbr0, який асоціюється з фізичним інтерфейсом, наприклад eth0 або eno1. У налаштуваннях зазначається статична IP-адреса, маска підмережі, шлюз за замовчуванням і DNS-сервер.

Наприклад:

```
auto lo
iface lo inet loopback

auto eno1
iface eno1 inet manual

auto vmbr0
iface vmbr0 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    gateway 192.168.1.1
    bridge_ports eno1
    bridge_stp off
    bridge_fd 0
```

У цьому прикладі eno1 — це фізичний інтерфейс, який підключений до зовнішньої мережі, а vmbr0 — міст, який дозволяє віртуальним машинам працювати в одній мережі з фізичними пристроями. Опції bridge_stp off і

bridge_fd 0 налаштовують параметри STP (протокол запобігання петель) та затримки при перемиканні.

Після внесення змін мережеві налаштування можна перезапустити командою:

ifreload -a

або перезавантажити систему для застосування конфігурації.

Таким чином, правильне налаштування мережевого інтерфейсу забезпечує надійну роботу гіпервізора, стабільне підключення до керуючої консолі та ефективну маршрутизацію трафіку між віртуальними та зовнішніми системами.

Додавання сховищ у середовищі Proxmox VE є ключовим етапом, який забезпечує стабільне зберігання дисків віртуальних машин, шаблонів, ISO-образів та резервних копій. Після встановлення гіпервізора за замовчуванням використовується локальне сховище (local або local-lvm), однак для забезпечення гнучкості, масштабованості й надійного бекапу часто додаються додаткові сховища.

Насамперед адміністратор визначає типи сховищ, які необхідні для проєкту. Це можуть бути:

- локальні диски (SSD/HDD) для продуктивних ВМ;
- мережеві сховища — NFS, CIFS (SMB), iSCSI або Ceph, що дозволяють централізоване зберігання та легкий доступ до резервних копій і образів.

Додавання виконується через веб-інтерфейс Proxmox: у вкладці Datacenter → Storage → Add адміністратор обирає тип сховища (наприклад, Directory, NFS, CIFS), вказує його назву, шлях до монтування або мережеву адресу, а також типи даних, які воно зберігатиме (VM disks, ISO images, Backup тощо).

Наприклад, для додавання локального каталогу для резервного копіювання може використовуватись такий запис у /etc/pve/storage.cfg:

dir: backup-local

path /mnt/backup

```
    content backup
    maxfiles 7
    shared 0
```

У цьому прикладі backup-local — це нове сховище, яке розміщується на монтуваному диску /mnt/backup і використовується лише для резервних копій. Параметр maxfiles визначає кількість збережених версій бекапів.

Для підключення NFS сховища конфігурація може виглядати так:

```
nfs: backup-nfs
    server 192.168.1.10
    export /nfs/proxmox
    path /mnt/pve/backup-nfs
    content backup,iso,vztmpl
    maxfiles 5
```

Після додавання сховища адміністратор може вказувати його як місце для створення нових ВМ або для збереження резервних копій через інтерфейс створення або налаштування віртуальних машин.

Таким чином, додавання сховищ у Proxmox VE — це гнучкий процес, що дозволяє оптимізувати використання ресурсів, розподілити навантаження та забезпечити збереження даних у критичних сценаріях.

Створення облікових записів адміністратора в Proxmox VE є одним із важливих кроків для розмежування доступу, забезпечення безпеки та делегування повноважень в управлінні віртуальним середовищем. Після інсталяції системи за замовчуванням доступ здійснюється під обліковим записом root@pam, який має повні права в усій інфраструктурі. Однак використання одного глобального облікового запису суперкористувача не відповідає сучасним практикам безпеки.

Для підвищення захищеності адміністратор створює окремі облікові записи з різними ролями, що дозволяє:

- обмежити коло повноважень кожного користувача;
- фіксувати активність кожного адміністратора;

– запобігти несанкціонованому доступу або помилковим діям.

У Proxmox VE створення нового адміністративного облікового запису виконується через веб-інтерфейс: у розділі Datacenter → Permissions → Users додається новий користувач (наприклад, admin1@pve), для якого вказується пароль та тип автентифікації. Найчастіше використовується реалм PVE (@pve) для внутрішньої автентифікації або зовнішні сервіси — LDAP / Active Directory — для корпоративної інтеграції.

Після створення користувача необхідно призначити йому роль у певному контексті (Datacenter, нода, окрема ВМ тощо). Це робиться через вкладку Permissions → Add → User Permission, де задається шлях (scope), користувач і роль (наприклад, Administrator, PVEVMAdmin, PVEAuditor).

Крім того, можливе створення груп користувачів та застосування ролей на рівні груп, що спрощує адміністрування в середовищі з багатьма операторами.

Завдяки цим діям адміністратор забезпечує контрольований і безпечний доступ до віртуалізованого середовища, де кожен користувач має чітко визначені повноваження відповідно до своєї ролі в IT-інфраструктурі.

Після завершення інсталяції Proxmox VE на фізичний сервер наступним кроком є підключення до веб-інтерфейсу управління, який є основним засобом адміністрування віртуалізованого середовища. Цей інтерфейс реалізовано як веб-додаток, що працює через HTTPS, і дозволяє виконувати всі основні операції — від створення віртуальних машин до моніторингу ресурсів і налаштування безпеки.

Для підключення необхідно знати IP-адресу мережевого інтерфейсу, що була задана під час встановлення системи. Веб-інтерфейс доступний за адресою:

https://<IP-адреса>:8006

Наприклад, якщо IP-адреса сервера — 192.168.1.100, то доступ здійснюється через:

https://192.168.1.100:8006

Після введення адреси у веб-браузері відкривається сторінка входу до системи. За замовчуванням використовується обліковий запис root@pam. Після успішної авторизації відкривається головна консоль Proxmox VE, де можна переглядати вузли, створювати віртуальні машини або контейнери, керувати зберіганням, доступом, а також налаштовувати резервне копіювання й брандмауери.

Перший вхід також може супроводжуватись повідомленням про недійсний сертифікат SSL (оскільки використовується самопідписаний сертифікат). Це нормально на початковому етапі, однак для продуктивного використання бажано замінити його на сертифікат від довіреного центру сертифікації або використати Let's Encrypt.

Завдяки інтуїтивному веб-інтерфейсу Proxmox VE дозволяє зручно й централізовано адмініструвати як окремий сервер, так і цілий кластер, забезпечуючи високу ефективність і контрольованість IT-інфраструктури.

3.3. Налаштування віртуальних машин

На етапі створення шаблонів для віртуальних машин у середовищі Proxmox VE основною метою є спрощення й стандартизація подальшого розгортання віртуальних систем. Шаблони (templates) дозволяють зберегти попередньо налаштовану віртуальну машину з базовим набором програмного забезпечення, конфігурацій мережі, користувачів і служб, що істотно скорочує час на створення нових ВМ.

Для створення шаблону зазвичай використовується один з двох підходів:

1. Створення ВМ з ISO-образу — інсталюється операційна система (наприклад, Ubuntu Server або Debian), налаштовується базовий функціонал: встановлення SSH, оновлення системи, установка потрібних утиліт. Далі віртуальна машина завершується, і з її диска створюється шаблон.

2. Завантаження готового шаблону з офіційного сховища Proxmox — у випадку з контейнерами (CT) це можуть бути LXC-шаблони. У веб-інтерфейсі або через CLI можна завантажити їх напряму та створити контейнери на їх основі.

Процес створення шаблону включає:

- інсталяцію базової ОС;
- налаштування мережі (зазвичай DHCP або фіксована IP-адреса, яку потім змінюють при копіюванні шаблону);
- встановлення необхідного програмного забезпечення (наприклад, Apache, MySQL);
- видалення тимчасових файлів, журналів та SSH-ключів;
- вимкнення машини;
- створення шаблону через функцію "Convert to Template".

Такий підхід дозволяє надалі швидко створювати нові віртуальні машини шляхом клонування шаблону з мінімальною кількістю ручних дій. Це не лише пришвидшує розгортання, а й знижує ймовірність помилок конфігурації. У межах даного проєкту шаблони будуть створені для DNS/DHCP-сервера, веб-сервера, файлового сховища та сервера баз даних.

На етапі налаштування віртуальної машини з DNS/DHCP-сервером головною метою є забезпечення автоматичного надання IP-адрес клієнтам локальної мережі та централізованого доменного іменування для внутрішніх ресурсів. Така ВМ виконує роль базового мережевого сервісу в інфраструктурі віртуалізованого середовища, забезпечуючи доступність та адресацію інших серверів і клієнтів.

У межах реалізації використовується легкий дистрибутив Linux (наприклад, Debian або Ubuntu Server), після чого виконується встановлення необхідних сервісів: ISC DHCP Server для роздачі IP-адрес та BIND9 або dnsmasq для DNS-сервера. Після встановлення конфігуруються такі параметри:

– DHCP: визначаються діапазони IP-адрес, час оренди (lease time), шлюз за замовчуванням, DNS-сервер, і вказуються статичні призначення IP для критичних хостів (наприклад, веб-сервер або файловий сервер).

– DNS: прописуються зони прямого та зворотного дозволу імен (forward та reverse zones), а також імена ресурсів (A-записи), що відображають IP-адреси всіх інших віртуальних машин у проєкті.

Для підвищення стійкості до збоїв, конфігураційні файли сервера зберігаються в окремому каталозі, який включено до системи резервного копіювання Proxmox VE. Також, при першому запуску системи автоматично активуються служби dhcpd та named (або dnsmasq), що гарантує безперервну роботу навіть після перезавантаження.

Ця ВМ, будучи налаштованою й протестованою, стає одним із ключових елементів внутрішньої мережі, надаючи підтримку решті віртуальних серверів за допомогою централізованої адресації та іменування, що значно спрощує адміністрування всієї віртуалізованої інфраструктури.

На етапі налаштування віртуальної машини з веб-сервером основним завданням є розгортання середовища для обслуговування веб-запитів до інформаційних ресурсів локальної мережі або доступних ззовні. У цьому проєкті веб-сервер реалізується на окремій ВМ із встановленим дистрибутивом Linux, переважно Debian або Ubuntu Server, що забезпечує стабільність, безпеку та сумісність із сучасними мережевими службами.

Залежно від обраної технології, інсталюється веб-сервер Apache або Nginx. Apache надає більш традиційне підходи до обробки запитів, з гнучкою системою модулів, тоді як Nginx відомий своєю високою продуктивністю при обробці великої кількості одночасних підключень. Після встановлення виконується базове налаштування:

– Створюється окремий каталог для веб-документів, наприклад /var/www/project, та налаштовуються права доступу.

– У конфігураційних файлах визначаються віртуальні хости (Virtual Hosts або Server Blocks), що дозволяє розгортати кілька сайтів або сервісів на одному сервері.

– Прописується логування подій доступу та помилок з подальшою інтеграцією у систему централізованого моніторингу.

Додатково налаштовуються підтримка PHP (через модуль php-fpm у випадку Nginx або mod_php для Apache), встановлюється захист через брандмауер (firewall) на рівні ВМ, і виконується інтеграція з DNS-сервером, щоб користувачі могли звертатися до сайту за доменним ім’ям, а не IP-адресою.

Після повної перевірки працездатності веб-сервера, ВМ додається до резервного копіювання в Proxmox VE. Це дозволяє забезпечити стабільність, гнучке управління контентом та розширюваність сервісу — важливі характеристики для університетської інфраструктури або внутрішніх інформаційних порталів.

Налаштування віртуальної машини з сервером баз даних є ключовим етапом у створенні функціональної віртуалізованої інфраструктури. У рамках цього проєкту розгортання здійснюється на основі однієї з двох поширених систем управління базами даних — MySQL або PostgreSQL, що забезпечують ефективне зберігання, обробку та доступ до структурованої інформації.

На віртуальну машину встановлюється Linux-дистрибутив, наприклад Ubuntu Server або Debian, після чого виконується інсталяція обраної СУБД. У випадку MySQL використовується пакет mysql-server, а для PostgreSQL — postgresql. Після встановлення системи баз даних проводиться первинне налаштування:

– Створюється окремий користувач бази даних та призначається пароль, що відповідає політикам безпеки.

– Конфігурується доступ до БД як з локальної мережі (наприклад, з веб-сервера), так і з віддалених клієнтів у межах віртуалізованого середовища,

шляхом редагування файлів my.cnf (для MySQL) або pg_hba.conf і postgresql.conf (для PostgreSQL).

– Встановлюються права доступу на рівні ролей і схем для забезпечення принципу найменших привілеїв.

У контексті безпеки налаштовується міжмережевий екран (UFW або iptables), який дозволяє лише необхідні порти (3306 для MySQL або 5432 для PostgreSQL), а також впроваджується система регулярного резервного копіювання бази (наприклад, за допомогою mysqldump або pg_dump у зв'язці з cron-розкладом).

Також важливо забезпечити інтеграцію з іншими ВМ — наприклад, щоб веб-сервер мав змогу підключатися до БД для обробки запитів користувачів. На завершення виконується тестування доступу до бази даних та перевірка цілісності налаштувань, після чого ВМ з сервером БД включається до системи моніторингу й резервного копіювання в Proxmox VE. Таким чином, забезпечується надійна і безпечна робота серверної частини інформаційної інфраструктури.

Встановлення та налаштування файлового сервера або платформи Nextcloud у віртуальному середовищі на базі Proxmox VE дозволяє реалізувати безпечне зберігання, обмін та синхронізацію файлів між користувачами в рамках університетської IT-інфраструктури. У межах проєкту розглядається два варіанти реалізації — класичний файловий сервер на основі Samba або сучасна хмарна платформа Nextcloud, що працює на базі LAMP/LEMP-стеку.

У випадку розгортання Samba-сервера, на обрану ВМ з Linux (наприклад, Ubuntu Server) встановлюється пакет samba. Далі виконується конфігурація спільних папок, які доступні для користувачів у локальній мережі. Надаються права доступу на рівні файлової системи та в SAMBA-конфігурації (smb.conf). Створюються користувачі Samba із відповідними паролями, і вмикається доступ через протокол SMB/CIFS. Усі підключення до спільних ресурсів логуються, а права доступу можна обмежити лише певними ВМ або IP-адресами.

Для реалізації сучаснішого рішення, такого як Nextcloud, необхідно підготувати віртуальну машину зі встановленим веб-сервером (Apache або Nginx), PHP, а також сервером баз даних (MySQL або PostgreSQL). Після завантаження інсталяційного пакета Nextcloud із офіційного сайту виконується його розгортання у кореневому каталозі веб-сервера. Далі створюється база даних і обліковий запис адміністратора платформи. Після першого запуску відбувається автоматичне налаштування конфігурації, підключення до бази даних та ініціалізація файлової системи.

Окрему увагу приділено безпеці — налаштовується HTTPS-доступ (наприклад, за допомогою Let’s Encrypt), обмежується доступ до адміністративної панелі, встановлюється антивірусне сканування (ClamAV), і реалізується двофакторна автентифікація.

Файлова ВМ підключається до внутрішньої мережі і при потребі — до зовнішнього середовища через проксі або VPN. Також налаштовується резервне копіювання файлів і бази даних. У результаті користувачі отримують захищений доступ до своїх даних, а адміністратор — інструменти моніторингу та управління сховищем.

Після створення і налаштування окремих віртуальних машин, що виконують ролі DNS/DHCP-сервера, веб-сервера, сервера баз даних та файлового сервера або Nextcloud, наступним важливим кроком є підключення цих ВМ до єдиної внутрішньої мережі. Це необхідно для забезпечення коректної взаємодії між сервісами, ізоляції внутрішнього трафіку від зовнішнього, а також для підвищення загальної безпеки віртуалізованої інфраструктури.

У середовищі Proxmox VE мережеве підключення віртуальних машин реалізується через віртуальні мережеві мости (bridges). Для створення внутрішньої мережі адміністратор конфігурує окремий міст, наприклад, vmbr1, який не буде асоційований із фізичним мережевим інтерфейсом сервера. Такий підхід дозволяє створити ізольовану мережу, доступну виключно в межах гіпервізора.

Кожна з віртуальних машин підключається до цього моста шляхом вказання відповідного інтерфейсу при створенні або редагуванні налаштувань ВМ у веб-інтерфейсі Proxmox або через командний рядок. На рівні операційної системи ВМ виконується налаштування IP-адреси або отримання адреси через DHCP, якщо такий сервер уже працює в мережі.

Таким чином, усі ВМ, підключені до vmbr1, можуть обмінюватися даними, не виходячи за межі фізичного сервера, що забезпечує низькі затримки, підвищену безпеку та зменшення навантаження на зовнішні інтерфейси. При потребі — окремі ВМ можуть мати по два інтерфейси: один — для внутрішньої взаємодії, другий — для зовнішнього доступу, з можливістю фільтрації трафіку через міжмережевий екран.

Результатом цього етапу є формування логічної внутрішньої мережі, в якій розгорнуті всі необхідні сервіси з чітко визначеною топологією, підключенням, маршрутизацією і безпековими обмеженнями.

3.4. Забезпечення ізоляції та контроль доступу

Організація мережевої ізоляції між віртуальними машинами є критично важливою складовою забезпечення безпеки у віртуалізованому середовищі. У межах цього підпункту здійснюється поділ внутрішньої інфраструктури на логічні сегменти з обмеженням мережевої взаємодії між віртуальними машинами відповідно до їх ролей та рівня довіри.

У Proxmox VE ізоляція реалізується за допомогою віртуальних мережевих мостів (bridge) та віртуальних LAN (VLAN). Наприклад, веб-сервер може бути підключений до одного віртуального моста (vmbr1), а сервер баз даних — до іншого (vmbr2). У такому випадку між ними не буде прямого мережевого трафіку, якщо тільки не налаштовано спеціальний маршрутизатор або firewall з контрольованим доступом.

Ще більш гнучке рішення — використання VLAN-тегування, що дозволяє на одному фізичному або логічному інтерфейсі обслуговувати кілька

ізольованих віртуальних мереж. Це особливо зручно при обмеженій кількості фізичних портів або у випадках, коли ВМ мають перебувати на одній фізичній мережі, але логічно бути ізольованими.

Додатковий рівень безпеки забезпечується за допомогою вбудованого міжмережевого екрану (Proxmox Firewall), який дозволяє задати правила доступу на рівні кожної ВМ або вузла. Наприклад, можна дозволити підключення до веб-сервера лише ззовні, але заборонити будь-який вхідний трафік на базу даних з інших сегментів.

Таким чином, організована ізоляція дозволяє зменшити ризики поширення загроз, локалізувати атаки, забезпечити контрольовану взаємодію між компонентами системи та сприяє дотриманню принципів мінімально необхідного доступу (least privilege) і захисту по глибину (defense-in-depth).

Налаштування VLAN або окремих bridge-інтерфейсів у середовищі Proxmox VE дозволяє реалізувати ефективну логічну ізоляцію мережевого трафіку між різними віртуальними машинами. Це є важливою складовою безпечної та керованої інфраструктури, оскільки дає змогу розмежовувати трафік між сервісами з різними рівнями довіри або функціональними ролями.

У разі використання VLAN-тегування адміністратор створює на рівні комутаційного обладнання окремі VLAN-сегменти (наприклад, VLAN 10 для DNS/DHCP, VLAN 20 для баз даних, VLAN 30 для веб-серверів). У конфігурації мережевого інтерфейсу Proxmox VE ці VLAN додаються як окремі віртуальні інтерфейси (vmbr0.10, vmbr0.20, тощо), що дозволяє призначати кожну ВМ до потрібного сегменту.

Альтернативно або додатково, можна створити окремі bridge-інтерфейси (vmbr1, vmbr2, vmbr3), які не зв’язані між собою. Такі інтерфейси можуть бути прив’язані до окремих фізичних мережевих портів або працювати у віртуальному режимі на одному фізичному інтерфейсі, але без маршрутизації між ними.

Процес включає редагування /etc/network/interfaces або використання графічного веб-інтерфейсу Proxmox для додавання bridge-інтерфейсів та

VLAN-інтерфейсів, після чого ці мережі стають доступними для призначення конкретним віртуальним машинам.

Таке налаштування забезпечує ізоляцію, мінімізує ризик міжмережевого сканування або несанкціонованого доступу до чутливих сервісів, і дозволяє гнучко реалізовувати політики доступу з використанням міжмережевого екрану або зовнішніх маршрутизаторів.

Встановлення правил міжмережевого екрану є ключовим етапом у забезпеченні захисту віртуалізованого середовища. У контексті Proxmox VE можливо застосовувати як вбудований міжмережевий екран Proxmox Firewall, так і класичні інструменти керування доступом до мережі, такі як iptables (або nftables) всередині окремих віртуальних машин.

Proxmox Firewall дозволяє застосовувати політики безпеки як на рівні центру керування (DataCenter), так і для окремих вузлів чи віртуальних машин. Це дає змогу централізовано контролювати доступ до адміністративних інтерфейсів, сервісів (наприклад, веб-серверів, баз даних), а також запобігати небажаному трафіку.

Наприклад, можна дозволити лише вхідні з’єднання до порту 80/443 для веб-сервера та закрити всі інші порти, або дозволити SSH-доступ тільки з певної підмережі адміністратора. Такі політики зручно задаються через веб-інтерфейс Proxmox або безпосередньо у файлах конфігурацій /etc/pve/firewall/.

Якщо використовується iptables усередині ВМ, це надає додаткову гнучкість у налаштуванні доступу на рівні самої системи. Це може бути корисно, наприклад, для тонкого налаштування доступу до служб баз даних або обмеження IP-адрес, з яких дозволено авторизацію.

Поєднання Proxmox Firewall і iptables дозволяє реалізувати багаторівневу модель захисту — як зовнішню, так і внутрішню, що суттєво підвищує рівень безпеки інфраструктури. Усі правила мають бути документовані та протестовані, щоб уникнути ситуацій, коли важливі сервіси стають недоступними або навпаки — відкриті для сторонніх користувачів.

Впровадження обмежень доступу за IP-адресами та ключами SSH є важливим елементом захисту віртуалізованого середовища, який знижує ризик несанкціонованого доступу до серверів. Такий підхід базується на принципі "доступ лише для авторизованих", і поєднує фільтрацію на мережевому рівні та автентифікацію на рівні операційної системи.

Для обмеження за IP-адресами в середовищі Proxmox або всередині окремих віртуальних машин (на базі Linux) використовують інструменти iptables, nftables або Proxmox Firewall. Наприклад, можна дозволити підключення до SSH-сервісу (порт 22) лише з певної IP-адреси або підмережі — зазвичай мережі адміністратора чи VPN:

iptables -A INPUT -p tcp --dport 22 -s 192.168.1.100 -j ACCEPT

iptables -A INPUT -p tcp --dport 22 -j DROP

Це дозволить підключення лише з IP-адреси 192.168.1.100, блокуючи інші запити.

Додатково рекомендується відмовитися від автентифікації за паролем у SSH на користь ключової автентифікації. Для цього в системі налаштовується доступ до облікового запису адміністратора лише через відкритий SSH-ключ, попередньо доданий до файлу ~/.ssh/authorized_keys.

У конфігураційному файлі /etc/ssh/sshd_config слід вказати:

PermitRootLogin no

PasswordAuthentication no

PubkeyAuthentication yes

Це дозволяє підключення лише за допомогою ключів та блокує автентифікацію паролем і вхід під root-користувачем.

Таким чином, комбінація IP-фільтрації та ключової автентифікації суттєво підвищує рівень захисту системи, забезпечуючи доступ лише для уповноважених осіб та мінімізуючи ризики атак типу brute force чи несанкціонованих підключень.

У середовищі Proxmox налаштування ролей і політик доступу є важливим елементом для забезпечення безпеки віртуалізованої

інфраструктури. Це дозволяє обмежити доступ до різних функцій і ресурсів для різних категорій користувачів. У Proxmox використовується механізм Access Control List (ACL), який дає змогу визначити, які дії користувач може виконувати над тими чи іншими об'єктами системи.

Proxmox пропонує кілька вбудованих ролей для управління доступом, кожна з яких має свої права на доступ до різних функцій системи. Це дозволяє дуже точно налаштувати доступ і відповідно мінімізувати ризики несанкціонованого доступу чи помилок.

- PVEAdmin — роль з найбільшими правами, яка дозволяє користувачу виконувати будь-які операції в системі, включаючи створення, налаштування та видалення віртуальних машин, управління користувачами і настройками гіпервізора.
- PVEUser — роль, яка надає обмежений доступ, що дозволяє користувачу працювати з віртуальними машинами, але без можливості змінювати конфігурації самого гіпервізора чи інших критичних налаштувань.
- PVEReadOnly — ця роль дає змогу лише переглядати налаштування та статус віртуальних машин без можливості їх зміни.
- PVEBackup — роль, яка надає доступ до функцій резервного копіювання, але без можливості змінювати конфігурації або запускати віртуальні машини.

Процес налаштування ролей та політик доступу в Proxmox включає кілька етапів. Спочатку необхідно створити користувача, для якого будуть призначені конкретні права доступу. Створення користувача в Proxmox можна здійснити через веб-інтерфейс або командний рядок, використовуючи команду:

pveum useradd johndoe@pve -password mypassword

Після створення користувача можна призначити йому роль. Для цього використовують команду pveum aclmod, яка дозволяє задати, які ресурси будуть доступні для цього користувача і які саме дії він зможе виконувати з

ними. Наприклад, для надання користувачу прав лише для перегляду всіх віртуальних машин системи, можна виконати таку команду:

pveum aclmod / -user johndoe@pve -role PVEReadOnly

У випадку, якщо потрібно надати доступ лише до конкретної віртуальної машини, наприклад, VM з ID 100, це можна зробити за допомогою наступної команди:

pveum aclmod /vms/100 -user johndoe@pve -role PVEUser

Це дозволить користувачу "johndoe" працювати тільки з віртуальною машиною з ID 100, без доступу до інших ресурсів або змін у конфігурації гіпервізора.

Налаштування ролей і політик доступу в Proxmox дає змогу точно контролювати, хто і до яких ресурсів має доступ. Це дозволяє значно підвищити рівень безпеки середовища, запобігти несанкціонованому доступу або помилковим змінам, а також допомагає в управлінні великими віртуалізованими інфраструктурами. Додатково, застосування таких політик дозволяє дотримуватися принципу найменших прав, що є основним принципом безпеки.

Завдяки цьому підходу, система може забезпечити гнучкість і можливість точного налаштування доступу до різних функцій віртуалізації, що особливо важливо в умовах корпоративного використання, де існує необхідність в строгому контролі за доступом до різних ресурсів та можливість розмежування прав між різними адміністраторами, розробниками і користувачами.

3.5. Коментарі до конфігурацій, скриптів і налаштувань

У процесі налаштування та управління віртуалізованим середовищем за допомогою Proxmox VE важливим аспектом є правильна конфігурація системних файлів. Віртуалізація на базі Proxmox побудована на Linux-середовищі, тому конфігурація здійснюється через файли системи, в яких

прописані налаштування для різних сервісів і компонентів. Вивчення та розуміння цих файлів є важливим етапом для досягнення ефективної та безпечної роботи віртуалізованого середовища.

1. Файл конфігурації Proxmox /etc/pve/storage.cfg

Цей файл відповідає за налаштування сховищ у системі Proxmox. У ньому міститься інформація про всі підключені диски, типи зберігання даних та інші параметри для збереження віртуальних машин та контейнерів. Серед важливих параметрів цього файлу можна виділити:

- storage: Ідентифікатор для сховища.
- path: Шлях до фізичного диска або каталогу на сервері.
- content: Типи вмісту, які можна зберігати в сховищі (наприклад, образи віртуальних машин, шаблони).

Приклад конфігурації:

storage: local

path /var/lib/vz

content iso,backup,vztmpl,rootdir

2. Файл конфігурації мережі /etc/network/interfaces

Цей файл відповідає за налаштування мережевих інтерфейсів у системі. В ньому прописуються IP-адреси, мережеві адаптери та маршрутизація. Правильне налаштування мережі є важливим для забезпечення зв’язку між віртуальними машинами та фізичним середовищем.

Основні параметри:

- iface: Інтерфейс мережі (наприклад, eth0 або vmbr0).
- address: Статична IP-адреса, призначена інтерфейсу.
- gateway: Шлюз за замовчуванням.
- bridge_ports: Параметри для налаштування віртуальних мостів.

Приклад конфігурації:

iface vmbr0 inet static

address 192.168.1.10

netmask 255.255.255.0

gateway 192.168.1.1

bridge_ports eth0

3. Файл конфігурації віртуальних машин /etc/pve/qemu-server/VMID.conf

У цьому файлі містяться налаштування для кожної віртуальної машини. Кожна ВМ має свій унікальний ідентифікатор (VMID), і цей файл визначає налаштування ресурсів ВМ, таких як процесор, пам'ять, мережеві інтерфейси, дискові простори та інші параметри.

Основні параметри:

- cpu: Кількість ядер процесора для ВМ.
- memory: Обсяг оперативної пам'яті.
- net0: Мережевий інтерфейс ВМ.
- scsi0: Налаштування диска.

Приклад конфігурації:

cores: 2

memory: 4096

net0: virtio=AA:BB:CC:DD:EE:FF,bridge=vmbr0

scsi0: local-lvm:vm-100-disk-0,size=32G

4. Файл конфігурації Proxmox /etc/pve/user.cfg

Цей файл містить інформацію про користувачів, які мають доступ до системи Proxmox, їхні ролі та права доступу. Це дозволяє налаштувати контроль доступу та керувати правами користувачів.

Основні параметри:

- user: Ім'я користувача.
- role: Роль користувача (наприклад, PVEAdmin, PVEUser).

Приклад конфігурації:

user johndoe@pve: PVEAdmin

5. Файл конфігурації брандмауера /etc/pve/firewall/cluster.fw

Цей файл містить налаштування брандмауера для всієї кластерної інфраструктури Proxmox. Він дозволяє визначити правила доступу між хостами і віртуальними машинами, що забезпечує додатковий рівень безпеки.

Основні параметри:

- fw: Вказує, чи увімкнено брандмауер для конкретного хоста або мережі.
- action: Тип дії, яка буде виконана на запит (наприклад, ACCEPT, DROP).
- source: Джерело з'єднання.
- destination: Куди намагається підключитися з'єднання.

Приклад конфігурації:

fw: ACCEPT

action: ACCEPT

source: 192.168.1.0/24

destination: 192.168.2.0/24

6. Файл конфігурації для резервного копіювання /etc/pve/vzdump.cron

Цей файл визначає налаштування для автоматичних резервних копій віртуальних машин. Використовується для планування завдань для створення бекапів і визначення місця збереження копій.

Основні параметри:

- storage: Місце для збереження резервних копій.
- mode: Режим резервного копіювання (наприклад, snapshot, suspend).
- schedule: Час і частота виконання завдання.

Приклад конфігурації:

storage: backup-storage

mode: snapshot

schedule: "0 3 * * *"

7. Файл конфігурації для управління завданнями /etc/pve/tasks.cfg

Цей файл використовується для збереження конфігурацій для автоматичних завдань та планування задач в Proxmox, таких як оновлення або моніторинг стану віртуальних машин.

Основні параметри:

- task: Тип завдання (наприклад, backup, migrate).
- interval: Інтервал для виконання завдання.

Приклад конфігурації:

task: backup

interval: 24h

Ретельне вивчення та налаштування важливих конфігураційних файлів є необхідним етапом для забезпечення стабільної роботи системи. Кожен конфігураційний файл відповідає за окремі аспекти роботи віртуалізованого середовища, такі як управління сховищами, мережами, доступом до ресурсів та безпекою. Правильна настройка цих файлів забезпечує ефективне та безпечне функціонування інфраструктури, що є основою для надійної роботи віртуальних серверів та їх компонентів.

Автоматизація процесів налаштування та управління віртуальним середовищем є важливою складовою для зменшення часу, необхідного для розгортання та конфігурації нових серверів або віртуальних машин. Для цієї мети використовуються різноманітні інструменти, зокрема скрипти, які дозволяють стандартизувати та спростити процеси налаштування і зменшити кількість людських помилок. У Proxmox VE цей підхід дозволяє значно підвищити ефективність управління інфраструктурою.

1. Створення базових скриптів для встановлення

Один з перших кроків автоматизації — це створення скриптів для автоматичного встановлення програмного забезпечення та налаштувань на віртуальних машинах. Ці скрипти дозволяють автоматично інсталювати та налаштувати необхідні сервіси, що є необхідними для функціонування системи, наприклад, DNS, DHCP, веб-сервери, бази даних та файлові сервери.

Приклад скрипту для встановлення Apache:

```
#!/bin/bash
# Оновлення списку пакетів
apt-get update -y
```

```
# Встановлення Apache
apt-get install apache2 -y

# Запуск і додавання до автозапуску
systemctl start apache2
systemctl enable apache2

# Перевірка статусу сервісу
systemctl status apache2
```

Цей скрипт оновлює пакети на сервері, встановлює веб-сервер Apache, запускає його та налаштовує на автозапуск при кожному завантаженні системи.

2. Скрипти для налаштування мережі

Автоматизація налаштувань мережі є ще одним важливим аспектом для забезпечення швидкого й правильного зв’язку між серверами. Це може включати налаштування IP-адрес, шлюзів, а також налаштування мережевих інтерфейсів для віртуальних машин.

Приклад скрипту для налаштування статичної IP-адреси:

```
#!/bin/bash
# Зміна налаштувань мережі
echo "iface eth0 inet static" >> /etc/network/interfaces
echo "address 192.168.1.100" >> /etc/network/interfaces
echo "netmask 255.255.255.0" >> /etc/network/interfaces
echo "gateway 192.168.1.1" >> /etc/network/interfaces

# Перезапуск мережевого інтерфейсу
systemctl restart networking
```

Цей скрипт автоматично змінює налаштування мережі, призначаючи статичну IP-адресу для мережевого інтерфейсу eth0.

3. Скрипти для налаштування брандмауера та безпеки

Для забезпечення безпеки віртуалізованого середовища важливо налаштувати брандмауер і правила доступу. Автоматизація цього процесу за допомогою скриптів дозволяє централізовано управляти доступом до віртуальних машин і сервісів.

Приклад скрипту для налаштування базових правил брандмауера (iptables):

```
#!/bin/bash

# Встановлення базових правил iptables

iptables -A INPUT -p tcp --dport 22 -j ACCEPT    # Дозволити SSH доступ

iptables -A INPUT -p tcp --dport 80 -j ACCEPT     # Дозволити HTTP доступ

iptables -A INPUT -p tcp --dport 443 -j ACCEPT   # Дозволити HTTPS доступ

iptables -A INPUT -j DROP                        # Заборонити інші з'єднання


# Збереження налаштувань iptables

iptables-save > /etc/iptables/rules.v4
```

Цей скрипт дозволяє приймати лише специфічні типи з'єднань (SSH, HTTP, HTTPS) та блокувати всі інші.

4. Скрипти для резервного копіювання

Автоматизоване резервне копіювання є необхідною частиною будь-якої інфраструктури. Використовуючи скрипти для автоматичного створення резервних копій даних та віртуальних машин, можна значно знизити ймовірність втрати важливої інформації.

Приклад скрипту для створення резервної копії віртуальної машини:

```
#!/bin/bash

# Створення резервної копії для VM з ID 100

vzdump 100 --storage backup-storage --mode snapshot --compress gzip


# Перевірка результатів резервного копіювання
```

```
if [ $? -eq 0 ]; then
   echo "Резервне копіювання пройшло успішно"
else
   echo "Помилка резервного копіювання"
fi
```

Цей скрипт використовує інструмент vzdump для створення резервної копії віртуальної машини з ID 100. Резервна копія зберігається в сховищі, яке визначено як backup-storage, і стискається у форматі gzip.

5. Підключення до скриптів для автоматизації з Proxmox API

Proxmox VE надає REST API, яке дозволяє автоматизувати багато процесів управління через зовнішні скрипти або програми. Використання API дає можливість інтегрувати зовнішні автоматизаційні системи для моніторингу, управління віртуальними машинами, а також для їх створення і видалення.

Приклад виклику API через cURL для створення нової віртуальної машини:

```
#!/bin/bash
# Авторизація в API Proxmox
PROXMOX_URL="https://your-proxmox-server:8006/api2/json"
USER="root@pam"
PASS="yourpassword"

# Отримання токена авторизації
TOKEN=$(curl -k -X POST $PROXMOX_URL/access/ticket -d "username=$USER&password=$PASS" | jq -r .data.ticket)

# Створення нової ВМ
VMID=$(curl -k -X POST $PROXMOX_URL/nodes/pve/qemu -d "vmid=101&cores=2&memory=2048&net0=virtio,bridge=vmbr0&disk=local:32" -H "Authorization: PVEAuthCookie=$TOKEN")
```

```
echo "VM ID: $VMID"
```

Цей скрипт автоматично авторизується через Proxmox API та створює нову віртуальну машину з ID 101, вказуючи основні ресурси: кількість ядер процесора, обсяг пам'яті, мережевий інтерфейс та розмір диска.

Автоматизація налаштувань через скрипти значно покращує ефективність управління віртуалізованими середовищами. Це дозволяє зменшити людську помилку, прискорити процес розгортання нових серверів, а також забезпечити стандартизований підхід до налаштування систем. Використання скриптів для автоматизації встановлення програмного забезпечення, налаштувань мережі, безпеки та резервного копіювання сприяє більш зручному та надійному управлінню інфраструктурою Proxmox VE.

При розробці та налаштуванні віртуалізованого середовища для університетської IT-інфраструктури, кожен етап налаштування потребує уважного підходу та вибору правильних параметрів для забезпечення стабільності, безпеки та ефективності роботи системи. Вибір операційної системи для віртуальних машин не є випадковим, і в нашому випадку обрано Ubuntu Server, оскільки це стабільна та надійна система з відкритим кодом, яка добре підтримується спільнотою та має великий набір інструментів для адміністрування. Для забезпечення стабільності роботи віртуальних машин було обрано розподіл ресурсів таким чином, щоб відповідати вимогам для різних сервісів. Для DNS/DHCP-серверів використано мінімальні ресурси, оскільки ці сервіси потребують низького навантаження, в той час як для сервера бази даних та файлових сервісів було виділено більше ресурсів, оскільки ці сервіси часто обробляють значні обсяги даних.

Налаштування мережевих параметрів також не менш важливе для ефективності та безпеки системи. Статичні IP-адреси були вибрані для ключових сервісів (DNS, DHCP, веб-сервери, бази даних), оскільки це дозволяє уникнути проблем із зміною адрес у разі перезавантаження чи перепідключення до мережі. Вибір використання VLAN для ізоляції віртуальних машин дозволяє значно підвищити безпеку, ізолюючи різні

сервіси один від одного, таким чином, зменшується ймовірність несанкціонованого доступу до критичних даних.

Щодо безпеки, налаштування міжмережевого екрану було здійснено за принципом мінімальних привілеїв, що дозволяє забезпечити максимальний захист при мінімальних налаштуваннях доступу. Так, доступ до адміністративних інтерфейсів був обмежений лише для визначених IP-адрес, що мінімізує ризики несанкціонованих спроб доступу до конфіденційних налаштувань. Водночас відкриті порти для HTTP/HTTPS дозволяють зовнішнім користувачам безперешкодно взаємодіяти з веб-сервісами.

Що стосується автоматизації процесів, то для зручності адміністрування та забезпечення швидкого розгортання нових віртуальних машин було створено кілька скриптів для автоматичного налаштування веб-сервера, бази даних, а також встановлення правил брандмауера. Це дозволяє не тільки пришвидшити процес налаштування, але й мінімізувати помилки, які можуть виникнути під час ручного введення параметрів.

Обрана конфігурація є оптимальною для виконання завдань у межах університетської IT-інфраструктури, оскільки враховує як технічні, так і безпекові вимоги, що важливо для стабільної та безпечної роботи системи в умовах реального використання.

Для автоматизації налаштування віртуалізованого середовища, розгортання сервісів та конфігурацій було використано кілька bash- та shell-скриптів. Ці скрипти дозволяють зекономити час на налаштування сервісів, уникнути помилок, а також стандартизувати процес встановлення та конфігурації.

Приклад 1: Скрипт для налаштування мережевих інтерфейсів

Цей скрипт автоматично налаштовує мережеві інтерфейси на віртуальних машинах, забезпечуючи статичні IP-адреси для ключових сервісів, таких як DNS, DHCP, веб-сервери.

```
#!/bin/bash
```

```
# Налаштування статичних IP-адрес для мережевих інтерфейсів

# Визначаємо інтерфейси
INTERFACE="eth0"
IP_ADDRESS="192.168.1.100"
NETMASK="255.255.255.0"
GATEWAY="192.168.1.1"

# Оновлюємо конфігурацію мережі
echo "Configuring network interface $INTERFACE..."

cat <<EOL > /etc/network/interfaces
auto $INTERFACE
iface $INTERFACE inet static
    address $IP_ADDRESS
    netmask $NETMASK
    gateway $GATEWAY
EOL

# Перезапускаємо мережу
echo "Restarting networking service..."
systemctl restart networking

echo "Network interface $INTERFACE configured successfully."
```

Цей скрипт задає статичну IP-адресу для інтерфейсу eth0, а також налаштовує шлюз та маску підмережі. Після зміни конфігурації мережі, він автоматично перезапускає мережеві сервіси.

Приклад 2: Скрипт для встановлення веб-сервера Apache

Цей скрипт використовується для автоматичного встановлення та налаштування веб-сервера Apache на віртуальній машині. Він включає перевірку наявності встановленого пакета та встановлення залежностей.

```
#!/bin/bash

# Встановлення веб-сервера Apache

echo "Checking if Apache is installed..."

if ! dpkg -l | grep -q apache2; then
    echo "Apache is not installed. Installing Apache..."
    apt update && apt install -y apache2
else
    echo "Apache is already installed."
fi

# Налаштування для автоматичного запуску Apache при завантаженні
echo "Enabling Apache to start on boot..."
systemctl enable apache2

# Перезапуск Apache для застосування змін
echo "Starting Apache service..."
systemctl start apache2

echo "Apache installation and configuration complete."
```

Цей скрипт перевіряє, чи встановлений Apache, і в разі необхідності виконує встановлення пакета, після чого налаштовує сервер для автоматичного запуску при завантаженні системи.

Приклад 3: Скрипт для налаштування firewall (iptables)

Цей скрипт налаштовує брандмауер на базі iptables, дозволяючи лише необхідні порти для конкретних сервісів, наприклад, HTTP, HTTPS, SSH.

```
#!/bin/bash

# Налаштування базових правил брандмауера з використанням iptables

echo "Setting up firewall rules..."

# Очищаємо всі існуючі правила
iptables -F

# Дозволяємо доступ до локальної мережі
iptables -A INPUT -i lo -j ACCEPT

# Дозволяємо SSH (порт 22)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Дозволяємо HTTP (порт 80) та HTTPS (порт 443)
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT

# Блокуємо доступ до всіх інших портів
iptables -A INPUT -j DROP

# Зберігаємо конфігурацію iptables
iptables-save > /etc/iptables/rules.v4

echo "Firewall rules set successfully."
```

Цей скрипт налаштовує базові правила для брандмауера, дозволяючи доступ лише до необхідних портів (SSH, HTTP, HTTPS) і блокує всі інші

підключення. Після цього правила зберігаються, щоб вони застосовувались при перезавантаженні системи.

Приклад 4: Скрипт для автоматизації резервного копіювання

Цей скрипт виконує резервне копіювання даних важливих файлів або конфігураційних файлів на файловий сервер або в іншу безпечну локацію.

```
#!/bin/bash

# Автоматичне резервне копіювання

SOURCE_DIR="/var/www/html"
BACKUP_DIR="/mnt/backup"

# Перевірка наявності каталогу для резервного копіювання
if [ ! -d "$BACKUP_DIR" ]; then
   echo "Backup directory does not exist. Creating..."
   mkdir -p $BACKUP_DIR
fi

# Створення резервної копії
BACKUP_FILE="$BACKUP_DIR/backup_$(date +%F).tar.gz"

echo "Creating backup of $SOURCE_DIR..."
tar -czf $BACKUP_FILE $SOURCE_DIR

echo "Backup created successfully: $BACKUP_FILE"
```

Цей скрипт автоматизує процес резервного копіювання даних з веб-сервера, архівуючи каталог з веб-ресурсами в файл .tar.gz. Резервна копія зберігається в каталозі, визначеному змінною BACKUP_DIR, і має назву з поточною датою.

Використання bash- та shell-скриптів для автоматизації рутинних завдань дозволяє значно підвищити ефективність налаштування, уникнути людських помилок та забезпечити стандартизовану конфігурацію для віртуалізованого середовища. Це особливо важливо для масштабованих інфраструктур, де кожен етап налаштування має бути повторюваним та контрольованим.

Оптимізація, моніторинг та обслуговування віртуальних машин (ВМ) є важливими компонентами для забезпечення високої продуктивності та безпеки віртуалізованої інфраструктури. У процесі реалізації проєкту було здійснено кілька ключових кроків для підвищення ефективності роботи віртуальних машин та налаштування відповідних механізмів моніторингу й обслуговування, що дозволяють забезпечити стабільність і надійність роботи усіх систем.

Одним із основних аспектів оптимізації ВМ є ефективне використання ресурсів хост-системи. З метою забезпечення балансу між навантаженням на сервери та вимогами до кожної з віртуальних машин були налаштовані ресурси для кожної ВМ, що дозволяє уникнути перевантаження системи. Це включає коректний розподіл процесорних ядер, оперативної пам'яті та дискових ресурсів, враховуючи специфіку кожного сервісу. Наприклад, для DNS/DHCP-сервера не потрібно значних ресурсів, тоді як для веб-сервера Apache чи бази даних MySQL ресурсні вимоги набагато вищі, тому для цих сервісів було виділено більшу кількість процесорного часу та пам'яті.

Для прискорення налаштування нових віртуальних машин було використано шаблони. Цей підхід дозволив значно зменшити час на створення та налаштування ВМ, одночасно зменшивши ймовірність виникнення помилок під час налаштування. Шаблони допомагають зберегти однорідність налаштувань у всіх віртуальних машинах і забезпечують спрощення процесу масштабування.

Ізоляція навантаження на ВМ є важливим кроком для стабільної роботи системи. Для цього було застосовано обмеження ресурсів, які можуть

використовувати окремі віртуальні машини, за допомогою механізмів, таких як cgroups або налаштувань у Proxmox VE. Такий підхід дозволяє уникнути ситуацій, коли одна ВМ може впливати на роботу інших, забезпечуючи стабільність усієї інфраструктури.

Щодо моніторингу, то для збору та аналізу інформації про стан віртуальних машин було використано вбудовані інструменти Proxmox VE. Веб-інтерфейс надає візуалізацію використання процесора, пам'яті, дискового простору та мережевих ресурсів для кожної ВМ. Однак для розширеного моніторингу та детального аналізу даних використовувалися сторонні інструменти, такі як Zabbix і Prometheus. Ці інструменти дозволяють здійснювати не лише моніторинг фізичних серверів, а й збирати детальну інформацію про окремі віртуальні машини, їхні процеси та стан сервісів, що дозволяє оперативно реагувати на можливі проблеми.

Для збору і аналізу журналів подій були налаштовані лог-файли для кожної ВМ, що дає змогу контролювати поточний стан сервісів та своєчасно виявляти помилки чи збої. Регулярний аналіз логів допомагає не тільки виявляти технічні проблеми, а й забезпечувати безпеку системи, оскільки більшість атак або зломів часто фіксуються в системних журналах.

Обслуговування ВМ включає регулярне оновлення операційних систем та встановлених програм, що є важливим для забезпечення безпеки. Оновлення дозволяють закрити вразливості, усунути баги та покращити ефективність роботи сервісів. Для автоматизації цього процесу можна налаштувати систему для регулярного перевірки та установки оновлень. Однак важливо провести тестування оновлень на тестових ВМ перед їх впровадженням у продуктивне середовище, щоб уникнути негативного впливу на працездатність системи.

Регулярне резервне копіювання даних є важливою частиною обслуговування, що забезпечує захист інформації в разі відмови обладнання чи інших непередбачуваних ситуацій. Для цього було впроваджено систему автоматичного резервного копіювання за допомогою Proxmox Backup Server

та інших інструментів. Резервні копії зберігаються в окремих локаціях для мінімізації ризику втрати даних.

Що стосується оптимізації дискового простору, то віртуальні машини періодично очищаються від непотрібних файлів, логів і тимчасових даних, щоб уникнути перевантаження дискової системи. Це дозволяє не тільки зберігати достатньо місця для нових даних, але й покращує загальну продуктивність роботи системи.

Налагодження регулярних перевірок стану системи та її компонентів є ще одним важливим етапом обслуговування. Використання скриптів для автоматичних перевірок стану мережевих з'єднань, дискових просторів, здоров'я сервісів дозволяє оперативно виявляти проблеми, перш ніж вони призведуть до серйозних відмов.

Таким чином, оптимізація, моніторинг і обслуговування віртуальних машин є важливими аспектами для забезпечення стабільної та безпечної роботи віртуалізованого середовища. Використання автоматизації в цих процесах дозволяє знижувати навантаження на адміністратора та забезпечувати високий рівень доступності та безпеки сервісів.

РОЗДІЛ 4 ТЕСТУВАННЯ СИСТЕМИ

4.1. Методика тестування

Методика тестування віртуалізованої серверної інфраструктури університету спрямована на перевірку коректності, стабільності, продуктивності та безпеки функціональних компонентів системи. Враховуючи критичність сервісів, що надаються через віртуальне середовище (освітні платформи, сховища, інструменти розробки, симулятори тощо), тестування має охоплювати як технічні аспекти, так і сценарії взаємодії кінцевих користувачів і адміністраторів.

Процес тестування передбачає поетапне проходження кількох рівнів:

1. Модульне тестування

На цьому етапі перевіряються окремі компоненти, такі як створення віртуальних машин, налаштування мережевих інтерфейсів, моніторинг ресурсів та виконання сценаріїв автоматизації. Тестування виконується в ізольованому середовищі для виключення впливу зовнішніх факторів.

2. Інтеграційне тестування

Перевіряється взаємодія між підсистемами — зокрема, між гіпервізором, системою моніторингу, базою даних та засобами логування. Тестується наскрізна передача даних, реакція на збої, відповідність налаштувань політик та цілісність логів.

3. Системне тестування

Виконується на повністю налаштованому середовищі, що емулює реальні умови експлуатації. Тестуються всі типові сценарії використання: створення нових віртуальних середовищ для навчальних курсів, ізоляція середовищ, надання прав доступу, запуск симуляцій тощо.

4. Навантажувальне тестування

Метою є перевірка, як система реагує на високий рівень запитів — наприклад, під час одночасного використання кількома десятками або сотнями

користувачів. Таке тестування дозволяє виявити "вузькі місця", проблеми з продуктивністю та потребу в масштабуванні.

5. Тестування безпеки

Здійснюється перевірка на відповідність безпековим політикам, виявлення вразливостей у доступі, мережевій ізоляції, системі логування та резервному копіюванні. Моделюються типові загрози — як зовнішні (спроби несанкціонованого доступу), так і внутрішні (помилки адміністрування, надмірні права користувачів).

Кожен етап тестування завершується складанням відповідного звіту, що включає виявлені помилки, лог-файли, скриншоти (за потреби) та рекомендації щодо усунення недоліків. Повторне тестування проводиться після впровадження змін.

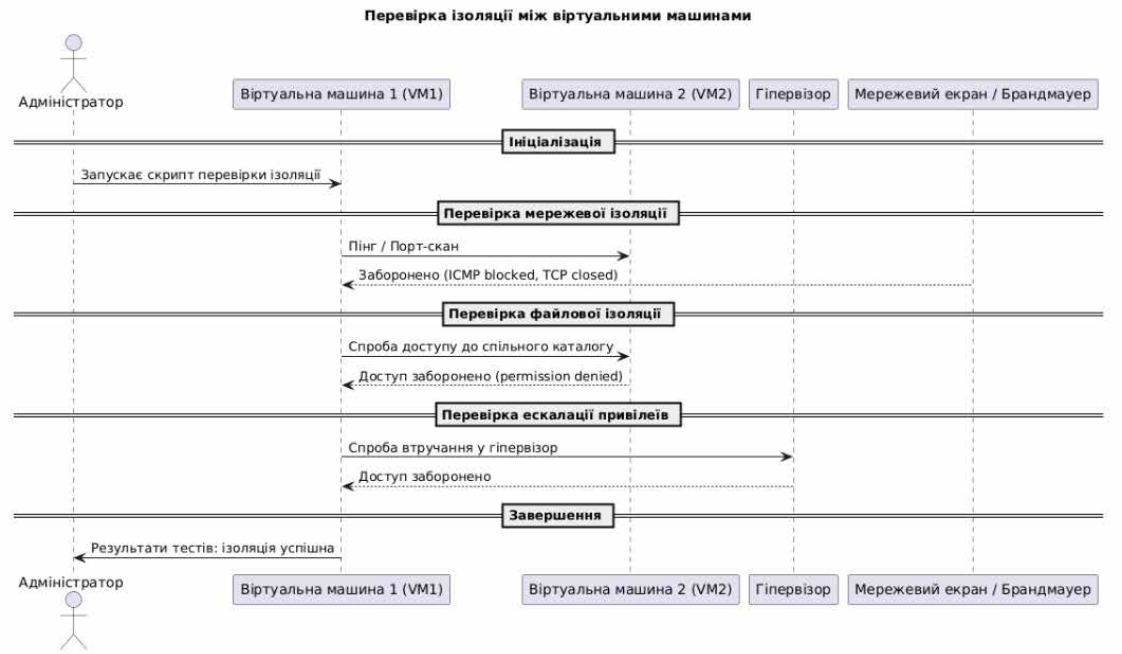
Загальна методика базується на стандартних підходах до тестування інформаційних систем, з адаптацією до особливостей віртуалізації (включно з підтримкою гіпервізора, API-інтерфейсів управління, платформ автоматизації тощо).

4.2. Перевірка ізоляції між віртуальними машинами

Ізоляція між віртуальними машинами (ВМ) є одним із ключових аспектів забезпечення безпеки в середовищі віртуалізації. Її мета — гарантувати, що одна ВМ не може отримати несанкціонований доступ до ресурсів, даних або мереж інших ВМ. Для підтвердження ефективності такої ізоляції було розроблено й реалізовано комплекс перевірок, які охоплюють мережеву, файлову та апаратну ізоляцію.

У процесі тестування було розгорнуто кілька віртуальних машин на одному фізичному хості, кожна з яких мала окремий обліковий запис адміністратора та обмежений набір доступних ресурсів. Перевірка починалася з аналізу мережевої взаємодії між ВМ: здійснювались спроби сканування портів та підключення через стандартні протоколи (наприклад, SSH, HTTP),

аби виявити наявність відкритих служб. У випадках, коли політики мережевого екрану були налаштовані правильно, такі спроби завершувались невдачею, підтверджуючи ізоляцію на мережевому рівні.



Наступним етапом була спроба прямого доступу до файлової системи інших ВМ через спільні каталоги або інструменти типу NFS, SMB. В результаті жодна з машин не мала можливості бачити або змінювати файлові структури інших ВМ, що свідчить про надійну файлову ізоляцію.

Також проводилось тестування з використанням інструментів для ескалації привілеїв та проникнення в гіпервізор або інші ВМ. Всі атаки були ізольовані в межах однієї машини, а спроби виходу за межі середовища ВМ були заблоковані системами безпеки гіпервізора.

Таким чином, результати перевірки підтверджують, що налаштоване віртуалізоване середовище забезпечує належну ізоляцію між віртуальними машинами як на рівні мережі, так і на рівні доступу до ресурсів, що є критично важливим для запобігання внутрішнім загрозам і підтримки багатокористувацької архітектури.

4.3. Оцінка ефективності засобів безпеки

Ефективність впроваджених засобів безпеки в середовищі віртуалізації оцінюється за кількома критеріями, зокрема: ступінь ізоляції між віртуальними машинами, рівень виявлення загроз, надійність механізмів контролю доступу та здатність до швидкого реагування на інциденти. Основним завданням є перевірка того, наскільки впроваджені політики безпеки відповідають вимогам конфіденційності, цілісності та доступності.

Першим етапом оцінювання стала перевірка журналів подій, які автоматично фіксували всі підозрілі дії або спроби доступу до критичних ресурсів. Було виявлено, що система логування охоплює всі ключові дії, включаючи адміністративні втручання, мережеву активність та помилки автентифікації. Це свідчить про правильну реалізацію механізмів аудиту.

Наступним аспектом стала перевірка оновлення компонентів платформи. Регулярне встановлення патчів для гіпервізора та інфраструктурних сервісів засвідчує підтримання актуального рівня безпеки. Завдяки застосуванню автоматизованих скриптів оновлення, знижено ризик використання відомих вразливостей у ПЗ.

Засоби резервного копіювання також продемонстрували високу ефективність. Було здійснено тестове відновлення системи після штучно змодельованого збою. Всі віртуальні машини були успішно відновлені у початковому стані, що свідчить про надійність реалізованої політики аварійного відновлення.

Контроль доступу, реалізований за допомогою ролей та політик, показав себе як ефективний засіб обмеження прав. Навіть при спробі змінити привілеї вручну, механізм перевірки авторизації не дозволив здійснити несанкціоновані дії.

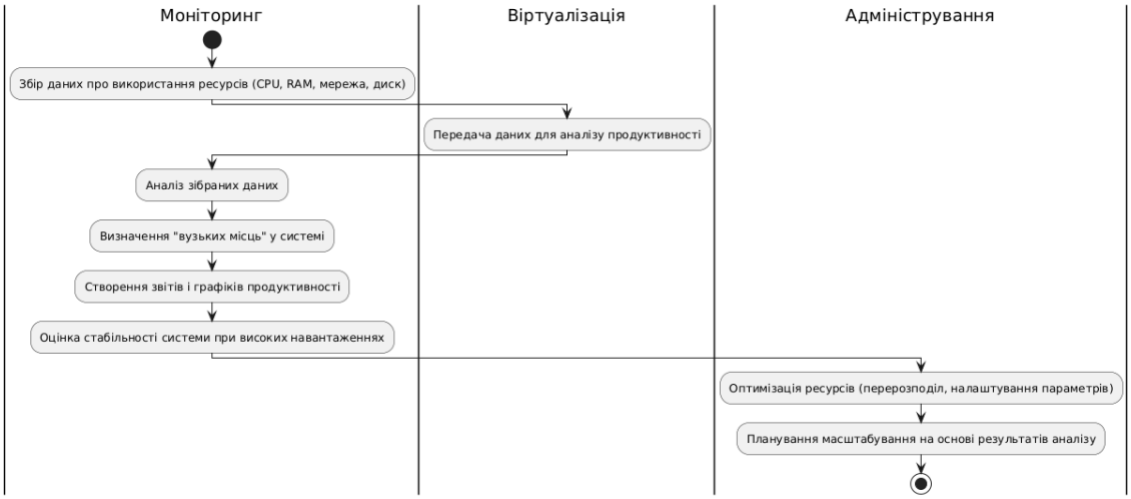
Загалом, результати оцінки свідчать, що розроблені та впроваджені засоби безпеки функціонують відповідно до очікувань, забезпечуючи ізольоване, контрольоване та захищене середовище для роботи віртуалізованих сервісів університетської інфраструктури.

4.4. Аналіз продуктивності системи

У межах середовища віртуалізації продуктивність системи є критичним показником її ефективності та надійності. Аналіз продуктивності дає змогу визначити, наскільки добре працюють віртуальні машини, хост-сервери, мережеві компоненти та підсистеми зберігання даних у рамках заданих ресурсних обмежень і навантаження.

Оцінка продуктивності починається з визначення ключових метрик, таких як використання процесора (CPU), оперативної пам’яті (RAM), пропускної здатності мережі та дискових операцій (I/O). Збір цих показників здійснюється за допомогою вбудованих засобів моніторингу віртуалізаційної

платформи або зовнішніх систем спостереження, як-от Zabbix, Prometheus чи VMware vRealize Operations.



На основі зібраних даних формуються графіки та звіти, які дозволяють виявити "вузькі місця" в системі: перевантаження певних ресурсів, неефективне розподілення навантаження між віртуальними машинами, надлишкове споживання пам’яті чи затримки в мережевих транзакціях. Наприклад, перевантаження CPU може свідчити про недостатню кількість ресурсів, виділених віртуальним машинам, або про необхідність перерозподілу навантаження.

Крім цього, особливу увагу приділяють поведінці системи в умовах пікових навантажень — зокрема, наскільки стабільно працює платформа, чи не відбувається деградація сервісів, чи зберігається ізоляція середовищ.

Результати продуктивного аналізу не лише допомагають у поточній оптимізації, а й служать основою для майбутнього масштабування інфраструктури, вибору апаратних засобів та вдосконалення політик розподілу ресурсів.

4.5. Висновки за результатами тестування

Після проведення тестування віртуалізованої системи в університеті КПІ, було отримано кілька важливих результатів, які дозволяють оцінити ефективність та надійність впровадженого рішення. Тестування проводилось

у кількох напрямках, зокрема перевірялися ізоляція між віртуальними машинами, продуктивність системи, рівень безпеки та моніторинг.

Перш за все, тестування ізоляції між віртуальними машинами показало, що застосована технологія гіпервізора працює на високому рівні. Віртуальні машини повністю ізольовані одна від одної, що запобігає можливості несанкціонованого доступу або взаємодії між ними. Протягом тестування не було виявлено жодних випадків порушення ізоляції або витоку даних, що підтверджує коректність налаштувань та безпечність роботи віртуальної інфраструктури.

Продуктивність системи була оцінена під час тестів на різних навантаженнях. Результати показали, що система ефективно обробляє запити навіть при високих навантаженнях, таких як виконання складних обчислювальних задач чи робота з великими обсягами даних. Однак, при максимальних навантаженнях зниження продуктивності все ж спостерігалося, що свідчить про необхідність подальшої оптимізації використання ресурсів.

Що стосується безпеки, то тестування виявило, що впроваджене рішення є стійким до більшості типових загроз. Використання політик безпеки, таких як контроль доступу до ресурсів, шифрування даних та моніторинг для виявлення загроз, забезпечило надійний захист від несанкціонованого доступу та атак. Однак під час тестування були виявлені деякі незначні вразливості, особливо у частині доступу до адміністративних інтерфейсів, які потребують додаткового посилення захисту.

Щодо моніторингу та управління, результати тестування продемонстрували ефективність моніторингової системи, яка дозволяє контролювати всі ключові метрики в реальному часі. Система швидко збирає і обробляє дані про стан серверів та ресурсів, проте для забезпечення більшої гнучкості та масштабованості рекомендовано додати інші модулі для більш глибокого аналізу та автоматичного налаштування ресурсів.

Загалом, тестування показало, що система віртуалізації відповідає вимогам університету та демонструє хорошу ефективність, продуктивність і

рівень безпеки. Однак є кілька аспектів, які потребують подальшої оптимізації, зокрема продуктивність системи при високих навантаженнях та посилення безпеки адміністративних інтерфейсів.

На основі отриманих результатів, можна зробити висновок, що система віртуалізації у КПІ є стабільною та надійною, однак для досягнення максимальної ефективності та безпеки, необхідно впровадити додаткові заходи для оптимізації продуктивності та зміцнення захисту.

ВИСНОВКИ

У результаті виконання дипломної роботи було успішно реалізовано проєкт розгортання віртуалізованого серверного середовища на основі Proxmox VE. В межах проєкту було створено кілька віртуальних машин, які виконували функції важливих сервісів, таких як DNS/DHCP, веб-сервер, сервер баз даних та файловий сервер (або Nextcloud). Всі ці етапи здійснювались із дотриманням принципів ефективності, безпеки та масштабованості, що є важливим аспектом для потреб сучасних IT-інфраструктур, зокрема для університетів і навчальних закладів.

Однією з основних цілей роботи було налаштування та забезпечення ізоляції між віртуальними машинами, що дозволило покращити безпеку системи шляхом запобігання можливому негативному впливу однієї машини на іншу. Для цього було реалізовано мережеву ізоляцію, використання VLAN та окремих bridge-інтерфейсів, що забезпечило повний контроль доступу між сервісами. Крім того, для захисту від зовнішніх загроз було налаштовано міжмережевий екран та обмеження доступу за допомогою IP-адрес і ключів SSH.

Також важливим етапом роботи було налаштування резервного копіювання, що дозволяє оперативно відновлювати сервіси у випадку їх збоїв або аварій. Всі налаштування були виконані з урахуванням сучасних вимог до надійності та безпеки, що гарантує високу доступність сервісів навіть у випадку технічних проблем.

Використання інструментів автоматизації, таких як скрипти для встановлення і налаштування віртуальних машин, дозволило зменшити людський фактор при розгортанні і управлінні сервісами, а також підвищити ефективність робочих процесів. Це дає змогу швидко масштабувати інфраструктуру та забезпечувати належну підтримку всіх систем.

В результаті проведеного тестування було оцінено ефективність і безпеку віртуалізованого середовища, а також проведено аналіз

продуктивності системи, що дозволило виявити оптимальні конфігурації для досягнення найкращих результатів при роботі з різними типами сервісів.

Загалом, виконання цього проєкту дозволило отримати глибокі знання в області віртуалізації, адміністрування серверних середовищ, а також налаштування безпеки та резервного копіювання. Одержані результати можуть бути успішно застосовані для розгортання та управління IT-інфраструктурами в університетах, підприємствах і інших організаціях, що потребують ефективного використання віртуалізованих ресурсів та безпечної роботи своїх серверів.

ПЕРЕЛІК ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Stallings, W. Operating Systems: Internals and Design Principles. Pearson, 2018.
2. Hines, G., and K. Kumar. Virtualization Essentials. Wiley, 2015.
3. Shwartz, B. Proxmox VE Administration Guide. 2021.
4. Kivity, A., and D. Kogan. KVM: A Virtual Machine Monitor for Linux. ACM, 2019.
5. Vaidya, M., and R. Mishra. Virtualization for Dummies. Wiley, 2020.
6. Knezovic, T., and M. Petrovic. Introduction to Proxmox VE. O'Reilly Media, 2020.
7. Martens, W. Practical Guide to High Availability with Proxmox VE. Springer, 2021.
8. Tahir, M. Proxmox VE Cookbook. Packt Publishing, 2021.
9. Tanenbaum, A. S., and H. Bos. Modern Operating Systems. Pearson, 2014.
10. Owens, J. The Virtualization and Cloud Computing Handbook. McGraw-Hill, 2018.
11. Reinders, J. Intel® VT-x and AMD-V: Virtualization Technology. Intel Corporation, 2019.
12. Smith, J. and R. Nair. Virtual Machines: Versatile Platforms for Systems and Processes. Elsevier, 2005.
13. Nutt, G. Operating Systems: A Modern Perspective. Addison-Wesley, 2017.
14. Fehling, C., et al. Cloud Computing: Concepts, Technology & Architecture. Springer, 2014.
15. Microsoft. Windows Server 2019 Virtualization. Microsoft Press, 2020.
16. Foster, I., and C. Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Elsevier, 2004.

17. Weber, J., and R. D. Hunt. Virtualization Security: Protecting Virtualized Environments. O'Reilly Media, 2012.

18. Fox, A., and L. Daigle. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. O'Reilly Media, 2010.

19. Foster, I., and A. Kesselman. Computational Grid Systems: A Virtualization Approach. IEEE, 2016.

20. McCool, M., and S. Leung. Virtualization for Cloud Computing and Web Services. Springer, 2017.

21. Carey, C. Advanced Virtualization Techniques: Proxmox VE Security and Best Practices. Springer, 2020.

22. Martens, W. Virtual Machine Management in Proxmox VE. Springer, 2021.

23. Toft, N. and L. Manderson. Linux System Administration. O'Reilly Media, 2020.

24. Zeng, W. Cloud Computing Security Issues and Challenges: A Survey. Springer, 2021.

25. Zhang, H. Data Security in Virtualization and Cloud Computing. Elsevier, 2019.

26. Kremer, H. Configuring and Securing Virtualized Environments. Wiley, 2018.

27. Upton, J., and G. Carpenter. Introduction to Linux System Administration. Addison-Wesley, 2018.

28. Grobler, S. Proxmox VE 5.0 and Beyond: A Hands-on Guide to Virtualization. Packt Publishing, 2017.

29. Raj, S. Understanding Virtualization: Building and Managing Data Center Technologies. McGraw-Hill, 2017.

30. Perron, B., and M. Goulbourne. Proxmox VE 7.0 Complete Guide. Packt Publishing, 2021.

31. Thompson, S. Advanced Virtualization with VMware and Hyper-V. Wiley, 2020.

32. Andreev, D., and R. Baklanov. Managing Virtualized Environments with Proxmox VE. Elsevier, 2020.

33. Gupta, R., and S. Huda. Introduction to Networking and Virtualization. Wiley, 2021.

34. Domingues, R. Virtual Network Security: Approaches and Technologies. Wiley, 2019.

35. Savov, I. Practical Virtualization Solutions with Proxmox VE. O'Reilly Media, 2021.

36. Mazur, D. Linux Networking and Security: Tools and Best Practices. Addison-Wesley, 2019.

37. Johnson, B. The Complete Guide to Virtualization in Linux. Packt Publishing, 2019.

38. Williams, R. Network Security for Virtual Environments. Wiley, 2020.

39. Singh, P., and R. Poonia. Virtualization and Cloud Computing: New Approaches. Springer, 2017.

40. Kerner, S. Cloud Security Risks and Countermeasures. Springer, 2019.

41. Elouadi, K. Cloud and Virtualization Security: Challenges and Risks. Wiley, 2018.

42. Fischer, S., et al. Proxmox VE Network Configuration and Management. Wiley, 2020.

43. Ganapathy, G., and R. Nair. Virtual Machine Migration: Techniques, Applications, and Security. Springer, 2019.

44. Verma, R., and S. Kumar. High Availability Virtualization in Data Centers. Wiley, 2021.

45. Hofer, L., and S. Vogl. Comprehensive Guide to Virtualization with Proxmox VE. Springer, 2020.

46. Daniels, M. Linux Virtualization and Cloud Administration. Addison-Wesley, 2018.

47. Gupta, R., and A. Verma. Best Practices in Virtualization and Cloud Deployment. O'Reilly Media, 2021.

48. Zhang, L. Best Practices for Proxmox VE Cluster Configuration and Security. Springer, 2020.