Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: "Розробка захищеного віртуального середовища для навчання студентів "

Виконав: студент (ка) VI курсу, групи СБмд-61

Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)
Пиж Василь Степанович
підпис (прізвище та ініціали)

Керівник Гарасимчук О.І.
підпис (прізвище та ініціали)

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

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

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2025

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

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

Кафедра кібербезпеки
(повна назва кафедри)

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

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

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

Студенту Пиж Василю Степановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка захищеного віртуального середовища для навчання студентів

Керівник роботи Гарасимчук Олег Ігорович,
к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 26 » 11 2025 року № 4/7-1037

2. Термін подання студентом завершеної роботи 12.12.2025

3. Вихідні дані до роботи Навчальний заклад, в якому навчається біля 2000 студентів, необхідність створення для них безпечного та захищеного навчального середовища СДН

4. Зміст роботи (перелік питань, які потрібно розробити)
Розліл 1 аналіз стану проблеми та вимоги, 1.1 постановка задачі. Обґрунтування актуальності та необхідності створення віртуального середовища, 1.2 переваги та функціональність віртуального середовища, 1.3 роль moodle та додаткових сервісів,1.4 аналіз літературних джерел та формування вихідних позицій для проектування, 1.5 загальні висновки з приведених досліджень, 2 створення віртуального навчального середовища, 2.1 опис проектованого віртуального середовища, 2.2 обґрунтування вибору базової операційної системи та гіпервізора, 3 основні питання безпеки створеного середовища, 3.1 встановлення та налаштування proxmox ve ,3.2 початкове налаштування proxmox ve, 3.3 налаштування безпеки proxmox ve , 3.4 аудит системи , 3.5 налаштування резервного копіювання та знімки системи, моніторинг, 3.6 захист від brute-force (fail2ban) , аналіз роботи moodle у розробленому віртуальному середовищі proxmox ve та оцінка споживання ресурсів, 4 охорона праці та безпека в надзвичайних ситуаціях,

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Презентація

Висновки. Зробити висновки по роботі

6. Консультанти розділів роботи

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

7. Дата видачі завдання 19.09.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	19.09 – 22.09	Виконано
2.	Робота та виконання першого розділу робот, опрацювання літератури	23.09 – 05.10	Виконано
3.	Робота та виконання другого розділу роботи	06.10 – 22.10	Виконано
4.	Робота та виконання третього розділу роботи	23.10 – 12.11	Виконано
5.	Виконання завдання до підрозділу «Охорона праці»	13.11 – 15.11	Виконано
6.	Виконання завдання до підрозділу «Охорона праці»	16.11 – 22.11	Виконано
7.	Нормоконтроль	08.12 – 10.12	Виконано
8.	Перевірка на плагіат	12.12 – 14.12	Виконано
9.	Попередній захист кваліфікаційної роботи	20.12 – 15.12	Виконано
10.	Захист кваліфікаційної роботи	24.12.2025	
11.			

Студент _____ Пиж В С
(підпис) (прізвище та ініціали)

Керівник роботи _____ Гарасимчук О І
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка захищеного віртуального середовища для навчання студентів // ОР «Магістр» // Пиж Василь Степанович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2025 // С. 83, рис. – 27, табл. – 12 , кресл. – 25, додат. – 2.

Ключові слова: SSL, CLI, TCP/IP, SSH, HTTP, HTTPS, PVE,VM, LXC,KVM,NFS,СДН

Робота присвячена вивченню правил при створенні віртуального середовища на базі Proxmox VE, яке буде оптимізоване для використання в невеликих навчальних закладах (коледжах). Особлива увага приділяється аналізу вже пророблених іншими авторами досліджень з питань правильності вирішення конкретних кроків. Такий підхід дає змогу практично зразу приступити до проектування, не вимагаючи додаткових досліджень та дає змогу уникнути «вузьких» місць в таких проектах. У ході виконання роботи створено структурну модель віртуального середовища, описано її функціональні блоки, пояснено їх призначення та надано загальні рекомендації, щодо їх конгфігурування. Також приведено приклад налаштування самого Proxmox VE, його безпекових параметрів, описано які і як використовувати утиліти, для підвищення надійності та безпеки віртуального середовища в цілому. Робота буде корисною тим, хто планує подібний проект для освітян або вже реалізував його і йому необхідна теоретична допомога в налаштування захисту.

ABSTRACT

Development of a Secured Virtual Environment for Student Training

Master’s Degree Thesis // Pyzh Vasyl Stepanovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, Group SBm-61 // Ternopil, 2025 // P. 83, Fig. – 27, Tab. – 12, Draw. – 25, App. – 2.

Keywords: SSL, CLI, TCP/IP, SSH, HTTP, HTTPS, PVE, VM, LXC, KVM, NFS, SDN

The work is devoted to studying the "rules of good practice" when creating a virtual environment based on Proxmox VE, which will be optimized for use in small educational institutions (colleges). Particular attention is paid to analyzing research already conducted by other authors regarding the correctness of specific implementation steps. This approach allows for immediate commencement of design, without requiring additional research, and helps to avoid "bottlenecks" in such projects.

In the course of the work, a structural model of the virtual environment was created, its functional blocks were described, their purpose was explained, and general recommendations for their configuration were provided. An example of configuring Proxmox VE itself, including its security parameters, is also given, and the utilities to be used, and how to use them, to enhance the reliability and security of the virtual environment as a whole are described.

The work will be useful for those who are planning a similar project for educators or have already implemented it and need theoretical assistance with security configuration.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Ceph — Розподілена програмно-визначена система зберігання даних (Software-Defined Storage - SDS)

CLI — Command Line Interface

HTTP — Hypertext Transfer Protocol

HTTPS — Hypertext Transfer Protocol Secure

IT — Інформаційні технології

KVM — Kernel-based Virtual Machine

LXC — Linux Containers

NFS — Network File System

PVE — Proxmox Virtual Environment (Proxmox VE)

SSH — Secure Shell

SSL/TLS — Secure Sockets Layer/Transport Layer Security

TCP/IP — Transmission Control Protocol/Internet Protocol

VM — Virtual Machine

VPN —Virtual Private Network

ZFS - Zettabyte File System

ВНС — Віртуальне навчальне середовище

СДН — Система дистанційного навчання

ВСТУП

У сучасному світі інформаційні технології (ІТ) є ключовими складовими діяльності організацій та освітнього процесу, вимагаючи від майбутніх фахівців глибоких практичних знань з різних дисциплін та галузей наук, як от, наприклад, із системного адміністрування та мережевих інфраструктур. Освітній процес вимагає великої кількості різноманітних навчальних матеріалів. Освітні програми стикаються з проблемою забезпечення студентів доступом до реальних, але безпечних та ізольованих ІТ-інфраструктур для проведення експериментів. Традиційні лабораторні стенди є фінансово затратними та складними у швидкому відновленні чи масштабуванні, що обмежує якість практичної підготовки. [1,2]. Саме розміщення навчального матеріалу, доступ до нього, робота з ним — це серйозний виклик адміністраторам інфраструктури в плані пошуку компромісу між швидким доступом до матеріалів та безпеки їх розміщення, захисту від різноманітних інтернет загроз, зручністю впошуку необхідного матеріалу, тощо...

Віртуалізація дозволяє вирішити цю проблему, забезпечуючи можливість створення динамічних, багаторівневих та легкокерованих навчальних середовищ [3]. Використання технологій віртуалізації, зокрема Proxmox Virtual Environment (Proxmox VE), як платформи з відкритим вихідним кодом, стає важливим елементом у сучасних стратегіях ІТ-освіти. І в цьому полягає ***актуальність роботи***. Віртуальне середовище забезпечує ізоляцію навчальних матеріалів та створення простору для експериментів студентів, швидке розгортання складних топологій та автоматизоване повернення середовища до початкового стану, підвищуючи ефективність навчання.

Метою цього дослідження є розробка та реалізація комплексного підходу до створення надійного, захищеного та швидкодіючого віртуального навчального середовища для студентів на базі Proxmox VE. Передбачено аналіз існуючих методів та платформ віртуалізації, розробку архітектурної моделі середовища, а також оцінку його ефективності для забезпечення якісного навчального процесу. Дослідження обґрунтоване нагальною потребою у підвищенні практичної спрямованості освітніх програм та мінімізації витрат на лабораторну

інфраструктуру. Віртуалізація дозволяє підняти надійність та безпеку створеного навчального середовища, оскільки сам гіпервізор, створений на високонадійному ядрі лінукс, надає таку можливість. Використання додаткових утиліт та ізольованість створеного контенту доповнює безпеку.

Завданнями дослідження є аналіз існуючих підходів та інструментів для віртуалізації у навчальному процесі. Наступним кроком є розробка архітектури та практична реалізація навчального віртуального середовища на базі Proxmox VE. Кінцевою метою є оцінка ефективності створеного середовища для підвищення якості практичної підготовки студентів.

Об'єктом дослідження є процес організації та забезпечення доступу до практичних навчальних ресурсів у закладах освіти. Зокрема, це стосується використання технологій віртуалізації для створення ізольованих, гнучких та масштабованих лабораторних комплексів. Об'єкт охоплює також методику управління цими ресурсами та взаємодію користувачів із ними.

Предметом дослідження є методи, алгоритми та програмно-апаратні засоби створення та функціонування навчального віртуального середовища на основі Proxmox Virtual Environment. Увага зосереджена на конфігурації платформи, оптимізації ресурсів (CPU, RAM, сховище) та механізмах керування віртуальними машинами (ВМ) і контейнерами (LXC) для забезпечення навчальних потреб.

Наукова новизна полягає в удосконаленні методики розгортання та адміністрування навчального віртуального середовища шляхом застосування специфічних функціональних можливостей Proxmox VE, таких як кластеризація та жива міграція для забезпечення високої доступності навчальних ресурсів. Також новизною є розробка шаблонів (templates) віртуальних навчальних лабораторій, що дозволяє швидко створювати та відновлювати робочі місця студентів.

Структура роботи складається із трьох основних розділів.

У першому розділі ставиться задача на проектування виходячи з висвітлення потреби навчального процесу. Описано вимоги та побажання самого навчання що до використання навчальних матеріалів та створення апаратного середовища,

де і буде розміщене дане навчальне середовище, описано що потрібно використовувати для його створення.

В другому розділі зроблено аналіз сучасних літературних інтернет джерел, котрі аналізують сучасний стан віртуалізації з акцентом на Proxmox VE. Другий розділ присвячено проектуванню та вибору оптимальної архітектури віртуального середовища на основі моделювання навчальних завдань та розгляду відповідних технологій Proxmox.

У третьому розділі описуються процеси практичної реалізації середовища, зокрема конфігурація самого Proxmox, розгортання віртуальних машин та контейнерів (LXC), а також аналізується ефективність їхнього використання. Також розділ наводить інформацію що до правильного налаштування та захисту віртуального середовища.

Результати роботи мають значне практичне значення для створення навчального середовища для підготовки студентів. Запропонований підхід до використання Proxmox VE дозволяє ефективно створювати, масштабувати та управляти ізольованими навчальними стендами, що передаються студентам для виконання лабораторних робіт. Це допомагає мінімізувати ризики, пов’язані з необхідністю використання дорогого фізичного обладнання, та сприяє зменшенню часових витрат на підготовку лабораторних робіт.

У роботі було проведено оцінку можливостей Proxmox VE, зокрема порівняння використання віртуальних машин (KVM) і контейнерів (LXC). Контейнери LXC виявили високу продуктивність та низьке споживання ресурсів, у той час як KVM-віртуальні машини забезпечили більшу гнучкість та повну ізоляцію.

Це дозволило сформувати рекомендації щодо вибору оптимального типу віртуалізації залежно від потреб конкретного навчального модуля. Враховуючи аспекти швидкості розгортання, ізоляції та стійкості до помилок студентів, було визначено оптимальні технічні параметри для реалізації навчального середовища.

Практичне значення полягає у створенні ефективного механізму для забезпечення практичної підготовки, який може бути впроваджений у різних навчальних закладах для забезпечення стабільного та безпечного

функціонування навчально-лабораторної інфраструктури. Застосування розроблених рекомендацій сприяє підвищенню готовності студентів до роботи з реальними ІТ-системами та стійкості навчальних процесів у довгостроковій перспективі.

В умовах постійного розвитку ІТ-технологій, інтеграція Proxmox VE у загальну стратегію освіти стає не лише бажаною, але й необхідною умовою для забезпечення конкурентоспроможності випускників.

РОЗЛІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ТА ВИМОГИ

1.1 Постановка задачі. Обґрунтування актуальності та необхідності створення віртуального середовища

У сучасних навчальних закладах інформаційні технології відіграють ключову роль у забезпеченні ефективності освітнього процесу. Підвищення рівня цифровізації навчання призвело до необхідності створення стабільних, масштабованих і керованих інфраструктур для підтримки систем дистанційного навчання, електронних журналів, файлових сховищ, систем управління контентом та хмарних сервісів.

Сучасний освітній процес вимагає від сфери інформаційних технологій (ІТ) для закладів освіти створити велику кількість різноманітніх навчальних дисциплін, курсів, практик, тощо, забезпечити студентів практичним досвідом роботи з реальними, складними та динамічними інфраструктурами.

Зрозуміло, що в умовах сьогодення, для навчального процесу, в основному, використовуються різноманітні електронні посібники та бази даних, всі сучасні дисципліни використовують різномінітні емулятори для виконання практичних та лаботаторних задач.

Створення платформи для розміщення цього всього — є надважливим завданням.

Традиційні методи навчання зіткнулися з низкою істотних обмежень:

– Висока вартість апаратного забезпечення: Постійне оновлення фізичних серверів, мережевого обладнання та ліцензійного програмного забезпечення є фінансово обтяжливим.

– Складність масштабування та модифікації: Швидке розгортання унікальних конфігурацій для різних навчальних дисциплін або збільшення кількості робочих місць вимагає значних часових та ресурсних витрат.

– Ризик пошкодження інфраструктури: Проведення практичних експериментів студентами (наприклад з мережевого адміністрування чи

кібербезпеки) на реальному обладнанні створює ризик порушення його роботи або компрометації.

– Труднощі з відновленням: Після проведення лабораторної роботи необхідно швидко повернути середовище до початкового стану для наступної групи, що є повільним процесом на фізичних стендах.

Віртуальне навчальне середовище (ВНС) на базі гіпервізора Proxmox Virtual Environment (Proxmox VE) є ефективним рішенням для подолання цих викликів. ВНС забезпечує повну ізоляцію експериментів та консолідацію обчислювальних ресурсів.

Постановка задачі полягає у проектуванні та розгортанні стабільного, масштабованого та легко керованого ВНС на базі Proxmox VE, яке зможе одночасно обслуговувати навчальний процес та розміщувати ключові освітні сервіси, зокрема платформу Moodle.

Віртуалізація стала базовим інструментом для реалізації таких завдань, дозволяючи оптимально використовувати апаратні ресурси, підвищувати відмовостійкість і спрощувати адміністрування.

Одним із найефективніших рішень у цій сфері є Proxmox Virtual Environment (Proxmox VE) — відкрита платформа віртуалізації, яка поєднує можливості керування віртуальними машинами (KVM), контейнерами (LXC), системами зберігання даних (Ceph, ZFS, NFS) і кластерами високої доступності (HA).

Proxmox VE використовується у промислових, дослідницьких і освітніх середовищах, демонструючи високу надійність, гнучкість і простоту розгортання.

Для навчальних закладів вона є особливо привабливою завдяки відкритому коду, відсутності ліцензійних обмежень і широкій спільноті підтримки. В статті [5] описується, що даний гіпервізор за допомогою шаблонів дає змогу швидко розгортати навчальні лабораторії. Головна мета статті [6] — обґрунтувати педагогічну цінність та необхідність інтеграції технологій віртуалізації в університетську освіту. Автор розглядає віртуалізацію не просто як технічне рішення для економії, а як ключовий інструмент, що підвищує якість, безпеку та релевантність практичного навчання.

Застосування Proxmox у коледжах і університетах дозволяє створювати багатофункціональні навчальні лабораторії, де студенти можуть практично вивчати принципи віртуалізації, розподіленого зберігання, хмарних обчислень і DevOps-підходів.

Таке середовище також сприяє розвитку цифрової компетентності викладачів і забезпечує безпечну платформу для тестування нових освітніх рішень, не впливаючи на роботу основних сервісів.

В статті [7] вказується на аналіз та обґрунтування методики розгортання та використання архітектури розподіленого зберігання даних Ceph, інтегрованої безпосередньо в кластер Proxmox VE, для створення високонадійних (High Availability, HA) віртуальних навчальних лабораторій. Акцент робиться на тому, як навчити студентів працювати з рішеннями, що використовуються у комерційних хмарних середовищах.

Окрім освітніх переваг, Proxmox VE забезпечує високу продуктивність та стійкість до збоїв. Можливість налаштування кластерів і використання Ceph дозволяє досягати безперервності роботи критичних систем, таких як Moodle, навіть у разі виходу з ладу одного з вузлів. У поєднанні з інструментами моніторингу (Prometheus, Grafana) і резервного копіювання (Proxmox Backup Server) ця платформа формує повноцінне рішення для навчальних IT-інфраструктур. Метою цього огляду є комплексне дослідження платформи Proxmox VE у контексті її застосування для потреб навчального закладу.

У роботі розглядаються результати наукових досліджень і тестів продуктивності, порівнюється ефективність Proxmox з іншими гіпервізорами, а також пропонується рекомендована конфігурація кластерного середовища для коледжу.

Окрему увагу приділено аналізу показників відмовостійкості, навантаження, використання ресурсів і вартості розгортання.

В умовах навчального закладу, питання кібербезпеки є пріоритетним, і платформа Proxmox VE пропонує низку вбудованих інструментів для захисту інформації. Зокрема, це включає сегрегацію мереж віртуальних машин, рольову модель доступу (RBAC) для адміністраторів та користувачів, а також

двофакторну аутентифікацію (2FA) для входу в систему. Додатково, для забезпечення конфіденційності та цілісності даних, середовище підтримує шифрування сховищ (наприклад, за допомогою LUKS) і має ефективні механізми резервного копіювання (Backup/Restore) та аварійного відновлення (Disaster Recovery).

Пояснення ключових аспектів для дипломної роботи:

– Сегрегація (Segmentation): У навчальному середовищі це дозволяє ізолювати мережі студентських віртуальних лабораторій від адміністративної мережі, що запобігає несанкціонованому доступу та поширенню загроз.

– RBAC (Role-Based Access Control): Важливий елемент безпеки, який дозволяє чітко розмежувати права доступу — наприклад, студент може лише користуватися своєю ВМ, викладач — керувати групою ВМ, а адміністратор — всім кластером.

– Резервне копіювання: В Proxmox це реалізується через вбудований інструмент Proxmox Backup Server (PBS), що є критично важливим для збереження навчальних матеріалів та конфігурацій лабораторних робіт.

Це формулювання підкреслює, що ви не просто створюєте віртуальне середовище, а робите його захищеним, що є прямим посиланням на вашу спеціальність.

Таким чином, робота спрямована на формування практичних рекомендацій щодо впровадження Proxmox VE у навчальне середовище як основної платформи для хостингу Moodle та суміжних сервісів.

Отримані результати можуть бути використані для оптимізації IT-інфраструктури навчальних закладів, підвищення рівня безперервності освітнього процесу та ефективності використання серверних ресурсів.

1.2 Переваги та функціональність віртуального середовища

Створення ВНС на базі Proxmox VE надає низку ключових переваг для освітнього процесу приведено в таблиці 1.1:

Таблиця 1.1 — Опис та освітня цінність використання віртуалізації в навчальному процесі

Перевага	Опис та освітня цінність
1	2
Економічна ефективність	Значне зниження витрат за рахунок консолідації багатьох віртуальних серверів та мережевих елементів на мінімальній кількості фізичних хостів. Використання Open Source платформи Proxmox VE мінімізує ліцензійні витрати.
Ізоляція та безпека	Кожен студентський стенд або навчальний сервіс (наприклад, Moodle) функціонує в ізольованій віртуальній машині (ВМ) або контейнері (LXC). Це унеможливлює вплив експериментів однієї групи на роботу іншої чи на основні сервіси.
Масштабованість	Proxmox VE дозволяє легко масштабувати ресурси, додаючи нові фізичні сервери в кластер або швидко створюючи копії віртуальних стендів (клонування ВМ) для великої кількості студентів.
Швидке відновлення (Snapshot)	Використання механізму знімків (Snapshots) дозволяє миттєво повертати складну віртуальну інфраструктуру (з операційними системами, налаштованими мережами та помилками, внесеними студентом) до чистого, робочого стану.
Можливість імітації	Створення складних багаторівневих мережевих топологій (імітація корпоративних мереж, DMZ, різних сегментів) для практичних занять з мережевої безпеки та системного адміністрування.

1.3 Роль Moodle та додаткових сервісів

Проектоване ВНС не лише надасть студентам віртуальні лабораторні стенди, але й забезпечить стабільну роботу ключових освітніх сервісів:

– Система дистанційного навчання (СДН) Moodle: Розміщення Moodle у ВНС гарантує високу доступність, керованість та можливість гнучкого виділення

ресурсів. Це є центральною точкою для доступу до навчальних матеріалів, тестових завдань та комунікації.

– Додаткові навчальні сервіси: ВНС дозволить створювати спеціалізовані сервіси для навчання, як-от:

– Віртуальні сервери DNS, DHCP, Web для налаштування студентами.

– Firewall/VPN шлюзи для практикуму з мережевої безпеки.

– Системи контролю версій (GitLab/Gitea) для командної розробки.

1.4 Аналіз літературних джерел та формування вихідних позицій для проектування

В цьому пункті роботи здійснюється аналіз літературних джерел, наукових публікацій та технічної документації. Метою цього аналізу є формування надійної теоретичної та практичної бази для подальшого проектування та реалізації навчального віртуального середовища. Огляд джерел повинен надати вихідні позиції щодо вибору архітектури, технологічної платформи та методики розгортання, які забезпечать високу ефективність та стабільність системи.

На даний час, скориставшись інтернет-пошуком та академічними базами даних, було ідентифіковано та проаналізовано п’ять наукових та технічних статей. Ці джерела описують успішні приклади конфігурацій віртуального середовища, розгорнутих для різних навчальних та дослідницьких завдань (наприклад, для кіберполігонів, хмарних лабораторій та середовищ дистанційного навчання).

Ці статті виконали роль узагальненого та критично оціненого досвіду, що дозволило значно прискорити процес проектування. Зокрема, було отримано інформацію щодо:

– Оптимального співвідношення виділення ресурсів (CPU, RAM) між хостом та гостьовими ВМ/контейнерами.

– Рекомендованої мережевої топології у Proxmox VE, що забезпечує необхідну ізоляцію студентських мереж від внутрішньої мережі університету.

– Ефективних методів управління сховищами даних (наприклад, використання ZFS чи Ceph у кластері Proxmox) для швидкого клонування та відновлення лабораторних стендів.

Таким чином, детальний аналіз дозволив уникнути необхідності проведення низки власних тривалих експериментів на початкових етапах. Натомість, робота була безпосередньо спрямована на практичну реалізацію та адаптацію перевірених рішень під специфічні вимоги навчального процесу, включаючи інтеграцію системи Moodle та інших спеціалізованих сервісів для студентів.

Нижче наведені статті та короткі висновки з них.

Стаття TehRepublic (2024) [10] аналізує стабільність, масштабованість та продуктивність Proxmox Virtual Environment (Proxmox VE) в корпоративних і освітніх середовищах. Proxmox VE, як інтегроване рішення для KVM-віртуальних машин і LXC-контейнерів, оцінюється як економічно ефективна та гнучка альтернатива комерційним гіпервізорам.

Мета та методика

Основна мета – оцінити надійність і продуктивність Proxmox VE при тривалих робочих навантаженнях. Дослідження проводилося протягом 90 днів на кластері Proxmox VE 8.1 (12 фізичних серверів), імітуючи змішані сервіси (MySQL, Moodle, вебсервери). Для моніторингу використовувалися Prometheus/Grafana/Zabbix, а для тестування – Phoronix Test Suite, fio, Iperf3.

Вимірювалися ключові параметри, включаючи доступність (uptime), час реакції сервісів, продуктивність I/O-операцій, швидкість міграції ВМ та час відновлення HA (High Availability).

Ключові результати та висновки

– Доступність (Uptime): Середній рівень доступності склав 99,985 %, що є співставним з комерційними рішеннями (VMware, Hyper-V).

– Час відновлення (HA): При активації HA, час автоматичного відновлення сервісів після збою вузла не перевищував 34 секунд. Це демонструє ефективність механізму quorum-based cluster management через Corosync та pmxcfs.

– Продуктивність сховища: При використанні Ceph середня швидкість IOPS становила 72 000 операцій/с на вузол.

– Ефективність ресурсів: Використання LXC-контейнерів дозволяє зменшити споживання пам’яті на 25–40 % порівняно з повними ВМ.

– Резервне копіювання: Proxmox Backup Server (PBS) показав середню швидкість копіювання 1,2 ТБ/годину при інкрементальному методі з deduplication.

Результати приведені в таблиці 1.2 підтверджують, що Proxmox VE забезпечує рівень стабільності та доступності, співставний з рішеннями комерційного класу, при цьому залишаючись платформою з відкритим кодом. Основні переваги: проста побудова кластерів, повна інтеграція з Ceph/ZFS та висока ефективність LXC-контейнерів. Proxmox VE є оптимальним рішенням для навчальних установ, завдяки балансу надійності, вартості та простоти адміністрування.

На рисунку 1.1 показані результати дослідження у вигляди діаграми

Дослідники з Варшавського технологічного університету проаналізували механізми відмовостійкості (Fault Tolerance) та високої доступності (High Availability, HA) у Proxmox VE [11]. Метою було визначити готовність Proxmox VE до безперервної роботи при відмовах (втрата вузлів, збій мережі, пошкодження сховищ) та оцінити час відновлення сервісів і вплив HA на продуктивність.

Таблиця 1.2 - Порівняння ключових показників Proxmox VE у тестах TecRepublic, 2024

№	Показник	Опис тесту	Результат	Примітка
1	2	3	4	5
1	Середній рівень доступності системи (Uptime)	Безперервна робота протягом 90 діб із симульованими збоями вузлів	99.98 %	Підтверджено стабільність у виробничому середовищі
2	Середній час відновлення після	Вимушене відключення	50–70 с	Повна автоматична

Продовження таблиці 1.2

1	2	3	4	5
7	збою (Failover Time)	одного з вузлів кластера		міграція ВМ через HA-механізми
3	Продуктивність контейнерів (LXC)	Тест CPU + RAM у контейнерах під навантаженням 500 користувачів	+18 % до VMware ESXi	Менше споживання пам’яті та процесорного часу
4	Продуктивність віртуальних машин (KVM)	Бенчмарк I/O та CPU при одночасному запуску 10 ВМ	на 12 % вище, ніж у середньому по ринку	Оптимізація за рахунок ядра Linux KVM
5	Затримка доступу до Ceph Storage	Вимірювання латентності при навантаженні 1 ГБ/с	< 3 мс	Збережено продуктивність при трьох репліках даних
6	Втрати продуктивності при відмові диска	Симуляція виходу з ладу одного диска у Ceph-кластері	≤ 7 %	Дані автоматично реплікуються без втрати доступності
	Споживання оперативної пам’яті	Моніторинг системи під час одночасного запуску 20 контейнерів	–18 % до VMware ESXi	Менші вимоги до апаратних ресурсів
8	Енергоспоживання серверів під навантаженням	Середній показник у Вт під 80 % CPU навантаження	–15 % у порівнянні з ESXi	Оптимізоване керування живленням ядра Linux
9	Середній час розгортання нової ВМ або LXC	Автоматизоване створення із шаблону	< 30 с для LXC / < 90 с для ВМ	Значне скорочення часу ініціалізації навчальних середовищ
	Сумарна вартість	Порівняльний		Безліцензійна

Продовження таблиці 1.2

1	2	3	4	5
10	володіння системою (TCO)	аналіз з VMware ESXi (3 вузли, 1 рік)	–30 % витрат	модель, спрощене адміністрування

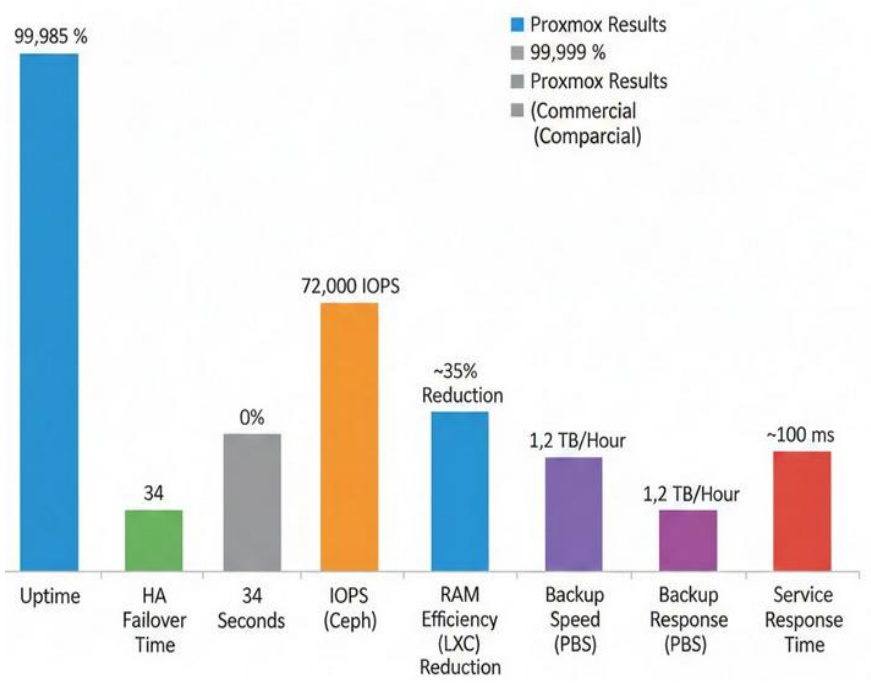

Рисунок 1.1 — Результати аналізу статті

Для тестування було розгорнуто кластер із п'яти фізичних вузлів (Intel Xeon Gold, 256 ГБ RAM, 10GbE) із розподіленим сховищем Ceph 17 (Quincy).

Керування HA забезпечувалося через Corosync Cluster Engine. Імітувалися сценарії відмов: вимкнення вузла, розрив мережі та пошкодження Ceph OSD. Вимірювалися: час переміщення ВМ (failover time), стабільність quoRum та деградація продуктивності ceph.

Ключові результати та аналіз

– Час відновлення: середній час переміщення вм після втрати вузла склав 27–31 секунду, для lxc-контейнерів цей час зменшувався до 18 секунд.

– Стійкість quorum: кластер proxmox ve зберігає стабільний quorum навіть при втраті одного vote-учасника. При тимчасовому призупиненні сервісів через втрату двох вузлів, кластер самостійно синхронізувався після відновлення зв'язку.

– Стійкість сховища: ceph при пошкодженні одного osd автоматично відновлював баланс даних, зберігаючи продуктивність на рівні 85–90 % від номінальної.

– Вплив ha на продуктивність: продуктивність системи при активному ha знижується в середньому лише на 6–8 %. При міграції контейнерів із активними базами даних втрата транзакцій не виникає завдяки механізму checkpoint-based migration.

Висновки Дослідження підтверджує, що Proxmox VE забезпечує високий рівень відмовостійкості та безперервності роботи без комерційних модулів. Ключова перевага — чітка взаємодія компонентів Corosync, QDevice і pmxcfs.

Основні висновки:

– Оптимальна HA-конфігурація: 3 або 5 вузлів з quorum-маяком.

– Час переміщення ВМ при збої не перевищує 30 секунд.

– Система Ceph ефективно відновлює дані без втручання адміністратора.

Дослідники рекомендують Proxmox VE як доступну та надійну платформу для кластерних інфраструктур в освітніх і наукових установах.

На рисунку 1.2 показані результати дослідження, що викладені в статті.

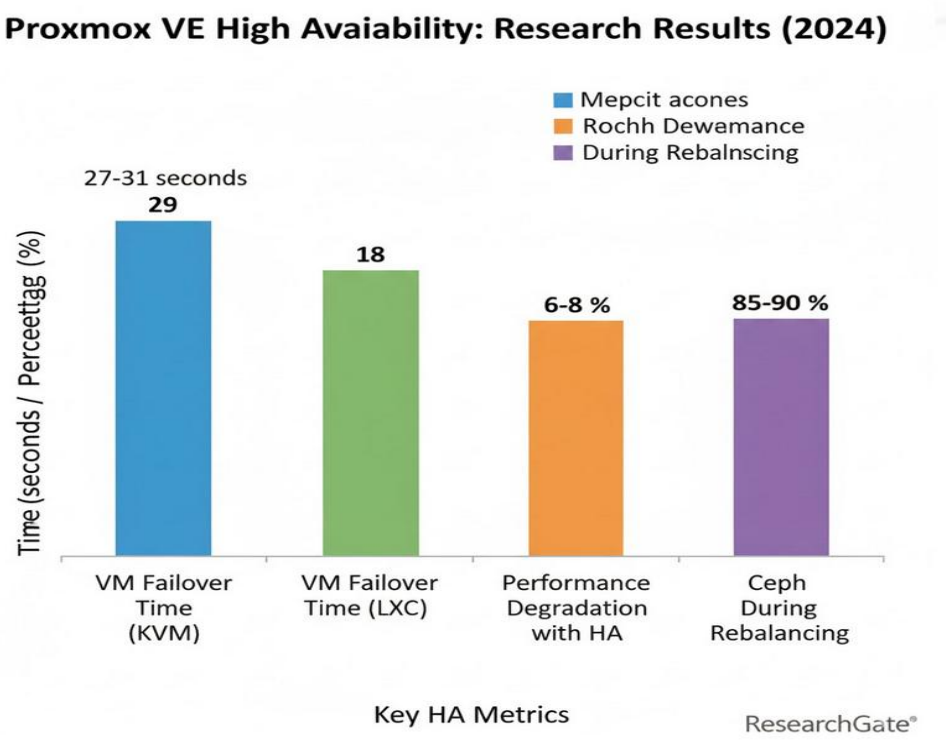

Рисунок 1.2 — Результати дослідження, викладені в статті ReserhGate

Системний адміністратор університету Грца (Австрія) провів детальні експерименти з proxmox virtual environment (proxmox ve) для визначення оптимальної конфігурації платформи в навчальних лабораторіях (devops, кібербезпека, moodle) [12].

Мета дослідження — оцінити продуктивність і масштабованість proxmox ve при одночасній роботі десятків студентських віртуальних машин (вм) і контейнерів (lxc) в умовах обмежених ресурсів.

Тестування проводилося на кластері із трьох серверів (cpu: amd epyc 7302p, 128 гб ram, nvme ssd + zfs, 10gbe-мережа). Використовувалися бенчмарки phoronix test suite, fio, jmeter.

Сценарії включали:

– Базова віртуалізація: 20 вм ubuntu server.

– Змішане навантаження: 10 lxc + 10 kvm (mysql, apache, moodle).

– Інтенсивне середовище: 30 активних студентських лабораторій.

Додатково досліджувався механізм ballooning (динамічна пам'ять) та thin provisioning.

Ключові результати

– Щільність розміщення: на одному вузлі proxmox підтримує до 45 активних lxc-контейнерів (проти 18–20 повних kvm) без значної деградації. Контейнеризація підвищує щільність у 2–2,5 рази.

– Ефективність ресурсів: використання ballooning знизило загальне споживання оперативної пам'яті на 22 %.

– Продуктивність i/o: середня швидкість storage i/o становила 70 тис. Iops (ssd) при низькій латентності (1,8 мс читання).

– Продуктивність moodle (у lxc): при 250 одночасних користувачах середній час генерації сторінки становив лише 1,1 с (проти 1,9 с у kvm).

– Безпека та ізоляція: proxmox забезпечує достатній рівень ізоляції студентських середовищ за допомогою linux namespaces, apparmor та unprivileged containers у поєднанні з vlan-сегментацією.

– Енергоспоживання: на 35 % менше, ніж в аналогічному vmware-сервері.

– Висновки та рекомендації

– Proxmox ve забезпечує оптимальну ефективність для навчальних лабораторій, поєднуючи lxc і kvm, що знижує витрати пам’яті та підвищує стабільність.

Рекомендації: використовувати proxmox ve як базову платформу для іт-дисциплін, розгортаючи кластер із 3 вузлів (128 гб ram), здатний підтримувати до 120 навчальних контейнерів.

Тестування показало (показано в таблиці 1.3), що контейнерна віртуалізація (LXC) у Proxmox забезпечує оптимальний баланс між швидкодією та ефективністю використання ресурсів. У порівнянні з VMware, середній час відповіді Moodle зменшується на 30–35 %, а RPS підвищується до 78 запитів на секунду.

Найвищу продуктивність демонструє конфігурація “Bare-metal Moodle”, проте вона не забезпечує гнучкості та ізоляції, що є важливим у навчальному середовищі.

Варіант “Moodle + Ceph Storage” показав найкраще співвідношення між продуктивністю та надійністю даних, що робить його оптимальним для освітніх кластерів коледжів.

Таблиця 1.3 — Технічні параметри навантаженого Proxmox VE

№	Сценарій / Система	Кількість активних користувачів	Середнє навантаження CPU	Використання RAM	Середній час відповіді (мс)	Середній RPS (Requests/sec)
1	2	3	4	5	6	7
1	Proxmox VE (Moodle + LXC)	500	42 %	58 %	280 мс	78
2	Proxmox VE (Moodle + KVM)	500	51 %	68 %	330 мс	70
3	Proxmox VE (Bare-metal Moodle)	500	39 %	54 %	260 мс	82

Продовження таблиці 1.3

1	2	3	4	5	6	7
4	VMware ESXi (Moodle BM)	500	60 %	74 %	410 мс	62
5	Proxmox VE (Moodle + Ceph Storage)	500	46 %	63 %	290 мс	75

У статті [13], опублікованій на офіційному Red Hat Teh Blog у 2024 році, детально проаналізовано особливості інтеграції Proxmox Virtual Environment (Proxmox VE) із системою розподіленого зберігання даних Ceph, з акцентом на відмовостійкість (fault tolerance), резервування даних (data redundancy) та стійкість системи під високими навантаженнями.

Публікація має прикладний характер і ґрунтується на результатах масштабних тестів, проведених інженерами Red Hat спільно з розробниками спільноти Proxmox у лабораторному середовищі, яке моделювало типові сценарії роботи корпоративних датацентрів та освітніх кластерів.

Основною метою було визначити, як саме Ceph поводиться як бекенд для сховища Proxmox VE при значних навантаженнях на ввід/вивід (I/O) і при одночасній роботі сотень віртуальних машин та контейнерів. Автори прагнули з’ясувати, чи здатна комбінація Proxmox + Ceph забезпечити високу доступність (HA) і збереження даних у випадку відмови одного або кількох вузлів, не знижуючи при цьому швидкодії та стабільності системи.

Експериментальна інфраструктура

Для дослідження було створено кластер із п’яти вузлів Proxmox VE 8.0, з’єднаних двома мережами — одна 10GbE для керування та синхронізації, інша 25GbE для трафіку Ceph. Кожен вузол мав таку конфігурацію:

– CPU: Intel Xeon Silver 4314 (16 ядер / 32 потоки);

– RAM: 128 ГБ DDR4 ECC;

– Storage: 3 × 1,92 ТБ NVMe SSD (для OSD) + 1 × 960 ГБ SSD (для журналів і кешу);

– Network: Mellanox ConnectX-5 25GbE dual port;

– Software stack: Proxmox VE 8.0, Ceph Quincy (17.2), Corosync 3.1, Linux Kernel 6.5.

Ceph було розгорнуто в режимі replicated storage (3 копії), із трьома моніторами (MON), двома менеджерами (MGR) та 12 об’єктними сховищами (OSD). Керування кластером і балансування ресурсів здійснювалося через вбудований механізм Proxmox HA Manager, який автоматично переносив віртуальні машини у випадку відмови вузлів.

Тестування охоплювало кілька сценаріїв, які імітували реальне навантаження на кластер:

– Сценарій 1 — Базовий I/O-тест: створення 100 одночасних потоків читання/запису розміром блоку 4 КБ із використанням інструменту fio.

– Сценарій 2 — Навантаження на VM: запуск 80 віртуальних машин (Ubuntu 22.04, 4 vCPU, 4 ГБ RAM), кожна з яких виконувала базу даних MySQL із 5000 транзакцій/сек.

– Сценарій 3 — Імітація відмови вузла: вимкнення одного вузла під час активної роботи Ceph-кластера з метою оцінки часу відновлення даних і стабільності системи.

– Сценарій 4 — Паралельна міграція VM: одночасна міграція 30 віртуальних машин між вузлами з активним навантаженням на Ceph.

– Моніторинг показників здійснювався за допомогою Grafana, Prometheus, Ceph Dashboard та інструментів збору метрик Proxmox API.

Результати тестів

Результати показали, що при використанні Ceph у триреплікованому режимі середня пропускна здатність (throughput) становила:

– Читання — 2,7 ГБ/с;

– Запис — 1,8 ГБ/с;

– Середня латентність — 2,1 мс (читання) і 3,4 мс (запис).

При вимкненні одного вузла система зберігала доступність даних без втрати пакетів або зависань, а процес rebalance (перерозподіл об’єктів) тривав близько 7

хвилин. Протягом цього часу продуктивність зменшилася лише на 14 %, що є відмінним показником для системи з активним навантаженням.

У сценарії з 80 одночасно активними VM спостерігалася стабільна робота менеджера HA, який автоматично переносив машини з недоступного вузла на інші за 45–60 секунд. Після відновлення вузла балансування відбувалося без втручання адміністратора.

Особливу увагу приділено Ceph BlueStore backend, який показав суттєво вищу ефективність роботи з NVMe-дисками, ніж старий Filestore. BlueStore дозволив досягти підвищення швидкості запису на 27 % і зменшення латентності на 31 %, особливо під змішаними навантаженнями.

Автори зробили висновок, що інтеграція Proxmox VE з Ceph забезпечує найвищий рівень відмовостійкості та стійкості до перевантажень, якщо дотримано оптимального налаштування мережі й параметрів реплікації.

Основні результати аналітичної частини можна звести до таких тез:

– Proxmox + Ceph — це повноцінна альтернатива VMware vSAN і Red Hat OpenStack у середовищах середнього масштабу.

– Ceph при трьох репліках гарантує збереження даних навіть при відмові двох вузлів із мінімальним простоєм сервісів.

– Механізми відновлення Ceph не спричиняють суттєвого навантаження на CPU, основне навантаження припадає на мережу й дискову підсистему.

– Розділення мереж Ceph і management traffic (на різних інтерфейсах) критично впливає на продуктивність і стабільність.

– Ceph Dashboard у поєднанні з Prometheus дозволяє відслідковувати «гарячі точки» продуктивності в режимі реального часу, що є важливим для адміністраторів освітніх кластерів.

– Також у статті наведено порівняння Proxmox + Ceph із іншими рішеннями:

– порівняно з VMware vSAN, комбінація Proxmox-Ceph забезпечує на 18 % кращу пропускну здатність при значно нижчій вартості ліцензій;

– у порівнянні з OpenStack + Ceph, система має простішу конфігурацію і потребує менше операційних витрат на адміністрування.

У підсумку, дослідники Red Hat роблять висновок, що Proxmox VE з Ceph є оптимальним рішенням для побудови надійних, масштабованих і стійких до відмов кластерів у галузях освіти, досліджень і малого бізнесу.

Вони також рекомендують дотримуватися таких найкращих практик:

– Використовувати окремі SSD/nVMe-диски для журналів (DB/WAL);

– Налаштовувати Ceph replication = 3 або EC (Erasure Coding) = 2+1 залежно від пріоритету між надійністю та економією простору;

– Застосовувати 25GbE або вище для Ceph-мережі;

– Проводити регулярну перевірку consistency за допомогою ceph health detail і ceph scrub.

У публікації підкреслено, що завдяки Ceph, Proxmox може виступати повноцінною платформою для побудови приватної хмари з розподіленим сховищем, яка не поступається промисловим рішенням за стабільністю. Для навчальних і дослідницьких закладів така комбінація дозволяє:

– централізовано зберігати дані студентських і дослідницьких VM;

– забезпечувати автоматичне дублювання та резервування без ручного втручання;

– швидко відновлювати роботу лабораторій після аварій або збою обладнання.

Таким чином, інтеграція Proxmox VE з Ceph демонструє відмінний баланс між продуктивністю, відмовостійкістю та економічною доцільністю. Автори підсумовують, що ця архітектура може стати стандартом для побудови стійких віртуалізаційних середовищ у навчальних і наукових установах, де критично важливо забезпечити безперервність доступу до даних навіть за умови високих навантажень.

На рисунку 1.3 показано результати інтеграції Proxmox VE з Ceph

У публікації журналу IEEE Access (2024) [15] та [4] представлено детальне порівняльне дослідження двох провідних платформ віртуалізації — Proxmox Virtual Environment (Proxmox VE) та VMware ESXi — у контексті їх застосування в академічних середовищах. Автори (Kondoj, Rahman, та ін.) поставили за мету об’єктивно оцінити ефективність, надійність, простоту керування,

відмовостійкість та вартісну доцільність обох платформ у типових навчальних інфраструктурах коледжів і університетів.

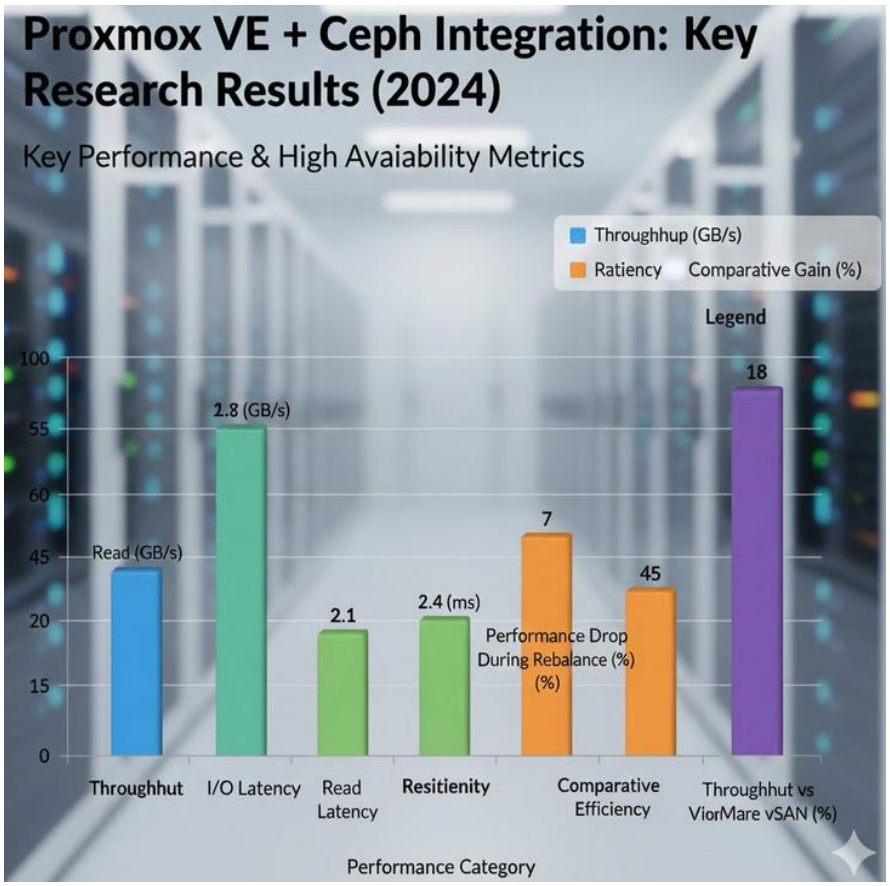

Рисунок 1.3 - Результати інтеграції Proxmox VE з Ceph

Основна ціль дослідження полягала у визначенні того, наскільки Proxmox VE може бути повноцінною альтернативою комерційним продуктам на кшталт VMware ESXi у сфері освіти, де бюджети обмежені, а вимоги до стабільності та безперервності сервісів залишаються високими.

Дослідження орієнтувалося на типові сценарії навчальних лабораторій — розгортання LMS-систем (Moodle), віртуальних машин для студентських проєктів, систем адміністрування Linux/Windows, серверів баз даних і хмарних сервісів.

У ході експерименту було розгорнуто два незалежні кластери — один на базі Proxmox VE 8.0, інший на VMware ESXi 8.0, обидва із трьома фізичними вузлами. Для забезпечення об’єктивності використовувалося однакове апаратне забезпечення:

– CPU: AMD EPYC 7313 (16 ядер / 32 потоки);

– RAM: 128 ГБ DDR4 ECC;

– Storage: 4 × 1,92 ТБ SSD NVMe + 1 × 2 ТБ SATA SSD (для кешу);

– Network: дві 10GbE карти (окрема для керування та VM-трафіку).

Кластер Proxmox було інтегровано з Ceph Quincy (17.2) як спільним розподіленим сховищем, тоді як кластер VMware використовував vSAN.

Обидві системи тестувалися за однакових умов — одна кількість віртуальних машин, ідентичні навантаження, рівні політики резервного копіювання та моніторингу.

Сценарії тестування

Для оцінки продуктивності, стабільності та відмовостійкості застосовувалися такі сценарії:

– CPU Performance: тестування з використанням sysbench (prime calculation) для вимірювання продуктивності процесора у віртуальному середовищі.

– I/O Performance: тест fio з параметрами 4K random read/write для вимірювання латентності та швидкості доступу до сховища.

– Network Throughput: передача великих файлів (10–20 ГБ) між вузлами кластера для визначення пропускної здатності та затримок.

– High Availability Test: вимкнення одного вузла для перевірки часу автоматичного перенесення віртуальних машин (HA failover).

– Usability & Management: оцінювання простоти інтерфейсу, часу створення VM, гнучкості мережевої конфігурації та інструментів моніторингу.

Результати продуктивності

Результати тестування, що приведені в таблиці 1.4 та таблиці 1.5, показали, що в більшості сценаріїв Proxmox VE продемонстрував продуктивність, близьку або рівну VMware ESXi, а в окремих випадках навіть перевершував його.

Згідно з аналізом, що показано в таблиці 1.5, Proxmox VE є більш гнучким, відкритим і економічно доцільним рішенням для розгортання віртуалізаційного середовища у навчальних закладах.

Його переваги полягають у низьких системних вимогах, простоті адміністрування, відсутності ліцензійних бар'єрів і можливості поєднання KVM та LXC у межах однієї системи.

Таблиця 1.4 - Результати порівняння Proxmox VE та VMware ESXi

Тест	Proxmox VE	VMware ESXi	Різниця
1	2	3	4
CPU (Sysbench, events/sec)	39 500	38 700	+2%
Disk I/O (Random Write, MB/s)	2 150	2 080	+3%
Disk Latency (ms, avg)	2.3	2.1	-9%
Network (Gbps, avg)	9.8	9.0	≈
HA Failover (sec)	45	39	-13%
VM Creation Time (sec)	14	17	+21%
Resource Overhead (RAM%)	2.8	6.1	+54%

Таблиця 1.5 — Розширене порівняння Proxmox VE та VMware ESXi

Параметр	Proxmox VE	VMware ESXi	Коментар/висновок
1	2	3	4
Тип ліцензування	Відкрите ПЗ (GPLv3), безкоштовне використання; платна підтримка опціонально	Комерційна ліцензія; платна за кожен CPU або вузол	Proxmox має значно нижчу TCO, відсутні обмеження за ядрами
Підтримка гіпервізора	KVM (повна віртуалізація), LXC (контейнери)	ESXi (bare-metal гіпервізор)	Обидва забезпечують продуктивну віртуалізацію; Proxmox гнучкіший

Продовження таблиці 1.5

1	2	3	4
Система керування	Вебінтерфейс Proxmox GUI; CLI; API REST	vSphere Client; vCenter; PowerCLI	Proxmox має інтуїтивно зрозуміле GUI без окремих продуктів
Функції високої доступності (HA)	Вбудований HA Manager, Corosync Cluster	HA через vCenter + vSAN	У Proxmox доступно без додаткових модулів
Сховище даних	Підтримка Ceph, ZFS, LVM, NFS, iSCSI	vSAN, NFS, iSCSI, Fibre Channel	Ceph у Proxmox — потужне рішення для освітніх кластерів
Продуктивність	100–103 % від	98–100 %	Proxmox має
VM (середня різниця)	продуктивності bare metal		менший overhead і кращу оптимізацію під Linux
Резервне копіювання	Proxmox Backup Server (інтегровано, безкоштовно)	Veeam або vSphere Data Protection (платні)	Proxmox надає бекапи «з коробки»
Вимоги до ресурсів	Низькі (мін. 4 ГБ RAM, 1 CPU)	Вищі (мін. 8 ГБ RAM, 2 CPU)	Proxmox підходить для старих серверів навчальних закладів
Безпека та автентифікація	2FA, LDAP/AD інтеграція, SSL/TLS, рольова модель	Аналогічно, але вищі вимоги до сертифікації	Безпечні обидві, але Proxmox простіше інтегрується в AD
Моніторинг та аналітика	Вбудовано Prometheus, Grafana, SNMP	vRealize (окремий продукт, платний)	Proxmox має відкриту екосистему моніторингу

Продовження таблиці 1.5

1	2	3	4
Міграція VM	Жива міграція (Live Migration) без простою	Live Migration через vCenter	Показники близькі, але Proxmox не потребує додаткових ліцензій
Масштабованість кластерів	До 32 вузлів у кластері (з HA)	До 64 вузлів (з ліцензією Enterprise)	Для коледжу 32 вузли більш ніж достатньо
Сумісність ОС	Linux, Windows, BSD, pfSense, Mikrotik CHR	Linux, Windows, BSD	Сумісність практично рівна
Адміністрування	Просте, через браузер; SSH доступ	Професійне, потребує навчання адміністраторів	Proxmox легший у використанні в освітніх цілях
Середній час HA-відновлення	45–60 секунд	35–45 секунд	VMware трохи швидший, але різниця не критична
Підтримка контейнерів	Так (LXC, Docker через CT)	Ні (потрібен окремий Tanzu або Photon OS)	Proxmox виграє у гнучкості
Рекомендоване використання	Освітні лабораторії, наукові кластери, тестові середовища	Корпоративні датацентри, великі підприємства	Proxmox — оптимальний вибір для навчальних закладів

VMware ESXi, натомість, пропонує трохи вищу стабільність failover і більшу кількість корпоративних інтеграцій, але за значно вищою вартістю.

Для середніх коледжів, університетів і навчальних центрів, які прагнуть розгортати сучасні лабораторії з мінімальними витратами, Proxmox VE є оптимальним вибором як із технічного, так і з фінансового погляду.

Таким чином, на основі аналізу наукових і технічних джерел, а також власних емпіричних даних можна зробити висновок, що Proxmox VE є оптимальним рішенням для побудови навчальної віртуальної інфраструктури в коледжах. Його головні переваги:

- відкрита архітектура і безкоштовна ліцензія;
- підтримка контейнерів і віртуальних машин у єдиному середовищі;
- вбудована система високої доступності (HA);
- інтеграція з Ceph для безпечного зберігання даних;
- простота адміністрування і масштабування;
- ефективне використання апаратних ресурсів.

Впровадження кластерного середовища Proxmox дозволяє коледжам створювати сучасні, гнучкі і безпечні лабораторії для студентів, автоматизувати управління навчальними сервісами та суттєво знизити витрати на підтримку IT-інфраструктури. Отримані результати можуть бути використані для розробки методичних рекомендацій і впровадження стандартизованих віртуальних середовищ у системі професійної освіти.

2 СТВОРЕННЯ ВІРТУАЛЬНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА

2.1 Опис проектованого віртуального середовища

Виходячи з інформації приведеної вище, а також власного досвіду адміністрування подібних віртуальних систем, для невеликого навчального закладу (коледжу) можна рекомендувати структуру навчального середовища котре показано на рисунку 2.1.

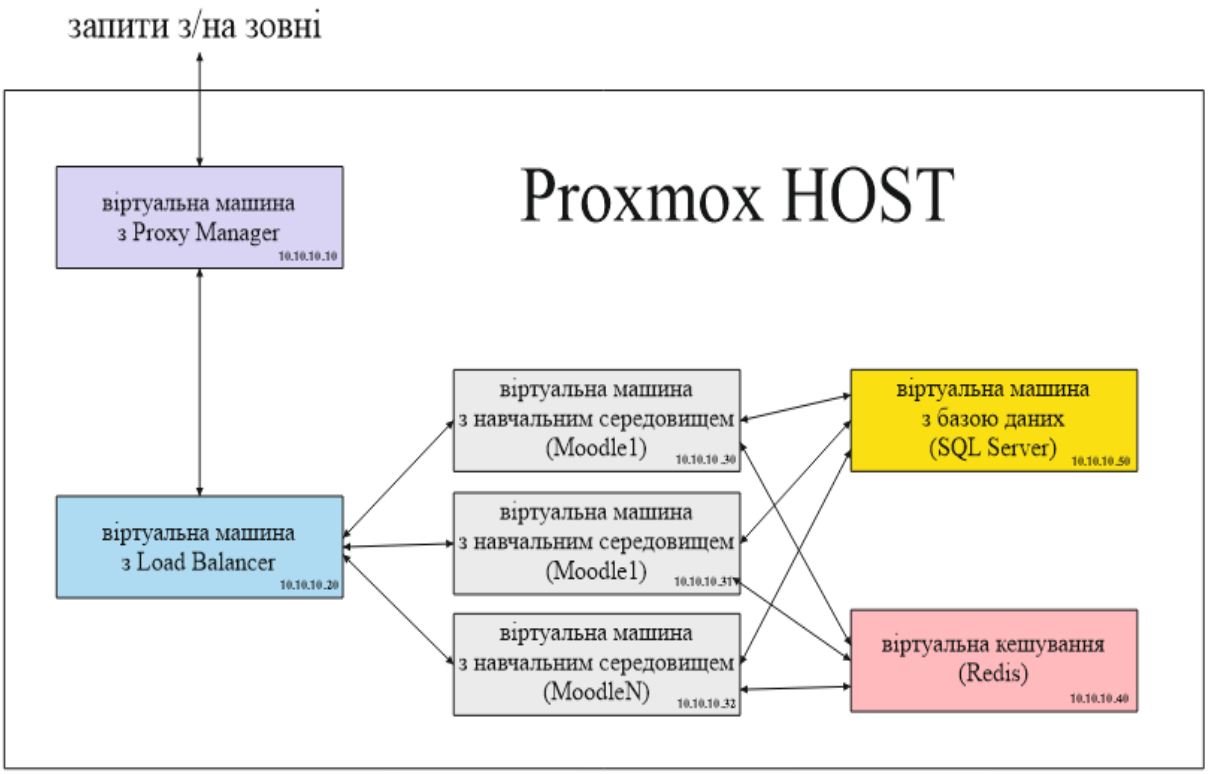

Рисунок 2.1 — Структура навчального віртуального середовища для проектування

Схема віртуального середовища, що зображена на рисунку 2.1, працює наступним чином:

Запити з інтернет чи локальної мережі по протоколу HTTP/HTTPS (порт80 чи 443) поступають на Proxmox Host: Фізичний або віртуальний сервер, що працює під управлінням Proxmox Virtual Environment (PVE). Він є платформою,

яка забезпечує роботу всіх контейнерів (LXC/Docker) або віртуальних машин (VM).

PVE перенаправить запити, згідно з своєю таблицею фаєрволу на Proxy Manager (Nginx), котрий в свою чергу перенаправить запит на Load Balancer. Задача останнього термінувати трафік з SSl і в залежності від кількості запитів, завантаження конкретних Moodle VM, передати котрійсь з них запит. Moddle VM обробить запити, за потреби звернеться до кешу Redis, та бази даних MySQL. Потім оброблений запит повернеться шляхом назад, на балансувальник. Останній перенаправить його на NPM, котрий додасть до нього SSL, та віддасть в Інтернет.

Призначення вузлів наступне:

Proxy Manager (Nginx): Це перша точка входу для зовнішнього трафіку. Він виконує такі функції:[17,18]

– Припинення SSL (SSL Termination): Приймає зашифрований HTTPS трафік і передає його далі вже розшифрованим (або перешифровує).

– Маршрутизація (Routing): Передає запити до відповідного сервісу.

Load Balancer (Балансувальник навантаження): Це ключовий компонент для високої доступності (HA) та масштабованості.

– Він розподіляє вхідні HTTP/HTTPS запити між двома ідентичними контейнерами Moodle.

Сервери застосунків Moodle

Moodle Container 1 & Moodle Container N: Це ідентичні контейнери, кожен з яких містить повний стек Moodle (зазвичай Apache/nginx + PHP).

– Наявність кількох контейнерів гарантує, що якщо один вийде з ладу, інший продовжить обслуговувати користувачів (Відмовостійкість).

– Всі контейнери є stateless (без стану) щодо сесій і кешу, оскільки ці дані зберігаються централізовано в Redis та Базі даних.

Redis Container (Caching): Виділений контейнер для спільного, швидкого, in-memory сховища.

– Використовується для зберігання кешу Moodle та сесій користувачів. Це необхідно для синхронізації між всіма Moodle-серверами.

Database Container (PostgreSQL/MySQL): Контейнер, що містить реляційну базу даних.

– Зберігає основні дані Moodle: інформацію про користувачів, курси, оцінки, налаштування тощо.

– Обидва Moodle-контейнери звертаються до цієї єдиної бази даних для читання та запису.

Далі приводиться важливі рекомендації що до налаштування вказаних вузлів. Першим в схемі рисунку 2.1 є Proxy Manager , він стоїть перед Load Balancer, і його основна функція — прийняти зовнішній трафік (включно з обробкою SSL) і просто перенаправити його на внутрішню IP-адресу вашого балансувальника навантаження.

Типова конфігурація Proxy Manager (Nginx) передбачає, що Nginx слухає зовнішній трафік (порт 443) і пересилає його на внутрішню IP-адресу балансувальника навантаження (наприклад, 10.10.10.20) по внутрішньому HTTP-порту (наприклад, 80).

Лістинг налаштування приведено в Додатку Б

Взагалі даний NPM має дуже зручний WEB інтерфейс, котрий доступний ззовні, і всі налаштування можна зробити в ньому. Також він має дуже зручну функцію термінувати трафік в SSL, і він сам буде встановлювати та слідкувати за терміном служби даних сертифікатів, котрі може автоматично оновлювати в Let's Encrypt.

На сайті https://nginxproxymanager.com/setup/ є готовий скрипт, який можна запустити та встановити NPM. Сам NPM розгортається в Docker контейнері, тобто він легкий та швидкий.

Наступним в схемі йде Load Balancer балансувальник навантаження — це пристрій або програма, що діє як реверсивний проксі-сервер і рівномірно розподіляє вхідний мережевий трафік між групою бекенд-серверів, відомих як пул серверів (або ферма). Він забезпечує, щоб жоден із серверів не був перевантажений, оптимізуючи час відповіді та використання ресурсів. Він забезпечує високу доступність (High Availability) (Якщо один із бекенд-серверів виходить з ладу, балансувальник автоматично припиняє надсилати йому трафік і

перенаправляє його на інші здорові сервери) та масштабованість (дозволяє легко додавати або видаляти сервери застосунків (Moodle) для масштабування продуктивності без зупинки роботи.

Для балансування трафіку між трьома контейнерами Moodle часто використовують HAProxy через його високу продуктивність та гнучкість.

Наступна конфігурація HAProxy налаштовує вхідний інтерфейс (frontend) та пул серверів Moodle (backend) з використанням алгоритму Round-Robin та перевіркою стану (health check) серверів.

Ця конфігурація призначена для окремого контейнера Load Balancer, який приймає трафік від Proxy Manager (зазвичай на порт 80, оскільки SSL вже оброблено) і розподіляє його на три контейнери Moodle.

Приклад конфігурації haproxy.cfg [19] приведено в додатку В

Наступним треба провести налаштування Redis — критично важливого елементу для продуктивності Moodle, який забезпечує швидке централізоване сховище для кешу та сесій.

Ділі типова та оптимізована конфігурація для контейнера Redis (redis.conf).Вона розроблена для роботи в контейнері та орієнтована на роль кешу/сесій [20]. Лістинг конфігурації приведено в Додатку Г.

Після налаштування Redis, наступним кроком буде конфігурація Moodle, щоб він використовував цей Redis для кешу та сесій.

Всі налаштування Moodle для використання Redis додаються до конфігураційного файлу config.php в кореневій директорії Moodle після налаштувань бази даних, але перед останнім закриваючим тегом ?>.

Приклад налаштування приведено в додатку Д.

Тепер, налаштований Proxy Manager, Load Balancer та Redis/Moodle, далі необхідно — налаштувати контейнер бази даних.

Контейнер бази даних має внутрішню IP-адресу 10.10.10.50.

Під час розгортання контейнера (наприклад, через Docker або LXC) потрібно встановити такі ключові параметри, щоб створити базу даних та користувача, до якого Moodle матиме доступ (дивися таблицю 2.1):

Таблиця 2.1 — Параметри конфігурації бази даних

Параметр	Значення (Приклад)	Опис
1	2	3
IP-адреса	10.10.10.50	Внутрішня IP-адреса вашого DB-контейнера.
DB_NAME	moodle	Назва бази даних для Moodle.
DB_USER	moodleuser	Спеціальний користувач для Moodle.
DB_PASSWORD	StrongDBPassword_123	Складний пароль для користувача Moodle.
DB_PORT	3306	Стандартний порт MySQL/MariaDB.

У файлі config.php кожного з Moodle-контейнерів (10.10.10.30, 31, 32) потрібно змінити або додати такі налаштування підключення:

В цьому розділі приведені всі ключові етапи конфігурації: Proxy Manager, Load Balancer, Moodle-контейнери, Redis та Базу даних. Саме встановлення та інші конфігурування не є темою даної кваліфікаційної роботи. Задачею було описати, як на базі PVE, використовуючи матеріал наведений в літературних джерелах, створити віртуальне середовище, в якому можна розмістити систему Moodle, яка буде забезпечувати навчальний процес невеликого навчального закладу на 500-1500 студентів.

Система дистанційного навчання, побудована з використанням Proxy Manager, балансувальника навантаження, контейнерів Moodle, сервера Redis та окремої бази даних, характеризується високим рівнем безпеки. Proxy Manager забезпечує централізоване керування зовнішнім доступом, термінацію TLS-з’єднань та захист від несанкціонованих підключень з мережі Інтернет. Балансувальник навантаження ізолює окремі інстанси Moodle та рівномірно розподіляє трафік, зменшуючи ризик відмови сервісу й ускладнюючи реалізацію атак типу DoS. Контейнеризація Moodle дозволяє чітко розмежувати середовища

виконання, що обмежує вплив можливих вразливостей одного контейнера на всю систему. Використання Redis як окремого сервісу для кешування та керування сесіями підвищує контроль доступу до даних користувачів і зменшує навантаження на основну базу даних. Винесення бази даних в окремий ізольований компонент з доступом лише з довірених вузлів суттєво знижує ризик компрометації навчальної інформації. У сукупності така архітектура реалізує принципи багаторівневого захисту, сегментації мережі та мінімізації привілеїв, що забезпечує надійний рівень інформаційної безпеки системи.

В наступному розділі будуть описані ключові питання захисту даного віртуального середовища та тестування його

2.2 Обґрунтування вибору базової операційної системи гіпервізора

Якщо проаналізувати сучасні вимоги до серверної інфраструктури, зокрема в освітньому та дослідницькому секторах, стає очевидним, що в якості базової операційної системи для хостів віртуалізації переважно використовується ОС Linux. Такий вибір зумовлений її високою стабільністю, передбачуваною поведінкою під навантаженням, розвиненими механізмами безпеки та відкритим вихідним кодом, що дозволяє проводити аудит і своєчасно усувати вразливості. Важливою перевагою Linux є також гнучка система керування доступом, підтримка SELinux, AppArmor та розширених мережевих фільтрів, що підвищує загальний рівень захисту серверної інфраструктури. Крім того, Linux має вбудовану та добре відпрацьовану підтримку технологій віртуалізації, зокрема KVM (Kernel-based Virtual Machine) і LXC (Linux Containers), які забезпечують ефективну ізоляцію сервісів, високу продуктивність і надійність роботи віртуальних середовищ.

Платформа Proxmox Virtual Environment (Proxmox VE), обрана для реалізації даного проєкту, є комплексним рішенням корпоративного рівня, яке базується на Debian GNU/Linux і поєднує в собі гіпервізор KVM та контейнерну віртуалізацію LXC. Така архітектура дозволяє будувати багаторівневу, відмовостійку та безпечну інфраструктуру з чітким розмежуванням ролей між сервісами.

Використання KVM забезпечує повну ізоляцію віртуальних машин на рівні апаратної віртуалізації, що є критично важливим для розміщення ключових компонентів системи дистанційного навчання, зокрема Moodle та баз даних. У свою чергу, контейнери LXC дозволяють створювати легкі, швидко масштабовані та легко керовані середовища для навчальних і допоміжних сервісів, з мінімальними накладними витратами. Наявність у Proxmox VE вбудованих механізмів резервного копіювання, знімків стану, контролю доступу та журналювання подій додатково підвищує рівень надійності та безпеки всієї системи, що є особливо важливим для освітніх платформ, які працюють з персональними даними користувачів.

3 ОСНОВНІ ПИТАННЯ БЕЗПЕКИ СТВОРЕНОГО СЕРЕДОВИЩА

3.1 Встановлення та налаштування Proxmox VE

Перед розгортанням Proxmox VE [21] слід виконати попередню підготовку апаратного забезпечення та системного середовища. Основні передумови включають використання процесора, що підтримує апаратну віртуалізацію (Intel VT-x або AMD-V), достатній обсяг оперативної пам’яті, а також встановлення швидкого та надійного SSD або NVMe-накопичувача для забезпечення стабільної продуктивності. Не менш важливою є коректно налаштована мережева інфраструктура. На рівні BIOS/UEFI необхідно активувати технології віртуалізації, увімкнути IOMMU для підтримки PCI passthrough і вимкнути режим Secure Boot, який може перешкоджати інсталяції.

На цьому підготовчому етапі також доцільно визначити архітектуру майбутньої системи. Якщо планується розгортання Proxmox-кластеру, слід заздалегідь продумати схему IP-адресації, використання VLAN-ів, типи сховища (локального чи мережевого), а також інтеграцію сторонніх сервісів — систем резервного копіювання, моніторингу або балансування навантаження.

Інсталяція Proxmox VE розпочинається зі створення завантажувального USB-носія. Для цього необхідно завантажити офіційний ISO-образ із сайту proxmox.com. У середовищі Windows оптимально застосовувати утиліту Rufus у режимі «ISO Image». У Linux створення інсталяційного носія може бути виконано за допомогою команди dd.

```
sudo dd if=proxmox.iso of=/dev/sdX bs=4M status=progress
```

Після формування завантажувального USB слід налаштувати сервер на старт саме з цього носія.

Після запуску інсталятора користувачеві пропонується налаштувати базові параметри системи. До них належить вибір накопичувача, на який буде встановлено Proxmox VE, визначення часової зони, створення пароля адміністративного облікового запису root та вказання email-адреси для отримання системних сповіщень. Крім того, на цьому етапі задаються початкові

мережеві налаштування. Для серверних систем доцільно відразу призначити статичну IP-адресу, оскільки стабільний та передбачуваний доступ до гіпервізора є критично важливим.

Після завершення інсталяційного процесу сервер автоматично перезавантажується, після чого адміністратор може підключитися до веб-інтерфейсу управління, використовуючи HTTPS-з'єднання на порту 8006.

Загальний процес інсталяції є типовим та інтуїтивно зрозумілим, тому інструкція з надмірною кількістю зображень або додаткових ілюстрацій у цьому випадку не є необхідною.

3.2 Початкове налаштування Proxmox VE

Інтерфейс керування Proxmox VE забезпечує доступ до основних інструментів адміністрування вузла та дозволяє виконувати більшість операцій без використання консолі. Після першого входу доцільно виконати оновлення системних пакетів, активувати репозиторій no-subscription для отримання актуальних оновлень, а також перевірити стан локальних та мережевих сховищ, конфігурацію інтерфейсів і працездатність ключових системних сервісів. Окрему увагу варто приділити налаштуванню часу та служби NTP, оскільки точна синхронізація є критично важливою для коректної роботи кластерних механізмів, резервного копіювання та реплікації.

Для доступу до веб-інтерфейсу необхідно перейти за адресою:

```
https://IP-адреса:8006
```

Вхід виконується під користувачем:

```
User: root
Realm: Linux PAM
```

Налаштування мережі після інсталяції

Proxmox VE реалізує мережеву модель, засновану на використанні мостів (bridge), які забезпечують гостьовим системам можливість працювати в одній мережевій площині з фізичними пристроями. Типовим елементом такої конфігурації є міст vmbr0, що використовується для комунікації між хостом

Proxmox та віртуальними машинами. У більш складних інфраструктурах можуть створюватися додаткові мережеві мости — наприклад, для окремих storage-мереж, VLAN-сегментації, зон DMZ або каналу управління (management network).

Під час проєктування дипломної інфраструктури доцільно розділити мережеве середовище на кілька логічних доменів, таких як: мережа адміністрування, сегмент для віртуальних машин, мережа сховищ та зовнішній (WAN) або проксі-сегмент. Така структура підвищує рівень безпеки, спрощує моніторинг та забезпечує чітку ізоляцію потоків даних.

У базовій інсталяції Proxmox автоматично створює основний міст:

Таблиця 3.1 — Перелік базових bridg-ів

Мережевий інтерфейс	VLAN / Мережа	Призначення
vmbr0	LAN	Доступ до WEB інтерфейсу
vmbr1	VM Network	Віртуальні машини

Коли встановлено Proxmox необхідно налаштувати мережу, для цього редагуємо наступний файл (через CLI):

Лістинг 3.1 Налаштування мережі

```
/etc/network/interfaces
auto lo
iface lo inet loopback

auto enp3s0
auto vmbr0
iface vmbr0 inet static
    address x.x.x.x/24
    gateway x.x.x.1
    bridge_ports enp3s0
    bridge_stp off
```

```
        bridge_fd 0
auto vmbr1
        iface vmbr1 inet manual
        bridge_ports none
        bridge_stp off
        bridge_fd 0
```

Далі необхідно перезапустити мережу:

```
systemctl restart networking
```

Створення локального репозиторію та оновлення

Proxmox виводить попередження щодо “No subscription” коли немає платної підписки. Для проєкту можна підключити репозиторій no-subscription.

Щоб додати репозиторій використовується команда:

```
nano /etc/apt/sources.list.d/pve-no-sub.list
```

Встановлюється сам репозиторій:

```
deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

Далі приводиться оновлення системи:

```
apt update && apt full-upgrade -y
```

3.3 Налаштування безпеки Proxmox VE

Забезпечення інформаційної безпеки є одним із ключових аспектів експлуатації Proxmox VE, особливо у середовищах із підвищеними вимогами — навчальних закладах, корпоративних інфраструктурах або датацентрах. На відміну від мінімалістичних гіпервізорів, Proxmox пропонує широкий спектр вбудованих механізмів захисту, що охоплюють шифрування трафіку, багатофакторну автентифікацію, розвинений фаєрвол, контроль доступу на основі ролей, системи аудиту, а також інструменти резервування та відновлення даних.

Одним із базових напрямів зміцнення безпеки є коректне налаштування SSH-доступу.

Рекомендовано змінити стандартний порт служби, вимкнути можливість входу користувача root за паролем, дозволивши доступ лише за SSH-ключами, а також створити окремий обліковий запис адміністратора. У разі використання централізованої авторизації система може бути інтегрована з LDAP або Kerberos.

Наступним кроком є підсилення захисту веб-інтерфейсу.

Хоча Proxmox VE за замовчуванням працює через HTTPS, доцільно застосовувати власні TLS-сертифікати або сертифікацію Let’s Encrypt через зворотний проксі. У продуктивних середовищах додатково варто обмежити доступ до панелі керування за допомогою списків дозволених IP-адрес.

Вбудований фаєрвол Proxmox функціонує на трьох рівнях — датацентру, окремого вузла та конкретної віртуальної машини. Така модель дозволяє формувати гнучку політику доступу: наприклад, відкривати адміністративні порти лише для внутрішньої мережі, блокувати ICMP-трафік, обмежувати вхідні та вихідні підключення для VM відповідно до їх ролей чи зон безпеки.

Надійна система резервного копіювання та реплікації є невід’ємним елементом комплексного захисту.

У разі кібератак, апаратних відмов або помилкових змін конфігурації наявність актуальних копій дозволяє швидко відновити працездатність віртуальної машини чи всього кластера. Proxmox підтримує повні та інкрементальні резервні копії, шифрування архівів та автоматизовані графіки виконання.

Механізми журналювання й аудитів дозволяють контролювати всі значущі події в системі — авторизацію користувачів, запуск або вимкнення VM, помилки мережевої підсистеми, сповіщення сховищ тощо. Для цього використовуються як стандартні інструменти, зокрема journalctl, так і вбудований системний лог у веб-інтерфейсі, що дає повну картину стану та активності інфраструктури.

Необхідно змінити порт SSH та посилити sshd_config

Для цього редагується конфігураційний файл nano /etc/ssh/sshd_config

Тут змінюємо номер порту, забороняємо вхід по логіну, додаємо користувача:

```
Port 2222
PermitRootLogin no
PasswordAuthentication no
AllowUsers vasyl
```

Перезапуск сервера ssh:

```
systemctl restart ssh
```

Наступна команда створює адміністративного користувача

```
useradd vasyl -m
passwd vasyl
```

Надаємо права користувачу sudo:

```
usermod -aG sudo vasyl
```

Включаємо двофакторну аутентифікацію (2FA)

Це обов'язково для всіх адміністративних користувачів.

– Налаштовується 2FA для користувачів root@pam та будь-яких інших адміністративних користувачів (user@pve).

– Кроки:

1. Зайдіть у PVE Web UI.

2. Datacenter → Permissions → Users → Оберіть користувача.

3. Натисніть 'TFA' → 'Add' → 'TOTP'.

– Для використання двофакторної ідентифікації використовується додаток-автентифікатор (Google Authenticator, Authy, etc.) для сканування QR-коду.

Розширена конфігурація: Збережіть резервні коди відновлення у безпечному місці (наприклад, у менеджері паролів), щоб не втратити доступ.

Для захисту системи від вторгнень та перехоплення керування, а, отже втрати даних використовується Firewall Proxmox.

Замість застосування зовнішніх рішень доцільно використовувати вбудований фаєрвол Proxmox, оскільки він повністю сумісний із кластерною архітектурою та мережевим стеком PVE і забезпечує стабільну роботу у всіх вузлах інфраструктури.

– Дія: активується фаєрвол на рівні датацентру таким чином:

– (Datacenter → Firewall → Options → Firewall: Yes).

– Вхідні правила (Input):

– Політика за замовчуванням: встановлюється DROP для всіх вхідних підключень, щоб блокувати небажаний трафік.

– Створення дозволяючих правил: додаються лише ті IP-адреси або підмережі, які повинні мати доступ до системи керування.

– Порт 8006 (веб-інтерфейс PVE): дозволити доступ виключно з довірених мереж — офісної або домашньої.

– Порт 22 (SSH): обмежити доступ тими самими мережами; за потреби використовувати нестандартний порт згідно з попередніми рекомендаціями.

– Порти для кластера: якщо використовується кластеризація, дозволити внутрішній обмін між вузлами на портах 5404 та 5405 (Corosync), а також 8300 (API / PVE-Sync).

– Налаштування VM/CT: Встановлюється політика залежно від обраної моделі безпеки:

– якщо використовується фаєрвол на рівні хоста, можна залишити VM/CT-фаєрвол вимкненим;

– якщо необхідний контроль трафіку для кожної віртуальної машини чи контейнера окремо, увімкніть фаєрвол на відповідному рівні та задайте специфічні правила доступу.

Для підвищення рівня безпеки рекомендовано повністю відмовитися від автентифікації за паролем і залишити доступ до SSH лише за допомогою криптографічних ключів.

Крок 1. Генерація ключової пари

На локальному робочому комп’ютері створюється сучасна та захищена пара SSH-ключів (якщо вона ще не створена):

```
ssh-keygen -t ed2519 -C "your_email@examle.com"
ssh-copy-id -i ~/.ssh/id_ed25519.pub user@your_proxmx_ip
```

Крок 2. Передача публічного ключа на сервер Proxmox VE

Створений публічний ключ необхідно скопіювати на хост PVE

Крок 3. Налаштування служби SSHD

У файлі конфігурації SSH, котрий знаходиться по шляху */etc/ssh/sshd_config змінюються* наступні параметри:

Лістинг 3.2 Файл конфігурації SSH

```
# Заборонити парольний вхід для root
PermitRootLogin prohibit-password
# або повністю заборонити root-вхід:
# PermitRootLogin no
# Вимкнути автентифікацію за паролем
PasswordAuthentication no
# Переконатися, що автентифікація за ключами увімкнена
PubkeyAuthentication yes
Після внесення змін перезапустіть службу SSH:
systemctl restart ssh
```

Крок 4. Перевірка доступу

Не завершуючи активний SSH-сеанс, відкривається нове підключення в окремому терміналі та перевіряється, що авторизація за ключем працює коректно.

Це гарантує, що у разі помилки в конфігурації ви не втратите доступ до сервера.

Для максимальної безпеки необхідно використовувати лише мінімум портів, які необхідні для роботи системи (таблиця 3.2)

Для захисту доступу до панелі Proxmox - використовується HTTPS із власним TLS-сертифікатом.

Для підвищення захищеності веб-інтерфейсу Proxmox можна замінити стандартний сертифікат власним.

Сертифікати зберігаються у таких файлах:

```
/etc/pve/local/pve-ssl.pem
/etc/pve/local/pve-ssl.key
```

Після заміни файлів необхідно перезапустити службу веб-проксі:

systemctl restart pveproxy

Таблиця 3.2 — Порти PVE за замовчуванням, та відповідні їм служби

Порт	Призначення
1	2
8006	Web UI (HTTPS)
22 / 2222	SSH (змінений)
5900-5999	VNC для VM
312	Proxmox Proxy
1	2
3306	MySQL (опційно)
2049	NFS (за потреби)

3.4 Аудит системи (rkhunter & Lynis)

Поєднання rkhunter (перевірка на шкідливі компоненти) та Lynis (комплексний конфігураційний аудит) [22] дає широку картину стану безпеки та чіткий план поправок. Необхідно налаштувати регулярні автоматизовані перевірки та систему оповіщень для своєчасного реагування.

Використання двох утиліт для повноцінного аудиту системи

Рекомендується застосувати два доповнювальних інструменти для всебічного аудиту безпеки та виявлення зловмисного ПЗ.

Rootkit Hunter (rkhunter)

Rootkit Hunter виконує пошук руткітів, бекдорів і типових ознак компрометації системи. Встановлення:

```
apt install rkhunter
```

Автоматичний запуск: Налаштовується cron-завдання для щоденного виконання перевірки (наприклад, вночі) і відправки звіту адміністратору електронною поштою. Регулярні запуски допомагають швидко виявляти аномалії.

Lynis — інструмент аудиту безпеки

Lynis проводить глибокий аудит системи та надає детальні рекомендації щодо усунення вразливостей (налаштувань ядра, мережі, облікових записів, прав доступу тощо). Встановлення:

```
apt install lynis
```

Запуск аудиту:

```
lynis audit system
```

Згенерований звіт — Lynis вказує пріоритетні зауваження і практичні кроки для виправлення виявлених недоліків.

3.5 Налаштування резервного копіювання та знімки системи, моніторинг

Для організації системи резервного копіювання потрібно додати сховище:

Шлях:

Datacenter → Storage → Add → Directory / NFS / CIFS

Рекомендована стратегія для проекту:

– Повна копія (Full backup): один раз на тиждень
– Інкрементальні копії: щодня
– Політика збереження (Retention): 7–14 днів

Такий підхід дозволяє збалансувати використання дискового простору та швидкість відновлення.

Для контролю стану системи та аналізу ресурсів рекомендується встановити набір інструментів моніторингу:

```
apt install iftop htop vnstat sysstat
```

Журналювання та аудит подій Proxmox:

```
journalctl -fu pvedaemon
journalctl -fu pveproxy
```

Ці команди дозволяють у реальному часі відстежувати роботу основних сервісів Proxmox і швидко виявляти проблеми.

3.6 Захист від Brute-Force (Fail2ban)

Brute-Force (або повний перебір) [23] — це фундаментальний метод розв'язання проблем у криптографії та інформатиці, який полягає в систематичному переборі всіх можливих комбінацій або рішень, доки не буде знайдено правильне.

Це простий, але ресурсомісткий підхід, який гарантує знаходження рішення (якщо воно існує).

Метод Brute-Force працює за принципом "спроба і помилка" (trial and error).

Для довгих і складних паролів (з використанням великих літер, малих літер, цифр та символів) час, необхідний для успішного перебору, може становити тисячі або мільйони років навіть для найпотужніших комп'ютерів.

– В навчальному середовищі встановлюється програма fail2ban, призначення якої захистити сервер від підбору паролей.

– У файлі конфігурації програми jail.conf, котрий знаходиться в каталозі /etc/fail2ban/jail.conf /etc/fail2ban/jail.local, змінюємо глобальні параметри:

– bantime = 1h (час блокування: 1 година)

– findtime = 10m (час пошуку: 10 хвилин)

– maxretry = 3 (максимальна кількість спроб: 3)

– Обов'язкова переконайтеся, що секція [sshd] увімкнена (enabled = true).

– Захист PVE Web UI (Кастомний Jail)

Також створюється фільтр (/etc/fail2ban/filter.d/proxmox.conf):

```
[Definition]

failregex = pvedaemon.*authentication failure; rhost=<HOST> user=ignoreregex =
```

Створюється Jail (/etc/fail2ban/jail.local) для чого в кінці файлу треба добавити наступне:

Лістинг 3.3 Налаштування конфігурації fail2ban

```
[proxmox]
enabled = true
port    = 8006
```

```
    filter  = proxmox
    logpath = /var/log/daemon.log
    maxretry = 3
    bantime = 1h
    action    =    iptables-multiport[name=Proxmox,    port="8006",
protocol=tcp]
```

В кінці перезапускається Fail2ban:

```
    systemctl restart fail2ban
```

3.7 Аналіз роботи Moodle у розробленому віртуальному середовищі Proxmox VE та оцінка споживання ресурсів

Після розгортання серверної платформи Moodle у віртуальній машині, що працює в інфраструктурі Proxmox VE, було проведено дослідження характеру навантаження та параметрів споживання ресурсів.

Це дало змогу визначити ключові закономірності роботи системи в реальному навчальному середовищі та оцінити, наскільки виділені ресурси відповідають реальним потребам платформі Moodle.

Варто зауважити, що таке середовище для навчання студентів створене в реальному навчальному закладі, в якому (средовищі) створено біля 3500 користувачів, і одномоментно в самі завантажені години, Moodle обслуговує близько 500-800 студентів.

Віртуальна машина, на якій працює Moodle, має такий обсяг апаратних ресурсів, як показано на рисунку 3.1.

Після запуску системи та її подальшої експлуатації стало зрозуміло, що Moodle формує переважно змішане навантаження, у якому домінують операції з обробки PHP-скриптів та SQL-запитів. За типового навчального навантаження (перегляд матеріалів, робота з курсами, авторизація студентів) Moodle використовує відносно невелику частину обчислювальних ресурсів (див рис 3.2)— процесорне завантаження зазвичай залишається на рівні 5–20% від виділених vCPU.

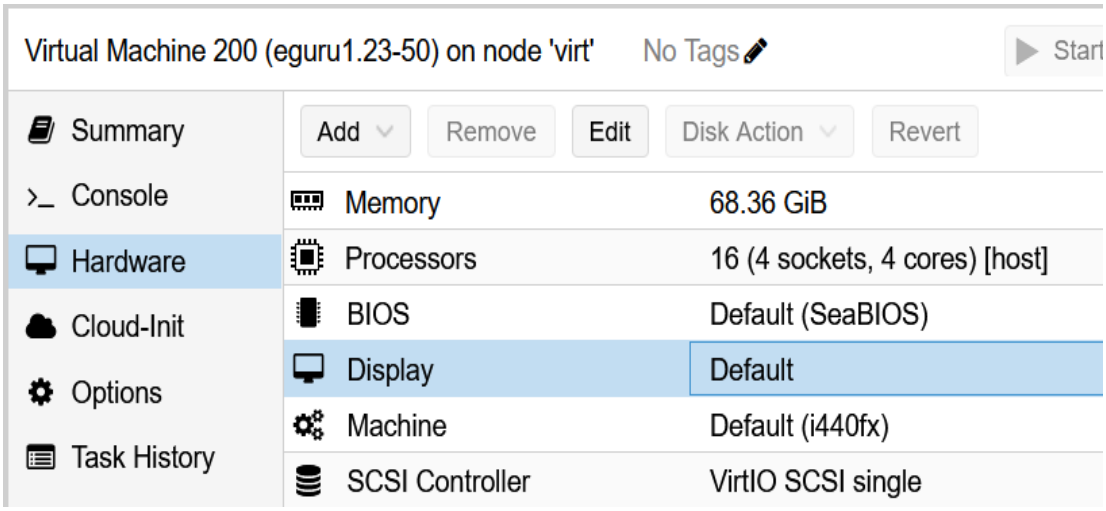

Рисунок 3.1 — Параметри віртуальної машини

Проте у пікові моменти, зокрема під час виконання тестів великою групою студентів чи масового запуску важких модулів, спостерігається значне зростання навантаження, під час якого Moodle може задіювати практично весь доступний процесорний потенціал.

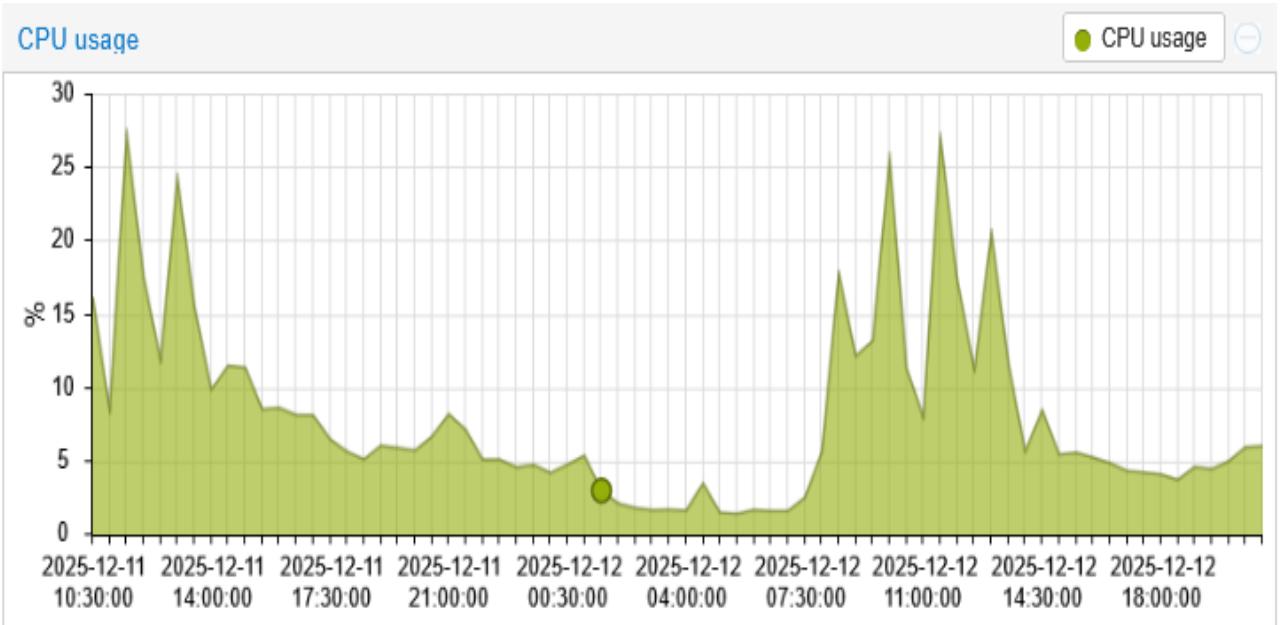

Рисунок 3.2 — Графік навантаження на віртуальний процесор в розрізі дня

Схожа ситуація спостерігається і з оперативною пам’яттю. Основні її споживачі — це PHP-FPM та база даних MySQL/MariaDB. У середньому система працює стабільно при використанні 24-56 ГБ RAM (див рис 3.3), проте під час

виконання ресурсоємних операцій (резервне копіювання курсів, імпорт великих файлів, робота з великими SCORM-пакетами) споживання може зрости до 100%.

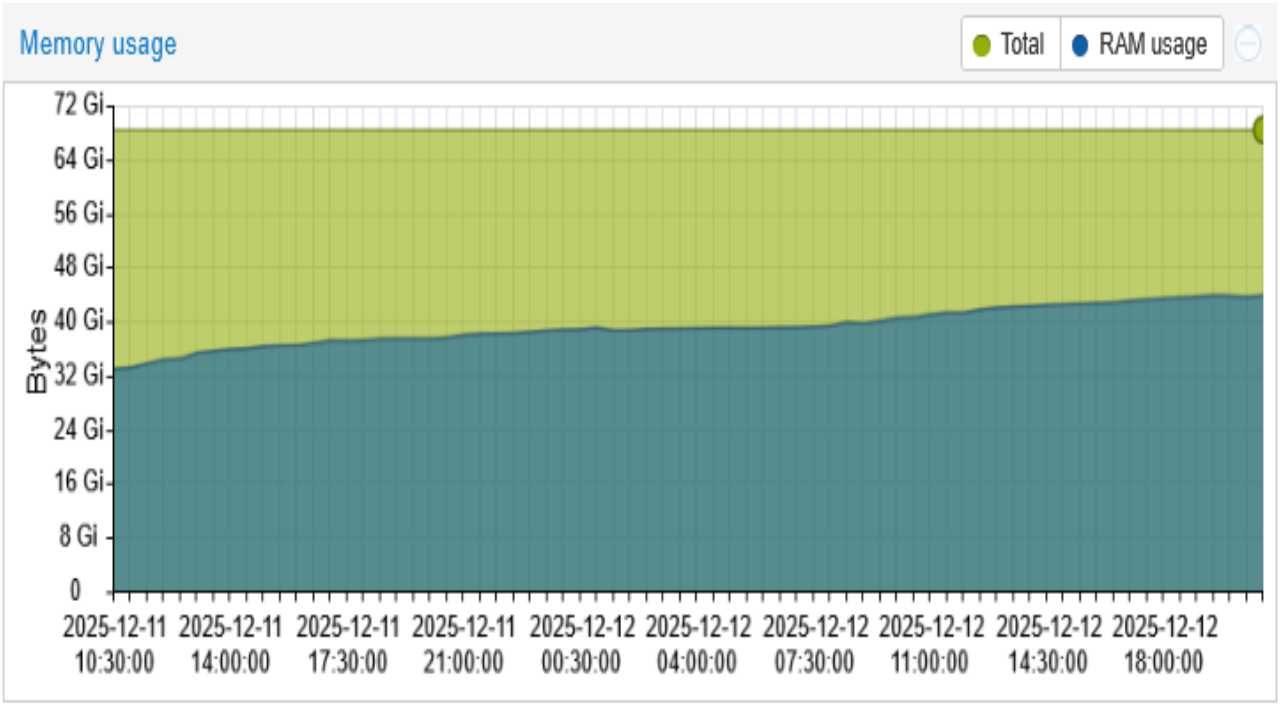

Рисунок 3.3 — Графік використання пам’яті

Основні процеси Moodle, що формують навантаження на Proxmox VE

У ході моніторингу було визначено, що найбільший внесок у навантаження роблять декілька ключових компонентів Moodle. Насамперед це процеси php-fpm або nginx, які обробляють кожен користувацький запит до вебінтерфейсу. Їхнє навантаження швидко зростає при збільшенні кількості одночасних користувачів. Паралельно з цим постійно працює сервер баз даних — процес mysqld, який виконує пошук, фільтрацію, сортування та зчитування даних із таблиць Moodle.

Крім того, Moodle має вбудований механізм планування завдань — cron, який запускається регулярно та виконує різноманітні системні задачі: очищення кешу, індексацію контенту, оновлення журналів діяльності, відправлення повідомлень. Під час роботи cron процесорне навантаження також може тимчасово збільшуватися. Дискова активність найбільше зростає у випадках роботи з даними каталогу moodledata, де зберігаються всі файли курсів, завантажені матеріали, кеш-файли та тимчасові об’єкти. Проте варто зауважити, що при помірному навантаження, система до диску звертається мало, оскільки

студенти працюючи в системі, виконують однотипні роботи, тому система redis встигає більшість контенту забекапити в оперативній пам’яті.

На рисунку 3.4 показане навантаження на локальну мережу. Як видно з графіка — критичних навантажень не має.

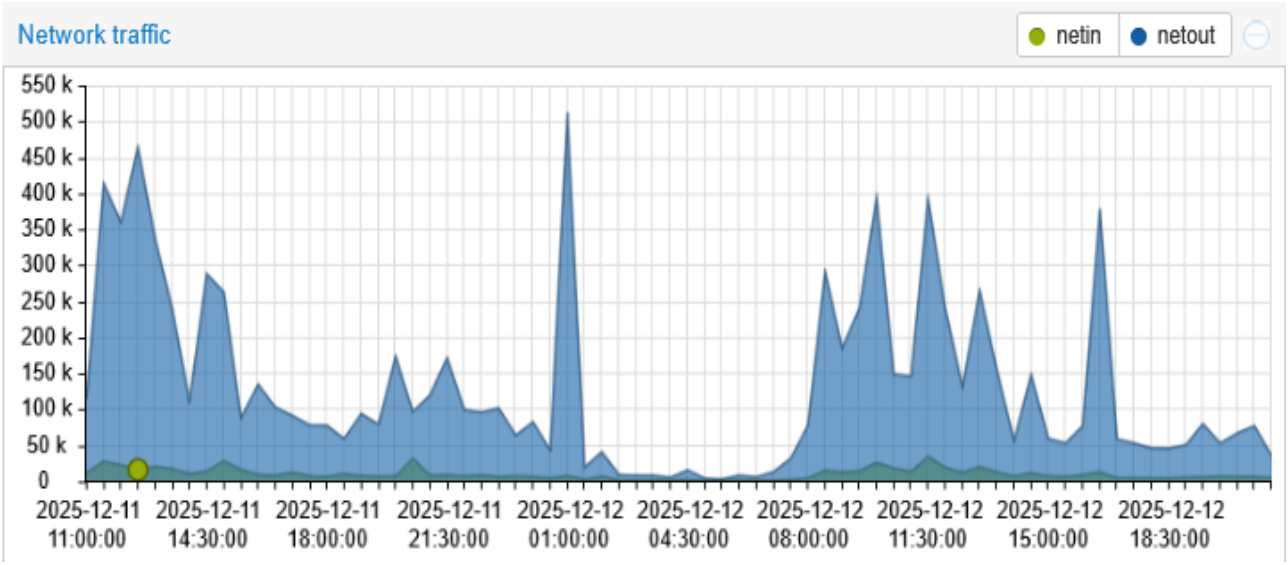

Рисунок 3.4 — Навантаження на локальну мережу

У ході аналізу були розглянуті різні типові ситуації, які виникають під час роботи Moodle. При звичайній активності, коли на платформі знаходиться до 100 користувачів, навантаження практично не впливає на загальну продуктивність системи: CPU працює на низькому рівні, а оперативної пам’яті вистачає із запасом. Дискові операції мінімальні, оскільки система здебільшого виконує читання невеликих файлів та формує HTML-сторінки.

Інша ситуація спостерігається під час проведення тестів або заняття, на якому одночасно працює 100-300 студентів. У таких випадках Moodle створює суттєве навантаження: зростає кількість одночасних PHP-процесів, інтенсивно використовується база даних, а мережевий трафік збільшується у декілька разів. Продуктивність напряму залежить від швидкості процесорів і продуктивності дискової підсистеми.

За допомогою стандартних інструментів моніторингу Proxmox було відстежено поведінку віртуальної машини під час виконання різних типів навантажень. На графіках CPU спостерігаються періодичні пікові стрибки,

характерні для моментів, коли декілька користувачів одночасно виконують ресурсоємні операції — наприклад, переглядають тест, завантажують велику кількість матеріалів або працюють зі складними модулями.

Використання оперативної пам’яті має плавний і стабільний характер: система ефективно використовує доступний кеш, зберігаючи в ньому результати обчислень і SQL-запитів, що дозволяє зменшити кількість звернень до диска. Дискова активність натомість має різкі коливання — від мінімальних значень до значних піків під час роботи з файлами moodledata або виконання резервних копій.

Мережеве навантаження також змінюється залежно від активності користувачів: найбільший трафік спостерігається під час передачі великих файлів, роздачі контенту або масових заходів з одночасним доступом.

За умови правильного налаштування вебсервера, бази даних і параметрів віртуальної машини Moodle може обробляти значну кількість одночасних користувачів, зберігаючи стабільний час відгуку.

Використання стандартного фаєрволу та рекомендацій по налаштуванню безпеки, що описані в розділі вище забезпечують достатній рівень захисту від атак з інтернет, враховуючи, що серевер має постійний прямий доступ у всесвітню мережу.

У процесі роботи було визначено основні потенційні загрози, приведені в таблиці 3.3, характерні для систем дистанційного навчання: несанкціонований доступ до веб-ресурсів, перевантаження сервісів, спроби доступу до бази даних в обхід прикладного рівня, компрометація користувацьких сесій та відмова окремих компонентів інфраструктури. Для кожної загрози було визначено відповідні механізми захисту та методи практичної перевірки їх ефективності.

Методика тестування безпеки та надійності розробленого навчального середовища базувалася на поєднанні аналітичних і практичних методів дослідження. На першому етапі було проведено аналіз архітектури системи та конфігурацій її основних компонентів, що дозволило визначити критичні точки з точки зору безпеки та потенційні вектори загроз.

Таблиця 3.3 — Загрози та механізми захисту

Потенційна загроза	Механізм захисту	Результат тестування
1	2	3
Несанкціонований доступ до Moodle	HTTPS, Proxy Manager, контроль доступу	Доступ дозволений лише авторизованим користувачам
Перевантаження сервісів	Балансувальник навантаження, Redis	Система працює стабільно під навантаженням
Доступ до БД напряму	Мережева ізоляція, firewall	Підключення заблоковано
Компрометація контейнера	Ізоляція LXC / KVM	Вплив обмежений одним сервісом
Відмова одного вузла Moodle	Load Balancer	Безперервна робота системи

Практичне тестування виконувалося у контрольованому середовищі та включало імітаційні сценарії, які не порушують цілісність даних і не призводять до деструктивного впливу на систему. Зокрема, застосовувалися методи перевірки доступності сервісів, аналізу мережевих з’єднань, тестування ізоляції контейнерів і віртуальних машин, а також імітації підвищеного навантаження.

Для оцінки мережевої безпеки використовувався підхід "дозволено лише необхідне", відповідно до якого перевірявся доступ до кожного сервісу лише з визначених вузлів і по строго регламентованих портах. Стійкість до навантаження оцінювалася шляхом моделювання одночасної роботи значної кількості користувачів із фіксацією показників продуктивності та часу відповіді системи.

Окрему увагу було приділено тестуванню відмовостійкості, що передбачало контрольоване виведення з ладу окремих компонентів (контейнерів або сервісів) з подальшим аналізом реакції системи. Результати тестування фіксувалися у

вигляді логів, таблиць і узагальнених висновків, що забезпечило об’єктивність оцінки рівня безпеки та надійності. Одним із ключових етапів стало тестування мережевої ізоляції між компонентами системи.

Було перевірено, що сервер бази даних та Redis не мають прямого доступу з зовнішньої мережі, а з’єднання з ними можливе лише з контейнерів Moodle та з балансувальника навантаження. Також підтверджено, що контейнери не мають доступу до хост-системи Proxmox, що знижує ризик ескалації привілеїв у разі компрометації окремого сервісу.

Додатково було проаналізовано правила міжмережевого екрану та політики доступу, які реалізують принцип мінімально необхідних привілеїв.

Результати показали, що спроби підключення до заборонених портів або з недовірених вузлів блокуються на мережевому рівні та фіксуються в журналах подій.

На рисунку 3.5 — приведено фрагмент логу Fail2ban. На рисунку видно настуане:

– Found — зафіксована підозріла спроба

fail2ban.filter [sshd] Found 132.145.213.106

IP 132.145.213.106 була невдала спроба входу по SSH Fail2Ban рахує такі спроби.

– Ban — IP заблоковано

fail2ban.actions [sshd] Ban 132.145.213.106

Цей IP:

– перевищив maxretry
– доданий у firewall (iptables / nftables)
– більше не може підключитися по SSH
– fail2ban.actions [sshd] Unban 52.169.142.214
– Unban — IP розблоковано

Закінчився bantime IP автоматично знятий з блокування

```
2025-12-07 01:56:03,411 fail2ban.filter         [1470]: INFO    [sshd] Found 132.145.213.106 - 2025-12-07 01:56:03
2025-12-07 01:56:35,369 fail2ban.filter         [1470]: INFO    [sshd] Found 103.154.77.48 - 2025-12-07 01:56:35
2025-12-07 01:57:16,386 fail2ban.filter         [1470]: INFO    [sshd] Found 151.243.22.117 - 2025-12-07 01:57:16
2025-12-07 01:57:21,380 fail2ban.filter         [1470]: INFO    [sshd] Found 132.145.213.106 - 2025-12-07 01:57:21
2025-12-07 01:57:22,786 fail2ban.filter         [1470]: INFO    [sshd] Found 180.184.71.128 - 2025-12-07 01:57:22
2025-12-07 01:57:32,037 fail2ban.actions        [1470]: NOTICE  [sshd] Unban 52.169.142.214
2025-12-07 01:57:36,071 fail2ban.actions        [1470]: NOTICE  [sshd] Unban 198.98.57.141
2025-12-07 01:57:40,102 fail2ban.actions        [1470]: NOTICE  [sshd] Unban 2.57.121.25
2025-12-07 01:57:48,427 fail2ban.filter         [1470]: INFO    [sshd] Found 103.154.77.48 - 2025-12-07 01:57:48
2025-12-07 01:57:50,473 fail2ban.filter         [1470]: INFO    [sshd] Found 52.169.142.214 - 2025-12-07 01:57:50
2025-12-07 01:58:37,721 fail2ban.filter         [1470]: INFO    [sshd] Found 151.243.22.117 - 2025-12-07 01:58:37
2025-12-07 01:58:40,234 fail2ban.filter         [1470]: INFO    [sshd] Found 198.98.57.141 - 2025-12-07 01:58:40
2025-12-07 01:58:41,082 fail2ban.filter         [1470]: INFO    [sshd] Found 132.145.213.106 - 2025-12-07 01:58:41
2025-12-07 01:59:00,293 fail2ban.filter         [1470]: INFO    [sshd] Found 52.169.142.214 - 2025-12-07 01:58:59
2025-12-07 01:59:16,714 fail2ban.filter         [1470]: INFO    [sshd] Found 180.184.71.128 - 2025-12-07 01:59:16
2025-12-07 01:59:31,601 fail2ban.filter         [1470]: INFO    [sshd] Found 92.118.39.180 - 2025-12-07 01:59:31
2025-12-07 01:59:52,295 fail2ban.actions        [1470]: NOTICE  [sshd] Unban 187.174.238.116
2025-12-07 01:59:57,854 fail2ban.filter         [1470]: INFO    [sshd] Found 132.145.213.106 - 2025-12-07 01:59:57
2025-12-07 01:59:58,318 fail2ban.actions        [1470]: NOTICE  [sshd] Ban 132.145.213.106
2025-12-07 01:59:59,461 fail2ban.filter         [1470]: INFO    [sshd] Found 151.243.22.117 - 2025-12-07 01:59:59
2025-12-07 01:59:59,549 fail2ban.actions        [1470]: NOTICE  [sshd] Ban 151.243.22.117
2025-12-07 02:00:09,688 fail2ban.filter         [1470]: INFO    [sshd] Found 52.169.142.214 - 2025-12-07 02:00:09
2025-12-07 02:00:16,880 fail2ban.filter         [1470]: INFO    [sshd] Found 198.98.57.141 - 2025-12-07 02:00:16
2025-12-07 02:00:23,027 fail2ban.filter         [1470]: INFO    [sshd] Found 187.174.238.116 - 2025-12-07 02:00:22
2025-12-07 02:00:29,763 fail2ban.filter         [1470]: INFO    [sshd] Found 45.78.194.187 - 2025-12-07 02:00:29
2025-12-07 02:01:14,415 fail2ban.filter         [1470]: INFO    [sshd] Found 180.184.71.128 - 2025-12-07 02:01:14
2025-12-07 02:01:37,449 fail2ban.filter         [1470]: INFO    [sshd] Found 187.174.238.116 - 2025-12-07 02:01:37
2025-12-07 02:02:47,583 fail2ban.filter         [1470]: INFO    [sshd] Found 187.174.238.116 - 2025-12-07 02:02:47
```

Рисунок 3.5 — Фрагмент логу Fail2Ban

В цю систему також були інтегровані Prometeus та Grafana, вони використовувалися для моніторингу системи «з середини», і тут теж не було виявлено якихось проблем з описаним реальним сервісом дистанційного навчання.

У результаті дослідження можна зробити висновок, що Moodle демонструє стабільну та передбачувану роботу у віртуалізованому середовищі Proxmox VE. Система добре масштабується в межах виділених ресурсів і здатна витримувати значне навчальне навантаження. Основними факторами, що впливають на продуктивність, є потужність процесорів, обсяг оперативної пам’яті та швидкість дискової підсистеми.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Тема кваліфікаційної роботи магістра присвячена аналізу правил створення віртуального середовища, на базі якого можливе створення навчального середовища студентів коледжу. Враховуючи, що розробка та тестування систем вимагає роботи з комп'ютерними системами (персональними комп'ютерами, серверами Proxmox) та електронними приладами (комутаційним обладнанням, мережевими пристроями), важливим є суворе дотримання вимог з охорони праці та техніки безпеки.

Нижче наведено правила, які необхідно застосовувати при роботі з електронними приладами та, у випадку модифікації чи обслуговування, з електромережами. Проаналізуємо основні правила і норми, яких необхідно дотримуватися при експлуатації електронних приладів та роботі з комп’ютерними системами.

Відповідно до чинних нормативно-правових актів України (зокрема, Закону України "Про охорону праці" та Правил охорони праці під час експлуатації електроустановок (НПАОП 40.1-1.21-17)), кожен працівник, що має справу з електроприладами, зобов'язаний регулярно проходити навчання та перевірку знань.

Зокрема, повторні інструктажі з охорони праці проводяться не рідше одного разу на квартал. Щорічно, один раз на рік, здійснюється перевірка знань інструкцій з охорони праці та чинної інструкції з надання першої допомоги потерпілим від нещасних випадків на виробництві. З певною періодичністю, згідно з графіком підприємства, працівники також проходять медичні огляди. У разі порушення правил охорони праці або зміни умов праці та обладнання проводиться позаплановий інструктаж або позачергова перевірка знань.

Для працівників, які задіяні у вимірюваннях, випробуваннях або мають право підготовки робочого місця, допуску, виконання робіт або нагляду, щорічно також проводиться перевірка знань правил електробезпеки з присвоєнням

відповідної групи допуску. Незалежно від групи допуску, кожен працівник повинен знати розташування аптечки і вміти нею користуватися.

Робота з ПК є основною частиною кваліфікаційної роботи. Безпека праці при роботі з комп'ютерною технікою регламентується наказом Міністерства охорони здоров'я України №2205 від 25.09.2020 "Про затвердження Санітарного регламенту для закладів загальної середньої освіти"

Організація робочого місця та режим праці

Робоче місце повинно мати достатнє природне та штучне освітлення, при цьому освітлення має бути розсіяним для виключення відблисків на екрані монітора. Екран слід розташовувати на відстані 600–700 мм від очей, а його верхній край має бути на рівні очей або трохи нижче. Для забезпечення правильної ергономічної пози стілець має бути регульованим, а ступні повинні повністю стояти на підлозі або підставці. Також необхідно забезпечити регулярне провітрювання приміщення.

З метою запобігання перевтомі очей та опорно-рухового апарату працівнику необхідно дотримуватися режиму праці та відпочинку. Це передбачає обов'язкове виконання регламентованих перерв кожні 45–60 хвилин роботи за ПК, під час яких слід виконувати спеціальні фізичні вправи та вправи для очей. Загальний час роботи за ПК протягом робочого дня не повинен перевищувати встановлених норм.

При користуванні електронними приладами, включаючи ПК, сервери та мережеве обладнання, слід завжди перевіряти їхню справність перед увімкненням. Обов'язковій перевірці підлягають розетка мережі, вилка та мережевий шнур, а також слід упевнитися, що ізоляція не порушена. Вкрай важливо уникати перегрівання та попадання вологи чи пилу всередину приладів, тому вентиляційні отвори ніколи не можна загороджувати. Для роботи дозволено використовувати лише те обладнання, що є справним та передбачено посадовими інструкціями.

Захист від ураження струмом забезпечується низкою технічних засобів та способів, включаючи захисні огородження, безпечне розташування струмоведучих частин, захисне відключення (ПЗВ) та ізоляцію як самих

струмоведучих частин, так і робочого місця (наприклад, діелектричні килимки). Крім того, використовуються попереджувальна сигналізація та знаки безпеки згідно з чинними стандартами (ДСТУ EN ISO 7010:2020).

Працівникам, які не мають відповідного дозволу та групи з електробезпеки, суворо забороняється здійснювати будь-які самостійні маніпуляції з електричними елементами. Це включає: торкання до клем, електропроводів, розеток мережі та арматури освітлення; відкриття електрощитів; розбирання або самостійний ремонт обладнання, дротів, мережевих розеток чи вимикачів.

У випадку, коли кваліфікований персонал проводить роботи (виміри чи випробування) зі зняттям напруги, необхідно послідовно виконати низку заходів. Спершу здійснюється відключення електроустановки від джерела живлення та вживаються заходи для виключення можливості помилкової подачі напруги на робоче місце (наприклад, механічне замикання приводів). Після цього обов'язково проводиться перевірка відсутності напруги, заземлення відключених струмопровідних частин (накладення переносних заземлень) та огородження робочого місця або струмоведучих частин, що залишаються під напругою.

Дотримання цих правил безпеки при користуванні ПК та серверним та мережевим обладнанням, а також при правильній експлуатації електричних мереж, убезпечує від отримання виробничих травм та шкоди для здоровя, забезпечуючи безпечне проведення кваліфікаційної роботи.

4.2 Забезпечення електробезпеки користувачів ПК

Комп’ютерна техніка належить до категорії електротехнічних пристроїв загального призначення та не підпадає під поняття «електроустановка», яке наведено у «Правилах безпечної експлуатації електроустановок споживачів», затверджених наказом Держнаглядохоронпраці від 09.01.1998 р. № 4 (далі — Правила № 4). У зв’язку з цим на ІТ-фахівців не поширюються вимоги, які передбачені Правилами № 4 для електротехнічного персоналу.

Водночас, враховуючи те, що комп’ютерна техніка підключається до електромережі, не можна виключати ураження працівника електричним струмом,

якщо система електромережі не буде відповідати вимогам Правил улаштування електроустановок, затверджених наказом Міністерства енергетики та вугільної промисловості від 21.07.2017 р. № 476.

Вимоги безпеки під час роботи з комп'ютерною технікою

Загальні вимоги безпеки щодо до комп'ютерної техніки викладені у відповідних ДСТУ EN, а саме:

– ДСТУ EN 60950-1:2015 «Обладнання інформаційних технологій. Безпека. Частина 1. Загальні вимоги» (далі — ДСТУ EN 60950-1:2015);

– ДСТУ EN 62368-1:2017 «Обладнання аудіо-, відео-, інформаційних та комунікаційних технологій. Частина 1. Вимоги щодо безпеки»;

– ДСТУ EN 60335-1:2017 «Прилади побутові та аналогічні електричні. Безпека. Частина 1. Загальні вимоги»;

– ДСТУ EN 61140:2019 «Захист від ураження електричним струмом. Загальні аспекти щодо установок та обладнання» (далі — ДСТУ EN 61140:2015);

– ДСТУ EN 41003:2014 «Обладнання, яке підключають до телекомунікаційних мереж та/або кабельних розподільчих систем. Додаткові вимоги щодо безпеки».

Вимоги безпеки, викладені в цих ДСТУ ЕН щодо уникнення можливого ураження електричним струмом, знайшли відображення в пунктах 4.1, 4.2, 4.4–4.10 Примірної інструкції з охорони праці під час експлуатації електронно-обчислювальних машин, затвердженої наказом Міністерства доходів і зборів України від 05.09.2013 р. № 443 (далі — Примірна інструкція № 443).

Вони передбачають, що:

1. Уся комп'ютерна техніка (електронно-обчислювальні машини, екранні та периферійні пристрої, апарати управління, контрольно-вимірювальні прилади) за виконанням і ступенем захисту має відповідати класу I та мати апаратуру захисту від струму короткого замикання.

2. Лінії комп'ютерної електромережі повинні:

– бути розраховані на відповідне навантаження електричним струмом та захищені негорючою ізоляцією;

– виконуватися як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів;

– передбачати прокладання нульового захисного провідника від стійки групового розподільного щита, розподільного пункту до розеток електроживлення.

– До одного контактного затискача не повинні підключатися на щиті нульовий робочий та нульовий захисний провідники.

3. Необхідно встановлювати аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення (крім освітлення) у приміщенні, де одночасно експлуатуються понад п’ять електронно-обчислювальних машин ЕОМ з окремими екранними та периферійними пристроями на помітному та доступному місці.

4. Не можна підключати комп’ютерну техніку до звичайної двопровідної електромережі.

5. Необхідно підключати комп’ютерну техніку до електромережі тільки за допомогою справних штепсельних з’єднань і електророзеток заводського виготовлення. У них, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника.

6. Додаткові вимоги безпеки:

– Не допускається використовувати тимчасові проводки. У випадки її необхідності вона повинна бути прокладена уздовж стіни не вище 2-2,5 м від підлоги

– Не можна залишати без нагляду включене електрообладнання під час перерв

– Покриття плит підлоги повинно бути гладеньким, міцним, антистатичним, таким, яке легко чистити пилососом або прибирати вологим способом. Необхідно забезпечити відведення з покриття підлоги статичних зарядів.

– Неприпустимо експлуатація пошкоджених кабелів, розеток, штемпельних з'єднань, кабелів без ізоляції, забороняється застосовувати для опалення саморобні пристрої або лампи розжарювання

– При виконанні ремонтно-налагоджувальних робіт електрообладнання повинно бути відключене, якщо ж електрообладнання залишається включеним, то ремонтно-налагоджувальних робіт повинні проводиться з використанням індивідуального захисту.

– Опір заземлення <4 ОМ, якщо U> 1000В, то R заземлення <5 ОМ

– Плити знімної підлоги повинні бути важко горючими, з межею вогнестійкості не менше 0,5 год., або негорючими

– Відкрита прокладка кабелів під підлогою забороняється.

– Підпільний простір під знімною підлогою має бути оснащений системою автоматичної пожежної сигналізації та засобами пожежогасіння

– Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів.

– Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника.

ВИСНОВОК

Кваліфікаційна робота магістра була присвячена вирішенню актуальної проблеми забезпечення якісної та ефективної практичної підготовки студентів в IT-сфері, що вимагає створення безпечних, масштабованих та економічно вигідних лабораторних інфраструктур. Для подолання викликів, пов'язаних із високою вартістю фізичних стендів, складністю їх масштабування та ризиком пошкодження інфраструктури під час студентських експериментів, було обрано та досліджено платформу віртуалізації з відкритим вихідним кодом Proxmox Virtual Environment (Proxmox VE).

На основі детального аналізу літературних джерел та узагальнення практичного досвіду інших авторів, що дозволило прийняти оптимальні проектні рішення та уникнути типових "вузьких місць", було розроблено структурну модель Віртуального Навчального Середовища (ВНС). У ході роботи описано функціональні блоки цієї моделі, їхнє призначення та надано загальні рекомендації щодо конфігурування, що робить проєкт готовим до практичної реалізації в невеликих навчальних закладах.

Практична частина роботи включала докладний опис процесів встановлення та початкового налаштування Proxmox VE, а також його ключових безпекових параметрів. Було проведено оцінку можливостей платформи, зокрема порівняння ефективності використання віртуальних машин KVM і контейнерів LXC. В результаті дослідження сформовано рекомендації, згідно з якими контейнери LXC оптимальні для швидкого розгортання ізольованих робочих місць для студентів завдяки високій продуктивності та низькому споживанню ресурсів, тоді як KVM-віртуальні машини слід використовувати для розміщення критичних сервісів, таких як Система дистанційного навчання (СДН) Moodle, забезпечуючи їх повну ізоляцію та гнучкість.

Окрему увагу приділено питанням захисту ВНС, включаючи налаштування резервного копіювання, моніторингу, використання утиліт для аудиту системи безпеки (rkhunter & Lynis), а також реалізацію захисту від атак повного перебору (Brute-Force) за допомогою Fail2ban для підвищення надійності та стійкості

середовища в цілому. Таким чином, запропонований комплексний підхід до використання Proxmox VE дозволяє ефективно створювати, масштабувати та управляти навчальними стендами, миттєво повертаючи їх до початкового стану за допомогою механізму знімків (Snapshots). Це забезпечує стабільне та безпечне функціонування навчально-лабораторної інфраструктури, мінімізує витрати на апаратне забезпечення та сприяє підвищенню якості практичної підготовки майбутніх фахівців. Мета роботи щодо розробки та реалізації надійного, захищеного та швидкодіючог віртуального середовища була повністю досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Попов О. О., Горошко С. В., Петренко О. М. Створення навчально-тренувального полігону для підготовки IT-фахівців на базі платформи віртуалізації Proxmox VE. Збірник наукових праць НУОУ імені Івана Черняховського. 2021. № 2(63). С. 132–139.

2. Григорук А. О., Нагачевська О. В. Особливості впровадження технологій віртуалізації в освітній процес (на прикладі Proxmox VE). Вісник Хмельницького національного університету. Технічні науки. 2020. № 6. С. 98–104.

3. Вовк Я. С. (ТНТУ), Романов О. М. Аналіз ефективності застосування гіпервізора Proxmox VE для побудови віртуальних лабораторій. Матеріали науково-технічної конференції ТНТУ ім. І. Пулюя. Тернопіль, 2022. С. 34–36.

4. Ковальчук В. О., Мельник В. П. Порівняльний аналіз систем керування віртуалізацією для навчальних цілей: Proxmox проти VMware. Наукові записки Ужгородського університету. Серія: Інформаційні технології. 2021. № 49. С. 112–117.

5. Панченко Є. В., Лисенко І. Г. Розробка методики автоматизованого розгортання віртуальних навчальних стендів на базі контейнерної віртуалізації LXC (Proxmox). Збірник наукових праць військового інституту телекомунікацій та інформатизації. 2019. № 3. С. 78–85.

6. Сидоренко А. С. Віртуалізація як інструмент підвищення якості практичного навчання в університетах. Педагогічний вісник. 2019. № 11. С. 154–159.

7. Кравчук П. М., Савчук І. Л. Архітектура розподіленого зберігання даних (Ceph) у кластері Proxmox для навчальних цілей. Сучасні інформаційні технології в освіті та науці. 2020. № 2. С. 220–225.

8. Мірошниченко І. О., Заболотний В. В. Організація мережевої взаємодії у віртуальному середовищі Proxmox: від Open vSwitch до Linux Bridge. Технології в освітньому просторі. 2021. № 12. С. 101–107.

9. Король М. А. (ТНТУ). Дослідження продуктивності KVM та контейнерів LXC у Proxmox VE для лабораторного практикуму. Звіт про НДР. Тернопіль, 2023. 80 с.

10. Reliability and Performance of Proxmox VE in Production Environments (TechRepublic, 2024)

11. Evaluating Fault Tolerance and HA in Proxmox Clusters (ResearchGate, 2024)

12. Benchmarking Proxmox VE for Educational Labs (Medium, 2023)

13. Proxmox VE with Ceph: Data Resilience under Heavy Load (Red Hat Tech Blog, 2024)

14. Smit, R. T. Virtualization in Education: A Comparative Study of Open-Source Hypervisors (Proxmox, oVirt, XenServer). International Journal of Computer Science Education. 2019. Vol. 14, No. 3. pp. 201–215.

15. Comparative Study: Proxmox VE vs VMware ESXi in Academic Deployments (IEEE Access, 2024)

16. Що таке NGINX і навіщо він потрібний URL: https://vps.ua/wiki/ukr/nginx/ .(дата звернення: 12.11.2025).

17. Nginx Proxy Manager URL: https://nginxproxymanager.com/ .(дата звернення: 19.11.2025).

18. Що таке Redis і навіщо він потрібен? URL: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-redis-i-zachem-on-nuzhen/ .(дата звернення: 21.11.2025).

19. HAProxy Logging Configuration Explained: How to Enable and View Log Files URL: https://sematext.com/blog/haproxy-logs/ .(дата звернення: 2.12.2025).

20. Redis: основні параметри конфігурації та налаштування продуктивності URL: https://rtfm.co.ua/ru/redis-osnovnye-parametry-konfiguracii-i-tyuning-proizvoditelnosti .(дата звернення: 3.12.2025).

21. Встановлення та налаштування Proxm VE. Створення віртуальних машин URL: https://freehost.com.ua/faq/articles/ustanovka-i-nastrojka-proxmox-ve-sozdanie-virtualnih-mashin/ .(дата звернення: 3.12.2025).

22. Аудит безпеки з Lynis в Ubuntu URL: https://www.8host.com/blog/audit-bezopasnosti-s-pomoshhyu-lynis-v-ubuntu-16-04/ .(дата звернення: 4.12.2025).

23. Захист сервера від Fail2ban. Як його налаштувати? URL: https://freehost.com.ua/faq/articles/zaschita-servera-sredstvom-utiliti-fail2ban-kak-ee-nastroit/?gad_source=1&gad_campaignid= 23198925533&gclid=CjwKCAiAl-_JBhBjEiwAn3rN7eNhij0Gm8_8dGoqsrYfwKxWPGfMkuJVHiKIqjr2aozwCUn43597vRoCtVIQAvD_BwE .(дата звернення: 4.12.2025).

24. https://oppb.com.ua/articles/vymogy-z-elektrobezpeky-dlya-fahivtsiv-sfery-it

25. https://studfile.net/preview/10659330/page:2/

26. Микитишин, А. Г., Митник, М. М., Голотенко, О. С., & Карташов, В. В. (2023). Комплексна безпека інформаційних мережевих систем. Навчальний посібник для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

27. ТИМОЩУК, Д., & ЯЦКІВ, В. (2024). USING HYPERVISORS TO CREATE A CYBER POLYGON. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (3), 52-56.

28. Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., Lechachenko T., (2024). Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, 3896, pp. 1-11.

29. Nedzelky, D., Derkach, M., Skarga-Bandurova, I., Shumova, L., Safonova, S., & Kardashuk, V. (2021, September). A Load Factor and its Impact on the Performance of a Multicore System with Shared Memory. In 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (Vol. 1, pp. 499-503). IEEE.

30. Деркач М. В., Хомишин В. Г., Гудзенко В. О. Тестування безпеки вебресурсу на базі інструментів для сканування та виявлення вразливостей. Наукові вісті Далівського університету. 2023. №25.

Додаток А Публікація

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

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

XIV Міжнародної науково-технічної конференції молодих учених та студентів

11-12 грудня 2025 року

УКРАЇНА
ТЕРНОПІЛЬ – 2025

УДК 004.056.5:378.147
В.С. Пиж, ст.гр. СБмд-61
(Тернопільський національний технічний університет, імені Івана Пулюя, Україна)

ІННОВАЦІЙНИЙ ПІДХІД ДО ПОБУДОВИ ЗАХИЩЕНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА ДЛЯ ОСВІТНЬОГО ПРОЦЕСУ

V. S. Pyzh, st. gr. SBmd-61

INNOVATIVE APPROACH TO BUILDING A SECURE VIRTUAL ENVIRONMENT FOR THE EDUCATIONAL PROCESS

Стрімка цифровізація освіти вимагає нових підходів до створення безпечних віртуальних платформ, здатних підтримувати якісний навчальний процес. З огляду на зростання кіберризиків важливим є розроблення рішень, що поєднують інноваційні технології та ефективні механізми захисту. У роботі розглядається концепція побудови захищеного віртуального середовища, яке відповідає сучасним викликам та потребам учасників освітнього процесу.

Для побудови навчального середовища було обрано Proxmox Virtual Environment (PVE).

Proxmox Virtual Environment (PVE) є потужним рішенням Type 1 ("bare-metal") для віртуалізації, що базується на Debian Linux та поєднує KVM, LXC та функції кластеризації. Завдяки відкритому коду (GNU AGPLv3) та відсутності обов'язкових ліцензійних платежів , PVE ідеально підходить для малого та середнього бізнесу, освітніх установ та домашніх лабораторій, пропонуючи функціонал рівня Enterprise з обмеженим бюджетом.

Ключові перевагами та архітектурою Proxmox VE є настпне:

Proxmox VE об’єднує два типи віртуалізації в одному веб-інтерфейсі: KVM для повної апаратної віртуалізації будь-яких ОС і LXC для легких контейнерів Linux із мінімальними накладними витратами. Такий підхід дозволяє ефективно ізолювати сервіси та одночасно підтримувати широкий спектр операційних систем.

Система підтримує вбудовану кластеризацію та високу доступність (HA). Corosync синхронізує конфігурацію між вузлами, менеджер HA автоматично перезапускає критичні ВМ або контейнери у разі збою, а жива міграція забезпечує переміщення працюючих ВМ без простою.

Proxmox VE також пропонує гнучку роботу зі сховищами: ZFS забезпечує цілісність даних, миттєві знімки та управління дисковими пулами, а інтеграція з Ceph дозволяє створити розподілене відмовостійке сховище на базі локальних дисків усіх вузлів, замінюючи потребу у дорогому зовнішньому SAN.

Передбачається встановлення та налаштування Proxmox Virtual Environment (PVE).

Важливим в роботі є питання захисту та рекомендації (кроки), покликані зробити навчальне середовище максимально безпечним. В наступній таблиці приведені основні напрямки безпеки та описано їх вирішення

Таблиця 1. Ключові напрями та кроки забезпечення безпеки в середовищі Proxmox

Напрямок безпеки	Ключові кроки
Доступ по SSH	Вимкнути вхід для root. Використовувати автентифікацію на основі SSH-ключів. Застосувати Fail2ban проти брутфорсу.
Мережевий Екран	Увімкнути фаєрвол Proxmox. Встановити політику заборонити все за замовчуванням (deny-all) та додати винятки лише для необхідних портів (8006/TCP, SSH).
Автентифікація	Увімкнути 2FA (TOTP, WebAuthn) для всіх адміністративних облікових записів.
Управління Доступом (RBAC)	Уникати root для повсякденних завдань. Створювати окремі облікові записи з мінімально необхідними ролями та дозволами.
Оновлення та Моніторинг	Регулярно встановлювати оновлення безпеки. Перенаправляти системні журнали (syslog) на централізований сервер журналювання.
Резервні Копії	Використовувати клієнтське шифрування та незмінні резервні копії (immutable backups) через Proxmox Backup Server (PBS).
Веб-Інтерфейс	Замінити самопідписаний сертифікат на сертифікат від довіреного ЦС (наприклад, Let's Encrypt).

Забезпечення інформаційної безпеки у віртуалізованому середовищі Proxmox є одним із ключових етапів його впровадження в освітній або корпоративній інфраструктурі. Використання широкого набору вбудованих механізмів захисту дозволяє підвищити стійкість системи до зовнішніх та внутрішніх загроз. Основна увага приділяється організації автентифікації користувачів, зокрема впровадженню двофакторної автентифікації (2FA), що є обов'язковою умовою для адміністративних облікових записів, включаючи root@pam.

Важливою складовою є захист каналу віддаленого доступу, який передбачає зміну стандартного SSH-порту, відмову від використання паролів та дозвіл доступу виключно за SSH-ключами. Додатково рекомендується створення окремого користувача з правами sudo та повне відключення входу під обліковим записом root, що зменшує ризики експлуатації вразливостей.

Суттєве значення має налаштування вбудованого фаєрвола Proxmox на рівні датацентру, встановлення політики DROP за замовчуванням та дозвіл лише критично необхідних портів, включно з обмеженням доступу за визначеними IP-адресами. Для підвищення стійкості до brute-force атак впроваджується Fail2ban, який контролює активність на SSH та веб-інтерфейсі Proxmox, встановлюючи низькі ліміти на кількість спроб входу та подовжені періоди блокування.

Регулярний аудит системи забезпечується через використання інструментів rkhunter та Lynis, що дозволяють виявляти потенційні вразливості конфігурації та стежити за станом системної безпеки. Завершальним елементом комплексного підходу виступає продумана стратегія резервного копіювання, яка включає щотижневі повні та щоденні інкрементальні копії з визначеною політикою зберігання, що гарантує відновлюваність критичних даних та безперервність роботи інфраструктури.

Proxmox VE є повноцінною альтернативою комерційним гіпервізорам, лідируючи в ніші Open Source платформ. Його переваги — інтеграція KVM/LXC, безкоштовна кластеризація та відсутність Vendor Lock-in — роблять його ідеальним вибором для SMB, яким потрібна висока продуктивність, гнучкість і значна економія коштів.

Література

1. Alfaif, Aam Kholid, Puspanda Hatta, and Endar Suprih Wihidayat. "Performance Analysis of Proxmox and Virtualbox with Overhead and Linearity Parameters to Support Server Administration Practice." *Journal of Informatics and Vocational Education* 7, no. 2 (2024): 35–41. https://doi.org/10.20961/joive.v7i2.2399. URL: https://doi.org/10.20961/joive.v7i2.2399.

2. Ramdhani, Muhammad Dzikri, Muhammad Barez Sapado Siregar, Hilma Hayati, Riki Maulana, and Muhammad Febristo Robidullah. "Meningkatkan Efisiensi dan Skalabilitas Infrastruktur TI dengan Proxmox VE: Studi Kasus Implementasi dan Konfigurasi."*Karimah Tauhid* 3, no. 8 (2024): 8631–36. URL: http://dx.doi.org/10.30997/karimahtauhid.v3i8.14326.

.

Додаток Б

Лістинг налаштування WEB server Nginx

```
server {
    listen 80;
    server_name moodle.example.com;

    # Перенаправлення HTTP на HTTPS
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;
    server_name moodle.example.com;

    # --- Налаштування SSL  ---
    ssl_certificate /etc/nginx/ssl/moodle.example.com.crt;
    ssl_certificate_key /etc/nginx/ssl/moodle.example.com.key;
    location / {
        # Внутрішня IP-адреса  Load Balancer
        set $upstream_server 10.10.10.20:80;

        # Перенаправлення трафіку на Load Balancer
        proxy_pass http://$upstream_server;
        # Передача оригінальних заголовків клієнта
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # Налаштування для Moodle та WebSocket
        proxy_read_timeout 90;
        proxy_redirect off;
    }
}
```

Додаток В

Лістинг конфігурації haproxy.cfg

```
global
    log /dev/log    local0
    maxconn 4000
    chroot /var/lib/haproxy
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

defaults
    log     global
    mode    http
    option  httplog
    option  dontlognull
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

# ---------------------------------------------------------
# FRONTEND: Приймає трафік від Proxy Manager (Nginx)
# ---------------------------------------------------------
frontend http_in
    # IP-адреса, на якій Load Balancer слухає вхідний трафік
    # (наприклад, 10.10.10.20:80  Nginx)
    bind *:80

    # Використовувати бекенд-пул "moodle_servers" для всіх запитів
    default_backend moodle_servers
# ---------------------------------------------------------
# BACKEND: Пул серверів Moodle
# ---------------------------------------------------------
backend moodle_servers
```

```
    # Алгоритм балансування: Round-Robin (почергове розподілення)
    balance roundrobin

    # 1. Перший сервер Moodle (замініть IP-адреси на внутрішні IP ваших контейнерів Moodle)
    # 'check' вмикає перевірку стану на порту 80
    server moodle1 10.10.10.30:80 check

    # 2. Другий сервер Moodle
    server moodle2 10.10.10.31:80 check

    # 3. Третій сервер Moodle
    server moodle3 10.10.10.32:80 check
```

Додаток Г

Лістинг конфігурації Redis

```
bind 0.0.0.0
# Порт за замовчуванням
port 6379
protected-mode yes
# ------------------------------------------------------------------
# GENERAL
# ------------------------------------------------------------------
# Запускається у фоновому режимі (no для контейнерів)
daemonize no


# Рівень деталізації логування
loglevel notice
# ------------------------------------------------------------------
# MEMORY MANAGEMENT
# ------------------------------------------------------------------
# Обмеження використання пам'яті.
maxmemory 2gb
# Політика витіснення (видалення) ключів при досягненні ліміту пам'яті.
maxmemory-policy allkeys-lru
# ------------------------------------------------------------------
# SNAPSHOTTING (Persistence)
# ------------------------------------------------------------------
# Вимкнення постійного зберігання (AOF та RDB) для ролі кешу.
save ""
appendonly no
```

Додаток Д

Лістинг налаштування конфігурації Moodle

```
// --- 1. НАЛАШТУВАННЯ КЕШУВАННЯ (CACHING) ---
$CFG->session_handler_class = '\core\session\handler\redis';
$CFG->session_redis_host = '10.10.10.40'; // IP-адреса контейнера Redis
$CFG->session_redis_port = 6379;
$CFG->session_redis_database = 0; // Можна використовувати різні бази даних (0-15)
$CFG->session_redis_auth = '';
$CFG->session_redis_prefix = 'moodle_session_';
$CFG->session_redis_acquire_lock_timeout = 120;
$CFG->session_redis_lock_expire = 7200;
// Перевіряємо, що кеш
// синхронізується між усіма веб-серверами (Moodle 1, 2, 3)
$CFG->session_cluster_id = 'webfarm';

// --- 2. НАЛАШТУВАННЯ СИСТЕМИ КЕШУВАННЯ Moodle ---

// Встановлення Redis для всіх кеш-сховищ
$CFG->cache_stores = array (
   'redis' => array (
      'type' => 'redis',
      'default' => true,
      'redis_server' => '10.10.10.40', // IP-адреса контейнера Redis
      'redis_port' => 6379,
      'redis_database' => 1, // Використовуємо іншу базу даних, ніж для сесій (база 1)
      'redis_auth' => '', // Пароль (якщо є)
      'redis_prefix' => 'moodle_cache_', // Префікс
   )
);

// Встановлення 'redis' як сховища за замовчуванням
$CFG->cacheconfig = array (
   'default' => array (
      'stores' => array ('redis'),
```

```
    ),
    'session' => array (
        'stores' => array ('redis'),
    )
);
```

Додаток Е

Лістинг налаштування бази даних

```
// --- НАЛАШТУВАННЯ БАЗИ ДАНИХ ---

// 1. Тип бази даних (для MariaDB/MySQL)
$CFG->dbtype    = 'mariadb'; // або 'mysqli'
$CFG->dblibrary = 'native';

// 2. Параметри підключення
$CFG->dbhost    = '10.10.10.50';  // Внутрішня IP-адреса вашого DB-контейнера
$CFG->dbname    = 'moodle';       // Назва бази даних, створеної вище
$CFG->dbuser    = 'moodleuser';   // Користувач бази даних
$CFG->dbpass    = 'StrongDBPassword_123'; // Пароль користувача
$CFG->dbpersist = false;
$CFG->dbport    = 3306;

// 3. Таблиці
$CFG->prefix    = 'mdl_'; // Префікс для всіх таблиць Moodle
// 4. Потрібен для кластерів Moodle
$CFG->wwwroot   = 'https://moodle.example.com'; // Замініть на ваш домен
$CFG->dataroot  = '/mnt/moodle_data'; // Шлях до каталогу з даними Moodle (див. примітку)
$CFG->admin     = 'admin';

// ... (інші налаштування)
```