КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.34.1247
УДК 004.7:004.451

Булатецький Віталій Вікторович
кандидат фіз.-мат. наук, доцент, доцент кафедри комп'ютерних наук та кібербезпеки
Волинський національний університет імені Лесі Українки, Луцьк, Україна
ORCID:0000-0002-9883-4550
Bulatetsky.Vitaly@vnu.edu.ua

АРХІТЕКТУРА ДОСТУПУ ДО ЛОКАЛЬНИХ ВЕБСЕРВІСІВ НА БАЗІ ВІДКРИТОЇ ІНФРАСТРУКТУРИ ВІРТУАЛІЗАЦІЇ

Анотація. В роботі запропоновано модель використання серверного середовища віртуалізації на основі відкритого програмного забезпечення Proxmox VE для розгортання веб-сервісів. Актуальність дослідження зумовлена необхідністю модернізації IT-інфраструктури закладів вищої освіти в умовах суворих апаратних обмежень та дефіциту доступних публічних IPv4-адрес. Вебсервіси пропонується реалізувати на окремих віртуальних машинах з приватними IP-адресами та відкритими гостьовими серверними операційними системами для легкого та простого резервного копіювання та незалежної безпечної роботи кожної із них. Шляхом використання reverse-proxy на окремій віртуальній машині-шлюзі, за допомогою мостів забезпечується можливість доступу до цільових вебсервісів з використанням лише однієї публічної IP-адреси з перспективою масштабування даної інфраструктури. Централізоване керування сертифікатами безпеки на рівні проксі-сервера усуває необхідність їх генерування для кожної віртуальної машини окремо. У роботі детально проаналізовано переваги використання файлової системи ZFS. Встановлено, що застосування алгоритмів стиснення даних (lz4) та механізму ZFS-знімків дозволяє не лише забезпечити цілісність інформації, а й суттєво підвищити продуктивність дискових операцій на застарілому обладнанні, мінімізуючи час відновлення системи після збоїв. Описано практичний досвід інтеграції розробленої інфраструктури з хмарним DNS-сервісом Cloudflare, що надає додатковий рівень захисту від зовнішніх загроз та оптимізує мережевий трафік. Проведене порівняльне дослідження з комерційними аналогами (VMware ESXi, Microsoft Hyper-V) підтвердило, що обрана архітектура на базі Proxmox VE демонструє значно нижчі накладні витрати на віртуалізацію, що є критичним фактором для ефективної експлуатації серверів із обмеженим обсягом оперативної пам'яті. Запропонована модель характеризується високим потенціалом до масштабування через можливість об'єднання вузлів у кластерні структури. Отримані результати можуть бути впроваджені у структурних підрозділах університетів як надійна, економічно вигідна та технологічно гнучка альтернатива пропрієтарним рішенням.

Ключові слова: Proxmox VE, Гіпервізор, KVM, віртуалізація, Ubuntu Server, Debian Linux, вебсервіс, файлова система ZFS, реверс-проксі, Cloudflare.

ВСТУП

Розвиток цифрових сервісів та збільшення їх кількості у закладах вищої освіти (ЗВО) зумовлює необхідність створення гнучких та безпечних моделей керування мережевою інфраструктурою. Традиційний підхід, що передбачає використання окремих фізичних серверів для кожного веб-ресурсу, на сьогодні є економічно та технічно неефективним.

У цій роботі розглядається впровадження архітектури доступу, що базується на поєднанні середовища віртуалізації та спеціалізованого вузла для здійснення доступу до ресурсів локальної мережі. Такий підхід дозволяє консолідувати обчислювальні ресурси на базі наявного апаратного забезпечення та забезпечити ефективну роботу великої кількості ізольованих локальних вебсервісів через єдину точку входу в умовах обмежених мережевих, апаратних та програмних ресурсів.

Постановка проблеми. Аналіз методів розгортання IT-ресурсів у сучасних закладах вищої освіти свідчить про домінування моделі, де для кожного окремого сервісу (або невеликої кількості однотипних сервісів) використовується виділений фізичний сервер із власною публічною IP-адресою. Доступ до ресурсів, розміщених безпосередньо на накопичувачах цих серверів, зазвичай організовується за допомогою спеціалізованого вільного ПЗ.

Такий підхід має низку суттєвих недоліків:

- неефективне використання апаратних ресурсів, оскільки фізичні сервери часто мають надлишкову потужність для одного сервісу, що призводить до простою обладнання та зайвих витрат на електроенергію;
- обмежений фонд доступних публічних IPv4-адрес ускладнює суттєве збільшення кількості незалежних вебресурсів, різні типи ресурсів вимагають одних і тих самих системних бібліотек, але різних версій для кожного із ресурсів, що унеможливлює розгортання кількох таких ресурсів на одному фізичному сервері;
- необхідність окремого налаштування та обслуговування для кожного фізичного вузла, що підвищує ризик виникнення помилок та вразливостей;
- відсутність централізованого шару засобів взаємодії ускладнює створення та відновлення повних образів систем у разі збоїв апаратного чи програмного забезпечення;
- складність швидкого перерозподілу обчислювальних ресурсів між різними завданнями залежно від актуальних потреб підрозділів ЗВО.

Задача виникла із реальної ситуації в одному із структурних підрозділів Волинського національного університету імені Лесі Українки, де з'явилась потреба розгортання комплексу вебресурсів та інших сервісів у межах однієї локальної мережі за умов суворих ресурсних обмежень. Ключовою проблемою стала доступність лише однієї публічної IPv4-адреси, що унеможливлює прямий доступ до великої кількості незалежних сервісів стандартними засобами та зумовлює необхідність використання адресного простору для приватних мереж [1]. Додатковими факторами, що визначають складність задачі, є можливість використання лише одного наявного вільного фізичного сервера**,** відсутність доступу до ліцензійних дистрибутивів пропрієтарних серверних ОС та інтеграція з зовнішнім DNS-сервісом Cloudflare у межах безкоштовного тарифного плану, що накладає обмеження на доменні імена пов'язані із субдоменами.

За таких умов постає задача створення архітектури, яка б забезпечила стабільне та безпечне функціонування сервісів та раціонального розподілу наявних потужностей.

Необхідність подолання вказаних недоліків традиційної моделі розгортання сервісів зумовлює пошук альтернативних архітектурних рішень на базі систем віртуалізації. Сучасний стан розвитку технологій із відкритим кодом пропонує широкий спектр інструментів, здатних забезпечити необхідний рівень продуктивності навіть за обмежених ресурсів. Однією з найбільш перспективних платформ для побудови таких інфраструктур є Proxmox VE, аналіз ефективності якої широко представлений у сучасних дослідженнях.

Аналіз останніх досліджень і публікацій. Сучасні тенденції в IT-інфраструктурі вказують на поступове переорієнтування від комерційних пропрієтарних рішень до систем з відкритим вихідним кодом. Платформа Proxmox Virtual Environment (віртуальне середовище Proxmox), яка працює на основі гіпервізора KVM (Kernel-based Virtual Machine – віртуальна машина на базі ядра Linux) та контейнеризації LXC (Linux Containers – технологія контейнерів у Linux), привертає значну увагу дослідників завдяки своїй гнучкості, високій продуктивності та економічній ефективності.

Ключовим аспектом при впровадженні технологій віртуалізації є оцінка рівня накладних витрат (overhead), тобто додаткових ресурсів, які витрачаються системою порівняно з роботою на фізичному обладнанні. Згідно з дослідженням [2], платформа Proxmox VE демонструє високу ефективність у задачах високопродуктивних обчислень (HPC). Зокрема, накладні витрати на використання процесора становлять лише 0,5-1,1% відносно рівня роботи на «голому залізі», що є кращим показником порівняно з використанням стандартного гіпервізора KVM. Крім того, система ефективно управляє оперативною пам'яттю. Завдяки механізмам динамічного розподілу ресурсів Proxmox VE споживає на 15-20% менше RAM у порівнянні з аналогічними конфігураціями KVM [2].

У результаті порівняльного аналізу з комерційною платформою VMware ESXi встановлено, що Proxmox VE у багатьох сценаріях демонструє суттєво вищу продуктивність. Зокрема, під час тестування файлових операцій у середовищах веб і поштових серверів його показники виявилися до трьох разів кращими [3]. Такий результат дослідники роботи [3] пов'язують із особливостями реалізації підсистеми зберігання даних. У Proxmox використовується менш абстрагований підхід до роботи з файловою системою, ніж у випадку VMFS у VMware ESXi.

У роботі [4] центральною визначено проблему недостатнього використання серверних ресурсів. На основі аналізу телеметричних даних реальних корпоративних систем встановлено, що така неефективність здебільшого зумовлена не обмеженнями платформи, а помилками планування, зокрема надлишковим виділенням ресурсів (over-provisioning). Автор [4] пропонує застосовувати стратегії динамічного балансування та консолідації навантажень у періоди зниженої активності, що дає змогу істотно зменшити загальну вартість володіння (TCO) та рівень енергоспоживання.

Для вищих навчальних закладів Proxmox VE виступає як база для створення «академічних хмар» [5]. Як засвідчують результати впровадження, описані у джерелах [5] і [6], Proxmox демонструє вищу зручність обслуговування, ніж Apache CloudStack, забезпечуючи при цьому рівноцінну продуктивність. Важливою перевагою для навчання є наявність просунутого консольного інтерфейсу та вбудованої системи резервного копіювання (Proxmox Backup Server), що критично для збереження робіт здобувачів освіти [5]. У роботі [7] проаналізовано досвід користувачів і встановлено перевагу Proxmox VE над такими десктопними рішеннями, як VirtualBox. Зокрема підкреслено, що використання гіпервізора 1-го типу є критично важливим для підготовки фахівців із системного адміністрування

У порівняльних дослідженнях з Xen та XenServer, Proxmox часто отримує перевагу через свою «гібридність» – одночасну підтримку повної віртуалізації (KVM) та контейнеризації (LXC/OpenVZ). Це робить його універсальним інструментом для організацій будь-якого розміру, незалежно від типу навантажень [8, 9].

Аналіз літератури підтверджує, що Proxmox VE є технологічно досконалою, високопродуктивною платформою, яка не лише не поступається комерційним аналогам, а й часто перевершує їх у специфічних сценаріях (Linux-орієнтовані навантаження, HPC). Для університетського середовища платформа є оптимальною завдяки поєднанню простоти управління, потужних інструментів резервного копіювання та відсутності ліцензійних витрат.

Метою роботи є реалізація та дослідження ефективності архітектури на базі Proxmox VE для консолідації вебсервісів в умовах обмежених ресурсів та аналіз особливостей її масштабування.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ ДОСЛІДЖЕННЯ

Запропонована та реалізована типова інфраструктура, яка використовується в подібних ситуаціях (дефіцит мережевих ідентифікаторів), але з використанням середовища віртуалізації. На рис. 1 представлена логічна структура системи. Запити з мережі Інтернет через Cloudflare DNS потрапляють на фізичний хост Dell PowerEdge 1950, де гіпервізор Proxmox VE спрямовує їх через віртуальний міст vmbr0 на Reverse Proxy (VM100). Після SSL-термінації трафік через ізольований міст vmbr1 розподіляється між цільовими серверами (VM101-VM103) за протоколом HTTP.

В якості фізичного сервера було використано класичний Dell PowerEdge 1950 з наступними характристиками:

CPU: Intel(R) Xeon(R) CPU X5355 @ 2.66GHz (1 Socket)
RAM: DDR2 FB-DIMM 13GB 667 MT/s ECC 72 bit (64 data + 8 parity control)
HDD: FS/RAID**:** ZFS RAID 1 (Mirr.) WD RED Plus WD10EFRX-68F SATA III.
NIC**:** 2 x Broadcom NetXtreme II BCM5708 Gigabit Ethernet (10/100/1000 Mbps).

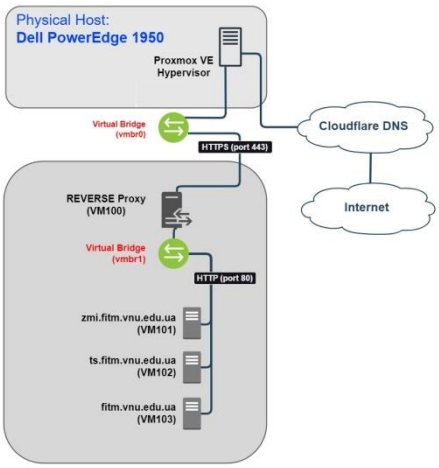

Рис.1. Блок-схема запропонованої інфраструктури на базі гіпервізора Proxmox VE. Наявна дворівнева ізоляція мережевих сервісів та логіка механізму централізованої SSL-термінації через реверс-проксі.

Доступ до глобальної мережі забезпечувався симетричним каналом зв’язку з пропускною здатністю 50-100 Мбіт/с (Full-duplex). Дана швидкість є достатньою для стабільної обробки вхідних HTTPS-запитів та віддачі контенту з внутрішніх віртуальних машин без виникнення мережевих затримок. Використання гігабітних внутрішніх інтерфейсів сервера дозволило уникнути "вузьких місць" при маршрутизації трафіку між реверс-проксі та цільовими серверами всередині гіпервізора.

Основою інфраструктури став Proxmox Virtual Environment 8.4 (остання стабільна версія на момент розгортання), як поєднання двох технологій віртуалізації на одній платформі. Це забезпечує максимальну гнучкість робочого середовища. Система дозволяє використовувати повну віртуалізацію KVM для образів Windows і Linux, а також легковагові контейнери для запуску ізольованих Linux-застосунків без виникнення системних конфліктів [10]. Версія 8.4 базується на Debian 12.5 та ядрі Linux 6.8, і підтримує застаріле обладнання, яке ми вимушені використовувати.

На відміну від vSphere (ESXi), яка має досить жорсткі вимоги до CPU та RAM, вибір Proxmox VE 8.4 замість VMware vSphere 8.0 обумовлений не лише ліцензійною політикою, а й апаратними обмеженнями: сучасні версії vSphere припинили підтримку процесорів архітектури Intel Clovertown (Xeon X53xx) та вимагають значно більшого обсягу зарезервованої оперативної пам'яті (від 8 ГБ) для функціонування самого гіпервізора проти 1 ГБ у Proxmox VE 8.4 , що було б критичним при загальному обсязі RAM у 13 ГБ. [10, 11, 12]

Порівнючи з Hyper-V (це рішення часто обирають як «стандартне» для Windows-інфраструктур), який вимагає Windows Server і споживає багато ресурсів, Proxmox споживає їх мінімум, залишаючи більше місця для ваших віртуальних машин. Proxmox VE забезпечує нативну підтримку LXC-контейнерів, що дозволяє запускати ізольовані сервіси з мінімальними накладними витратами, тоді як Hyper-V орієнтований переважно на повну віртуалізацію, що потребує більше апаратних потужностей [13, 14].

Дискова підсистема сервера базується на двох накопичувачах WD Red. Для забезпечення відмовостійкості та цілісності даних використано файлову систему ZFS. Кожен фізичний диск має ідентичну структуру розділів: BIOS boot (partition 1) розділ для завантажувача, EFI (partition 2) розділ для підтримки сучасних інтерфейсів завантаження та ZFS (partition 3) – основний розділ об’ємом ~999 ГБ, який є частиною дзеркального пулу (Mirror).

Незважаючи на те, що контейнери споживають менше ресурсів, для даного проекту було обрано повну віртуалізацію KVM, оскільки, на відміну від контейнерів, які використовують спільне ядро з хостом, кожна ВМ має власне віртуалізоване ядро. Це створює бар’єр для потенційних атак: якщо зловмисник отримає доступ до одного з бекенд-серверів (zmi або ts), він залишиться без прямого доступу до ресурсів гіпервізора або інших сервісів.

Використання VirtIO адаптерів у поєднанні з віртуальними мостами vmbr0 та vmbr1 дозволяє на програмному рівні імітувати фізичне розділення мереж, що значно складніше реалізувати та адмініструвати в контейнерному середовищі без втрати безпеки. KVM дозволяє жорстко зарезервувати ресурси (RAM та CPU) за кожною машиною. Це виключає ситуації, коли одна ВМ може "захопити" всі ресурси хоста, що іноді трапляється в контейнерах, призводячи до збою всієї системи. Використання блочних пристроїв ZVOL для віртуальних машин забезпечує найвищий рівень стабільності при роботі з дзеркальним рейдом. Такий підхід до організації сховища корелює з позитивним досвідом використання Proxmox VE для розгортання віртуалізованих об’єктів в академічних середовищах, проте у нашому випадку він був адаптований для специфічних вимог продуктивності серверної частини [6]. Це гарантує, що операції запису будуть завершені коректно навіть при високому навантаженні на дискову підсистему [15]. Завдяки повній віртуалізації, створення Snapshot в Proxmox фіксує повний стан системи, включаючи стан ядра та пам’яті. Кожна ВМ живе власним життєвим циклом, що спрощує адміністрування та планування технічних робіт.

На базі розгорнутого гіпервізора створено серію віртуальних машин, всі під керуванням Ubuntu 24.04.3 LTS. Розподіл апаратних ресурсів було оптимізовано з урахуванням загального обсягу оперативної пам’яті сервера (13 ГБ) та специфіки роботи вебсервісів.

VM100 (rproxy) – Reverse Proxy.1 vCPU (kvm64), оперативна пам’ять: 3 ГБ, дискова підсистема**:** 50 ГБ на local-zfs (VirtIO SCSI) (Рис.2). Функціонує у режимі мережевого мосту між двома сегментами мережі. Перший інтерфейс підключено до зовнішнього мосту vmbr0 для прийому вхідних запитів, тоді як другий інтерфейс інтегровано у внутрішню мережу vmbr1. Це дозволяє rproxy виступати єдиним легітимним шлюзом, через який зовнішній трафік може потрапити до бекенд-серверів, що не мають прямого виходу в інтернет.

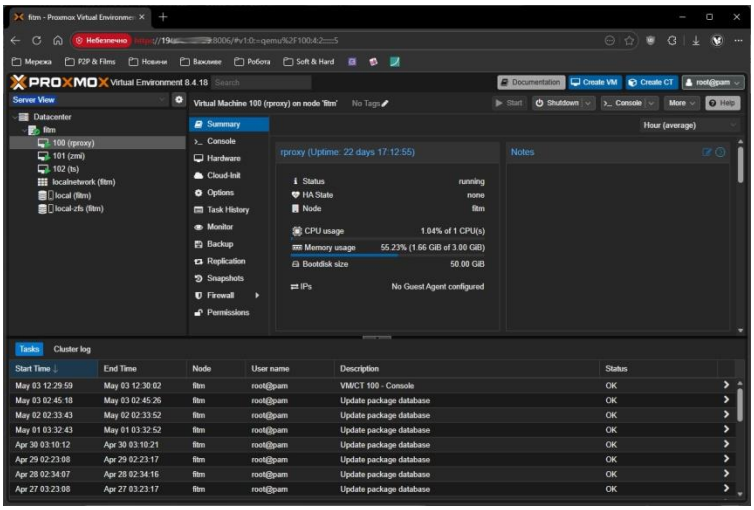

Рис.2. Панель керування віртуальною машиною-шлюзом (VM100) у середовищі Proxmox VE 8.4

Для віртуальної машини VM100, що виконує роль реверс-проксі, обрано операційну систему Ubuntu 24.04.3 LTS із розгорнутим вебсервером Nginx. Це дозволяє централізовано керувати вхідним трафіком, здійснюючи SSL-термінацію на рівні шлюзу та розвантажуючи обчислювальні потужності бекенд-серверів. Завдяки механізму маршрутизації на основі заголовків, система забезпечує коректну трансляцію запитів до різних внутрішніх ресурсів, використовуючи лише одну публічну IP-адресу [16, 17].

Особливістю налагодження трансляції через цей вузол є його одночасна присутність у двох ізольованих мережевих сегментах через віртуальні мости vmbr0 та vmbr1. Nginx приймає зовнішні пакети, перевіряє їх на відповідність правилам безпеки та перенаправляє оброблений трафік на внутрішні IP-адреси віртуальних машин у закритій мережі. Такий підхід гарантує, що цільові сервери залишаються фізично недоступними для прямого впливу, що значно підвищує загальний рівень безпеки інфраструктури [18].

Реалізація IP-таблиць на VM100 базується на Netfilter і дозволяє створити жорсткий набір правил фільтрації на межі двох мережевих мостів. Конфігурація IP-таблиць на VM100 чітко відповідає схемі руху трафіку: на зовнішньому інтерфейсі: дозволено лише порти 80 та 443 для взаємодії з Cloudflare та порт 22 для адміністративного доступу. При цьому фільтрація налаштована таким чином, що вхідний трафік на порт 443 термінується проксі-сервером, а подальша трансляція запитів у внутрішній сегмент vmbr1 відбувається виключно на порт 80. Це дозволяє поєднати високий рівень безпеки зовнішнього з'єднання з мінімальним навантаженням на внутрішні вебвузли (VM101,VM102, …), які завдяки правилам Firewall залишаються повністю закритими для будь-яких прямих запитів ззовні.

Стосовно інших віртуальних машин, їхня роль в інфраструктурі полягає у безпосередньому хостингу вебсайтів підрозділу. На відміну від шлюзу, вони знаходяться у глибокій ізоляції. Кожна з цих машин працює під керуванням Ubuntu 24.04.3 LTS. Використання ідентичної ОС для всіх бекенд-вузлів дозволяє стандартизувати адміністрування та спростити процес розгортання оновлень. На них розгорнуто стандартний вебстек (наприклад, Apache або Nginx), який обробляє запити, що надходять від реверс-проксі. Вузли мають лише внутрішні адреси (наприклад, з діапазону 10.0.0.x), що робить їх невидимими для сканерів портів та автоматизованих атак з інтернету. Весь вхідний трафік на ці машини потрапляє тільки з внутрішнього інтерфейсу VM100. Це дозволяє серверам "довіряти" запитам, оскільки вони вже пройшли очищення та SSL-термінацію на рівні шлюзу. Для завантаження оновлень ПЗ налаштовано шлюз через VM100, що дозволяє контролювати навіть вихідну активність серверів.

Зважаючи на обмеження фізичного сервера кожній віртуальній машині виділено від 3 до 4 ГБ RAM без використання динамічного перерозподілу. Це гарантує стабільну швидкість роботи PHP/Python скриптів та баз даних без затримок на рівні гіпервізора. Образи дисків розміщені на local-zfs. Завдяки активованому стисненню lz4 резервних копій, реальний об'єм, який займає Ubuntu Server на дисках WD Red, значно менший за номінально виділені 100 ГБ. Це також підвищує швидкість читання даних, оскільки процесор Xeon X5355 швидше розпаковує дані в пам'яті, ніж старі диски зчитують їх фізично. Оскільки сервери сайтів часто потребують оновлення контенту або плагінів, використання ZFS-знімків дозволяє адміністратору швидко і оперативно зробити "зліпок". У разі невдалого оновлення, сайт повертається до робочого стану однією командою в консолі Proxmox.

Організація віддаленого доступу для адміністраторів віртуальних серверів реалізована через механізм прокидання портів (Port Forwarding) на шлюзі. Замість використання стандартного 22-го порту, для кожного внутрішнього сервера виділено унікальний зовнішній порт на публічній IP-адресі реверс-проксі, що дозволяє суб-адміністраторам підключатися безпосередньо до своїх робочих середовищ Ubuntu Server за допомогою стандартних SSH-клієнтів. Таке рішення забезпечує ізоляцію адміністративного трафіку та дозволяє централізовано обмежувати доступ до цих портів на рівні IP-таблиць VM100, надаючи фахівцям можливість повноцінно керувати програмним стеком та контентом сайтів без надання їм доступу до загального інтерфейсу керування гіпервізором Proxmox.

Оскільки проект реалізується в межах університету, інтеграція з Cloudflare стала логічним завершенням архітектури, забезпечуючи зв'язок між публічними доменними іменами та внутрішніми ресурсами на Proxmox. У межах проекту використовується основне доменне ім'я факультету, на якому розгорнуто дерево субдоменів для кожного віртуального сервера. Налаштування DNS-записів типу A у панелі керування Cloudflare спрямовує запити для zmi.fitm.vnu.edu.ua, ts.fitm.vnu.edu.ua та fitm.vnu.edu.ua на єдину публічну IP-адресу сервера Dell PowerEdge 1950. Така структура дозволяє Reverse Proxy (VM100) ідентифікувати, до якого саме ресурсу звертається користувач, і перенаправляти трафік у внутрішню мережу через vmbr1 до відповідної ВМ.

На момент написання статті успішно розгорнуто середовище Proxmox VE 8.4 на фізичному сервері з однією публічною IP-адресою, у межах якого створено ряд віртуальних машин для забезпечення реверс-проксіювання на базі Ubuntu 24.04.3 LTS та Nginx, а також хостингів із гостьовими операційними системами Ubuntu 24.04.3 LTS зі стеком LAMP. Практична експлуатація підтвердила високу стабільність обраної архітектури на застарілому обладнанні, що ілюструється безперервним часом роботи шлюзу понад 22 дні при мінімальному навантаженні на центральний процесор (близько 1-2%) та раціональному використанні оперативної пам'яті. Наше дослідження підтверджує висновки Л. Еммунгіла [4] щодо того, що в корпоративних дата-центрах часто спостерігається системне недовантаження CPU при критичному дефіциті оперативної пам'яті через неоптимальний розподіл ресурсів ВМ.

Наразі відповідні ресурси підрозділу університету функціонують у штатному режимі та знаходяться в процесі наповнення контентом, а користувачам віртуальних серверів надані необхідні облікові дані та інструкції для автономного керування своїми робочими середовищами.

Аналіз продуктивності дискової підсистеми показав, що впровадження файлової системи ZFS із алгоритмом стиснення lz4 дозволило значно прискорити операції читання та запису даних, компенсуючи апаратні обмеження інтерфейсів SATA. Це забезпечило швидкий відгук вебсервісів навіть при одночасному зверненні до декількох віртуальних машин. Реалізована схема мережевої ізоляції через віртуальні мости досить ефективна, оскільки бекенд-сервери залишаються повністю закритими для зовнішніх атак, отримуючи лише очищений трафік через шлюз [19]. Організація віддаленого адміністрування через Port Forwarding на нестандартні SSH-порти дозволила розмежувати зони відповідальності фахівців без необхідності надання їм доступу до загального інтерфейсу керування гіпервізором.

Під час інтеграції з Cloudflare було виявлено обмеження безкоштовного тарифного плану щодо проксіювання глибоко вкладених субдоменів та відсутності підтримки wildcard-сертифікатів для них. Зокрема, стандартний проксі-статус обмежує роботу лише вебпортами (80, 443), що робить неможливим пряме проксіювання TCP-запитів для SSH на виділені порти суб-адміністраторів без переведення цих записів у режим «DNS Only» [20]. На даний момент опрацьовується рішення щодо переходу до корпоративного тарифного плану Cloudflare для повноцінної підтримки ієрархічної структури доменних імен, а на перехідний період для субдоменних імен вимкнено проксіювання Cloudflare. В цілому, отримані результати підтверджують доцільність використання відкритого ПЗ віртуалізації для ревіталізації наявного парку серверного обладнання та побудови масштабованої ІТ-інфраструктури навчального закладу в умовах дефіциту ресурсів.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Впроваджена архітектура на базі Proxmox VE 8.4 підтвердила ефективність використання відкритого ПЗ для консолідації вебсервісів в умовах обмежених апаратних ресурсів та дефіциту публічних IP-адрес. Ключовою перевагою обраного підходу стала можливість легкого централізованого резервного копіювання та швидкого відновлення всієї інфраструктури або окремих вузлів через механізм ZFS-знімків, що мінімізує час простою сервісів у разі збоїв. Застосування реверс-проксіювання у поєднанні з ізольованими віртуальними мостами дозволило надати безпечний доступ до внутрішніх ресурсів через єдину точку входу без суттєвої втрати продуктивності системи.

Важливою перевагою обраної архітектури є її висока масштабованість: вона дозволяє в подальшому об'єднувати додаткові фізичні сервери локальної мережі у єдиний кластер під керуванням Proxmox VE, забезпечуючи централізоване керування ресурсами та гнучкий розподіл навантаження між вузлами.

Водночас дослідження виявило певні обмеження, зокрема залежність мережі від працездатності вузла шлюзу та апаратні ліміти застарілого обладнання. Також встановлено, що використання безкоштовних тарифів DNS-сервісів ускладнює автоматизацію керування сертифікатами для розгалужених структур субдоменів та накладає суворі обмеження на використання нестандартних портів [20], що вимагає додаткового налаштування VPN або переведення записів у режим «DNS Only», знижуючи рівень захищеності вузлів. Попри це, створена модель є життєздатною та придатною для впровадження у структурних підрозділах ЗВО як економічно вигідна і надійна альтернатива пропрієтарним рішенням віртуалізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., & Lear, E. (1996). *Address allocation for private internets* (RFC 1918). Internet Engineering Task Force. https://doi.org/10.17487/RFC1918
2. Izoulet, A., Beserra, D., Espie, M., Endo, P. T., & Araujo, J. (2025). Performance evaluation of Proxmox for high-performance computing in resource-shared environments. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 4096-4101). IEEE. https://doi.org/10.1109/SMC58881.2025.11342990
3. Đorđević, B., Timčenko, V., Nedeljković, N., & Davidović, N. (2021). ESXi and Proxmox: FileSystem performance comparison for type-1 hypervisors. In *Proceedings of the 8th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2021)*. https://www.etran.rs/2021/zbornik/Papers/102_RTI_2.3.pdf
4. Emmungil, L. (2025). An empirical analysis of server utilization and optimization strategies in a Proxmox VE environment. *Uygulamalı Mühendislik ve Tarım Dergisi, 2*(2), 23-30. https://izlik.org/JA54XG52BP
5. Oleksiuk, V., Oleksiuk, O., & Spirin, O. (2021). Comparative study of the support of academic clouds based on Apache CloudStack and Proxmox VE platforms. In *Myroslav I. Zhaldak Symposium on Advances in Educational Technology*. SCITEPRESS – Science and Technology Publications. https://doi.org/10.5220/0012064300003431
6. Oleksiuk, V., & Oleksiuk, O. (2022). The practice of developing the academic cloud using the Proxmox VE platform. *Educational Technology Quarterly*. https://doi.org/10.55056/etq.36
7. Simon, M., & Huraj, L. (2023). VirtualBox and Proxmox VE in network management: A user-centered comparison for university environments. In R. Silhavy & P. Silhavy (Eds.), *Networks and Systems in Cybernetics* (Lecture Notes in Networks and Systems, Vol. 723). Springer. https://doi.org/10.1007/978-3-031-35317-8_44
8. Algarni, S. A., Ikbal, M. R., Alroobaea, R., Ghiduk, A. S., & Nadeem, F. (2018). Performance evaluation of Xen, KVM, and Proxmox hypervisors. *International Journal of Open Source Software and Processes, 9*(2), 39–54. https://doi.org/10.4018/IJOSSP.2018040103
9. Ally, S. (2018). Comparative analysis of Proxmox VE and XenServer as type 1 open source based hypervisors. *International Journal of Scientific & Technology Research, 7*(3), 72-77.
10. Proxmox Server Solutions GmbH. (n.d.). *Features*. Proxmox Virtual Environment. https://www.proxmox.com/en/products/proxmox-virtual-environment/features
11. Broadcom. (n.d.). *Broadcom compatibility guide*. https://compatibilityguide.broadcom.com/
12. Broadcom. (2026, April 17). *VMware vSphere 8.0*. TechDocs. https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere/8-0.html
13. Microsoft. (n.d.). *Hardware requirements for Windows Server*. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/get-started/hardware-requirements
14. Microsoft. (n.d.). *Hyper-V virtualization in Windows Server and Windows*. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview
15. OpenZFS Project. (n.d.). *OpenZFS documentation*. https://openzfs.github.io/openzfs-docs/
16. F5, Inc. (n.d.). *NGINX reverse proxy*. NGINX Documentation. https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/
17. Nemeth, E., Snyder, G., Hein, T. R., Whaley, B., & Mackin, D. (2017). *UNIX and Linux system administration handbook* (5th ed.). Addison-Wesley Professional.
18. SANS Institute. (n.d.). *Designing a DMZ*. https://www.sans.org/white-papers/950

19. National Institute of Standards and Technology. (n.d.). *Defense in depth*. NIST Computer Security Resource Center. https://csrc.nist.gov/glossary/term/defense_in_depth
20. Cloudflare. (n.d.). *Network ports*. Cloudflare Docs. https://developers.cloudflare.com/fundamentals/reference/network-ports/

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023

Vitalii Bulatetskyi
PhD, Associate Professor, Associate Professor at the Department of Computer Science and Cybersecurity
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
ORCID:0000-0002-9883-4550
Bulatetsky.Vitaly@vnu.edu.ua

ACCESS ARCHITECTURE FOR LOCAL WEB SERVICES BASED ON OPENSOURCE VIRTUALIZATION INFRASTRUCTURE

Abstract. The paper proposes a model for using a server virtualization environment based on the Proxmox VE open-source software for web service deployment. The relevance of the study is driven by the need to modernize the IT infrastructure of higher education institutions under conditions of severe hardware constraints and a shortage of available public IPv4 addresses. It is suggested to implement web services on separate virtual machines with private IP addresses and open-source guest server operating systems to ensure easy backup and the independent, secure operation of each instance. By utilizing a reverse proxy on a dedicated gateway virtual machine, bridging enables access to target web services using only a single public IP address, with the prospect of further infrastructure scaling. Centralized management of security certificates at the proxy server level eliminates the need to generate them for each virtual machine individually. The paper provides a detailed analysis of the advantages of using the ZFS file system. It was established that the application of data compression algorithms (lz4) and the ZFS snapshot mechanism not only ensures information integrity but also significantly enhances disk I/O performance on legacy hardware, minimizing system recovery time after failures. The practical experience of integrating the developed infrastructure with the Cloudflare cloud DNS service is described, providing an additional layer of protection against external threats and optimizing network traffic. A comparative study with commercial counterparts (VMware ESXi, Microsoft Hyper-V) confirmed that the chosen Proxmox VE-based architecture demonstrates significantly lower virtualization overhead, which is a critical factor for the efficient operation of servers with limited RAM. The proposed model is characterized by high scaling potential through the possibility of combining nodes into cluster structures. The results obtained can be implemented in university departments as a reliable, cost-effective, and technologically flexible alternative to proprietary solutions.

Keywords: Proxmox VE, KVM hypervisor, virtualization, Ubuntu Server, Debian Linux, web service, ZFS file system, reverse proxy, virtual machine, Cloudflare.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., & Lear, E. (1996). *Address allocation for private internets* (RFC 1918). Internet Engineering Task Force. https://doi.org/10.17487/RFC1918
2. Izoulet, A., Beserra, D., Espie, M., Endo, P. T., & Araujo, J. (2025). Performance evaluation of Proxmox for high-performance computing in resource-shared environments. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 4096-4101). IEEE. https://doi.org/10.1109/SMC58881.2025.11342990
3. Đorđević, B., Timčenko, V., Nedeljković, N., & Davidović, N. (2021). ESXi and Proxmox: FileSystem performance comparison for type-1 hypervisors. In *Proceedings of the 8th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2021)*. https://www.etran.rs/2021/zbornik/Papers/102_RTI_2.3.pdf
4. Emmungil, L. (2025). An empirical analysis of server utilization and optimization strategies in a Proxmox VE environment. *Uygulamalı Mühendislik ve Tarım Dergisi, 2*(2), 23-30. https://izlik.org/JA54XG52BP
5. Oleksiuk, V., Oleksiuk, O., & Spirin, O. (2021). Comparative study of the support of academic clouds based on Apache CloudStack and Proxmox VE platforms. In *Myroslav I. Zhaldak Symposium on Advances in Educational Technology*. SCITEPRESS – Science and Technology Publications. https://doi.org/10.5220/0012064300003431
6. Oleksiuk, V., & Oleksiuk, O. (2022). The practice of developing the academic cloud using the Proxmox VE platform. *Educational Technology Quarterly*. https://doi.org/10.55056/etq.36

7. Simon, M., & Huraj, L. (2023). VirtualBox and Proxmox VE in network management: A user-centered comparison for university environments. In R. Silhavy & P. Silhavy (Eds.), *Networks and Systems in Cybernetics* (Lecture Notes in Networks and Systems, Vol. 723). Springer. https://doi.org/10.1007/978-3-031-35317-8_44
8. Algarni, S. A., Ikbal, M. R., Alroobaea, R., Ghiduk, A. S., & Nadeem, F. (2018). Performance evaluation of Xen, KVM, and Proxmox hypervisors. *International Journal of Open Source Software and Processes, 9*(2), 39–54. https://doi.org/10.4018/IJOSSP.2018040103
9. Ally, S. (2018). Comparative analysis of Proxmox VE and XenServer as type 1 open source based hypervisors. *International Journal of Scientific & Technology Research, 7*(3), 72-77.
10. Proxmox Server Solutions GmbH. (n.d.). *Features*. Proxmox Virtual Environment. https://www.proxmox.com/en/products/proxmox-virtual-environment/features
11. Broadcom. (n.d.). *Broadcom compatibility guide*. https://compatibilityguide.broadcom.com/
12. Broadcom. (2026, April 17). *VMware vSphere 8.0*. TechDocs. https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere/8-0.html
13. Microsoft. (n.d.). *Hardware requirements for Windows Server*. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/get-started/hardware-requirements
14. Microsoft. (n.d.). *Hyper-V virtualization in Windows Server and Windows*. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview
15. OpenZFS Project. (n.d.). *OpenZFS documentation.* https://openzfs.github.io/openzfs-docs/
16. F5, Inc. (n.d.). *NGINX reverse proxy*. NGINX Documentation. https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/
17. Nemeth, E., Snyder, G., Hein, T. R., Whaley, B., & Mackin, D. (2017). *UNIX and Linux system administration handbook* (5th ed.). Addison-Wesley Professional.
18. SANS Institute. (n.d.). *Designing a DMZ*. https://www.sans.org/white-papers/950
19. National Institute of Standards and Technology. (n.d.). *Defense in depth*. NIST Computer Security Resource Center. https://csrc.nist.gov/glossary/term/defense_in_depth
20. Cloudflare. (n.d.). *Network ports*. Cloudflare Docs. https://developers.cloudflare.com/fundamentals/reference/network-ports/

Отримано редакцією журналу / Received: 23.01.26
Прорецензовано / Revised: 05.02.26
Схвалено до друку / Accepted: 24.09.26