ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система адміністрування серверів та управління мережевою інфраструктурою веб-сайтів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Ілля КУЛИКОВСЬКИЙ*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-42
Ілля КУЛИКОВСЬКИЙ
Ім'я, ПРІЗВИЩЕ

Керівник: Ірина КУЗЬМІЧ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерної інженерії

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2026р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Куликовський Ілля Володимирович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система адміністрування серверів та управління мережевою інфраструктурою веб-сайтів»
керівник кваліфікаційної роботи Ірина КУЗЬМІЧ,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» лютого 2026р. № 51
2. Строк подання кваліфікаційної роботи «8» травня 2026р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література; науково-технічна література з питань мережевих технологій та адміністрування серверів; сучасні серверні платформи та операційні системи (Linux); веб-сервери (Nginx, Apache); системи управління базами даних (MySQL, PostgreSQL); технології віртуалізації (Docker, VMware, Proxmox); хмарні технології (IaaS, PaaS, SaaS); мережеві протоколи (TCP/IP, HTTP/HTTPS, DNS, DHCP); технології адресації IPv4/IPv6; моделі мереж (клієнт-серверна, багаторівнева архітектура); засоби автоматизації (Bash, Ansible); системи моніторингу (Netdata, Prometheus, Grafana); засоби забезпечення безпеки (Firewall, Fail2Ban, SSL/TLS); методи резервного копіювання та відновлення даних;
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Теоретичні основи мережевої інфраструктури веб-сайтів та адміністрування серверів
   2. Сучасні технології та рішення у побудові серверного середовища веб-сайтів
   3. Розгортання, адміністрування, моніторинг та забезпечення безпеки серверної інфраструктури веб-сайту
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «20» лютого 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-24.03.2026	Виконано
2	Обробка матеріалу	23.03-05.04.2026	Виконано
3	Розробка теоретичної частини	06.04-10.04.2026	Виконано
4	Розробка моделі мережі	11.04-20.04.2026	Виконано
5	Висновки за результатами аналізу	21.04-06.05.2026	Виконано
6	Оформлення пояснювальної записки	07.05-08.05.2026	Виконано
7	Розробка презентації	09.05-11.05.2026	Виконано
8	Передзахист	11.05-15.05.2026	Виконано
9	Здача в деканат	18.05-21.05.2026	Виконано

Здобувач вищої освіти _____ Ілля КУЛИКОВСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ірина КУЗЬМІЧ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 58 стор., 14 рис., 10 табл., 50 джерел.

Мета роботи – вивчення системи адміністрування серверів та управління мережевою інфраструктурою веб-сайтів.

Об'єкт дослідження – функціональні процеси та процеси розвитку мережевої інфраструктури веб-сайтів у контексті забезпечення їх стабільної, безпечної та ефективної роботи.

Предмет дослідження – методи, технології та інструменти системи адміністрування серверів та управління мережевою інфраструктурою веб-сайтів, включаючи інструменти моніторингу, безпеки та оптимізації ресурсів.

Короткий зміст роботи: Сучасні веб-сайти та інформаційні системи залежать від надійної мережевої інфраструктури та ефективного серверного середовища. Зростання кількості користувачів і обсягів даних підвищує вимоги до стабільності, безпеки та продуктивності, тому важливим є впровадження сучасних підходів до розгортання, адміністрування та захисту серверів.

У роботі розглянуто основи мережевої інфраструктури веб-сайтів, принципи адміністрування серверів і сучасні технології, зокрема віртуалізацію, контейнеризацію та хмарні рішення. У практичній частині досліджено процес розгортання серверного середовища, визначено етапи впровадження системи адміністрування та запропоновано підхід до моніторингу, безпеки й підтримки серверів, що забезпечує стабільну та безпечну роботу веб-ресурсів.

На основі проведеного дослідження визначено основні підходи до побудови системи адміністрування серверної інфраструктури веб-сайтів, обґрунтовано їх ефективність і доцільність використання в сучасних умовах.

КЛЮЧОВІ СЛОВА: МЕРЕЖЕВА ІНФРАСТРУКТУРА, АДМІНІСТРУВАННЯ СЕРВЕРІВ, ВЕБ-САЙТ, ВІРТУАЛІЗАЦІЯ, КОНТЕЙНЕРИЗАЦІЯ, ХМАРНІ ТЕХНОЛОГІЇ, IAAS, PAAS, SAAS.

ЗМІСТ

ВСТУП……... 8
1. ТЕОРЕТИЧНІ ОСНОВИ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ВЕБ-САЙТІВ ТА АДМІНІСТРУВАННЯ СЕРВЕРІВ……..... 11
1.1 Поняття мережевої інфраструктури веб-сайту та її складові…… 11
1.2 Основні принципи та моделі системи адміністрування серверів…….. 16
1.3 Рівні підтримки та організація серверного обслуговування……..... 19
2. СУЧАСНІ ТЕХНОЛОГІЇ ТА РІШЕННЯ У МЕРЕЖЕВІЙ ІНФРАСТРУКТУРІ ВЕБ-САЙТІВ..... 25
2.1 Серверні технології та платформи для розгортання веб-сайтів……..... 25
2.2 Мережеві протоколи, сервіси та засоби взаємодії……. 31
2.3 Віртуалізація та хмарні технології у веб-інфраструктурі……..... 35
2.4 Критерії вибору серверного обладнання та програмного забезпечення.. 44
3. ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ СИСТЕМИ АДМІНІСТРУВАННЯ СЕРВЕРІВ У МЕРЕЖЕВІЙ ІНФРАСТРУКТУРІ ВЕБ-САЙТІВ…..... 49
3.1 Принципи розгортання серверного середовища веб-сайту……..... 49
3.2 Основні етапи розробки та впровадження системи адміністрування..... 55
3.3. Моніторинг, безпека та підтримка серверів і мережі..... 59
ВИСНОВКИ…… 64
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ…….. 66

ВСТУП

У сучасних умовах стрімкої цифровізації суспільства та розвитку інформаційних технологій особливої значущості набуває ефективне адміністрування серверів і управління мережевою інфраструктурою веб-сайтів. Переважна більшість бізнес-процесів, державних сервісів і комунікаційних платформ функціонують на основі веб-технологій, що зумовлює підвищені вимоги до стабільності, безпеки та продуктивності серверних систем. Зростання кількості веб-додатків і обсягів оброблюваних даних супроводжується підвищенням навантаження на серверну інфраструктуру, що потребує впровадження сучасних підходів до адміністрування, моніторингу та оптимізації ресурсів. Крім того, актуальність теми обумовлюється постійним зростанням кіберзагроз, зокрема атак типу DDoS attack, витоків даних та несанкціонованого доступу, що вимагає впровадження ефективних механізмів захисту та контролю. Важливим чинником є також поширення хмарних технологій і моделей надання IT-послуг, таких як IaaS, PaaS та SaaS, які змінюють підходи до управління інфраструктурою та вимагають нових компетенцій від фахівців. Саме тому, виходячи із викладеного вище, дослідження процесів адміністрування серверів і управління мережевою інфраструктурою веб-сайтів є *актуальним* завданням, оскільки воно спрямоване на підвищення надійності функціонування інформаційних систем, забезпечення їх безпеки та ефективності в умовах зростаючих технологічних викликів.

Мета роботи – вивчення системи адміністрування серверів та управління мережевою інфраструктурою веб-сайтів. Відповідно до мети сформульовано такі *завдання дослідження*: вивчення теоретичних основ мережевої інфраструктури веб-сайтів та адміністрування серверів; розгляд серверних технологій та платформ для розгортання веб-сайтів; виокремлення мережевих протоколів, сервісів та засобів взаємодії; аналіз особливостей

віртуалізації та хмарних технологій у веб-інфраструктурі; визначення критеріїв вибору серверного обладнання та програмного забезпечення; відображення принципів розгортання серверного середовища веб-сайту; обґрунтування основних етапів розробки та впровадження системи адміністрування; дослідження моніторингу, безпеки та підтримки серверів і мережі. *Об'єкт дослідження* – функціональні процеси та процеси розвитку мережевої інфраструктури веб-сайтів у контексті забезпечення їх стабільної, безпечної та ефективної роботи. *Предмет дослідження* – методи, технології та інструменти системи адміністрування серверів та управління мережевою інфраструктурою веб-сайтів, включаючи інструменти моніторингу, безпеки та оптимізації ресурсів. *Методи дослідження*. У роботі використано комплекс загальнонаукових і спеціальних методів, зокрема: метод аналізу та узагальнення для дослідження теоретичних основ мережевої інфраструктури; порівняльний метод для оцінки сучасних серверних технологій та рішень; системний підхід для вивчення структури мережевої інфраструктури; метод моделювання для відображення процесів адміністрування серверів; методи спостереження та опису для характеристики практичних аспектів їх впровадження. *Наукова новизна і практична значущість отриманих результатів* полягають у комплексному підході до дослідження процесів адміністрування та управління серверами мережевої інфраструктури веб-сайтів з урахуванням сучасних технологій, особливо віртуалізації та хмарних рішень. У роботі узагальнено підходи до побудови ефективної серверної інфраструктури та визначено ключові фактори, що впливають на її продуктивність та безпеку.

Апробація результатів та публікації.

Теоретична, методична та практична значущість отриманих результатів полягає в систематизації теоретичних знань про мережеву інфраструктуру та систему адміністрування серверів, удосконаленні методологічних підходів до їх дослідження та можливості застосування

результатів у практичній роботі ІТ-фахівців з розгортання, налаштування та підтримки серверних систем та веб-ресурсів.

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1. ТЕОРЕТИЧНІ ОСНОВИ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ВЕБ-САЙТІВ ТА АДМІНІСТРУВАННЯ СЕРВЕРІВ

1.1 Поняття мережевої інфраструктури веб-сайту та її складові

Мережна інфраструктура є критичною основою для цифрового світу, складаючись з комбінації апаратного забезпечення, програмного забезпечення та послуг, необхідних для безперебійного функціонування мережевих комунікацій. Вона відіграє ключову роль у забезпеченні підключення, зв'язку, операцій та управління в рамках мережі, служачи основою, що підтримує всі мережеві функції та послуги. Сьогодні сучасна мережева інфраструктура розроблена для ефективної передачі даних, безпечних комунікацій та надійного підключення між різними пристроями і платформами [29].

Взагалі поняття «корпоративна мережа» (частіше – комп’ютерна мережа) набуло широкого поширення в межах країни наприкінці 1990-х років [15]. На той час мережеві технології були досить слабкі й примітивні. Підключення до мережі здебільшого здійснювалося через модеми зі швидкістю до 56 Кбіт/с, або через ADSL-з’єднання, яке давало максимум 10–20 Мбіт/с. Через невисоку швидкість передавання даних можливості зловмисників щодо поширення шкідливого програмного забезпечення та викрадення інформації були суттєво обмежені.

У результаті вся мережева інфраструктура фактично зводилася до простого набору кінцевих комп’ютерів та одного чи кількох серверів у межах локальної мережі (як на рис. 1.1). Тобто це була доволі закрита система, без складних інтеграцій чи широких каналів зв’язку.

Сьогодні спостерігається така ситуація, коли швидкість обміну даними як у межах локальних сегментів мережі, так і між локальною мережею та зовнішніми вебсерверами або віддаленими сегментами практично зрівнялася. Це стало наслідком розвитку мережевих технологій, що проявився у зростанні

пропускної здатності каналів мережі, а також удосконалення технологічної бази, зокрема збільшення обчислювальних потужностей мережевих та кінцевих вузлів. У результаті значно розширилися можливості приймання та обробки відносно великих обсягів даних в одиницю часу.

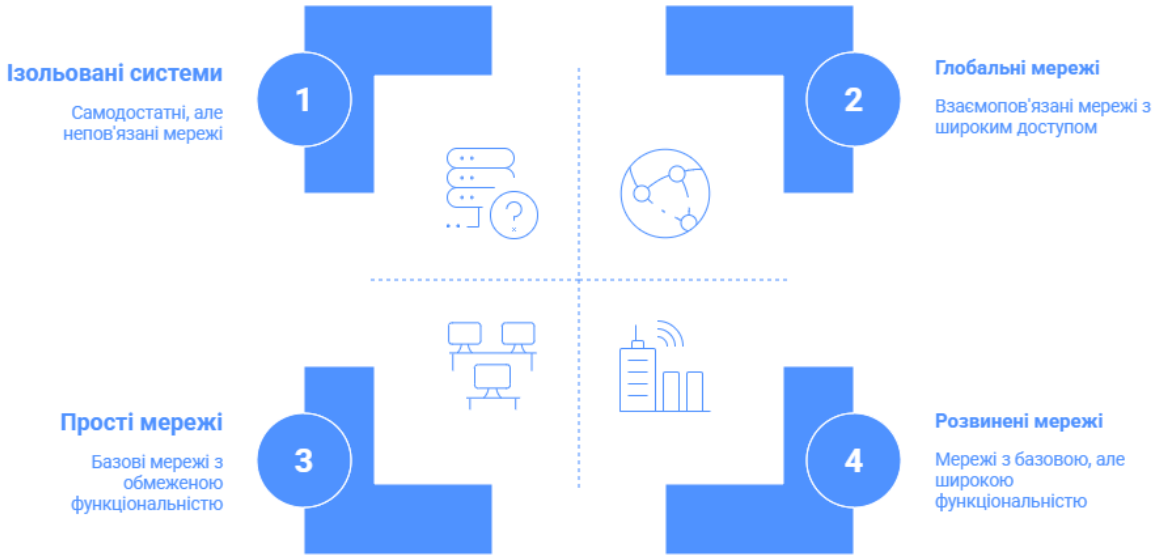


Рис. 1.1 – Складові мережевої інфраструктури кінця 90-х років XX століття

На сучасному етапі розвитку інформаційних технологій існує значна кількість комп'ютерних мереж, які відрізняються за своїм призначенням і базуються на різноманітних обчислювальних та телекомунікаційних технологіях. Їх побудова та принципи роботи залежать від обраної мережевої архітектури.

Мережева архітектура являє собою комплекс апаратних і програмних рішень, що визначають способи організації доступу до мережі та регламентують протоколи обміну інформацією між її компонентами [23]. Мережі надають користувачам можливість не тільки швидкого обміну інформацією, але і спільної роботи на принтерах та інших периферійних пристроях, і навіть одночасної обробки документів.

Комп'ютерна мережа (КМ) – це сукупність засобів в'язку, яка забезпечує обмін даними між двома чи більше комп'ютерами [42]. Основною її

характеристикою є швидкість передавання даних, яка вимірюється у бітах за секунду (б/с). Усе різноманіття комп’ютерних мереж можна класифікувати за групою ознак:

територіальна поширеність;

відомча належність;

швидкість передачі інформації;

тип середовища передачі інформації [36].

Для створення єдиної інформаційної системи підприємства, її коректної роботи та налагодженого процесу обміну інформації потрібна мережна інфраструктура, яка створить для цього необхідні умови.

Мережева інфраструктура – це комплекс технічних засобів і програмних продуктів, який дозволяє використовувати різні прикладні мережеві послуги для ефективної роботи [6]. Складові мережної інфраструктури відображено на рис. 1.2.



Рис. 1.2 – Складові мережної інфраструктури

Локальна мережа – основний компонент мережевої інфраструктури, який об’єднує всі комп’ютери підприємства та обчислювальні ресурси та забезпечує обмін даних між ними. Дозволяє використовувати спільні

обчислювальні ресурси всіх технічних засобів компанії на вирішення складних завдань.

Система передачі даних (СПД) – сукупність технічних засобів для надійної передачі та отримання інформації; забезпечує взаємодію системних додатків та дозволяє організовувати спільний доступ до інформаційних ресурсів компанії.

Бездротові мережі (Wi-Fi) – це комп'ютерні мережі, створені без використання проводів. Легші в організації з фізичного боку. Забезпечують ефективну взаємодію та обмін інформацією [18].

Окрім цього, основні складові включають веб-сервер, доменне ім'я (DNS), хостинг, базу даних, SSL-сертифікати та систему керування вмістом (CMS).

Веб-сервер – це як цифрова кімната, де зберігаються всі файли веб-сайта, такі як HTML, CSS, JavaScript та зображення. Якщо хтось хоче побачити веб- сторінку, сервер відправляє ці файли на свій комп' ютер [7].

Хостинг – це служба круглого часу для отримування веб-сайту на мережевому сервері.

Доменна назва, наприклад google.com, - це унікальна адреса в інтернеті, яка перекладається на числову IP- адресу системою DNS.

База даних – добре організоване місце, де можна знайти будь-яку інформацію на зразок текстів, подробиць користувачів та коментарів, що допомагає динамічним веб-сайтам працювати плавно [47].

Цифрові сертифікати – це цифрові файли, які забезпечують можливість ідентифікації (організації, особи, служби або програмного коду) та автентичності відкритих ключів шифрування та цифрового підпису. Сертифікат типу SSL/TLS зберігає дані у безпеці під час розмови з веб- сайтом, перевіряючи, чи все безпечно (це те, що означає HTTPS) [43].

CMS – це програмне забезпечення, що призначене для конструювання і керування веб-сайтами [17].

Веб-браузери – це програми, призначені для перегляду та сприйняття гіпертекстових документів [38].

Загалом мережна інфраструктура забезпечує швидкість завантаження, стабільність роботи та захищеність будь-якого веб-ресурсу. Мережна інфраструктура поділяється на кілька основних компонентів, кожен з яких виконує певну функцію (рис. 1.3).

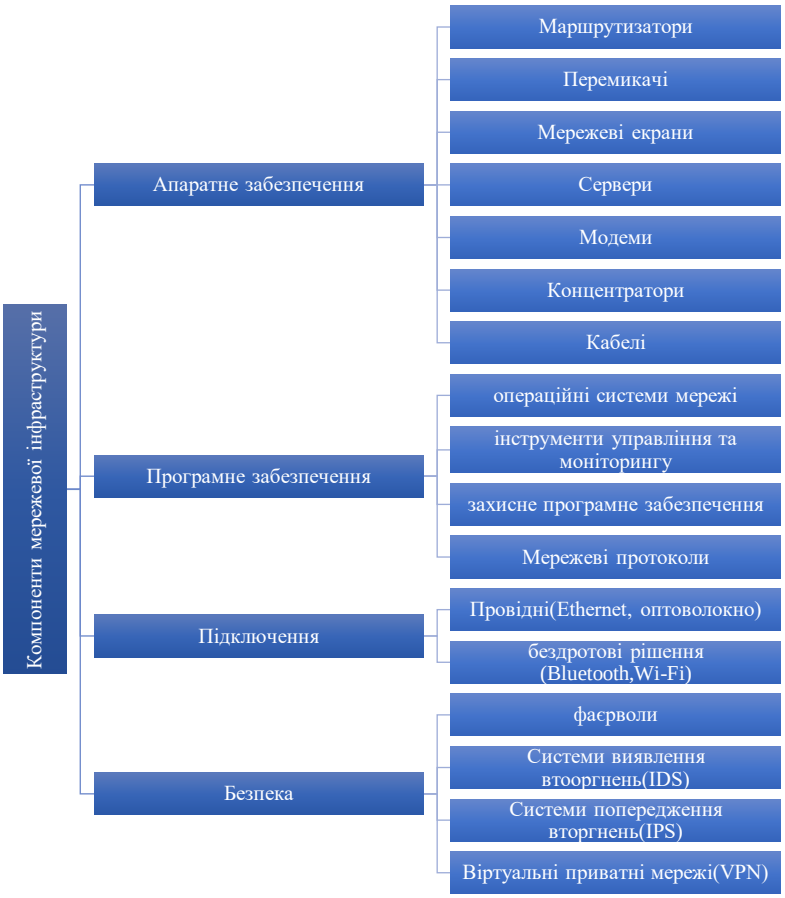


Рис. 1.3 – Компоненти мережевої інфраструктури

Як бачимо, на рисунку представлено узагальнену структуру компонентів мережевої інфраструктури, яка поділяється на чотири основні складові: арпаратне забезпечення, програмне забезпечення, підключення та безпека. Такий поділ є логічним і дозволяє комплексно відобразити всі елементи, необхідні для функціонування сучасної мережі.

Отже, мережева інфраструктура – це складна система, що складається з численних компонентів, що поєднують апаратне забезпечення, програмне забезпечення, канали зв’язку та функції безпеки для забезпечення

ефективного, стабільного та безпечного мережевого зв'язку. Її еволюція від простих локальних мереж з обмеженою ємністю до сучасних, інтегрованих, високошвидкісних систем є результатом швидкого розвитку технологій, обчислювальної потужності та вимог до передачі даних. Сьогодні мережева інфраструктура формує основу для роботи веб-ресурсів та інформаційних систем, забезпечуючи безперебійну взаємодію між пристроями, доступ до спільних ресурсів та належний рівень захисту інформації.

1.2 Основні принципи та моделі системи адміністрування серверів

Система адміністрування серверів є ключовим елементом функціонування сучасних інформаційних систем і мережевої інфраструктури. У процесі розвитку цифрових технологій зростає потреба у надійному, безпечному та ефективному управлінні серверними ресурсами, що забезпечують безперебійну роботу веб-сайтів, баз даних і корпоративних сервісів. Основні принципи адміністрування серверів ґрунтуються на забезпеченні стабільності, масштабованості, безпеки та доступності систем, а також на оптимізації їх продуктивності. Важливу роль відіграють різні моделі адміністрування, зокрема централізована, децентралізована та гібридна, які визначають підходи до управління ресурсами та розподілу відповідальності між адміністраторами. Розуміння цих принципів і моделей є необхідною передумовою для побудови ефективної серверної інфраструктури та забезпечення її надійного функціонування в умовах сучасних викликів.

Принципи адміністрування серверів – це комплексний підхід, що базується на забезпеченні безпеки, стабільності, продуктивності та автоматизації процесів. Ключові принципи адміністрування серверів включають регулярне оновлення ПЗ, налаштування моніторингу, резервне копіювання, керування правами доступу та швидке реагування на інциденти [49].

Основні принципи адміністрування зосереджені навколо безпеки, моніторингу та продуктивності, резервного копіювання, автоматизації, керування доступом, документування інформації та даних (табл. 1.1).

Таблиця 1.1 – Основні принципи адміністрування серверів

Принцип	Опис
Безпека (Security)	Пріоритет №1. Включає захист від злому, регулярне оновлення операційної системи та програмного забезпечення, використання безпечних протоколів (SSH, HTTPS), налаштування фаєрволів і антивірусів.
Моніторинг та продуктивність	Постійний контроль навантаження на CPU, RAM, мережу та диск (Disk I/O), що дозволяє своєчасно виявляти й усувати проблемні місця.
Резервне копіювання (Backup)	Регулярне створення резервних копій даних і перевірка можливості їх відновлення для гарантії збереження інформації.
Автоматизація (Automation)	Використання скриптів та інструментів (DevOps, CI/CD) для автоматизації рутинних процесів і зменшення людського фактору.
Керування доступом	Розмежування прав користувачів, застосування принципу мінімальних привілеїв (без зайвого використання root/administrator).
Документування	Ведення логів, фіксація конфігурацій і процедур відновлення для прозорості роботи та полегшення діагностики.

Як бачимо із даних табл. 1.1, адміністрування серверів забезпечує надійну роботу веб-серверів, баз даних, VPS і локальних мереж із використанням операційних систем Linux та Windows Server тощо [1-2; 20; 34].

Моделі адміністрування серверів визначають рівень відповідальності між клієнтом та провайдером (або системним адміністратором) за обслуговування інфраструктури: від повного контролю (Dedicated) до надання готових рішень (SaaS). Основні моделі – IaaS (інфраструктура як послуга), PaaS (платформа) та SaaS (програмне забезпечення), що включають налаштування ОС, безпеку, моніторинг та масштабування.

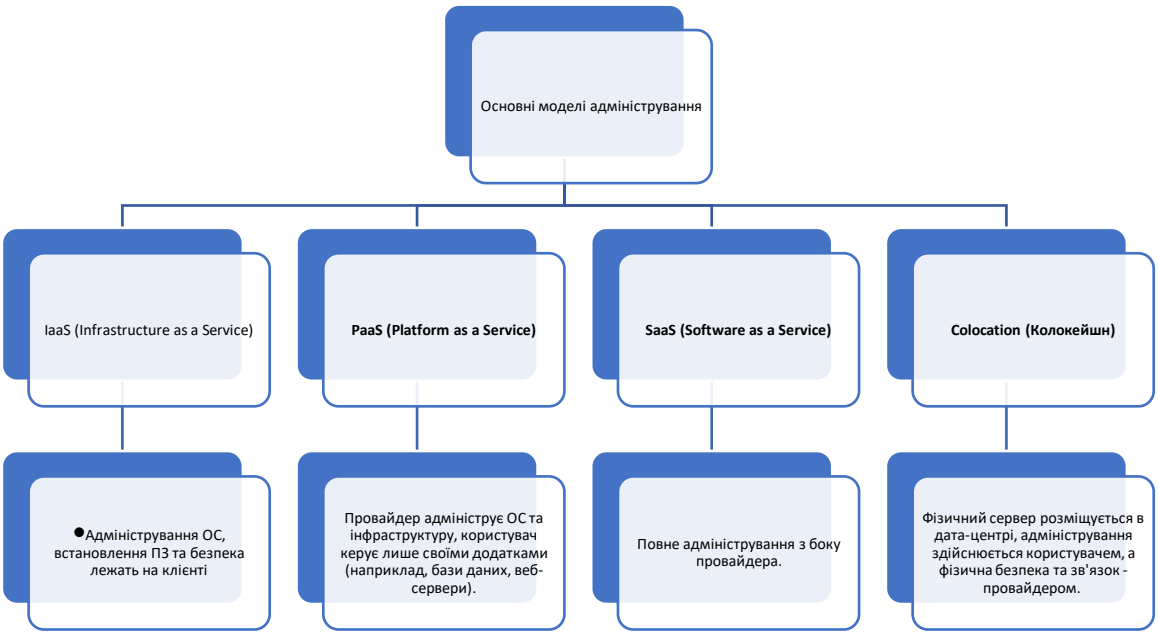


Рис. 1.4 – Основні моделі адміністрування серверів [3; 44]

Підходи до системи адміністрування серверної інфраструктури можна класифікувати за рівнем участі адміністратора та ступенем автоматизації процесів.

До першої категорії належить віддалене адміністрування, яке передбачає керування сервером за допомогою захищених мережевих протоколів, зокрема таких як SSH, RDP та VPN. Такий підхід забезпечує можливість контролю та налаштування системи незалежно від фізичного місцезнаходження адміністратора.

Іншим важливим напрямом є автоматизоване адміністрування, що реалізується в межах підходів DevOps. Воно базується на використанні спеціалізованих інструментів конфігураційного управління, таких як Ansible, Puppet та Chef, які дозволяють описувати інфраструктуру у вигляді коду та забезпечують її відтворюваність, масштабованість і стандартизацію.

Окрему модель становить використання керованих серверів, за якої значна частина або повний обсяг адміністративних функцій передається провайдеру послуг. У цьому випадку постачальник забезпечує моніторинг, оновлення, безпеку та технічну підтримку інфраструктури, що дозволяє

знизити навантаження на внутрішні ресурси організації.

Вибір конкретної моделі системи адміністрування визначається рівнем технічної підготовки персоналу та стратегічними цілями підприємства. Зокрема, модель IaaS передбачає наявність кваліфікованих системних адміністраторів, тоді як PaaS і SaaS є більш доцільними для швидкого розгортання прикладних рішень із мінімальними вимогами до адміністрування.

1.3 Рівні підтримки та організація серверного обслуговування

У сучасних інформаційних системах угоди про рівень обслуговування (Service Level Agreement, SLA) є ключовим інструментом регламентації якості надання IT-послуг, визначаючи очікувані параметри функціонування серверної інфраструктури, включаючи доступність, продуктивність, час реагування на інциденти та відновлення сервісів.

SLA виступає формалізованою угодою між постачальником послуг і замовником, яка встановлює кількісні та якісні показники ефективності обслуговування. Особлива увага приділяється безперервності роботи сервісів, що є критично важливим для бізнес-процесів [19].

Рівні підтримки серверного обслуговування (SLA) визначають швидкість реакції та обсяг послуг: від базового моніторингу (NBD – наступний робочий день) до критичної підтримки 24/7 з гарантованим часом відновлення (часто 1-4 години). Вони включають абонентське обслуговування, оновлення, резервне копіювання та захист, адаптуючись під навантаження та важливість IT-інфраструктури.

До базових характеристик SLA належать:

- доступність сервісу (Availability) – визначається у відсотках часу безперервної роботи системи (наприклад, 99,9% або 99,99% uptime) [6].
- час реагування (Response Time) – максимальний допустимий

інтервал між повідомленням про інцидент і початком його обробки [31].

- час відновлення (Recovery Time Objective, RTO) – час, необхідний для повного відновлення працездатності сервісу [48].
- допустима втрата даних (Recovery Point Objective, RPO) – максимальний період втрати даних у разі збою [10].
- пропускна здатність і продуктивність – показники швидкодії системи під навантаженням [28].

Рівні підтримки зазвичай структуруються відповідно до складності інцидентів та ступеня залучення спеціалістів (табл. 1.2).

Таблиця 1.2 – Рівні технічної підтримки серверів [26]

Рівень	**Характеристика**	**Основні функції**
L1 (базовий рівень)	Початковий рівень підтримки	Прийом звернень, первинна діагностика, вирішення типових проблем
L2 (середній рівень)	Поглиблена технічна підтримка	Аналіз складніших інцидентів, налаштування сервера, робота з конфігураціями
L3 (експертний рівень)	Висококваліфікована підтримка	Усунення критичних збоїв, оптимізація системи, розробка рішень
L4 (вендор/розробник)	Підтримка виробника	Виправлення помилок програмного забезпечення, оновлення продуктів

Залежно від вимог до безперервності бізнес-процесів, SLA поділяються на декілька категорій (табл. 1.3).

Таблиця 1.3 – Типові рівні SLA [50]

Рівень SLA	**Доступність**	**Час реагування**	**Сфера застосування**
Базовий	95–99%	До 24 годин	Некритичні системи
Стандартний	99–99,9%	4–8 годин	Корпоративні системи
Розширений	99,9–99,99%	1–2 години	Бізнес-критичні сервіси
Критичний (Premium)	≥99,99%	До 15 хвилин	Високонавантажені та фінансові системи

Згідно з аналізованими підходами, ефективність SLA залежить від:

- впровадження систем моніторингу (continuous monitoring);
- автоматизації процесів реагування на інциденти;
- використання резервування та відмовостійких архітектур;

- чіткого визначення відповідальності сторін.

Особливу роль відіграє інтеграція SLA з практиками управління IT-послугами, такими як ITIL, що забезпечує стандартизацію процесів підтримки.

Таким чином, SLA є критичним елементом управління серверною інфраструктурою, який забезпечує прозорість, передбачуваність і контроль якості IT-послуг. Рівні підтримки дозволяють ефективно розподіляти ресурси та забезпечувати належний рівень обслуговування відповідно до вимог бізнесу. Раціональний вибір параметрів SLA сприяє підвищенню надійності систем та мінімізації ризиків простою [50].

Вибір рівня підтримки серверного обслуговування залежить від критичності сервера для бізнесу – чим вища вартість простою, тим вищий рівень підтримки потрібен [9].

Окрім підтримки серверного обслуговування не менш важливим є і технічне обслуговування серверів, яке являє собою комплекс необхідних дій, який забезпечує безперебійну роботу сервера.

Технічне обслуговування серверної інфраструктури включає комплекс робіт, спрямованих на забезпечення стабільної, безпечної та безперервної роботи інформаційних систем. До основних завдань обслуговування серверної інфраструктури належать:

- інсталяція та налаштування програмного забезпечення;
- організація та контроль процесів резервного копіювання даних;
- встановлення і конфігурація антивірусних засобів захисту;
- налаштування мережевих протоколів;
- встановлення та адміністрування операційних систем;
- обслуговування поштових серверів;
- проведення моніторингу та профілактичних заходів;
- тестування й оновлення систем інформаційної безпеки;
- діагностика та усунення технічних несправностей.

Регулярне виконання зазначених завдань дозволяє суттєво знизити

ризик виникнення збоїв у роботі операційних систем, а також мінімізувати загрози, пов’язані з впливом шкідливого програмного забезпечення.

Апаратна підтримка охоплює сукупність робіт, пов’язаних із підбором, встановленням, заміною та модернізацією серверного обладнання. До цієї категорії серверного обслуговування також належать:

- організація серверного приміщення відповідно до технічних вимог;
- забезпечення безперебійного електроживлення;
- оптимізація апаратної конфігурації відповідно до навантаження.

Підготовка сервера до роботи є складним процесом, що потребує професійного підходу. Перелік дій з підготовки сервера залежить від його типу та функціонального призначення, проте цей процес зазвичай включає такі етапи:

- підключення до електромережі;
- встановлення операційної системи та прикладного програмного забезпечення;
- налаштування доступу користувачів;
- організацію системи резервного копіювання.

Вибір серверної платформи визначається специфікою задач. Наприклад, сервери на базі Windows Server доцільно використовувати для роботи з прикладними програмами корпоративного рівня (зокрема, бухгалтерськими системами), що забезпечує централізоване зберігання даних і можливість віддаленого доступу. Водночас системи на базі Linux широко застосовуються як веб-сервери завдяки високій стабільності, гнучкості налаштування та підвищеному рівню безпеки. Не менш важливою є продуктивніть платформ. Відповідно до цього вимоги до продуктивності серверних платформ комп’ютерної мережі формулюються на основі стандартних методик вимірювання продуктивності [39].

Супровід серверної інфраструктури передбачає систематичний моніторинг її стану та виконання профілактичних заходів, навіть за

відсутності явних збоїв. До ключових процедур, які застосовуються на цьому етапі належать:

- регулярне оновлення програмного забезпечення (рекомендовано не рідше одного разу на місяць);
- планові перезавантаження системи для оптимізації використання ресурсів;
- антивірусна перевірка з метою своєчасного виявлення загроз;
- оптимізація файлової системи (зокрема дефрагментація носіїв даних);
- аналіз системних журналів і результатів моніторингу для виявлення потенційних проблем.

Віддалене управління – це основний спосіб конфігурувати різне мережеве обладнання, апаратні і мультимедійні платформи. Як правило, такі пристрої володіють web-інтерфейсом адміністрування. Віддалене адміністрування є ефективним інструментом управління серверною інфраструктурою, що дозволяє оперативно реагувати на інциденти без фізичної присутності спеціаліста. Такий підхід забезпечує скорочення часу реагування та зниження експлуатаційних витрат. Водночас ключовою умовою його реалізації є наявність стабільного та високошвидкісного мережевого з’єднання [27].

У процесі експлуатації серверів можуть виникати різноманітні проблеми, зумовлені як програмними, так і апаратними факторами. Найпоширенішими з них є:

- помилки конфігурації системи;
- відмова або зношення апаратних компонентів;
- мережеві атаки типу DDoS attack;
- фізичні перебої в електроживленні або мережевому підключенні.

Тому зважаючи на викладене вище, ефективне виявлення та усунення таких проблем можливе лише за умови належного рівня адміністрування та

регулярного технічного обслуговування.

Таким чином, комплексне технічне обслуговування серверів є необхідною умовою забезпечення надійності, безпеки та безперервності функціонування інформаційних систем. Поєднання програмної та апаратної підтримки, а також використання сучасних методів адміністрування дозволяє мінімізувати ризики збоїв і підвищити ефективність роботи IT-інфраструктури.

2. СУЧАСНІ ТЕХНОЛОГІЇ ТА РІШЕННЯ У МЕРЕЖЕВІЙ ІНФРАСТРУКТУРІ ВЕБ-САЙТІВ

2.1 Серверні технології та платформи для розгортання веб-сайтів

Функціонування мережі Інтернет ґрунтується на клієнт-серверній архітектурі. У межах цієї моделі клієнтська програма, зазвичай веб-браузер, формує запити для отримання інформації або мережевих ресурсів і надсилає їх за URL-адресою. Обробка таких запитів здійснюється серверним програмним забезпеченням, що працює на серверних комп'ютерах, підключених до мережі та ідентифікованих IP-адресою. У відповідь сервер надає необхідні файли, контент та інші ресурси, а також виконує допоміжні сервісні функції (рис. 2.1) [7].

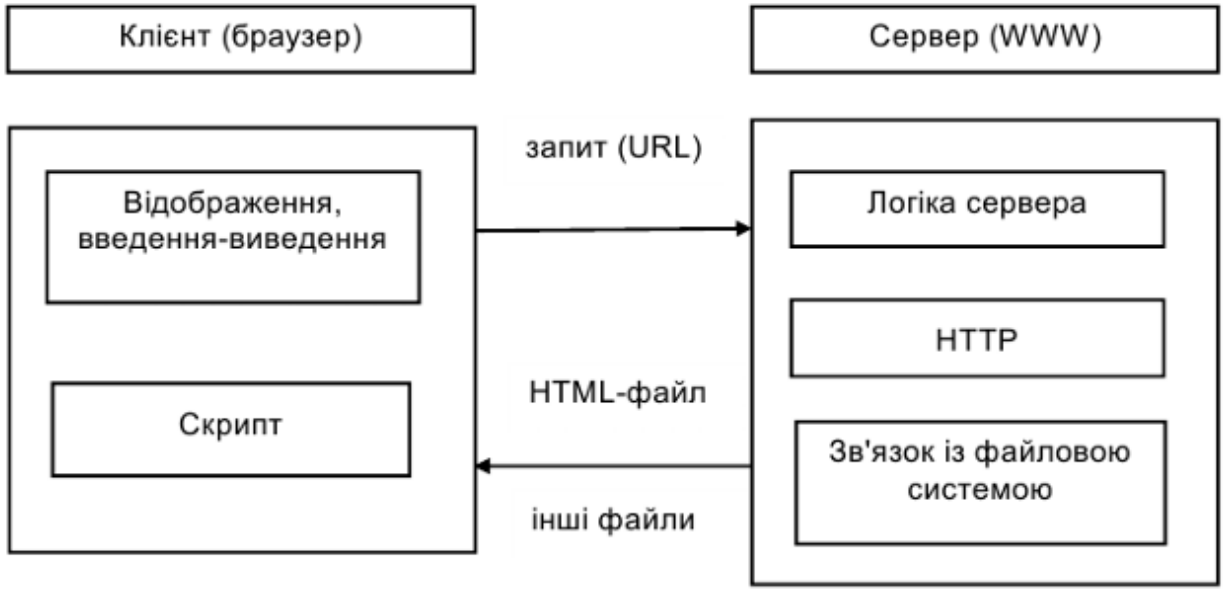


Рис. 2.1 – Клієнт-серверна архітектура у WEB

Клієнт-сервісна архітектура (client-server architecture) – є підходом до організації інформаційних мереж, у якому основні обчислювальні та інформаційні ресурси зосереджуються на серверній стороні. Сервери забезпечують надання послуг і ресурсів іншим учасникам мережі –клієнтам,

які формують відповідні запити. У межах такої моделі виділяють два основних типи компонентів: серверні системи та клієнтські пристрої.[5].

Сервером називають програмно-апаратний вузол, що виконує функції обробки запитів і надання сервісів іншим елементам мережі. Сервіс у цьому контексті розглядається як сукупність процедур і функцій, спрямованих на обслуговування запитів клієнтів та забезпечення доступу до необхідних ресурсів. [14].

Сервер працює за завданнями клієнтів і управляє виконанням їх завдань. Після виконання кожного завдання сервер посилає отримані результати клієнту, який послав це завдання.

Сервісна функція в архітектурі клієнт – сервер описується комплексом прикладних програм, відповідно до якого виконуються різноманітні прикладні процеси [15].

Процес, який викликає сервісну функцію за допомогою певних операцій, називається клієнтом. Ним може бути програма або користувач.

Клієнти – це робочі станції, які використовують ресурси сервера і надають зручні інтерфейси користувача. Інтерфейси користувача це процедури взаємодії користувача з системою або мережею [11].

Клієнт є ініціатором і використовує електронну пошту або інші сервіси сервера. У цьому процесі клієнт запитує вид обслуговування, встановлює сеанс, отримує потрібні йому результати і повідомляє про закінчення роботи.

У мережах з виділеним файловим сервером на виділеному автономному ПК встановлюється серверна мережева операційна система. Цей ПК стає сервером. Програмне забезпечення (ПЗ), встановлене на робочій станції, дозволяє їй обмінюватися даними з сервером. Найбільш поширені мережеві операційна системи:

– NetWare фірми Novel;

– Windows NT фірми Microsoft;

– UNIX фірми AT &T;

– Linux [15].

Крім мережевої операційної системи необхідні мережні прикладні програми, що реалізують переваги, надані мережею. Мережі на базі серверів мають кращі характеристики і підвищену надійність. Сервер володіє головними ресурсами мережі, до яких звертаються інші робочі станції.

У сучасній клієнт – серверній архітектурі виділяється чотири групи об’єктів: клієнти, сервери, дані і мережеві служби (рис. 2.2).

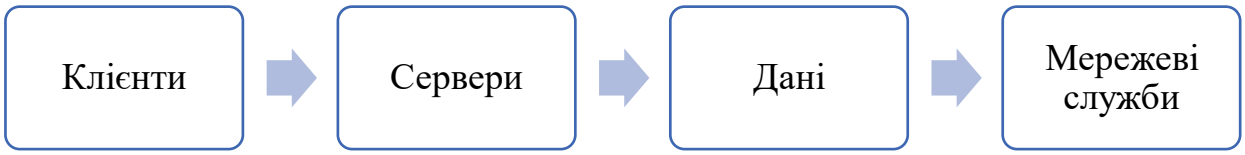


Рис.2.2 – структура сучасної клієнт-серверної архітектури

Клієнти розташовуються в системах на робочих місцях користувачів. Дані в основному зберігаються в серверах. Мережеві служби є спільно використовуваними серверами і даними. Крім того служби керують процедурами обробки даних [11].

За такої взаємодії сервер грає роль сховища. Для розширення можливостей WEB-сторінок із реалізації більш складних функцій необхідно розширювати логіку сервера за рахунок виконання спеціально розроблених програм. Ці програми можуть виконувати найрізноманітніші функції: оброблення сторінок перед відправкою, зв'язок із базами даних, проведення обчислень тощо. Таким чином під серверними технологіями будемо розуміти сукупність засобів і методів запуску виконуваних програм на сервері.

Для забезпечення функціонування сервісу WWW було розроблено досить багато серверів. Вони володіють різними характеристиками, але найбільш масовими є два: IIS (Internet Information Services), розроблений компанією "Майкрософт", і вільно поширювані Apache та Nginx.

У практиці розгортання веб-сайтів широко використовуються готові технологічні стеки, зокрема LAMP (Linux, Apache, MySQL, PHP) та LEMP (Linux, Nginx, MySQL, PHP), які об’єднують основні компоненти серверного

середовища в єдину узгоджену систему. Адже використання таких стеків забезпечує сумісність програмних компонентів, спрощує адміністрування та підвищує ефективність функціонування веб-застосунків.

Програми, які розробляються для виконання на сервері крім реалізації основних функцій, повинні мати кілька особливостей (порівняно з тими, що виконуються в середовищі операційної системи):

- запуск здійснюється сервером;
- має бути забезпечена можливість взаємодії із сервером;
- має бути забезпечена можливість відправляти результати (дані) клієнту, який видав запит (браузеру).

Для створення серверних розширень було розроблено кілька інтерфейсів (специфікацій на взаємодію між сервером та програмою), найбільшого поширення набули: CGI, FCGI і API-інтерфейси. Кожен із них має свої переваги, так само, як і певні недоліки [5].

Технологія активних серверних сторінок (включення коду в саму HTML сторінку) полягає в тому, що запитувана сторінка з розміченим текстом містить програмний код, який виконується на сервері перед відправкою клієнту, змінюючи запитувану сторінку. Цей код може витягувати дані з бази даних, будувати призначений для користувача інтерфейс залежно від отриманих даних, а потім відправляти HTML-сторінку назад клієнту.

Кінцевим результатом такого процесу буде відправка браузеру стандартної WEB сторінки на HTML. Цей процес являє собою динамічне створення WEB сторінок. Сторінка, яка показується в браузері, насправді є продуктом виконання коду, який був запущений після клієнтського запиту, на сервері такої сторінки фізично не існує [7].

Таким чином, клієнт-серверна архітектура та серверні технології формують основу функціональності сучасних веб-ресурсів. Вони дозволяють обробляти запити користувачів, взаємодіяти з базами даних та генерувати динамічний контент. Однак практична реалізація цих функцій вимагає складних програмних рішень, що спрощують розробку, розгортання та

адміністрування веб-сайтів. Саме такими інструментами виступають системи керування вмістом, платформи хостингу та технології віртуалізації, що дозволяють ефективно організувати серверне середовище та забезпечити стабільну роботу веб-застосунків.

Для організації сучасного веб-сайту та його швидкого розгортання одним з оптимальних рішень є залучення системи керування вмістом (CMS, Content Management System) з відкритим кодом WordPress. Завдяки такому вибору можна у найкоротший термін розробити універсальний веб-сайт, блог, Інтернет-магазин тощо. Завдяки наявності готових шаблонів тем, плагінів, у поєднанні з можливістю їх швидкої розробки, веб-сайт на WordPress можна налаштувати для рішення різноманітних завдань [4].

Зараз для розгортання веб-сайтів доступно чимало сервісів хостингу та ресурси хмарних обчислень (Cloud Computing). Хмарні обчислення можуть надаватися в вигляді: сервісів, технологій, програм, платформ і інфраструктури. Поруч із цим є можливість застосування виділеного серверу [37]. Таке рішення надає можливості одноразового інвестування коштів у обладнання та подальшу експлуатацію веб-серверу з оплатою тільки послуг центру обробки даних (сервісне обслуговування, енергоресурси та Інтернет-трафік) [4].

Для підвищення продуктивності та доступності веб-ресурсів у сучасних умовах широко застосовується балансування навантаження (load balancing), яке дозволяє розподіляти запити користувачів між кількома серверами, що, як наслідок, сприяє підвищенню відмовостійкості системи та забезпечує стабільну роботу навіть при значних навантаженнях.

На платформі виділеного серверу (Dedicated Server), який є окремим обчислювальним пристроєм з окремим виділеним каналом зв’язку, доцільно розгорнути систему віртуалізації, що надає уніфікацію середовищу розгортання програмного забезпечення веб-серверу та ефективне рішення резервного копіювання (Backup) всіх даних на рівні файлу сховища віртуальної машини (VPS, Virtual Private Server) [7].

Іншим підходом до організації серверного середовища може стати рішення на базі технології віртуалізації Docker та, у разі масштабування такого рішення, кластеру Kubernetes. Однак, цей підхід більш себе виправдовує у разі застосування методології DevOps і неперервного циклу інтеграції та доставки веб-застосунків користувачам (CI/DI, Continuous Integration / Continuous Delivery), що фактично стає стандартним в процесах розробки веб-продукту, а не за умови впровадження готових рішень. Таким чином, можна отримати на стороні серверної платформи зручні засоби розгортання операційної системи у середовищі віртуальної машини та задіяти наявні засоби управління й резервного копіювання на рівні самого серверу [22].

Контейнеризація, зокрема із використанням Docker, дозволяє ізолювати програмні середовища разом із усіма необхідними залежностями, що забезпечує стабільність роботи додатків незалежно від платформи розгортання та спрощує процес їх перенесення між серверами.

Наступним кроком є розгортання системи WordPress. Для цього на стороні серверу (або віртуальної машини) знадобиться розгорнути веб-сервер, наприклад, Apache або Nginx чи ін., інтерпретатор мови PHP, базу даних MySQL або MariaDB. У разі обслуговування декількох сайтів знадобиться настройка веб-серверу (засобами Virtual Host) або установка WordPress у режимі мультисайту (Multisite, також технологія «Network of Sites») [46].

Значно спростити процедуру розгортання серверних компонентів дозволяє застосування спеціалізованої панелі управління, наприклад, відкритої системи Vesta Control Panel, яка надає зручний веб-інтерфейс для керування та моніторингу серверу. Така панель управління дозволяє керувати обліковими записами користувачів, доменними (Domains) іменами та відповідними DNS-записами, поштовими скриньками (Mail), виконувати автоматичне розгортання баз даних, планувати виконанням фонових процесів серверу (Cron Jobs), здійснювати резервне копіювання, надавати статистичні дані щодо роботи серверу та ін. У такому середовищі достатньо просто

розгорнути сучасний веб-сайт. Також, відповідні рішення масштабуються на рівень залучення ресурсів хостинг-провайдерів чи хмарних обчислень [4].

Окрім цього, для оптимізації швидкості завантаження веб-ресурсів застосовуються мережі доставки контенту (CDN), які забезпечують кешування статичних даних на географічно розподілених серверах та їх доставку користувачам з найближчих вузлів, значно зменшуючи затримки та підвищуючи загальну продуктивність веб-сайтів.

Отже, сучасні серверні технології та платформи забезпечують широкі можливості для розгортання веб-сайтів, поєднуючи традиційні підходи з інноваційними рішеннями, такими як віртуалізація, контейнеризація та хмарні обчислення, підвищуючи продуктивність, масштабованість і надійність веб-ресурсів, а також спрощуючи процес їх адміністрування та підтримки.

2.2 Мережеві протоколи, сервіси та засоби взаємодії

У контексті стрімкої цифрової трансформації інформаційно-комунікаційних систем протоколи TCP/IP залишаються базисом мережевої взаємодії, забезпечуючи обмін даними між вузлами в глобальних та локальних мережах [26].

У світі все підпорядковано правилам або законам, що визначають якісну роботу та функціональний взаємозв’язок елементів системи, в нашому випадку, локальної та глобальної мережі людства. Тому необхідно щоб при обміні даними між комп’ютерами мережі передбачалося, що дані без спотворення та втрати будуть доставлені від відправника адресату. Для цього необхідно, щоб різнотипні комп’ютери, комунікаційні пристрої, мережне обладнання та програмне забезпечення виконували передавання даних за однаковими чітко визначеними правилами. Такі правила називаються мережними протоколами.

Більшість сучасних комп’ютерних мереж здійснюють передавання даних на основі набору протоколів, що має назву TCP/IP (англ. Transmission Control Protocol / Internet Protocol – протокол управління передаванням / міжмережний протокол) [18].

Дані, що передаються мережею, розбиваються на невеликі пакети та доповнюються даними передачі: адресами комп'ютерів-відправників та одержувачів, кількістю та довжиною пакетів тощо. Кожен пакет передається окремо по каналу зв'язку. Шлях передачі визначається маршрутизаторами, які також контролюють доставку пакетів [18].

Правила розділення даних на пакети, їх доставки одержувачу та повторного складання в повний пакет визначаються протоколом TCP. Пересилання пакетів між комп'ютерами з різною архітектурою, операційними системами та мережами базується на протоколі IP.

Поділяючи дані на окремі пакети, їх передача по мережі є швидкою та надійною, навіть у разі збоїв мережевих компонентів. У разі збою мережі маршрутизатори намагаються знайти новий маршрут для пакета, щоб обійти пошкоджену ділянку.

Мережевий протокол у комп'ютерних мережах базується на наборі стандартів, що визначають принципи взаємодії комп'ютерів у мережі [8]. Протокол також встановлює загальні правила взаємодії різних програм, мережевих вузлів або систем, створюючи таким чином єдиний простір передачі. Хости (кожен мережевий вузол, який надсилає або отримує дані по мережі, називається хостом) взаємодіють один з одним. Щоб правильно отримувати та обробляти повідомлення, вони повинні розуміти їх структуру та значення. Приклади використання різних форматів повідомлень у різних протоколах включають встановлення з'єднання з віддаленим комп'ютером, надсилання повідомлень електронною поштою та передачу файлів. Очевидно, що різні сервіси використовують різні формати повідомлень.

Протокол описує:

1. Формат повідомлення, якому застосунки зобов'язані слідувати;

2. Спосіб обміну повідомленнями між комп'ютерами в контексті визначеної дії, як, наприклад, пересилка повідомлення по мережі [18].

Маршрутизований протокол – це будь-який мережний протокол, адреса мережевого рівня якого надає достатньо інформації для доставки пакету від одного вузла мережі до іншого на основі використовуваної схеми адресації. Такий протокол задає формати полів всередині пакету. Пакети зазвичай передаються від однієї кінцевої системи до іншої. Маршрутизований протокол використовує таблицю маршрутизації для пересилки пакетів [30].

Прикладами протоколів маршрутизації є Інтернет-протокол (IP), Інтернет-протокол (IPX) тощо. Принцип протоколів маршрутизації найлегше зрозуміти, якщо пам'ятати, що вони є протоколами передачі даних [35].

Протокол маршрутизації – це протокол, який підтримує маршрутизовані протоколи та забезпечує механізми обміну маршрутною інформацією [12]. Повідомлення протоколу маршрутизації обмінюються між маршрутизаторами. Протокол маршрутизації дозволяє маршрутизаторам обмінюватися інформацією один з одним для оновлення записів та підтримки таблиці маршрутизації. Прикладами протоколів маршрутизації є RIP, IGRP, EIGRP та OSPF.

Щоб протокол маршрутизувався, він повинен містити механізми для призначення мережевої адреси та адреси хоста кожному пристрою в мережі. Деякі протоколи, такі як IPX, вимагають лише вказівки мережевої адреси, оскільки ця технологія використовує фізичну адресу пристрою (MAC-адресу) як адресу пристрою. Інші протоколи, такі як IP-протокол, вимагають явного вказівки повної адреси та маски підмережі.

Найпоширенішою системою класифікації мережевих протоколів (і методів мережевого зв'язку загалом) є так звана модель OSI (таблиця 1.1), згідно з якою протоколи поділяються на 7 рівнів за їх призначенням – від фізичного рівня (генерація та виявлення електричних чи інших сигналів) до прикладного рівня (API додатків для передачі інформації) [40].

Таблиця 2.1 – Список мережевих протоколів

Тип даних	№ рівня	Рівень OSI	Опис	Протоколи
Дані	7	Прикладний	доступ до мережевих служб	NTP, Gopher, Finger, NNTP, LDAP, DHCP, IRC, Jabber, WebDAV, DICT, BACnet, HTTPS та ін.
Дані	6	Представлення	представлення і кодування даних	Apple Filing Protocol, ASTERIX, DTLS, External Data Representation, MIME, NetWare Core Protocol, XML
Дані	5	Сеансовий	керування сеансом зв'язку	NFS, ADSP, ASP, H.245, iSNS, L2TP, NetBIOS, PAP, PPTP, RPC, RTCP, SMPP, SCP, SOCKS, ZIP, SDP
Блоки	4	Транспортний	безпечне та надійне з'єднання «точка – точка»	SPX, TCP, UDP, SCTP, RTP, IL
Пакети	3	Мережевий	визначення маршруту та логічних адрес	XNS, IP, IPv4, IPv6, ICMP, ICMPv6, IGMP, IPsec
Кадри	2	Канальний	MAC та LLC (фізична адресація)	ARP, CDP, DCAP, Ethernet, FDDI, HDLC, L2F, L2TP, PPP, PPTP, Fibre Channel, Token Ring, VTP
Біти	1	Фізичний	кабель, сигнали, бінарна передача	ISDN, PDH, RS-232, SDH, SONET, ITU-T V (модемні протоколи)

Джерело: [21]

Як бачимо із табл. 2.2, представлена класифікація мережевих протоколів за моделлю OSI дозволяє їх систематизувати на основі функціонального призначення та рівня взаємодії в мережі. Кожен рівень виконує чітко визначені завдання – від фізичної передачі сигналу до забезпечення доступу користувачів до мережевих сервісів, тим самим забезпечуючи скоординовану та ефективну роботу всієї мережевої інфраструктури.

Використання багаторівневої моделі сприяє стандартизації процесів обміну даними, спрощує розробку та впровадження нових технологій, а також забезпечує сумісність різних пристроїв та програмного забезпечення,

незалежно від виробника чи архітектури, що, у свою чергу, налаштовує надійний, масштабований та безпечний обмін інформацією в сучасних комп'ютерних мережах.

Отже, мережеві протоколи, які зведені в стек TCP/IP та структуровані за моделлю OSI, таким чином формують основу функціонування локальних та глобальних мереж і гарантують цілісність, доступність та надійність переданих даних. Їхня роль є вирішальною для розвитку інформаційного суспільства та зростання цифрового зв'язку.

2.3 Віртуалізація та хмарні технології у веб-інфраструктурі

Віртуалізація та хмарні технології у веб-інфраструктурі сьогодні належать до найбільш перспективних і економічно ефективних рішень у сфері IT. Вони забезпечують гнучкість, масштабованість і високу доступність веб-сервісів. Завдяки віртуалізації стає можливим абстрагування апаратних ресурсів від програмного забезпечення, що дозволяє ефективно розподіляти обчислювальні потужності між різними компонентами веб-інфраструктури.

Віртуалізація – процес представлення набору обчислювальних ресурсів або їх логічного об'єднання, який дає які-небудь переваги перед оригінальною конфігурацією. Віртуалізація здійснюється за допомогою віртуальних машин. Які фактично представлені програмним або апаратним середовищем, яке приховує справжню реалізацію якого-небудь процесу або об'єкту від його видимого представлення. Видове різноманіття віртуалізації к веб-інфраструктурі представлено у табл. 2.2.

Таблиця 2.2 – Видове різноманіття віртуалізації

№	Вид віртуалізації	Характеристика	Особливості
1	Повна віртуалізація	Використання немодифікованих гостьових ОС	Емуляція апаратного забезпечення, робота через хостову ОС

2	Паравіртуалізація	Модифікація ядра гостьової ОС	Використання спеціального API для взаємодії з апаратурою
3	Віртуалізація на рівні ОС	Використання одного ядра ОС для кількох середовищ	Паралельна робота ізольованих середовищ
4	Віртуалізація серверів	Запуск кількох віртуальних серверів на одному фізичному	Кожна ВМ має власну ОС і додатки
5	Віртуалізація додатків	Ізоляція програм без повної інсталяції	Віртуалізація файлів, реєстру, служб; запуск із носіїв
6	Віртуалізація представлень (робочих місць)	Виконання застосунків на сервері	Користувач отримує лише інтерфейс (віддалений доступ)

Отже, наведене видове різноманіття віртуалізації демонструє широкий спектр підходів до організації обчислювальних ресурсів, що дозволяє ефективно адаптувати IT-інфраструктуру до різних потреб користувачів і завдань. Кожен тип віртуалізації має свої особливості, переваги та сфери застосування: від повної ізоляції операційних систем до оптимізації використання ресурсів на рівні додатків або робочих середовищ. Правильний вибір виду віртуалізації є ключовим фактором при проєктуванні сучасних інформаційних систем, оскільки дозволяє досягти оптимального балансу між продуктивністю, надійністю, гнучкістю та витратами на обслуговування інфраструктури. Іншими словами віртуалізація має на увазі запуск на одному фізичному комп'ютері декількох віртуальних комп'ютерів (рис. 2.1).

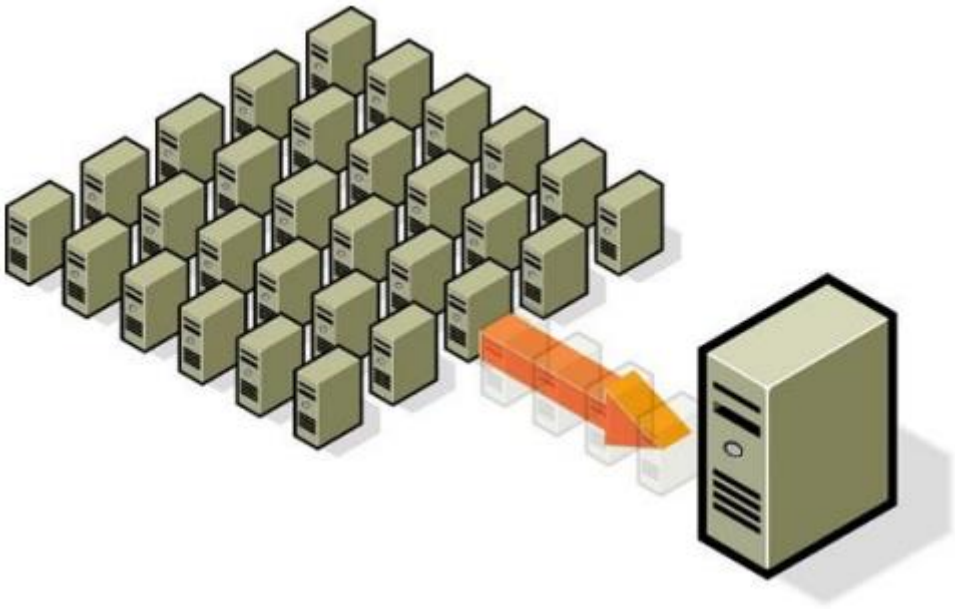


Рис. 2.1 – Візуальне зображення змісту поняття віртуалізації

Історія технологій віртуалізації налічує понад сорок років. IBM була першим постачальником, який розробив віртуальні середовища для різних завдань користувачів, спочатку на мейнфреймах. У 1960-х роках віртуалізація була суто науковим інтересом та інноваційним рішенням для ізоляції комп'ютерних систем у межах фізичної машини. Після появи ПК інтерес до віртуалізації дещо зменшився через швидкий розвиток операційних систем, які ставили високі вимоги до апаратного забезпечення того часу. Однак швидке зростання комп'ютерного обладнання наприкінці 1990-х років спонукало IT-індустрію заново відкрити технології віртуалізації для програмних платформ. Сьогодні постачальники технологій віртуалізації пропонують надійні та прості в управлінні платформи, і ринок цих технологій переживає справжній бум. За словами провідних експертів, віртуалізація зараз входить до трійки найперспективніших комп'ютерних технологій [25].

Сьогодні технології віртуалізації широко застосовуються для консолідації серверів, оптимізації використання ресурсів центрів обробки даних, а також для побудови хмарних платформ. У контексті веб-інфраструктури це дозволяє розгортати веб-сайти та сервіси на віртуальних машинах або контейнерах, що значно спрощує їх адміністрування, оновлення та масштабування залежно від навантаження.

Окрім цього, віртуалізація дає змогу підтримувати роботу застарілих веб-додатків у сучасному середовищі, ізолюючи їх у спеціальних віртуальних середовищах. Це особливо важливо для забезпечення безперервності бізнес-процесів і зниження витрат на модернізацію програмного забезпечення.

Важливим напрямом є також використання технологій віртуальних робочих середовищ і «тонких клієнтів», які дозволяють адміністраторам і користувачам отримувати доступ до веб-ресурсів і систем управління з будь-якої точки світу. При цьому всі дані та налаштування зберігаються у хмарній інфраструктурі, що підвищує рівень безпеки та спрощує управління.

Ключовою перевагою впровадження віртуалізації та хмарних технологій у веб-інфраструктурі є можливість об’єднання всіх обчислювальних ресурсів у єдине середовище – хмарний центр обробки даних. Це дозволяє значно підвищити ефективність використання ресурсів, зменшити витрати на обслуговування інфраструктури та забезпечити стабільну роботу веб-сервісів навіть за умов змінного навантаження.

Коли сьогодні говорять про технології віртуалізації, мають на увазі віртуалізацію серверів, оскільки вона стала найпопулярнішим рішенням на ринку IT. Віртуалізація серверів передбачає запуск кількох віртуальних серверів на одному фізичному сервері. Віртуальні машини, або сервери, – це програми, які працюють на хост-операційній системі та емулюють апаратне забезпечення фізичного сервера. На кожній віртуальній машині може бути встановлена власна операційна система, на яку потім можна встановлювати програми та служби. Типові представники це продукти VmWare (ESX, Server, Workstation) і Microsoft (Hyper - V, Virtual Serer, Virtual PC) [13].

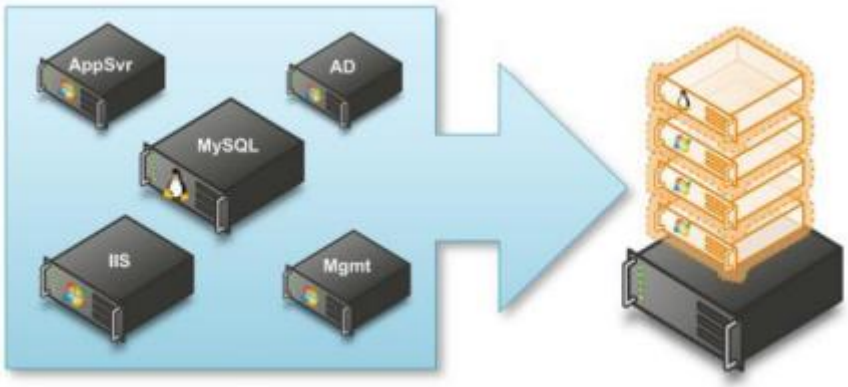


Рис. 2.2 – Візуальний приклад зображення процесу віртуалізації серверів

Платформи віртуалізації в серверній інфраструктурі представлені у табл. 2.3.

Таблиця 2.3 – Платформи віртуалізації в серверній інфраструктурі

№	Платформа / Виробник	Продукт	Тип віртуалізації	Призначення	Особливості

1	VMware	VMware ESX / ESXi	Серверна (гіпервізор)	Віртуалізація серверів	Висока продуктивність, корпоративний рівень
2	VMware	VMware vSphere	Комплексна платформа	Віртуалізація ЦОД, хмарні середовища	Централізоване управління, масштабованість
3	VMware	VMware vCenter	Управління інфраструктурою	Адміністрування віртуального середовища	Моніторинг, автоматизація, централізація
4	VMware	VMware Server	Серверна	Для малих і середніх інфраструктур	Безкоштовна, проста у розгортанні
5	Citrix	Xen (Open Source)	Паравіртуалізація	Дослідження, тестування	Відкрите ПЗ, інтеграція з Linux
6	Citrix	XenServer	Серверна	Віртуалізація серверів підприємства	Підтримка хмарних середовищ
7	Citrix	XenApp	Віртуалізація додатків	Доставка застосунків користувачам	Централізоване виконання
8	Citrix	XenDesktop	Віртуалізація робочих місць	Віддалені робочі столи	Підтримка різних клієнтів
9	Microsoft	Hyper-V	Серверна (гіпервізор)	Віртуалізація серверів	Інтеграція з Windows Server, висока продуктивність
10	Microsoft	Virtual Server	Серверна	Базова віртуалізація	Безкоштовний, обмежений функціонал

Таким чином, наведені у таблиці 2.4 платформи віртуалізації демонструють широкий спектр сучасних рішень, що використовуються для побудови та управління серверною інфраструктурою. Кожна з них має свої особливості, переваги та сферу застосування, що дозволяє обирати оптимальний інструментарій залежно від вимог конкретної інформаційної системи.

Зі стрімким розвитком веб-орієнтованих систем (веб-орієнтовані системи – це комп’ютерні системи, які доступні користувачеві з використанням програмних засобів таких, як веб-браузер) та хмарних технологій (хмарні технології – це модель забезпечення повсюдного та зручного доступу на вимогу через мережу до спільної бази даних

обчислювальних ресурсів, що підлягають налаштуванню і які можуть бути оперативно надані з мінімальними управлінськими затратами та зверненнями до провайдера [21]) важливо обрати відповідні та ефективні засоби для вирішення поставлених завдань, а саме – забезпечення доступності в будь-якій точці світу з можливістю зворотного зв'язку в реальному часі.

«Хмара – це великий пул легко використовуваних і легкодоступних віртуалізованних ресурсів (таких як апаратні комплекси, сервіси та ін.). Ці ресурси можуть бути динамічно перерозподілені (масштабовані) для підстроювання під змінну навантаження, забезпечуючи оптимальне використання ресурсів. Цей пул ресурсів зазвичай надається за принципом «оплата по мірі використання». При цьому власник хмари гарантує якість обслуговування на основі певних угод з користувачем» [45].

Історія хмарних технологій почалася досить давно. До 70-х – 80-х років відносять перші ідеї, які побічно торкалися того, що згодом і стало хмарними обчисленнями. Але все таки датою відліку сучасної історії cloud computing став 2006 рік, коли компанія Amazon, яка вже на той момент була однією з найбільших, презентувала свою інфраструктуру веб-сервісів, яка була здатна забезпечити користувачеві не лише хостинг, а й надати віддалені обчислювальні потужності клієнтові. Новинку сприйняли і схвалили такі гіганти, як Google, Sun і IBM, а в 2008 році про свій інтерес у цій галузі заявила корпорація Microsoft [32].

Трансформація сучасних управлінських моделей від ієрархій Web 2.0 до децентралізованих екосистем Web 3.0 передбачає еволюцію хмарних технологій, що докорінно змінює архітектуру управлінських комунікацій. Якщо ера Web 2.0 базувалася на централізованих хмарних платформах, де менеджер виконував роль ієрархічного контролера інформаційних потоків, то перехід до Web 3.0 вимагає відмови від моделі «центр–периферія». Концепція семантичної павутини, закладена Т. Бернерсом-Лі [45], стала каталізатором переходу до середовища, де дані отримують чітко визначене значення, зрозуміле для автоматизованих систем.

В таблиці 2.4 розглянуті можливості та організація найбільш поширених на сьогоднішній день платформ хмарних обчислень: Amazon Web Services, Google App Engine і Microsoft Windows Azure.

Таблиця 2.4 – Характеристики найбільш поширених платформ хмарних обчислень

Платформи / Характеристики	Amazon Web Services	Google App Engine	Microsoft Windows Azure
Сервіси	Обчислювальні сервіси, сервіси зберігання	Web-додатки, сервіси зберігання	Web-додатки, інші додатки, сервіси зберігання
Віртуалізація	Рівня ОС, із запущеним гіпервізором Xen	Контейнер додатків	Рівня ОС
Інтерфейс доступу користувача	Утиліти консолі Amazon EC2	Web-консоль	Портал адміністрування Microsoft Azure
Web APIs	Так	Так	Так
Середовище розробки	Відсутнє	Python, Java	Microsoft .NET

Хмарні технології пропонують масштабовану інфраструктуру і програмні засоби без прямої прив’язки до фізичних машин, при цьому економлячи витрати, серверні потужності і енергоспоживання під час простоювання.

Хмарні технології – це можливість безлічі фізичних серверів бути єдиним обчислювальним середовищем. В цілому, сервіси хмарних обчислень є додатками, доступ до яких забезпечується через Інтернет за допомогою браузера або інших мережевих застосувань, наприклад, FTP-клієнта [41].

Головна відмінність від звичного методу роботи з ПЗ полягає в тому, що користувач використовує не ресурси свого комп’ютера, або сервера своєї локальної мережі, а потужності, які надаються йому як Інтернет-послуга. При цьому користувач має повний доступ до власних даних і можливість роботи з ними з будь-якої точки світу і з будь-якого пристрою, але не обтяжений управлінням операційною системою, програмною базою, обчислювальними потужностями, за допомогою яких ця робота відбувається.

Зберігання в хмарі не лише даних, а й додатків змінює обчислювальну парадигму в бік традиційної клієнт-серверної моделі, при якій на стороні користувача зберігається мінімально необхідна функціональність [16]. Таким чином, необхідність встановлювати необхідні оновлення програмного забезпечення, проводити перевірку на віруси й інше обслуговування покладається на провайдера хмарного сервісу. Це також означає, що загальний доступ, управління версіями, спільне редагування стають набагато простішими, ніж коли додатки і дані розміщені на призначених для користувача комп'ютерах.

Основні типи хмарних обчислень та їх характеристики наведені у табл. 2.5.

Таблиця 2.5 – Основні типи хмарних обчислень та їх характеристики

Тип хмари	Хто використовує	Розміщення інфраструктури	Переваги	Недоліки
Приватна хмара (Private cloud)	Одна організація	Усередині організації або у стороннього провайдера	Високий рівень безпеки, контрольованість, керованість, нижча вартість порівняно з публічною	Обмежена масштабованість, складніше швидко розширювати ресурси
Публічна хмара (Public cloud)	Будь-які користувачі	У провайдера (зовнішня інфраструктура)	Висока масштабованість, доступність, гнучкість	Нижчий рівень контролю та безпеки, обмежені налаштування, ризики конфіденційності
Гібридна хмара (Hybrid cloud)	Організації, що мають критичні та некритичні дані	Частково у власній інфраструктурі, частково у публічній хмарі	Баланс безпеки і масштабованості, оптимізація ресурсів, гнучкість	Складність інтеграції та управління
Громадська хмара (Community cloud)	Група організацій	Спільна або змішана інфраструктура	Спільне використання ресурсів, відповідність спільним вимогам	Обмежена універсальність, залежність від спільних правил

Класифікація моделей обслуговування хмарних обчислень представлена в табл. 2.6.

Таблиця 2.6 – Класифікація моделей обслуговування хмарних обчислень

Модель	Розшифрування	Короткий опис	Основні функції / особливості
SaaS	Software as a Service (Програмне забезпечення як послуга)	Надання готових програмних додатків через Інтернет	Робота через браузер, орендна модель оплати, відсутність потреби в інсталяції та обслуговуванні ПЗ користувачем
PaaS	Platform as a Service (Платформа як послуга)	Надання платформи для розробки, тестування та розгортання додатків	Доступ до ОС, БД, інструментів розробки; керування ресурсами здійснює провайдер; гнучке масштабування
IaaS	Infrastructure as a Service (Інфраструктура як послуга)	Надання базових обчислювальних ресурсів	Віртуальні сервери, мережі, сховища; користувач керує ОС і ПЗ, але не фізичною інфраструктурою
DaaS	Data as a Service (Дані як послуга)	Надання доступу до даних через хмару	Централізоване зберігання даних, доступ через API або сервіси
WaaS	Workplace as a Service (Робоче місце як послуга)	Віртуальне робоче середовище для користувача	Віддалені робочі столи, централізоване управління робочими місцями
AaaS	All as a Service (Усе як послуга)	Комплексна модель надання всіх IT-ресурсів як сервісу	Поєднання різних хмарних сервісів у єдину інфраструктуру

Моделі хмарних обчислень утворюють багаторівневу систему для надання IT-ресурсів відповідно до потреб користувачів. Найбазовішою моделлю є IaaS (Інфраструктура як послуга), яка забезпечує доступ до ресурсів інфраструктури та дозволяє користувачам самостійно керувати операційними системами та програмним забезпеченням. Наступним рівнем є PaaS (Платформа як послуга), яка надає готове середовище для розробки та розгортання додатків без необхідності управління фізичною інфраструктурою.

Модель SaaS (Програмне забезпечення як послуга) є найбільш орієнтованою на користувача, оскільки вона пропонує доступ до готових програмних рішень через Інтернет без необхідності встановлення та адміністрування. Інші моделі, такі як DaaS (Дані як послуга), WaaS (Веб як послуга) та AaaS (Розширені ресурси як послуга), розширюють

функціональність хмарних технологій, надаючи доступ до даних, віртуальних робочих просторів та складних сервісних рішень.

Таким чином, класифікація сервісних моделей дозволяє розрізняти різні рівні абстракції в хмарних технологіях – від інфраструктури до повноцінних, готових сервісів, забезпечуючи гнучкість, масштабованість та ефективність при їх використанні в сучасних інформаційних системах.

2.4 Критерії вибору серверного обладнання та програмного забезпечення

Під час проектування та впровадження віртуалізованої веб-інфраструктури особливу увагу слід приділити вибору апаратного та програмного забезпечення сервера, оскільки саме ці компоненти формують основу функціональності всієї системи та визначають її продуктивність, надійність, гнучкість та безпеку. Враховуючи швидкий розвиток веб-технологій та зростаюче навантаження на інформаційні ресурси, вибір правильних інфраструктурних рішень має вирішальне значення для безперебійної роботи сервісів. Ключовим критерієм є продуктивність сервера, яка залежить від характеристик процесорів, обсягу оперативної пам'яті та ефективності системи зберігання даних. Для сучасних веб-інфраструктур важливо використовувати сервери з потужними багатоядерними процесорами, оскільки вони дозволяють одночасно обробляти велику кількість запитів. Достатній обсяг оперативної пам'яті необхідний для роботи безлічі віртуальних машин або контейнерів, кожен з яких споживає ресурси. Ще одним важливим аспектом є швидкість роботи системи жорстких дисків, зокрема використання SSD-накопичувачів, які значно скорочують час доступу до даних та підвищують загальну продуктивність системи.

Не менш важливим критерієм є масштабованість системи, яка дозволяє інфраструктурі адаптуватися до зростаючих потреб користувачів. Апаратне

забезпечення сервера повинно підтримувати поступове розширення – як шляхом додавання нових серверів, так і шляхом оновлення існуючих компонентів. Відзначимо, що блейд-системи є ефективним рішенням, оскільки вони забезпечують компактне розташування серверів, централізоване управління та спрощене масштабування. У контексті хмарних технологій масштабованість також означає можливість динамічного розподілу ресурсів залежно від робочого навантаження, тим самим оптимізуючи витрати.

Ще однією важливою вимогою є відмовостійкість та надійність інфраструктури. Оскільки веб-сервіси повинні бути доступні цілодобово, резервування критично важливих компонентів системи, включаючи блоки живлення, системи охолодження, мережеві інтерфейси та модулі керування, є надзвичайно важливим.

Наступним важливим критерієм є продуктивність мережі та гнучкість підключення. Сервери повинні підтримувати сучасні високошвидкісні інтерфейси передачі даних та різноманітні мережеві протоколи, забезпечуючи ефективну взаємодію між компонентами інфраструктури, такими як сервери, системи зберігання даних та зовнішні мережі. Низька затримка та висока пропускна здатність є особливо важливими, оскільки вони безпосередньо впливають на швидкість обробки запитів від користувачів веб-додатків.

Сумісність серверного обладнання з програмним забезпеченням для віртуалізації відіграє не менш вирішальну роль. Адже використання сертифікованого обладнання гарантує стабільну роботу гіпервізорів, таких як VMware ESXi та Microsoft Hyper-V, дозволяючи ефективно керувати та ізолювати віртуальні машини, а також оптимально розподіляти ресурси. Підтримка різних версій гіпервізора також пропонує гнучкість у виборі програмної платформи.

Сучасні програмні рішення дозволяють здійснювати моніторинг стану пристроїв, управління ресурсами, автоматичне виділення віртуальних машин та оновлення програмного забезпечення через єдиний інтерфейс, значно

спрощуючи роботу адміністраторів та зменшує ймовірність помилок. Тому централізовані системи управління інфраструктурою вимагають особливої уваги. Також відзначимо, що важливими є і функції автоматизації, які дозволяють швидко реагувати на зміни навантаження та оптимізувати використання ресурсів. Не менш важливим критерієм є безпека інформаційної інфраструктури. Сервери повинні підтримувати сучасні механізми захисту, такі як апаратне шифрування, безпечне завантаження операційної системи, багатофакторна автентифікація та контроль доступу.

Інтеграція з системами управління доступом для всього підприємства, особливо Microsoft Active Directory, дозволяє централізовано керувати правами користувачів та покращує захист даних. Підтримка протоколів безпечної передачі даних та механізмів виявлення загроз також має вирішальне значення.

Крім того, слід враховувати можливості інтеграції та автоматизації процесів. Сучасні серверні рішення повинні підтримувати відкриті інтерфейси прикладного програмування (API), такі як REST або Redfish, для інтеграції інфраструктури з іншими системами та сервісами. Така можливість підтримки відкриває переваги для створення складних, автоматизованих сценаріїв управління, що особливо актуально в середовищах хмарних обчислень.

Енергоефективність та оптимізація ресурсів також є важливими аспектами. Використання сучасних, енергоефективних компонентів знижує витрати на електроенергію та охолодження, що особливо важливо для великих центрів обробки даних. Віртуалізація, у свою чергу, сприяє більш ефективному використанню апаратних ресурсів, тим самим зменшуючи кількість фізичних серверів і, отже, витрати на їх обслуговування.

Останній, але не менш важливий критерій – це сервісна підтримка та технічне обслуговування. Надійна підтримка від виробника обладнання включає швидку заміну компонентів, доступ до оновлень програмного забезпечення, віддалену діагностику та цілодобову технічну підтримку. Тому вибір серверного обладнання та програмного забезпечення для віртуалізованої

веб-інфраструктури повинен базуватися на комплексному підході, який враховує продуктивність, масштабованість, надійність, безпеку та керованість. Тільки враховуючи всі ці критерії, можна створити ефективну, стабільну та сучасну інфраструктуру, яка гарантує довгострокову, бездоганну роботу веб-сервісів.

На підставі викладеного вище формулюємо узагальнену таблицю критерії вибору серверного обладнання та програмного забезпечення (табл.2.7).

Таблиця 2.7 – Критерії вибору серверного обладнання та ПЗ

№	Критерій	Характеристика	Що враховується
1	Продуктивність	Забезпечує швидку обробку запитів та стабільну роботу системи	Багатоядерні процесори, обсяг ОЗП, SSD-накопичувачі
2	Масштабованість	Можливість розширення інфраструктури відповідно до навантаження	Додавання серверів, модернізація компонентів, блейд-системи
3	Відмовостійкість та надійність	Забезпечення безперервної роботи сервісів	Резервування живлення, охолодження, мережевих інтерфейсів
4	Продуктивність мережі	Швидкість передачі даних та мінімальні затримки	Високошвидкісні інтерфейси, пропускна здатність, низька латентність
5	Сумісність з ПЗ віртуалізації	Стабільна робота віртуального середовища	Підтримка VMware ESXi, Microsoft Hyper-V, сертифіковане обладнання
6	Керованість та моніторинг	Спрощення адміністрування системи	Централізовані системи управління, моніторинг, автоматизація
7	Безпека	Захист даних та доступу до системи	Шифрування, MFA, контроль доступу, secure boot
8	Інтеграція та автоматизація	Взаємодія з іншими системами	API (REST, Redfish), інтеграція з Active Directory
9	Енергоефективність	Зменшення витрат на експлуатацію	Енергоощадні компоненти, оптимізація ресурсів
10	Сервісна підтримка	Швидке вирішення технічних проблем	Технічна підтримка 24/7, оновлення ПЗ, заміна компонентів

Отже, вибір серверного обладнання та програмного забезпечення для віртуалізованої веб-інфраструктури є багатокритеріальним завданням, що потребує комплексного аналізу технічних, організаційних та експлуатаційних

характеристик. Розглянуті критерії – від продуктивності та масштабованості до безпеки, енергоефективності й рівня сервісної підтримки, взаємопов’язані між собою та формують цілісну модель оцінювання інфраструктурних рішень. Їх збалансоване врахування дозволяє забезпечити стабільну роботу системи, ефективне використання ресурсів і можливість подальшого розвитку інфраструктури відповідно до зростаючих вимог веб-сервісів. Таким чином, лише системний підхід до вибору компонентів серверного середовища забезпечує створення надійної, масштабованої та безпечної платформи для сучасних веб-додатків.

3. ПРАКТИЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ СИСТЕМИ АДМІНІСТРУВАННЯ СЕРВЕРІВ У МЕРЕЖЕВІЙ ІНФРАСТРУКТУРІ ВЕБ-САЙТІВ

3.1 Принципи розгортання серверного середовища веб-сайту

Розгортання серверного середовища веб-сайту– це критично важливий крок у забезпеченні стабільної та високопродуктивної мережі. Від правильності прийнятих рішень на цьому етапі залежить не лише стабільність роботи веб-ресурсу, але й його продуктивність, безпека та можливість подальшого масштабування. Усі процеси, пов’язані зі встановленням сервера, зазвичай є складними та включатимуть планування, вибір відповідної архітектури, налаштування програмного забезпечення, а також забезпечення належного середовища для роботи сервера. Вони будуються на таких ключових принципах:

1. Орієнтація на вимоги веб-сайту та користувачів.
2. Вибір оптимальної архітектури серверного середовища.
3. Використання віртуалізації та ізоляції ресурсів.
4. Забезпечення надійності та відмовостійкості.
5. Безпеки серверного середовища.
6. Автоматизації процесів розгортання та налаштування.
7. Моніторингу та керованості системи.
8. Масштабованість серверного середовища.

По-перше, важливо зосередитися на вимогах веб-сайту та користувачів. Перед розгортанням необхідно чітко визначити, яке навантаження на сервер очікувати, який тип контенту (статичний чи динамічний) буде надаватися та які технології будуть використовуватися для розробки веб-сайту. Наприклад, веб-сайт з великою кількістю мультимедійного контенту або високим

трафіком вимагає потужнішого серверного обладнання та оптимізованої архітектури.

На початковому етапі ми визначаємо, що наш веб-сайт є динамічним (з використанням CMS) і розрахований на середнє навантаження до 1000-3000 користувачів на добу. Для реалізації такого завдання будемо розгортати сервер з такими технічними характеристиками, як відображені в табл. 3.1 характеристики.

Таблиця 3.1 – Технічні характеристики серверного обладнання

Компонент	Характеристика
Процесор (CPU)	4–8 ядер (наприклад, Intel Xeon E-23xx або AMD EPYC 7002/7003), частота від 2.5 ГГц
Оперативна пам’ять (RAM)	16–32 ГБ DDR4 ECC
Накопичувач (SSD)	від 500 ГБ NVMe SSD (рекомендовано 2×500 ГБ у RAID 1)
Дискова підсистема	RAID 1 (дзеркалювання) для підвищення надійності
Мережева карта	1 Gbps Ethernet (або 10 Gbps для перспективи масштабування)
Пропускна здатність	від 100 Mbps гарантованого каналу
Операційна система	Linux (Ubuntu Server 22.04 LTS / Debian 12)
Веб-сервер	Nginx (або Apache як альтернатива)
Сервер застосунків	PHP-FPM (версія 8.x) або Node.js
Система управління БД	MySQL / MariaDB або PostgreSQL
Віртуалізація	Docker (контейнеризація) або Proxmox / VMware (за потреби)
Система резервного копіювання	Автоматизовані backup (cron + rsync / mysqldump)
Система моніторингу	Netdata або Prometheus + Grafana
Безпека	Firewall (UFW), Fail2Ban, SSL (Let’s Encrypt), SSH-доступ за ключами
Джерело живлення	Надлишковий блок живлення (Redundant PSU) (для фізичного сервера)
Форм-фактор	Rack-сервер (1U/2U) або віртуальний сервер (VPS/VDS)

Виходячи із вище викладеного у табл. 3.1 відзначимо.ю що при виборі серверного обладнання основна увага приділяється швидкому SSD (краще NVMe), достатньому обсягу RAM для обробки запитів, відмовостійкості (RAID), можливості масштабування в майбутньому.

Наступний принцип – вибір оптимальної архітектури сервера. Сьогодні найчастіше використовуються клієнт-серверні або багаторівневі (трирівневі)

архітектури, де веб-сервер, сервер додатків та рівень бази даних розділені. Такий підхід дозволяє балансувати навантаження між компонентами системи, підвищує продуктивність та спрощує адміністрування. У нашому випадку доцільно передбачити майбутню масштабованість, наприклад, додаючи більше серверів або використовуючи балансувальники навантаження, тому доцільно реалізувати саме трирівневу архітектуру сервера, структура якої відображена на рис. 3.1.



Рис. 3.1 – Трирівнева архітектура сервера

При цьому кожен із рівнів відповідає за свій певний функціонал. Наприклад, Nginx відповідає за обробку HTTP/HTTPS запитів, розгорнутий PHP-FPM або Node.js – обробляє бізнес-логіку, тоді як MySQL або ж PostgreSQL зберігає всі дані. Таким чином у своїй розробці ми використовуємо один сервер з логічним розділенням ролей.

Принцип віртуалізації та ізоляції ресурсів відіграє вирішальну роль. Використання серверного середовища на основі віртуальних машин або контейнерів дозволяє ефективніше використовувати апаратні ресурси, а також пропонує гнучкість у конфігурації та управлінні системою. Крім того, ізоляція середовищ підвищує безпеку, оскільки один компонент системи не впливатиме на інші у разі збою або атаки. Згідно із цим принципом у своїй

розробці використовуємо віртуалізацію за рахунок Docker – для контейнеризації, створивши окремі середовища: один контейнер для веб-сервера та один контейнер для бази даних. Реалізація цього підходу відбувається за рахунок налаштування Docker Compose та встановлення обмежень ресурсів кожного контейнера. Таким чином, вдасться реалізувати швидку міграцію системи та забезпечити легке відновлення серверної інфраструктури після збоїв.

Ще одним важливим принципом є забезпечення надійності та відмовостійкості. Використання серверного середовища на основі віртуальних машин/контейнерів дозволяє ефективніше використовувати апаратні ресурси, забезпечуючи гнучкість у конфігурації та управлінні системою. Крім того, ізоляція середовищ підвищує безпеку, оскільки збій або атака на один компонент системи не впливає на інші. Виходячи із вище викладеного необхідним та невід'ємним кроком є першочергове закладення резервування через налаштування RAID 5. Резервне копіювання здійснюватиметься через щоденні бекапи бази даних; щотижневі повні бекапи системи.

Процес резервного копіювання організовується таким чином, щоб він запускався автоматично і не потребував постійної взаємодії адміністратора. Цей запуск досягається шляхом налаштування спеціальних запланованих завдань (cron-завдань) на сервері, які виконують скрипти резервного копіювання у заздалегідь визначений час. Наприклад, резервне копіювання бази даних виконується щоночі, коли трафік веб-сайту мінімальний, а файлів веб-сайту – кожні кілька днів або раз на тиждень (в залежності від зміни налаштувань).

Створені копії не зберігаються на одному сервері, з метою мінімізації їх втрати в разі збою. Тому файли резервних копій автоматично переносяться на інший сервер або у хмарне сховище (окремий фізичний сервер, віддалений VPS або сервіс зберігання даних). Такий підхід гарантує захист інформації навіть у разі серйозних технічних проблем з основною системою.

Окрім створення копій, важливо гарантувати можливість швидкого відновлення даних. Для цього регулярно тестується функціональність резервних копій – виконується тестове відновлення бази даних або окремих файлів, що гарантує швидке відновлення без значних втрат часу або даних у разі збою. Таким чином, резервні копії стають не просто формальністю, а справжнім механізмом захисту даних.

Не менш важливим є принцип безпеки сервера. Захист від несанкціонованого доступу має бути гарантований вже на етапі розгортання, мають бути налаштовані брандмауери (firewall (наприклад, UFW)), мають використовуватися безпечні протоколи передачі даних (HTTPS, SSH), а програмне забезпечення має регулярно оновлюватися. Особливу увагу слід приділяємо розподілу прав доступу користувачів та адміністраторів. Виходячи із цього необхідною умовою захисту сервера є обмеження прав доступу, створення окремого користувача для системного адміністрування, регулярне оновлення системи (apt update / upgrade).

Крім того, слід дотримуватися принципу автоматизації процесів розгортання та налаштування. Використання спеціалізованих інструментів (скриптів, систем управління конфігурацією) дозволяє зменшити кількість помилок, пришвидшити процес впровадження та забезпечити узгодженість налаштувань на різних серверах, що, на нашу думку, особливо важливо під час розгортання систем у великих масштабах.

Практично це реалізується через створення власних bash-скриптів, які виконують базове налаштування системи. У таких скриптах прописуються всі необхідні дії: оновлення операційної системи, встановлення потрібних пакетів, налаштування веб-сервера, бази даних та допоміжних сервісів. У результаті достатньо один раз запустити скрипт – і сервер автоматично приводиться до потрібного стану без необхідності виконувати десятки команд вручну.

Для розгортання самих сервісів використовується docker-compose, де описуються всі компоненти системи (веб-сервер, застосунок, база даних) у

вигляді єдиного конфігураційного файлу. Завдяки цьому вся інфраструктура підіймається однією командою, а всі налаштування при цьому залишаються однаковими незалежно від середовища.

У випадках, коли потрібно керувати кількома серверами або більш складною інфраструктурою, додатково застосовується Ansible. З його допомогою централізовано задається конфігурація серверів і автоматично застосовується до всіх вузлів.

Окремо варто виділити принцип моніторингу та керованості системи. Уже на етапі розгортання необхідно інтегрувати засоби моніторингу, які дозволяють відстежувати стан серверів, навантаження на ресурси, мережеву активність та інші ключові показники. Завдяки реалізації цього принципу можна оперативно реагувати на проблеми та запобігати їх виникненню.

Практична реалізація цього принципу починається одразу після запуску серверів. Встановлюється система моніторингу, зокрема Prometheus разом із Grafana для більш гнучкого аналізу даних. Після цього налаштовуємо збір основних показників: завантаження процесора, використання оперативної пам’яті, стан дисків і мережевий трафік. Далі підключаємо систему сповіщень – задаємо порогові значення, при перевищенні яких автоматично надсилаються повідомлення адміністратору (наприклад, у Telegram або на електронну пошту). У результаті адміністратор не просто спостерігає за станом системи, а отримує сигнал одразу при виникненні проблеми і може швидко відреагувати ще до того, як це вплине на роботу веб-сайту.

Завершальним принципом є гнучкість та масштабованість серверного середовища. Система проєктується таким чином, щоб її можна було легко розширити у разі зростання кількості користувачів або обсягів даних.

Система не створюється по принципу «раз і назавжди», а одразу планується розширення, зокрема: передбачається можливість додати ще один веб-сервер, підключити балансувальник навантаження (Nginx); при зростанні

виноситься база даних на окремий сервер та використовуються хмарні сервіси.

Таким чином, у процесі розгортання серверного середовища ми не просто дотримуємось теоретичних принципів, а реалізуємо їх через конкретні інструменти, технології та послідовні дії. Кожне рішення (від вибору архітектури до налаштування безпеки) приймається з урахуванням реальних умов експлуатації веб-сайту, що дозволяє створити стабільну, керовану та масштабовану систему.

Отже, розгортання серверного середовища веб-сайту є складним багатокомпонентним процесом, який вимагає врахування технічних, організаційних та безпекових аспектів. Дотримання наведених принципів дозволяє створити стабільну, продуктивну та захищену інфраструктуру, що відповідає сучасним вимогам до функціонування веб-ресурсів.

3.2 Основні етапи розробки та впровадження системи адміністрування

Після розгортання серверного середовища наступним логічним кроком є побудова системи адміністрування, яка забезпечує контроль, управління та підтримку всієї інфраструктури веб-сайту. У межах даної роботи система адміністрування розглядається не як окремий інструмент, а як комплекс рішень, що поєднує технічні засоби, політики доступу та процедури обслуговування.

Розробка та впровадження такої системи здійснюється поетапно, з урахуванням уже обраної архітектури, технологій та принципів розгортання серверного середовища.

Основні етапи розробки та впровадження системи адміністрування передбачають:

1. Аналіз вимог до адміністрування системи.
2. Проєктування структури системи адміністрування.
3. Налаштування доступу та керування користувачами.

4. Впровадження інструментів адміністрування.

5. Організація системи резервного копіювання.

6. Впровадження системи моніторингу.

7. Забезпечення безпеки адміністрування.

8. Тестування та введення в експлуатацію.

На першому етапі визначаємо, які саме задачі має виконувати система адміністрування (рис. 3.2).

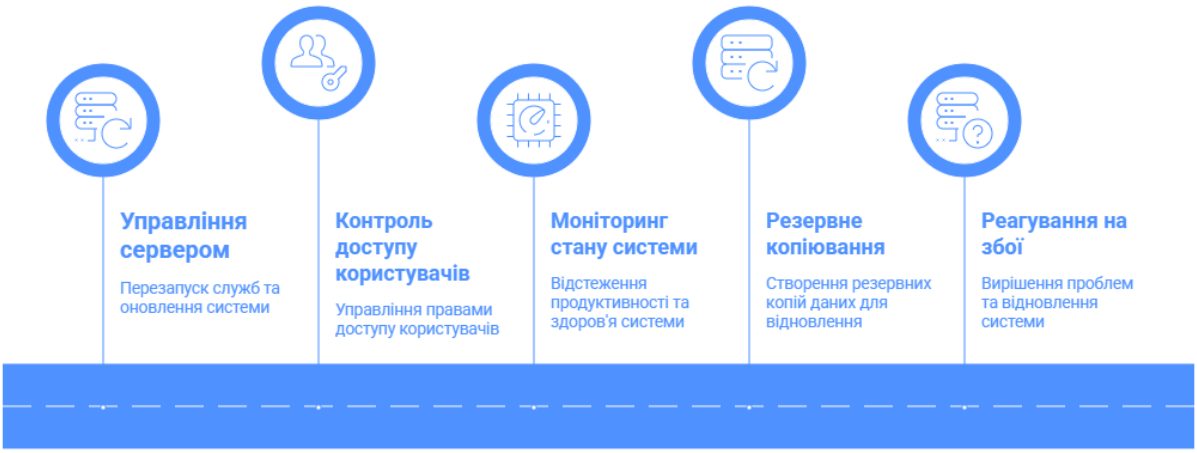


Рис. 3.2 – Задачі системи адміністрування

На цьому, першому етапі розробки та впровадження системи адміністрування важливо не ускладнювати систему, а чітко визначити, що саме потрібно контролювати.

З урахуванням попереднього пункту (трирівневої архітектури), на другому етапі будується структура адміністрування. При цьому ця побудова здійснюється окремо для трьох компонентів: веб-сервера, сервера застосунків, бази даних.

На практиці поставлена задача буде реалізуватись шляхом забезпечення доступу до серверів здійснюється через SSH, за рахунок створення єдиної схема доступу (ключі, користувачі, групи), та, у підсумку, на підставі визначення централізованого підходу до управління (наприклад, через один адміністративний вузол). Окрім цього важливо відзначити, що структура системи (IP-адреси, ролі серверів, сервіси) документується.

Без правильно організованого доступу система адміністрування не буде безпечною, тому налаштування доступу та керування користувачами є не менш важливим етапом. На цьому етапі здійснюється створення окремих користувачів, а саме: адміністратора. технічного користувача (для сервісів) та проводяться налаштування із заборони входу root через пароль та SSH-доступу за ключами.

Для розмежування прав використовуватиметься файл sudoers, при цьому доступ до критичних файлів буде обмежуватись та супроводжуватись веденням обліку дій, тобто логуванням.

На етапі впровадження інструментів адміністрування здійснюється вибір та встановлення конкретних засобів

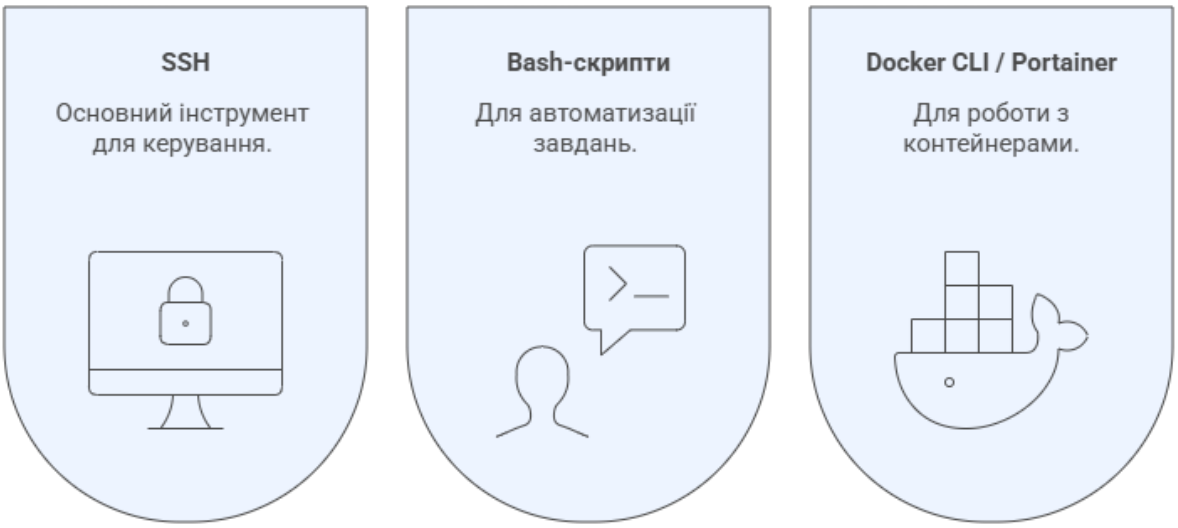


Рис. 3.3 – Встановлення конкретних засобів адміністрування

Практично реалізація цих засобів буде відбуватись наступним чином:

Першочергово будуть створені крипти для виконання таких операцій як оновлення системи, перезапуск сервісів, очищення логів.

Після цього буде реалізовано централізоване керування у вигляді управління процесами з боку одного адміністратора, який матиме доступ до усіх серверів.

На п’ятому етапі із організації системи резервного копіювання буде здійснено регулярне резервне копіювання баз даних (щоденно), файлів сайту

(щотижнево), оскільки адміністрування неможливе без надійної системи backup. Практично ця функція буде забезпечена через налаштування cron-завдання, використання mysqldump для БД, архівацію (tar) та збереження копій на окремому сервері, а за потреби у хмарному сховищі. Додатково звертаємо увагу на необхідність перевірки можливості відновлення (test restore) та забезпечуємо її надійну реалізацію.

Щоб адміністрування було ефективним, необхідно бачити стан системи в реальному часі, впровадження системи моніторингу у процесі адміністрування серверів у мережевій інфраструктурі веб-сайтів є життєво необхідним. Передбачаємо встановлення системи моніторингу Prometheus + Grafana з метою контролінгу CPU, RAM, диска; навантаження на сервер; доступності сервісів (uptime) (рис. 3.4).

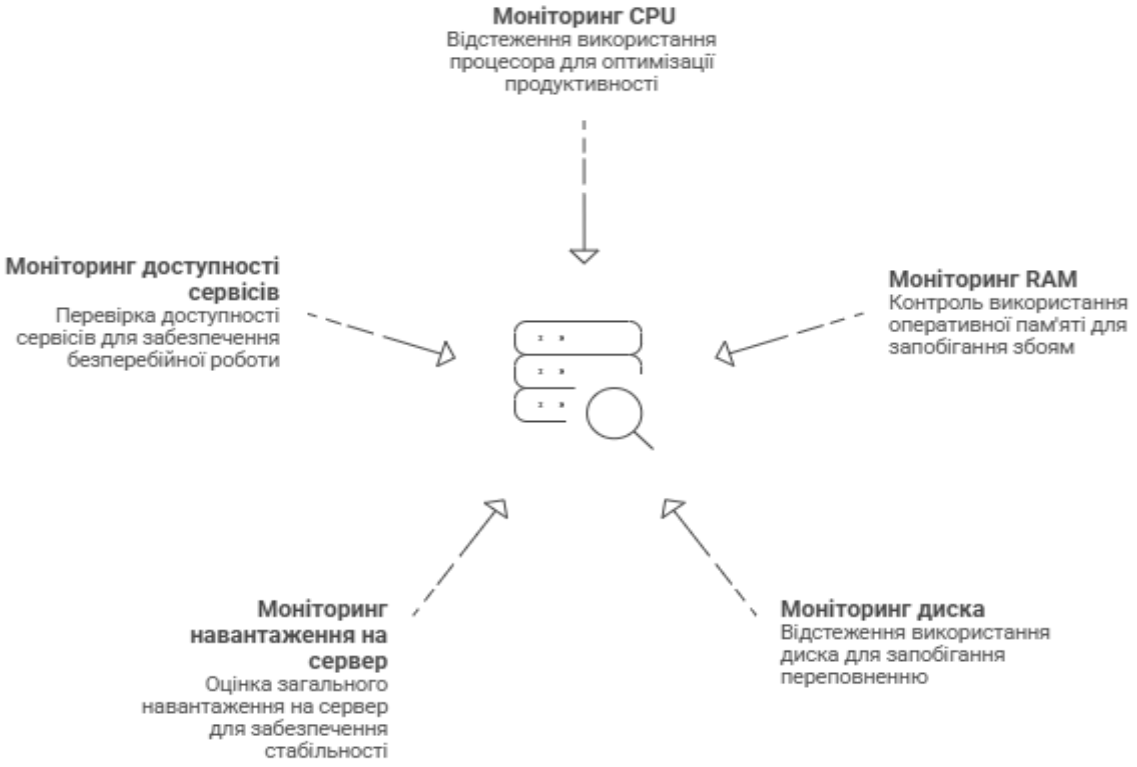


Рис. 3.4 – Система моніторингу для контролю сервера

Важливо відзначити, що у даній системі за рахунок налаштування системи сповіщення (alerts), адміністратор отримує можливість отримувати повідомлення про критичні ситуації.

Переходячи до наступного етапу, зміст якого полягає у забезпеченні безпеки адміністрування, здійснюємо посилання захисту, яке передбачає налаштування firewall (UFW / iptables), Fail2Ban (захист від brute-force) та впровадження регулярних оновлень системи, а також контролю логів.

Останній восьмий етап розробки та впровадження системи адміністрування має на меті тестування та введення в експлуатацію доступу до серверів, роботи скриптів, резервного копіювання, відновлення даних. На практиці з метою виявлення та усунення можливих помилок здійснюємо моделювання аварійних ситуацій та перевірку реакції системи у відповідь. Як результат система вводиться в експлуатацію, а саме адміністрування переходить у режим постійної підтримки.

Таким чином, розробка та впровадження системи адміністрування серверного середовища є поетапним процесом, який тісно пов’язаний із архітектурою та принципами розгортання інфраструктури. У межах даної роботи всі етапи реалізуються з урахуванням практичних потреб: від налаштування доступу до впровадження автоматизації, моніторингу та резервного копіювання. Такий підхід дозволяє забезпечити не лише ефективне управління системою, а й її стабільну та безпечну роботу в умовах реального використання веб-сайту.

3.3. Моніторинг, безпека та підтримка серверів і мережі

Після налаштування серверного середовища та впровадження системи адміністрування наступним важливим кроком є забезпечення постійного моніторингу, захисту та стабільної роботи всієї інфраструктури. Моніторинг, безпека та технічна підтримка – це не одноразові заходи, а постійні процеси, що гарантують надійність веб-сайту в реальних умовах експлуатації.

В рамках цієї розробки ці компоненти реалізуються як взаємопов’язані елементи єдиної системи управління серверною інфраструктурою.

Моніторинг є основою ефективного адміністрування, оскільки він дозволяє отримувати інформацію про стан серверів та мережі в режимі реального часу.

У нашій реалізації впроваджується система моніторингу на базі Prometheus + Grafana (для більш гнучкого та масштабованого рішення). Контроль здійснюється за структурою, відображеною на рис. 3.5.

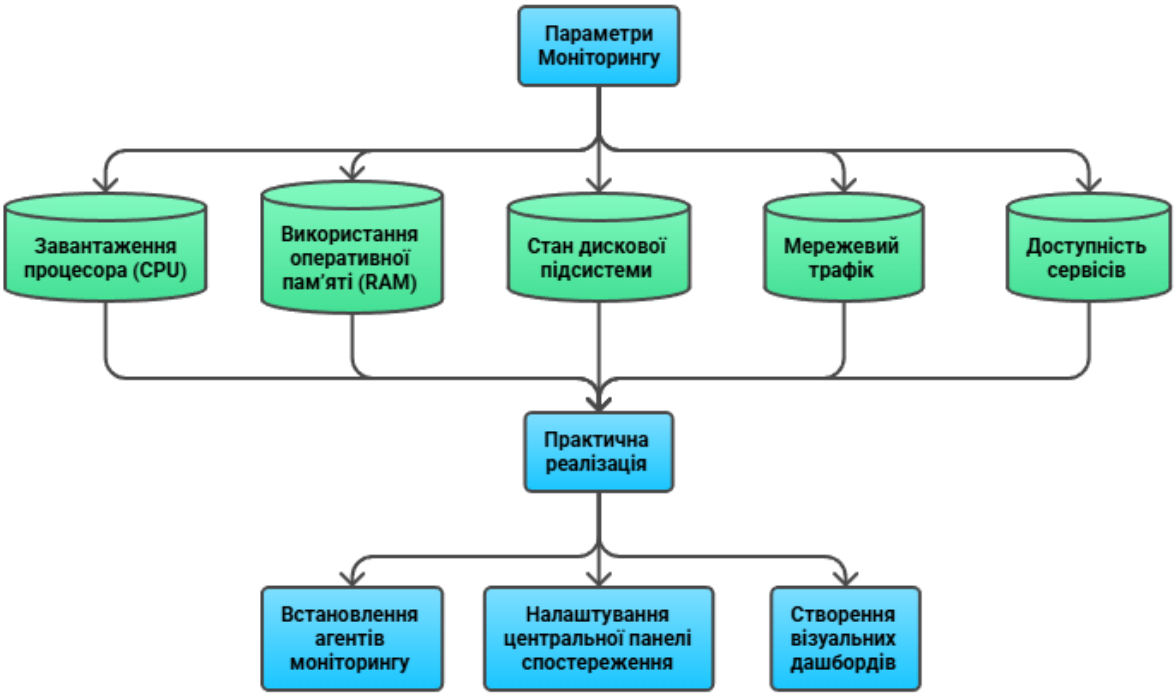


Рис. 3.5 – Структура забезпечення контролю за реалізацією моніторингу серверів і мережі

Також впроваджуються системи сповіщень, які повідомляють адміністратора про можливі збої, а саме:

- відмову сервісів;
- нестачу ресурсів;
- перевищення навантаження.

Забезпечення інформаційної безпеки серверного середовища є не менш важливим процесом, адже саме безпека є одним із ключових аспектів функціонування серверної інфраструктури, особливо в умовах відкритого доступу до веб-сайту через мережу Інтернет.

У нашій системі реалізовано багаторівневий підхід до захисту, який відображено на рис. 3.6.

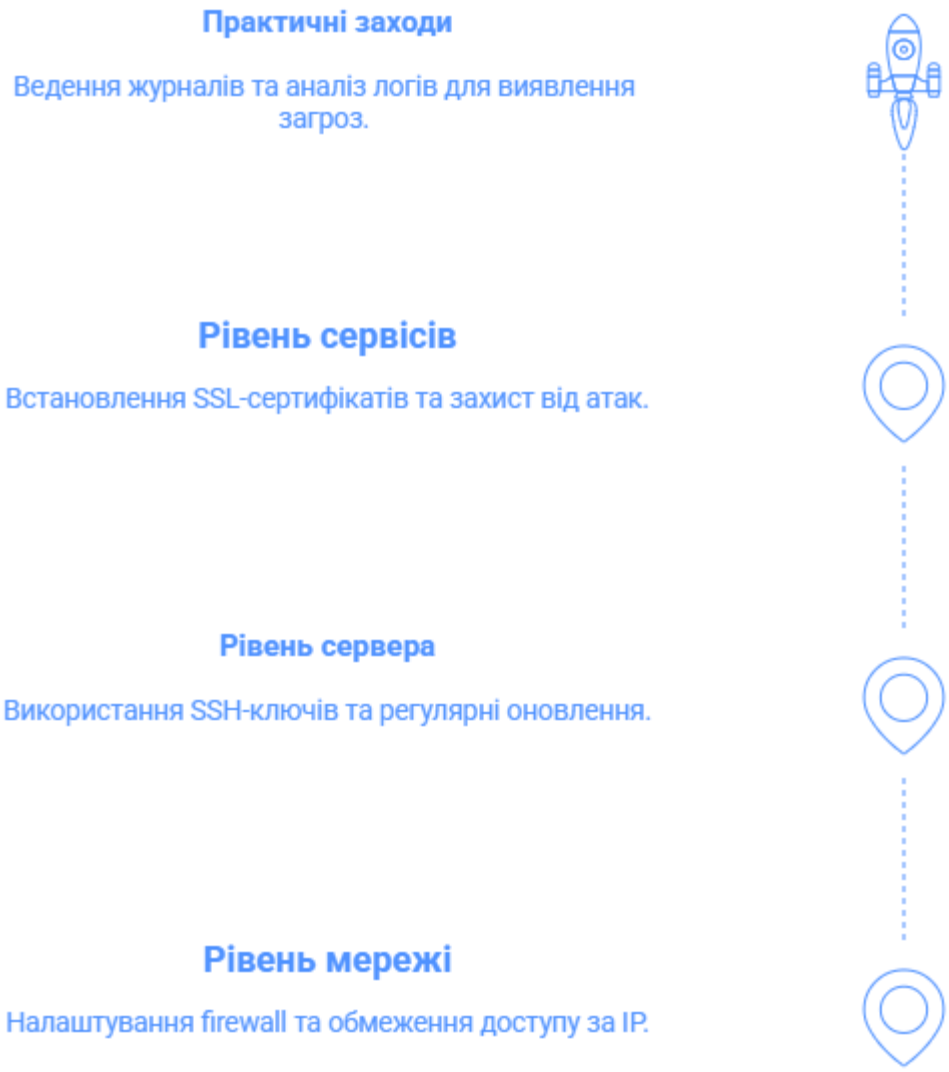


Рис. 3.6 – Багаторівневий підхід до захисту серверів і мережі

Також зазначимо, що іще однією невід'ємною частиною підтримки серверів є організація системи резервного копіювання, особливості забезпечення та організації якої були описані у попередніх пунктах.

Особливої уваги заслуговує технічна підтримка та обслуговування серверів, яка включає комплекс регулярних дій, спрямованих на забезпечення її стабільної роботи. До основних заходів із цього числа належать:

– оновлення системи та програмного забезпечення;

– оптимізація роботи бази даних;
– очищення логів і тимчасових файлів;
– перевірка стану апаратних ресурсів.

На практиці виконуються такі заходи:

– створення автоматизованих скриптів для обслуговування;
– здійснення планових перевірок системи;
– реєстрація всіх змін у конфігурації.

Крім того, значна увага приділяється веденню журналу адміністрування та документуванню виконаних операцій.

Реагування на інциденти та відновлення працездатності є іще одним заходом забезпечення безпеки сервера і мережі, оскільки у процесі експлуатації можливі збої або атаки, тому необхідно мати чіткий алгоритм дій. З метою попередження та швидкого реагування на можливі загрози передбачено локалізацію несправностей, швидке виявлення проблем через моніторинг, відновлення сервісів. На цьому етапі першопочатково здійснюємо перезапуск служб (nginx, база даних), після цього – відновлення з резервної копії та на завершення – аналіз причин збою. За результатами такого моніторингу впроваджується, як наслідок, базова стратегія disaster recovery та обов’язково визначається пріоритет відновлення критичних сервісів.

Окрім серверів, важливою є стабільність мережі. Тому варто віддати належне місце і підтримці мережевої інфраструктури. У пропонованій нами розробці реалізації алгоритму впровадження та адміністрування серверів у мережевій інфраструктурі веб-сайтів контролюється пропускна здатність мережі, обов’язково відстежуються затримки та втрати пакетів, а також перевіряється доступність серверів. На цьому етапі для забезпечення практичної реалізації заходів використовуються інструменти ping, traceroute, мережеві модулі систем моніторингу з метою забезпечення стабільного підключення до Інтернету, пошуку резервних каналів.

Таким чином, моніторинг, захист та підтримка серверів і мереж є останнім, але постійним компонентом роботи серверного середовища веб-

сайту. Запропонована модель моніторингу, безпеки та підтримки серверів і мережі – від впровадження систем моніторингу до організації резервного копіювання даних та реагування на інциденти безпеки. Вважаємо, що такий підхід забезпечує безперервну, безпечну та ефективну роботу веб-сайту та закладає основу для його подальшого розвитку та масштабування.

ВИСНОВКИ

Мережева інфраструктура є основою функціонування сучасних веб-сайтів та інформаційних систем, оскільки забезпечує взаємодію між пристроями, доступ до ресурсів і належний рівень захисту даних. У ході виконання роботи було встановлено, що ефективне функціонування веб-ресурсів неможливе без надійного серверного середовища та правильно організованого системного адміністрування. Адміністрування серверів розглядається як комплексний процес, що поєднує технічні, організаційні та безпекові аспекти, а його основними принципами є забезпечення безпеки, моніторинг, резервне копіювання, автоматизація та контроль доступу.

У процесі дослідження було проаналізовано теоретичні основи мережевої інфраструктури веб-сайтів та моделей системи адміністрування серверів, визначено їхні особливості, переваги та сфери застосування. Розглянуто сучасні серверні технології, зокрема віртуалізацію, контейнеризацію та хмарні обчислення, а також мережеві протоколи, що забезпечують стабільність і надійність передачі даних. Окрему увагу приділено критеріям вибору серверного обладнання та програмного забезпечення, що дозволяє сформувати ефективну, масштабовану та безпечну інфраструктуру.

У практичній частині роботи розроблено підхід до розгортання серверного середовища веб-сайту з урахуванням реальних умов експлуатації. Визначено основні принципи побудови серверної інфраструктури, обґрунтовано вибір архітектури, програмного забезпечення та засобів віртуалізації. Розроблено та описано етапи впровадження системи адміністрування, включаючи налаштування доступу, автоматизацію процесів, організацію резервного копіювання та використання сучасних інструментів управління.

Також у роботі реалізовано підхід до моніторингу, безпеки та підтримки

серверів і мережі. Запропоновано використання систем моніторингу, засобів захисту та механізмів реагування на інциденти, що дозволяє забезпечити безперервну та стабільну роботу веб-сайту. Визначено заходи щодо технічного обслуговування та підтримки інфраструктури, які мінімізують ризики збоїв і підвищують ефективність роботи системи.

Отже, у результаті виконання роботи досліджено теоретичні та практичні аспекти побудови мережевої інфраструктури веб-сайтів, розроблено підхід до розгортання та адміністрування серверного середовища, а також обґрунтовано використання сучасних технологій і засобів управління. Запропоновані рішення забезпечують створення надійної, безпечної та масштабованої інфраструктури.

Перспективи подальшого розвитку полягають у вдосконаленні підходів до адміністрування шляхом ширшого впровадження хмарних технологій, контейнеризації та автоматизації процесів, а також у розвитку систем моніторингу з елементами прогнозування та підвищенні рівня кібербезпеки, що дозволить створювати більш гнучкі, ефективні та адаптивні серверні інфраструктури відповідно до сучасних вимог інформаційного середовища.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адміністрування веб-сервера: Налаштування серверного середовища. IT рейтинг UA. URL: https://surl.lu/rcwekf

2. Адміністрування операційних систем. URL: https://e-tk.lntu.edu.ua/mod/resource/view.php?id=10464

3. Адміністрування серверів – що це і для чого потрібно. URL: https://tuthost.ua/uk/blog/administruvannya-serveriv-shho-tse-i-dlya-chogo-potribno/

4. Алексієв В.О., Єрещенко О.Д., Скороход О.А. Особливості розгортання сучасного веб-сайту. URL: https://repository.hneu.edu.ua/bitstream/123456789/23567/1/Aleksiyev_KhNUE.pdf

5. Архітектура клієнт-сервер. URL: http://inter.ptngu.com/kompyuterni-merezhi/arhitektura-kliyent-server

6. Архітектура розподілених програмних систем: методичні вказівки до виконання лабораторних/ уклад. Р. В. Мазуренко, С. В. Білощицька. Київ : КНУБА, 2025. 40 с. URL: https://repositary.knuba.edu.ua/server/api/core/bitstreams/644378d5-54bf-4882-9278-9b1d05644e0c/content

7. Бунке О.С. Серверні web-технології. Навчальний посібник Київ КПІ ім. Ігоря Сікорського 2023. 109 с. URL : https://ela.kpi.ua/server/api/core/bitstreams/5e3e77bd-fd03-4154-8841-dc16f26a7d1c/content

8. Використання локальних комп’ютерних мереж. URL: http://nkkep.com/wp-content/uploads/2020/11/P-22-Inform-16.11.pdf

9. Віддалене адміністрування й підтримка серверів та додатків. URL: https://surl.li/axjtfr

10. Войцеховська О. В., Чехместрук Р. Ю., Марценюк Д. В.,

Войцеховський О. В. «Система автоматичного аварійного відновлення інформаційних вебресурсів у хмарному середовищі». *Вісник ВПІ*. Вип. 1. 2026. С. 159–166. URL: https://visnyk.vntu.edu.ua/index.php/visnyk/article/view/3416

11. Волосюк Ю. В. Комп'ютерні мережі : курс лекцій. Миколаїв : МНАУ, 2019. 203 с.

12. Гайда Н. Порівняння протоколів маршрутизації RIP I EIRGP. *VIII Всеукраїнська студентська науково - технічна конференція «Природничі та гуманітарні науки. Актуальні питання».* С. 101. URL : https://elartu.tntu.edu.ua/bitstream/123456789/12668/2/Conf_2015v1_Gaida_N_V-Comparison_of_RIP_and_EIRGP_101.pdf

13. Галицький О.В., Микитенко П.В., Франчук В.М. Хмарні технології як засіб підтримки онлайн-заходів. *Новітні комп'ютерні технології спецвипуск «Хмарні технології навчання».* Т. XIII. 2015. С. 158-166. URL: https://www.researchgate.net/profile/Pavlo-Mykytenko-2/publication/363201657_Cloud_computing_as_a_tool_to_support_online_activities/links/6339dd3776e39959d68ff97c/Cloud-computing-as-a-tool-to-support-online-activities.pdf

14. Голь В.Д., Ірха М.С. Телекомунікаційні та інформаційні мережі6 навчальний посібник. Київ: ІСЗЗІ КПІ ім. Ігоря Сікорського, 2021. 250 с.

15. Городецька О. С., Гикавий В. А., Онищук О. В. Комп'ютерні мережі : навчальний посібник Вінниця: ВНТУ, 2017. 129 с.

16. Гриб'юк О.О. Перспективи впровадження хмарних технологій в освіті. URL: https://lib.iitta.gov.ua/id/eprint/1111/1/grybyuk-stattya1-hmary%2B_Copy.pdf

17. Жаврук Н. В., Вакалюк Т.А. Технічні аспекти створення сайтів за допомогою CMS. URL : https://eprints.zu.edu.ua/25717/1/%D0%96%D0%B0%D0%B2%D1%80%D1%83%D0%BA%20%D0%9D.%20%D0%92..pdf

18. Жураківський Б. Ю., Зенів І. О. Розробка та реалізація мережних протоколів: Навчальний посібник: навч. посіб. для студ. спеціальності 121

«Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології», спеціалізації «Інженерія програмного забезпечення інформаційно управляючих систем» та «Інформаційне забезпечення робототехнічних систем». КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2020. 462 с.

19. Заїка В. Ф., Варфоломеєва О. Г., Колченко Г. Ф., Миронов Д. В., Перепелиця Н. Л. Методологія застосування угод про рівень обслуговування (SLA). Зв'язок. No 3. 2025. С. 8-13. URL: https://con.dut.edu.ua/index.php/communication/article/view/2871/2764

20. Захарченко С.М., Суприган О.І. основи системного адміністрування комп'ютерних мереж на базі ОС Windows. Навчальний посібник. Вінниця: ВНТУ, 2008. 100 с. URL: https://surl.li/qtbimq

21. Зінченко О.В., Іщеряков С.М., Прокопов С.В., Сєрих С.О., Василенко В.В. «Хмарні технології». 2020. 74 с. URL: https://duikt.edu.ua/uploads/l_2048_32915773.pdf

22. Киричек Г., Смірнов В., Тягунова М. Дослідження застосування контейнерних технологій для розгортання програм на суперкомп'ютерах. *Комп'ютерні науки. Вісник КрНУ імені Михайла Остроградського.* Випуск 3/2023 (140). С. 29-35. URL: https://visnikkrnu.kdu.edu.ua/statti/2023_3_2023_3_29.pdf

23. Комп'ютерні мережі. Конспект лекцій / Укл.: Зав'ялець Ю.А. Чернівці, 2015. 183 с.

24. Комп'ютерні мережі: частина 1 навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології», спеціалізації «Інженерія програмного забезпечення інформаційно управляючих систем» та «Інформаційне забезпечення робототехнічних систем»/ Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2020. – 336 с.

25. Конспект лекцій з дисципліни «Грід-системи та технології хмарних обчислень» для студентів освітніх рівнів «бакалавр», «магістр» /

Укладачі : Шимчук Г.В., Маєвський О.В., Назаревич О.Б., Стадник М.А. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2016. 340 с.

26. Костюк Ю.В., Складаний П.М., Рзаєва С.Л., Мазур Н.П., Черевик В.М., Аносов А.О. Особливості реалізації мережевих атак через tcp/ip-протоколи. *Кіберббезпека: освіта, наука, техніка*. №1 (29), 2025. С. 571-597. URL ; https://csecurity.kubg.edu.ua/index.php/journal/article/view/915/755

27. Кузнецов Д.В., Люта М.В. Віддалене адміністрування комп'ютерних мереж. *І Всеукраїнська конференція здобувачів вищої освіти і молодих учених «Інноватика в освіті, науці та бізнесі: виклики та можливості».* С. 267-272.https://er.knutd.edu.ua/bitstream/123456789/17464/1/Innovatyka2020_P267-272.pdf

28. Логвин Д. А., Божуха Л. М. Метрики оцінювання ефективності програмного забезпечення для обробки природної мови. С. 266-276. URL : https://actualproblems.dp.ua/index.php/APAIT/article/download/313/284

29. Мережева інфраструктура. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/network-infrastructure?srsltid=AfmBOoqHWIdq_vr6Jj5eZudB0TDv9MTVt78wbQccZvMBPt19x29r4hRM

30. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі. Львів : «Магнолія 2006», 2013. 256 с.

31. Морозов А. В., Шапіренко С. Л. Автоматизований аналіз клієнтських відгуків для прийняття управлінських рішень у сервісних системах. *Технічна інженерія*. № (2(96). 2025. С. 108–116.

32. Нагаєв В.М.. Чалий І.В., Гіржева О.М. Хмарні технології WEB 3.0 ТА WEB 4.0 цифрового лідерства та кіберзахисту у професійному становленні сучасного менеджера. *Economic paradigm.* URL: https://humanitarian.com.ua/index.php/economics/article/download/783/808

33. Полотай О., Дубик А.-О. Розроблення моделі технічного захисту мережево ї інфраструктури організації. URL: https://sci.ldubgd.edu.ua/bitstream/123456789/12348/1/Tezy_Dubyk.pdf

34. Принципи захисту сервера від злому. IT для Бізнесу. URL: https://itfb.com.ua/uk/printsipi-zahistu-servera-vid-zlomu/

35. Протокол IP. URL: https://comsys.kpi.ua/wp-content/uploads/2025/02/02.pdf

36. Проценко Н.М. Економічна інформатика: навч. посіб. Харків, 2020. 212 с.

37. Смірнова Т.В.. Поліщук Л.І., Смірнов О.А., Буравченко К.О., Макевнін А.О. Дослідження хмарних технологій як сервісів. *Кібербезпека: освіта, наука, техніка.* № 3 (7), 2020. URL: RESEARCH-OF-CLOUDY-TECHNOLOGIES-AS-A-SERVICES.pdf

38. Сніцаренко А., Коробань О. Конкуренція на ринку браузерів: порівняльна характеристика . *Наука. Освіта. Молодь.* Частина 3 . С. 63-66. URL : https://library.udpu.edu.ua/library_files/stud_konferenzia/2017_3/23.pdf

39. Сумцов Д.В., Уваров Д.А. Узагальнена оцінка продуктивності серверної платформи комп'ютерної мережі. *Інфокомунікаційні системи*. URL: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/soi_2011_4_50.pdf

40. Термінологічний словник з використання інформаційних технологій та електронних бібліотечних систем / упорядник: Кільченко А. В. Київ : ІІТЗН НАПН України, 2014. 15с. URL: https://ekt.elit.sumdu.edu.ua/wp-content/uploads/2022/12/Posibnyk_Rozrobka-ta-realizatsia-merezhnykh-protokoliv.pdf

41. Ткач В.А. Хмарні сервіси в освіті. Удосконалення інформаційно-ресурсного забезпечення освіти і науки в умовах євроінтеграції : тези доповідей XVIII Міжнародної науково-практичної конференції (26-28 трав. 2016 р.) / Білоцерківський інститут економіки та управління Університету

«Україна» ; за заг. ред. П. І. Юхименка ; наук. ред. І. Г. Романченко. Біла Церква, 2016. 194 с. URL: https://eir.kristti.com.ua/storage/app/uploads/public/641/2d3/33e/6412d333e29cc832323130.pdf

42. Хоменко В.Г., Павленко М.П. Комп'ютерні мережі : Навчальний посібник. Донецьк : ЛАНДОН-ХХІ, 2011. 316 с.

43. Чевардін Є.В. Організаційно-технічні рішення щодо побудови захищених інформаційно-комунікаційних систем. *Системи і технології зв'язку, інформатизації та кібербезпеки.* № 6. 2024. С. 250-263. URL: https://journal.viti.edu.ua/index.php/cicst/article/download/109/26_Article_21.pdf

44. Черніков І. Моделі хмарних сервісів: різниця між IaaS, SaaS, PaaS та приклади. URL: https://surl.li/mfqjek

45. Шемпур В.П., Безпоясний Б.С. Термінологія хмарних технологій. URL: http://library.ippro.com.ua/attachments/article/498/%D0%A5%D0%9C%D0%90%D0%A0%D0%9D%D0%86%20%D0%A2%D0%95%D0%A5%D0%9D%D0%9E%D0%9B%D0%9E%D0%93%D0%86%D0%87.pdf

46. Юскович-Жуковська В. І., Богут О. М. Web-програмування : Підручник. Рівне : Волин. обереги, 2023. 384 с.

47. David T. Bourgeois. Дані та бази даних. 2022. URL: https://surl.li/uxndji

48. DR та Backup рішення – як комплексний захист бізнесу. URL: https://denovo.ua/storage/files/mIb3muibnnkm7u2iqpYUl5iR6GE0hbEVQKwQszoz.pdf

49. Romanov M. O., Badalin A. O., Rybalka D. V., Borysov V. V.Methods for Enhancing Security in Linux Server Administration: A Study of Protection Practices in Server Management – Permission Configuration, Auditing, Two-Factor Authentication, SSH Access Control. *Программные системы и вычислительные методы*. 2025. № 4. URL: https://www.researchgate.net/publication/399441213_Methods_for_Enhancing_Se

curity_in_Linux_Server_Administration_A_Study_of_Protection_Practices_in_Server_Management_-_Permission_Configuration_Auditing_Two-Factor_Authentication_SSH_Access_Control

50. Swain A. K., Garza V. R. Key factors in achieving service level agreements (SLA) for information technology (IT) incident resolution. *Information Systems Frontiers*. 2022. Vol. 25, No. 2. P. 819–834.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
на тему: «Система адміністрування серверів та управління мережевою інфраструктурою веб-сайтів»

Виконав студент групи КІД-42
_Ілля КУЛИКОВСЬКИЙ_____

Керівник кваліфікаційної роботи:
_Ірина КУЗЬМІЧ_____,

Київ – 2026

Мета роботи – вивчення системи адміністрування серверів та управління мережевою інфраструктурою веб-сайтів.

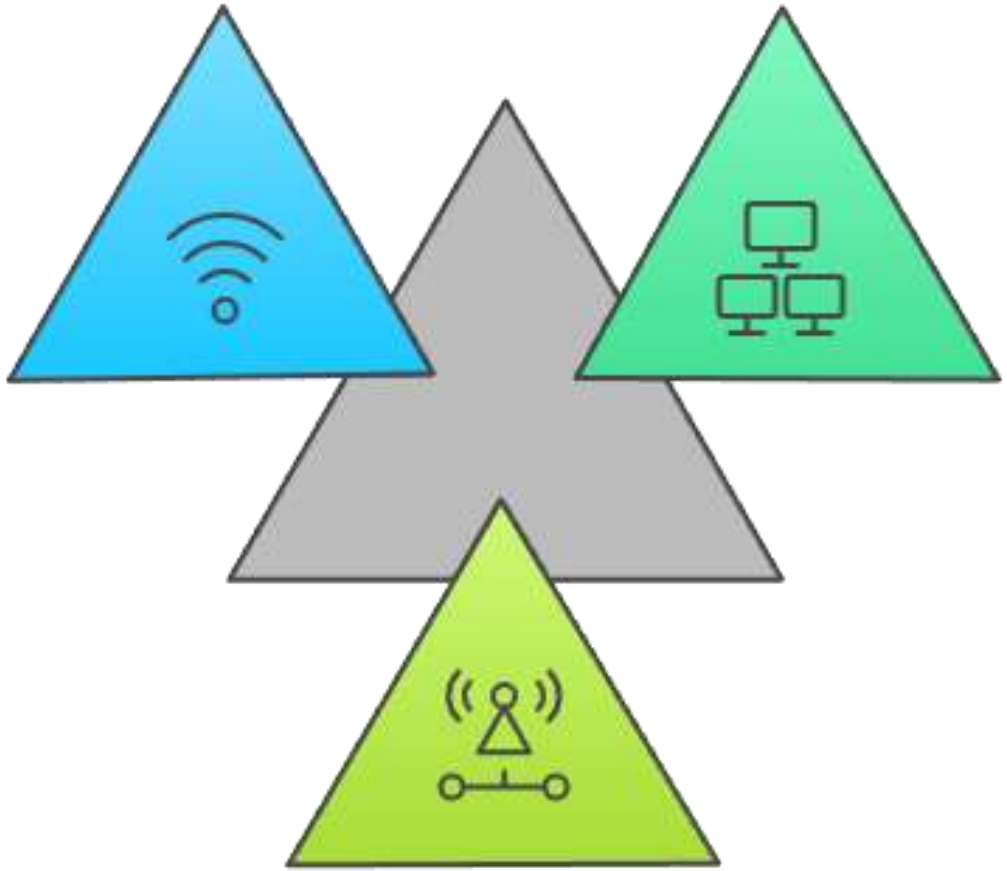
Об`єкт дослідження – функціональні процеси та процеси розвитку мережевої інфраструктури веб-сайтів у контексті забезпечення їх стабільної, безпечної та ефективної роботи.

Предмет дослідження – методи, технології та інструменти системи адміністрування серверів та управління мережевою інфраструктурою веб-сайтів, включаючи інструменти моніторингу, безпеки та оптимізації ресурсів.

Актуальність обраної теми

У сучасних умовах стрімкої цифровізації суспільства та розвитку інформаційних технологій особливої значущості набуває ефективне адміністрування серверів і управління мережевою інфраструктурою веб-сайтів. Переважна більшість бізнес-процесів, державних сервісів і комунікаційних платформ функціонують на основі веб-технологій, що зумовлює підвищені вимоги до стабільності, безпеки та продуктивності серверних систем. Зростання кількості веб-додатків і обсягів оброблюваних даних супроводжується підвищенням навантаження на серверну інфраструктуру, що потребує впровадження сучасних підходів до адміністрування, моніторингу та оптимізації ресурсів.

Складові мережевої інфраструктури



Локальна мережа

Об'єднує комп'ютери та ресурси в межах підприємства

Система передачі даних

Забезпечує надійну передачу інформації між системами

Бездротові мережі

Створює мережі без проводів для ефективної взаємодії

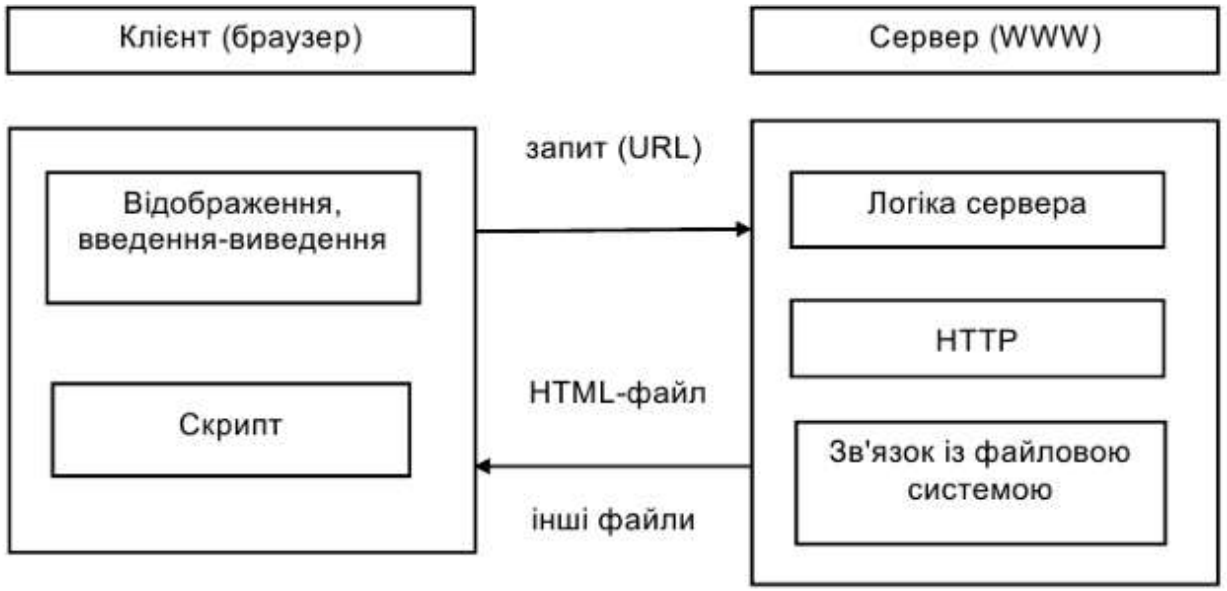
Рівні технічної підтримки серверів

Рівень	Характеристика	Основні функції
L1 (базовий рівень)	Початковий рівень підтримки	Прийом звернень, первинна діагностика, вирішення типових проблем
L2 (середній рівень)	Поглиблена технічна підтримка	Аналіз складніших інцидентів, налаштування сервера, робота з конфігураціями
L3 (експертний рівень)	Висококваліфікована підтримка	Усунення критичних збоїв, оптимізація системи, розробка рішень
L4 (вендор/розробник)	Підтримка виробника	Виправлення помилок програмного забезпечення, оновлення продуктів

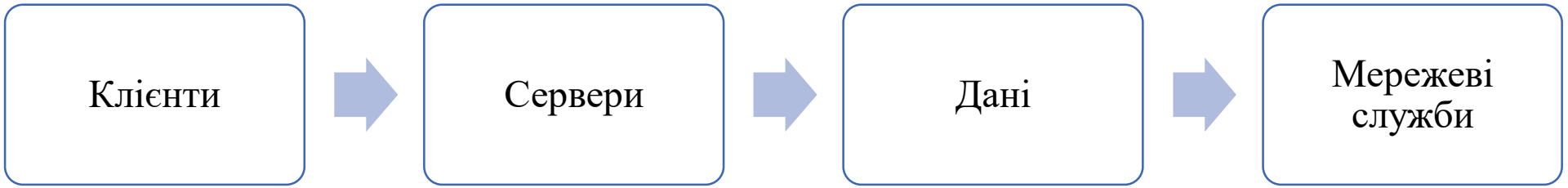
Типові рівні SLA

Рівень SLA	Доступність	Час реагування	Сфера застосування
Базовий	95–99%	До 24 годин	Некритичні системи
Стандартний	99–99,9%	4–8 годин	Корпоративні системи
Розширений	99,9–99,99%	1–2 години	Бізнес-критичні сервіси
Критичний (Premium)	≥99,99%	До 15 хвилин	Високонавантажені та фінансові системи

Клієнт-серверна архітектура у WEB



Структура сучасної клієнт-серверної архітектури

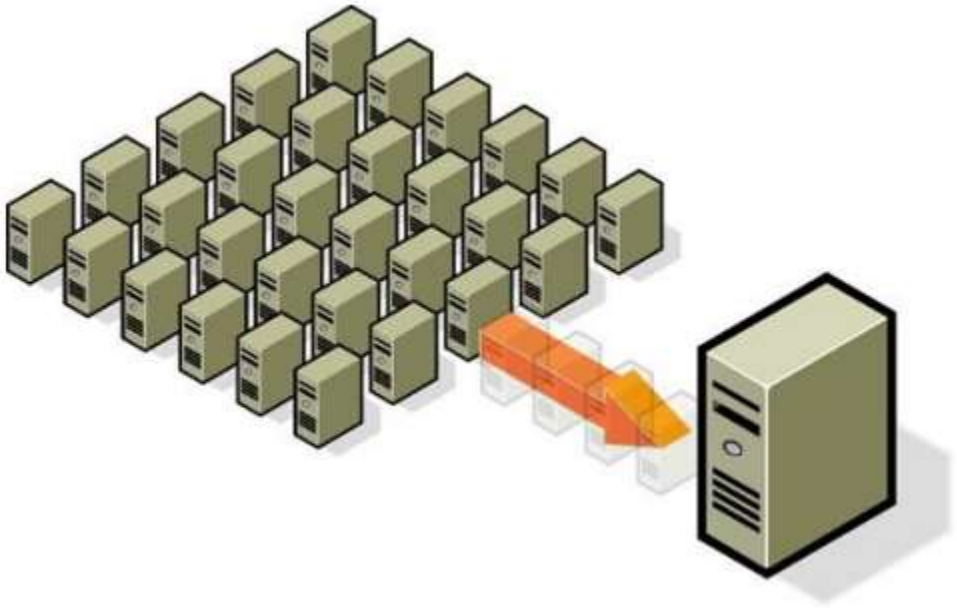


Список мережевих протоколів

Тип даних	№ рівня	Рівень OSI	Опис	Протоколи
Дані	7	Прикладний	доступ до мережевих служб	NTP, Gopher, Finger, NNTP, LDAP, DHCP, IRC, Jabber, WebDAV, DICT, BACnet, HTTPS та ін.
Дані	6	Представлення	представлення і кодування даних	Apple Filing Protocol, ASTERIX, DTLS, External Data Representation, MIME, NetWare Core Protocol, XML
Дані	5	Сеансовий	керування сеансом зв'язку	NFS, ADSP, ASP, H.245, iSNS, L2TP, NetBIOS, PAP, PPTP, RPC, RTCP, SMPP, SCP, SOCKS, ZIP, SDP
Блоки	4	Транспортний	безпечне та надійне з'єднання «точка – точка»	SPX, TCP, UDP, SCTP, RTP, IL
Пакети	3	Мережевий	визначення маршруту та логічних адрес	XNS, IP, IPv4, IPv6, ICMP, ICMPv6, IGMP, IPsec
Кадри	2	Канальний	MAC та LLC (фізична адресація)	ARP, CDP, DCAP, Ethernet, FDDI, HDLC, L2F, L2TP, PPP, PPTP, Fibre Channel, Token Ring, VTP
Біти	1	Фізичний	кабель, сигнали, бінарна передача	ISDN, PDH, RS-232, SDH, SONET, ITU-T V (модемні протоколи)

Візуальний приклад зображення процесу віртуалізації серверів





Платформи віртуалізації в серверній інфраструктурі

№	Платформа / Виробник	Продукт	Тип віртуалізації	Призначення	Особливості
1	VMware	VMware ESX / ESXi	Серверна (гіпервізор)	Віртуалізація серверів	Висока продуктивність, корпоративний рівень
2	VMware	VMware vSphere	Комплексна платформа	Віртуалізація ЦОД, хмарні середовища	Централізоване управління, масштабованість
3	VMware	VMware vCenter	Управління інфраструктурою	Адміністрування віртуального середовища	Моніторинг, автоматизація, централізація
4	VMware	VMware Server	Серверна	Для малих і середніх інфраструктур	Безкоштовна, проста у розгортанні
5	Citrix	Xen (Open Source)	Паравіртуалізація	Дослідження, тестування	Відкрите ПЗ, інтеграція з Linux
6	Citrix	XenServer	Серверна	Віртуалізація серверів підприємства	Підтримка хмарних середовищ
7	Citrix	XenApp	Віртуалізація додатків	Доставка застосунків користувачам	Централізоване виконання
8	Citrix	XenDesktop	Віртуалізація робочих місць	Віддалені робочі столи	Підтримка різних клієнтів
9	Microsoft	Hyper-V	Серверна (гіпервізор)	Віртуалізація серверів	Інтеграція з Windows Server, висока продуктивність
10	Microsoft	Virtual Server	Серверна	Базова віртуалізація	Безкоштовний, обмежений функціонал

Основні типи хмарних обчислень та їх характеристики

Тип хмари	Хто використовує	Розміщення інфраструктури	Переваги	Недоліки
Приватна хмара (Private cloud)	Одна організація	Усередині організації або у стороннього провайдера	Високий рівень безпеки, контрольованість, керованість, нижча вартість порівняно з публічною	Обмежена масштабованість, складніше швидко розширювати ресурси
Публічна хмара (Public cloud)	Будь-які користувачі	У провайдера (зовнішня інфраструктура)	Висока масштабованість, доступність, гнучкість	Нижчий рівень контролю та безпеки, обмежені налаштування, ризики конфіденційності
Гібридна хмара (Hybrid cloud)	Організації, що мають критичні та некритичні дані	Частково у власній інфраструктурі, частково у публічній хмарі	Баланс безпеки і масштабованості, оптимізація ресурсів, гнучкість	Складність інтеграції та управління
Громадська хмара (Community cloud)	Група організацій	Спільна або змішана інфраструктура	Спільне використання ресурсів, відповідність спільним вимогам	Обмежена універсальність, залежність від спільних правил

Класифікація моделей обслуговування хмарних обчислень

Модель	Розшифрування	Короткий опис	Основні функції / особливості
SaaS	Software as a Service (Програмне забезпечення як послуга)	Надання готових програмних додатків через Інтернет	Робота через браузер, орендна модель оплати, відсутність потреби в інсталяції та обслуговуванні ПЗ користувачем
PaaS	Platform as a Service (Платформа як послуга)	Надання платформи для розробки, тестування та розгортання додатків	Доступ до ОС, БД, інструментів розробки; керування ресурсами здійснює провайдер; гнучке масштабування
IaaS	Infrastructure as a Service (Інфраструктура як послуга)	Надання базових обчислювальних ресурсів	Віртуальні сервери, мережі, сховища; користувач керує ОС і ПЗ, але не фізичною інфраструктурою
DaaS	Data as a Service (Дані як послуга)	Надання доступу до даних через хмару	Централізоване зберігання даних, доступ через API або сервіси
WaaS	Workplace as a Service (Робоче місце як послуга)	Віртуальне робоче середовище для користувача	Віддалені робочі столи, централізоване управління робочими місцями
AaaS	All as a Service (Усе як послуга)	Комплексна модель надання всіх IT-ресурсів як сервісу	Поєднання різних хмарних сервісів у єдину інфраструктуру

Критерії вибору серверного обладнання та ПЗ

№	Критерій	Характеристика	Що враховується
1	Продуктивність	Забезпечує швидку обробку запитів та стабільну роботу системи	Багатоядерні процесори, обсяг ОЗП, SSD-накопичувачі
2	Масштабованість	Можливість розширення інфраструктури відповідно до навантаження	Додавання серверів, модернізація компонентів, блейд-системи
3	Відмовостійкість та надійність	Забезпечення безперервної роботи сервісів	Резервування живлення, охолодження, мережевих інтерфейсів
4	Продуктивність мережі	Швидкість передачі даних та мінімальні затримки	Високошвидкісні інтерфейси, пропускна здатність, низька латентність
5	Сумісність з ПЗ віртуалізації	Стабільна робота віртуального середовища	Підтримка VMware ESXi, Microsoft Hyper-V, сертифіковане обладнання
6	Керованість та моніторинг	Спрощення адміністрування системи	Централізовані системи управління, моніторинг, автоматизація
7	Безпека	Захист даних та доступу до системи	Шифрування, MFA, контроль доступу, secure boot
8	Інтеграція та автоматизація	Взаємодія з іншими системами	API (REST, Redfish), інтеграція з Active Directory
9	Енергоефективність	Зменшення витрат на експлуатацію	Енергоощадні компоненти, оптимізація ресурсів
10	Сервісна підтримка	Швидке вирішення технічних проблем	Технічна підтримка 24/7, оновлення ПЗ, заміна компонентів

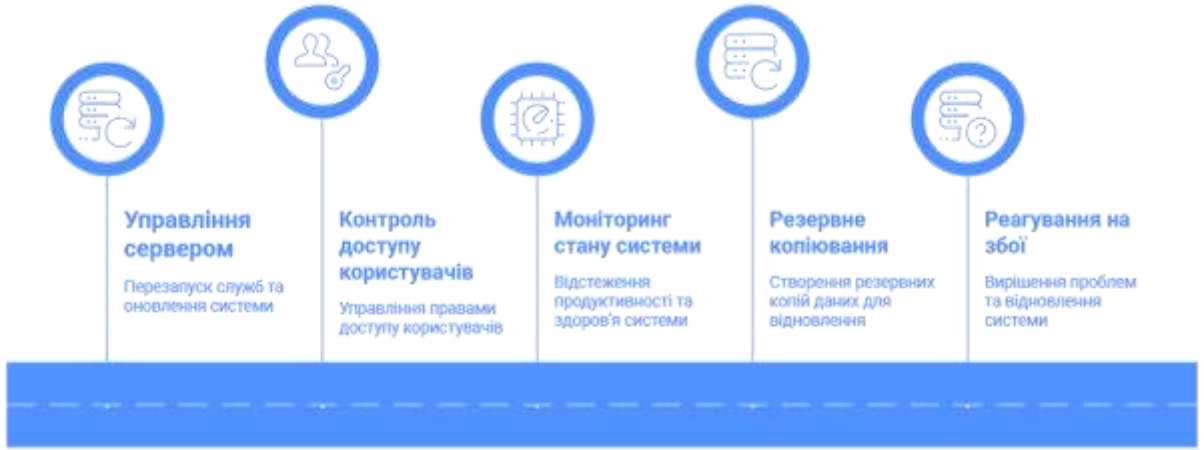
Технічні характеристики серверного обладнання

Компонент	Характеристика
Процесор (CPU)	4–8 ядер (наприклад, Intel Xeon E-23xx або AMD EPYC 7002/7003), частота від 2.5 ГГц
Оперативна пам’ять (RAM)	16–32 ГБ DDR4 ECC
Накопичувач (SSD)	від 500 ГБ NVMe SSD (рекомендовано 2×500 ГБ у RAID 1)
Дискова підсистема	RAID 1 (дзеркалювання) для підвищення надійності
Мережева карта	1 Gbps Ethernet (або 10 Gbps для перспективи масштабування)
Пропускна здатність	від 100 Mbps гарантованого каналу
Операційна система	Linux (Ubuntu Server 22.04 LTS / Debian 12)
Веб-сервер	Nginx (або Apache як альтернатива)
Сервер застосунків	PHP-FPM (версія 8.x) або Node.js
Система управління БД	MySQL / MariaDB або PostgreSQL
Віртуалізація	Docker (контейнеризація) або Proxmox / VMware (за потреби)
Система резервного копіювання	Автоматизовані backup (cron + rsync / mysqldump)
Система моніторингу	Netdata або Prometheus + Grafana
Безпека	Firewall (UFW), Fail2Ban, SSL (Let’s Encrypt), SSH-доступ за ключами
Джерело живлення	Надлишковий блок живлення (Redundant PSU) (для фізичного сервера)
Форм-фактор	Rack-сервер (1U/2U) або віртуальний сервер (VPS/VDS)

Трирівнева архітектура сервера

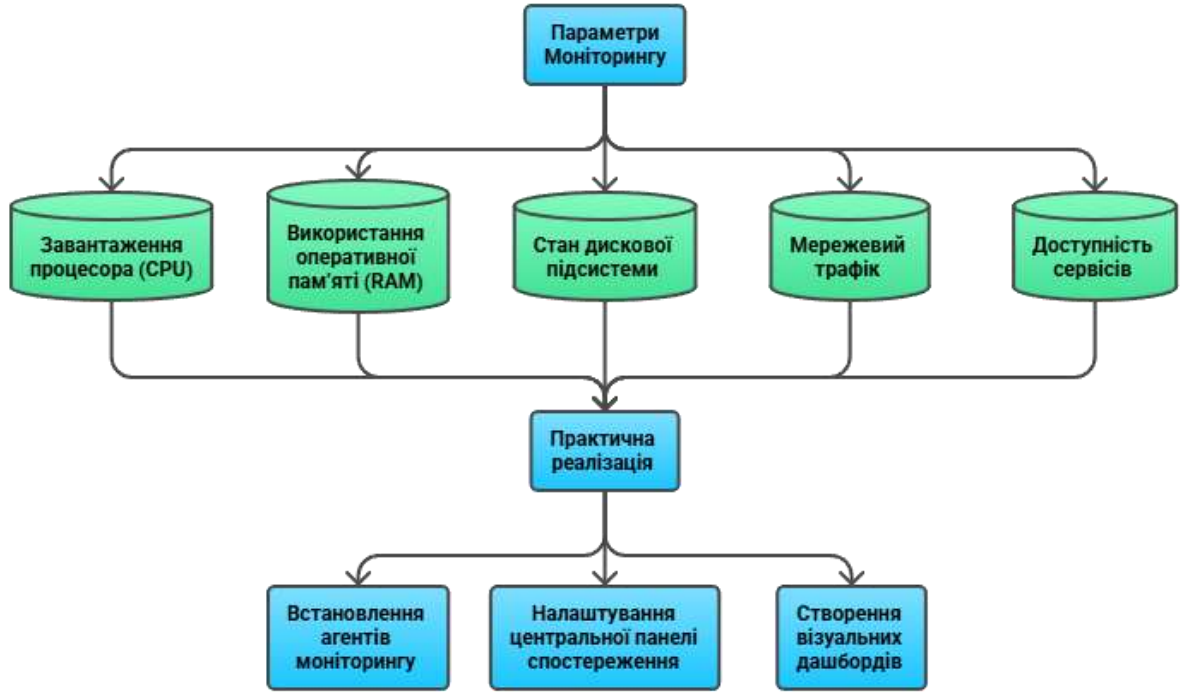


Задачі системи адміністрування





Конкретні засоби адміністрування



Структура забезпечення контролю за реалізацією моніторингу серверів і мережі

ВИСНОВКИ

У ході виконання роботи було встановлено, що ефективне функціонування веб-ресурсів неможливе без надійного серверного середовища та правильно організованого системного адміністрування. Адміністрування серверів розглядається як комплексний процес, що поєднує технічні, організаційні та безпекові аспекти, а його основними принципами є забезпечення безпеки, моніторинг, резервне копіювання, автоматизація та контроль доступу.

У процесі дослідження було проаналізовано теоретичні основи мережевої інфраструктури веб-сайтів та моделей системи адміністрування серверів, визначено їхні особливості, переваги та сфери застосування. Розглянуто сучасні серверні технології, зокрема віртуалізацію, контейнеризацію та хмарні обчислення, а також мережеві протоколи, що забезпечують стабільність і надійність передачі даних. Окрему увагу приділено критеріям вибору серверного обладнання та програмного забезпечення, що дозволяє сформувати ефективну, масштабовану та безпечну інфраструктуру.

У практичній частині роботи розроблено підхід до розгортання серверного середовища веб-сайту з урахуванням реальних умов експлуатації. Визначено основні принципи побудови серверної інфраструктури, обґрунтовано вибір архітектури, програмного забезпечення та засобів віртуалізації. Розроблено та описано етапи впровадження системи адміністрування, включаючи налаштування доступу, автоматизацію процесів, організацію резервного копіювання та використання сучасних інструментів управління.

ПРОДОВЖЕННЯ ВИСНОВКІВ

Також у роботі реалізовано підхід до моніторингу, безпеки та підтримки серверів і мережі. Запропоновано використання систем моніторингу, засобів захисту та механізмів реагування на інциденти, що дозволяє забезпечити безперервну та стабільну роботу веб-сайту. Визначено заходи щодо технічного обслуговування та підтримки інфраструктури, які мінімізують ризики збоїв і підвищують ефективність роботи системи.

Отже, у результаті виконання роботи досліджено теоретичні та практичні аспекти побудови мережевої інфраструктури веб-сайтів, розроблено підхід до розгортання та адміністрування серверного середовища, а також обґрунтовано використання сучасних технологій і засобів управління. Запропоновані рішення забезпечують створення надійної, безпечної та масштабованої інфраструктури.

Перспективи подальшого розвитку полягають у вдосконаленні підходів до адміністрування шляхом ширшого впровадження хмарних технологій, контейнеризації та автоматизації процесів, а також у розвитку систем моніторингу з елементами прогнозування та підвищенні рівня кібербезпеки, що дозволить створювати більш гнучкі, ефективні та адаптивні серверні інфраструктури відповідно до сучасних вимог інформаційного середовища.

РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ

1.