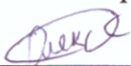


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

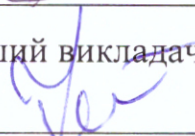
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра
на тему: «Методи та засоби захисту та оптимізації
вебстеку у віртуалізованому середовищі»

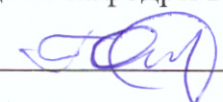
Виконав студент 2-го курсу групи КБм-24-1
спеціальності 125 «Кібербезпека та захист інформації»

 Олександр ХІМІЧУК

Керівник: Старший викладач кафедри КІ та КБ,

 Ігор ФАЛЬКОВСЬКИЙ

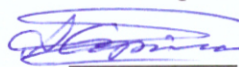
Керівник: Доцент кафедри КІ та КБ, к.пед.н., доцент,

 Олена ГОЛОВНЯ

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека та захист інформації»
Освітня програма «Кібербезпека»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій ЄФІМЕНКО

В жовтні 2025р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу магістра

Студент: Хімічук Олександр Вікторович

Тема роботи: «Методи та засоби захисту та оптимізації вебстеку у
віртуалізованому середовищі»

затверджена Наказом університету від “06” жовтня 2025р. № 566/с

Термін здачі студентом закінченої роботи “ 24 ” жовтня 2025р.

Вихідні дані роботи: Предметом дослідження є інтегрований процес підвищення стійкості, безпеки та високої доступності багатокомпонентного вебстеку. Об'єктом дослідження є віртуалізована серверна інфраструктура, а також інструменти, технології та методи конфігурації, моніторингу та управління вебдодатками на платформі Microsoft.

Обладнання та програмне забезпечення:

1. Microsoft Hyper-V 10
2. Windows Server 2022 Standart
3. MS SQL Server 2022 Developer
4. Umbraco 13

5. Zabbix 7.4.5
6. Wazuh 4.14.1
7. Apache JMeter 5.6.3

Консультанти розділів проекту (роботи)

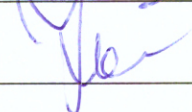
Розділ	Консультант	Підпис, дата	
		Завдання видав	завдання прийняв
1	Фальковський І. Г.	30.01.2025	30.01.2025
2	Фальковський І. Г.	25.11.2025	25.11.2025
3	Фальковський І. Г.	25.11.2025	25.11.2025

Студент Олександр ХІМІЧУК

Керівник Ігор ФАЛЬКОВСЬКИЙ

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів проекту	Примітки
1	Аналіз отриманого завдання та уточнення поставлених завдань	14.01.2025-31.01.2025 р.	виконано
2	Аналіз літературних джерел	14.01.2025-31.01.2025 р.	виконано
3	Складання плану проекту	14.01.2025-31.01.2025 р.	виконано
4	Аналіз об'єкта дослідження	01.09.2025-29.09.2025 р.	виконано
5	Проектування	30.09.2025-30.10.2025 р.	виконано
6	Налаштування	30.10.2025-10.12.2025 р.	виконано
7	Тестування	10.12.2025-12.12.2025 р.	виконано
7	Оформлення пояснювальної записки	14.01.2025-14.12.2025 р.	виконано
8	Оформлення презентації	15.12.2025-17.12.2025 р.	виконано
9	Проходження етапів підготовки до захисту	15.12.2025-19.12.2025 р.	виконано
10	Захист	26.12.2025 р.	виконано

Студент  Олександр ХІМІЧУК
Керівник  Ігор ФАЛЬКОВСЬКИЙ

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему «Методи та засоби захисту та оптимізації вебстеку у віртуалізованому середовищі» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Обсяг основного тексту звіту складає 58 сторінок при загальному обсязі роботи 75 сторінок. Список використаних джерел містить 20 найменувань і займає 2 сторінки. В роботі наведено 56 рисунків та 1 таблицю.

Метою є розробка та реалізація віртуалізованої інфраструктури вебстеку, яка забезпечує високу доступність, відмовостійкість на рівні бази даних та додатку, а також інтегрує засоби активного моніторингу та захисту.

У ході виконання роботи було обґрунтовано вибір методів та засобів оптимізації, моніторингу та захисту системи, розглянуто переваги використання Hyper-V для розгортання кластерних рішень Microsoft, розроблено трирівневу архітектуру вебстека з Active Directory, балансуванням NLB та автоматичним Failover. У проекті реалізовано горизонтальне масштабування системи та захист каналу зв'язку через HTTPS та підтверджено життєздатність інфраструктури результатами тестування.

Ключові слова: ВЕБСТЕК, ВІРТУАЛІЗАЦІЯ, ЗАХИСТ, ОПТИМІЗАЦІЯ, МОНІТОРИНГ.

ANNOTATION

The explanatory note to the qualification work on the topic "Methods and means of protecting and optimizing the web stack in a virtualized environment" consists of a list of conventional abbreviations, an introduction, three sections, conclusions, a list of used literature and appendices.

The volume of the main text of the report is 58 pages, with a total volume of work of 75 pages. The list of used sources contains 20 items and occupies 2 pages. The work contains 56 figures and 1 table.

The goal is to develop and implement a virtualized web stack infrastructure that provides high availability, fault tolerance at the database and application levels, and also integrates active monitoring and protection tools.

During the work, the choice of methods and means of optimizing, monitoring and protecting the system was justified, the advantages of using Hyper-V for deploying Microsoft cluster solutions were considered, a three-tier web stack architecture with Active Directory, NLB balancing and automatic Failover was developed. The project implemented horizontal scaling of the system and protection of the communication channel via HTTPS, and confirmed the viability of the infrastructure by testing results.

Keywords: WEBSTACK, VIRTUALIZATION, PROTECTION, OPTIMIZATION, MONITORING.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. Теоретичні відомості та основи побудови високодоступних віртуалізованих вебстеків.....	12
1.1. Вступ до проблематики розгортання вебстеків	12
1.1.1. Визначення поняття вебстека та його компонентів	12
1.1.2. Значення ефективності розгортання вебстеків у сучасних умовах ...	13
1.1.3. Вплив віртуалізації, оптимізації та моніторингу на продуктивність	15
1.2. Аналіз основних платформ для розгортання вебстеків	16
1.3. Сучасні методи віртуалізації і контейнеризації для вебстеків та їх порівняння.....	17
1.4. Оптимізація платформ для розгортання вебстеків	19
1.5. Огляд засобів захисту працездатності вебстека	23
1.6. Огляд інструментів моніторингу вебстеків.....	25
1.7. Постановка завдання дослідження	30
Висновки до розділу 1	31
РОЗДІЛ 2. Проектування та налаштування оптимізованої та захищеної інфраструктури вебстека	33
2.1. Архітектура віртуалізованого середовища розгортання.....	33
2.2. Налаштування систем моніторингу	35
2.3. Налаштування балансування навантаження NLB.	40
2.4. Налаштування WSFC та Always On Availability Groups	45
2.5. Налаштування захищеного з'єднання HTTPS за допомогою SSL-сертифіката.....	54
Висновки до розділу 2	58
РОЗДІЛ 3. Експериментальне дослідження та оцінка ефективності.....	58

3.1. Тестування розподілу навантаження.	60
3.2. Тестування відмовостійкості рівня даних.	63
Висновки з розділу 3.....	65
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IIS — Internet Informational Services

CMS — Content Management System

VM — Virtual Machine

ОС — Операційна Система

БД — База Даних

WSFC — Windows Server Failover Cluster

HA — High Availability

AGs — Availability Groups

AD – Active Directory

NLB — Network Load Balancing

ARR — Application Request Routing

CDN — Content Delivery Network

HTTP — HyperText Transfer Protocol

DNS — Domain Name System

SIEM — Security Information and Event Management

ВСТУП

Актуальність теми. У сучасному цифровому середовищі критично важливі вебдодатки вимагають безперервної доступності та нульової втрати даних. Будь-який збій або незапланований простій призводить до прямих фінансових збитків, втрати довіри клієнтів та репутаційних ризиків. Актуальність обраної теми обумовлена необхідністю розробки та впровадження комплексних, інтегрованих рішень, що поєднують механізми високої доступності на рівні бази даних та додатку з ефективними засобами моніторингу та захисту від сучасних кіберзагроз, що є фундаментальною вимогою до будь-якої комерційної IT-інфраструктури.

Метою кваліфікаційної роботи є розробка та реалізація віртуалізованої інфраструктури вебстеку, яка забезпечує високу доступність, відмовостійкість на рівні бази даних та додатку, а також інтегрує засоби активного моніторингу та захисту.

Об'єктом дослідження є віртуалізована серверна інфраструктура, а також інструменти, технології та методи конфігурації, моніторингу та управління вебдодатками на платформі Microsoft.

Предметом дослідження є інтегрований процес підвищення стійкості, безпеки та високої доступності багатокomпонентного вебстеку.

Визначена мета обумовлює наступні завдання:

- Розгорнути доменне середовище та базову віртуалізовану інфраструктуру для розміщення вебстеку.
- Оптимізувати роботу вебсервера
- Впровадити механізми захисту функціонування інфраструктури
- Налаштувати засоби моніторингу для постійного контролю стану системи.

У процесі виконання кваліфікаційної роботи було розроблено проект відмовостійкої інфраструктури для вебплатформи Umbraco CMS на базі стека технологій Microsoft. Проєкт реалізує багаторівневий захист через впровадження кластерів балансування навантаження NLB та груп доступності SQL Server Always On у віртуалізованому середовищі Hyper-V. Створена система забезпечує

автоматичне відновлення після збоїв, горизонтальне масштабування ресурсів та комплексний моніторинг стану вузлів.

РОЗДІЛ 1. Теоретичні відомості та основи побудови високодоступних віртуалізованих вебстеків

1.1. Вступ до проблематики розгортання вебстеків

1.1.1. Визначення поняття вебстека та його компонентів

Вебстек (web stack) — це сукупність технологій, інструментів та програмного забезпечення, які використовуються для створення, розгортання, обслуговування та управління вебдодатками. У загальному розумінні вебстек охоплює серверну, клієнтську частини та інфраструктуру, що забезпечує взаємодію між ними. Основною метою використання вебстека є забезпечення ефективного функціонування вебдодатка, включаючи швидкість, стабільність, масштабованість і безпеку.

Вебстек зазвичай включає чотири основні компоненти:

1. Операційна система
2. Вебсервер
3. Сервер додатків або бекенд
4. База даних

Операційна система — це основа, на якій працює все програмне забезпечення. Найпоширенішими операційними системами для серверів є Linux (наприклад, дистрибутиви Ubuntu, CentOS) і Windows Server. Операційна система забезпечує управління ресурсами, мережеву комунікацію та виконання процесів.

Вебсервер відповідає за прийом HTTP(S)-запитів від клієнтів і повернення відповідей, таких як вебсторінки або дані. Популярними вебсерверами є Apache HTTP Server, Nginx, Microsoft IIS.

Сервер додатків виконує бізнес-логіку вебдодатка. Це програмне середовище, де обробляються дані, виконуються алгоритми, і формується

кінцевий результат для користувача. Для розробки серверної частини часто використовують ASP.Net Core, Node.js, Django (Python), Spring Boot (Java).

База даних використовується для зберігання та управління інформацією, необхідною для роботи вебдодатка. Розрізняють два основних типи баз даних:

- Реляційні (SQL): MySQL, PostgreSQL, Microsoft SQL Server.
- Нереляційні (NoSQL): MongoDB, Redis, Cassandra.

Компоненти вебстека працюють у взаємозв'язку, забезпечуючи повноцінний цикл роботи вебдодатка. Клієнт надсилає запит через браузер, який проходить через вебсервер до серверу додатків. Сервер додатків обробляє дані, звертається до бази даних (за необхідності) і формує відповідь, яка повертається клієнту. Ефективність кожного компонента і їхнє налаштування впливають на загальну продуктивність системи.

Розуміння структури та принципів роботи вебстека є ключовим для розробників і адміністраторів, оскільки це дозволяє ефективно створювати, оптимізувати та підтримувати сучасні вебдодатки. Правильний вибір і конфігурація компонентів забезпечують масштабованість, безпеку та високу продуктивність системи [\[1\]](#).

1.1.2. Значення ефективності розгортання вебстеків у сучасних умовах

Ефективне розгортання вебстеків відіграє вирішальну роль у сучасних умовах стрімкого розвитку інформаційних технологій і цифрової економіки. Зростаючий обсяг даних, вимоги до швидкості та стабільності вебдодатків, а також конкуренція на ринку диктують необхідність оптимального підходу до розгортання й підтримки вебстеків.

Ключові аспекти значення ефективності:

1. Підвищення продуктивності: Оптимальне розгортання вебстека дозволяє досягти максимальної швидкодії вебдодатків, забезпечуючи швидке

завантаження сторінок і обробку запитів. Це важливо для користувацького досвіду, особливо в умовах глобальної конкуренції.

2. Скорочення витрат: Ефективність у розгортанні сприяє раціональному використанню ресурсів, зокрема серверів, енергоспоживання та витрат на підтримку. Це особливо актуально для компаній, які використовують хмарні технології, де ресурси оплачуються за фактом споживання.
3. Масштабованість: У сучасному світі вебдодатки повинні легко адаптуватися до зростання навантаження. Ефективно налаштований вебстек дозволяє масштабувати інфраструктуру з мінімальними зусиллями, забезпечуючи стабільну роботу навіть під час пікових навантажень.
4. Безпека: Оптимізація процесів розгортання включає впровадження сучасних механізмів безпеки, таких як шифрування, контроль доступу та регулярні оновлення компонентів. Це знижує ризики кібератак і втрати даних.
5. Швидкість виведення на ринок: Ефективність у розгортанні вебстека дозволяє скоротити час на запуск нових продуктів і послуг. Це особливо важливо для стартапів і компаній, які прагнуть швидко реагувати на змінні потреби ринку.

Ефективне розгортання вебстеків безпосередньо впливає на успіх компаній у цифровому середовищі. Підприємства, які забезпечують надійну й швидку роботу своїх вебдодатків, отримують конкурентні переваги, залучають більше клієнтів і підвищують свою репутацію.

У сучасних умовах ефективність розгортання вебстеків є важливим фактором успішної роботи як окремих компаній, так і цілих галузей. Інтеграція передових технологій, оптимізація процесів і забезпечення безпеки дозволяють не лише покращити технічні показники, але й зробити бізнес більш стійким і конкурентоспроможним.

1.1.3. Вплив віртуалізації, оптимізації та моніторингу на продуктивність

Віртуалізація, оптимізація та моніторинг є ключовими складовими забезпечення високої продуктивності сучасних вебстеків. Кожен із цих елементів відіграє специфічну роль у забезпеченні стабільної роботи системи та її здатності адаптуватися до змінних умов експлуатації.

Віртуалізація дозволяє ізолювати обчислювальні ресурси та розділяти їх між різними сервісами чи додатками. Це забезпечує гнучкість у розгортанні вебстеків, покращує використання апаратних ресурсів і підвищує рівень безпеки. Завдяки віртуалізації адміністратори можуть швидко створювати або масштабувати середовища для розробки, тестування чи продакшн-експлуатації, що зменшує час на впровадження нових функцій.

Оптимізація стосується як апаратного, так і програмного забезпечення. Налаштування серверних параметрів, таких як кешування, стиснення даних і оптимізація запитів до бази даних, дозволяють значно зменшити затримки та покращити загальну продуктивність системи. Наприклад, оптимізація алгоритмів маршрутизації запитів у вебсервері чи використання більш ефективних форматів стиснення дозволяють досягти швидшого завантаження сторінок для кінцевого користувача.

Моніторинг забезпечує можливість постійного спостереження за станом вебстеку та оперативного виявлення проблем. Завдяки інструментам моніторингу можна відстежувати ключові метрики продуктивності, зокрема використання ресурсів, швидкість обробки запитів та час простою. Аналіз отриманих даних дозволяє виявити потенційні вузькі місця у системі та вжити заходів для їх усунення до того, як вони стануть критичними.

Моніторинг також сприяє підтримці високого рівня безпеки, оскільки дозволяє виявляти підозрілу активність, як-от спроби несанкціонованого доступу чи DDoS-атаки. Завдяки інтеграції з сучасними системами оповіщення

адміністратори можуть оперативно реагувати на будь-які інциденти, що мінімізує ризик тривалих простоїв та збоїв у роботі вебстеку.

Загалом, поєднання віртуалізації, оптимізації та моніторингу дозволяє не лише підвищити продуктивність вебстеку, а й забезпечити його надійність, масштабованість та безпеку, що є критично важливим у сучасних умовах високої конкуренції та швидкого розвитку технологій [\[2\]\[3\]\[4\]](#).

1.2. Аналіз основних платформ для розгортання вебстеків

У сучасній веб-розробці домінують кілька архітектурних рішень, які різняться за принципами роботи, продуктивністю та областю застосування. Розглянемо три ключові стеки, що демонструють різні підходи до розгортання та забезпечення стійкості.

IIS-ASP.Net(Core) (IIS, ASP.NET Core, SQL Server)

Стек на базі Microsoft Windows (використовується у цьому проєкті) складається з Windows Server, вебсервера Internet Information Services (IIS), платформи ASP.NET Core (.NET) та бази даних SQL Server.

Ключова перевага цього стека полягає в нативній інтеграції з інструментами корпоративного рівня, такими як Active Directory для безпеки та Windows Server Failover Cluster (WSFC) для відмовостійкості. SQL Server є галузевим стандартом для транзакційної обробки даних (OLTP), гарантуючи високу цілісність [\[5\]\[6\]](#).

LAMP (Linux, Apache, MySQL, PHP/Python/Perl)

Стек LAMP є класичною архітектурою з відкритим вихідним кодом (Open Source), що складається з операційної системи Linux, вебсервера Apache HTTP Server, бази даних MySQL/MariaDB та мови програмування PHP, Python або Perl.

Основними перевагами цього стека є відсутність ліцензійних витрат та велика, зріла спільнота, що забезпечує гнучкість та швидкість розробки. Однак, забезпечення високої доступності та масштабування на рівні бази даних

(особливо для складних транзакцій) вимагає значних знань з адміністрування Linux. LAMP традиційно використовується для розміщення популярних CMS, таких як WordPress або Drupal, а також для невеликих і середніх вебсайтів. [7]

MERN / MEAN (MongoDB, Express.js, React/Angular, Node.js)

Стеки MERN / MEAN побудовані навколо платформи JavaScript та Node.js, що дозволяє використовувати єдину мову програмування на всіх рівнях (від фронтенду до бекенду). Завдяки неблокуючому вводу/виводу Node.js, ці стеки відомі своєю високою продуктивністю та ідеально підходять для створення динамічних додатків реального часу (API-сервісів) та односторінкових додатків (SPA). Використання бази даних NoSQL, як-от MongoDB, підвищує гнучкість, але може бути менш придатним для додатків, які вимагають жорсткої цілісності даних та складних транзакцій. Їхнє основне застосування — це високо навантажені API та сучасні вебсервіси.

Вибір стеку на базі Microsoft Windows, IIS та SQL Server є стратегічно обґрунтованим, оскільки він забезпечує найвищий рівень надійності та інтеграції для цілей високої доступності. Технології SQL Server Always On Availability Groups та WSFC є нативними компонентами Windows Server, що надає найкращу підтримку для створення кластерів з автоматичним перемиканням. Жоден інший стек не пропонує такої глибокої, зрілої та оптимізованої інтеграції для цілей НА. Цей стек є ідеальним вибором для корпоративних порталів та критично важливих CMS (наприклад, Umbraco), де потрібна максимальна надійність та відсутність втрати даних.

1.3. Сучасні методи віртуалізації і контейнеризації для вебстеків та їх порівняння

Сучасні технології віртуалізації та контейнеризації є ключовими елементами в управлінні та оптимізації вебстеків. Вони забезпечують ефективне використання апаратних ресурсів, спрощують процеси розгортання та управління вебдодатками. У цьому підрозділі розглядаються основні методи віртуалізації та контейнеризації, а також їх порівняння.

Контейнеризація забезпечує ізоляцію додатків разом із усіма їхніми залежностями, дозволяючи запускати їх у стандартизованих середовищах. Цей підхід є формою віртуалізації на рівні операційної системи, оскільки контейнери спільно використовують ядро хостової ОС, що забезпечує їхню швидкість та легкість.

Популярні Інструменти:

- Docker — надійна платформа для створення та управління контейнерами. Docker дозволяє розгорнути вебтек як набір мікросервісів, використовуючи легкі образи, зокрема для ASP.NET Core.
- Podman — альтернатива Docker, яка не потребує демонів (daemonless) і має покращену безпеку, оскільки контейнери запускаються безпосередньо від імені користувача.

Переваги: Легка портативність між різними середовищами; висока швидкість запуску контейнерів, що сприяє швидкому масштабуванню; просте оновлення й управління версіями компонентів вебстека [\[9\]](#).

Традиційна віртуалізація передбачає створення віртуальних машин (VM), які повністю емулюють апаратне забезпечення. Це забезпечує повну ізоляцію повноцінних операційних систем (ОС) за допомогою гіпервізора.

Гіпервізори промислового рівня:

- Hyper-V — гіпервізор типу 1 від Microsoft, вбудований у Windows Server, що є природним вибором для віртуалізації IIS-ASP.NET Core.
- VMware vSphere/ESXi — лідер ринку гіпервізорів для корпоративних середовищ, відомий своєю зрілістю та широким функціоналом управління.

- Oracle VM — гіпервізор, побудований на базі технології Xen, що часто використовується у корпоративних середовищах Oracle для віртуалізації серверів.

Переваги: Повна ізоляція повноцінних операційних систем; підтримка різних ОС на одному фізичному сервері; гнучкість у налаштуванні середовища [\[10\]](#).

У контексті забезпечення високої доступності (HA) корпоративного рівня та впровадження кластерних технологій Microsoft, використання традиційної віртуалізації (VM) є доцільнішим, ніж контейнеризація. По-перше, такі критично важливі компоненти, як Windows Server Failover Cluster (WSFC) та SQL Server Always On Availability Groups, вимагають повної ізоляції операційної системи, стабільної ідентичності машини в домені Active Directory та прямого доступу до віртуалізованого апаратного забезпечення, що найкраще забезпечується повноцінними віртуальними машинами. По-друге, контейнеризація вносить додатковий рівень складності у конфігурацію WSFC та забезпечення кворуму, тоді як віртуалізація надає необхідну апаратну емуляцію та повну ізоляцію ОС для досягнення максимальної відмовостійкості.

Вибір гіпервізора Hyper-V є найбільш обґрунтованим, оскільки це гіпервізор типу 1 від Microsoft, який є нативною частиною екосистеми Windows Server. Такий вибір забезпечує максимальну сумісність, стабільність та продуктивність для хостингу компонентів вебстеку на базі IIS/ASP.NET Core та SQL Server, мінімізуючи накладні витрати на інтеграцію та забезпечуючи глибоку підтримку кластерних функцій Microsoft.

1.4. Оптимізація платформ для розгортання вебстеків

Продуктивність вебстеку є ключовим фактором, що визначає швидкість, стабільність та загальну якість роботи вебзастосунків. Вона залежить від комплексного набору взаємопов'язаних архітектурних, апаратних та програмних факторів, які мають бути враховані під час проектування розгортання.

На рівні апаратної інфраструктури (у нашому випадку — віртуальної машини, VM) критичне значення мають обчислювальна потужність ЦП, що безпосередньо впливає на швидкість виконання коду, та обсяг оперативної пам'яті, необхідний для роботи IIS, ASP.NET Core та кешу. Швидкість зберігання даних є фундаментальною; використання SSD-накопичувачів замість традиційних HDD значно знижує час доступу до даних, особливо для бази даних. Крім того, мережа — її пропускна здатність і мінімізація затримок (*latency*) — є життєво важливою для швидкості обміну даними між серверами та клієнтами.

Архітектурні фактори стосуються структури та організації компонентів. Ефективність залежить від вибору серверного ПЗ, наприклад, IIS, його можливостей обробки запитів і підтримки сучасних протоколів (HTTP/2). Хоча мікросервісна архітектура (як альтернатива) сприяє розподіленому виконанню завдань, вона додає накладні витрати на взаємодію компонентів, що вимагає ретельного налаштування. Важливою є сумісність компонентів: узгодженість між сервером додатків (IIS), кодом (ASP.NET Core), базою даних (MS SQL Server) та кеш-системами впливає на загальну продуктивність.

Нарешті, програмні фактори включають оптимізацію коду (використання асинхронної обробки в ASP.NET Core) та ефективне застосування кешування на різних рівнях. Сучасні практики також вимагають впровадження віртуалізації (для ізоляції та HA-кластерів) або контейнеризації (для швидкого масштабування). Постійний моніторинг (Prometheus, Grafana) і аналіз логів є невід'ємними складовими, оскільки дозволяють вчасно ідентифікувати та усувати вузькі місця, забезпечуючи неперервну оптимізацію.

Ефективна робота вебстеків залежить від оптимального використання ресурсів сервера, що забезпечується шляхом впровадження методів кешування, балансування навантаження та інших підходів. У цьому підрозділі розглянуто ключові методи, які сприяють покращенню продуктивності та стабільності вебзастосунків.

Використання кешування

Кешування є фундаментальним методом оптимізації, що дозволяє зберігати часто використовувані дані у швидкодоступних місцях, зменшуючи навантаження на основні сервери та скорочуючи час відгуку.

Кешування на стороні сервера (Серверне та Розподілене) включає використання розподілених служб, таких як Redis або Memcached, для зберігання динамічних даних у пам'яті, що дозволяє уникнути повторних запитів до бази даних і прискорити час відгуку. Також застосовується Output Caching — увімкнення кешування динамічних і статичних відповідей на рівні IIS або самого ASP.NET Core-додатку.

Клієнтське Кешування досягається налаштуванням заголовків HTTP, таких як Cache-Control і ETag, для оптимізації кешування контенту на стороні браузера користувача [\[12\]](#).

Балансування навантаження

Балансування навантаження забезпечує рівномірний розподіл запитів між серверами, що дозволяє підвищити доступність і продуктивність системи.

1. Network Load Balancing (NLB) працює на рівні мережі і забезпечує розподіл вхідних запитів між кількома серверами. Цей метод балансування навантаження є універсальним і не потребує змін у конфігурації IIS, що робить його простим і зручним для впровадження. NLB інтегрується з операційною системою Windows і може використовуватися з різними веб-стеками, забезпечуючи гнучкість та масштабованість для різноманітних веб-додатків.

Однією з переваг NLB є те, що всі сервери можуть активно брати участь в одночасній обробці вхідних запитів протоколу передачі гіпертексту (HTTP). Ще одна перевага полягає в тому, що в середовищі NLB IIS може бути набагато легше підтримувати поточні оновлення та відкат, забезпечуючи високу доступність веб-додатків.

2. Application Request Routing (ARR) працює на рівні вебсервера і забезпечує розподіл вхідних HTTP-запитів між серверами на основі заголовків

HTTP, змінних сервера та алгоритмів балансування навантаження. Цей підхід вимагає змін конфігурації IIS, але надає більше можливостей для тонкого налаштування і контролю над трафіком. ARR дозволяє налаштовувати правила маршрутизації на основі різних параметрів запиту, таких як URL, заголовки HTTP або вміст запиту [\[13\]](#).

Інші Підходи до Масштабування та Відмовостійкості

Ефективне розгортання вимагає інтеграції низки додаткових технологій, що впливають на швидкість і надійність.

Оптимізація Баз Даних та HA Database: База даних (MS SQL Server) є критичним компонентом, який повинен мати власні механізми відмовостійкості. Використання Always On Availability Groups або Clustering гарантує, що у разі збою основного вузла БД, робота автоматично перемикається на резервний вузол. Додаткова оптимізація включає використання індексів, оптимізованих запитів та кешування запитів до бази даних.

Масштабування Системи: Розрізняють два основні типи масштабування. Вертикальне масштабування передбачає додавання ресурсів (ЦП, ОЗП) до існуючих серверів, тоді як горизонтальне масштабування — це додавання нових серверів до пулу, що є більш гнучким і кращим підходом для вебстеків.

CDN та Географічна Відмовостійкість (Multi-Region Architecture): Використання CDN для статичного контенту забезпечує його швидке завантаження та знижує навантаження на основні сервери. Для проєктів із найвищими вимогами до надійності застосовується Multi-Region Architecture — розгортання ідентичних вебстеків у географічно розподілених дата-центрах. Це забезпечується інтелектуальним DNS-балансуванням, що гарантує, що збій цілого регіону не призведе до зупинки роботи вебсайту.

Протоколи та Резервування: Увімкнення HTTP/2 для пришвидшення передачі даних через мультиплексування та забезпечення безпеки через використання HTTPS є стандартною вимогою. Нарешті, резервне копіювання та

відновлення (*Backup/Restore*) VM, конфігураційних файлів та БД є основою для швидкого відновлення після критичного збою, що визначається стратегіями RPO/RTO.

Використання методів кешування, балансування навантаження та інших підходів дозволяє значно підвищити ефективність вебзастосунків. Інтеграція сучасних інструментів і технологій є важливим етапом для забезпечення стабільності та продуктивності вебстеків у сучасних умовах.

1.5. Огляд засобів захисту працездатності вебстека

Захист інфраструктури є критично важливим елементом її стійкості. У цьому проєкті використано багат шаровий підхід до безпеки, що включає як нативні механізми операційної системи, так і спеціалізовану платформу для виявлення загроз та реагування на інциденти.

Брандмауер фільтрує мережевий трафік, дозволяючи або блокуючи пакети даних на основі встановлених правил. Ці правила зазвичай ґрунтуються на IP-адресах, номерах портів та протоколах.

Для вебстека брандмауер контролює доступ до критично важливих служб. Наприклад, він гарантує, що вебсервер IIS доступний лише через стандартні порти HTTP (80) і HTTPS (443), а доступ до сервера бази даних (SQL Server) відкритий лише для вебсерверів (зазвичай, порт 1433 або 5022 для реплікації) та адміністраторів. Це запобігає прямому впливу зовнішніх загроз на внутрішні компоненти, які забезпечують роботу системи.

У проєкті використовується Брандмауер Windows Server — міжмережевий екран на рівні хоста, який пропонує детальне управління трафіком для кожного окремого вузла інфраструктури.

Резервування та Кластери Високої Доступності (High Availability Clusters)

Концепція резервного сервера реалізується через різні технології резервування даних, що є ключовим для гарантування високої доступності (High

Availability, HA). Резервний сервер — це пасивна або активна копія основного сервера, що готова взяти на себе його функції у разі збою.

У сучасних умовах кластеризація є базовим принципом ефективного розгортання вебстеку. Кластери високої доступності (HA Clusters) — це об'єднання кількох фізичних або віртуальних серверів, які спільно працюють для забезпечення неперервної роботи застосунків.

Втрата однієї VM або фізичного вузла не призводить до простою вебсайту. Технології, такі як Hyper-V Failover Clustering або VMware High Availability (HA), автоматично переносять робоче навантаження на інший доступний вузол кластера. Це є основою для забезпечення SLA (Service Level Agreement) та надійності вебдодатків.

Реалізація вебстеку на віртуалізованому середовищі вимагає налаштування віртуальних машин у кластері (або використання кластерних рішень хмарного провайдера), що є необхідною умовою для High Availability та відмовостійкості вебдодатку.

Для забезпечення стійкості до збоїв та безперервності роботи критично важливих систем SQL Server пропонує кілька технологій резервування, які різняться за складністю, режимом синхронізації та можливостями автоматичного відновлення. Три основні підходи — це Логістичні журнали (Log Shipping), Дзеркальне відображення бази даних (Database Mirroring) та Групи доступності Always On (Always On Availability Groups).

Метод Log Shipping є найпростішим і найстарішим. Його принцип полягає у послідовному копіюванні та відновленні файлів журналів транзакцій з основного сервера на один або кілька вторинних серверів. Через природу операції відновлення, синхронізація відбувається асинхронно, що призводить до невеликої, але постійної затримки даних на вторинній копії (іноді до 30 хвилин). Цей метод не підтримує автоматичне перемикання (Failover). У разі збою основного сервера, адміністратор має вручну ініціювати процес переходу, що

робить його придатним переважно для аварійного відновлення, де невеликий час простою є прийнятним.

Database Mirroring (Дзеркальне відображення) було проміжним рішенням, що значно підвищило доступність порівняно з Log Shipping. Ця технологія встановлює прямий зв'язок між двома екземплярами: Головним (Principal) і Дзеркальним (Mirror). Вона підтримує як синхронний режим (без втрати даних), так і асинхронний режим. Ключова перевага полягала в тому, що за умови наявності третього екземпляра — Свідка (Witness) — Mirroring міг забезпечити автоматичне перемикавання (Failover) протягом кількох секунд. Однак, Mirroring має суттєве обмеження: він захищає лише одну базу даних за раз і може мати лише одну вторинну копію.

Always On Availability Groups (AG) — це сучасна, найбільш гнучка та масштабована технологія. На відміну від Mirroring, яке працює на рівні бази даних, AGs захищають групу баз даних одночасно і вимагають наявності Windows Server Failover Cluster (WSFC) для роботи. AGs підтримують до восьми вторинних реплік, які можуть бути як синхронними (для швидкого автоматичного перемикавання), так і асинхронними (для географічно віддаленого DR). Найважливішою перевагою є використання Прослуховувач (Listener) — єдиної віртуальної точки підключення — яка автоматично перенаправляє клієнтський трафік на активний вузол. Це робить збій повністю прозорим для додатка, який просто продовжує підключатися до тієї ж адреси, навіть коли активний вузол змінюється. Завдяки цьому AGs є стандартом для забезпечення максимальної доступності критично важливих систем [\[14\]](#).

1.6. Огляд інструментів моніторингу вебстеків

Ефективний моніторинг є ключовим елементом для забезпечення стабільної роботи вебстеків. Сучасні інструменти моніторингу дозволяють відслідковувати продуктивність, виявляти проблеми та оптимізувати роботу систем.

Для забезпечення цілісного контролю, моніторинг вебстеку IIS-ASP.Net(Core) поділяється на такі ключові класи:

1. Системний Моніторинг (System Monitoring): Фокусується на базових показниках здоров'я віртуальної машини (Windows Server): завантаження CPU, використання RAM, дискові операції (IOPS) та базове мережеве навантаження.
2. Мережевий Моніторинг (Network Monitoring): Контроль мережевого трафіку, затримки (latency) та втрат пакетів. Включає використання протоколів SNMP та NetFlow/sFlow для аналізу потоків даних і стану мережевого обладнання.
3. Моніторинг Застосунку та Вебсервера (APM/Web Server Monitoring): Глибокий аналіз продуктивності коду ASP.NET Core та метрик IIS (час відповіді, кількість активних запитів, стан пулів додатків).
4. Безпековий Моніторинг (Security Monitoring): Відстеження подій, пов'язаних із безпекою, включаючи спроби несанкціонованого доступу, аномалії в логах автентифікації/авторизації, зміни критичних конфігурацій та активність мережевого екрану. Часто реалізується через SIEM-системи (Security Information and Event Management) або спеціалізовані модулі логування.
5. Моніторинг Логів (Logging Monitoring): Централізований збір, індексація та аналіз лог-файлів з усіх компонентів (OC, IIS, ASP.NET Core).

Ефективне розгортання вимагає комбінації інструментів, кожен з яких спеціалізується на своєму типі даних (метрики, статус, логи), забезпечуючи повне покриття віртуалізованого вебстеку IIS-ASP.NET Core.

Zabbix

Zabbix є класичною універсальною платформою моніторингу, що працює за клієнт-серверною моделлю. Zabbix Server збирає дані від Zabbix Agents, які

встановлюються на віртуальну машину (Windows Server VM) для моніторингу її внутрішнього стану.

Він ідеально підходить для системного моніторингу, відстежуючи базові показники здоров'я VM, такі як завантаження CPU, використання RAM, дискові операції (IOPS) та базове мережеве навантаження. Zabbix також підтримує протокол SNMP для збору інформації з мережевого обладнання.

Його ключові переваги — це гнучкість (підтримка агентного та безагентного моніторингу) та висока масштабованість. Головний недолік — це високі вимоги до ресурсів Zabbix Server при роботі з великими історичними даними, а його інтерфейс може бути менш інтуїтивним порівняно з сучасними інструментами [\[15\]](#).

Prometheus

Prometheus — це сучасна система моніторингу, розроблена для збору часових рядів метрик. Вона використовує модель Pull, коли Prometheus Server періодично запитує дані (метрики) через HTTP-протокол у спеціальних Експортерів.

У контексті IIS-ASP.NET Core, Prometheus використовує Windows Exporter та IIS Exporter для отримання детальних метрик вебсервера. Він також легко інтегрується безпосередньо з кодом ASP.NET Core, що дозволяє відстежувати продуктивність коду та час виконання запитів (APM). Його потужна мова запитів PromQL дає змогу виконувати складний аналіз метрик у реальному часі.

Ключові переваги — висока точність і швидкість збору метрик, а також ідеальна придатність для моніторингу продуктивності застосунку. Основний недолік — обмежені можливості для довгострокового зберігання даних без інтеграції з додатковими рішеннями [\[16\]](#).

Nagios

Nagios є однією з найстаріших і найбільш перевірених систем моніторингу в ІТ-індустрії, яка спеціалізується на контролі доступності та статусу критично важливих сервісів.

На відміну від Prometheus, Nagios фокусується на моделі Push/Polling (запиту або отримання даних від агента). Він використовує плагіни для виконання перевірок на локальному сервері або віддалено (наприклад, через NRPE - Nagios Remote Plugin Executor). Його основна функція полягає у швидкому визначенні, чи є компонент "живим" і чи відповідає він базовим критеріям (наприклад, чи працює процес IIS, чи є вільне місце на диску).

Ідеально підходить для контролю доступності критичних елементів віртуалізованого вебстеку: перевірка, чи запущена VM, чи працює сервіс IIS, чи доступний MS SQL Server. Це важливий елемент системного моніторингу.

Головна перевага Nagios — його надійність, стабільність та низькі вимоги до ресурсів. Він дуже ефективний для перевірки статусу. Основний недолік — це складність конфігурації, яка часто виконується через текстові файли, а також слабка візуалізація порівняно з Grafana. Він є менш гнучким для збору та аналізу глибоких метрик продуктивності порівняно з Prometheus [\[17\]](#).

Grafana

Grafana — це платформа вебвізуалізації, що не збирає дані самостійно, а слугує інтерфейсом для відображення даних, отриманих з різних джерел (Data Sources).

Архітектура та Застосування: Вона є невід'ємним доповненням до Prometheus та Zabbix. Grafana інтегрується з ними та іншими джерелами (наприклад, MS SQL Server, Elasticsearch) для створення інтерактивних, налаштовуваних дашбордів. Це дозволяє створювати єдину панель, яка відображає всі аспекти розгортання: від здоров'я VM до продуктивності вебзастосунку.

Переваги та Недоліки: Головна перевага — краща в класі візуалізація та гнучкість у налаштуванні звітів і графіків. Grafana також має функціонал для налаштування сповіщень, доповнюючи функціонал джерел даних. Основний недолік — повна залежність від зовнішніх джерел даних [\[18\]](#).

ELK Stack, Graylog та Wazuh

Централізоване логування є необхідним для швидкої діагностики помилок та забезпечення безпекового моніторингу.

ELK Stack (Elasticsearch, Logstash, Kibana) є стандартом для роботи з логами. Logstash збирає, фільтрує та парсить лог-файли з усіх компонентів (IIS, ASP.NET Core, Windows Event Log). Elasticsearch індексує ці логи, забезпечуючи високу швидкість повнотекстового пошуку. Kibana надає інтерфейс для пошуку, аналізу та візуалізації логів.

Цей стек критично важливий для безпекового моніторингу, оскільки дозволяє аналізувати логи IIS на предмет спроб атак (SQL Injection, XSS) та відстежувати події безпеки Windows Server.

Graylog — це альтернативне рішення, яке також пропонує потужні можливості для збору та аналізу логів. Він часто вважається простішим у впровадженні, ніж ELK, для малих і середніх інфраструктур.

Wazuh дозволяє збирати та аналізувати журнали подій Windows (Event Logs) та IIS з усіх вузлів кластера в єдиному інтерфейсі. Це дає змогу виявляти складні патерни атак (наприклад, розподілені спроби підбору паролів), які важко помітити при ручному аналізі окремих серверів.

Важливою функцією, що аналізується, є здатність системи в реальному часі відстежувати зміни в критичних файлах вебдодатка Umbraco. Це дозволить миттєво виявляти несанкціоноване втручання в код або конфігураційні файли системи.

Розглядається можливість використання Wazuh для автоматичного аудиту серверів на відповідність кращим практикам безпеки (hardening) та пошуку відомих вразливостей у встановленому ПЗ (SQL Server, IIS). Wazuh у даній архітектурі виконує функції HIDS та частково SIEM-системи, забезпечуючи централізований збір, кореляцію та аналіз подій безпеки. [\[19\]](#).

Інтеграція Wazuh розглядається як логічне доповнення до Zabbix. У той час як Zabbix фокусується на доступності та ресурсах, Wazuh забезпечує глибоку видимість подій безпеки, що є критичним для захисту корпоративних даних та підтримання цілісності вебстеку в умовах сучасних кіберзагроз.

Спеціалізовані Протоколи (SNMP, NetFlow/sFlow)

SNMP (Simple Network Management Protocol) — це класичний протокол, що використовується для мережевого моніторингу. Він дозволяє інструментам, таким як Zabbix, віддалено збирати статус та метрики мережевих пристроїв і самої VM Windows Server щодо використання мережевих інтерфейсів та основних ресурсів.

NetFlow/sFlow — це технології для аналізу потоків трафіку. Вони не просто вимірюють обсяг даних, а фіксують метадані про комунікацію (хто, з ким, коли, який додаток). Це є критичним елементом безпекового моніторингу, оскільки дозволяє виявляти мережеві аномалії, такі як DDoS-атаки або несанкціоновану передачу великих обсягів даних.

1.7. Постановка завдання дослідження

Підбиваючи підсумки аналізу теоретичних відомостей та огляду сучасних джерел, можна констатувати, що в умовах стрімкої цифровізації бізнес-процесів виникає гостра потреба у впровадженні комплексних методів забезпечення стійкості вебсистем. Теоретичний огляд показав, що фрагментарне використання засобів резервування не здатне забезпечити необхідний рівень доступності при виникненні каскадних збоїв інфраструктури.

Актуальність дослідження зумовлена необхідністю створення єдиної екосистеми, яка поєднує високу доступність на рівні вебдодатків, відмовостійкість на рівні сховищ даних (SQL Server Always On) та проактивний моніторинг стану системи. Такий підхід дозволяє мінімізувати час відновлення сервісів та гарантувати цілісність інформації в критичних умовах.

Метою роботи є розробка та практична реалізація віртуалізованої інфраструктури для вебплатформи Umbraco CMS, яка здатна функціонувати в режимі високої доступності та забезпечувати безперервність процесів при виході з ладу окремих вузлів.

Для досягнення поставленої мети визначено наступні завдання:

- Спроекувати мережеву топологію та розгорнути доменну інфраструктуру на базі Active Directory для централізованого управління доступом та конфігураціями вузлів.
- Імплементувати механізми балансування навантаження (NLB) для розподілу вхідного трафіку між вебсерверами, забезпечуючи горизонтальне масштабування та стійкість до відмов рівня додатків.
- Розгорнути кластерну систему SQL Server Always On Availability Groups, налаштувавши синхронну реплікацію даних та віртуальний прослуховувач (Listener) для автоматичного перемикання ролей без переривання роботи користувачів.
- Впровадити засоби системного моніторингу продуктивності для збору та аналізу метрик стану інфраструктури в реальному часі.

Висновки до розділу 1

У першому розділі виконано теоретичний аналіз сучасних підходів до побудови високонадійних та відмовостійких вебсистем, а також досліджено методи забезпечення високої доступності, масштабованості та безпеки серверної інфраструктури.

На основі проведеного аналізу обґрунтовано доцільність використання технологічного стека на базі платформи Microsoft для реалізації корпоративних вебрішень. Встановлено, що застосування середовища віртуалізації Hyper-V у поєднанні з доменною інфраструктурою Active Directory забезпечує високий рівень сумісності, керованості та інтеграції кластерних рішень у середовищі Windows Server.

Проаналізовано основні методи забезпечення відмовостійкості баз даних та визначено, що використання технології SQL Server Always On Availability Groups є одним із найбільш ефективних рішень для критично важливих вебсистем, оскільки вона забезпечує автоматичне перемикання між репліками та мінімізацію втрати даних у разі збоїв.

Також встановлено, що застосування механізму Network Load Balancing є ключовим чинником реалізації горизонтального масштабування вебсерверів IIS, що дозволяє рівномірно розподіляти навантаження та підвищувати загальну доступність сервісів.

Окрему увагу приділено питанням моніторингу та безпеки інфраструктури. Обґрунтовано доцільність поєднання системи моніторингу продуктивності Zabbix та засобів аналізу подій безпеки Wazuh для забезпечення комплексного контролю стану системи, своєчасного виявлення технічних збоїв і реагування на потенційні кіберзагрози.

Проведений теоретичний аналіз підтвердив актуальність обраної теми дослідження та сформував методологічну основу для практичної реалізації оптимізованої та захищеної відмовостійкої вебінфраструктури, що розглядається у наступних розділах роботи.

РОЗДІЛ 2. Проектування та налаштування оптимізованої та захищеної інфраструктури вебстека

2.1. Архітектура віртуалізованого середовища розгортання

Інфраструктура вебстека проєкту розгортається у віртуалізованому середовищі на базі гіпервізора Hyper-V.

Основна архітектура побудована на трирівневій моделі, що забезпечує чітке розділення відповідальності між рівнем представлення/додатків та рівнем даних.

Використання повноцінних віртуальних машин є необхідним для реалізації Windows Server Failover Cluster та SQL Server Always On Availability Groups, оскільки ці технології вимагають стабільної ідентифікації вузла та глибокої інтеграції з операційною системою Windows Server.

Кластер складається із двох ідентичних віртуальних машин, які виконують подвійну роль: вебсерверів та реплік бази даних.

Між двома серверами налаштовано Мережеве балансування навантаження (NLB). Принцип роботи NLB полягає у розподіленні вхідного клієнтського трафіку (HTTP/HTTPS) між двома серверами, тим самим оптимізуючи використання ресурсів та запобігаючи перевантаженню одного окремого вузла.

Вебдодаток Umbraco розгорнуто на обох серверах, але обидва екземпляри підключаються до єдиної віртуальної точки доступу (Listener).

Для забезпечення відмовостійкості даних використовується Windows Server Failover Cluster. WSFC забезпечує роботу функції SQL Server Always On Availability Groups. Ця функція створює синхронізовану копію бази даних (БД) на вторинному сервері (Secondary Replica) і автоматично перемикає роботу сайту на неї під час збою в роботі первинного сервера (Primary Replica).

Для реалізації автоматичного та надійного перемикавання додано ще один компонент інфраструктури — окрема VM (VM3 Windows Server 2022). Цей

сервер виконує роль Свідка файлового ресурсу (File Share Witness), за допомогою чого кластер вирішує, який сервер матиме роль Primary у разі порушення зв'язку між VM1 та VM2.

За сумісництвом, на цьому сервері розгорнуто домен Active Directory, який вирішує проблему довіри та забезпечує централізовану автентифікацію для всіх вузлів, що є обов'язковою вимогою для роботи WSFC.

Запропонована архітектура забезпечує як високий рівень доступності (High Availability), так і елементи відмовостійкості (Fault Tolerance) на рівні вебдодатка та бази даних.

У проєкт впроваджено системи системного, мережного моніторингу та логування для інформування про роботоздатність системи та події, пов'язані з роботою інфраструктури.

Система моніторингу продуктивності збирає метрики про навантаження на центральний процесор, оперативну пам'ять, файлову систему, швидкість запитів до IIS, використання кешу, мережевий трафік, тощо. Це дозволяє проактивно виявляти вузькі місця та оптимізувати ресурси.

Система виявлення вторгнень на хості (HIDS) та управління безпекою (SIEM) Wazuh збирає та аналізує логи подій Windows та IIS, що є критично важливим для виявлення спроб несанкціонованого доступу та моніторингу цілісності файлів.

На етапі проєктування інфраструктури було виконано якісне моделювання загроз (threat modeling), у межах якого розглядалися основні вектори атак: компрометація вебдодатка, несанкціонований доступ до БД, відмова в обслуговуванні та порушення цілісності даних.

Замість повноцінного penetration testing у роботі застосовано підхід керованої симуляції загроз, що дозволяє оцінити здатність системи виявляти аномальні події.

Усі пристрої мережі з'єднуються між собою через віртуальний комутатор гіпервізора Hyper-V із доступом до інтернету, створюючи ізольоване, але функціонально повне корпоративне середовище.

Детальна схема архітектури інфраструктури та таблиця адресації наведені в додатку А та Б.

2.2. Налаштування систем моніторингу

Налаштування проекту варто почати із завантаження і додавання у гіпервізор необхідних віртуальних машин. У даному проекті використовується 5 віртуальних машин. AD_DC_Controller — це ВМ зі встановленою ОС Windows Server 2022, яка буде виконувати роль контролера домену. IIS-WebStack та IIS-WebStack2 — віртуальні машини із завчасно налаштованим вебстеком IIS-ASP.net(Core). Wazuh і Zabbix — виконують ролі серверів моніторингу, вони будуть збирати необхідні дані про роботу перших трьох віртуальних машин.

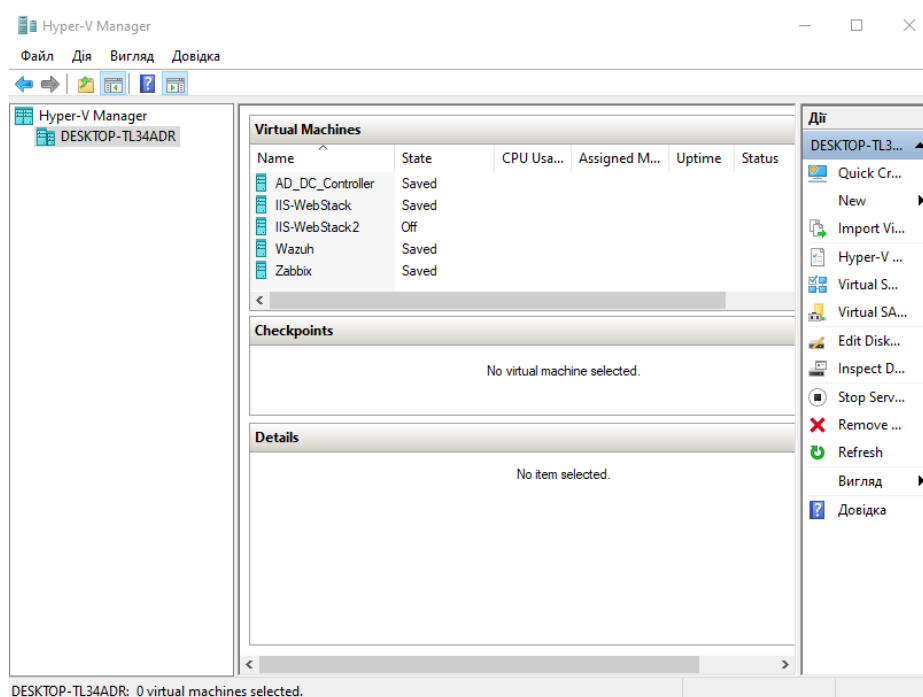


Рис. 2.1. Вікно Hyper-V із відображенням доданих VM

Для зв'язку пристроїв між собою і доступу до інтернету у гіпервізорі Hyper-V потрібно створити віртуальний комутатор із відповідним типом з'єднання.

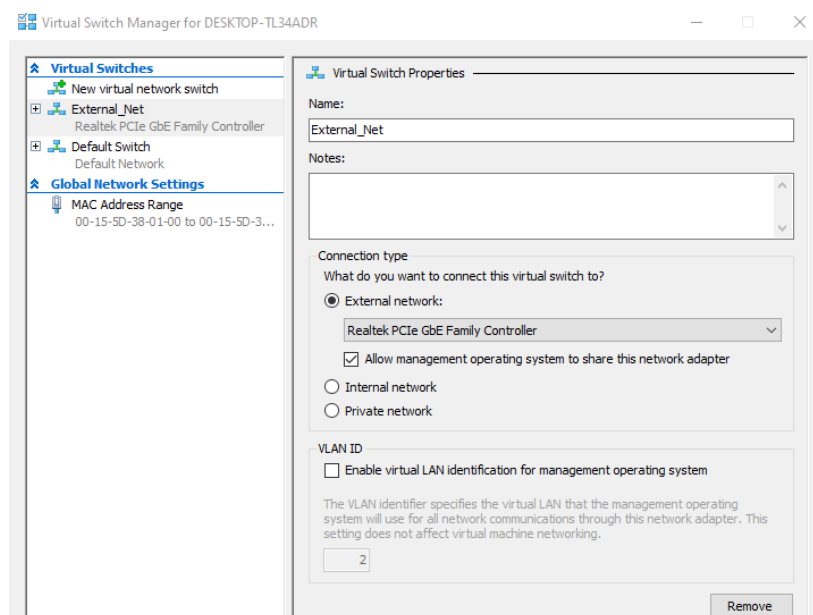


Рис. 2.2. Створення і налаштування віртуального комутатора

Налаштування моніторингу на хоста починається із встановлення на ньому агента моніторингу. Під час встановлення Zabbix Agent необхідно ввести лише кілька значень: назву хоста, який буде моніторитись, IP-адреса сервера Zabbix, порт, за яким буде здійснюватися підключення та IP-адреса для активних перевірок (вона співпадає із адресою сервера Zabbix).

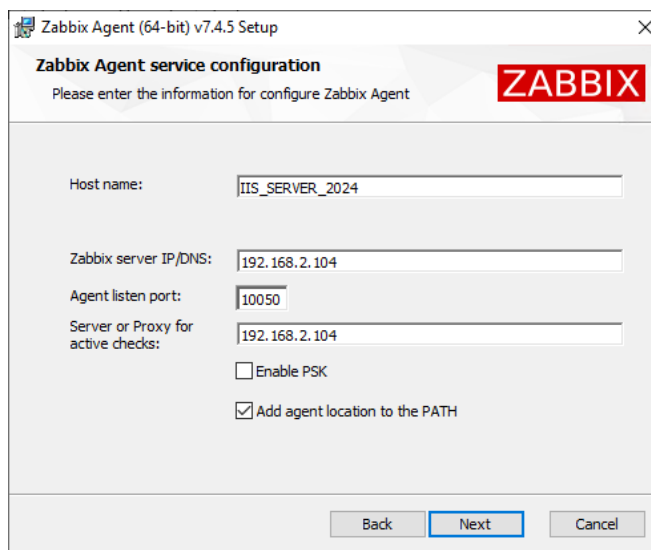


Рис. 2.3. Налаштування агента Zabbix

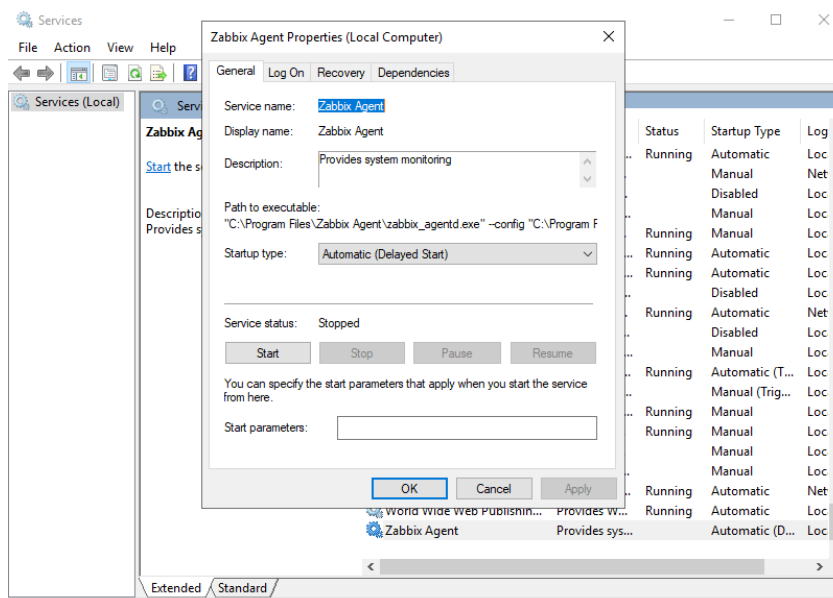


Рис. 2.4. Увімкнення служби Zabbix Agent

Щоб брандмауер Windows не блокував з'єднання Zabbix Agent із Zabbix Server, важливо додати правило для вхідних/вихідних з'єднань для порту 10050, який використовується Zabbix для спілкування за замовчуванням.

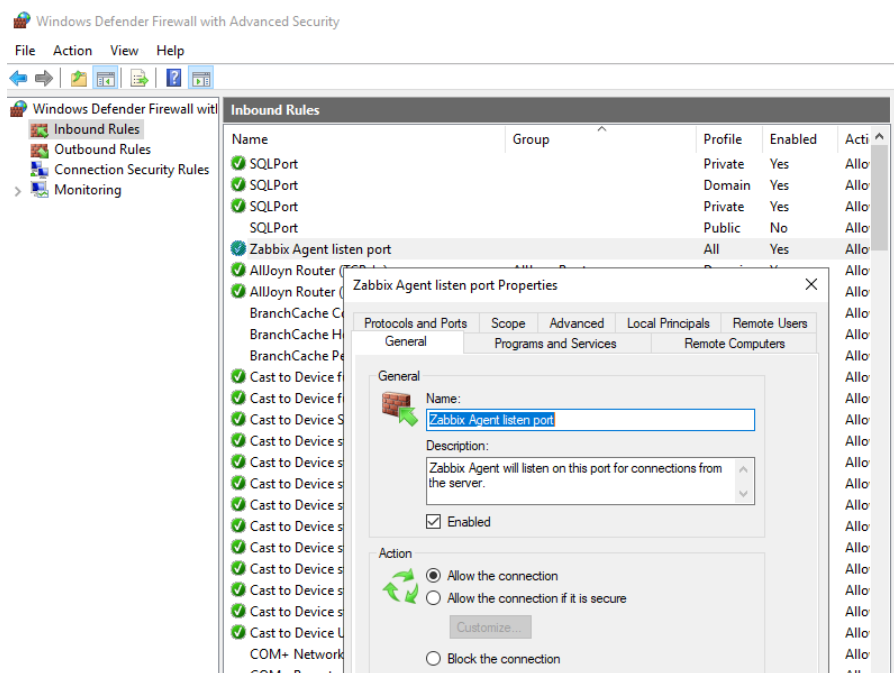


Рис. 2.5. Налаштування правила вхідних з'єднань для Zabbix Agent

У вебінтерфейсі Zabbix створюється новий хост моніторингу. У вікні створення вказується назва хоста, його IP-адреса, порт 10050, а також вибираються шаблони моніторингу. Шаблони моніторингу містять готові набори метрик, які будуть відображатися на Dashboard.

Host

Host IPMI Tags Macros 1 Inventory Encryption Value mapping

* Host name IIS_Server_2024

Visible name IIS_Server_2024

Templates

Name	Actions
IIS by Zabbix agent	Unlink Unlink and clear
Windows by Zabbix agent	Unlink Unlink and clear

type here to search Select

* Host groups Virtual machines X type here to search Select

Interfaces

Type	IP address	DNS name	Connect to	Port	Default
Agent	192.168.2.115		☒ IP ☐ DNS	10050	☒ Remove

Add

Description

Update Clone Delete Cancel

Рис. 2.6. Додавання моніторингу хосту у вебінтерфейсі Zabbix

Система Wazuh має подібну процедуру налаштування, але вона ще використовує ключ автентифікації, щоб з'єднатися зі своїм агентом на хості моніторингу. Тому перед встановленням агента на хості потрібно згенерувати цей ключ у Wazuh Manager. Для цього існує спеціальна утиліта на стороні менеджера Wazuh, вона використовує тільки IP-адресу хоста, на якому буде встановлений Wazuh Agent, для генерації ключа. Далі цей ключ вводиться у вікні агента разом із IP-адресою Zabbix Server, після чого з'являється можливість відслідковувати логи через вебінтерфейс Wazuh.

```
wazuh-user@wazuh-server:~
** No agent available. You need to add one first.
** Press ENTER to return to the main menu.

*****
* Wazuh v4.14.1 Agent manager.          *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: A

- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
    * A name for the new agent: IIS_SERVER_WAZUH_AGENT
    * The IP Address of the new agent: 192.168.2.105
  Confirm adding it?(y/n): y
  Agent added with ID 001.

*****
* Wazuh v4.14.1 Agent manager.          *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: L

Available agents:
  ID: 001, Name: IIS_SERVER_WAZUH_AGENT, IP: 192.168.2.105
** Press ENTER to return to the main menu.

*****
* Wazuh v4.14.1 Agent manager.          *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: E

Available agents:
  ID: 001, Name: IIS_SERVER_WAZUH_AGENT, IP: 192.168.2.105
Provide the ID of the agent to extract the key (or '\q' to quit): 001

Agent key information for '001' is:
MDAxIE1JU19TRVJWRVJFV0FaVUhfQUdFTlQgMTkyLjE2OC4yLjEwNSA3MTVjMGYyZDg5NjYk2MjU1NDY2
OGNlNjklM2U5ZmUxYzUzOWVhY2VlYzVhYzE4NjNhODZmMzc4OWE1N2RlMTUx
```

Рис. 2.7. Генерування ключа автентифікації для Wazuh Agent

Wazuh використовує порт 1514 для прослуховування, тому і для нього також необхідно створити правила вхідних/вихідних з'єднань, щоб брандмауер не блокував трафік між агентом і сервером Wazuh.

New Outbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Does this rule apply to TCP or UDP?

☐ TCP

☒ UDP

Does this rule apply to all remote ports or specific remote ports?

☐ All remote ports

☒ Specific remote ports:

Example: 80, 443, 5000-5010

Рис. 2.8. Створення правила вхідних з'єднань для Wazuh Agent

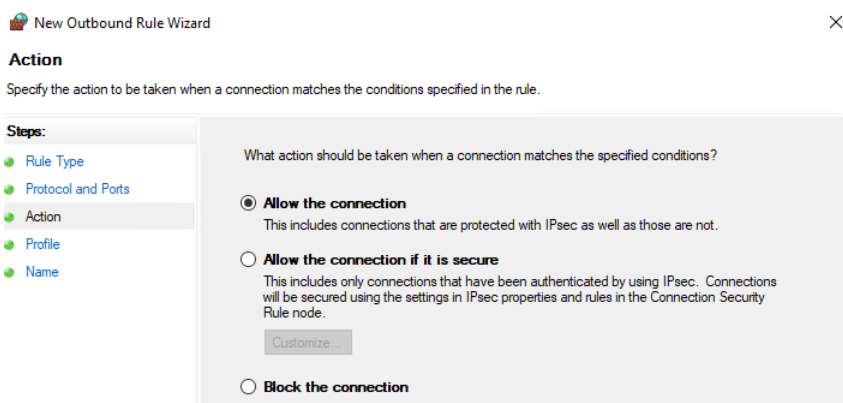


Рис. 2.9. Створення правила вхідних з'єднань для Wazuh Agent

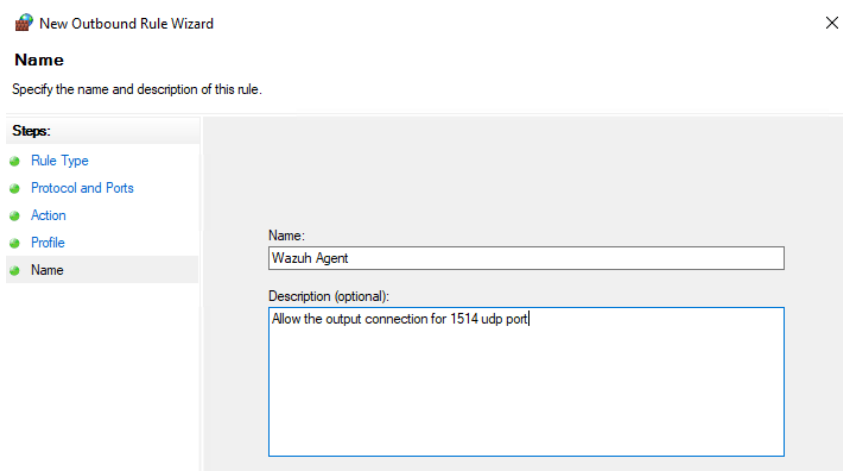


Рис. 2.10. Створення правила вхідних з'єднань для Wazuh Agent

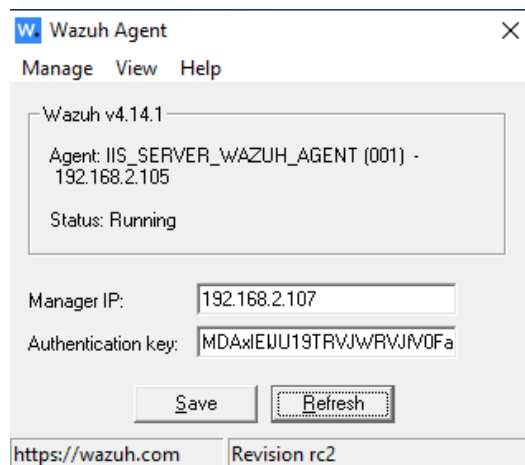


Рис. 2.11. Вікно Wazuh Agent, підключення до Wazuh Server

2.3. Налаштування балансування навантаження NLB.

Балансування навантаження є необхідним етапом для досягнення високої доступності на рівні додатків і забезпечення масштабованості вебстека. У цьому

проєкті для розподілу клієнтського трафіку між двома вебсерверами було обрано технологію Мережевого балансування навантаження Windows (NLB).

Основна мета NLB — забезпечити рівномірний розподіл вхідного трафіку (HTTP/HTTPS) між групою серверів.

NLB не лише оптимізує використання ресурсів, зменшуючи навантаження на окремий вузол, але й виконує функцію відмовостійкості: у разі виходу з ладу одного з вебсерверів (наприклад, VM1), NLB автоматично виключає його з пулу та перенаправляє весь трафік, що залишився, на робочий сервер (VM2), забезпечуючи безперервність роботи сайту Umbraco.

NLB створює єдину віртуальну IP-адресу та DNS-ім'я, до якої підключаються кінцеві користувачі. Таким чином, збій на фізичному сервері стає прозорим для клієнта.

Щоб отримати можливість створювати кластери, потрібно встановити функцію Network Load Balancing через консоль Server Manager.

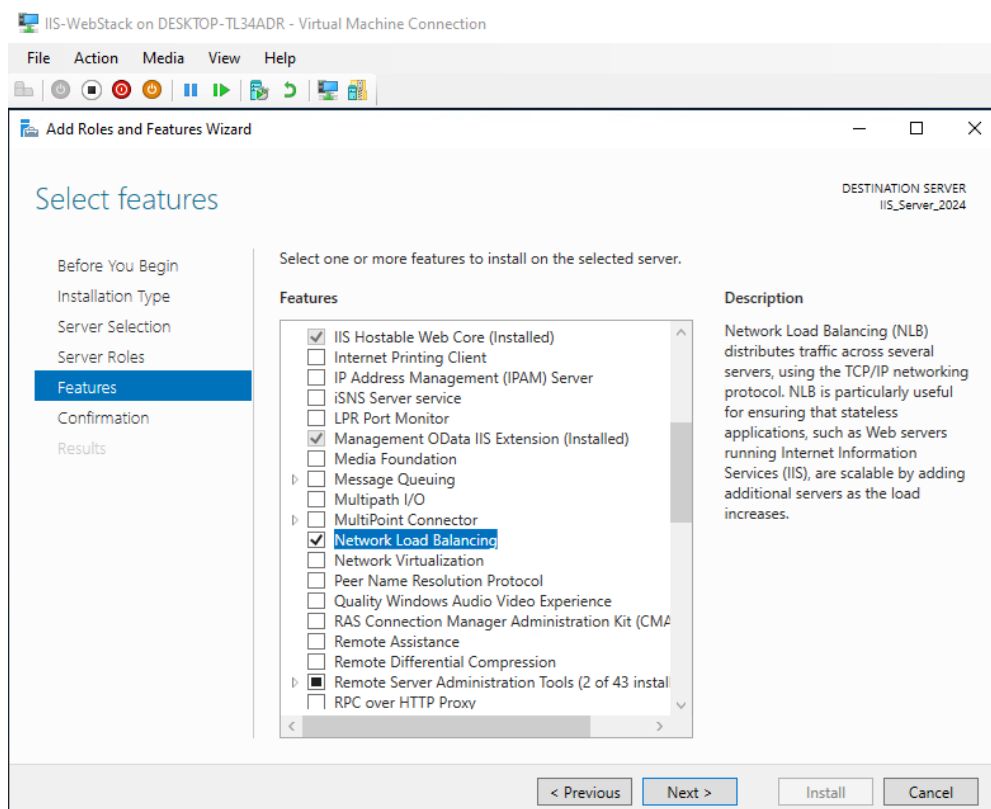


Рис. 2.12. Додавання функції NLB у Server Manager

Був створений новий кластер NLB, до якого були додані обидва вебсервери (VM1 та VM2). Кластеру була призначена єдина віртуальна IP-адреса, що фігурує у Таблиці Адресації (192.168.2.120).

Був обраний Режим Single Network Adapter та встановлений режим роботи кластера Unicast для мінімізації мережевого трафіку.

Правила Портів: Були створені правила для обробки трафіку:

- Порт 80 (HTTP): Для нешифрованого вебтрафіку.
- Порт 443 (HTTPS): Для захищеного вебтрафіку, що є критично важливим для виробничого середовища.

Параметр спорідненості було встановлено на Single (Єдиний). Це гарантує, що запити від одного і того ж клієнта протягом сесії завжди надходять на один і той же вебсервер. Це є важливою вимогою для коректної роботи більшості CMS (включно з Umbraco), які можуть зберігати певні дані сесії або кешу на рівні сервера.

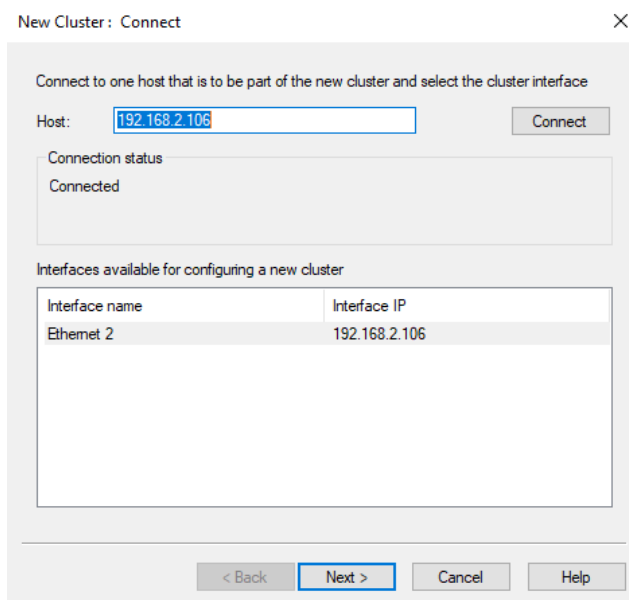


Рис. 2.13. Створення нового NLB-кластера. Додавання першого хоста до кластера

New Cluster : Host Parameters

Priority (unique host identifier): 1

Dedicated IP addresses

IP address	Subnet mask
192.168.2.106	255.255.255.0

Add... Edit... Remove

Initial host state

Default state: Started

☐ Retain suspended state after computer restarts

< Back Next > Cancel Help

Рис. 2.14. Створення нового NLB-кластера. Вибір пріоритету

New Cluster : Cluster IP Addresses

The cluster IP addresses are shared by every member of the cluster for load balancing. The first IP address listed is considered the primary cluster IP address and used for cluster heartbeats.

Cluster IP addresses:

IP address	Subnet mask
192.168.2.120	255.255.255.0

Add... Edit... Remove

< Back Next > Cancel Help

Рис. 2.15. Створення нового NLB-кластера. Додавання IP-адреси кластера

New Cluster : Cluster Parameters

Cluster IP configuration

IP address: 192.168.2.120

Subnet mask: 255 . 255 . 255 . 0

Full Internet name: webcluster.local

Network address: 02-bf-c0-a8-02-78

Cluster operation mode

☒ Unicast

☐ Multicast

☐ IGMP multicast

< Back Next > Cancel Help

Рис. 2.16. Створення нового NLB-кластера. Вибір режиму роботи кластера

New Cluster : Port Rules

Defined port rules:

Cluster IP address	Start	End	Prot...	Mode	Priority	Load	Affinity
All	443	443	Both	Multiple	--	Equal	Single
All	80	80	Both	Multiple	--	Equal	Single

< >

Add... Edit... Remove

Port rule description

TCP and UDP traffic directed to any cluster IP address that arrives on port 443 is balanced equally across all members of the cluster. Client IP addresses are used to assign client connections to a specific cluster host.

< Back Finish Cancel Help

Рис. 2.17. Створення нового NLB-кластера. Визначення правил портів

Рис. 2.18. Створення нового NLB-кластера. Налаштування правил портів

Обидва вебсервери налаштовані на підключення до єдиного джерела даних — віртуального ресурсу Listener (UMBRACO-LSTNR), що гарантує, що обидва сервери працюють з актуальною копією даних.

Хоча Umbraco може кешувати деякі дані локально, контент (медіафайли, зображення) на обох серверах синхронізується, щоб забезпечити, що клієнт, перенаправлений з VM1 на VM2, завжди бачить однакове наповнення сайту.

Успішне налаштування NLB завершило формування високої доступності рівня додатків, забезпечуючи як відмовостійкість, так і здатність до горизонтального масштабування.

2.4. Налаштування WSFC та Always On Availability Groups

Рівень даних є найкритичнішим компонентом інфраструктури, оскільки саме тут зберігаються всі дані вебдодатка. Для забезпечення нульової втрати даних та автоматичного перемикання (Automatic Failover) було імплементовано комплексне кластерне рішення на базі технологій Microsoft.

Windows Server Failover Cluster є обов'язковою основою для функціонування Always On Availability Groups.

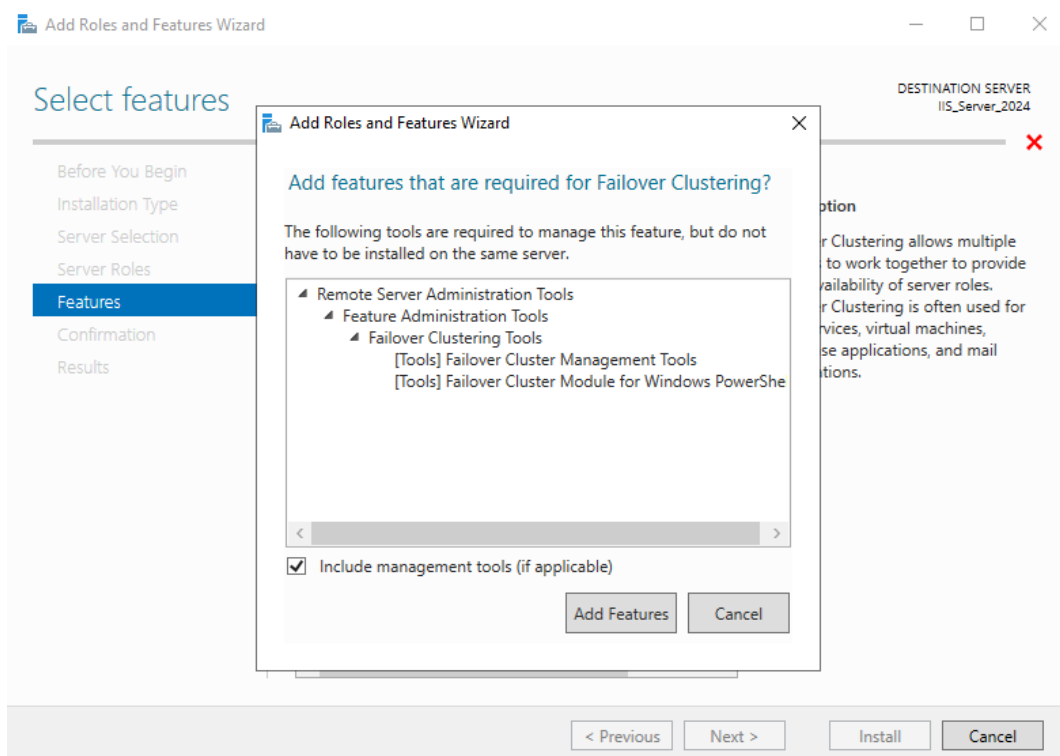


Рис. 2.19. Додавання функції Failover Clustering у Server Manager

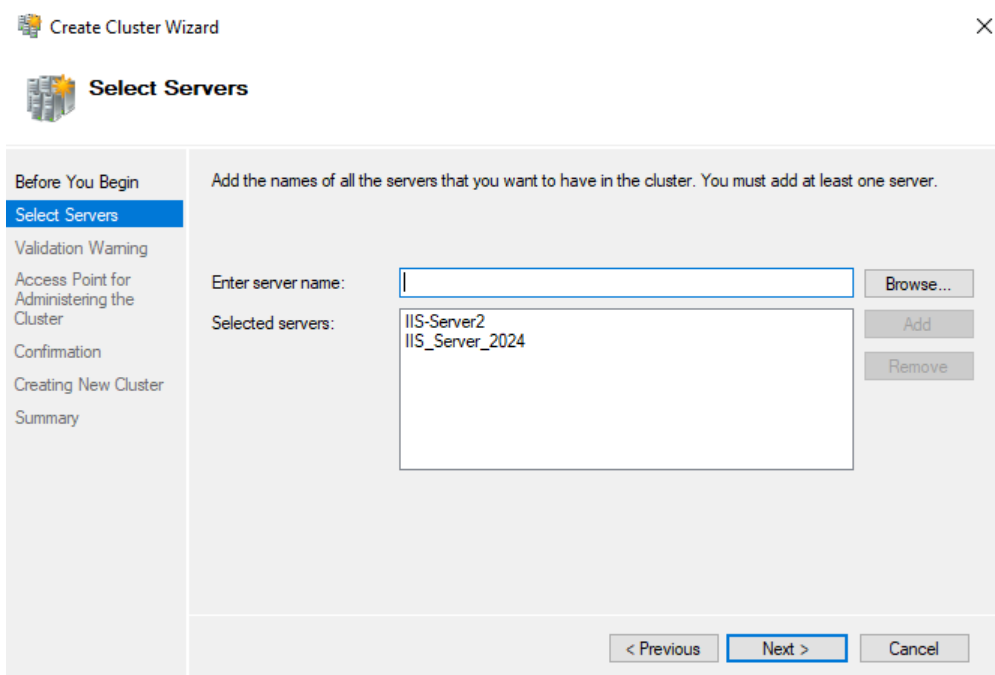


Рис. 2.20. Створення Failover Cluster. Вибір хостів, які стануть частинами кластера

Create Cluster Wizard

Access Point for Administering the Cluster

Before You Begin
Select Servers
Validation Warning
Access Point for Administering the Cluster
Confirmation
Creating New Cluster
Summary

Type the name you want to use when administering the cluster.

Cluster Name:

i The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☒	192.168.2.0/24	192.168.2.118

< Previous **Next >** Cancel

Рис. 2.21. Створення Failover Cluster. Вибір імені та IP-адреси кластера

Create Cluster Wizard

Confirmation

Before You Begin
Access Point for Administering the Cluster
Confirmation
Creating New Cluster
Summary

You are ready to create a cluster.
The wizard will create your cluster with the following settings:

Cluster	WEB-CLUSTER
Node	IIS_Server_2024.mysite.local IIS-Server2.mysite.local
Cluster registration	DNS and Active Directory Domain Services
IP Address	192.168.2.118

☒ Add all eligible storage to the cluster.

To continue, click Next.

< Previous **Next >** Cancel

Рис. 2.22. Створення Failover Cluster. Перегляд вибраних налаштувань кластера

WSFC створює логічну групу з двох серверів, які постійно відстежують стан один одного. Кластер керує віртуальною мережевою назвою та координує процеси перемикавання ролей.

Для надійної роботи та запобігання ситуації "розділення мозку" (Split-Brain) було налаштовано механізм кворуму. У тривузловій конфігурації серверу AD_DC_Controller було призначено як Свідок файлового ресурсу. Ця непарна третя точка голосування дозволяє кластеру однозначно визначити Primary вузол у разі збою одного з серверів, що є критичним для автоматичного перемикавання.

Після успішного налаштування WSFC було створено групу доступності для бази даних UmbracoDB.

Обидва екземпляри SQL Server були додані як репліки до групи доступності. VM1 була встановлена як Первинна Репліка, з якою відбуваються всі операції запису. VM2 була встановлена як Вторинна Репліка, яка отримує та зберігає копію даних.

Обрано Синхронний режим. Це гарантує, що транзакція підтверджується лише після того, як вона була успішно записана як на Primary, так і на Secondary репліках. Ця конфігурація забезпечує нульову втрату даних.

Налаштовано Автоматичне перемикавання, яке дозволяє WSFC самотійно, без втручання адміністратора, підвищити роль VM2 до Primary у разі відмови VM1.

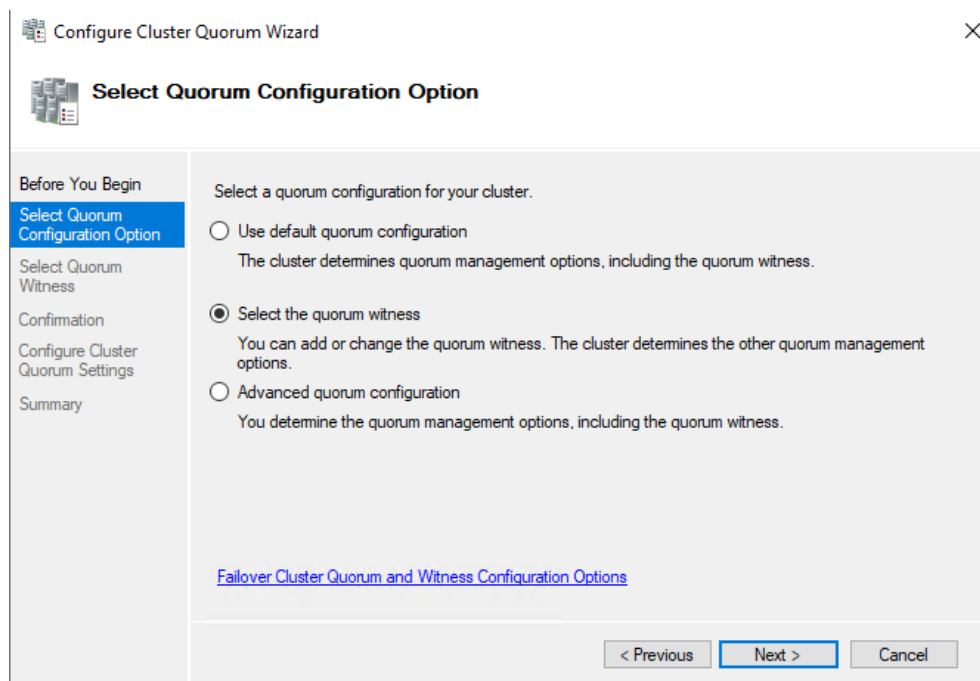


Рис. 2.23. Налаштування кворуму. Обрання свідка кворуму

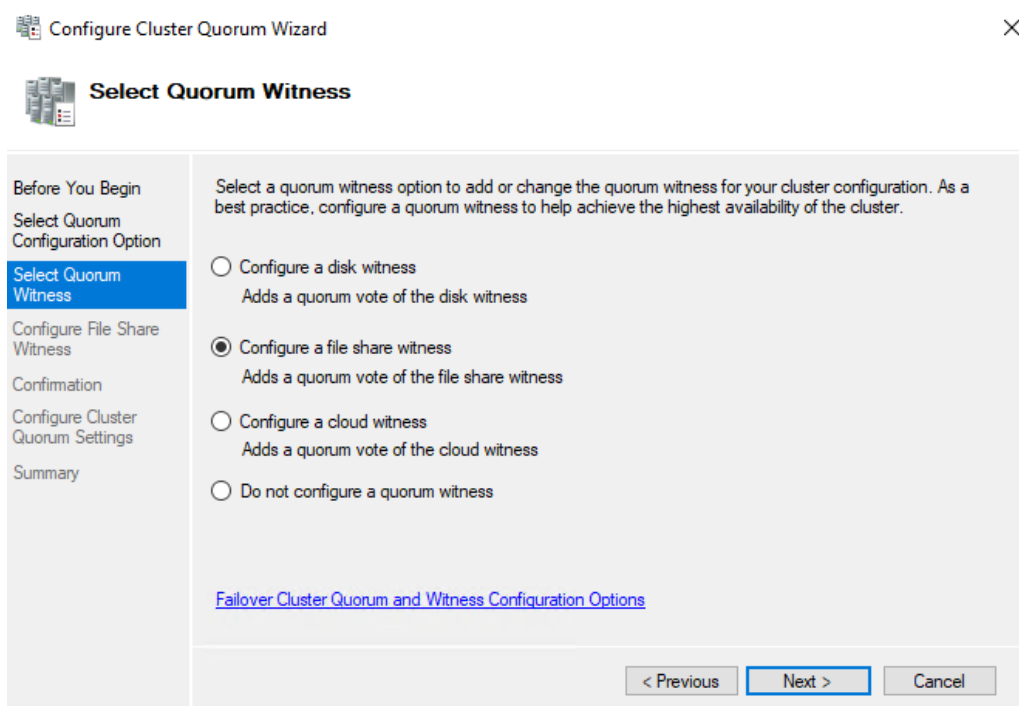


Рис. 2.24. Налаштування кворуму. Обрання свідка кворуму

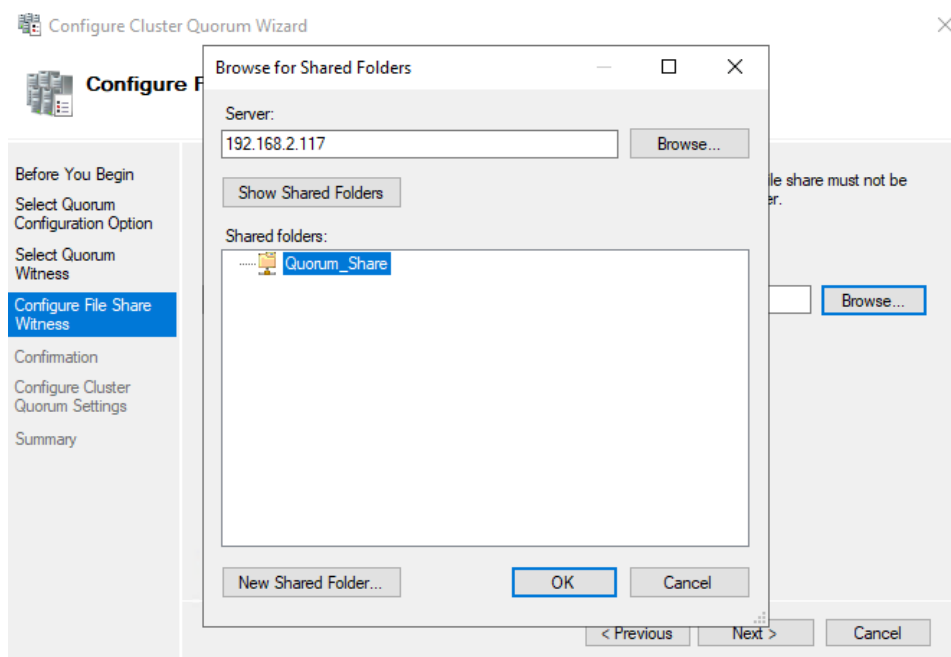


Рис. 2.25. Налаштування кворуму. Обрання спільної папки

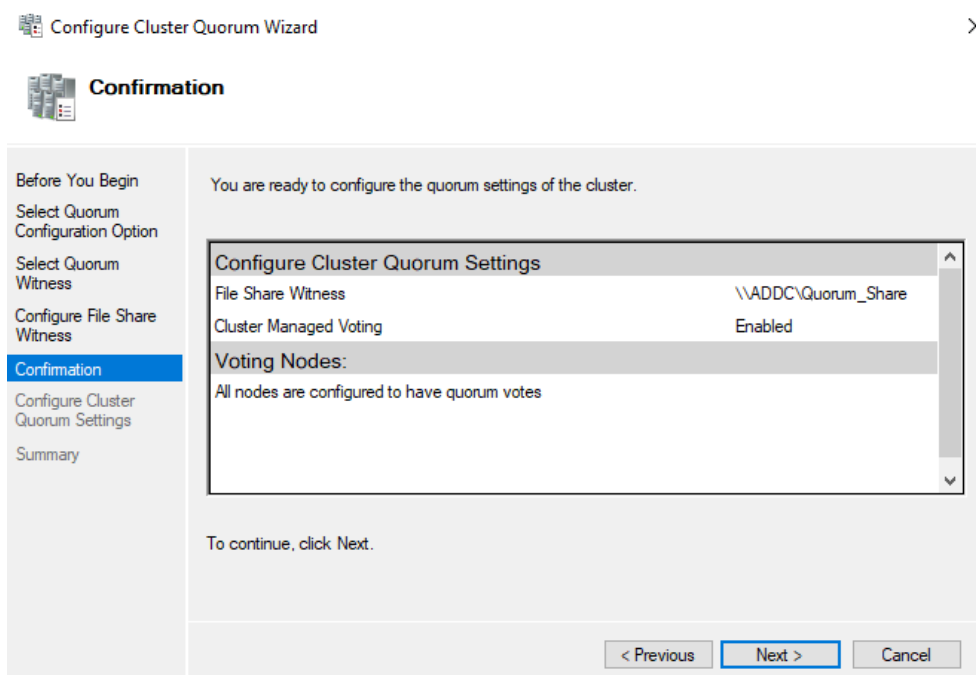


Рис. 2.26. Перегляд обраних налаштувань

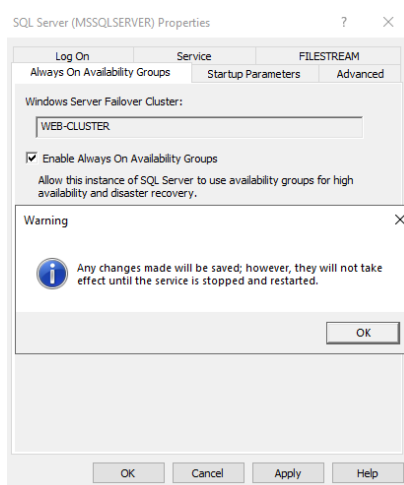


Рис. 2.27. Увімкнення Always On Availability Groups для служби SQL Server

Listener є найважливішим елементом архітектури AG, що забезпечує прозорість збою для вебдодатка. Це віртуальна мережева точкою, яка залишається незмінною, незалежно від того, яка фізична машина є Primary.

Вебдодаток Umbraco налаштовано на підключення саме до цього віртуального імені. У разі перемикання Listener автоматично переміщується на нову Primary Replica, і додаток не потребує зміни конфігурації чи перезапуску, що гарантує максимальну безперервність роботи.

Налаштування WSFC та Always On AGs забезпечило надійний, високодоступний рівень даних, готовий до автоматичного відновлення після будь-якого апаратного чи програмного збою.

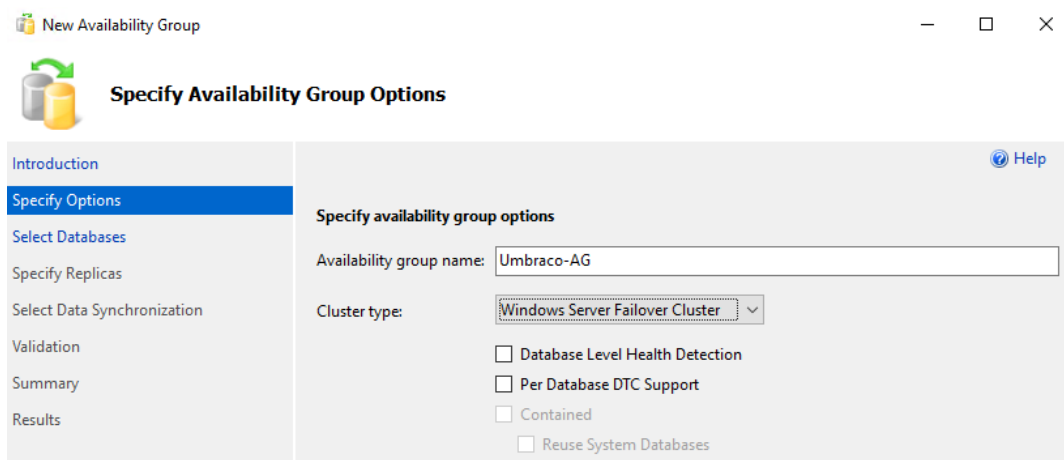


Рис. 2.28. Створення групи на сервері SQL

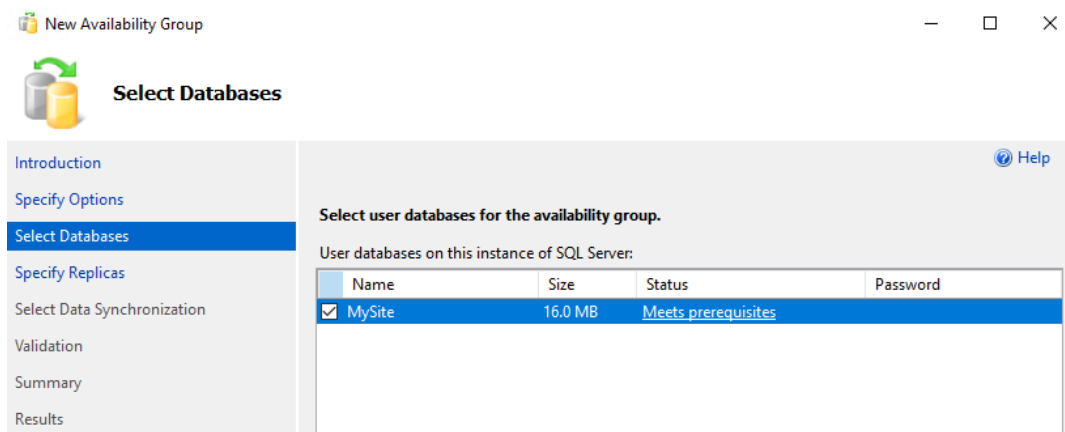


Рис. 2.29. Вибір бази даних для реплікації

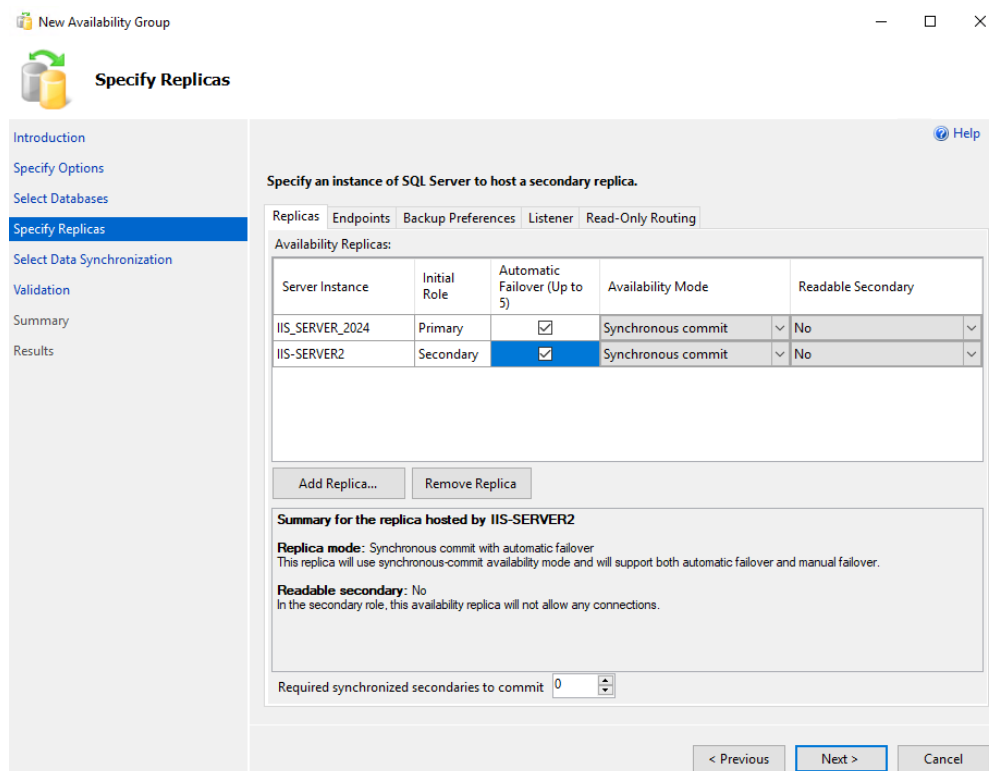


Рис. 2.30. Вибір первинного і вторинного серверів

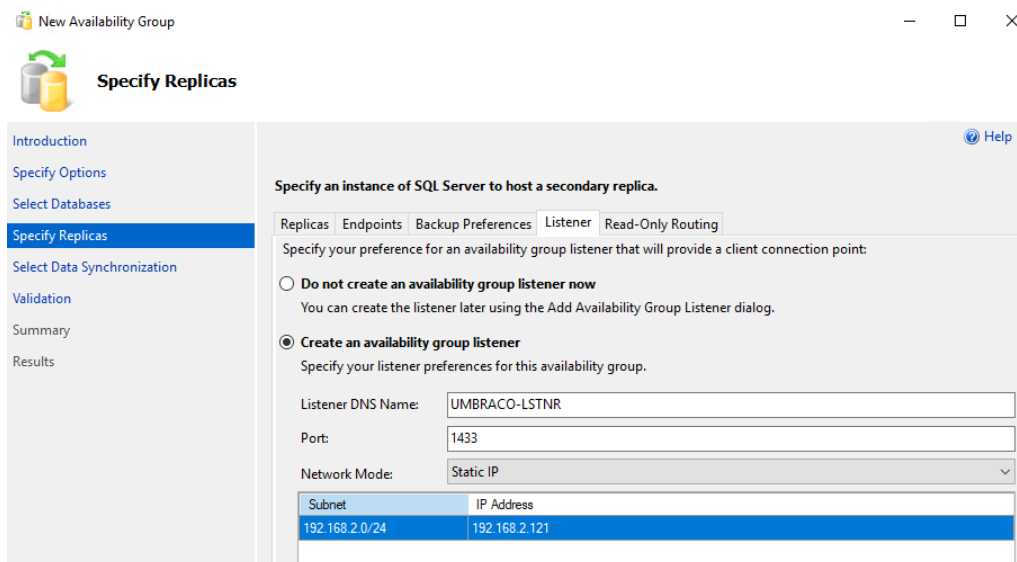


Рис. 2.31. Вибір назви слухача, номеру порту і адреси

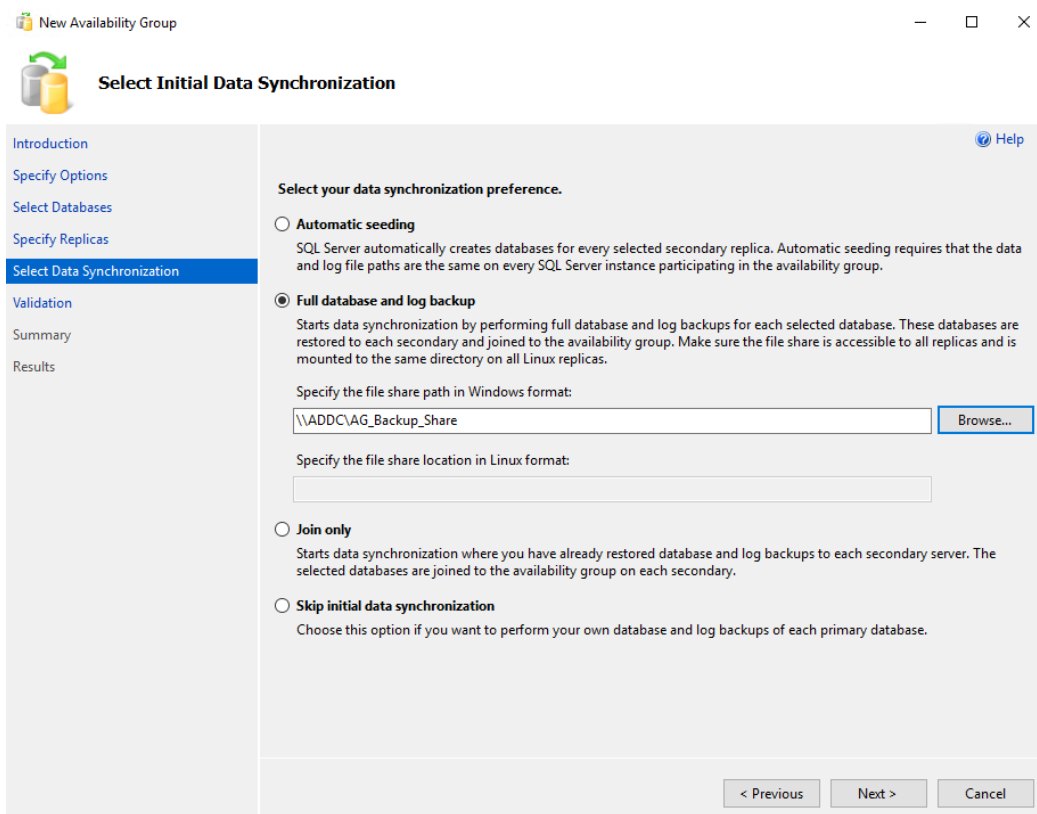


Рис. 2.32. Вибір спільної папки для копіювання БД

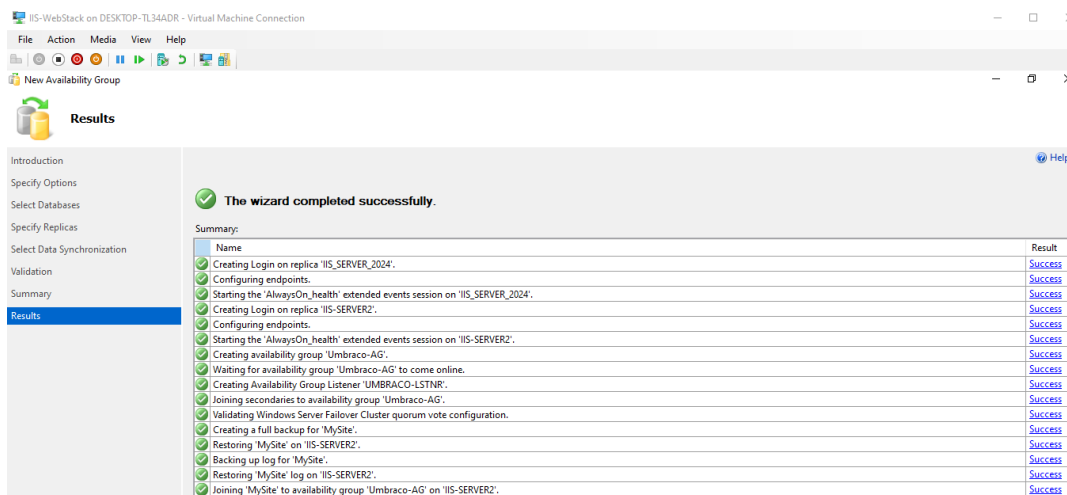


Рис. 2.33. Результат створення групи успішний

2.5. Налаштування захищеного з'єднання HTTPS за допомогою SSL-сертифіката

Для впровадження протоколу HTTPS та забезпечення шифрування трафіку між клієнтом і сервером необхідно використовувати SSL/TLS-сертифікат. У промислових середовищах зазвичай застосовуються сертифікати від довірених центрів сертифікації (CA), таких як Let's Encrypt (безкоштовні) або комерційні сертифікати (DigiCert, Sectigo тощо). Оскільки дана інфраструктура розгорнута в ізольованому лабораторному середовищі без доступу до публічних доменних імен, було прийнято рішення використовувати самопідписаний сертифікат (Self-Signed Certificate).

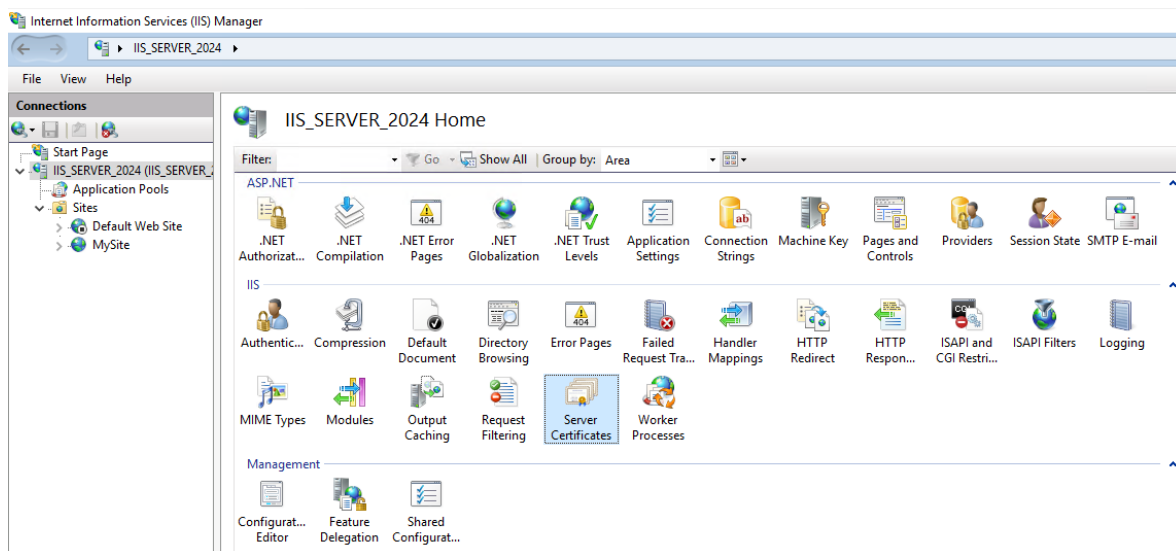


Рис. 2.34. Інструменти вебсервера

Для створення сертифіката засобами графічного інтерфейсу необхідно перейти до консолі управління IIS Manager. На рівні сервера у розділі IIS обирається інструмент Server Certificates. На панелі дій Actions слід обрати пункт Create Self-Signed Certificate, що запустить майстер налаштування.

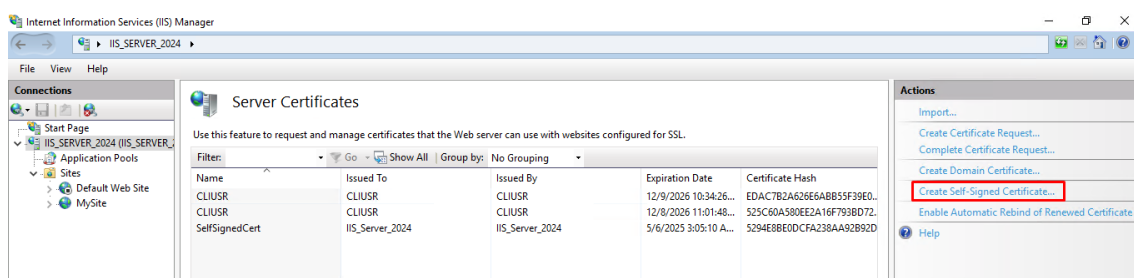


Рис. 2.35. Перехід до створення сертифіката

При заповненні параметрів необхідно вказати дружню назву сертифіката та обрати сховище для його зберігання. У контексті IIS обирається тип сховища Personal. Це зумовлено тим, що самопідписаний сертифікат призначений для локального використання на конкретному сервері, і його закритий ключ повинен зберігатися в особистому сховищі сертифікатів комп'ютера для коректної роботи механізмів автентифікації. [20].

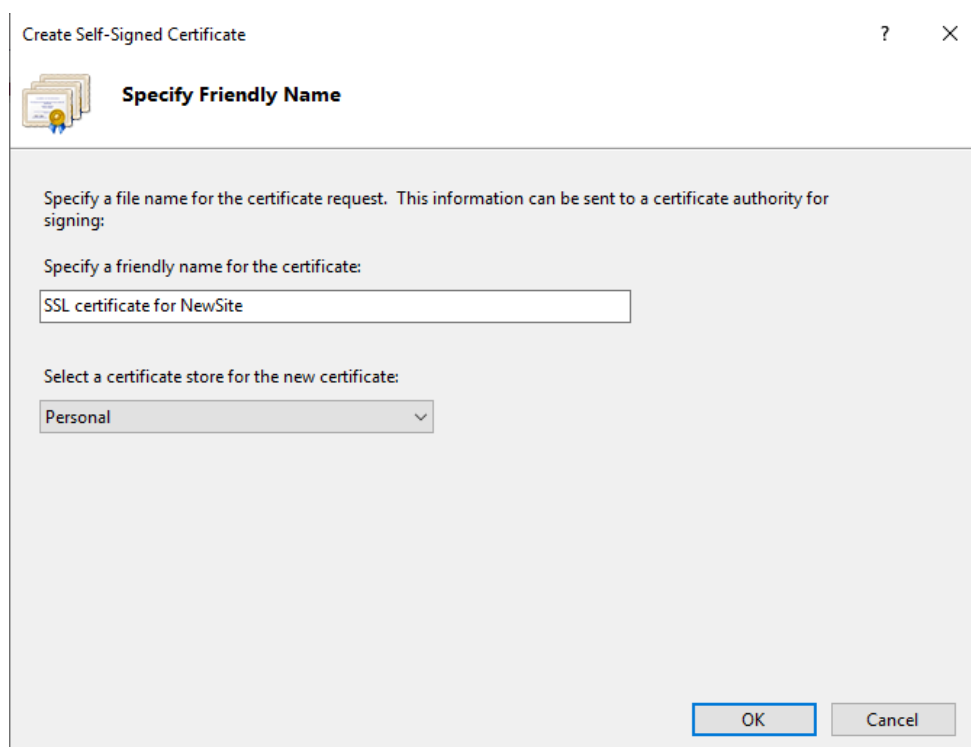


Рис. 2.36. Створення сертифікату

Тепер створений сертифікат відображається у списку усіх підключених сертифікатів.

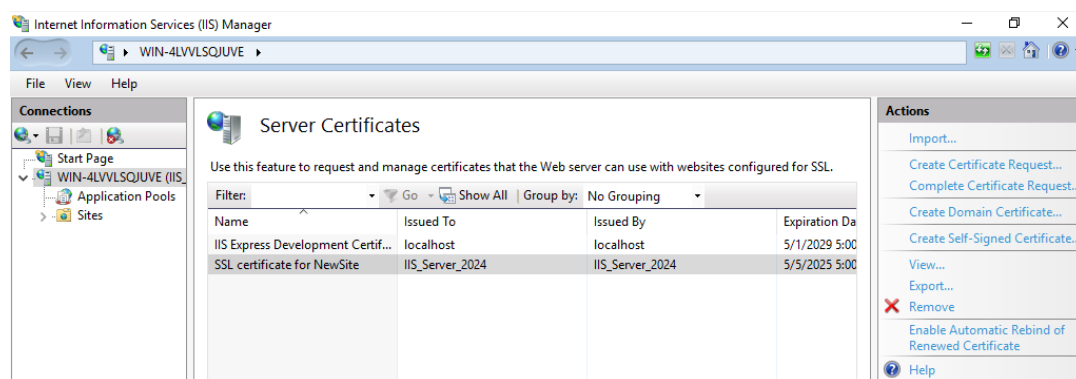
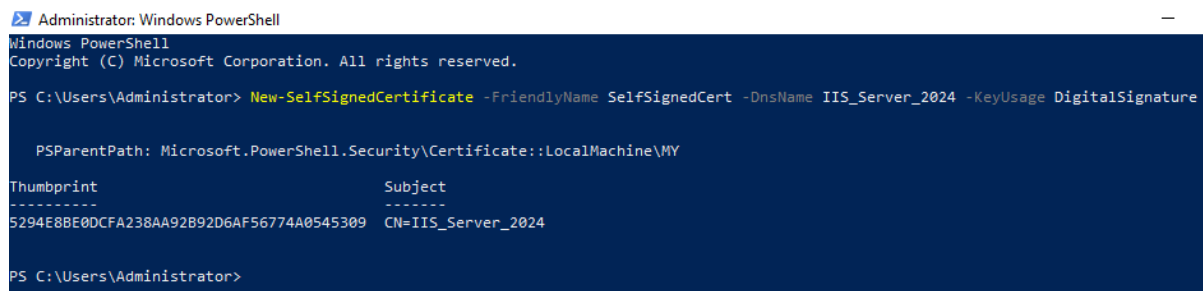


Рис. 2.37. Список сертифікатів

Альтернативним і більш гнучким способом є створення сертифіката за допомогою оболонки PowerShell. Використання командлета New-SelfSignedCertificate дозволяє задати додаткові параметри, такі як термін дії сертифіката (наприклад, 10 років) або альтернативні імена суб'єкта (Subject Alternative Name), що часто є необхідним для сучасних браузерів.



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> New-SelfSignedCertificate -FriendlyName SelfSignedCert -DnsName IIS_Server_2024 -KeyUsage DigitalSignature

PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\MY

Thumbprint                               Subject
-----
5294E8BE0DCFA238AA92B92D6AF56774A0545309  CN=IIS_Server_2024

PS C:\Users\Administrator>

```

Рис. 2.38. Створення SSL-сертифікату за допомогою команди PowerShell

Після створення сертифіката його необхідно підключити до конкретного вебсайту. Для цього у дереві консолі IIS обирається відповідний сайт, і на панелі дій натискається пункт Bindings (Прив'язки).

У вікні редагування прив'язок слід додати новий запис:

- Type: змінити з http на https.
- Port: за замовчуванням встановлюється 443.
- SSL certificate: обрати створений раніше сертифікат із випадаючого списку.

Після збереження налаштувань вебсайт стає доступним за захищеним протоколом. Важливо зауважити, що при використанні самопідписаного сертифіката браузер видаватиме попередження про недовіру до видавця, що є нормальним явищем для внутрішніх тестових середовищ і вирішується шляхом ручного додавання сертифіката до списку «Довіраних корневих центрів сертифікації» на клієнтських машинах.

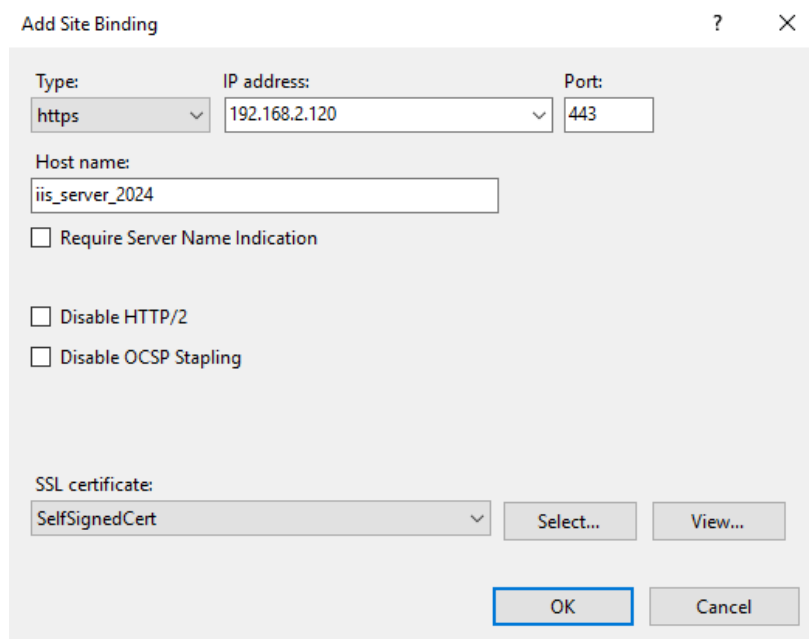


Рис. 2.39. Створення нової прив'язки

Висновки до розділу 2

У другому розділі було виконано проєктування та практичну реалізацію комплексної оптимізованої й захищеної інфраструктури вебстека на базі технологій Microsoft. У результаті проведеної роботи розроблено та впроваджено трирівневу архітектуру з використанням віртуалізації Nureg-V, яка забезпечує логічне розділення функціональних ролей між контролером домену, вебсерверним рівнем та рівнем бази даних. Такий підхід дозволив створити ізольоване, кероване та масштабоване середовище, придатне для подальшого розвитку.

У межах вебсерверного рівня успішно реалізовано механізм балансування навантаження Network Load Balancing між вузлами VM1 та VM2, що забезпечило рівномірний розподіл клієнтського трафіку, підвищення продуктивності та досягнення високої доступності вебдодатка Umbraco. Запропоноване рішення мінімізує ризики перевантаження окремих серверів і створює основу для горизонтального масштабування.

Ключовим елементом забезпечення відмовостійкості на рівні даних стало розгортання кластера Windows Server Failover Cluster та реалізація груп доступності SQL Server Always On. Використання файлового свідка кворуму та віртуального прослуховувача SQL Listener забезпечило автоматичне перемикання ролей між репліками без втрати даних і без необхідності змін у конфігурації вебдодатка у разі виникнення відмов.

Окрему увагу було приділено формуванню основ багатошарового захисту інфраструктури, зокрема шляхом впровадження захищеного каналу зв'язку HTTPS з використанням SSL-сертифікатів та підготовки середовища до інтеграції систем моніторингу та безпеки Zabbix і Wazuh. Спроектowana та реалізована архітектура продемонструвала високу готовність до експлуатації, поєднуючи механізми оптимізації трафіку, автоматичного відновлення сервісів і підвищеного рівня захищеності на всіх рівнях вебстека а також забезпечує як високий рівень доступності (High Availability), так і елементи відмовостійкості (Fault Tolerance) на рівні вебдодатка та бази даних.

РОЗДІЛ 3. Експериментальне дослідження та оцінка ефективності

3.1. Тестування розподілу навантаження.

Цей тест є фінальним підтвердженням того, що система ефективно розподіляє ресурси між вузлами. Використання JMeter як джерела навантаження та Zabbix як інструменту візуалізації дозволяє отримати об'єктивні докази успішної роботи NLB.

Метою тесту є перевірка коректності роботи кластера мережевого балансування навантаження (Network Load Balancing, NLB) та підтвердження того, що вхідні запити обробляються обома серверами одночасно.

Для створення реалістичного потоку запитів було конфігуровано план тестування у JMeter. Встановлено кількість віртуальних користувачів — 50, з періодом розгону у 60 секунд. Необхідно додати елемент HTTP Request, за допомогою нього здійснюється підключення до цілі тесту. Цільовою адресою обрано віртуальну IP-адресу NLB-кластера. Також можна додати елементи Summary Request, View Results Tree, HTTP Header Manager для кращого контролю над процесом.

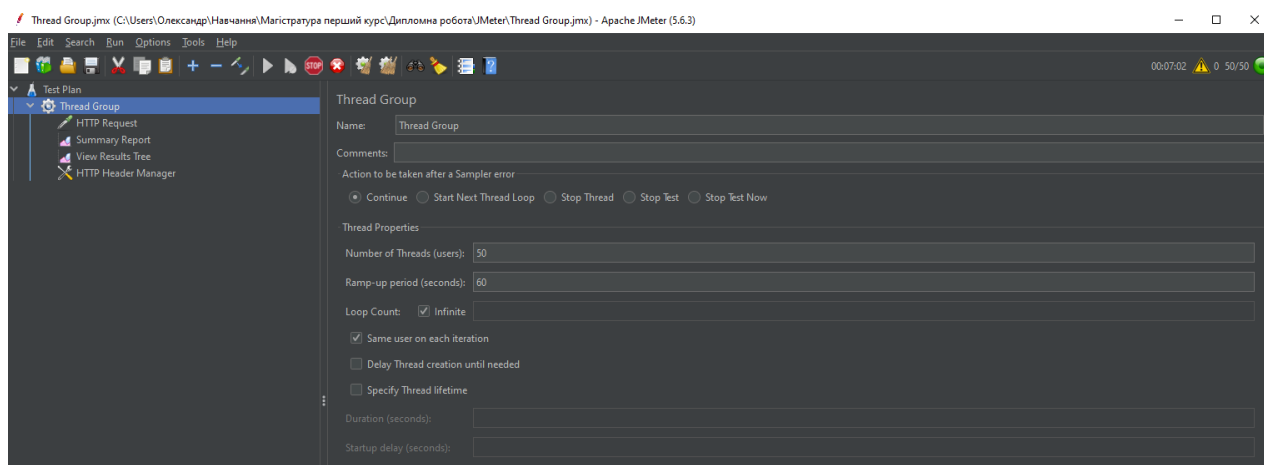


Рис. 3.1. Створення нового плану

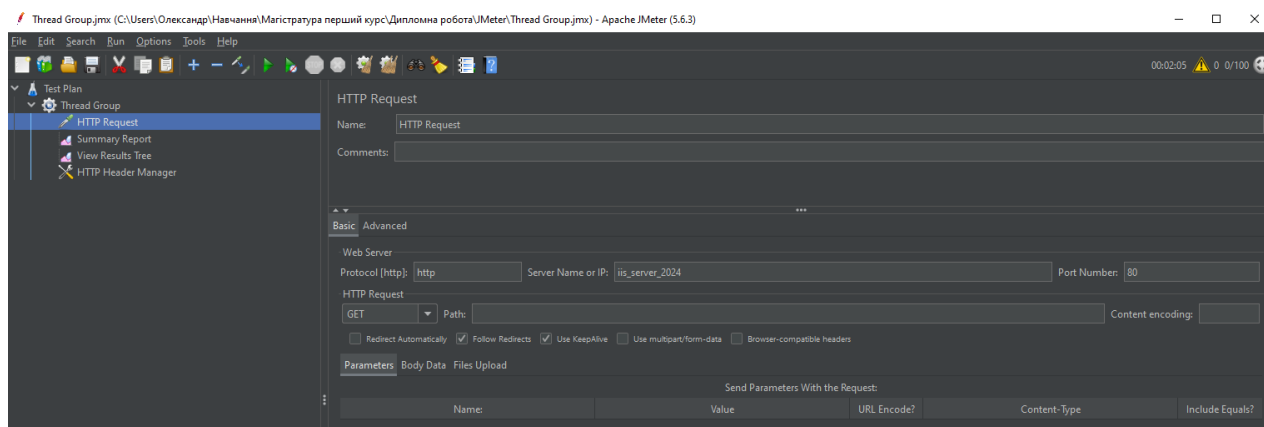


Рис. 3.2. Налаштування підключення до сайту в елементі HTTP Request

Після старту тесту в JMeter починається масове надсилення HTTP-запитів до вебстека. У консолі JMeter (View Results Tree / Aggregate Report) відстежується відсутність помилок (Error % має бути значення близьке 0) та стабільний час відповіді.

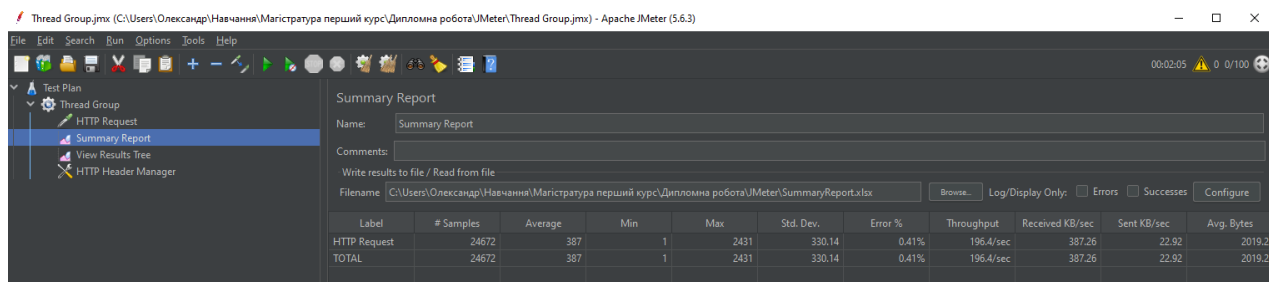


Рис. 3.3. Відомості елемента Summary Report

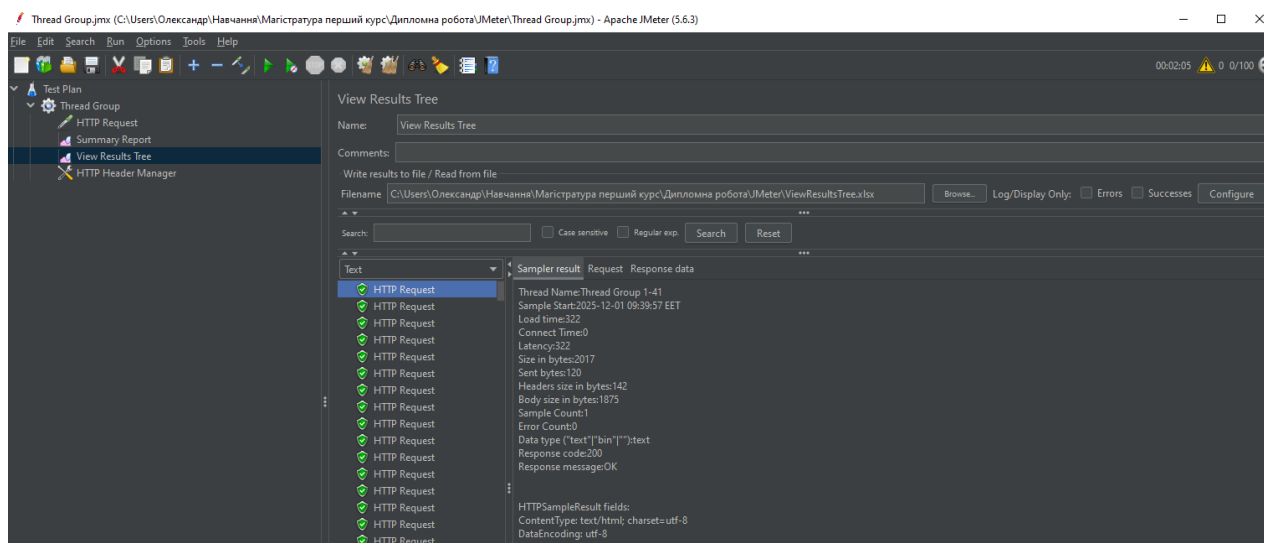


Рис. 3.4. Відомості елемента View Results Tree

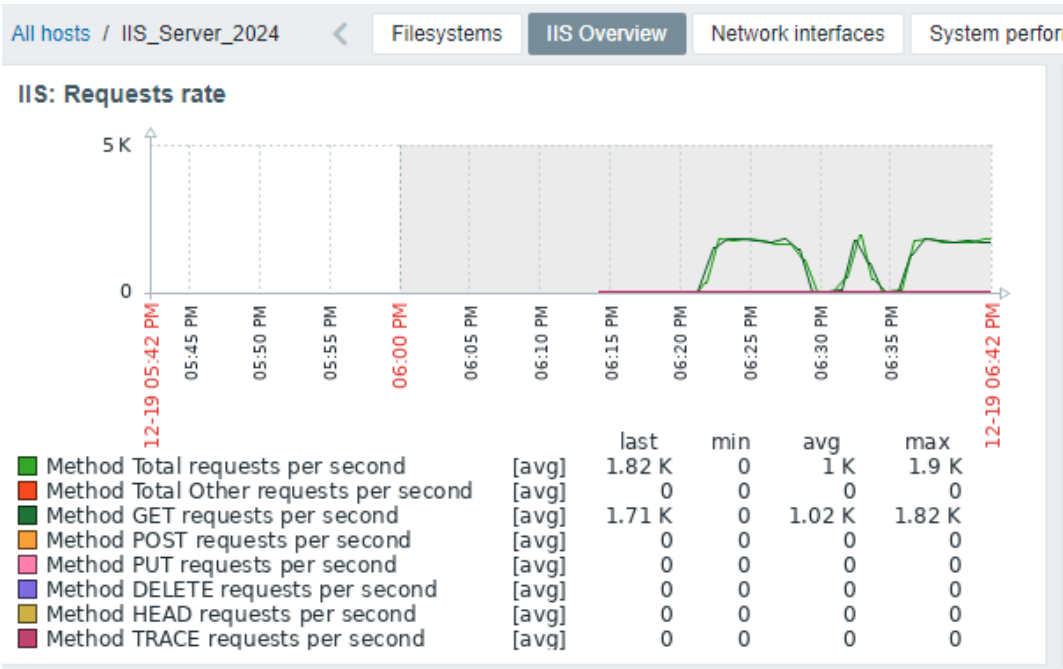


Рис. 3.5. Графік швидкості надходження запитів на сервер IIS_Server_2024

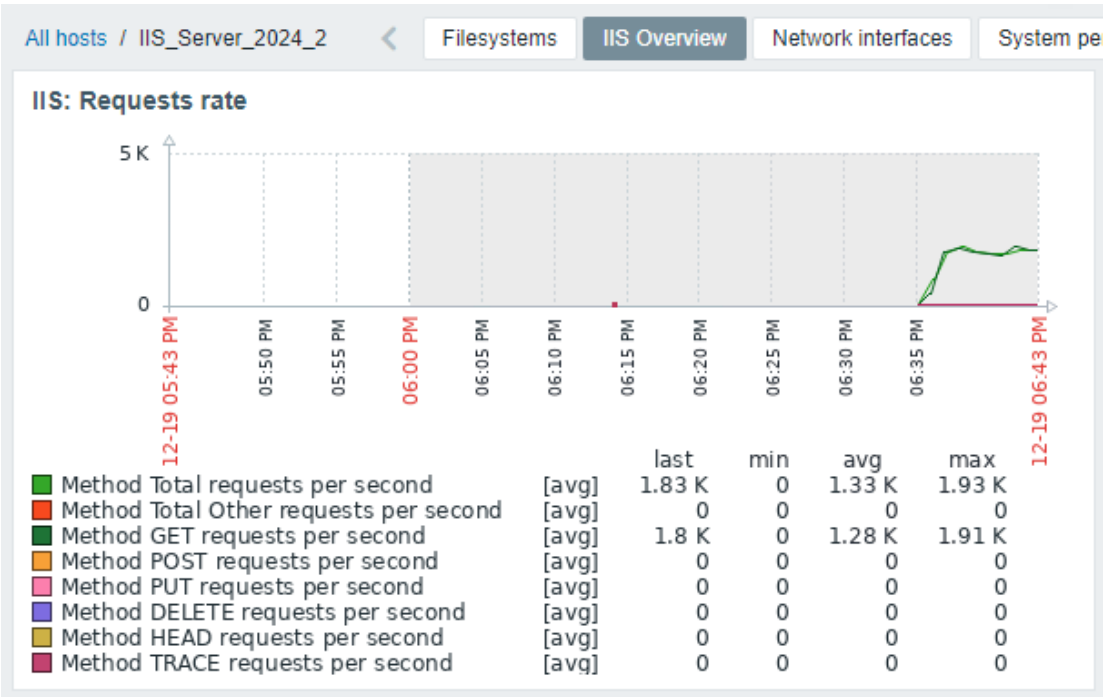


Рис. 3.6. Графік швидкості надходження запитів на сервер IIS_Server_2024_2

На рисунках 3.5 і 3.6 видно, що запити надходять до обох серверів одночасно, а це означає, що балансування навантаження працює належним чином.

Проведене тестування з використанням Apache JMeter продемонструвало ефективність налаштованого NLB-кластера. Дашборди системи Zabbix

зафіксували рівномірний розподіл вхідного трафіку та обчислювального навантаження між вузлами VM1 та VM2. Отримані дані підтвердили, що впровадження балансування дозволяє уникнути перевантаження окремих серверів та забезпечує стабільну швидкість відгуку вебдодатка Umbraco навіть при інтенсивному потоці звернень. Це доводить успішну оптимізацію продуктивності розробленої інфраструктури.

3.2. Тестування відмовостійкості рівня даних.

Це ключовий тест для підтвердження відмовостійкості рівня даних. Цей тест покаже, що система здатна не просто працювати на двох серверах, а й автоматично відновлюватися після критичного збою.

Метою тесту є перевірка здатності системи автоматично перемикає базу даних на вторинну репліку у разі повної відмови первинного сервера та підтримання безперервної роботи сайту.

Перед початком тесту необхідно переконатися, що група доступності знаходиться у стабільному стані:

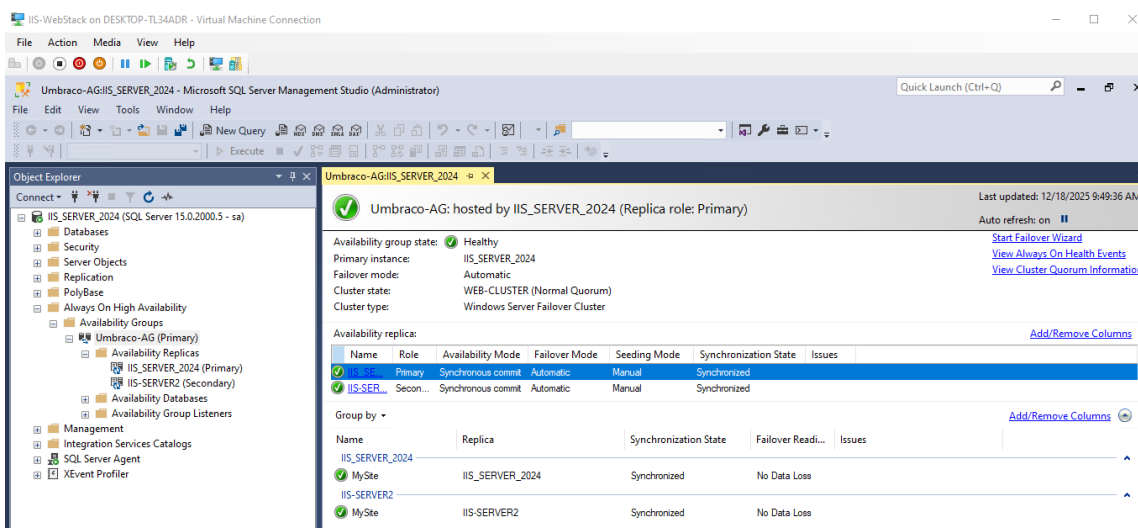


Рис. 3.7. Перегляд Dashboard про стан реплік

Для імітації найгіршого сценарію (апаратної відмови) проводиться раптове вимкнення первинного вузла.

Virtual Machines					
Name	State	CPU Usa...	Assigned M...	Uptime	S
AD_DC_Controller	Running	11%	1988 MB	00:35:11	
IIS-WebStack	Saved				
IIS-WebStack2	Running	5%	3038 MB	00:30:31	
Wazuh	Saved				
Zabbix	Running	0%	3692 MB	00:34:35	

Рис. 3.8. Вимкнення одного сервера

У цей момент кластер WSFC на VM3 (Свідок кворуму) та VM2 фіксують втрату зв'язку з VM1. Завдяки налаштованому кворуму, VM2 автоматично перебирає на себе роль Primary.

Object Explorer		Umbraco-AG: hosted by IIS-SERVER2 (Replica role: Primary)		Last updated: 18.12.2025 9:51:41	
Connect		Availability group state: Critical --- Critical (1), Warnings (3)		Auto refresh: on	
IIS-SERVER2 (SQL Server 15.0.2000.5 - sa)		Primary instance: IIS-SERVER2		Start Failover Wizard	
Databases		Failover mode: Automatic		View Always On Health Events	
Security		Cluster state: WEB-CLUSTER (Normal Quorum)		View Cluster Quorum Information	
Server Objects		Cluster type: Windows Server Failover Cluster			
Replication		Availability replica:		Add/Remove Columns	
PolyBase		Name	Role	Availability Mode	Failover Mode
Always On High Availability		Seeding Mode	Synchronization State	Issues	
Availability Groups		IIS-SER2	Primary	Synchronous commit	Automatic
Umbraco-AG (Secondary)		IIS_SERVER_2024	Secon.	Synchronous commit	Automatic
Availability Replicas		IIS-SER2	Primary	Synchronous commit	Automatic
IIS-SERVER2_2024		IIS_SERVER_2024	Secon.	Synchronous commit	Automatic
IIS-SERVER2 (Secondary)		IIS-SER2	Primary	Synchronous commit	Automatic
Availability Databases		IIS_SERVER_2024	Secon.	Synchronous commit	Automatic
Availability Group Listeners		IIS-SER2	Primary	Synchronous commit	Automatic
Management		IIS_SERVER_2024	Secon.	Synchronous commit	Automatic
Integration Services Catalogs		IIS-SER2	Primary	Synchronous commit	Automatic
SQL Server Agent		IIS_SERVER_2024	Secon.	Synchronous commit	Automatic
XEvent Profiler		IIS-SER2	Primary	Synchronous commit	Automatic

Рис. 3.9. Повторна перевірка стану реплік

Сайт може завантажуватися на кілька секунд довше, але він відкривається без помилок. Користувач не бачить повідомлень про неможливість підключення до БД.

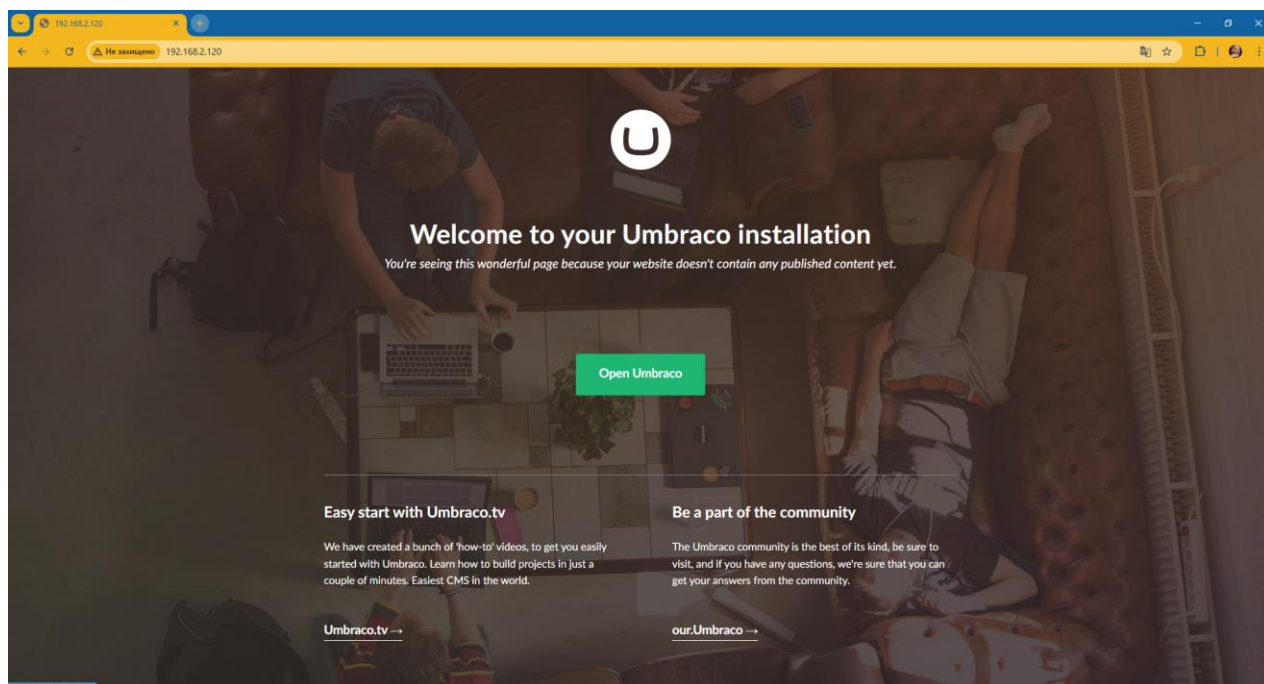


Рис. 3.10. Спроба відкрити сайт, під час перемикання

Результати тестування Always On Availability Groups підтвердили високу ефективність обраної архітектури. Примусове вимкнення сервера з основною реплікою бази даних призвело до автоматичного спрацювання механізму Failover. Завдяки використанню SQL Listener та синхронного режиму реплікації, вебдодаток Umbraco зберіг повну працездатність та доступ до даних. Час простою (Downtime) під час перемикання становив менше 10 секунд, що повністю відповідає вимогам до високонавантажених відмовостійких систем.

Висновки з розділу 3

У третьому розділі було проведено комплексне тестування та експериментальну перевірку працездатності розгорнутої віртуалізованої інфраструктури вебстека, що дозволило підтвердити ефективність та доцільність обраних архітектурних і технологічних рішень.

За результатами навантажувального тестування, виконаного із застосуванням інструмента Apache JMeter, встановлено, що використання механізму мережевого балансування навантаження NLB забезпечує стабільний час відгуку вебдодатка та рівномірний розподіл вхідних HTTP-запитів між вузлами кластера. Отримані показники продуктивності та використання ресурсів

були зафіксовані й проаналізовані за допомогою системи моніторингу Zabbix, що підтвердило відсутність критичних перевантажень окремих серверів під час пікових навантажень.

Окрему увагу було приділено перевірці механізмів відмовостійкості на рівні бази даних. Експеримент із примусовим вимкненням первинної репліки SQL Server підтвердив коректну роботу технології Always On Availability Groups, оскільки автоматичне перемикання ролей на резервний вузол відбулося без втрати цілісності даних та без необхідності втручання адміністратора. При цьому вебдодаток Umbraco зберіг доступність для кінцевих користувачів, а час простою під час перемикання залишався в межах допустимих значень.

Проведені випробування засвідчили, що розроблена інфраструктура відповідає вимогам високої доступності, відмовостійкості та продуктивності, а також є готовою до експлуатації в умовах реального навантаження.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено та практично реалізовано проєкт відмовостійкої та захищеної інфраструктури вебстека на базі технологій Microsoft. Створена система комплексно реалізує концепцію високої доступності, поєднуючи механізми балансування навантаження, кластеризації баз даних, централізованого управління та моніторингу безпеки. За результатами дослідження та практичної реалізації досягнуто поставлених у роботі цілей і виконано визначені завдання.

Спроектовано та розгорнуто комплексну мережеву інфраструктуру, створено ізольоване віртуалізоване середовище на базі гіпервізора Hyper-V, а також розгорнуто доменну інфраструктуру Active Directory для централізованого управління доступом і конфігураціями вузлів. Високу доступність вебрівня реалізовано за допомогою механізму Network Load Balancing, що дозволило об'єднати ресурси двох вебсерверів у єдиний логічний кластер. Результати навантажувального тестування в Apache JMeter підтвердили здатність системи

до горизонтального масштабування та стабільної роботи під час пікових навантажень без деградації сервісу.

Впроваджено відмовостійку систему зберігання даних на базі технології SQL Server Always On Availability Groups у складі Windows Server Failover Cluster. Практичні випробування показали, що використання віртуального SQL Listener забезпечує автоматичне перемикання на резервну репліку у разі відмови основного вузла без втрати даних та необхідності втручання з боку адміністратора, що гарантує безперервність роботи вебдодатка. Для захисту переданих даних налаштовано захищений канал зв'язку з використанням протоколу HTTPS та SSL-сертифікатів, що забезпечило конфіденційність і цілісність користувацького трафіку.

Сформовано багаторівневу систему моніторингу та безпеки шляхом інтеграції Zabbix для збору та аналізу метрик продуктивності в реальному часі, а також системи Wazuh для моніторингу подій безпеки, контролю цілісності файлів і виявлення потенційних загроз. Комплексна верифікація інфраструктури шляхом тестів на відмовостійкість підтвердила, що спроектована архітектура відповідає сучасним вимогам до високодоступних та безпечних вебсистем і здатна до автоматичного відновлення працездатності без порушення цілісності даних.

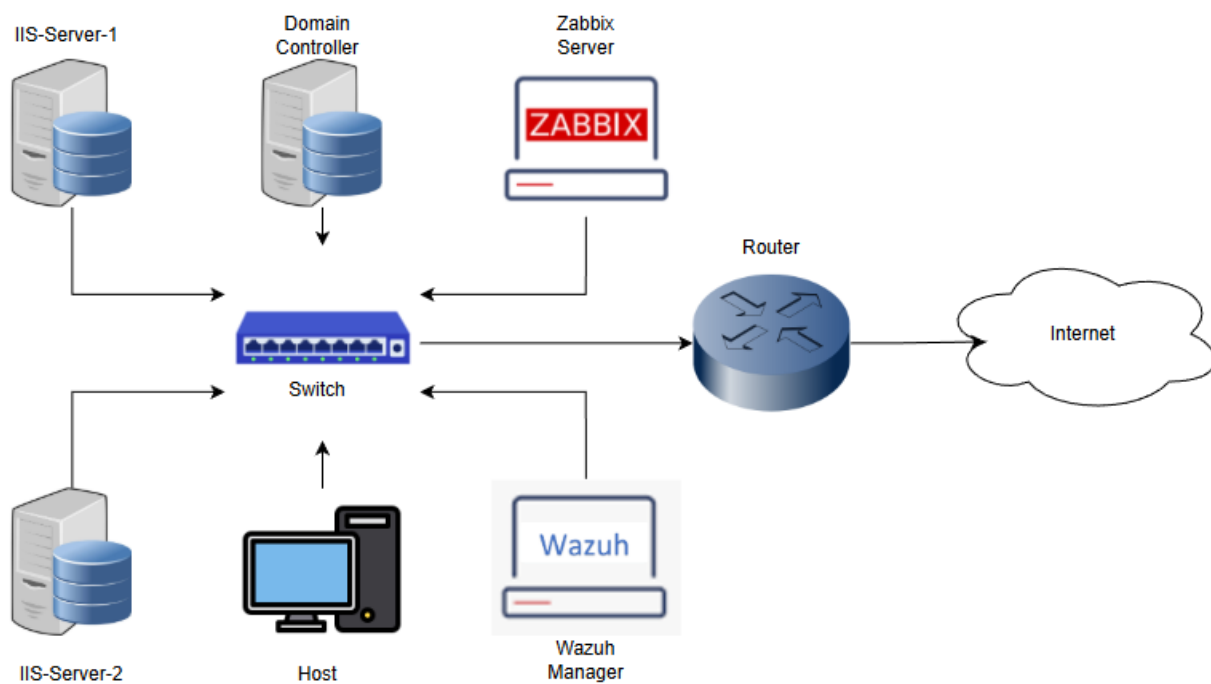
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Steve Souders. High Performance Web Sites: Essential Knowledge for Front-End Engineers. Sebastopol : O'Reilly Media, 2007. 160 с.
2. Mike Julian. Practical Monitoring: Effective Strategies for the Real World, Mike Julian. Sebastopol : O'Reilly Media, 2017. 156 с.
3. Jennifer Petoff, Betsy Beyer, Mike Julian, Chris Jones. Site Reliability Engineering: How Google Runs Production Systems. Sebastopol : O'Reilly Media, 2016. 552 с.
4. Sam Newman. Building Microservices: Designing Fine-Grained Systems. Sebastopol : O'Reilly Media, 2015. 280 с.
5. IIS Web Server Overview. URL: <https://learn.microsoft.com/uk-ua/iis/get-started/introduction-to-iis/iis-web-server-overview>. (дата звернення: 09.12.2024).
6. Internet Information Services. URL: https://uk.wikipedia.org/wiki/Internet_Information_Services. (дата звернення: 09.12.2024).
7. Apache HTTP Server Documentation. URL: <https://httpd.apache.org/docs/> (дата звернення 21.01.2025).
8. Using the Application Request Routing Module. URL: <https://learn.microsoft.com/en-us/iis/extensions/planning-for-arr/using-the-application-request-routing-module>. (дата звернення: 09.12.2024).
9. OS-level virtualization. URL: https://en.wikipedia.org/wiki/OS-level_virtualization. (дата звернення 21.01.2025).
10. Virtualization. URL: <https://en.wikipedia.org/wiki/Virtualization>. (дата звернення 21.01.2025).
11. Nginx documentation. URL: <https://nginx.org/en/docs/>. (дата звернення 21.01.2025).
12. Як пришвидшити систему з допомогою кешування. Стратегії та підходи. URL: <https://dou.ua/forums/topic/43076/>. (дата звернення 21.01.2025).
13. Achieving High Availability and Scalability - ARR and NLB. URL: <https://learn.microsoft.com/en-us/iis/extensions/configuring-application->

- [request-routing-arr/achieving-high-availability-and-scalability-arr-and-nlb.](#)
(дата звернення 21.01.2025).
14. What is an Always On availability group? URL:
[https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/](#)
(дата звернення 21.01.2025).
15. Zabbix Manual. URL: [https://www.zabbix.com/documentation/7.0/en/manual.](#)
(дата звернення 21.01.2025).
16. Prometheus Overview. URL:
[https://prometheus.io/docs/introduction/overview/.](#) (дата звернення
21.01.2025).
17. Nagios Documentation. URL: [https://www.nagios.org/documentation/.](#) (дата
звернення 21.01.2025).
18. Grafana documentation. URL: [https://grafana.com/docs/grafana/latest/.](#) (дата
звернення 21.01.2025).
19. Documentation. URL: [https://documentation.wazuh.com/current/index.html.](#)
(дата звернення 21.01.2025).
20. SSL-сертифікати: що це, для чого потрібно та як згенерувати запит на
підписання сертифіката. URL: [https://tucha.ua/uk/blog/support/ssl-certifikati-
scho-ce-dlya-chogo-potribno-ta-yak-zgeneruvati-zapit-na-pidpisannya-
sertifikata.](#) (дата звернення 21.01.2025).

ДОДАТКИ

Схема архітектури інфраструктури



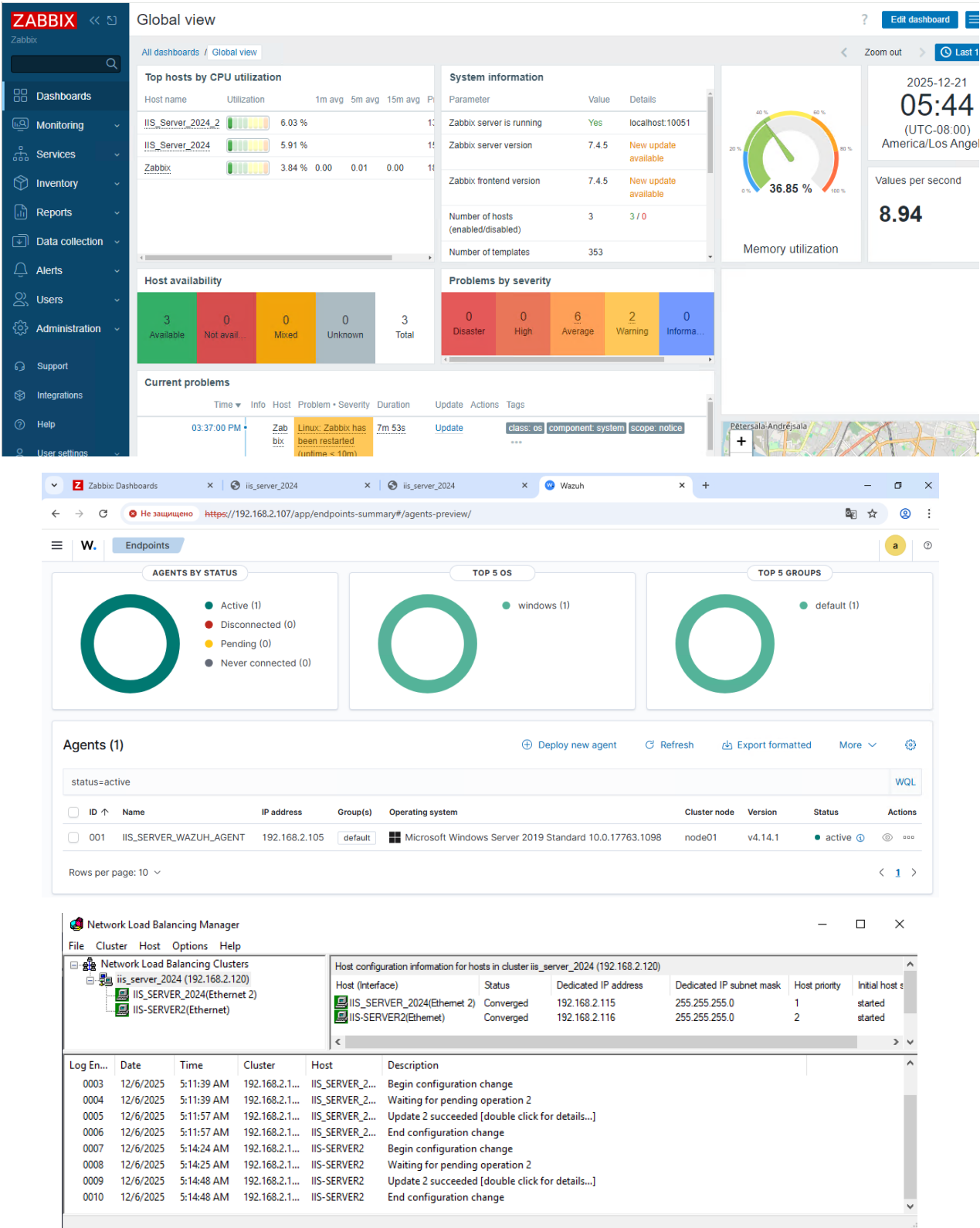
Додаток Б
Таблиця 2.1

Адресація компонентів інфраструктури

Компонент		IP-адреса/префікс	Призначення
Мережа	-	192.168.2.0/24	Приватний діапазон адрес
IIS Server 1	Мережний адаптер	192.168.2.115/24	Первинний вузол WSFC, Primary SQL Replica, IIS Web Server.
	Шлюз за замовчуванням	192.168.2.1/24	
	Основний DNS	192.168.2.119/24	
IIS Server 2	Мережний адаптер	192.168.2.116/24	Вторинний вузол WSFC, Secondary SQL Replica, IIS Web Server.
	Шлюз за замовчуванням	192.168.2.1/24	
	Основний DNS	192.168.2.119/24	
Domain Controller	Мережний адаптер	192.168.2.119/24	Active Directory, DNS Server, Свідок Кворуму (File Share Witness).
	Шлюз за замовчуванням	192.168.2.1/24	
	Основний DNS	127.0.0.1/24	
Zabbix Server	Мережний адаптер	192.168.2.104/24	Хостинг системи системного та

	Шлюз за замовчуванням	192.168.2.1/24	мережевого моніторингу Zabbix Server.
	Основний DNS	192.168.2.119/24	
Wazuh Manager	Мережний адаптер	192.168.2.102/24	Хостинг системи виявлення вторгнень та SIEM Wazuh Manager.
	Шлюз за замовчуванням	192.168.2.1/24	
	Основний DNS	192.168.2.119/24	
NLB Cluster		192.168.2.120/24	Єдина точка входу для клієнтського трафіку (HTTP/HTTPS) до вебсерверів.
WSFC		192.168.2.118/24	Управління кластером Windows Server Failover Cluster (WSFC).
Listener		192.168.2.121/24	Єдина точка підключення для вебдодатка Umbraco до бази даних SQL Server.

Скріншоти, що підтверджують працездатність та коректність налаштування СИСТЕМИ



Failover Cluster Manager

File Action View Help

Failover Cluster Manager

WEB-CLUSTER

Nodes

Disks

Pools

Enclosures

Networks

Cluster Events

Nodes (2)

Search

Queries

NameStatusAssigned VoteCurrent VoteSite

IIS-Server2Up10

IIS_Server_2024Up11

Actions

Nodes

Add Node...

View

Refresh

Help

Umbraco-AG: hosted by IIS_SERVER_2024 (Replica role: Primary)

Last updated: 12/10/2025 4:04:10 AM

Auto refresh: on

Availability group state: Healthy

Primary instance: IIS_SERVER_2024

Failover mode: Automatic

Cluster state: WEB-CLUSTER (Normal Quorum)

Cluster type: Windows Server Failover Cluster

[Start Failover Wizard](#)

[View Always On Health Events](#)

[View Cluster Quorum Information](#)

Availability replica:

Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues
IIS_SERVER_2024	Primary	Synchronous commit	Automatic	Manual	Synchronized	
IIS_SERVER_2024	Secon...	Synchronous commit	Automatic	Manual	Synchronized	

[Add/Remove Columns](#)

Group by

Name	Replica	Synchronization State	Failover Read...	Issues
IIS_SERVER_2024				
MySite	IIS_SERVER_2024	Synchronized	No Data Loss	
IIS-SERVER2				
MySite	IIS-SERVER2	Synchronized	No Data Loss	

[Add/Remove Columns](#)

Site Bindings

Type	Host Name	Port	IP Address	Binding Informa...
https	iis_server_2024	443	*	
http		80	192.168.2.115	
http		80	192.168.2.120	

Add...

Edit...

Remove

Browse

Close