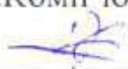


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект автоматизації адміністрування IT
інфраструктури за допомогою Windows Admin Center та
PowerShell DSC»

Виконав: студент 4-го курсу групи КІ-21-2
спеціальності 123 «Комп'ютерна інженерія»

 Даниїл ГЛУХЕНЬКИЙ

Керівник: старший викладач кафедри комп'ютерної
інженерії та кібербезпеки

 Ігор ФАЛЬКОВСЬКИЙ

Рецензент: доцент кафедри комп'ютерної інженерії та
кібербезпеки, кандидат педагогічних наук, доцент

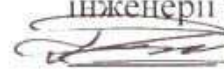
 Олена ГОЛОВНЯ

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко

3 березня 2025 р.

ЗАВДАННЯ

на випускн кваліфікаційну роботу бакалавра

Студент: Глухенький Даниїл Олегович

Тема роботи: «Проект автоматизації адміністрування ІТ інфраструктури за допомогою Windows Admin Center та PowerShell DSC»

затверджена Наказом університету від 3 березня 2025 р. № 79/с

Термін здачі студентом закінченої роботи 9 червня 2025 р.

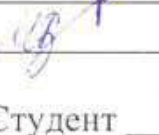
Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

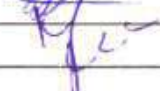
Мережева інфраструктура малої компанії створена на базі симулятора GNSv3 та гіпервізора VirtualBox.

Обладнання та програмне забезпечення:

1. Фізична топологія мережі
2. Графічний симулятор мереж GNSv3
3. Маршрутизатор Cisco 7200
4. Віртуальні машини Windows 10
5. Віртуальні машини Windows Server 2019
6. Операційна система Windows 10

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Ігор Фальковський	 31.03.2025	 31.03.2025
2	Ігор Фальковський	 26.05.2025	 26.05.2025
3	Вадим Миколайчук	 03.06.2025	 03.06.2025

Студент  Даниїл ГЛУХЕНЬКИЙ
Керівник  Ігор ФАЛЬКОВСЬКИЙ

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі	20.02.2025	
2	Пошук, огляд та аналіз аналогічних розробок	20.03.2025	
3	Формулювання технічного завдання	07.04.2025	
4	Опрацювання літературних джерел	20.04.2025	
5	Проектування структури	12.05.2025	
6	Наплаштування системи	16.05.2025	
7	Відлагодження	23.05.2025	
8	Написання пояснювальної записки	27.05.2025	
9	Захист	21.06.2025	

Студент  Даниїл ГЛУХЕНЬКИЙ
Керівник  Ігор ФАЛЬКОВСЬКИЙ

АНОТАЦІЯ

На кваліфікаційну роботу було поставлено завдання: забезпечити автоматизацію адміністрування мережі за допомогою утиліти Windows Admin Center та PowerShell DSC.

Пояснювальна записка до кваліфікаційної роботи складається з трьох розділів, висновків, списку використаної літератури та додатків. Текстова частина викладена на 16 сторінок додатків. Список використаних джерел містить 23 найменувань та займає 3 сторінки. В роботі наведено 80 рисунків. Загальний обсяг роботи - 134 сторінки.

В першому розділі кваліфікаційної роботи проведено аналіз заданих проблем та наведено декілька методів їх вирішення, завдяки яким склалися чіткі уявлення про функціонал проекту.

В другому розділі кваліфікаційної роботи наведено основні алгоритми проектування макетної схеми, які були описані в технічному завданні. Описуються основні методи створення проекту: безпосереднє проектування схеми та налаштування її елементів.

У третій частині кваліфікаційної роботи проводиться тестування та виправлення наявної версії створеного проєкту. Роботу створено на базі програмного забезпечення VirtualBox з використанням емулятора віртуальних мереж GNS v3.

Ключові слова: ПРОЄКТ, МЕРЕЖА, СХЕМА, СИМУЛЯЦІЯ, ВІРТУАЛЬНА МАШИНА, IP, ЗВ'ЯЗОК, НАЛАШТУВАННЯ, СКРИПТ, АДМІНІСТРУВАННЯ, ДОМЕН, СЕРВЕР.

ККІКБ.КР.Б-123-03-ПЗ

Зм.	Арк.	№ докум.	Піпис	Дата	Проект автоматизації адміністрування ІТ інфраструктури за допомогою Windows Admin Center та PowerShell DSC	Літ.	Арк.	Аркуші	
Розроб.	Глухенький Д.О.			16.06.25			4	135	
Перевір.	Фальковський І.Г.			16.06.25					
Керівник	Фальковський І.Г.			16.06.25					
Н. контр.	Фальковський І.Г.			16.06.25					
Зав. каф.	Сфіменко А.А.			16.06.25		ФІКТ Гр. КІ-21-2			

ANNOTATION

The qualification work was assigned the task of automating network administration using the Windows Admin Centre utility and PowerShell DSC.

The explanatory note to the qualification work consists of three chapters, conclusions, a list of references and appendices. The text part is set out on 16 pages of appendices. The list of references contains 23 titles and occupies 3 pages. The paper contains 80 figures. The total volume of the work is 134 pages.

The first chapter of the qualification work analyses the given problems and provides several methods of solving them, which helped to develop a clear understanding of the project's functionality.

The second section of the qualification work presents the basic algorithms for designing a layout scheme, which were described in the terms of reference. The main methods of creating a project are described: direct design of the circuit and configuration of its elements.

In the third part of the qualification work, the existing version of the created project is tested and possible errors are corrected. The work was created on the basis of VirtualBox software using the GNS v3 virtual network emulator.

Keywords: PROJECT, NETWORK, SCHEME, SIMULATION, VIRTUAL MACHINE, IP, COMMUNICATION, CONFIGURATION, SCRIPT, ADMINISTRATION, DOMAIN, SERVER.

		Глухенький Д.О.			ККИКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.Г.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Перелік умовних скорочень

ПК – персональний Комп'ютер

ПЗ – програмне забезпечення

ПД – програмний додаток

WAC – Windows Admin Center

GNS – Graphical Network Simulator-3

VB– Virtual Box

PING – Packet Inter-Network Groper

DSC – Desired State Configuration

AD – Active Directory

RDP –Remote Desktop Protocol

OU – Organization Unit

VPN – Virtual Private Network

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.Г.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗАЦІЇ АДМІНІСТРУВАННЯ ВІРТУАЛЬНИХ МЕРЕЖ	10
1.1 Вступ до автоматизації адміністрування віртуальних мереж	10
1.1.1 Роль автоматизації в управлінні сучасними IT-інфраструктурами: ключові технології.....	10
1.1.2 Визначення основних понять: віртуальні мережі, Windows Admin Center та PowerShell DSC	11
1.2 Технологічні основи windows admin center.....	12
1.2.1 Архітектура та принципи роботи веб-інтерфейсу Windows Admin Center.....	12
1.2.2 Основні функціональні можливості та інтеграції з IT-системами	13
1.2.3 Переваги та обмеження Windows Admin Center: аналіз можливостей.....	14
1.2.4 Порівняльний аналіз Windows Admin Center із аналогами: характеристика ефективності	15
1.3. Power shell desired state configuration (dsc): основи та застосування	16
1.3.1 Принцип роботи DSC: забезпечення бажаного стану інфраструктури.....	17
1.3.2 Порівняння DSC з традиційними скриптовими підходами: переваги та недоліки.....	18
1.3.3 Синергія Windows Admin Center і PowerShell DSC: методи автоматизації адміністрування	20
1.4. Перехід до практичного застосування	21
1.4.1 Роль автоматизації в оптимізації віртуальних мереж: ключові виклики та можливості.....	21
1.4.2 Підготовка до практичного впровадження: планування та інтеграція інструментів.....	22
1.4.3 Вимоги та специфікації проєкту: технічні та функціональні аспекти	23
Висновок першого розділу	26
РОЗДІЛ 2 ПРОЄКТУВАННЯ КОРПОРАТИВНОЇ IPV4-МЕРЕЖІ ОРГАНІЗАЦІЇ	27
2.1 Розробка фізичної топології мережі	27
2.1.1 Вибір і конфігурація комутаторів для забезпечення мережевої інфраструктури	27
2.1.2 Налаштування маршрутизаторів для міжмережевої взаємодії	28
2.1.3 Організація каналів зв'язку та їхня роль у забезпеченні стабільності мережі	30
2.2. Проєктування логічної топології мережі та розробка системи ір-адресації.....	31
2.2.1 Визначення та опис мережевих інтерфейсів.....	31
2.2.2 Розробка схеми розподілу IP-адрес для ефективного використання	32
2.2.3 Налаштування DHCP-сервера для автоматизації розподілу IP-адрес	33
2.3 Інструменти віртуалізації для реалізації мережі	34
2.3.1 Застосування VirtualBox для створення віртуальних машин	34
2.3.1.1 Порівняльний аналіз VirtualBox та Hyper-V: переваги та обмеження	36
2.3.2 Використання Windows Server 2019 як основи для серверних компонентів.....	40
2.3.3 Розгортання Windows 10 на клієнтських пристроях.....	41
2.4. Розробка служб та сервісів прикладного рівня	42
2.4.1 Налаштування Active Directory для управління користувачами та ресурсами	42

2.4.2 Аналіз ризиків і стратегії їхньої мінімізації.....	44
2.5. Проєктування підсистеми мережного захисту	46
2.5.1 Користувачі та базовий захист пристроїв	46
2.5.2 Віддалений доступ	48
2.5.3 Застосування брандмауера для захисту від зовнішніх загроз	49
Висновок до другого розділу	50
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ОЦІНКА СПРОЄКТОВАНО МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ	51
3.1. Вибір обладнання та операційних систем для реалізації мережі	51
3.2 Налаштування DHCP-сервера для автоматизації розподілу IP-адрес	52
3.3 Конфігурація Windows Admin Center для управління інфраструктурою	54
3.3.1 Перевірка функціональності Windows Admin Center за допомогою PowerShell-скрипту	62
3.3.2 Налаштування Role-Based Access Control для доступу до сервісів та DHCP	68
3.4 Автоматизація адміністрування за допомогою PowerShell Desired State Configuration	79
3.4.1 Автоматизація управління користувачами та доступом у домені	79
3.4.2 Автоматизація налаштування мережевих служб	87
3.4.3 Автоматизація заходів безпеки	94
3.4.4 Автоматизація моніторингу та оптимізації роботи системи	102
3.5 Тестування роботи побудованої мережі.....	105
3.5.1 Перевірка налаштувань DHCP через Windows Admin Center	110
Висновок до третього розділу	114
ВИСНОВКИ	115
Список використаних джерел.....	116
ДОДАТКИ.....	119

ВСТУП

У сучасному світі, де час набуває статусу критичного ресурсу, оптимізація його використання стає пріоритетним завданням. Це особливо актуально для корпоративного середовища, де автоматизація адміністрування виступає ключовим елементом ефективного управління сучасною ІТ-інфраструктурою. У цьому контексті програмне забезпечення Windows Admin Center у поєднанні з утилітою PowerShell Desired State Configuration (DSC) є надійними інструментами для забезпечення автоматизованого управління мережевими системами.

Стрімкий розвиток технологій увесь час змінює наше уявлення про світ. Це змушує організації адаптуватися до нових умов, впроваджуючи автоматизацію операційних процесів для підвищення ефективності ведення бізнесу. Використання сучасних інструментів дає змогу ІТ-фахівцям забезпечувати стабільність, надійність і високу продуктивність в управлінні ІТ-інфраструктурою.

Метою кваліфікаційної роботи є ґрунтовне вивчення основних принципів автоматизації адміністрування комп'ютерних мереж, а також практичне застосування цих принципів для побудови мереж у реальних умовах із використанням фізичного обладнання.

Об'єктом дослідження даної кваліфікаційної роботи є процес створення проєкту автоматизації адміністрування ІТ інфраструктури за допомогою Windows Admin Center та PowerShell DSC.

Предметом дослідження виступає мережева інфраструктура малої компанії та її функціональні можливості: взаємозв'язок між пристроями, підтримка віддаленого доступу, впровадження автоматизованого управління процесами та здійснення системного моніторингу.

Після визначення об'єкта та предмета дослідження здійснюється аналіз і проєктування відповідної мережевої структури з метою створення повноцінного та функціонального проєкту, який можна застосувати для налаштування та автоматизації адміністрування реальної мережі малої компанії.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗАЦІЇ АДМІНІСТРУВАННЯ ВІРТУАЛЬНИХ МЕРЕЖ

1.1 Вступ до автоматизації адміністрування віртуальних мереж

Автоматизація адміністрування відіграє ключову роль у сучасному управлінні віртуальними мережами, дозволяючи зменшити кількість рутинних дій і підвищити ефективність роботи ІТ-персоналу. Завдяки використанню спеціалізованих інструментів, таких як PowerShell DSC, Windows Admin Center, та інші можливо централізовано контролювати, оновлювати та підтримувати інфраструктуру. Це особливо важливо у віртуальних середовищах, де динаміка змін вимагає швидких і точних рішень.

1.1.1 Роль автоматизації в управлінні сучасними ІТ-інфраструктурами: ключові технології

Як вже було сказано вище автоматизація є основою ефективного управління сучасними ІТ-інфраструктурами, особливо в умовах швидкого зростання віртуальних мереж. Вона дозволяє організаціям оптимізувати ресурси, скоротити час на виконання рутинних завдань і мінімізувати людські помилки. У цьому контексті ключові технології автоматизації відіграють вирішальну роль, забезпечуючи централізоване управління та швидке реагування на зміни в інфраструктурі.

Одним із провідних інструментів є Windows Admin Center (WAC) - веб-інтерфейс, який надає адміністраторам зручний доступ до управління серверами, клієнтськими системами і віртуальними середовищами. WAC підтримує інтеграцію з різними Windows-сервісами, дозволяючи автоматизувати задачі, такі як оновлення систем чи моніторинг продуктивності. Ще однією важливою технологією є PowerShell Desired State Configuration (DSC), яка забезпечує декларативний підхід до конфігурації систем, дозволяючи задавати бажаний стан інфраструктури та автоматично підтримувати його.

Поряд із WAC і DSC, сучасні ІТ-інфраструктури використовують системи оркестрації, такі як Ansible чи Puppet, які автоматизують розгортання

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

програмного забезпечення та управління конфігураціями на великих масштабах. Хмарні платформи, наприклад Microsoft Azure Automation, дозволяють інтегрувати автоматизацію в хмарні середовища, забезпечуючи гнучкість і масштабованість. Також варто відзначити інструменти моніторингу, такі як System Center Operations Manager (SCOM), які автоматизують виявлення проблем і сповіщення адміністраторів.

Застосування цих технологій сприяє підвищенню ефективності адміністрування, дозволяючи ІТ-фахівцям зосередитися на стратегічних задачах, а не на рутинних операціях. Таким чином, автоматизація стає невід'ємною частиною управління сучасними ІТ-інфраструктурами, забезпечуючи їхню надійність і конкурентоспроможність.

1.1.2 Визначення основних понять: віртуальні мережі, Windows Admin Center та PowerShell DSC

Даний розділ має на меті надати чітке розуміння ключових термінів, що лежать в основі теми проєкту. Віртуальні мережі - це логічно розділені мережеві структури, які функціонують поверх фізичної мережі, дозволяючи ефективно керувати підключеннями, ізолювати трафік та масштабувати інфраструктуру без додаткового обладнання. Windows Admin Center (WAC) - це сучасний веб-інтерфейс від Microsoft, призначений для централізованого адміністрування серверів, клієнтських машин та кластерів, що поєднує низку інструментів керування в одному рішенні без потреби в окремих консолях. WAC забезпечує зручний доступ до таких функцій, як управління службами, журналами подій, мережевими налаштуваннями та ролями серверів. PowerShell Desired State Configuration (DSC) - це потужна технологія конфігураційного управління, яка дозволяє автоматизувати налаштування систем і підтримувати їх у заданому стані. Завдяки DSC адміністратори можуть створювати декларативні конфігурації, які легко масштабуються та повторно використовуються для уніфікації середовища в межах організації.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Загальне розуміння цих ключових понять є необхідною основою для подальшого аналізу та практичного застосування інструментів автоматизації в IT-інфраструктурі. Їх поєднання забезпечує ефективне, гнучке та безпечне адміністрування сучасних віртуалізованих мережових середовищ.

1.2 Технологічні основи windows admin center

Даний розділ присвячений аналізу технологічної бази Windows Admin Center як інструменту для централізованого управління IT-інфраструктурою. Нижче буде розглянуто його архітектуру, принципи роботи та ключові можливості, що роблять WAC ефективним рішенням для адміністраторів.

1.2.1 Архітектура та принципи роботи веб-інтерфейсу Windows Admin Center

Windows Admin Center (WAC) має модульну архітектуру, що ґрунтується на веб-технологіях і забезпечує зручне централізоване управління серверами, комп'ютерами та мережевою інфраструктурою. Основу WAC становить веб-сервер, який запускається локально на керуючій машині або сервері й надає адміністраторам доступ через браузер. Комунікація з цільовими пристроями відбувається через PowerShell, WMI та WinRM, що дозволяє WAC взаємодіяти з різними компонентами Windows без встановлення додаткових агентів на керовані машини. Завдяки такому підходу адміністратор отримує повний набір функцій – від керування ролями та службами до моніторингу системи – у єдиному інтерфейсі.

Принципи роботи WAC побудовані на безагентній моделі, що спрощує впровадження та забезпечує безперервне оновлення компонентів без втручання користувача. Важливою складовою архітектури є підтримка розширень, які дозволяють інтегрувати нові функції та сторонні рішення, розширюючи можливості інструмента без зміни його ядра. WAC також забезпечує безпечне з'єднання з використанням HTTPS, можливість ролевого доступу та інтеграцію з Azure, що відкриває можливості для гібридного адміністрування. Усе це робить

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Windows Admin Center гнучким та масштабованим рішенням для сучасних ІТ-середовищ.

1.2.2 Основні функціональні можливості та інтеграції з ІТ-системами

Windows Admin Center (WAC) надає потужний набір функціональних можливостей, які охоплюють широкий спектр завдань, пов'язаних з адмініструванням серверів, робочих станцій та мережевих компонентів в інфраструктурі підприємства. Інструмент дозволяє здійснювати централізоване управління як окремими пристроями, так і цілими групами систем у реальному часі.

До ключових можливостей відносяться:

- керування ролями та функціями Windows Server,
- відстеження системних ресурсів,
- моніторинг продуктивності процесора, оперативної пам'яті, мережевої активності та накопичувачів.

Крім цього, адміністратор може керувати службами, процесами, оновленнями, переглядати та аналізувати журнали подій, керувати локальними та віддаленими дисками, а також налаштовувати файлові системи. Також підтримується віддалене керування робочим столом, конфігурація мережевих інтерфейсів, брандмауерів, облікових записів і прав доступу, що забезпечує повноцінний контроль за всіма аспектами ІТ-інфраструктури з одного веб-інтерфейсу.

Окремої уваги заслуговує інтеграція WAC з іншими ІТ-системами та службами, що робить його особливо корисним у змішаних середовищах. Інструмент має тісну інтеграцію з Active Directory для централізованого управління ідентифікацією та доступом, а також підтримує запуск і використання скриптів PowerShell безпосередньо з інтерфейсу. Серед підтримуваних компонентів: Hyper-V, VirtualBox, засоби для управління віртуальними машинами, Failover Clustering для високої доступності, Storage Spaces Direct для побудови масштабованих кластерів зберігання.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Завдяки підтримці підключення до Microsoft Azure, Windows Admin Center дозволяє використовувати гібридні функції, такі як Azure Backup, Azure Monitor, Azure Update Management і Defender for Cloud. Це значно підвищує рівень безпеки, резервування та моніторингу в інфраструктурі. Також WAC підтримує систему розширень (extensions), що дає змогу встановлювати додаткові модулі відповідно до специфіки підприємства, роблячи платформу максимально адаптивною, масштабованою та відкритою до розвитку нових функцій у майбутньому.

1.2.3 Переваги та обмеження Windows Admin Center: аналіз можливостей

Windows Admin Center (WAC) є сучасним інструментом для централізованого управління інфраструктурою Windows, який має низку важливих переваг.

Передусім, його головна перевага полягає в наявності зручного та інтуїтивного веб-інтерфейсу, що дозволяє адміністраторам отримати доступ до більшості функцій Windows Server без потреби використовувати окремі консолі чи підключатись через віддалений робочий стіл. WAC об'єднує інструменти для моніторингу, конфігурації, оновлення, управління службами, дисками, мережами та подіями - усе в єдиному вікні. Це значно скорочує час на обслуговування систем і підвищує загальну ефективність ІТ-персоналу. Крім того, WAC не потребує встановлення агентів на керованих машинах, що спрощує впровадження та мінімізує вплив на продуктивність систем. Інтеграція з хмарними сервісами Azure забезпечує можливості гібридного адміністрування, дозволяючи поєднати локальні сервери з хмарною інфраструктурою.

Попри численні переваги, існують і певні обмеження, які варто враховувати під час впровадження Windows Admin Center. Найбільш помітне з них - підтримка лише операційних систем на базі Windows, що унеможливорює використання WAC для адміністрування Linux-серверів або пристроїв з іншими ОС. Крім того, для повноцінної роботи інструменту потрібно налаштувати

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

WinRM та відповідні дозволи, що може бути непростим завданням у складних середовищах. Деякі специфічні функції, такі як повне керування груповими політиками (GPO) або розширене налаштування ролей, все ще потребують запуску окремих класичних MMC-консолей. Також у середовищах з великою кількістю серверів можуть виникати затримки або зниження продуктивності інтерфейсу. Важливо зазначити, що розвиток Windows Admin Center триває, і з кожною новою версією функціональність розширюється, однак на момент впровадження слід ретельно враховувати як поточні можливості, так і обмеження цього інструменту.

1.2.4 Порівняльний аналіз Windows Admin Center із аналогами: характеристика ефективності

Для оцінки ефективності Windows Admin Center доцільно буде порівняти його з аналогічними інструментами, такими як System Center Configuration Manager (SCCM), Ansible і Puppet. Кожен із даних інструментів має свої особливості, які впливають на їхнє використання в різних сценаріях адміністрування.

Наприклад, SCCM є потужним рішенням від Microsoft, призначеним для управління великими мережами з тисячами вузлів. На відміну від WAC, SCCM вимагає ліцензування, що робить його дорожчим, проте сервіс надає розширені можливості, такі як розгортання програмного забезпечення та управління оновленнями на масштабованому рівні.

Ansible, інструмент із відкритим кодом, що збільшує його рівень підтримки спільнотою. Ansible підтримує кросплатформність, що робить його ідеальним для гетерогенних середовищ із Windows і Linux-системами.

Puppet - це ще один популярний інструмент для адміністрування, він також підтримує кросплатформність, проте його Enterprise-версія є платною, що може стати ключовим фактором при виборі програмного забезпечення для адміністрування малої мережі.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Нижче наведемо таблицю порівняння Windows Admin Center із зазначеними аналогами за низкою ключових характеристик:

Таблиця 1. Порівняння з аналогами

Критерій	Windows Admin Center	System Center Configuration Manager (SCCM)	Ansible	Puppet
Платформа	Windows	Windows	Кросплатформність (Windows, Linux)	Кросплатформність (Windows, Linux)
Тип доступу	Веб-інтерфейс	Десктопний додаток	Командний рядок/ Playbook	Командний рядок/ Маніфести
Легкість налаштування	Висока (графічний інтерфейс)	Середня (вимагає зрозуміння)	Середня (вимагає знання YAML)	Середня (вимагає знання Ruby)
Інтеграція	Windows-сервіси, Azure	Windows-сервіси, Azure	Широка (SSH, API)	Широка (агенти, API)
Масштабованість	Середня (до 100 вузлів)	Висока (тисячі вузлів)	Висока (тисячі вузлів)	Висока (тисячі вузлів)
Вартість	Безкоштовно	Платно (ліцензія Microsoft)	Безкоштовно (з платними модулями)	Платно (Enterprise-версія)
Підтримка спільноти	Середня	Висока	Висока	Висока

Порівняльний аналіз показує, що Windows Admin Center є оптимальним рішенням для малих і середніх мереж із Windows-системами, де важливі простота використання та економія. Однак для великих або кросплатформних інфраструктур Ansible і Puppet можуть бути кращими завдяки своїй масштабованості та універсальності. SCCM, у свою чергу, більше підходить для великих організацій, які вже інтегровані в екосистему Microsoft.

1.3. Power shell desired state configuration (dsc): основи та застосування

Даний розділ присвячено огляду технології PowerShell Desired State Configuration (DSC). Це - потужний інструмент для автоматизації налаштування та підтримки конфігурацій комп'ютерних систем у заданому стані. Його використання значно полегшує виконання нашого проєкту, оскільки дозволяє

мінімізувати ручну працю, забезпечити стабільність конфігурацій та швидко відновлювати систему до потрібного стану в разі змін або збоїв.

1.3.1 Принцип роботи DSC: забезпечення бажаного стану інфраструктури

PowerShell Desired State Configuration (DSC) є потужним інструментом автоматизації, розробленим Microsoft, який базується на декларативному підході до управління конфігурацією IT-інфраструктури. Принцип роботи DSC полягає в тому, щоб визначити бажаний стан систем і забезпечити його підтримку, незалежно від поточного стану. Замість того, щоб надавати детальні інструкції для кожної операції, адміністратор описує кінцеву конфігурацію (наприклад, встановлені програми, налаштування мережі чи параметри безпеки), а DSC автоматично застосовує необхідні зміни для досягнення цього стану.

Архітектура DSC включає три основні компоненти:

- конфігураційні скрипти,
- локального агента конфігурації (Local Configuration Manager (далі «LCM»)),
- ресурси DSC.

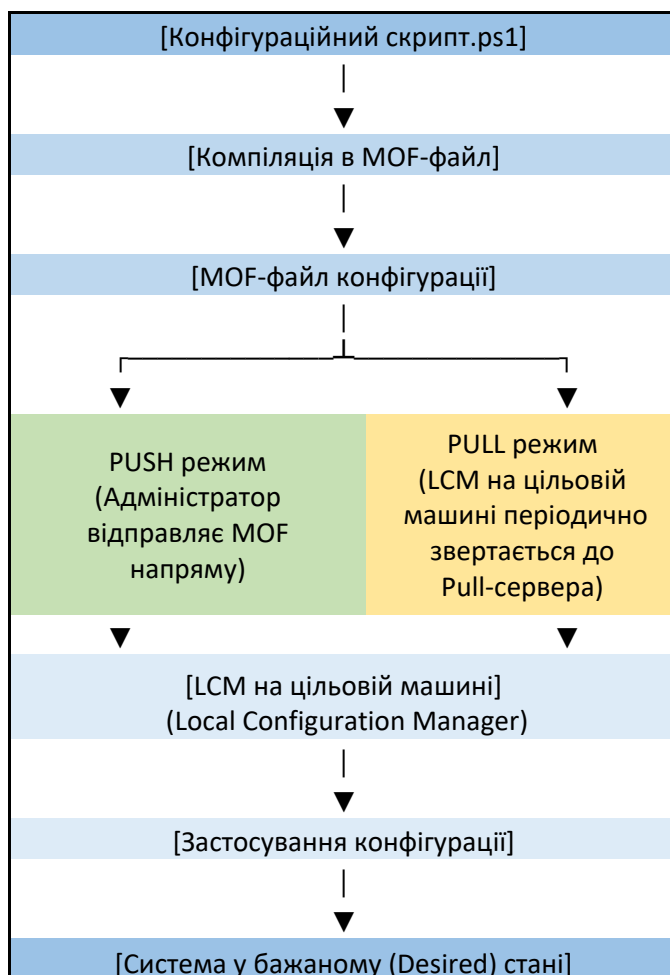
Конфігураційні скрипти створюються за допомогою PowerShell і визначають бажаний стан у вигляді блоків конфігурації. LCM, який є частиною кожної керованої системи, відповідає за виконання цих скриптів і періодичну перевірку відповідності поточного стану бажаному. Ресурси DSC - це модулі, які містять логіку для управління конкретними аспектами системи, такими як файли, реєстр чи служби.

Процес роботи DSC починається з написання конфігурації, яка компілюється в MOF-файли (Managed Object Format). Ці файли передаються на цільові вузли, де LCM аналізує їх і застосовує зміни. LCM також може працювати в режимі "Pull" або "Push": у режимі "Pull" вузли періодично отримують оновлення з центрального сервера, а в режимі "Push" конфігурація розгортається

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				17
Змн.	Арк.	№ докум.	Підпис	Дата		

централізовано адміністратором. Нижче прикріплена Таблиця 2, яка наглядно демонструє увесь процес.

Таблиця 2. Процес створення скрипта



Такий підхід забезпечує гнучкість і масштабованість, дозволяючи управляти як окремими машинами, так і великими мережами.

1.3.2 Порівняння DSC з традиційними скриптовими підходами: переваги та недоліки

PowerShell Desired State Configuration (DSC) відрізняється від традиційних скриптових підходів, які базуються на імперативному програмуванні, де адміністратор визначає послідовність команд для досягнення бажаної конфігурації. Традиційні PowerShell-скрипти виконують дії крок за кроком, наприклад, видаляючи файли, встановлюючи програми чи змінюючи параметри, що вимагає детального опису кожного етапу. На відміну від цього, DSC

використовує декларативний підхід, де адміністратор вказує лише кінцевий стан (наприклад, "служба запущена" чи "файл присутній"), а механізм DSC самостійно визначає, які дії потрібні для його досягнення.

Однією з ключових відмінностей є **спосіб обробки змін**. Традиційні скрипти виконуються одноразово і не стежать за станом системи після завершення, тоді як DSC періодично перевіряє відповідність поточного стану бажаному і вносить корективи при необхідності. Наприклад, якщо служба була зупинена вручну, DSC автоматично її перезапустить, тоді як традиційний скрипт цього не зробить без повторного запуску. Така функція "self-healing" робить DSC більш надійним для довготривалого управління.

Ще одна перевага DSC полягає в **модульності та повторному використанні**. Конфігурації DSC можна зберігати як шаблони та застосовувати до різних вузлів, тоді як традиційні скрипти часто вимагають модифікації для кожного нового сценарію. Наприклад, ресурс DSC може бути налаштований для управління мережевими параметрами на кількох серверах, тоді як традиційний скрипт потрібно адаптувати для кожної машини.

Однак, традиційні скрипти мають свої сильні сторони:

Вони **простіші для написання** та відлагодження в невеликих проєктах, де немає потреби в постійному контролі стану. Крім того, скрипти дозволяють виконувати складні логічні операції, які можуть бути обмеженими в DSC через його декларативну природу. Наприклад, традиційний скрипт може легко обробляти умовні оператори чи циклічні дії, тоді як DSC потребує додаткових ресурсів або конфігурацій для подібних задач.

Важливим аспектом є **продуктивність**. Традиційні скрипти зазвичай виконуються швидше, оскільки не включають механізми періодичної перевірки, як у DSC. Проте в масштабних середовищах, де важлива стабільність, витрати на періодичну перевірку DSC окупаються завдяки зменшенню ручного втручання. У віртуальних мережах, таких як мережа нашого проєкту, DSC є особливо корисним для забезпечення узгодженості конфігурацій між віртуальними машинами.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Вибір між DSC і традиційними скриптовими підходами залежить від конкретних потреб інфраструктури. DSC є ідеальним для автоматизації з акцентом на стабільність і масштабність, тоді як традиційні скрипти краще підходять для одноразових або специфічних задач, що не потребують постійного контролю.

1.3.3 Синергія Windows Admin Center і PowerShell DSC: методи автоматизації адміністрування

Окремо Windows Admin Center та PowerShell DSC є потужними інструментами для адміністрування IT-інфраструктури, кожен із яких виконує свою важливу роль у забезпеченні стабільності, безпеки та керованості систем. Проте, їх поєднання дозволяє значно підвищити ефективність управління, забезпечуючи як централізований моніторинг і керування так і автоматизоване налаштування та підтримку конфігурації.

Синергія Windows Admin Center і PowerShell Desired State Configuration створює ефективний інструментарій для автоматизації адміністрування IT-інфраструктури. WAC забезпечує графічний веб-інтерфейс для управління системами, тоді як DSC додає декларативну логіку для забезпечення бажаного стану, що дозволяє комбінувати зручність і автоматизацію. Ця інтеграція дозволяє адміністраторам як візуально контролювати інфраструктуру, так і автоматизувати складні процеси.

Можна виділити два основних підходи до синергії цих інструментів:

- Через PowerShell
- Через WAC

Одним із ключових підходів до синергії є використання WAC для розгортання та моніторингу конфігурацій DSC. WAC підтримує вбудований модуль DSC, який дозволяє створювати, редагувати та застосовувати конфігурації безпосередньо через веб-інтерфейс. Наприклад, адміністратор може налаштувати відключення USB-портів на кількох робочих станціях,

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				20
Змн.	Арк.	№ докум.	Підпис	Дата		

використовуючи DSC, а WAC забезпечить відображення стану виконання та звітів про помилки.

Інший підхід полягає в інтеграції через PowerShell. WAC дозволяє виконувати PowerShell-команди, які можуть активувати або оновлювати конфігурації DSC на цільових вузлах. Це особливо корисно в гібридних середовищах, де WAC керує як локальними, так і хмарними ресурсами, а DSC забезпечує узгодженість конфігурацій. Наприклад, можна налаштувати автоматичне розгортання оновлень у Azure через WAC із використанням DSC для підтримки їхнього стану.

Синергія також проявляється в спрощенні управління масштабними мережами. WAC може централізувати доступ до кількох серверів, а DSC розподіляє конфігурації між ними, забезпечуючи їхню однорідність. У контексті віртуальних мереж це дозволяє швидко застосовувати однакові політики безпеки чи мережеві налаштування до всіх віртуальних машин. Однак ефективність цієї синергії залежить від правильної конфігурації обох інструментів. Наприклад, WAC і DSC потребують активованого WinRM, а також узгоджених прав доступу для адміністраторів. Неправильне налаштування може призвести до збоїв у синхронізації або моніторингу.

Отже, поєднання WAC і DSC забезпечує комплексний підхід до автоматизації адміністрування, об'єднуючи графічний контроль із декларативною логікою. Ця інтеграція є особливо цінною для віртуальних мереж, де потрібна висока автоматизація та ефективне управління ресурсами.

1.4. Перехід до практичного застосування

1.4.1 Роль автоматизації в оптимізації віртуальних мереж: ключові виклики та можливості

Як зазначалося раніше автоматизація відіграє критичну роль в оптимізації віртуальних мереж, оскільки дозволяє зменшити вплив людського фактору, забезпечити стандартизоване розгортання налаштувань та пришвидшити реагування на інфраструктурні зміни.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Одним із ключових завдань є підтримка стабільності та безпеки мереж при мінімальних витратах часу та ресурсів. Автоматизовані засоби дають змогу централізовано впроваджувати політики, контролювати стан систем, розгортати оновлення й резервне копіювання без постійного ручного втручання.

Водночас впровадження таких рішень потребує продуманої підготовки: налаштування інструментів, навчання персоналу, а також забезпечення сумісності з наявною інфраструктурою. Проте усі ці зусилля виправдані, адже автоматизація дає змогу оптимізувати ресурси, підвищити продуктивність ІТ-персоналу та забезпечити масштабованість системи в довгостроковій перспективі. Таким чином, вона виступає не лише засобом підвищення ефективності, а й стратегічним елементом в управлінні сучасними віртуальними мережами.

1.4.2 Підготовка до практичного впровадження: планування та інтеграція інструментів

Підготовка до практичного впровадження автоматизації адміністрування віртуальних мереж є критичним етапом, який визначає успішність подальшого застосування технологій. Цей процес починається з аналізу потреб інфраструктури, що дозволяє визначити ключові задачі, які потребують автоматизації, та розробити загальну стратегію їхнього вирішення. Планування включає оцінку наявних ресурсів, визначення цільових систем і створення структури, яка забезпечить ефективне використання інструментів адміністрування.

Інтеграція інструментів є невід'ємною частиною підготовки, де Windows Admin Center і PowerShell Desired State Configuration відіграють взаємодоповнюючі ролі. Ця комбінація вимагає узгодження їхніх технічних компонентів, таких як протоколи зв'язку та механізми доступу, що має бути ретельно сплановано для забезпечення сумісності.

Процес інтеграції також передбачає розробку плану взаємодії між інструментами, включаючи визначення ролей адміністраторів і підготовку

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

базових конфігурацій, які будуть адаптовані до специфіки інфраструктури. Важливим є врахування потенційних викликів, таких як безпека доступу чи обмеження мережевих ресурсів, щоб мінімізувати ризики на етапі впровадження. Такий підхід дозволяє створити міцну основу для практичного застосування, готуючи інфраструктуру до автоматизованого управління.

Завершальний етап підготовки полягає в тестуванні інтеграції на теоретичному рівні, що дозволяє виявити можливі недоліки без прямого впливу на робочі системи. Цей процес закладає основу для переходу до практичного етапу, де будуть реалізовані та перевірені розроблені стратегії, вказуючи значне підвищення ефективності адміністрування в майбутньому.

1.4.3 Вимоги та специфікації проєкту: технічні та функціональні аспекти

Мета створення проєкту

Розробка проєкту обґрунтована необхідністю виконання завдання в рамках кваліфікаційної роботи, що передбачає створення рішення для автоматизації адміністрування мережевих систем.

Цільове призначення системи

Основна мета проєкту полягає в розробці системи, яка забезпечить автоматизоване управління мережею через інтеграцію інструментів Windows Admin Center (WAC) і PowerShell Desired State Configuration (DSC), підвищуючи ефективність адміністрування та оптимізуючи операційні процеси.

Функціональні та технічні характеристики проєкту

Для реалізації проєкту необхідно забезпечити виконання наступних умов:

- Мережева взаємодія: Система має гарантувати стабільне з'єднання між усіма компонентами мережі, забезпечуючи їхню безперебійну взаємодію.
- Основні завдання проєкту:
 1. Використання Windows Admin Center як основного інструменту управління.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				23
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Забезпечення автоматизації управління інфраструктурою з використанням PowerShell DSC для створення та підтримки конфігурацій.
 3. Реалізація механізмів безпеки, включаючи автентифікацію користувачів, шифрування даних і розмежування рівнів доступу для захисту мережевих ресурсів.
 4. Сумісність із наявною інфраструктурою, що дозволяє інтегрувати систему з уже розгорнутим обладнанням і програмним забезпеченням.
 5. Зручність використання завдяки інтуїтивно зрозумілому інтерфейсу, який спрощує налаштування та адміністрування мережі.
 6. Моделювання мережевої схеми, максимально наближеної до реального прототипу, для точного відтворення робочого середовища.
 7. Наявність тестового вузла, який використовуватиметься для перевірки створених користувачів і віддаленого тестування автоматизованих скриптів.
- Гнучкість застосування: Проєкт має бути адаптованим для автоматизації мереж у різних локальних середовищах, таких як навчальні заклади, офіси чи підприємства.

Системні вимоги для роботи проєкту

Для забезпечення коректного функціонування системи необхідно відповідати таким мінімальним технічним характеристикам:

- Процесор: Intel Core i5 або еквівалент.
- Оперативна пам'ять: 16 ГБ.
- Вільне місце на диску: 100 ГБ (HDD або SSD).
- Графічний адаптер: вбудована Intel HD Graphics.
- Операційна система: Windows 10, Windows 11 або Windows Server 2019 і новіші версії.

Очікування від інтерфейсу

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс системи має бути гнучким і адаптивним, дозволяючи користувачу налаштовувати розташування елементів, змінювати назви та інші параметри відповідно до своїх потреб. Важливо, щоб він реалістично відображав мережеве середовище, забезпечуючи точну візуалізацію пристроїв і їхньої поведінки. Користувач повинен мати можливість налаштовувати параметри мережевих пристроїв, щоб проводити детальний аналіз і тестування мережі.

Вимоги до сумісності

Система має бути сумісною з операційними системами Windows 10, 11, а також Windows Server 2019 і новішими версіями. Для моделювання мережі необхідно встановити GNS3 версії 3.0 або вище, а також VirtualBox версії 6.0 або новіший. На цьому етапі додаткові специфічні вимоги до програмної сумісності відсутні.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок першого розділу

Перший розділ кваліфікаційної роботи присвячений теоретичним основам автоматизації адміністрування віртуальних мереж, що є основою для подальшого практичного дослідження. Розглянуто роль автоматизації в управлінні сучасними IT-інфраструктурами, підкреслено ключові технології, такі як Windows Admin Center і PowerShell DSC, які забезпечують ефективне рішення для централізованого контролю. Аналіз архітектури, функціональних можливостей і порівняння WAC із аналогами виявив його сильні сторони, такі як зручність використання та безкоштовність, а також обмеження, пов'язані з масштабованістю.

Дослідження принципів роботи DSC і його відмінностей від традиційних скриптів підкреслило переваги декларативного підходу в забезпеченні стабільного стану інфраструктури. Синергія WAC і DSC продемонструвала потенціал для комплексної автоматизації, особливо в контексті віртуальних мереж. Завершальний підпункт визначив вимоги до проекту, що є логічним переходом до практичного впровадження, окресливши технічні та функціональні аспекти для успішної реалізації. Таким чином, теоретичний розділ закладає міцну основу для подальшого аналізу та практичних експериментів, спрямованих на оптимізацію адміністрування а зібрані теоретичні знання стали підґрунтям для розробки й реалізації практичного рішення з автоматизації адміністрування мережевої інфраструктури.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТУВАННЯ КОПРОРАТИВНОЇ IPV4-МЕРЕЖІ ОРГАНІЗАЦІЇ

2.1 Розробка фізичної топології мережі

У розділі розробки фізичної топології мережі розглядається створення фізичної структури мережі. Вона включає в себе: вибір та розміщення мережевого обладнання, підключення образів віртуальних машин, під'єднання маршрутизаторів та інших пристроїв для забезпечення ефективного функціонування мережевої інфраструктури.

2.1.1 Вибір і конфігурація комутаторів для забезпечення мережевої інфраструктури

Для побудови віртуальної мережі у даному проєкті буде використано інструмент Virtual Private Network version 3 (GNSv3), який реалізує концепцію віртуальної приватної мережі. Хоча GNS3 не містить вбудованих моделей комутаторів за замовчуванням, він підтримує інтеграцію зовнішніх IOS-образів, зокрема від Cisco, а також роботу з віртуальними комутаторами. Cisco IOU (IOS on Unix) або IOL (IOS on Linux) одні із багатьох видів комутаторів, які можна завантажити та інтегрувати в систему GNSv3. Вони забезпечують підтримку широкого спектра функцій комутації, що дозволяє створювати повноцінні симуляції мережевої інфраструктури. Варто відмітити такий пристрій як EtherSwitch router. Це віртуальний пристрій із функціями комутатора та маршрутизатора, використовуючи Cisco IOS. Цей пристрій дозволяє симулювати основні функції комутації, такі як VLAN, STP та інші, разом з маршрутизацією на базі одного образу IOS. EtherSwitch слугує гарним прикладом для аналізу, проте в даній роботі будуть використані інші технології які будуть розглянуті нижче.

Серед реального обладнання, яке може бути змодельоване або взяте за основу конфігурації, варто відзначити серії Cisco Catalyst 2960, 3560 і 3750. Catalyst 2960 - це комутатор рівня 2, який виконує базові завдання доступу, підтримуючи VLAN, STP та інші протоколи комутації. Catalyst 3560 і 3750 - це вже пристрої

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		27

рівня 3 з розширеною підтримкою маршрутизації між VLAN, що робить їх придатними для роботи як на рівні доступу, так і в ядрах корпоративних мереж.

Вибір конкретної моделі комутатора залежить від поставлених завдань: для невеликих мереж доцільно використовувати простіші моделі з обмеженим набором функцій, тоді як складніші топології потребують потужних пристроїв із підтримкою більшої кількості протоколів та механізмів керування трафіком.

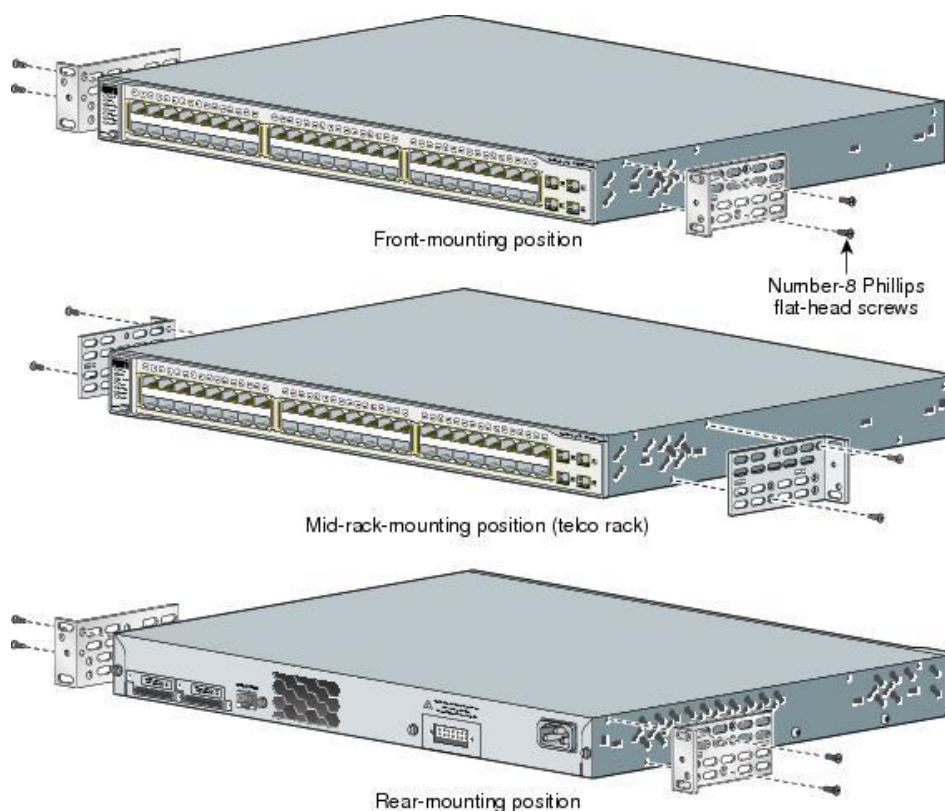


Рис. 1 Комутатор Cisco Catalyst 3750

2.1.2 Налаштування маршрутизаторів для міжмережевої взаємодії

Так само як і з комутаторами GNSv3 не надає велику кількість різних маршрутизаторів для роботи, але їх образи також можна знайти в мережі Інтернет і завантажити їх у програму. Для прикладу розглянемо декілька конкретних образів маршрутизаторів які нам пропонує Cisco.

Кожна модель маршрутизатора має свої характеристики та функціональні переваги. Наприклад, Cisco 3725 є пристроєм середнього класу, який підтримує основні функції маршрутизації та базову комутацію. Завдяки універсальності та надійності, він часто використовується в тестових середовищах для симуляції типових сценаріїв взаємодії мережевих пристроїв.

		Глухенький Д.О			KKIB.KP.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Особливої уваги заслуговує маршрутизатор Cisco 7200, який у контексті даного проєкту є одним з найкращих варіантів. Цей пристрій вирізняється гнучкістю, підтримкою великої кількості мережевих протоколів та високим рівнем функціональності, що робить його одним із найпоширеніших виборів серед фахівців для створення комплексних мережевих топологій у GNS3. Він підтримує такі протоколи, як OSPF, EIGRP, BGP, RIP, а також мультикастові протоколи, що критично важливо для побудови масштабованих і динамічних мереж.

Окрім базових можливостей маршрутизації, Cisco 7200 також забезпечує вбудовані механізми безпеки, включаючи ACL, VPN, IPsec та firewall-функції, що дозволяє ефективно захищати мережу від зовнішніх загроз та несанкціонованого доступу. Ще однією важливою перевагою цієї моделі є можливість використання модулів розширення, які дозволяють адаптувати пристрій під специфічні потреби, наприклад, додати інтерфейси для підключення до інших типів мереж або реалізувати підтримку голосових сервісів.

Загалом, Cisco 7200 є потужним та надійним симуляційним маршрутизатором, який здатний задовольнити потреби навіть складних мережевих проєктів. Його продуктивність, масштабованість та широкий функціонал роблять його логічним вибором для проєктування віртуальної мережі в рамках цього проєкту.



Рис. 2 Маршрутизатор Cisco 7200

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		29

2.1.3 Організація каналів зв'язку та їхня роль у забезпеченні стабільності мережі

Канали зв'язку є ключовим елементом будь-якої мережевої інфраструктури, оскільки забезпечують передачу даних між пристроями. Це фізичні або бездротові засоби передачі даних, які дозволяють обмінюватися інформацією між вузлами мережі.

Існує декілька різних типів каналів зв'язку між вузлами, які використовуються в мережевих системах. Коротко оглянемо кожен з них:

- Бездротові технології: Здійснюють передачу даних без використання фізичних кабелів, використовуючи радіохвилі, інфрачервоне випромінювання, або інші бездротові технології. Безпроводові технології включають Wi-Fi, Bluetooth, NFC та інші.
- Супутниковий зв'язок: Використовує комунікаційні супутники для передачі даних між вузлами мережі. Даний тип зв'язку особливо корисний у віддалених районах або у випадках, коли пряма мідна або оптична кабельна передача неможлива.
- Кабельна передача: Базується на використанні мідних або волоконно-оптичних кабелів для передачі даних. Мідний кабель працює шляхом передачі електричного сигналу, тоді як волоконно-оптичний кабель подає дані у вигляді світлових сигналів, що забезпечує вищу швидкість та менші витрати.
- Комутовані мережі: Використовується для передачі даних через телефонну мережу, або інші види комутованих мереж. Цей тип зв'язку часто використовується для підключення до Інтернету через модеми або DSL-лінії.

Кожен тип каналу зв'язку має свої переваги та обмеження, проте вибір конкретного типу каналу зв'язку залежить від потреб проєкту, фізичних обмежень, вартості та інших факторів.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

2.2. Проектування логічної топології мережі та розробка системи ір-адресації

Наступним кроком у реалізації проєкту є розробка оптимальної структури мережі та налаштування системи IP-адресації, що забезпечить ефективну взаємодію між усіма мережевими пристроями. У цьому розділі буде проаналізовано ключові фактори, які впливають на вибір мережевої топології та схеми розподілу IP-адрес, з метою формування надійного і функціонального мережевого середовища, що відповідає вимогам організації.

2.2.1 Визначення та опис мережевих інтерфейсів

Для створення віртуальної мережі необхідно спочатку розробити теоретичний макет, якому буде відповідати кінцева версія нашої схеми. Розглянувши теоретичні відомості про наявні маршрутизатори та комутатори можна створити макет схеми у вигляді описової таблиці IP-адресації. Одразу додамо в неї віртуальні машини які будуть детально розглянуті в РОЗДІЛІЗ, у якому буде проведено повне налаштування нашої віртуальної комп'ютерної мережі.

У таблицю запишемо наявні у нашій схемі пристрої та віртуальні машини, елементи до яких вони приєднані, і відповідні інтерфейси з обох сторін. Також на даному кроці складемо описову таблицю мережі для того, щоб краще орієнтуватися у елементах відповідної комп'ютерної мережі. Почнемо з створення описової таблиці.

Таблиця 3. Описова таблиця

Мережа/ Пристрій	Інтерфейс/Мережний адаптер/Шлюз	Підключений до пристрою	Підключений до інтерфейсу
Маршрутизатор R-12-2-1	Fa0/1	SW-12-2-1	Fa0/1
Комутатор SW-12-22-1	Fa0/1	R-12-2-1	Fa0/1
	Fa0/2	Srv-win-22-hluh-1	Fa0
	Fa0/3	Srv-win-22-hluh-2	Fa0

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

Комутатор SW-12-22-1 (продовження)	Fa0/4	Srv-win-22-hluh-3	Fa0
	Fa0/5	WS-12-2-1	Fa0
	Fa0/6	WS-12-2-2	Fa0
Сервер Srv-win-22-hluh-1	Fa0	SW-12-2-1	Fa0/2
Сервер Srv-win-22-hluh-2	Fa0	SW-12-2-1	Fa0/3
Сервер Srv-win-22-hluh-3	Fa0	SW-12-2-1	Fa0/4
Робоча станція WS-12-2-1	Fa0	SW-12-2-1	Fa0/5
Робоча станція WS-12-2-2	Fa0	SW-12-2-1	Fa0/6

2.2.2 Розробка схеми розподілу IP-адрес для ефективного використання

Після завершення проєктування описової таблиці можна приступати до складання таблиці адресації, сюди впишемо всі адреси всіх наявних пристроїв в мережі, шлюзи за замовчуванням та інтерфейси, на які налаштовано ці адреси.

Таблиця 4. Таблиця IP-адресації

Мережа/ Пристрій	Інтерфейс/Мережний адаптер/Шлюз	IP-адреса	Маска	Префікс
Мережа А	-	192.168.2.64	255.255.255.240	/28
Маршрутизатор R-12-2-1	Fa0/1	192.168.2.65	255.255.255.240	/28
Комутатор SW-12-2-1	Vlan 1	192.168.2.66	255.255.255.240	/28
Сервер Srv-win-22-hluh-1	Fa0	192.168.2.67	255.255.255.240	/28
	Шлюз за замовчуванням	192.168.2.65	-	-
Сервер Srv-win-22-hluh-2	Fa0	192.168.2.68	255.255.255.240	/28
	Шлюз за замовчуванням	192.168.2.65	-	-

Сервер Srv-win-22- hluh-3	Fa0	192.168.2.71	255.255.255.240	/28
	Шлюз за замовчуванням	192.168.2.65	-	-
WS-12-2-1	Fa0	192.168.2.69	255.255.255.240	/28
	Шлюз за замовчуванням	192.168.2.65	-	-
WS-12-2-2	Fa0	192.168.2.70	255.255.255.240	/28
	Шлюз за замовчуванням	192.168.2.65	-	-

2.2.3 Налаштування DHCP-сервера для автоматизації розподілу IP-адрес

Отже, було розроблено таблицю адресації для майбутньої мережі, яка відображає розподіл IP-адрес серед її компонентів. Для невеликої схеми, що включає обмежену кількість віртуальних станцій, створення адресного діапазону та призначення адрес вручну не становить значних труднощів і може бути виконано без особливих зусиль. Проте в умовах невеликої кількості вузлів такий підхід залишається прийнятним лише на початковому етапі. Зі збільшенням масштабу мережі, зокрема при необхідності налаштування сотень робочих станцій, ручне призначення адрес стає економічно невиправданим, оскільки витрати часу адміністратора зростають у геометричній прогресії.

Для вирішення цієї проблеми адміністратор може вдаватися до використання технології DHCP (Dynamic Host Configuration Protocol), яка автоматизує процес розподілу IP-адрес. DHCP - це мережевий протокол, що дозволяє серверу централізовано видавати динамічні IP-адреси пристроям у мережі, забезпечуючи їхню автоматичну конфігурацію з мінімальним втручанням. У нашому випадку доцільно виділити DHCP на окремий сервер, що дозволить розвантажити головний контролер домену та спростити управління окремими функціями мережі.

		Глухенький Д.О.			KKIKB.KP.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Для забезпечення стабільності роботи ключових серверів, таких як контролер домену чи DNS, доцільно зберегти для них статичну адресацію, уникаючи змін у їхніх конфігураціях. Водночас динамічна адресація, реалізована через DHCP, буде застосована для розподілу адрес серед усіх існуючих і нових робочих станцій. Цей підхід гарантує, що кожна машина, яка приєднується до домену, автоматично отримує адресу з заздалегідь визначеного діапазону, усуваючи необхідність ручного втручання. Така стратегія не лише спрощує початкове налаштування, а й автоматизує управління адресами, що є особливо цінним як для невеликих, так і для великих мереж, сприяючи їхній масштабованості та ефективності.

Крім того, використання DHCP дозволяє централізовано оновлювати параметри мережі, такі як адреси шлюзів чи DNS-серверів, що значно полегшує адміністрування в динамічних умовах. Наприклад, при зміні мережевої топології адміністратор може просто оновити конфігурацію DHCP-сервера, і всі підключені пристрої отримають нові налаштування автоматично. Такий підхід мінімізує ймовірність помилок і забезпечує гнучкість у розширенні інфраструктури, роблячи його незамінним інструментом для сучасних корпоративних мереж.

2.3 Інструменти віртуалізації для реалізації мережі

При проєктуванні мережевої схеми виникло питання використання реальних робочих станцій та серверів. Для продовження нашого дослідження прийнято рішення використовувати образи віртуальних машин. Це було зроблено для того, щоб максимально наблизити умови роботи системи до реальних, витративши на це мінімум зусиль та власних коштів. В роботі буде використано образи віртуальних машин Windows Server 2019 та робочі станції із інстальованою ОС Windows 10.

2.3.1 Застосування VirtualBox для створення віртуальних машин

VirtualBox є безкоштовною платформою для віртуалізації, розробленою компанією Oracle, яка дозволяє створювати та запускати віртуальні машини на

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				34
Змн.	Арк.	№ докум.	Підпис	Дата		

різних операційних системах, таких як Windows, Linux, macOS, Solaris та інші. Завдяки своїй мультиплатформності, VirtualBox підтримує широкий спектр гостьових ОС, що робить його популярним інструментом для тестування, навчання та створення домашніх лабораторій. Ця гнучкість забезпечує користувачам можливість працювати з віртуальними середовищами на різних хостових платформах, адаптуючи їх до специфічних умов.

Основні функціональні можливості:

VirtualBox пропонує низку ключових можливостей, які сприяють його ефективному використанню в проектуванні віртуальних мереж. Зокрема:

- Підтримка різноманітних ОС: Платформа сумісна з більш ніж 50 гостьовими операційними системами, включаючи популярні дистрибутиви Linux, версії Windows та macOS, що дозволяє створювати гетерогенні віртуальні середовища.
- Гнучка конфігурація: Користувачі можуть налаштовувати параметри віртуальних машин, зокрема обсяг оперативної пам'яті, кількість процесорних ядер та розмір віртуальних дисків, оптимізуючи продуктивність під конкретні задачі.
- Механізм знімків (Snapshots): Функція створення знімків стану дозволяє зберігати конфігурацію віртуальної машини на певному етапі, що полегшує експерименти та повернення до попереднього стану за потреби.
- Гнучкі мережеві налаштування: VirtualBox підтримує різні режими мережевої взаємодії, такі як NAT, мостові з'єднання, внутрішні мережі та мережі хостів, забезпечуючи можливість моделювання складних мережевих топологій.

Переваги та обмеження:

VirtualBox має як сильні сторони, так і певні недоліки, які впливають на його вибір для різних сценаріїв. До переваг належать:

- Мультиплатформність: Програма працює на хостових системах Windows, Linux і macOS, що розширює її доступність.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

- **Безкоштовність:** Як рішення з відкритим кодом, VirtualBox доступний для використання без додаткових витрат, що є перевагою для індивідуальних і невеликих проєктів.
- **Простота інтерфейсу:** Інтуїтивно зрозумілий графічний інтерфейс полегшує роботу як для початківців, так і для досвідчених користувачів.

Разом із тим, платформа має певні обмеження:

- **Нижча продуктивність:** У порівнянні з комерційними аналогами, такими як VMware чи Hyper-V, VirtualBox може демонструвати меншу ефективність у сценаріях із високим навантаженням.
- **Обмежена корпоративна підтримка:** VirtualBox не призначений для масштабних корпоративних інфраструктур, де потрібні розширені функції, як-от у VMware vSphere.
- **Обмеження з macOS:** Через ліцензійні обмеження Apple підтримка macOS як гостьової системи може бути неповною для деяких версій.

Сфери застосування:

VirtualBox є ідеальним рішенням для створення віртуальних машин у навчальних цілях, тестуванні програмного забезпечення та моделюванні невеликих мереж, зокрема в рамках проєктування корпоративної IPv4-мережі. Його універсальність і відсутність вартості роблять його привабливим для індивідуальних користувачів і невеликих організацій. Проте для складних корпоративних середовищ, де важливі висока продуктивність і масштабування, варто розглянути альтернативи, такі як Hyper-V або VMware, які пропонують розширені можливості для великих інфраструктур.

2.3.1.1 Порівняльний аналіз VirtualBox та Hyper-V: переваги та обмеження

VirtualBox і Hyper-V є двома популярними платформами для віртуалізації, але вони мають різні підходи роботи та особливості їх використання. Розберемо детальніше кілька ключових відмінностей між ними. Складемо порівняльну

		Глухенький Д.О.			KKIKB.KP.B-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

таблицю куди коротко запишемо спільні та схожі сторони. Далі детальніше оглянемо кожен із пунктів порівняння.

Таблиця 5. Порівняння гіпервізорів

Критерій	VirtualBox	Hyper-V
Тип гіпервізора	Тип 2 (hosted)	Тип 1 (bare-metal)
Вартість	Безкоштовно (open-source)	Безкоштовно (вбудовано у Windows)
Сумісність із ОС	Windows, Linux, macOS, Solaris	Тільки Windows
Продуктивність	Середня	Висока
Кількість віртуальних процесорів	До 32 на VM	До 240 на VM
Управління ресурсами	Гнучка, але менш ефективне	Динамічне, високоефективне
Мережеві налаштування	Гнучкі (NAT, міст, внутрішні)	Розширені (включно SDN)
Знімки/Чекпоінти	Знімки (Snapshots)	Чекпоінти (Checkpoints)
Підтримка 3-Д графіки	Обмежена, потребує Guest Additions	Обмежена, через RemoteFX
Корпоративні функції	Обмежені	Розширені (міграція, кластери)
Легкість використання	Проста, інтуїтивна	Складніша, потребує досвіду

1. Тип гіпервізора:

VirtualBox: Як тип 2 гіпервізора (hosted), VirtualBox працює поверх існуючої операційної системи хоста (наприклад, Windows чи Linux), що означає, що він залежить від хостової ОС для доступу до апаратних ресурсів. Це робить його менш ефективним у порівнянні з типом 1, але простішим у встановленні та використанні на звичайних комп'ютерах.

Hyper-V: Як тип 1 гіпервізора (bare-metal), Hyper-V працює безпосередньо на апаратному забезпеченні, займаючи місце хостової ОС. Це забезпечує вищу продуктивність і стабільність, але вимагає активації через Windows Features і сумісності з певними редакціями Windows (Pro, Enterprise).

2. Вартість

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

VirtualBox: Є безкоштовним open-source рішенням, що дозволяє використовувати його без обмежень як для особистих, так і для комерційних цілей. Додаткові розширення доступні за ліцензією, але базова версія повністю безкоштовна.

Hyper-V: Вбудований у Windows Pro, Enterprise і Server. Можна використовувати без додаткових витрат, якщо у вас уже є відповідна ліцензія на ОС. Це робить його економічно вигідним для користувачів Microsoft, але потребує відповідної інфраструктури. Проте можуть виникнути певні проблеми, якщо у користувача немає ліцензії на Windows.

3. Сумісність із ОС

VirtualBox: Підтримує широкий спектр хостових ОС (Windows, Linux, macOS, Solaris), що робить його універсальним інструментом для кросплатформного тестування. Як гостьові ОС також підтримує понад 50 систем, включаючи macOS (але з обмеженнями).

Hyper-V: Обмежений виключно Windows як хостовою ОС. Проте підтримує гостьові ОС Windows, Linux і FreeBSD. Також його інтеграція з екосистемою Microsoft є перевагою для Windows-оточення.

4. Продуктивність

VirtualBox: Має середню продуктивність, що впливає на швидкість у ресурсномістких задачах. Підходить для невеликих проєктів чи тестування, але може бути менш ефективним при великій кількості VM.

Hyper-V: Забезпечує високу продуктивність завдяки прямому доступу до апаратних ресурсів, що ідеально для великих навантажень і корпоративних середовищ.

5. Кількість віртуальних процесорів

VirtualBox: Підтримує до 32 віртуальних процесорів на одну VM, що достатньо для невеликих або середніх задач, але обмежує масштабування.

Hyper-V: Дозволяє використовувати до 240 віртуальних процесорів на VM, що робить його придатним для масштабних корпоративних застосувань із високими обчислювальними вимогами.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

6. Управління ресурсами

VirtualBox: Пропонує гнучке налаштування ресурсів (CPU, RAM, диск), але воно менш ефективне через відсутність динамічної оптимізації. Користувач вручну визначає бажані параметри, що може призвести до перевантаження чи недовикористання.

Hyper-V: Має функцію динамічного розподілу пам'яті та ресурсів, що автоматично адаптується до навантаження, оптимізуючи використання апаратних можливостей для кількох VM.

7. Мережеві налаштування

VirtualBox: Надає гнучкі опції налаштування, що дозволяє моделювати складні мережеві топології та є ідеальним варіантом для навчальних цілей. (NAT, мостові з'єднання, внутрішні мережі, мережі хостів)

Hyper-V: Підтримує розширені мережеві функції, включаючи програмно-визначені мережі (SDN) і ізоляцію, що робить його потужним інструментом для корпоративних мереж із високими вимогами до безпеки.

8. Знімки/чекпоінти

VirtualBox: Використовує знімки (Snapshots), які дозволяють зберігати стан VM і повертатися до нього в будь який зручний час. Це зручно для тестування, але може впливати на продуктивність при великій кількості знімків.

Hyper-V: Застосовує чекпоінти (Checkpoints – « з англ. контрольні точки»), які працюють схожим чином, але інтегровані з Windows-екосистемою, забезпечуючи швидке відновлення та сумісність із корпоративними інструментами.

9. Підтримка 3D-графіки

VirtualBox: Підтримка 3D-графіки доступна після встановлення Guest Additions, але обмежена і може бути нестабільною, особливо для сучасних додатків.

Hyper-V: Пропонує підтримку через RemoteFX, але також із обмеженнями, особливо для не-Windows-гостьових ОС. Ефективність залежить від апаратної підтримки.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		39

10. Корпоративні функції

VirtualBox: Має обмежені корпоративні можливості, такі як відсутність живого мігрування чи кластеризації, що робить його менш придатним для великих інфраструктур.

Hyper-V: Пропонує розширені функції, включаючи живу міграцію, кластеризацію та інтеграцію з System Center, що ідеально для великих корпоративних середовищ.

11. Легкість використання

VirtualBox: Має інтуїтивно зрозумілий графічний інтерфейс, що полегшує роботу для початківців і не потребує глибоких знань про адміністрування.

Hyper-V: Вимагає більше технічних навичок через складніший процес налаштування та інтеграцію з PowerShell, що може бути складним для новачків.

2.3.2 Використання Windows Server 2019 як основи для серверних компонентів

Windows Server є потужною серверною операційною системою від Microsoft, розробленою для забезпечення надійної інфраструктури для корпоративних мереж, хмарних рішень та віртуалізованих середовищ. Ця платформа призначена для підтримки складних робочих навантажень, від управління користувачами до розгортання сучасних додатків.

Основним нововведенням є підтримка гібридних сценаріїв роботи через інтеграцію з Azure, зокрема через інструменти, такі як Azure Arc, які дозволяють централізовано керувати як локальними, так і хмарними ресурсами. Ця інтеграція полегшує розширення інфраструктури організації в хмарне середовище, забезпечуючи гнучкість і сумісність із хмарними сервісами, такими як віртуальні машини та сховища даних.

Система також пропонує покращення в області безпеки, включаючи розширені механізми захисту від загроз, такі як шифрування даних і контроль доступу. Ці заходи спрямовані на підвищення стійкості інфраструктури до кібератак і забезпечення безпечного обміну інформацією в гібридних

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

середовищах. Варто відзначити інтеграцію з VirtualBox, що дозволяє використовувати цю платформу віртуалізації для розгортання та тестування серверних компонентів у віртуальному середовищі. Така сумісність розширює можливості адміністраторів, надаючи інструменти для моделювання та експериментів без впливу на основну інфраструктуру.

2.3.3 Розгортання Windows 10 на клієнтських пристроях

Віртуальна машина з Windows 10 - це середовище, створене за допомогою платформи віртуалізації, як-от Hyper-V, VMware чи VirtualBox, яке дозволяє використовувати всі можливості операційної системи Windows 10 без потреби встановлення її на фізичний комп'ютер. Вона ідеально підходить для тестування, навчання, розробки програмного забезпечення чи інтеграції з серверними ОС.

Процес розгортання починається з підготовки віртуальної машини в VirtualBox, де визначаються ключові параметри, такі як обсяг оперативної пам'яті, кількість процесорних ядер і розмір віртуального диска. Для Windows 10 важливо враховувати мінімальні системні вимоги, щоб забезпечити стабільну роботу, включаючи достатній обсяг дискового простору для встановлення системи та оновлень. Після створення віртуальної машини використовується образ установки Windows 10 (наприклад, у форматі ISO, OVA), який імпортується в VirtualBox для запуску процесу інсталяції.

Крім того, розгортання Windows 10 у VirtualBox підтримує використання знімків (Snapshots), що дає змогу зберігати ключові етапи конфігурації та повертатися до них у разі виникнення помилок. Така функція є особливо корисною для тестування різних сценаріїв роботи клієнтських пристроїв, наприклад, приєднання до домену чи оновлення системи. Завдяки інтуїтивному інтерфейсу VirtualBox процес налаштування може бути адаптований до потреб проекту, забезпечуючи реалістичне моделювання робочих станцій.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		41

2.4. Розробка служб та сервісів прикладного рівня

2.4.1 Налаштування Active Directory для управління користувачами та ресурсами

Active Directory (AD) - це каталогова служба, розроблена компанією Microsoft для операційних систем Windows, яка забезпечує централізоване управління мережевими ресурсами, зокрема користувачами, комп'ютерами, пристроями та іншими об'єктами в корпоративних інфраструктурах. AD є ключовим інструментом для адміністрування доступу, безпеки та політик у межах усієї мережі, що дозволяє організаціям ефективно координувати складні ІТ-системи.

Функціональні можливості Active Directory

Active Directory надає широкий спектр функцій для управління користувачами та ресурсами. Однією з основних можливостей є створення та адміністрування облікових записів користувачів і груп, що дозволяє централізовано контролювати доступ до мережесих ресурсів, таких як файли, принтери чи додатки. AD забезпечує автентифікацію користувачів (перевірку їхньої особи) та авторизацію (визначення рівня доступу) через використання протоколів, зокрема Kerberos і NTLM, що гарантує безпечну ідентифікацію та захист даних.

Крім того, AD підтримує налаштування групових політик (Group Policy), які дозволяють централізовано визначати параметри безпеки, конфігурації робочих станцій і користувацькі налаштування для всіх об'єктів у домені. Наприклад, адміністратор може встановити обмеження на використання паролів або автоматично розгорнути оновлення на всіх комп'ютерах. AD також забезпечує управління пристроями в мережі, надаючи можливості для моніторингу їхнього стану, централізованого налаштування та швидкого усунення неполадок.

Переваги використання Active Directory

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		42

- Застосування Active Directory у корпоративному середовищі має низку переваг:
- Централізоване управління: Усі налаштування, політики доступу та ресурси контролюються з єдиної точки, що значно спрощує адміністрування великих мереж.
- Масштабованість: AD підтримує розширення інфраструктури шляхом додавання нових доменів, контролерів домену чи організаційних одиниць, адаптуючись до зростання організації.
- Гнучкість: Система дозволяє створювати складні структури доменів і груп, що забезпечує детальне налаштування прав доступу відповідно до потреб компанії.
- Підвищена безпека: Централізоване управління політиками безпеки та інтеграція з іншими захисними механізмами підвищують рівень захисту мережі.
- Інтеграція з іншими системами: AD легко інтегрується з такими рішеннями Microsoft, як Exchange, SharePoint і Microsoft 365, створюючи єдину точку управління для всіх корпоративних сервісів.

Обмеження Active Directory

Незважаючи на численні переваги, Active Directory має певні недоліки, які варто враховувати:

- Складність налаштування: Початкове розгортання AD може бути складним, особливо в організаціях із великою кількістю користувачів або складними інтеграційними вимогами.
- Залежність від Windows: AD оптимізовано для роботи в середовищі Windows, що може ускладнювати інтеграцію з іншими операційними системами, такими як Linux чи macOS.
- Високі вимоги до ресурсів: Для ефективної роботи в масштабних мережах AD потребує значних апаратних ресурсів, зокрема потужних серверів і стабільної інфраструктури.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

Active Directory є фундаментальним компонентом управління IT-інфраструктурою в середовищі Windows, забезпечуючи централізоване адміністрування користувачів, комп'ютерів, доступу до ресурсів і політик безпеки. Її застосування в рамках проєктування корпоративної мережі дозволяє створити надійну основу для ефективного управління, що робить AD незамінним інструментом для організацій, які прагнуть оптимізувати свої мережеві процеси.

2.4.2 Аналіз ризиків і стратегії їхньої мінімізації

Автоматизація адміністрування IT-інфраструктури за допомогою WAC і PowerShell DSC сприяє підвищенню ефективності управління, однак супроводжується потенційними ризиками, які потребують ретельного аналізу та контролю. Далі розглянемо ключові категорії ризиків, пов'язаних із використанням цих технологій, а також спробуємо знайти оптимальні рішення для їх нейтралізації.

Ризики безпеки

Застосування автоматизованих інструментів, таких як WAC і PowerShell DSC, надає адміністраторам широкі повноваження для взаємодії з численними сервісами та системами. Недостатній контроль доступу до цих інструментів може спричинити несанкціоноване втручання або компрометацію конфіденційних даних. Зокрема, слабка автентифікація може стати вразливістю, що відкриває можливості для зловмисного доступу.

Щоб мінімізувати ці загрози, необхідно буде впровадити багатофакторну автентифікацію для доступу до WAC і DSC. Додатковим заходом є обмеження доступу через віртуальні приватні мережі (VPN) або конфігурацію брандмауерів для фільтрації мережевого трафіку. Регулярний аудит і оновлення стандартних параметрів безпеки під час інтеграції нових інструментів суттєво знижують ймовірність інцидентів. Окремо варто зазначити, що PowerShell DSC може бути вразливим до впровадження шкідливого коду або витоку даних. Для захисту від таких ризиків рекомендується застосовувати цифрові підписи для верифікації

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		44

автентичності скриптів і впроваджувати моніторинг PowerShell-сесій для виявлення підозрілих команд.

Ризики, пов'язані з доступом та управлінням правами

Неправильна конфігурація прав доступу до серверів через WAC або PowerShell DSC може призвести до надмірного або недостатнього рівня повноважень користувачів, що становить загрозу для цілісності даних. Наприклад, надмірні права можуть дозволити несанкціоновані зміни, тоді як недостатні - перешкоджати виконанню необхідних завдань.

Для подолання цієї проблеми необхідно суворо дотримуватися принципу найменших привілеїв (Principle of Least Privilege) під час визначення ролей і прав доступу. Крім того, регулярний аудит повноважень користувачів та їхнє періодичне переглядання допоможуть своєчасно виявляти та усувати небажані зміни, забезпечуючи належний контроль.

Ризики, пов'язані з недостатньою кваліфікацією персоналу

Впровадження автоматизованих рішень, таких як WAC і PowerShell DSC, може бути ускладненим через брак досвіду або знань серед адміністраторів. Неправильне налаштування може призвести до нестабільності системи або неефективного використання інструментів.

При налаштуванні реальних фізичних систем варто організувати регулярні тренінги та навчальні семінари для персоналу, відповідального за адміністрування. Залучення зовнішніх консультантів для підтримки на етапах налаштування та оптимізації також сприятиме підвищенню компетентності команди та вдосконаленню процесів автоматизації.

Використання Windows Admin Center і PowerShell DSC забезпечує значні переваги, зокрема зменшення людського фактору, підвищення операційної ефективності та спрощення управління ресурсами. Однак ці технології супроводжуються потенційними ризиками, які вимагають комплексного підходу до їхнього управління. Впровадження надійних заходів безпеки, регулярного тестування, безперервного моніторингу та професійного навчання персоналу

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

дозволяє мінімізувати загрози, гарантуючи стабільну та безпечну роботу автоматизованих систем у корпоративному середовищі.

2.5. Проєктування підсистеми мережного захисту

2.5.1 Користувачі та базовий захист пристроїв

Забезпечення базового захисту мережевих пристроїв є критично важливим як для емульованих систем, так і для реальних мереж. Цей процес включає фізичні, організаційно-технічні та програмні заходи, спрямовані на захист інфраструктури. Нижче розглянемо принципи управління користувачами та алгоритми захисту для реальних та емульованих типів мереж.

Управління користувачами

Управління користувачами є ключовим елементом забезпечення безпеки мережевої інфраструктури, оскільки дозволяє контролювати доступ до ресурсів і мінімізувати ризики несанкціонованих дій. Цей процес охоплює створення облікових записів, призначення ролей і прав доступу, а також їхню періодичну модифікацію чи видалення залежно від потреб організації. Наприклад, у корпоративному середовищі користувачі можуть бути поділені на групи з різними рівнями доступу: від базового для рядових співробітників до розширеного для адміністраторів, що забезпечує гнучкість і безпеку.

Для ефективного управління доцільно використовувати централізовані системи, такі як Active Directory, які дозволяють створювати організаційні одиниці (OU) для групування користувачів за підрозділами та застосовувати до них єдині політики. Важливо також впроваджувати вимоги до складності паролів, регулярного їхнього оновлення (наприклад, кожні 90 днів) і моніторингу активності користувачів. Такий підхід допомагає виявляти аномалії, як-от численні невдалі спроби входу, і оперативно реагувати на потенційні загрози, забезпечуючи стабільність системи.

Алгоритми базового захисту

Базові заходи захисту мережевих пристроїв спрямовані на створення першого рівня безпеки, що запобігає несанкціонованому доступу та вразливостям. Для

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		46

емульованих систем, таких як GNS3, захист включає встановлення складних паролів на віртуальні пристрої, використання списків доступу (ACL) для фільтрації трафіку та впровадження VPN із протоколами IPsec для безпечного зовнішнього доступу. Ці заходи дозволяють моделювати захищене середовище, яке відповідає реальним умовам корпоративних мереж.

У фізичних мережах акцент робиться на фізичну безпеку, таку як розміщення обладнання в захищених приміщеннях із контрольованим доступом, а також на захист мережевого трафіку через брандмауери та шифрування. Регулярне оновлення прошивок і програмного забезпечення пристроїв є критично важливим для усунення відомих вразливостей, що забезпечує стійкість до зовнішніх загроз. Такі базові алгоритми створюють основу для подальшого впровадження більш складних захисних механізмів.

Порівняння методів

Методи захисту емульованих і фізичних мереж мають значні відмінності, зумовлені їхньою природою та специфікою роботи. У фізичних мережах ключову роль відіграє фізична безпека, зокрема захист обладнання від несанкціонованого доступу чи пошкодження, а також управління кабельною інфраструктурою для уникнення збоїв. Наприклад, правильне маркування та ізоляція кабелів допомагають уникнути випадкових відключень, що не є актуальним для віртуальних систем.

В емульованих мережах, таких як GNS3, акцент зміщується на віртуальні заходи, такі як налаштування VPN для захисту зовнішнього доступу та використання ACL для контролю трафіку, оскільки фізичні ризики відсутні. Однак ці методи можуть бути менш ефективними в реальних мережах через складність апаратного налаштування. Розуміння цих відмінностей дозволяє адаптувати стратегії захисту до типу інфраструктури, забезпечуючи оптимальний рівень безпеки.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		47

2.5.2 Віддалений доступ

Graphical Network Simulator v3 (GNS3) підтримує кілька методів забезпечення віддаленого доступу до мережевих пристроїв, серед яких одним із ключових є використання протоколу SSH (Secure Shell). Цей протокол забезпечує безпечне шифрування даних і надійну автентифікацію, що дозволяє адміністраторам встановлювати захищені з'єднання з віртуальними пристроями в емульованому середовищі. У рамках проєкту передбачається інтеграція з Windows Admin Center, який пропонує інструменти, такі як віддалений робочий стіл і стандартні механізми віддаленого підключення до віртуальних машин, для реалізації цих функцій.

WAC забезпечує централізоване управління кількома машинами через єдиний веб-інтерфейс, що підвищує ефективність адміністрування. Процес включає додавання віддалених машин до платформи, після чого доступні різні інструменти управління, зокрема віддалений робочий стіл, який дозволяє виконувати конфігурацію систем без необхідності фізичного доступу до робочих станцій чи серверів. Крім того, WAC підтримує розширене адміністрування через вбудовані утиліти, що розширюють можливості контролю над інфраструктурою.

Інтеграція з PowerShell у WAC дозволяє виконувати віддалені командлети на серверах без необхідності підключення до робочого столу, забезпечуючи гнучкість у виконанні адміністративних задач. Функція доступу до диспетчера задач дає змогу моніторити процеси на віддалених машинах і керувати ними, тоді як перегляд журналів подій сприяє діагностиці потенційних проблем. Крім того, дана платформа надає можливість виконувати роботу з файлами на віддалених серверах і робочих станціях без використання мережевих спільних ресурсів. Це сильно підвищує рівень зручності адміністрування. Ці можливості роблять WAC ефективним інструментом для автоматизації та оптимізації управління мережею.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

Детальні аспекти налаштування мережі та інтеграції робочих станцій із Windows Admin Center будуть розглянуті в [Розділі 3](#), де буде представлено практичне впровадження цих рішень.

2.5.3 Застосування брандмауера для захисту від зовнішніх загроз

Брандмауер є одним із ключових елементів захисту мережевої інфраструктури, забезпечуючи контроль і фільтрацію трафіку для запобігання зовнішнім загрозам. Його використання дозволяє створювати бар'єр між внутрішньою мережею організації та зовнішнім середовищем, що є критично важливим для захисту від несанкціонованого доступу, атак типу "відмова в обслуговуванні" (DDoS) та інших кіберзагроз. У контексті проєктування корпоративної IPv4-мережі брандмауер відіграє важливу роль у забезпеченні безпеки як для фізичних, так і для емульованих систем.

Функціонал брандмауера базується на встановленні правил фільтрації, які визначають, які типи трафіку дозволені, а які блокуються, залежно від джерела, призначення, протоколів і портів. Наприклад, брандмауер може дозволяти вхідний трафік лише для певних служб, таких як HTTPS, блокуючи всі інші запити, що зменшує поверхню атаки. Додатково сучасні брандмауери підтримують функції інспекції пакетів (Deep Packet Inspection), що дозволяє аналізувати вміст даних для виявлення потенційно шкідливого коду. У поєднанні з можливостями логування та моніторингу брандмауер допомагає адміністраторам своєчасно виявляти підозрілу активність і реагувати на інциденти.

У фізичних мережах брандмауер часто розгортається як окремий апаратний пристрій або програмне забезпечення на межі мережі (наприклад, між внутрішньою мережею та Інтернетом). Для емульованих мереж, таких як GNS3, брандмауер може бути налаштований на віртуальних маршрутизаторах або серверах, що імітують захист реального середовища. У обох випадках важливо враховувати принципи зонування мережі, створюючи демілітаризовані зони (DMZ) для ізоляції публічних сервісів, таких як веб-сервери, від внутрішніх

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

ресурсів. Це дає змогу мінімізувати ризик проникнення в критичні системи в разі компрометації зовнішнього сервісу.

Застосування брандмауера також вимагає регулярного оновлення правил фільтрації та аналізу логів для адаптації до нових загроз. Наприклад, у разі виявлення нової вразливості в певному протоколі брандмауер може бути налаштований для блокування відповідного трафіку до встановлення патчів. Такий підхід, разом із інтеграцією з іншими захисними механізмами, такими як системи виявлення вторгнень (IDS), забезпечує комплексний захист від зовнішніх загроз, що є необхідною умовою для безпечної роботи корпоративної мережі.

Висновок до другого розділу

Другий розділ кваліфікаційної роботи присвячений всебічному проєктуванню корпоративної IPv4-мережі організації, яке охопило розробку фізичної та логічної топологій, інтеграцію віртуалізаційних технологій, а також створення системи IP-адресації та служб прикладного рівня. Було проаналізовано використання комутаторів, маршрутизаторів і каналів зв'язку для забезпечення стабільної інфраструктури мережі. Також було досліджено впровадження інструментів, зокрема VirtualBox і Windows Server 2019, для моделювання робочих станцій і серверів у віртуальному середовищі. Розробка логічної топології, включно з DHCP для автоматизації адрес, і налаштування Active Directory забезпечили основу для централізованого управління ресурсами.

Окрему увагу було приділено проєктуванню системи мережевої безпеки, зокрема управлінню користувачами, базовому захисту пристроїв і застосуванню брандмауера для захисту від зовнішніх загроз. Порівняння методів захисту для емульованих і фізичних мереж підкреслило необхідність адаптації стратегій до специфіки інфраструктури. Проведений аналіз налаштування віддаленого доступу через Windows Admin Center і PowerShell DSC розширило можливості адміністрування. Отримані результати закладають велику теоретичну основу для практичного впровадження, деталі якого будуть розглянуті в наступному розділі.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ОЦІНКА СПРОЄКТОВАНО МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ

Опрацювавши всі потрібні теоретичні відомості можна приступити до практичного виконання проєкту.

Необхідно виконати наступні завдання:

- створити віртуальну мережу на базі GNSv3,
- додати до неї віртуальні машини на базі Windows Server 2019,
- додати робочі станції які будуть працювати на базі ОС Windows 10,
- встановити на одну робочу станцію Windows Admin Center,
- провести повний аналіз його функціоналу,
- за допомогою PowerShell DSC створити групу скриптів які будуть задавати бажаний стан системи,
- провести аналіз колаборації WAC та DSC,
- проаналізувати отримані результати.

3.1. Вибір обладнання та операційних систем для реалізації мережі

Як зазначалося у Розділ 2 (п. [2.1.1](#) та [2.1.2](#)), для реалізації проєкту одну із ключових ролей грає вибір відповідних моделей комутаторів і маршрутизаторів, які відповідають вимогам до продуктивності, масштабованості та безпеки. З урахуванням потреб інфраструктури було обрано моделі від компанії Cisco Systems, а саме комутатор Cisco Catalyst 2960 та маршрутизатор Cisco 7200, які забезпечують необхідний функціонал для стабільної роботи мережі.

Комутатор Cisco Catalyst 2960 належить до серії керованих комутаторів другого рівня і оснащений 24 портами Ethernet/Fast Ethernet із підтримкою швидкості передачі даних до 100 Мбіт/с. Він підтримує стандартні мережеві протоколи для створення віртуальних локальних мереж, а також забезпечує функції управління трафіком, такі як якість обслуговування і обмеження широкомовних штормів. Додатково модель пропонує базові механізми безпеки, зокрема списки доступу (ACL) для фільтрації трафіку.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				51
Змн.	Арк.	№ докум.	Підпис	Дата		

Маршрутизатор Cisco 7200 відноситься до серії високопродуктивних маршрутизаторів, призначених для середніх і великих мереж, де потрібна ефективна обробка трафіку та підтримка складних протоколів. Ця модель забезпечує високу надійність завдяки модульній архітектурі, яка дозволяє адаптувати пристрій до потреб інфраструктури, і підтримує широкий спектр протоколів маршрутизації, зокрема IP, IPv6, EIGRP, MPLS і GET-VPN. Завдяки інтеграції з технологіями віртуалізації та шифрування, Cisco 7200 забезпечує безпечну передачу даних і гнучке управління міжмережевими з'єднаннями.



Рис. 3 Відповідні елементи у емуляторі GNSv3

3.2 Налаштування DHCP-сервера для автоматизації розподілу IP-адрес

Контролер домену відіграє центральну роль у мережевій інфраструктурі, забезпечуючи управління політиками та бажаним станом пристроїв. Однак його використання пов'язане зі значним навантаженням, оскільки він є єдиною точкою управління ключовими аспектами мережі, що створює ризик для безпеки та стабільності в разі відмови. Для підвищення надійності системи було вирішено розгорнути додатковий Сервер №2, основною функцією якого є реплікація даних контролера домену та забезпечення безперервного управління мережею у випадку збою основного контролера.

Розподіл IP-адрес у мережі є складним і трудомістким процесом, особливо в умовах масштабування. Для автоматизації цього процесу доцільно використовувати протокол DHCP, однак його розгортання на контролері домену є недоцільним через високий рівень концентрації критичних служб в одному місці, що підвищує вразливість системи. З метою зниження навантаження та підвищення безпеки було створено окремий Сервер №3, на якому буде встановлено службу DHCP. Цей сервер забезпечить автоматичний розподіл IP-адрес для всіх існуючих і нових робочих станцій, оптимізуючи управління

		Глухенький Д.О.			KKIKB.KP.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

мережею та зменшуючи ризик перевантаження основних компонентів інфраструктури. Після інсталяції образу Сервера №3 встановимо на нього роль DHCP сервера.

Одразу перевіримо список всіх інстальованих ролей та служб на поточному сервері, для цього використаємо команду **Get-WindowsFeature | Where-Object {\$_.Installed -eq \$true}**.

```

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

PS C:\Users\Administrator> Get-WindowsFeature | Where-Object {$_.Installed -eq $true}

Display Name                                     Name                                Install State
-----
[X] DHCP Server                                DHCP                                Installed
[X] File and Storage Services                  FileAndStorage-Services             Installed
[X] Storage Services                          Storage-Services                     Installed
[X] .NET Framework 4.7 Features                NET-Framework-45-Fea...             Installed
[X] .NET Framework 4.7                        NET-Framework-45-Core               Installed
[X] WCF Services                              NET-WCF-Services45                  Installed
[X] TCP Port Sharing                          NET-WCF-TCP-PortShar...              Installed
[X] Remote Server Administration Tools         RSAT                                Installed
[X] Role Administration Tools                  RSAT-Role-Tools                     Installed
[X] DHCP Server Tools                         RSAT-DHCP                           Installed
[X] System Data Archiver                      System-DataArchiver                  Installed
[X] Windows Defender Antivirus                 Windows-Defender                     Installed
[X] Windows PowerShell                        PowerShellRoot                       Installed
[X] Windows PowerShell 5.1                     PowerShell                           Installed
[X] Windows PowerShell ISE                     PowerShell-ISE                       Installed
[X] WoW64 Support                             WoW64-Support                       Installed
[X] XPS Viewer                                XPS-Viewer                          Installed
  
```

Рис. 4 Перевірка встановлених служб

Також перевіримо чи отримує робоча станція адресу від Сервера №3.

```

DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::3227:6f84:4853:5dc6%9(Preferred)
IPv4 Address. . . . . : 192.168.2.69(Preferred)
Subnet Mask . . . . . : 255.255.255.240
  
```

Рис. 5 Перевірка роботи DHCP

Станція успішно отримує адресу, це означає, що можна приступати до наступного кроку проектування роботи. Під час виконання кроку 3.5 буде розглянута окрема роль для керування DHCP налаштуваннями на даному сервері.

3.3 Конфігурація Windows Admin Center для управління інфраструктурою

У даному розділі буде проведено налаштування та налагодження веб-інтерфейсного інструменту Windows Admin Center для централізованого управління інфраструктурою. Будуть використані всі дані які були отримані раніше, та будуть підняті усі проблеми та питання на які не були дані відповіді у розділах вище.

Доменний контролер є критично важливим елементом інфраструктури, відповідаючи за автентифікацію користувачів і управління доменом. Встановлення WAC, який функціонує як веб-служба, на DC може створити вразливості, зокрема через відкриття мережових портів, що збільшує поверхню для атак, наприклад, через експлуатацію веб-додатків. Крім того, WAC споживає значні ресурси, такі як процесор і оперативна пам'ять, що може знизити продуктивність DC, особливо в умовах інтенсивного навантаження, коли численні запити до служби автентифікації можуть призвести до затримок або збоїв.

З метою забезпечення безпеки та стабільності WAC буде встановлено на окремій віртуальній робочій станції з операційною системою Windows 10, якій присвоєно назву Windows-WAC. Для цього завантажуюмо офіційний образ WAC із офіційного вебсайту Microsoft і виконуємо інсталяцію на станцію Windows-WAC. Після завершення встановлення перевіряємо коректність інсталяції, переконуючись, що веб-інтерфейс WAC доступний через браузер за відповідною адресою, що забезпечить подальше централізоване управління інфраструктурою.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		54

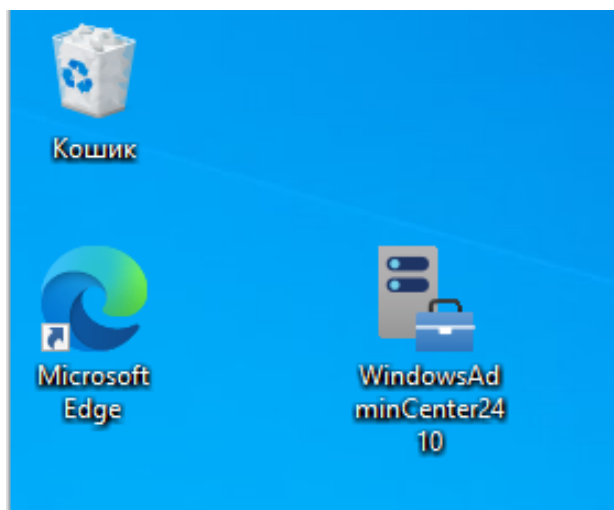


Рис. 6 Логотип WAC на робочому столі

Як уже зазначалося, Windows Admin Center є веб-інтерфейсом, що функціонує через вбудований або зовнішній браузер операційної системи, тому для його ефективного використання на робочій станції Windows-WAC рекомендується встановити актуальну та оновлену версію браузера, наприклад, Microsoft Edge. Це забезпечить стабільну роботу інтерфейсу та сумісність із сучасними веб-технологіями. Після завершення інсталяції та запуску WAC система вимагає авторизацію з використанням облікових записів, пов'язаних із доменом, що гарантує безпечний доступ до ресурсів.

Для зручності проведення майбутніх тестів передбачається використання облікового запису головного адміністратора домену для авторизації в WAC. Цей вибір забезпечить повний доступ до всіх необхідних програм і утиліт, дозволить здійснювати моніторинг ресурсів робочої станції та вносити віддалені зміни в конфігурацію системи. Такий підхід оптимізує адміністративні процеси та сприяє ефективному управлінню інфраструктурою через централізований інтерфейс.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				55
Змн.	Арк.	№ докум.	Підпис	Дата		



Sign in to Windows Admin Center

Username

hluhenkiy\ADMINISTRATOR

Password

Sign In

Рис. 7 Авторизація в програму

Після запуску Windows Admin Center користувач стикається з порожнім полем приєднаних пристроїв, що вимагає виконання початкових дій для інтеграції інфраструктури. Необхідно обрати опцію "Додати" в інтерфейсі WAC та підключити всі віртуальні машини в домені. Розпочнемо з контролера домену, позначеного як Srv-Win-22-Hluh. Цей етап забезпечує доступ до ключових серверів і робочих станцій, створюючи основу для подальшого адміністрування.

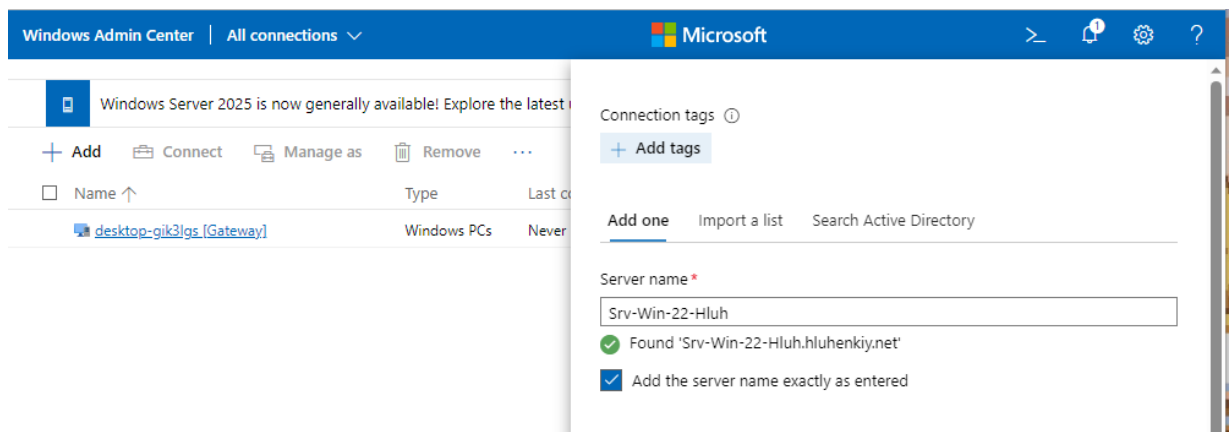


Рис. 8 Приєднання до контролеру домена

Для перевірки коректності роботи Windows Admin Center після підключення до сервера Srv-Win-22-Hluh виконуємо тестування функціоналу. На

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				56
Змн.	Арк.	№ докум.	Підпис	Дата		

сервері створюємо текстовий документ і відкриваємо в WAC інструмент керування теками для підтвердження його наявності.

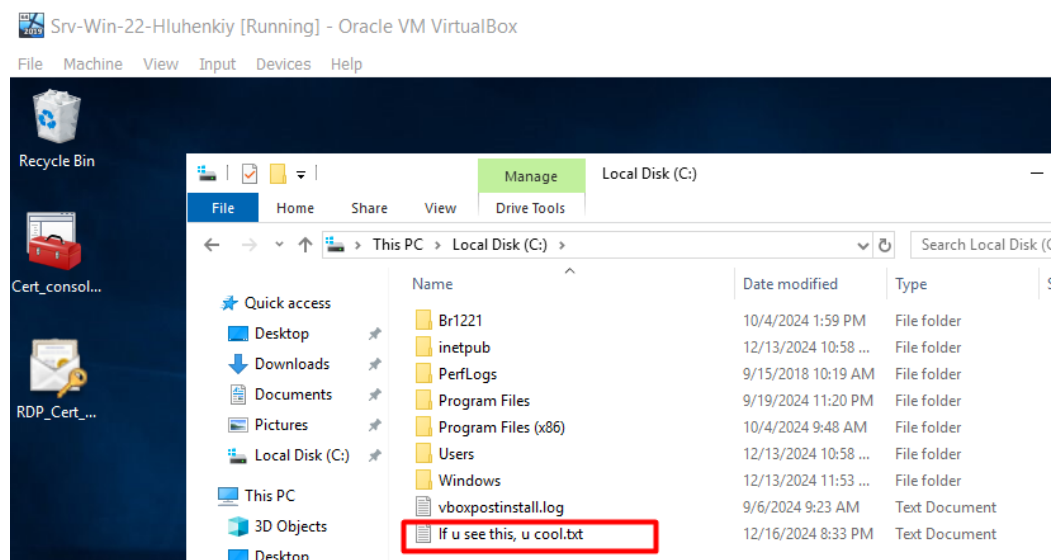


Рис. 9 Створення файлу на сервері

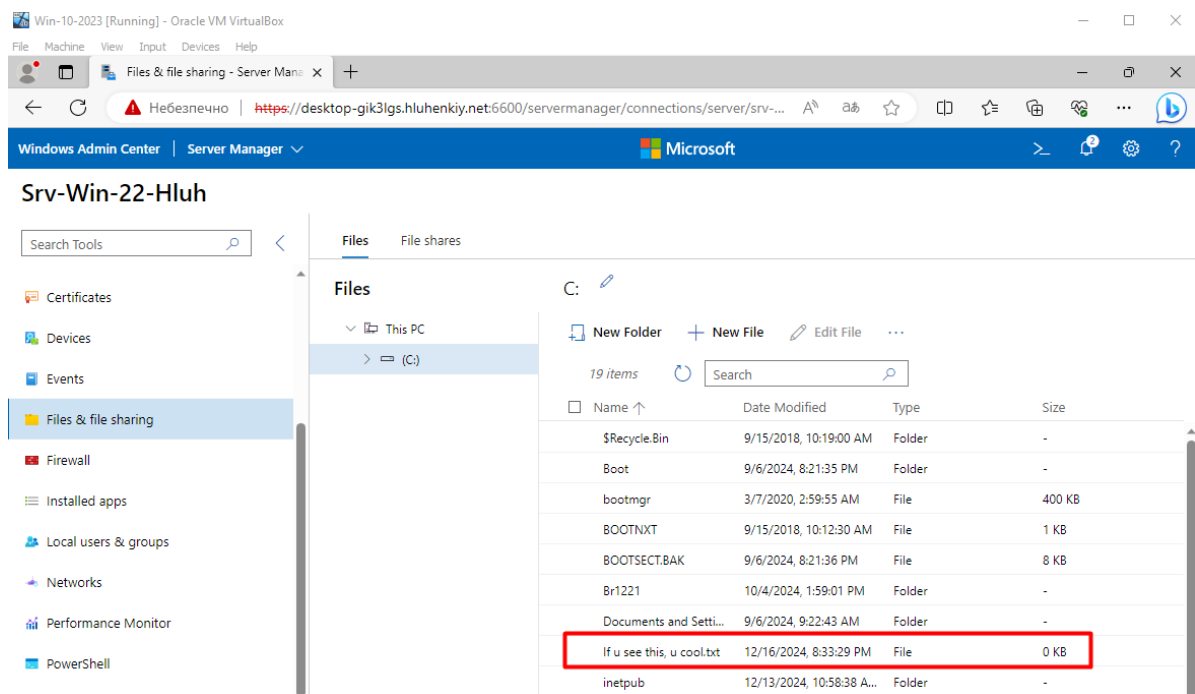


Рис. 10 Перевірка створення файлу на сервер

Успішне відображення файлу в інтерфейсі свідчить про правильну інтеграцію сервера з WAC. Додатково адміністратор може здійснювати моніторинг робочих станцій і серверів без необхідності ручного оновлення даних чи зупинки перевіряємих систем. Для подальшого тестування додаємо текст до документа, що збільшить його розмір, і перевіряємо, чи оновлюється відповідне значення ваги файлу в інтерфейсі WAC.

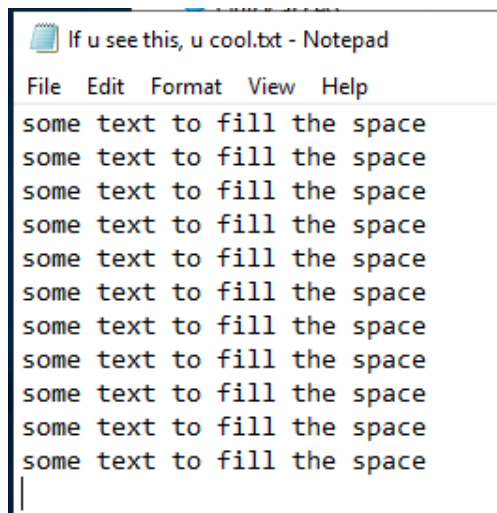


Рис. 11 Заповнення документа для збільшення ваги

Наступним кроком повернемося назад в WAC та переглянемо відповідне значення ваги файлу.

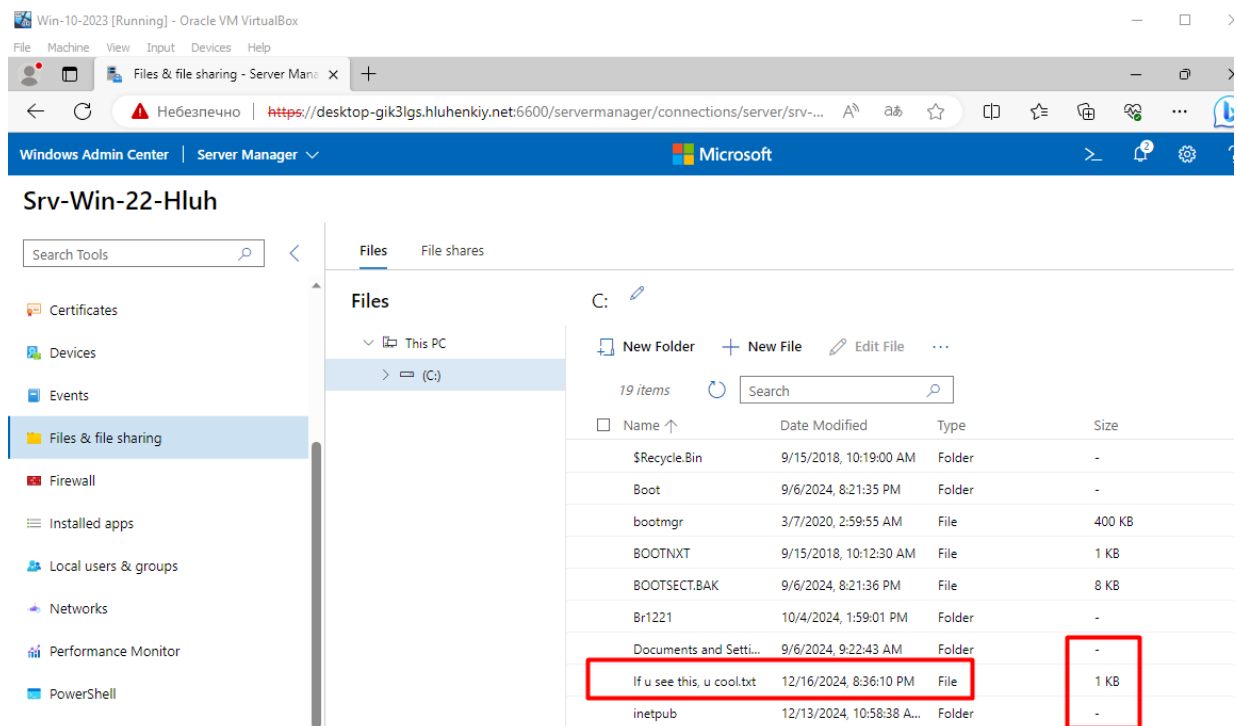


Рис. 12 Оновлення даних у реальному часі

Інтерфейс WAC надає не лише можливість переглядати наявні документи та папки, але й забезпечує функціонал для віддаленого управління та редагування їхнього вмісту. Ця можливість дозволяє адміністраторам виконувати адміністративні задачі централізовано, підвищуючи ефективність управління інфраструктурою. Для демонстрації цього функціоналу створюємо папку через інтерфейс WAC і перевіряємо її наявність на контролері домена Srv-

Win-22-Hluh. Успішне відображення папки підтверджує коректність інтеграції WAC із сервером і забезпечує основу для подальшого тестування можливостей віддаленого управління файлами та теками.

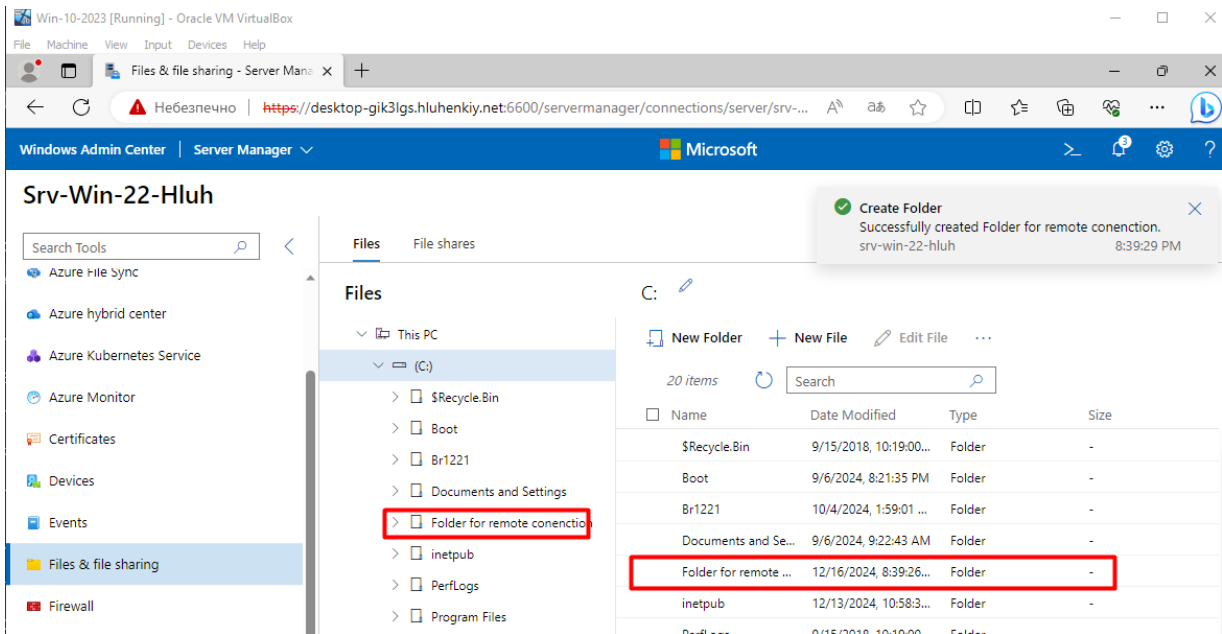


Рис. 13 Успішне створення папки на сервері

Після створення папки через інтерфейс WAC у верхньому правому куті відображається повідомлення про успішне виконання змін. У відповідній вкладці, позначеній як піктограма дзвінка, фіксуються записи про всі успішні та неуспішні операції в програмі. У разі виникнення помилки WAC забезпечує опис її суті та пропонує короткі рекомендації щодо її усунення, що полегшує діагностику та усунення проблем адміністратором. Для підтвердження коректності змін перевіряємо наявність створеної папки безпосередньо на сервері Srv-Win-22-Hluh.

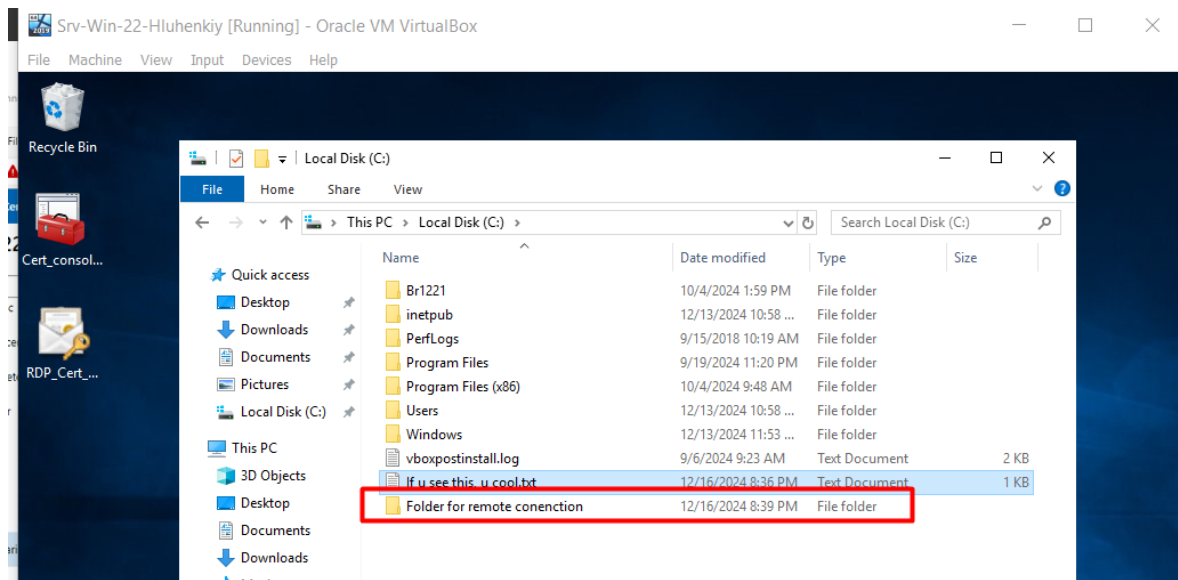


Рис. 14 Підтвердження створення папки на сервері

Перевіривши функціонал утиліти "File & Sharing", перейдемо до наступного інструменту "Remote Desktop" (віддалений робочий стіл), який є важливим компонентом WAC. Цей інструмент дозволяє здійснювати пряме підключення до цільової робочої станції для виконання необхідних конфігураційних змін. Під час сеансу віддаленого доступу станція переходить у режим очікування, забезпечуючи економію ресурсів. Якщо користувач локально увімкне машину та виконає авторизацію, сеанс віддаленого доступу буде перерваний, вимагаючи повторного запуску для відновлення з'єднання.

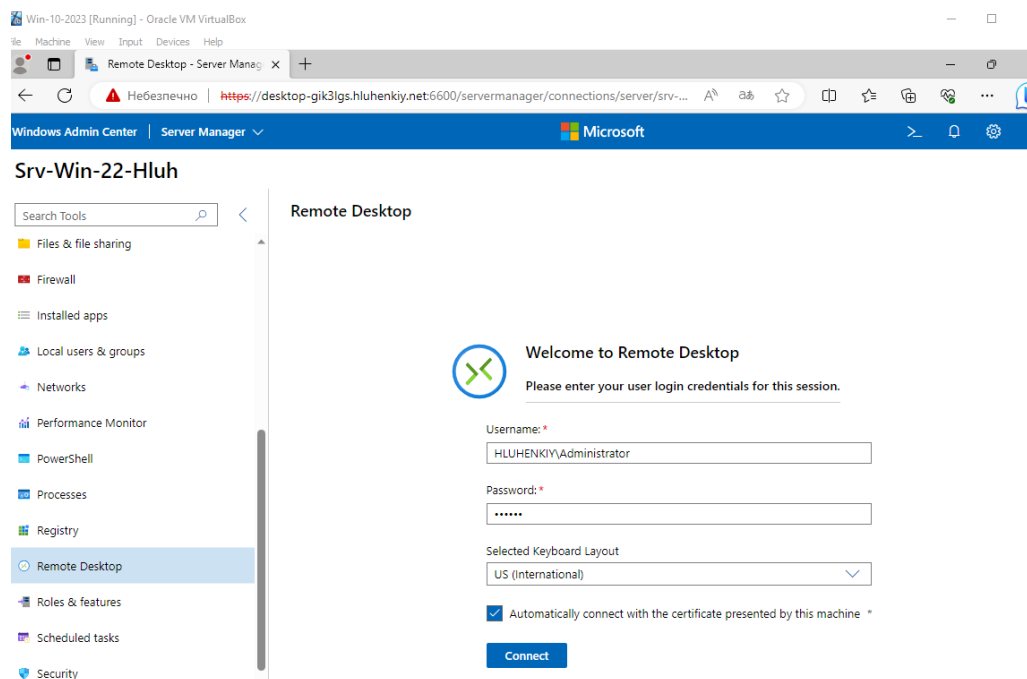


Рис. 15 Підключення через віддалений доступ

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

Для успішного підключення через віддалений робочий стіл необхідно ввести автентифікаційні дані користувача, який здійснюватиме з'єднання, а також вказати відповідну розкладку клавіатури для забезпечення коректної роботи. Після встановлення з'єднання відкривається вікно віртуальної машини, де доступні всі функції, характерні для прямого використання станції, включаючи налаштування системних параметрів і виконання адміністративних задач.

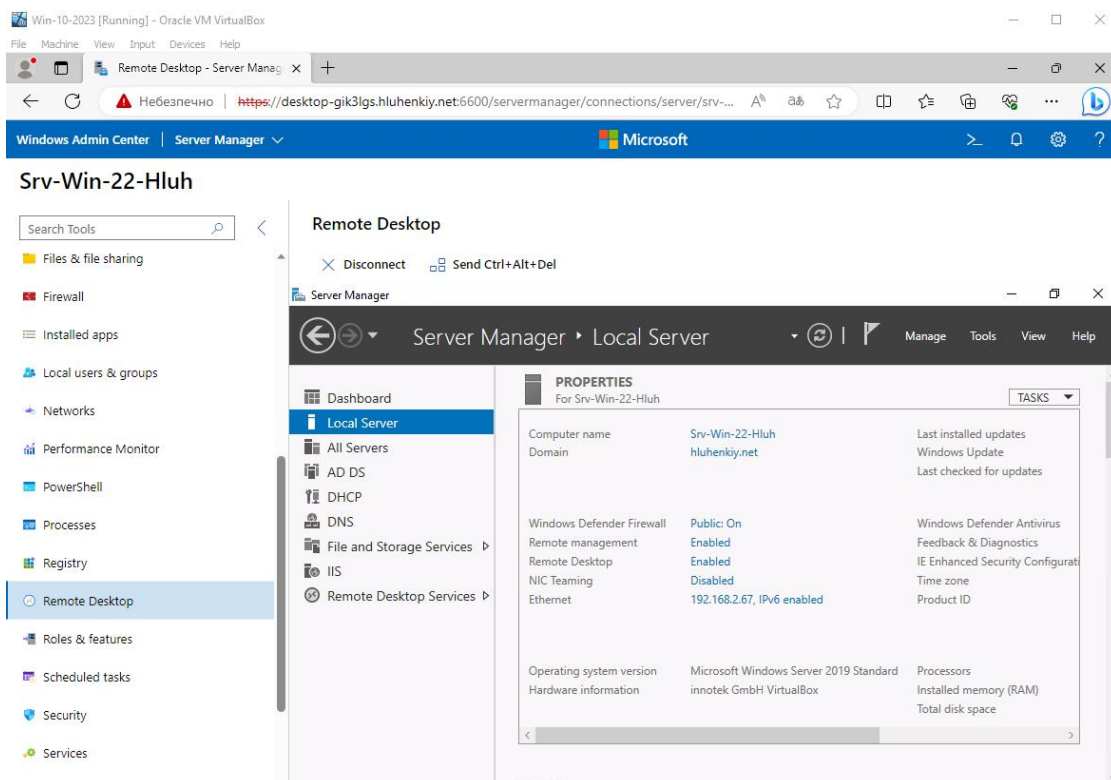


Рис. 16 Успішне під'єднання до віддаленого робочого столу

На основі попередньо описаних особливостей роботи віддаленого робочого столу через WAC виконаємо перевірку сценарію, коли користувач локально увійде в систему на машині, що використовується для віддаленого доступу. Цей тест дозволить підтвердити поведінку системи в умовах одночасного доступу та оцінити коректність роботи інструменту "Remote Desktop".

Для цього ініціалізуємо сеанс віддаленого робочого столу до робочої станції через WAC, а потім виконаємо локальний вхід у систему на цій же машині, використовуючи облікові дані користувача. Як очікується, локальний вхід призведе до переривання віддаленого сеансу, що вимагатиме повторного запуску

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

з'єднання через WAC для відновлення доступу, підтверджуючи описану раніше поведінку системи.

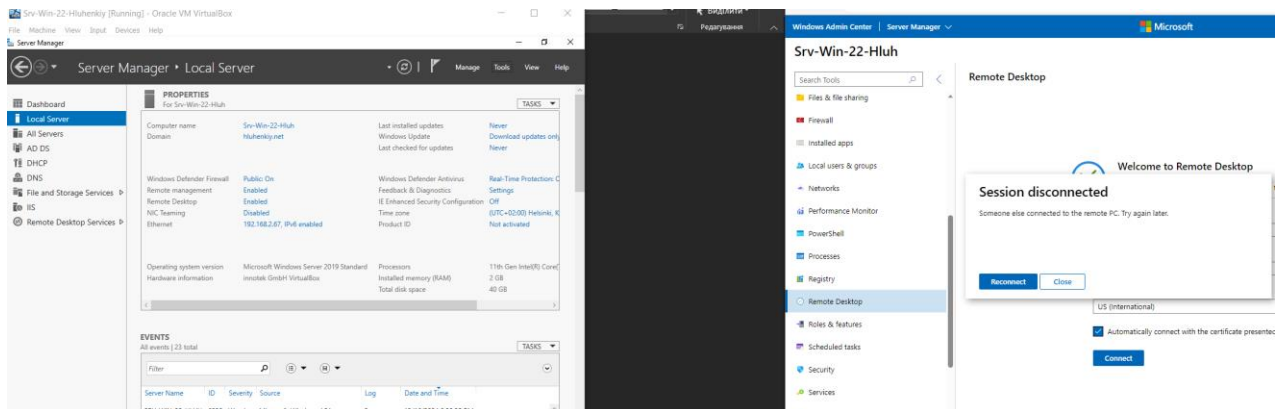


Рис. 17 Відключення від Remote Desktop

3.3.1 Перевірка функціональності Windows Admin Center за допомогою PowerShell-скрипту

Після встановлення WAC і підтвердження його коректної роботи перейдемо до наступного етапу - створення PowerShell DSC скрипту. Цей скрипт буде виконано віддалено на робочій станції та сервері з метою забезпечення моніторингу вільного місця на дисках. Даний PowerShell DSC скрипт передбачає автоматичне створення лог-файлу у разі перевищення встановленого порогового значення вільного простору, що сприятиме подальшому аналізу стану системи.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

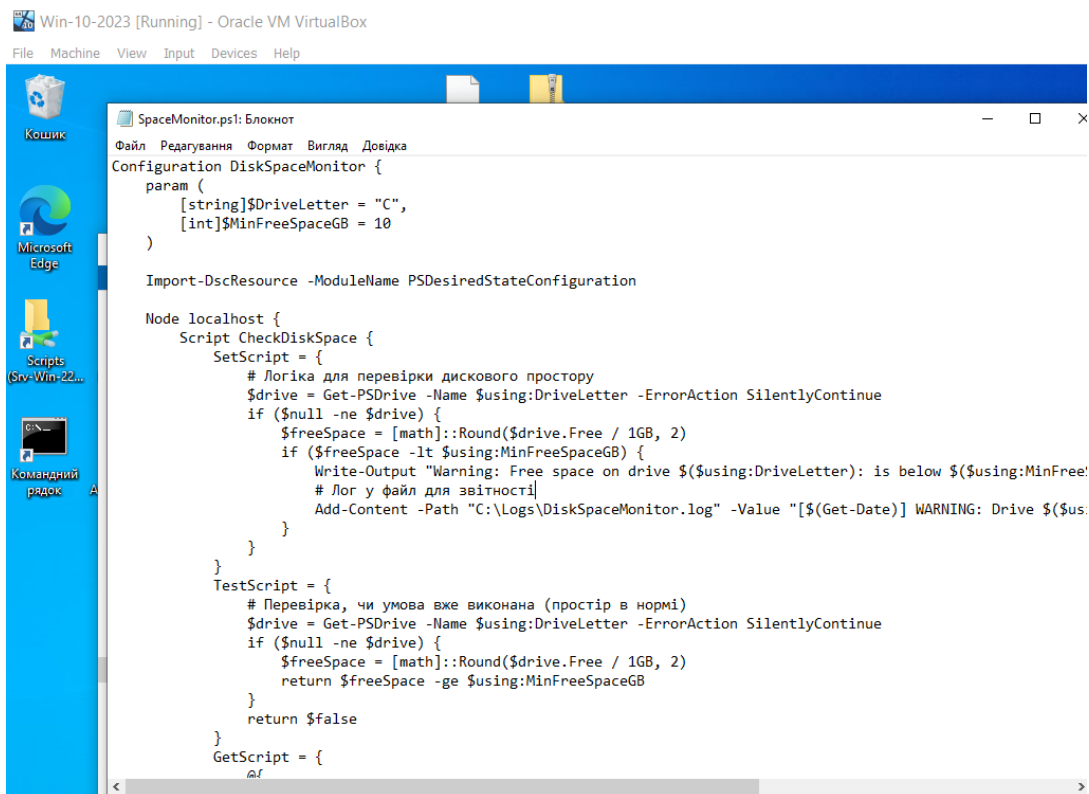


Рис. 18 Створення DSC-конфігурації

Логіка роботи PowerShell DSC-конфігурації базується на визначенні бажаного стану системи через спеціалізовані ресурси.

File забезпечує перевірку та створення директорії C:\Logs для збереження лог-файлів, якщо вона відсутня.

Script включає три ключові компоненти:

- **TestScript:** Аналізує стан дисків, визначаючи обсяг вільного простору. Якщо обсяг вільного місця на диску становить менше 50 ГБ, система вважається невідповідною бажаному стану.
- **SetScript:** У разі виявлення невідповідності генерує лог-файл із попередженням про проблему.
- **GetScript:** Повертає поточний результат перевірки для оцінки стану.

Мінімальний поріг вільного простору встановлено на рівні 50 ГБ навмисно, щоб забезпечити видимі результати під час кожного виконання скрипту. Це надає можливість генерувати новий лог-файл із інформацією про час виконання, статистику використаної пам'яті та стан дисків, що полегшує відстеження змін у системі.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

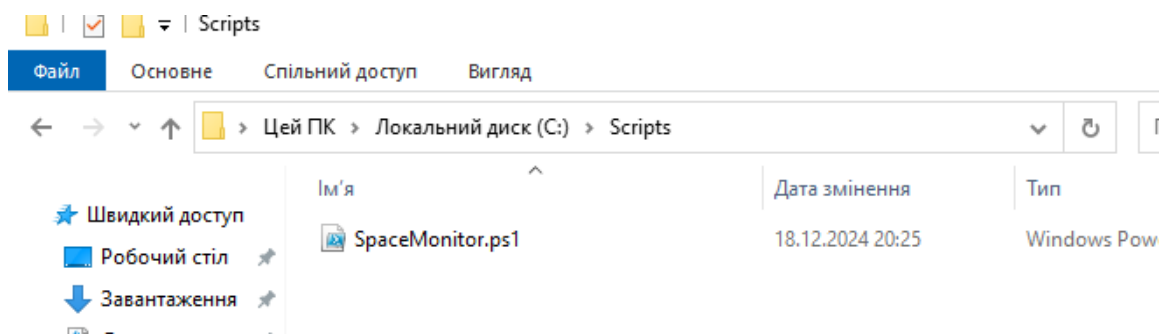


Рис. 19 Створення DSC-конфігурації

Для завершення завдання необхідно запустити створений PowerShell DSC-скрипт із застосуванням тимчасового виключення політики виконання, що дозволяє уникнути зміни постійних політик і мінімізує ризики безпеки під час запуску. Для цього використовуємо команду запуску з відповідним параметром виключення, забезпечуючи безпечне виконання скрипту на цільовій системі.

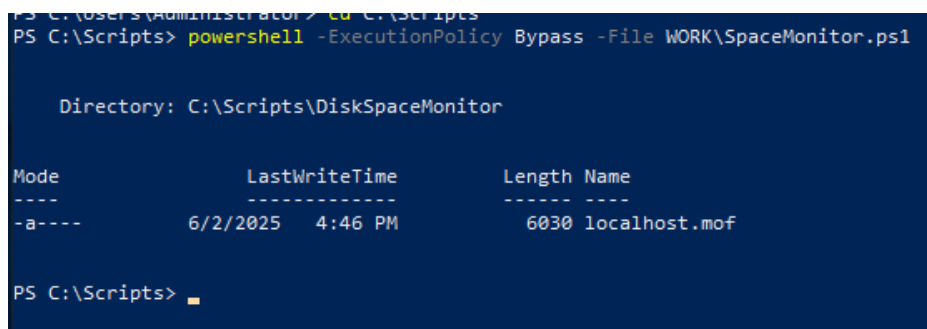


Рис. 20 Виконання конфігурації

Після виконання скрипту генерується MOF-файл, який слугує для перевірки стану системи, зокрема оцінки обсягу вільного місця на дисках. Для активації конфігурації застосовуємо команду `Start-DscConfiguration`, яка ініціалізує процес перевірки та, у разі виявлення невідповідностей, виконує дії, визначені в скрипті. У нашому випадку скрипт налаштовано на створення лог-файлу при обсязі вільного місця, меншому за встановлений поріг у 50 ГБ.

```

PS C:\Scripts> Start-DscConfiguration -Path C:\DiskSpaceMonitor -Wait -Verbose
VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, 'methodName' =
SendConfigurationApply,'className' = MSFT_DSCLocalConfigurationManager,'namespaceName' =
root\Microsoft\Windows\DesiredStateConfiguration'.
VERBOSE: An LCM method call arrived from computer SRV-WIN-22-HLUH with user sid
S-1-5-21-3247501395-3078649286-117273642-500.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Set ]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Resource ] [[Script]CheckDiskSpace]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Test ] [[Script]CheckDiskSpace]
VERBOSE: [SRV-WIN-22-HLUH]: [[Script]CheckDiskSpace] Perform operation 'Query CimInstances'
with following parameters, 'queryExpression' = SELECT * FROM Win32_LogicalDisk WHERE DeviceId='C:', 'queryDialect' =
WQL, 'namespaceName' = root\cimv2'.
VERBOSE: [SRV-WIN-22-HLUH]: [[Script]CheckDiskSpace] Operation 'Query CimInstances'
complete.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Test ] [[Script]CheckDiskSpace] in 0.0320 seconds.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Set ] [[Script]CheckDiskSpace]
VERBOSE: [SRV-WIN-22-HLUH]: [[Script]CheckDiskSpace] Performing the operation
"Set-TargetResource" on target "Executing the SetScript with the user supplied credential".
VERBOSE: [SRV-WIN-22-HLUH]: [[Script]CheckDiskSpace] Perform operation 'Query CimInstances'
with following parameters, 'queryExpression' = SELECT * FROM Win32_LogicalDisk WHERE DeviceId='C:', 'queryDialect' =
WQL, 'namespaceName' = root\cimv2'.
VERBOSE: [SRV-WIN-22-HLUH]: [[Script]CheckDiskSpace] Operation 'Query CimInstances'
complete.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ] [[Script]CheckDiskSpace] in 0.0620 seconds.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Resource ] [[Script]CheckDiskSpace]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ] in 0.1850 seconds.
VERBOSE: Operation 'Invoke CimMethod' complete.
VERBOSE: Time taken for configuration job to complete is 0.42 seconds
PS C:\Scripts>

```

Рис. 21 Виконання скрипту

Перевіряємо теку C:\Logs на наявність створеного лог-файлу. Оскільки обсяг вільного місця на диску становить 28 ГБ, що є нижчим за встановлений поріг у 50 ГБ, у лог-файл автоматично додається попередження з інформацією про невідповідність. Це підтверджує коректну роботу скрипту та його здатність виявляти критичні стани системи.

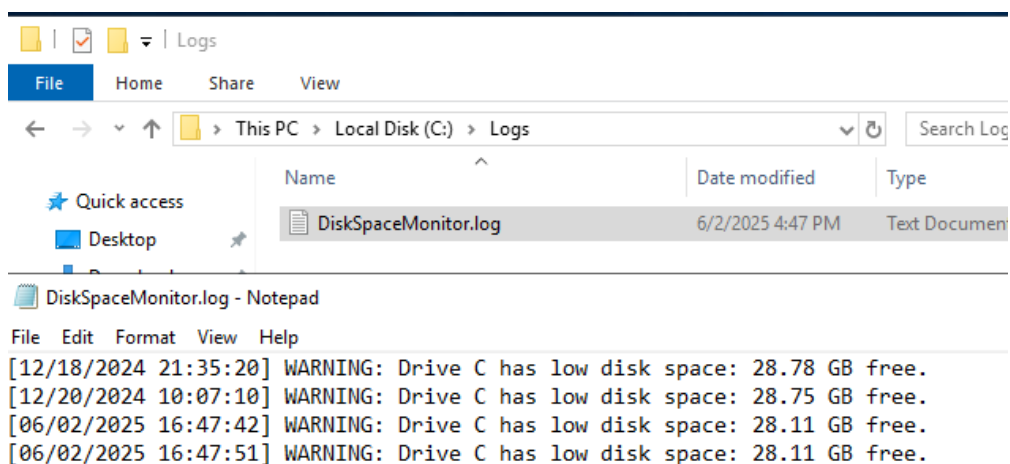


Рис. 22 Результат виконання скрипту

Далі виконуємо віддалений запуск цього ж скрипту на іншій віртуальній робочій станції через WAC. Для цього відкриваємо WAC, переходимо до утиліти віддаленого виконання PowerShell і запускаємо раніше створений скрипт. Одразу після виконання вводимо команду Get-Content для перевірки вмісту лог-файлу, що дозволяє переконатися в успішному застосуванні конфігурації та коректності віддаленої автоматизації.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

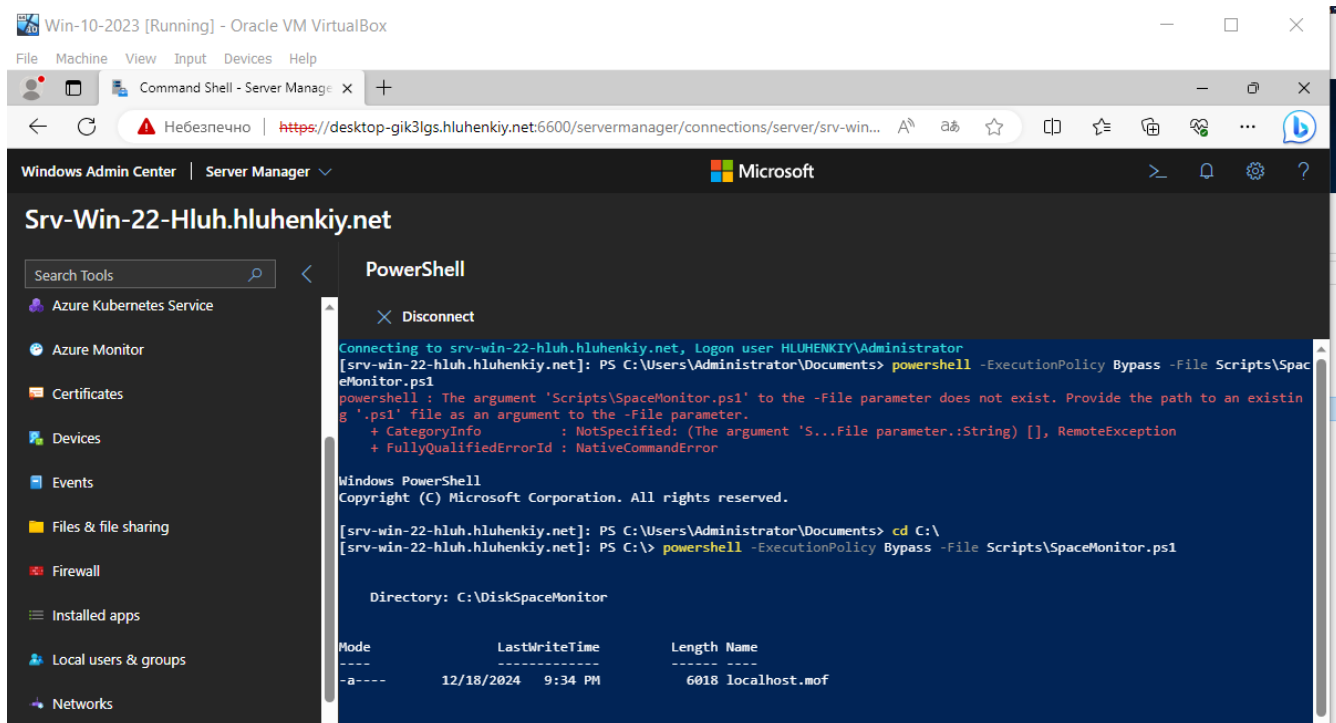


Рис. 23 Виконання скрипта на контролері домену

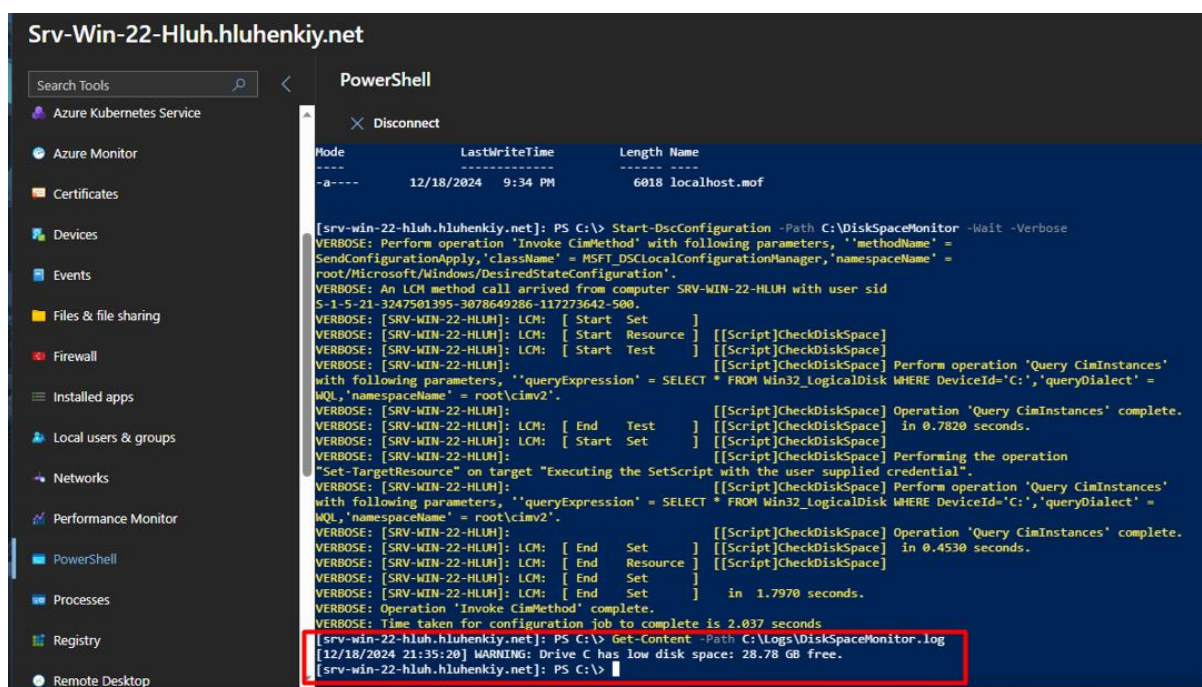


Рис. 24 Виконання скрипта на контролері домену

Виконаємо аналогічні дії, проте цього разу з іншою робочою станцією - Windows-Tester. Виконаємо скрипт та за допомогою системної команди «Get-Content» виведемо у консоль вміст log файлу, який створюється після успішного виконання команди.

		Глухенький Д.О.							Арк.
		Фальковський Л.І.							66
Змн.	Арк.	№ докум.	Підпис	Дата					

ККІКБ.КР.Б-123-03-ПЗ

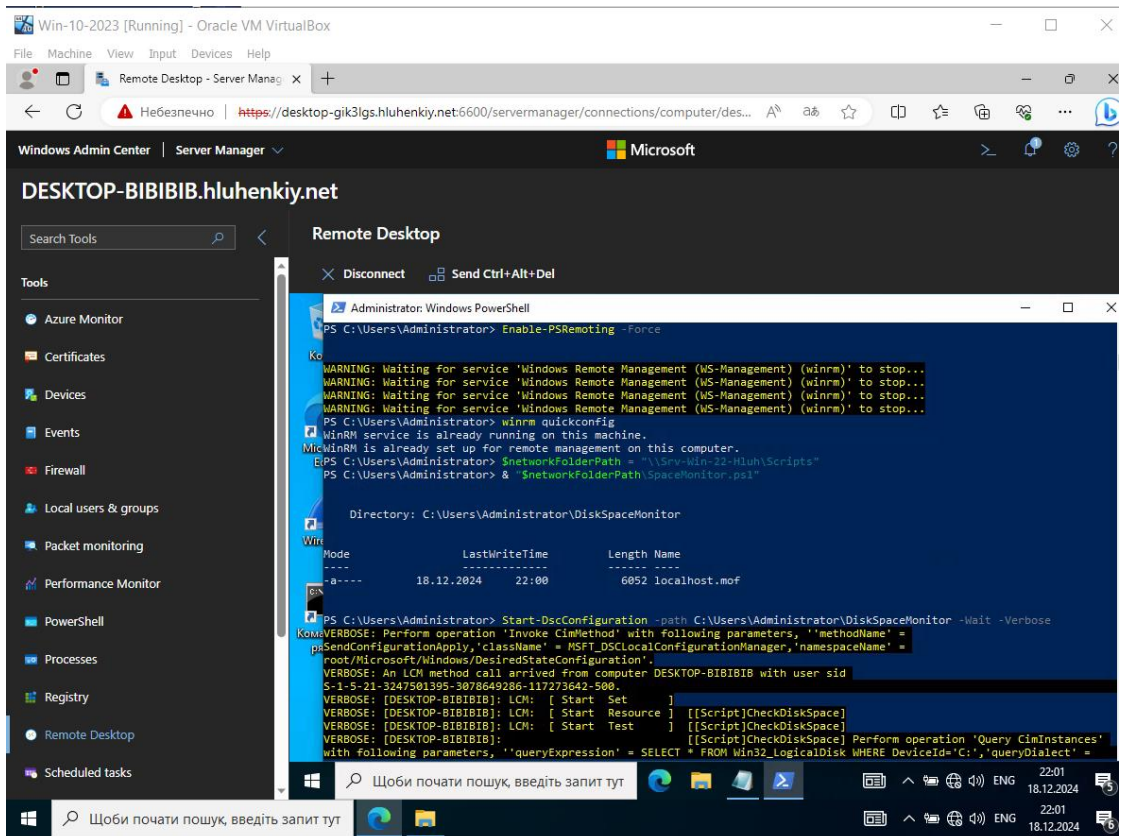


Рис. 25 Виконання скрипта на робочій станції

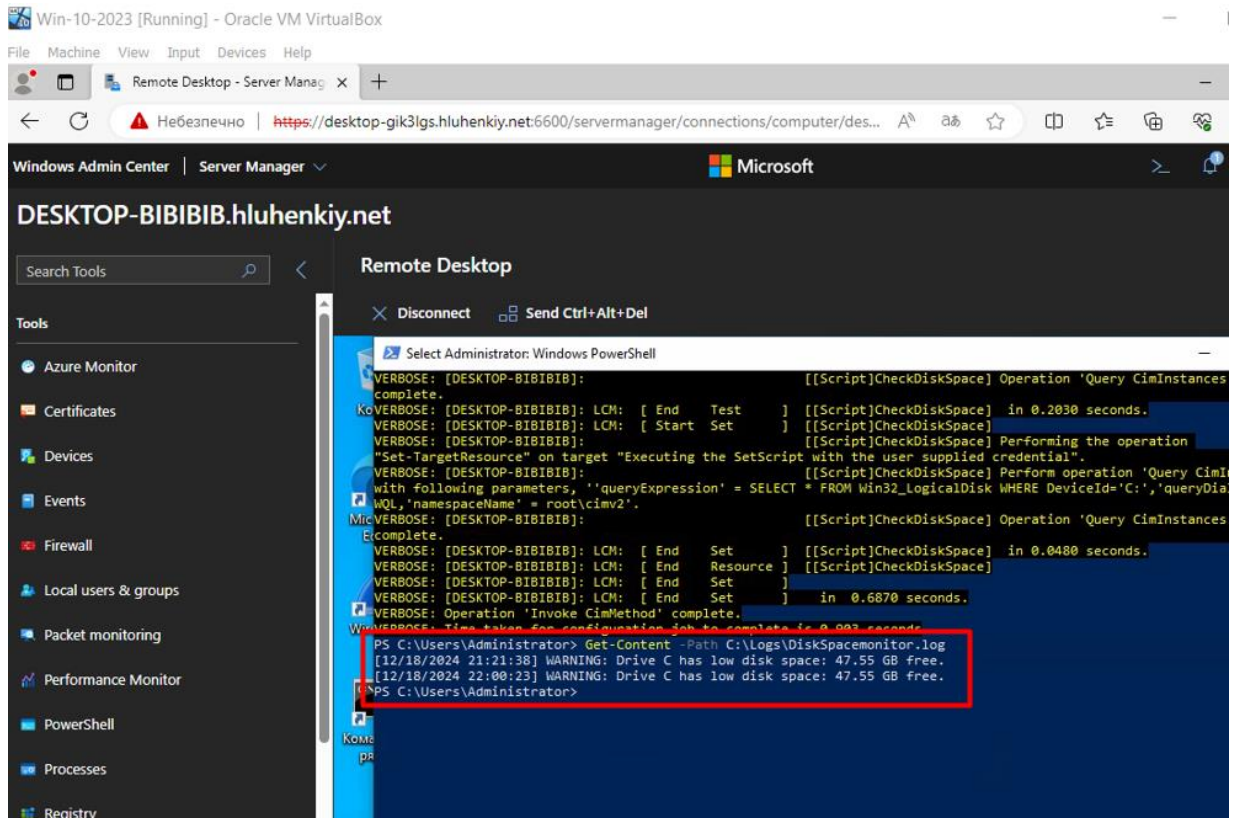


Рис. 26 Виконання скрипта на робочій станції

Таким чином, підтверджено коректну роботу Windows Admin Center, що дозволяє ефективно здійснювати моніторинг стану мережевої інфраструктури та

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

своєчасно виявляти потенційні загрози й проблеми. Переконавшись у відповідності виконаних налаштувань вимогам завдання, переходимо до наступного етапу - розробки проєкту. Повний вміст усіх розроблених скриптів і лог-файлів наведено в Додатку Б для детального аналізу та подальшого використання.

3.3.2 Налаштування Role-Based Access Control для доступу до сервісів та DHCP

У попередніх підпунктах було розглянуто встановлення та базову конфігурацію WAC. Також було проведено перевірку його функціональності за допомогою PowerShell-скрипту, це надало можливість практично підтвердити наші теоретичні висновки із минулих розділів. На даному етапі буде зосереджено всю увагу на налаштуванні технології Role-Based Access Control (RBAC) для управління доступом до сервісів і служб, що є важливим кроком для забезпечення безпеки та розподілу обов'язків між користувачами. У рамках цього підпункту буде описано створення користувачів, призначення їм відповідних ролей, зокрема для адміністрування DHCP, що дозволить оптимізувати управління доступом у межах домену.

Почнемо із створення конкретної ролі для адміністрування DHCP на Сервері №3. Це має бути група користувачів, яка буде мати обмежені права, та зможе під'єднуватись до Серверу №3 для проведення налаштувань. Створимо групу LimitedGroup куди одразу додамо користувача DHCPadmin:

		Глухенький Д.О.			KKIKB.KP.B-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

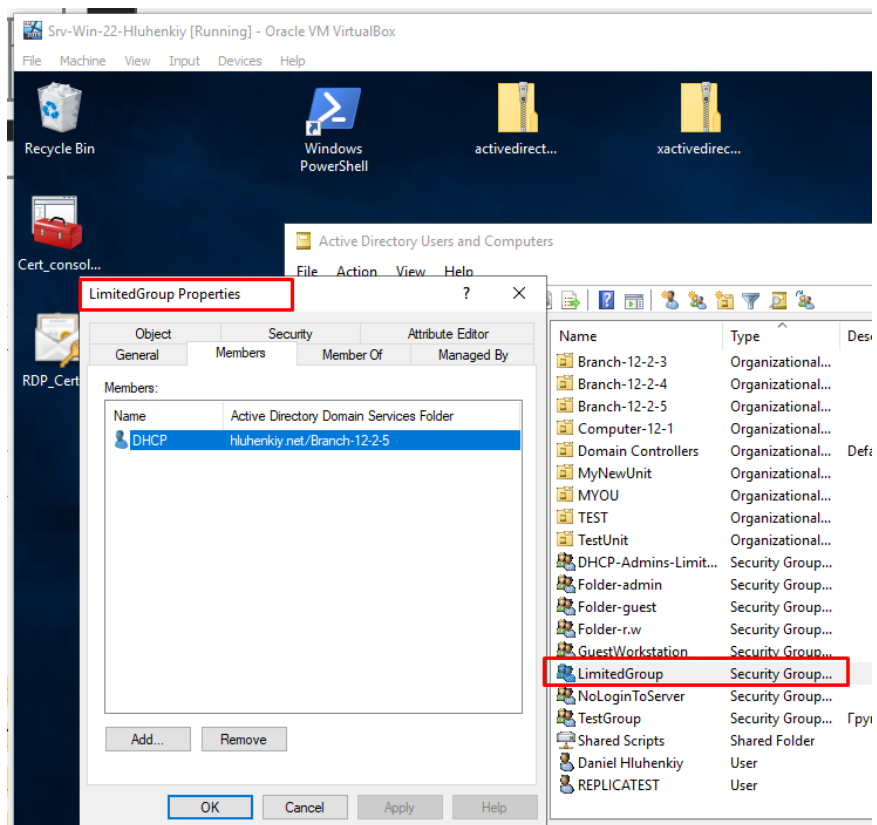


Рис. 27 Створення групи користувачів

Далі, використовуючи інструмент Group Policy Management, створюємо правило політик для групи LimitedGroup. Дана група не матиме доступу до інструментів налаштування домену. На Сервері №3 відповідна група користувачів матиме змогу відкрити налаштування DHCP server служби, проте доступ до інших буде заборонений.

Використовуючи облікові дані користувача із групи LimitedGroup здійснюємо вхід на віртуальній машині Сервер №3 та пробуємо запустити різні контрольні служби.

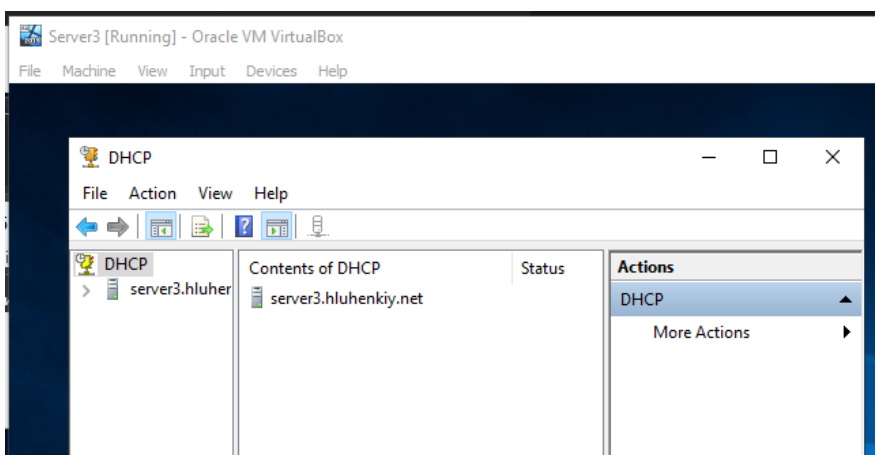


Рис. 28 Перевірка роботи політик

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

Тепер варто перевірити доступ до інших доменних служб, адже критично важливо, щоб користувачі з обмеженими правами не могли змінювати важливі налаштування в чутливих програмах.

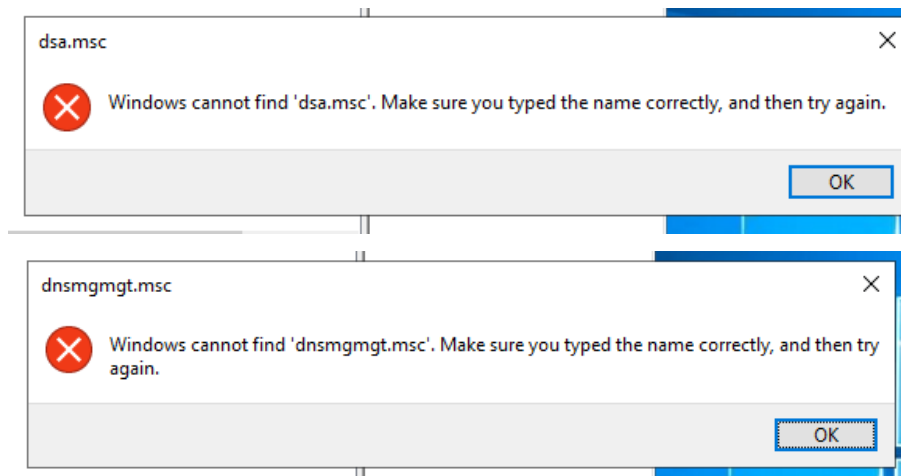


Рис. 29-30 Перевірка доступу до служб

На фото вище зображені дві системні помилки, а саме: доступ до служби active directory та доступ до служби керування DNS. Дані повідомлення підтверджують правильність налаштувань доступу, адже на відповідному сервері навіть не було встановлено дані служби. Це потрібно для того, щоб збільшити рівень безпеки домена від несанкціанованих зламів.

Наступним кроком буде перевірити доступ даної групи користувачів до головного контролеру домену. Згідно налаштуванням конкретна група користувачів не зможе виконувати адміністрування на головному Сервері №1 через відсутність достатньо високих привілеїв. Перевіримо налаштування здійснивши вхід на Сервер №1 використовуючи облікові дані користувача із групи LimitedGroup.

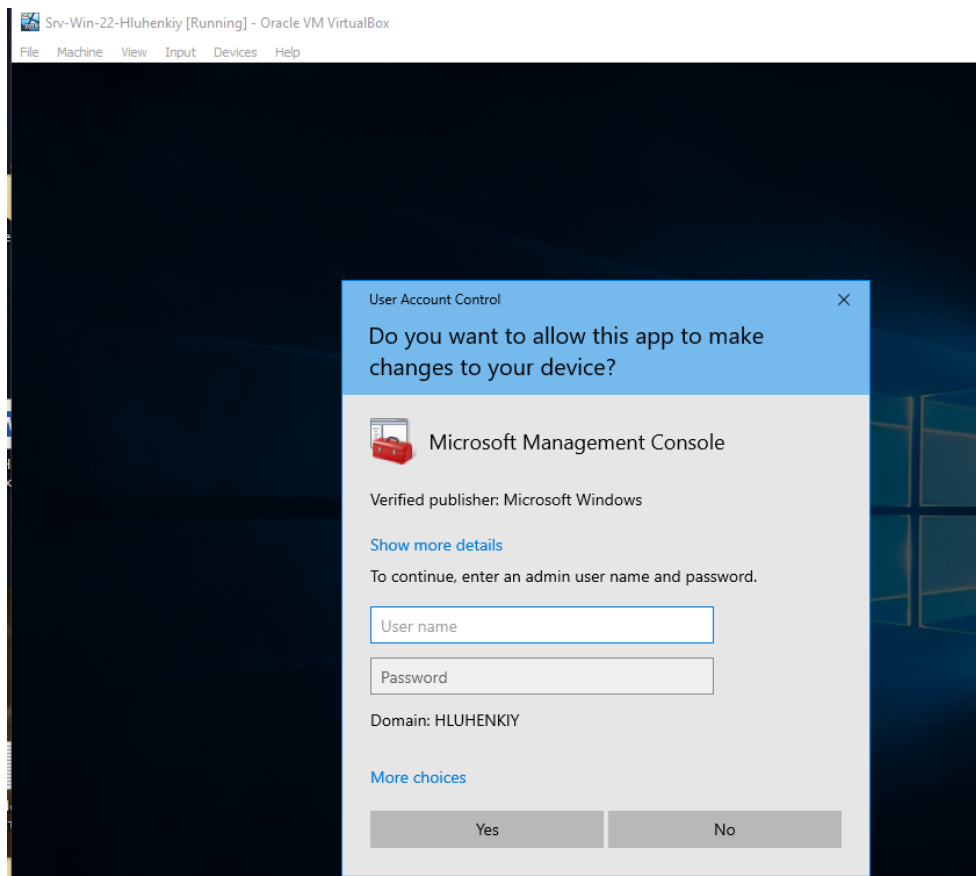


Рис. 31 Перевірка наданих прав

Фото вище демонструє правильність проведених раніше налаштувань. Користувач не може напряму запустити ту чи іншу програму або службу, для цього йому необхідно буде підтвердити облікові дані адміністратора домену, інакше Сервер №1 попередить про помилку входу.

Ще одним важливим питанням є список користувачів у яких наявний доступ до входу та керуванню серверами в мережі. Питання безпеки є ключовим аспектом виконання проєкту, тому було потрібно створити групу користувачів які не матимуть доступу входу до серверів. Назвемо її NoLigInToServer:

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				71
Змн.	Арк.	№ докум.	Підпис	Дата		

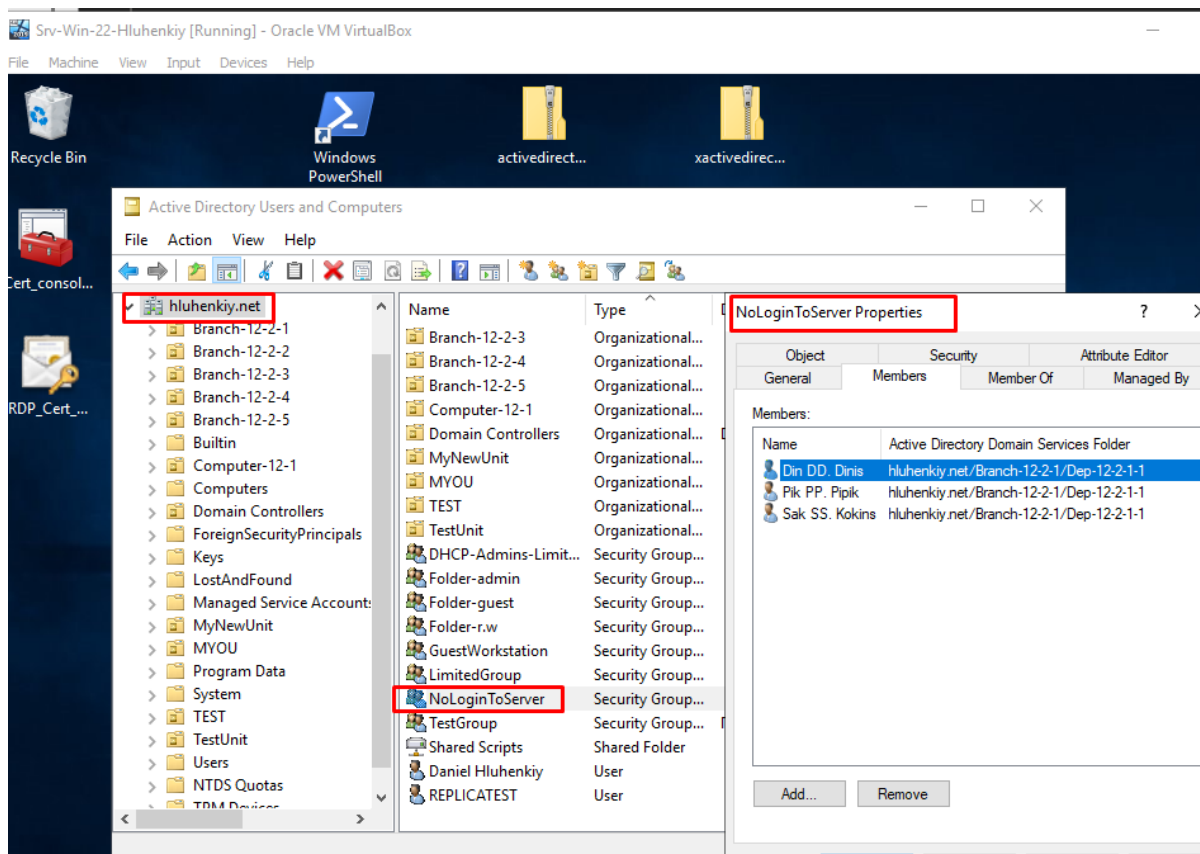


Рис. 32 Створення групи з обмеженням доступом до станцій

Відповідну групу додамо до списку «**Deny log on locally**» у службі **Local Secutiry Policy**. Це системне правило яке забороняє доступ до системи всім групам та користувачам які наявні в прикріпленому списку. Це дозволяє гнучко керувати правами доступу для небажаних користувачів, наприклад гостей, які повинні мати право лише на перегляд певної категорії сервісів. Це надає можливість оптимізувати управління доступом у межах конкретного домену.

Також служба **Local Secutiry Policy** дозволяє налаштувати доступ до підключення через віддалений робочий стіл (Remote Desktop), задавати правила налаштування паролів (складність, термін дії), керувати доступом до ресурсів через призначення прав користувачам і групам та інше. Крім того, можна обмежувати використання зовнішніх пристроїв (наприклад, USB), налаштовувати аудит подій і контролювати запуск програм, що підвищує загальний рівень безпеки системи.

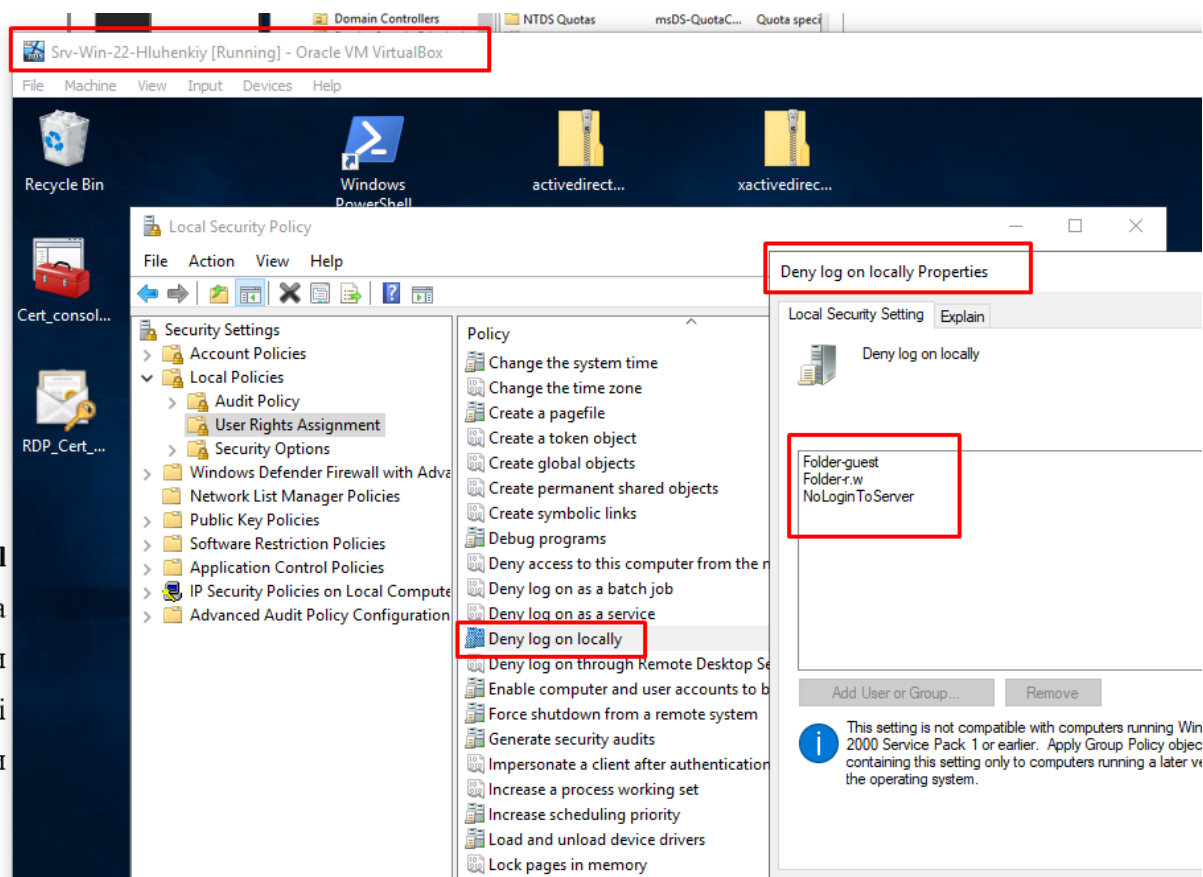


Рис. 33 Налаштування правила доступу до системи

Подібні права доступу користувачів необхідно налаштувати на кожному із трьох серверів домена. Такий підхід забезпечить додаткову безпеку та розподіл прав та обов'язків між користувачами.

Після налаштувань відповідних прав необхідно виконати перевірку проведеної роботи шляхом підключення до серверу під виглядом користувача із групи LimitedGroup. Використовуючи облікові дані користувача der-11-ppr здійснимо вхід на Сервер №1 та Сервер №3 та перевіримо якість налаштувань.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

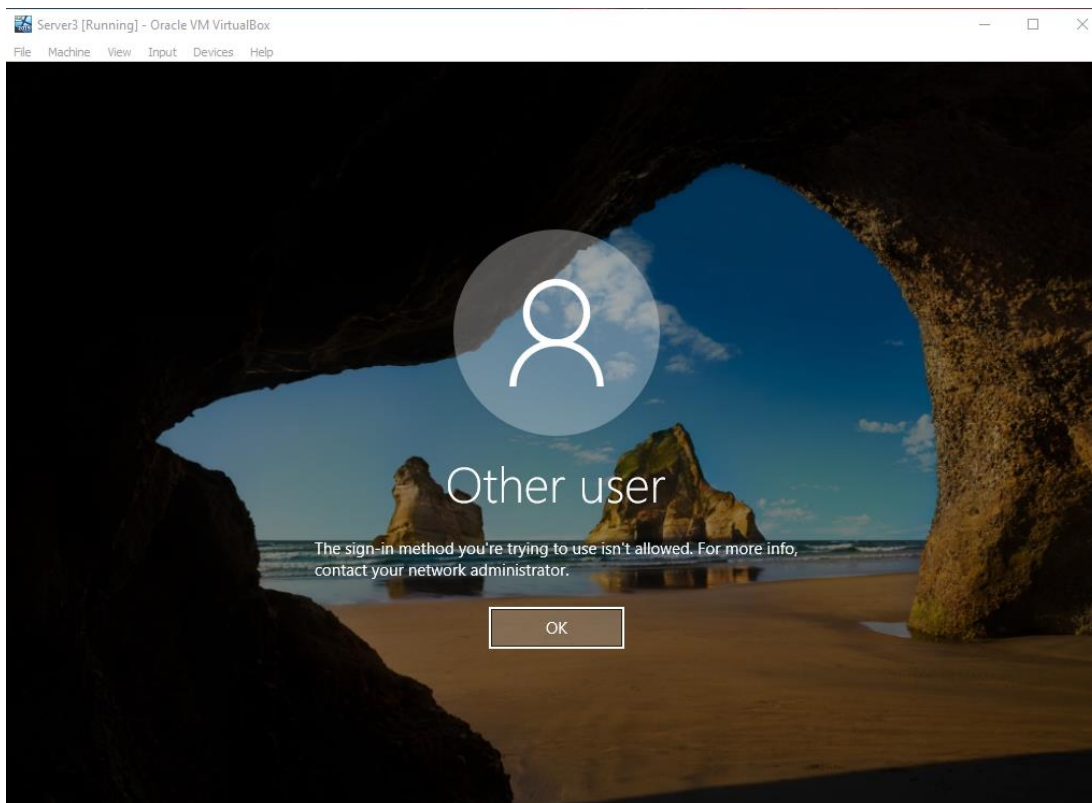


Рис. 34 Помилка входу на Сервері №3

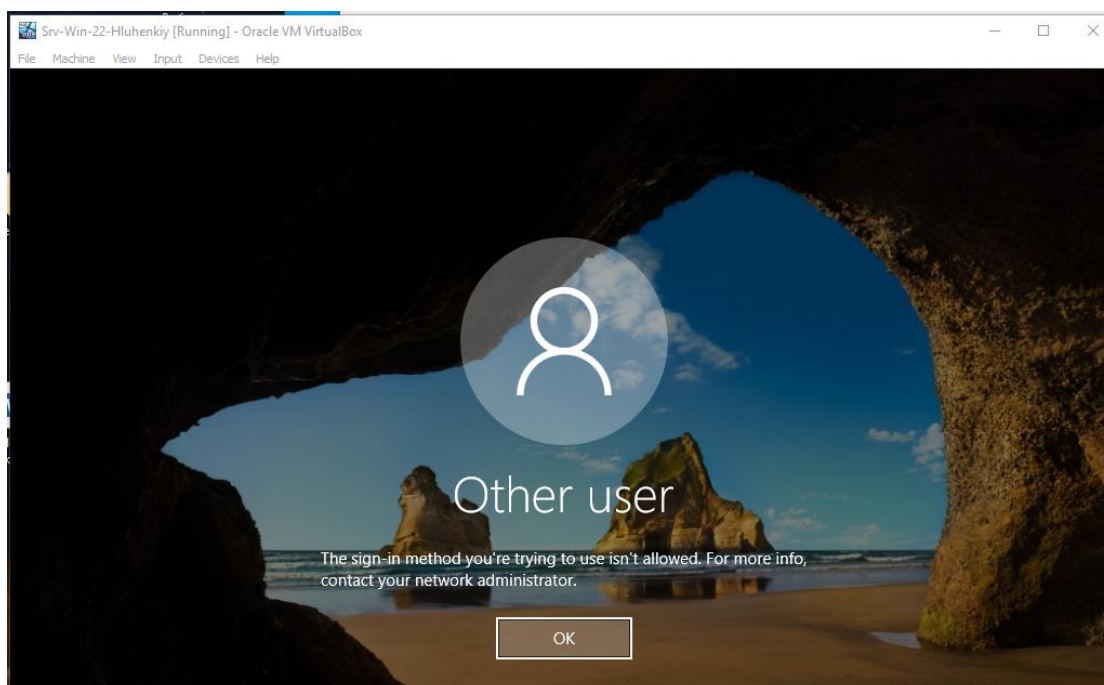


Рис. 35 Помилка входу на Сервері №1

Повідомлення на екрані блокування підтверджує правильність проведених налаштувань, що дає нам змогу перейти до наступного кроку розподілення ролей у домені. На цей раз створимо декілька системних папок на диску C:\ та налаштуємо до них доступ лише конкретним групам користувачів.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				74
Змн.	Арк.	№ докум.	Підпис	Дата		

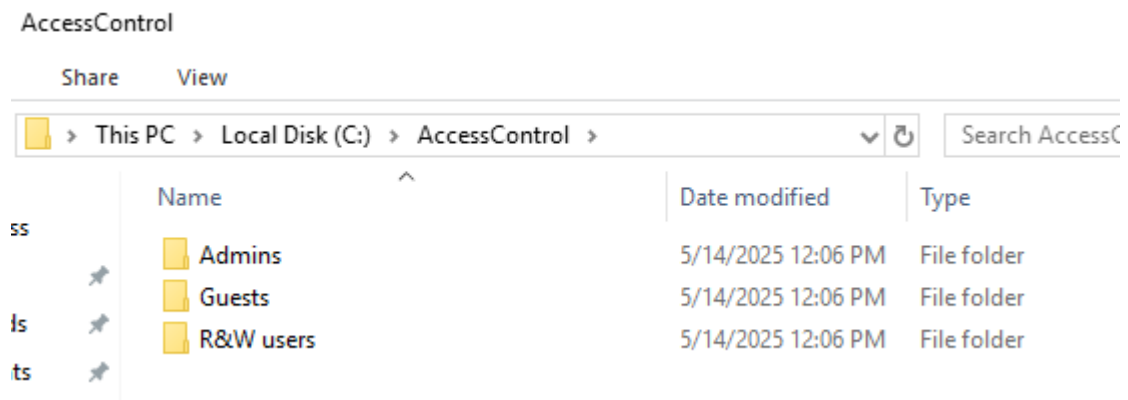


Рис. 36 Створення тек

Використовуючи функцію налаштування теки можна обрати бажаних та небажаних користувачів. В нашому випадку це будуть групи користувачів Folder-Admin, Folder-Guest, Folder-R.W. Як зрозуміло із назви груп, користувачі будуть мати доступ до своєї конкретної папки, та, якщо дозволяють привілеї, доступ до папки яка нижче за рівнем доступу. Нижче показана схема роботи даних дозволів.

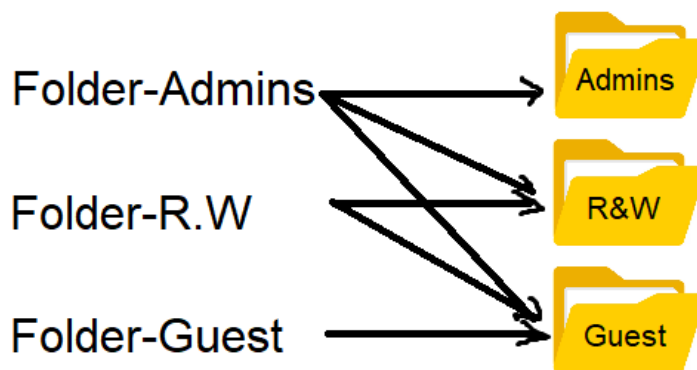


Рис. 37 Схема розподілу дозволів

Група Folder-Admin має доступ до всіх трьох папок, в той час як група Folder-Guest має лише дозвіл на читання власної папки. Цей принцип дозволяє забезпечити гнучкий контроль привілеїв у домені. Далі проведемо перевірку доступу до даних папок. Використовуючи облікові дані користувачів із цих груп підключемось до різних машин та перевіримо правильність виконаних налаштувань.

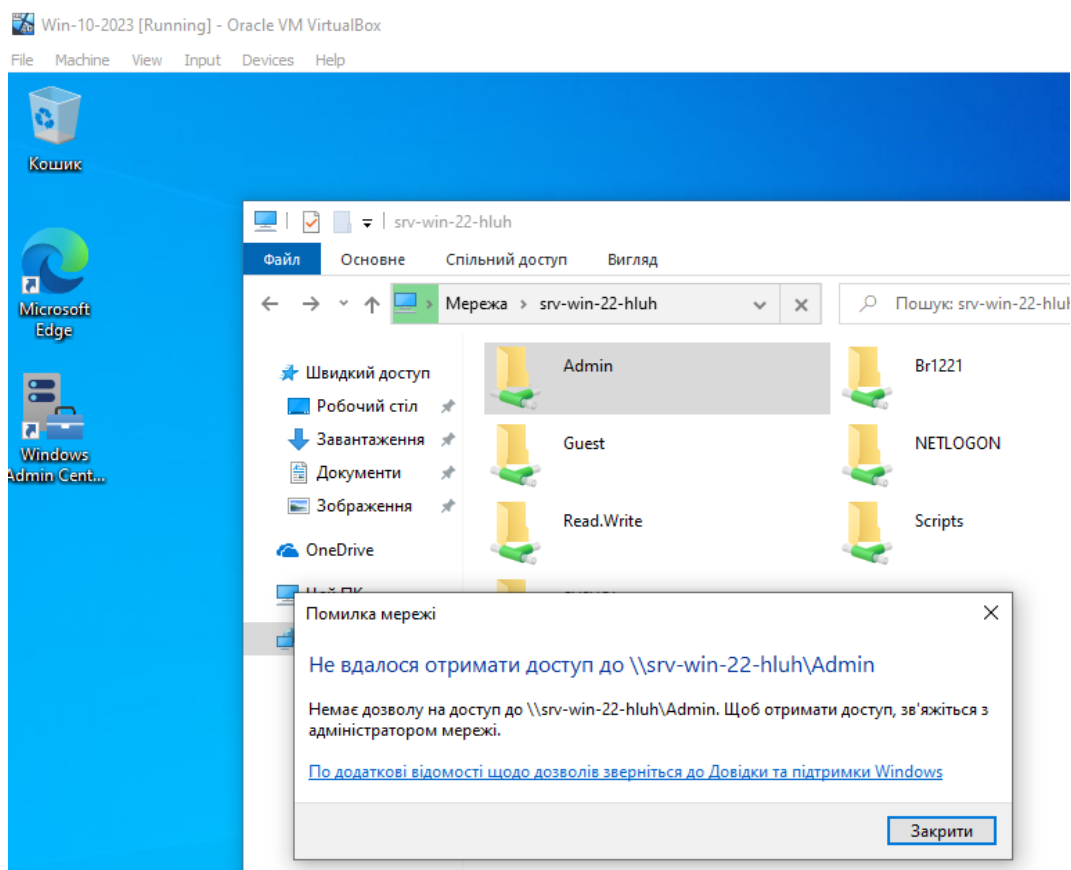


Рис. 38 Спроба входу до теки

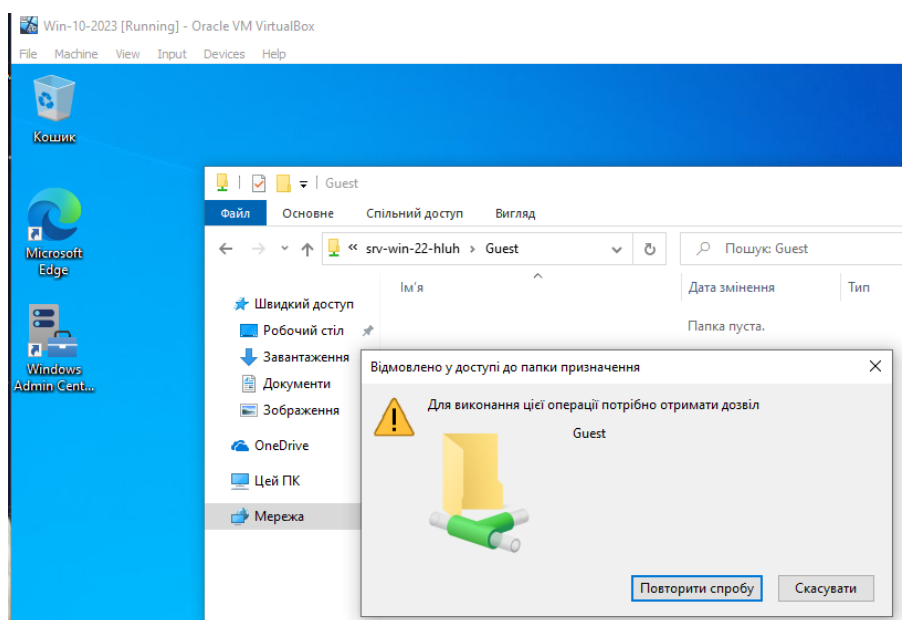


Рис. 39 Спроба редагування теки

Як видно із наведених вище додатків налаштування прав доступу були виконані правильно. Користувач із групи Folder-Guest спробував під'єднатись до теки Admins та Read.Write, проте у доступі йому було відмовлено. Проте користувач зміг зайти в теку Guest, хоча все ще мав заборону на редагування її

вмісту. Аналогічні дослідження можна провести для групи Folder-Admin та Folder-R.W, перевірити чи наявний доступ до власної папки, та чи можуть, або ж не можуть, під'єднатись до теки яка має вищий та нижчий пріорітети доступу. Для більш детального опрацювання матеріалу наявна можливість переглянути скріншоти дослідження які прикріплені у [Додатку Б](#).

Варто відмітити можливість налаштування доступу до окремих програм шляхом надання дозволів або встановлення заборон через інструмент AppLocker. AppLocker - це інтегрований у Windows інструмент який дозволяє адміністраторам створювати політики, які контролюють запуск програм, скриптів, бібліотек і інсталяційних файлів на основі їхніх атрибутів, таких як видавець, шлях або хеш файлу. Наприклад, можна дозволити запуск лише довірених додатків, заборонивши виконання неавторизованого програмного забезпечення. Це значно знижує ризик запуску шкідливих програм у системі. Цей інструмент налаштовується через Local Security Policy або групові політики домену, забезпечуючи гнучке управління безпекою на робочих станціях і серверах. Крім того, AppLocker підтримує аудит режиму, що дозволяє тестувати політики перед їхнім примусовим застосуванням, допомагаючи уникнути непередбачених обмежень для користувачів.

Під час проведення налаштувань Role Based Access Controll (RBAC) було залучено інструмент AppLocker для налаштування певних дозволів для конкретних груп користувачів. Основною групою для налаштувань була Guests. Для того, щоб заблокувати даним користувачам доступ до більшості важливих інструментів було використано групові політики та AppLoker відповідно. Нижче наведено фото процесу налаштувань.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				77
Змн.	Арк.	№ докум.	Підпис	Дата		

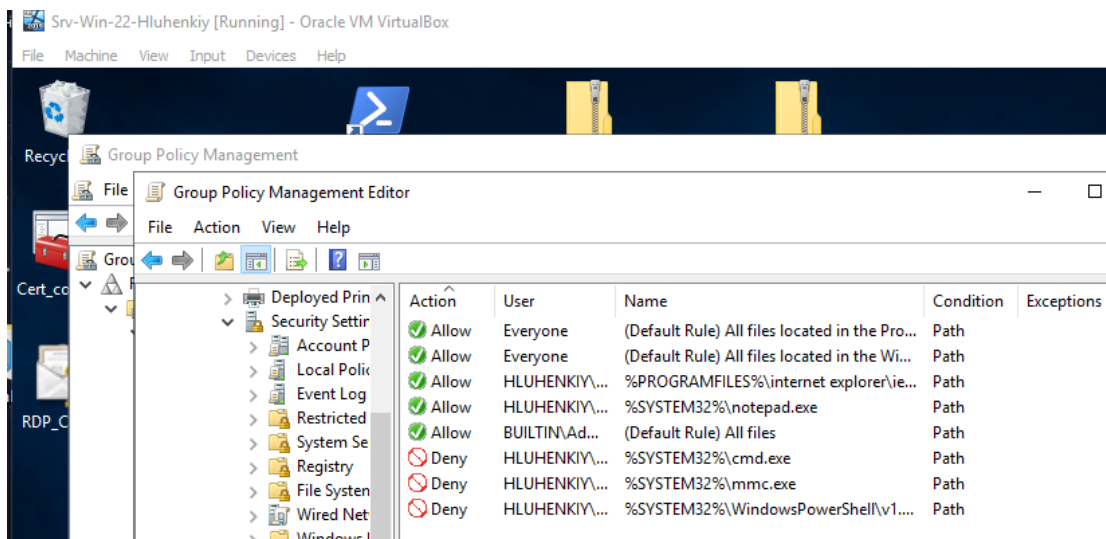


Рис. 40 Налаштування доступу для користувачів

Основною метою проведення налаштувань було заборонити групі користувачів користуватись важливими системними інструментами, неправильне використання яких могло б нашкодити системі чи домену в цілому. Для цього було заблоковано права доступу до таких інструментів як: Windows PowerShell, Command Prompt (CMD) та інструмент Microsoft Management Console (mmc.exe). Доступ до консолей powershell та cmd був заборонений щоб унеможливити несанкціоноване втручання в мережу. В свою чергу mmc також являє собою консоль керування системою.

ММС - це графічний інтерфейс, представлений файлом mmc.exe, який слугує для централізованого управління різними адміністративними інструментами (snap-ins), такими як Local Security Policy, Group Policy Editor або Event Viewer. Цей інструмент дозволяє адміністраторам конфігурувати політики, моніторити систему та управляти ресурсами через єдине вікно. Заборона гостям доступу до mmc.exe є доцільною через потенційні ризики безпеки. Гості, як правило, мають обмежені права, але доступ до ММС може дозволити їм змінити критичні політики. Це може призвести до несанкціонованих змін або компрометації системи, тому доступ до цього інструменту має бути обмеженим для адміністраторів із належними повноваженнями.

3.4 Автоматизація адміністрування за допомогою PowerShell Desired State Configuration

У попередньому підпункті було розглянуто ключові переваги використання PowerShell DSC для автоматизації адміністрування мережевої інфраструктури, зокрема створення скрипту для перевірки вільного місця на диску віртуальної машини через Windows Admin Center. На цьому етапі буде розроблено низку додаткових скриптів, які охоплюють роботу з Active Directory та операції на рівні домену, а також інші аспекти адміністрування. Нижче вказано як ці скрипти згруповано за функціональним призначенням у чотири основні категорії: управління користувачами та доступом, налаштування мережевих служб, забезпечення безпеки, а також моніторинг і оптимізація роботи системи.

3.4.1 Автоматизація управління користувачами та доступом у домені

Першою категорією для опрацювання буде управління користувачами та доступом у домені. Важливим аспектом є автоматизація простих рпоцесів, які зазвичай займають мало часу для налаштування. Проте при масштабуванні системи виконання навіть таких простих задач викликає складнощі. Саме тому варто створити скрипти для автоматизації таких дій, щоб адміністратор міг виконувати більш важливі налаштування під час автоматичного виконання скрипта. Для прикладу створимо DSC-скрипт який буде додавати організаційну одиницю з певною назвою в наш домен.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		79

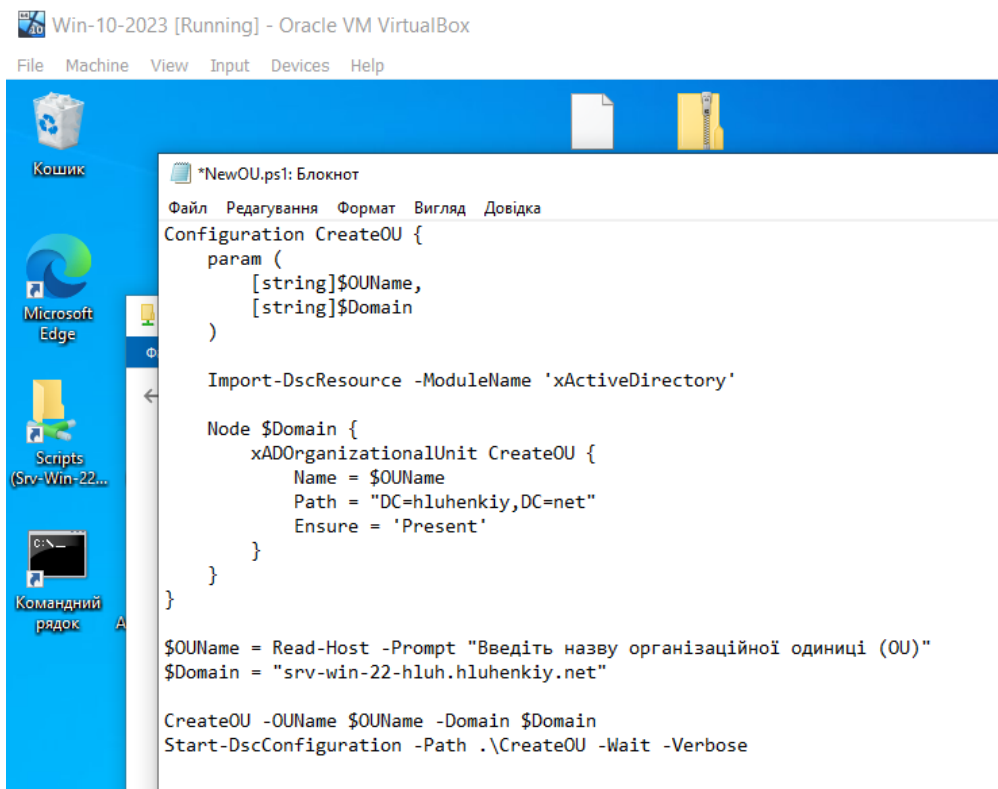


Рис. 41 Створення DSC-конфігурації

Поділимо написаний скрипт на декілька частин, та оглянемо детальніше кожен окрему:

- **Оголошення конфігурації CreateOU:**

Configuration CreateOU: Це блок DSC-конфігурації, що визначає дії, які потрібно виконати.

param: Оголошення вхідних параметрів. \$OUName - назва OU, \$Domain - домен, де створюється ця OU.

Import-DscResource: Підключає модуль xActiveDirectory, який надає ресурси для роботи з AD.

- **Опис дій для вузла (Node)**

Node \$Domain: Вказує, на якому вузлі (сервері) виконується конфігурація.

xADOrganizationalUnit CreateOU: Описує ресурс, який створює OU.

Параметри:

-Name = \$OUName: Назва OU.

-Path = "DC=hluhenkiy,DC=net": Шлях у домені, де створюється OU

-Ensure = 'Present': Гарантує, що OU існує (створюється, якщо її немає).

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

- Запит у користувача даних

Read-Host: Запитує у користувача назву OU.

\$Domain: Статично задає домен сервера, де буде створена OU.

- Виклик конфігурації

CreateOU: Передає значення параметрів до конфігурації CreateOU.

- Застосування DSC конфігурації

Start-DscConfiguration: Застосовує конфігурацію:

-Path .\CreateOU: Вказує шлях до скомпільованих .mof-файлів конфігурації.

-Wait: Очікує завершення виконання.

-Verbose: Виводить детальну інформацію під час виконання.

Останній крок при застосування конфігурації можна пропустити, та виконати потім вручну. Для наочності ще виконаємо дану команду після створення відповідного .mof файлу.

Тепер запусимо виконання нашого скрипта, та впевнимось в його працездатності:

```

PS C:\scripts>
PS C:\scripts>
PS C:\scripts> .\NewOU.ps1
Введіть назву організаційної одиниці (OU): MyNewUnit

Directory: C:\scripts\CreateOU

Mode                LastWriteTime         Length Name
-----
-a----          12/20/2024   3:53 PM             2090 srv-win-22-hluh.hluhenkiy.net.mof
VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, 'methodName' =
SendConfigurationApply,'className' = MSFT_DSCLocalConfigurationManager,'namespaceName' =
root/Microsoft/Windows/DesiredStateConfiguration'.
VERBOSE: An LCM method call arrived from computer SRV-WIN-22-HLUH with user sid
S-1-5-21-3247501395-3078649286-117273642-500.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Set ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Resource ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Test ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: [[xADOrganizationalUnit]CreateOU] Retrieving OU 'MyNewUnit'.
VERBOSE: [SRV-WIN-22-HLUH]: [[xADOrganizationalUnit]CreateOU] OU 'MyNewUnit' does not exist
when it should exist.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Test ] [[xADOrganizationalUnit]CreateOU] in 0.2970 seconds.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Set ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: [[xADOrganizationalUnit]CreateOU] Retrieving OU 'MyNewUnit'.
VERBOSE: [SRV-WIN-22-HLUH]: [[xADOrganizationalUnit]CreateOU] Creating OU 'MyNewUnit'.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ] [[xADOrganizationalUnit]CreateOU] in 0.0470 seconds.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Resource ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ] in 0.7030 seconds.
VERBOSE: Operation 'Invoke CimMethod' complete.
VERBOSE: Time taken for configuration job to complete is 0.879 seconds
  
```

Рис. 42 Виконання DSC-конфігурації

Як видно на фото скрипт виконався без помилок. Після його запуску виконались процеси створення .moф файлу, які були детально описані в консолі жовтим кольором. Також було описано дату та час створення файлу, а також зазначено його назву.

Тепер спробуємо запустити створений раніше .moф файл, та перевіримо чи створить він організаційну одиницю, як і було задано в плані.

```

VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ] in 0.7030 seconds.
VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, 'methodName' = SendConfigurationApply, 'className' = MSFT_DSCLocalConfigurationManager, 'namespaceName' = root/Microsoft/Windows/DesiredStateConfiguration.
VERBOSE: Time taken for configuration job to complete is 0.879 seconds
PS C:\scripts> Start-DscConfiguration -path C:\Scripts\CreateOU -wait -Verbose
VERBOSE: An LCM method call arrived from computer SRV-WIN-22-HLUH with user sid S-1-5-21-3247501395-9078649286-117273642-500.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Set ]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Resource ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Test ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Test ] [[xADOrganizationalUnit]CreateOU] Retrieving OU 'MyNewUnit'.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Start Test ] [[xADOrganizationalUnit]CreateOU] OU 'MyNewUnit' exists and is in the desired state.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Test ] [[xADOrganizationalUnit]CreateOU] in 0.0310 seconds.
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ Skip Set ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Resource ] [[xADOrganizationalUnit]CreateOU]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ]
VERBOSE: [SRV-WIN-22-HLUH]: LCM: [ End Set ] in 0.1100 seconds.
VERBOSE: Operation 'Invoke CimMethod' complete.
VERBOSE: Time taken for configuration job to complete is 0.18 seconds
PS C:\scripts>
  
```

Рис. 43 Виконання .moф-конфігурації

Виконання успішне. Останнім кроком буде перевірка. Зайдемо на контролері домену в Active Directory Users and Computers та перевіримо чи створилась відповідна організаційна одиниця.

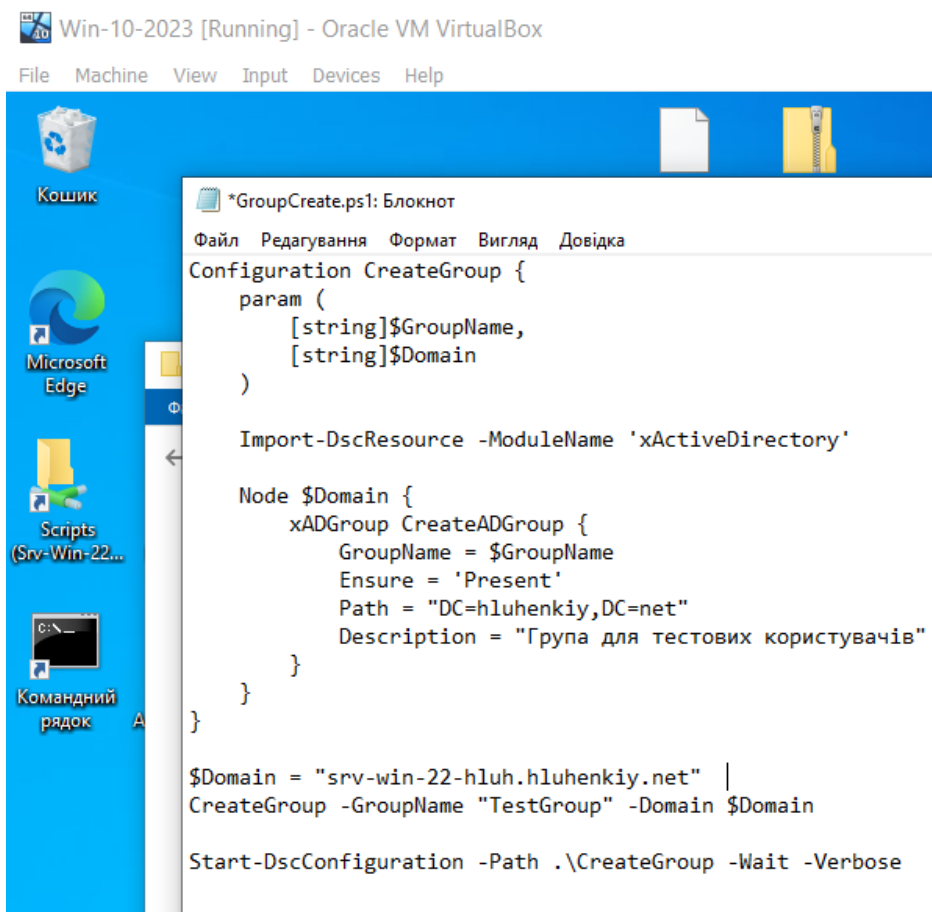


Рис. 45 Створення DSC-конфігурації

Детально розглянемо кожен модуль даного скрипта:

- **Оголошення конфігурації**

Configuration CreateGroup: Створюється конфігурація з назвою CreateGroup.

param: Приймає два параметри:

\$GroupName - ім'я групи, яку потрібно створити.

\$Domain - домен, на якому виконується конфігурація.

- **Імпорт DSC-ресурсу**

Імпортується DSC-модуль xActiveDirectory, який містить ресурси для роботи з Active Directory.

- **Створення групи**

xADGroup: Використовується DSC-ресурс для створення або управління групами в AD.

GroupName = \$GroupName: Назва групи, передана як параметр (TestGroup).

Ensure = 'Present': Гарантує, що група існує. Якщо її немає, вона буде створена.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

Path = "DC=hluhenkiy,DC=net": Вказує шлях у дереві AD, де буде створено групу (корінь домену hluhenkiy.net).

Description = "Група для тестових користувачів": Додає опис до групи.

- **Виклик конфігурації**

\$Domain: Вказується, що конфігурація буде застосована на домені srv-win-22-hluh.hluhenkiy.net.

CreateGroup: Назва групи - TestGroup.

Тепер перейдемо до етапу тестування даного скрипта. Запустимо його локально на Сервері №1. На цей раз в скрипті було вказано конкретну назву для групи користувачів, щоб продемонструвати можливості powershell скриптингу. Якщо у адміністратора, або у будь кого хто відповідальний за використання даного скрипта, з'явиться бажання змінити назву теки це можна запросто зробити змінивши відповідне поле в самому скрипті.

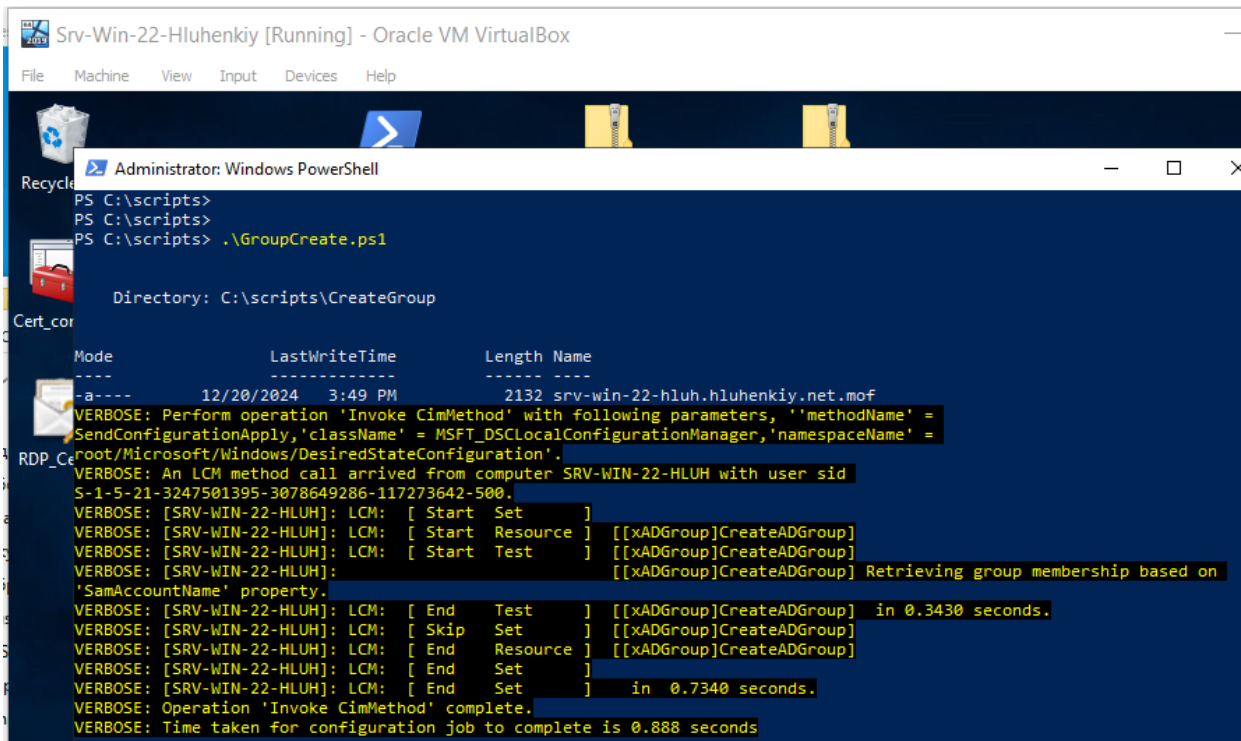


Рис. 46 Виконання DSC-конфігурації

Після успішного створення .mof файлу лишається лише запустити його конфігурацію та перевірити чи всі налаштування будуть виконані правильно.

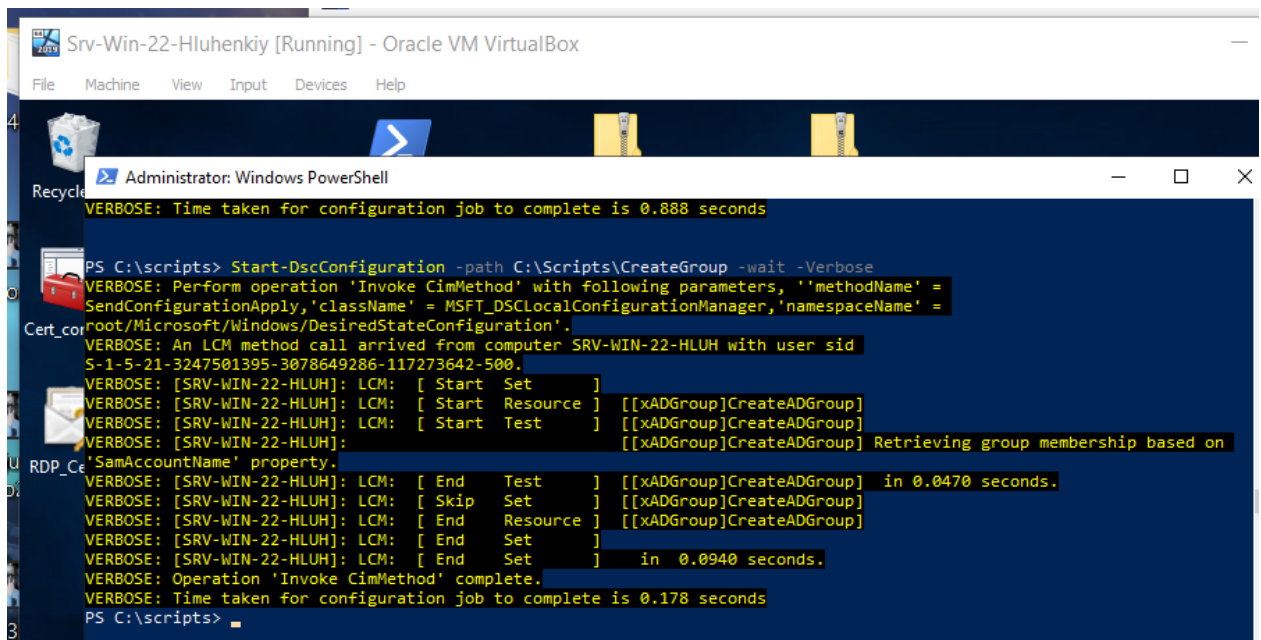


Рис. 47 Виконання .mof-конфігурації

Виконання конфігурації не дало помилок, отже можна зробити висновок, що скрипт спрацював правильно, та групу користувачів було успішно створено. Перейдемо в Active Directory Users and Computers та перевіримо чи створилася нова група.

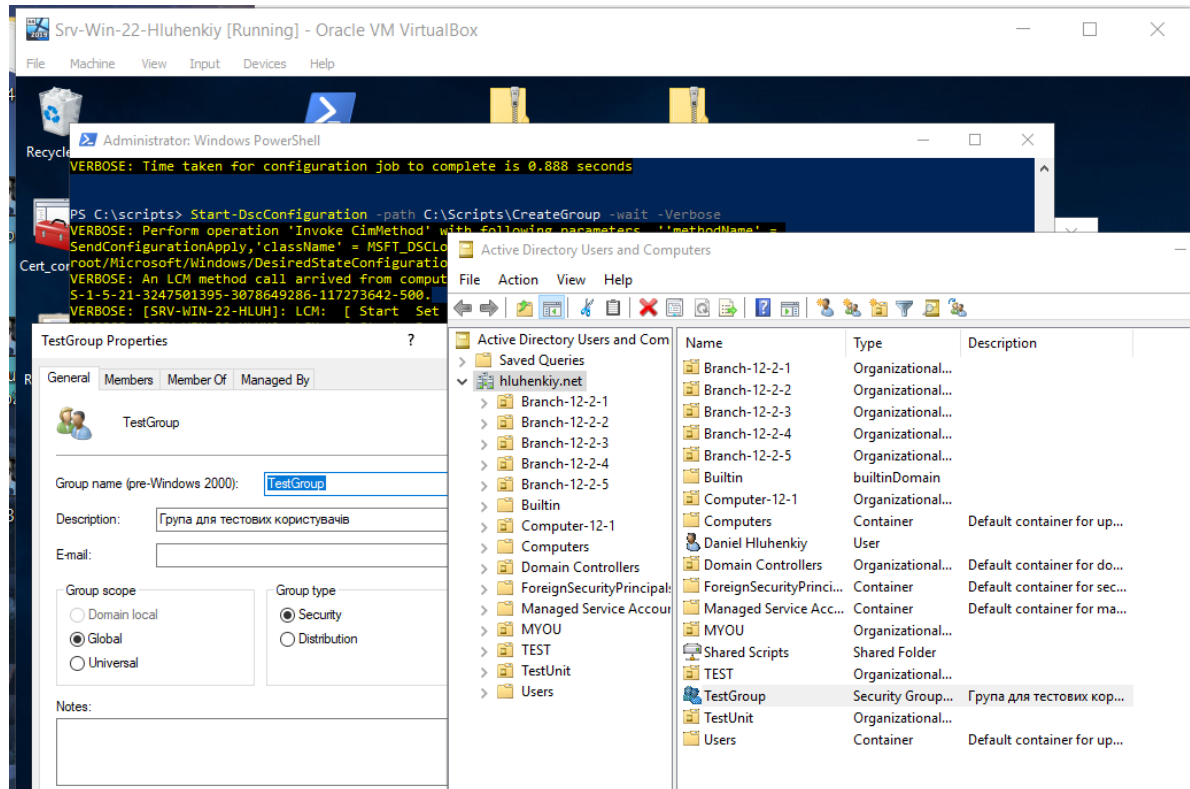


Рис. 48 Перевірка створення групи користувачів

Розроблені скрипти успішно автоматизували процеси управління доступом у домені. Це означає, що можна приступати до наступної групи скриптів для автоматизації.

3.4.2 Автоматизація налаштування мережевих служб

На цьому етапі буде зосереджено увагу на автоматизації налаштування ключових мережевих служб, зокрема служби DHCP, синхронізації часу та перевірки DNS-сервера, які є критичними для стабільної роботи мережі. У цьому підпункті буде описано розробку відповідних скриптів, їхню логіку, а також результати виконання.

У попередньому розділі було проведено успішне перенесення DHCP ролі на окрему робочу станцію Сервер №3. Налаштування передбачали призначення ролі одному серверу та видалення наявної ролі у іншого. В реаліях реальних мереж, де час цінується більше за все, такий процес може витратити забагато часу. Саме тому було розроблено скрипт, який автоматично буде перевіряти стан конкретної віртуальної машини: чи наявна у неї роль dhcp, та якщо ні, призначить її.

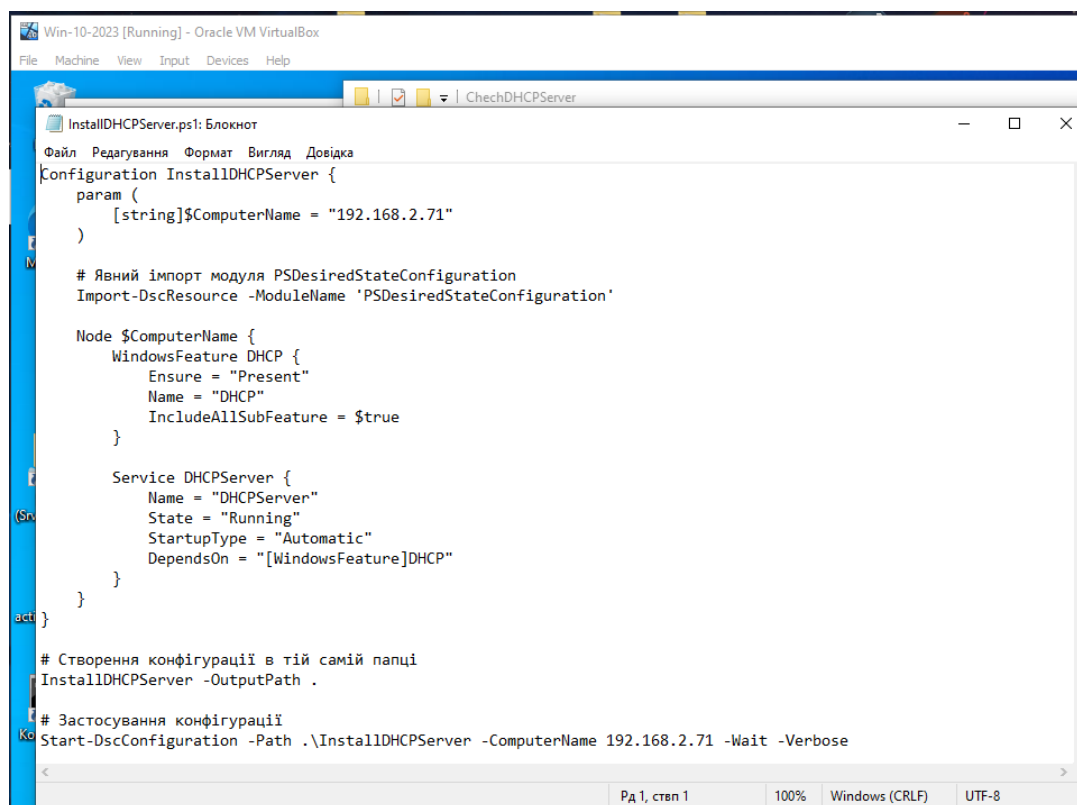


Рис. 49 Скрипт встановлення ролі DHCP сервера

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

Перш за все оглянемо його структуру:

Оголошення конфігурації:

Configuration InstallDHCPServer: Створюється конфігурація з назвою InstallDHCPServer, призначена для автоматизації встановлення та налаштування DHCP-сервера.

param: Приймає два параметри:

\$ComputerName - IP-адреса цільового вузла, на якому застосовується конфігурація (за замовчуванням "192.168.2.71")

Імпорт DSC-ресурсу:

Імпортується модуль PSDesiredStateConfiguration за допомогою Import-DscResource, який надає ресурси для управління функціями та службами Windows, зокрема встановлення ролей і налаштування служб.

Налаштування DHCP-сервера:

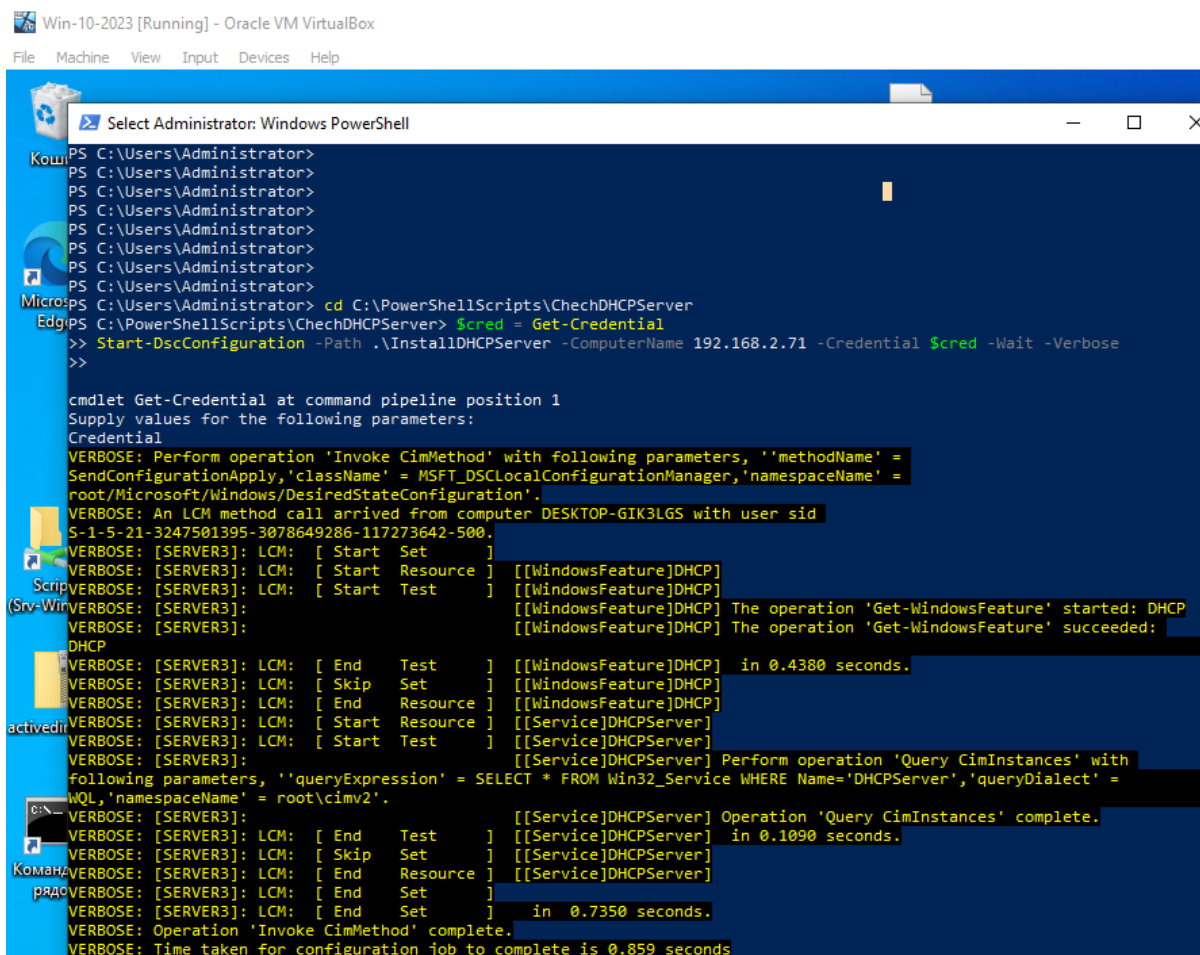
Node \$ComputerName: Визначає цільовий вузол із IP-адресою, переданою через параметр \$ComputerName.

- WindowsFeature DHCP: Використовується DSC-ресурс для встановлення ролі DHCP.
 - **Ensure = "Present":** Гарантує, що роль DHCP встановлена; якщо її немає, вона буде додана.
 - **Name = "DHCP":** Вказує назву ролі для встановлення (DHCP).
 - **IncludeAllSubFeature = \$true:** Включає всі підфункції ролі DHCP для повного встановлення.
- Service DHCPServer: Налаштовує службу DHCP-сервера.
 - **Name = "DHCPServer":** Вказує назву служби.
 - **State = "Running":** Забезпечує, що служба запущена.
 - **StartupType = "Automatic":** Встановлює автоматичний запуск служби під час завантаження системи.
 - **DependsOn = "[WindowsFeature]DHCP":** Вказує залежність від попереднього встановлення ролі DHCP.

Виклик конфігурації:

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		88

Тепер запусимо конфігурацію та перевіримо праивльність виконання налаштувань.



Скрипт перевірів наявність DHCP ролі на вказаній машині, в нашому випадку це Сервері №3, та вивів у консоль деталі виконання. Скрипт перевірів роль у рядку **Start Test “Operation Query CimInstances” succeeded**, вивішив повідомлення **succeeded**, далі він виконав **Skip Set**, тобто пропуск наступного кроку коду, адже DHCP роль на сервері наявна, тому сенсу виконувати блок налаштувань не було. Якщо ж скрипт бачить, що роль не назначена, вона автоматично призначає роль DHCP серверу вказаній машині.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.Г				89
Змн.	Арк.	№ докум.	Підпис	Дата		

Не менш важливим аспектом успішної роботи мережі є синхронізація часу на усіх серверах. Це забезпечує коректну роботу мережевих служб, таких як Active Directory, DHCP і DNS, а також дає змогу виконувати точне логування подій і синхронізацію даних між системами. Відхилення в часі може призвести до збоїв автентифікації, помилок у журналах або проблем із резервним копіюванням. Саме тому було розроблено скрипт, який автоматизує налаштування синхронізації часу з NTP-сервером, гарантуючи узгодженість часових параметрів на всіх серверах інфраструктури.

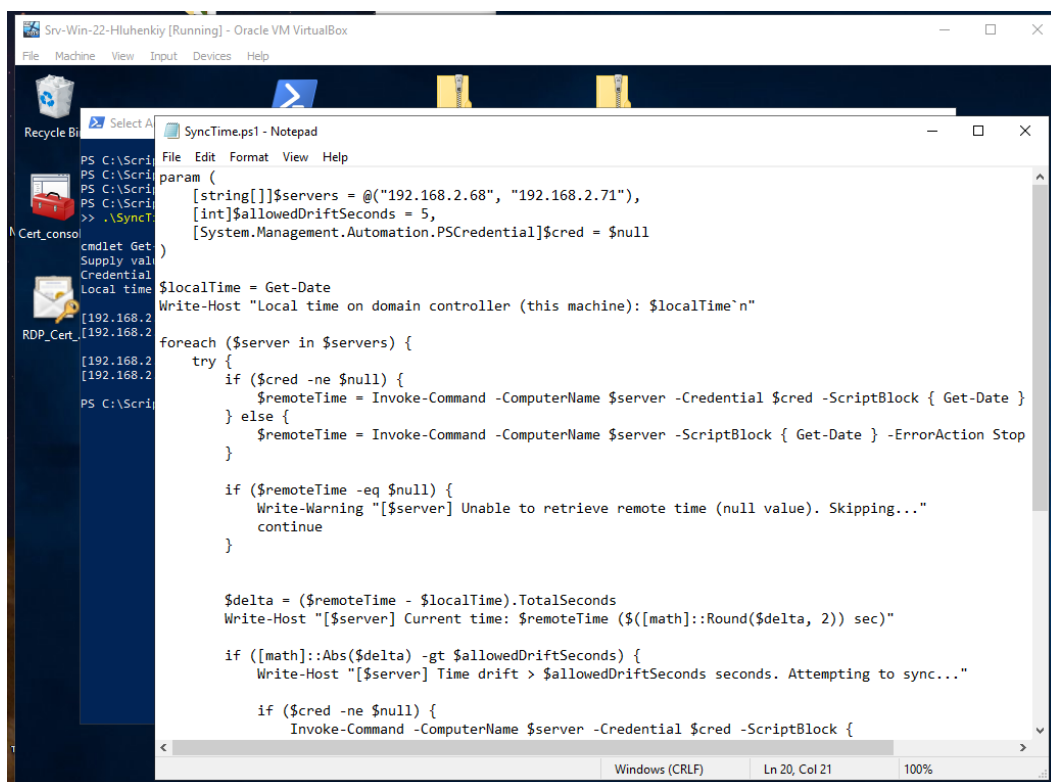


Рис. 51 Скрипт синхронізації часу

Даний скрипт призначений для моніторингу та синхронізації часу на віддалених серверах, визначаючи різницю з локальним часом доменного контролера та виконуючи корекцію при перевищенні допустимого відхилення.

Оглянемо його складові:

Параметри: Приймає масив IP-адрес серверів ($\$servers = "192.168.2.68", "192.168.2.71"$), допустиме відхилення в секундах ($\$allowedDriftSeconds = 5$) та необов'язковий об'єкт облікових даних ($\$cred$) для автентифікації.

Локальний час: Далі використовує Get-Date для отримання поточного часу на доменному контролері з виведенням у консоль.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

Цикл обробки: Наступним кроком перебирає кожен сервер у масиві \$servers, виконуючи віддалений виклик Invoke-Command для отримання часу через Get-Date, із підтримкою автентифікації за допомогою \$cred.

Перевірка та синхронізація: Наостанок обчислює різницю часу (\$delta) і порівнює її з \$allowedDriftSeconds; при перевищенні викликає w32tm /resync /nowait для синхронізації, виводячи відповідні повідомлення (Write-Host, Write-Warning) та обробляючи помилки через блок try-catch.

```

PS C:\Scripts\TimeSync>
PS C:\Scripts\TimeSync>
PS C:\Scripts\TimeSync> $cred = Get-Credential
>> .\SyncTime.ps1 -cred $cred

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
Local time on domain controller (this machine): 05/21/2025 16:08:11

[192.168.2.68] Current time: 05/21/2025 16:08:12 (1.14 sec)
[192.168.2.68] Time is within allowed drift.

[192.168.2.71] Current time: 05/21/2025 16:08:12 (1.48 sec)
[192.168.2.71] Time is within allowed drift.

PS C:\Scripts\TimeSync>
  
```

Рис. 52 Виконання скрипта

Перевіримо якість налаштувань порівнявши вказану дату та час на кожному сервері в мережі:

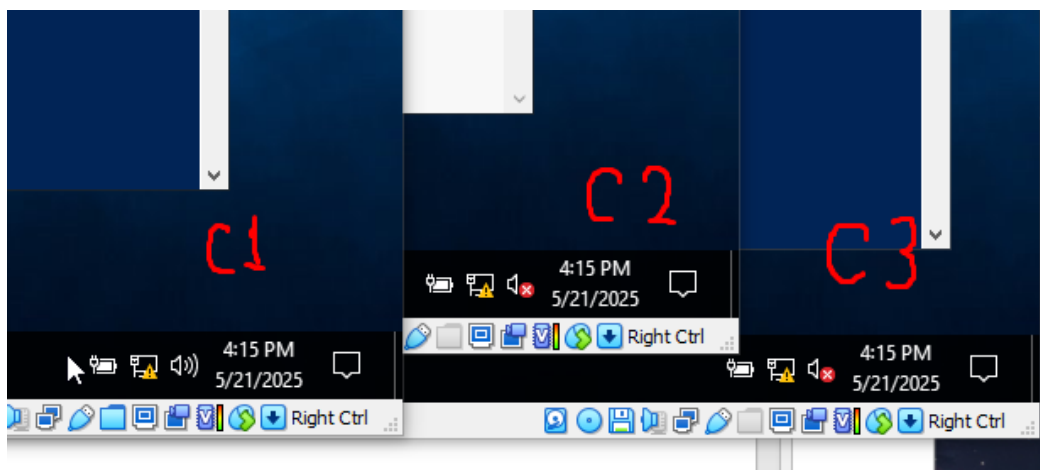


Рис. 53 Перевірка виконання скрипта

Даний скрипт забезпечує автоматизоване підтримання узгодженості часу, критично важливого для стабільної роботи мережеслужб, із гнучкістю у виборі серверів і безпечним доступом.

Наступним буде скрипт налаштування DNS-адреси на робочих станціях у мережі. При невірних налаштуваннях станція може втратити зв'язок з доменом, що є критичною помилкою, особливо при умові, що на станції може бути важливий інструмент керування мережею, наприклад Windows Admin Center. Якщо адміністратор втратить доступ до WAC, то він не зможе відстежувати якість роботи процесів, служб, та сервісів на станціях у мережі. Тому було написано скрипт, який перевіряє DNS-адресу у вказаних робочих станціях та, якщо наявні розбіжності, виправляє їх, забезпечуючи правильність роботи мережі.

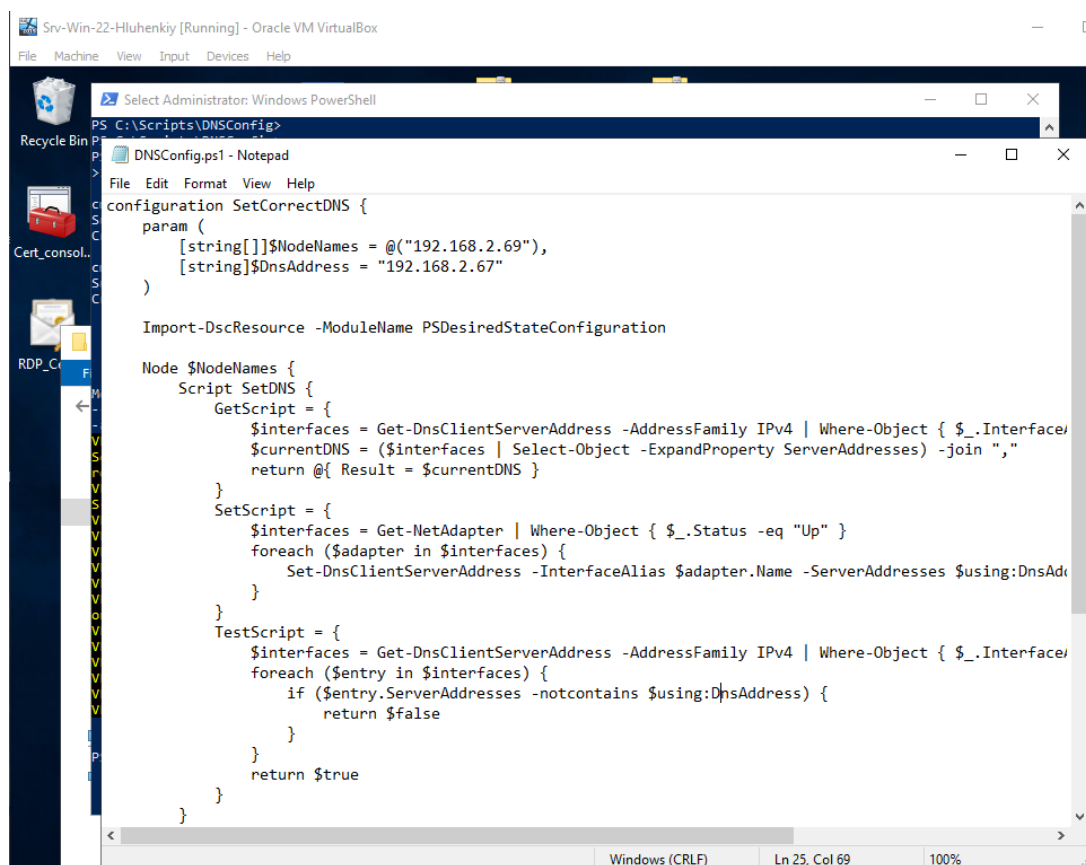


Рис. 54 Скрипт налаштування DNS

Параметри: Приймає масив вузлів (\$NodeNames = "192.168.2.69") та IP-адресу DNS-сервера (\$DnsAddress = "192.168.2.67") для налаштування.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

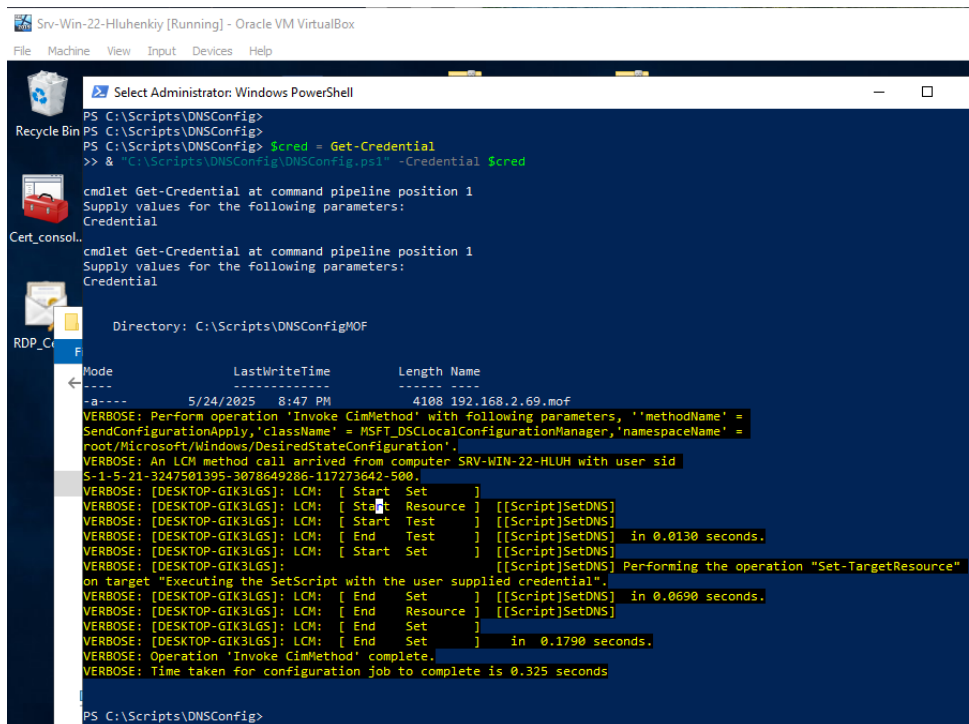
Імпорт ресурсу: Використовує модуль PSDesiredStateConfiguration для доступу до DSC-ресурсів.

Ресурс Script:

- **GetScript:** Отримує поточні DNS-адреси мережевих інтерфейсів через Get-DnsClientServerAddress, повертаючи їх у вигляді рядка.
- **TestScript:** Перевіряє, чи містять усі активні інтерфейси вказану DNS-адресу (\$DnsAddress), повертаючи \$true, якщо налаштування коректне, і \$false у протилежному випадку.
- **SetScript:** Налаштовує DNS-адресу на всіх активних мережевих адаптерах (Get-NetAdapter) за допомогою Set-DnsClientServerAddress.

Застосування конфігурації: Генерує MOF-файл у папці C:\Scripts\DNSConfigMOF і застосовує конфігурацію на кожному вузлі через Start-DscConfiguration, використовуючи автентифікацію (\$cred) із параметрами -Wait, -Force і -Verbose для синхронного виконання та детального виведення.

Запустимо конфігурацію та перевіримо правильність проведених налаштувань.



```
PS C:\Scripts\DNSConfig>
PS C:\Scripts\DNSConfig>
PS C:\Scripts\DNSConfig> $cred = Get-Credential
>> & "C:\Scripts\DNSConfig\DNSConfig.ps1" -Credential $cred

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential

Directory: C:\Scripts\DNSConfigMOF

Mode                LastWriteTime         Length Name
----                -
-a-----          5/24/2025   8:47 PM           4108 192.168.2.69.mof

VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, ''methodName' =
SendConfigurationApply,'className' = MSFT_DSCLocalConfigurationManager,'namespaceName' =
root\Microsoft\Windows\DesiredStateConfiguration'.
VERBOSE: An LCM method call arrived from computer SRV-WIN-22-HLUH with user sid
S-1-5-21-3247501395-3078649286-117273642-500.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set ]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Resource ] [[Script]SetDNS]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[Script]SetDNS]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Test ] [[Script]SetDNS] in 0.0130 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set ] [[Script]SetDNS]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ] [[Script]SetDNS]
on target "Executing the SetScript with the user supplied credential". Performing the operation "Set-TargetResource"
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ] [[Script]SetDNS] in 0.0690 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Resource ] [[Script]SetDNS]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ] in 0.1790 seconds.
VERBOSE: Operation 'Invoke CimMethod' complete.
VERBOSE: Time taken for configuration job to complete is 0.325 seconds
PS C:\Scripts\DNSConfig>
```

Рис. 55 Виконання скрипта

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

Як видно з фото скрипт створив .mof файл після чого одразу запустив його роботу, далі він детально описав всі пройдені кроки налаштувань з вказаним часом витраченим на кожен крок.

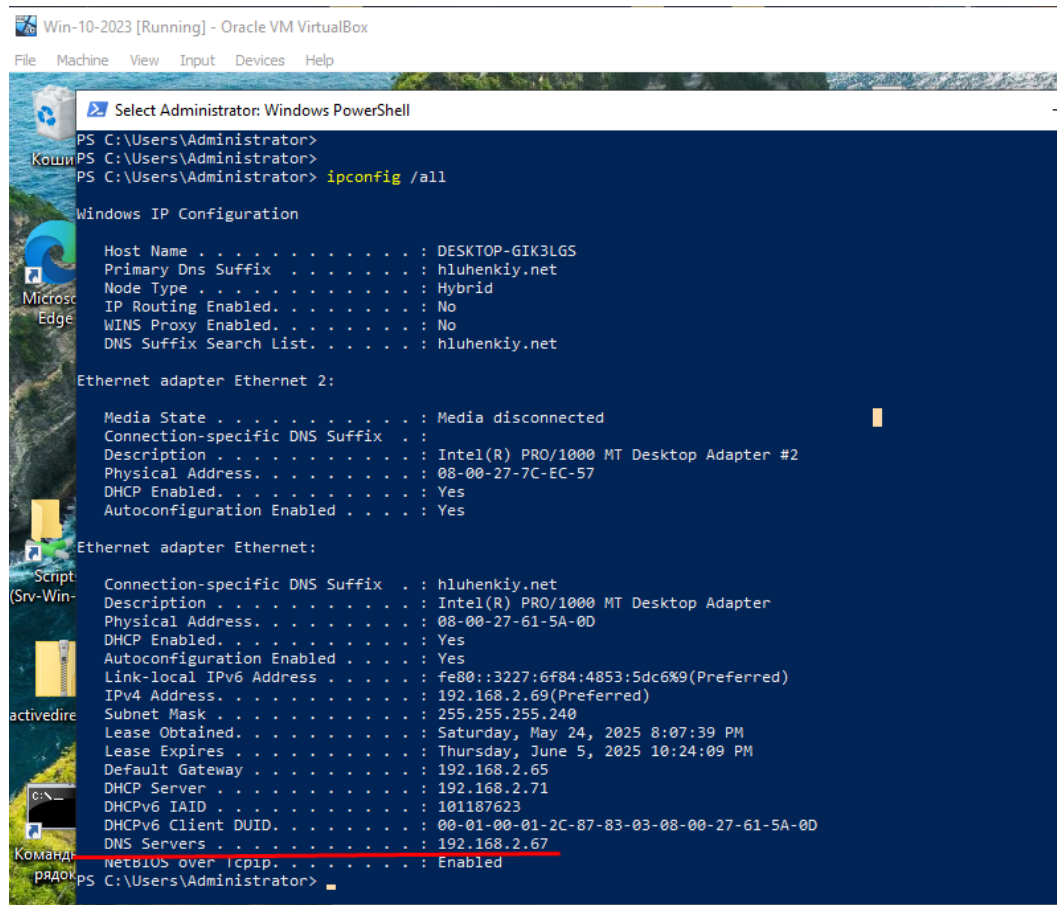


Рис. 56 Перевірка налаштувань DNS

Перевірка показала, що налаштування були виконані правильно. Отже, можна приступати до наступного кроку автоматизації адміністрування.

3.4.3 Автоматизація заходів безпеки

В даному підрозділі буде створено скрипти для автоматизації заходів безпеки, зокрема створення папок із обмеженим доступом і вимкнення USB-портів на робочих станціях, для підвищення захисту даних і запобігання несанкціонованим діям. Також буде розглянуто один із методів боротьби з шкідливими програмними забезпеченнями.

Першим скриптом для автоматизації заходів безпеки було написано скрипт для створення тек з спеціальними правами, виключно на конкретну групу користувачів.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				94
Змн.	Арк.	№ докум.	Підпис	Дата		

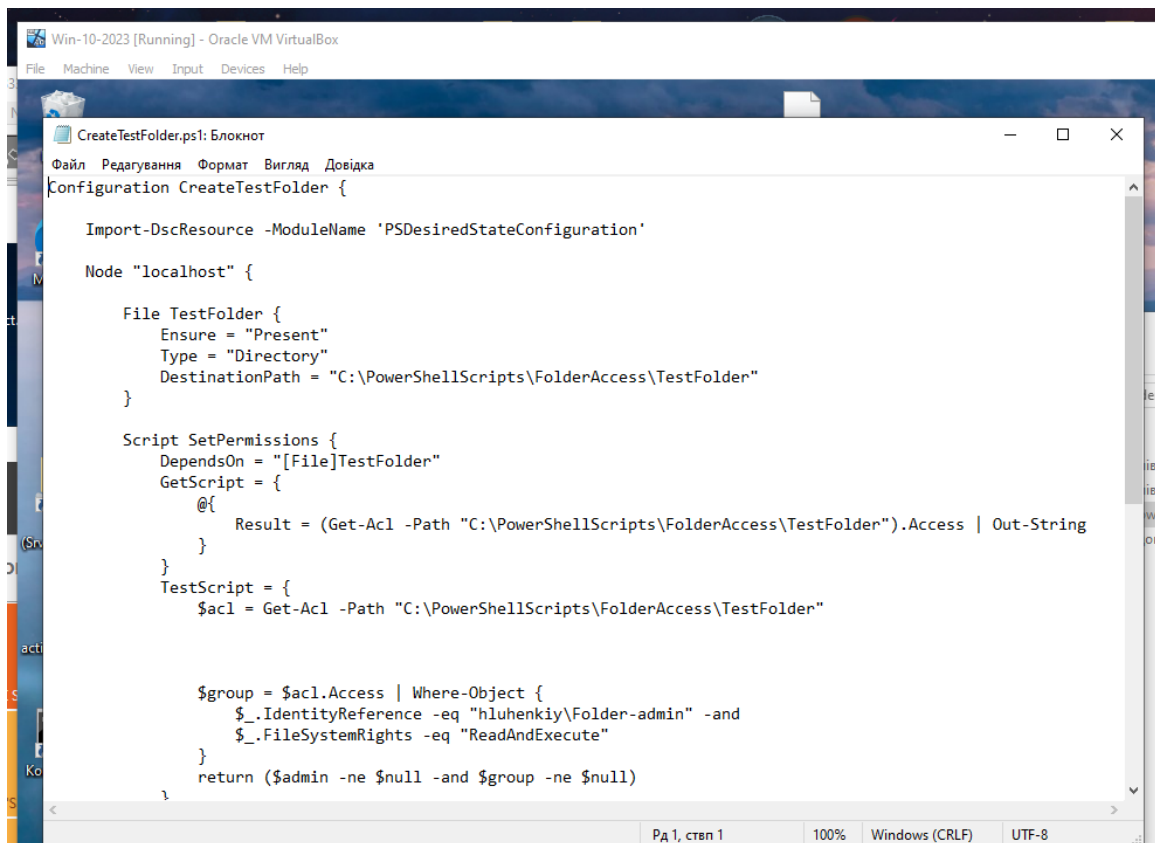


Рис. 57 Створення скрипта

Даний скрипт призначений для створення папки з обмеженими правами доступу на локальному комп'ютері. Одразу оглянемо його ключові елементи роботи:

Імпорт ресурсу: Використовується модуль PSDesiredStateConfiguration для доступу до DSC-ресурсів.

Вузол: Конфігурація застосовується на локальному вузлі (localhost).

Ресурс File:

TestFolder: Створює папку за шляхом C:\PowerShellScripts\FolderAccess\TestFolder із параметром Ensure = "Present", гарантуючи її наявність.

Ресурс Script:

- **SetPermissions:** Налаштовує права доступу до папки, залежачи від ресурсу TestFolder.
- **GetScript:** Повертає поточні права доступу (Get-Acl) для папки у вигляді рядка.
- **TestScript:** Перевіряє, чи має група hluhenkiy\Folder-admin права ReadAndExecute; повертає \$true, якщо права встановлені коректно.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

- **SetScript:** Встановлює нові права доступу, забороняючи наслідування (SetAccessRuleProtection), очищаючи існуючі права, і надаючи групі hluchenkiy\Folder-admin права ReadAndExecute із успадкуванням для підпапок і файлів через Set-Acl.

Тепер варто перевірити правильність роботи даного скрипта. Запустимо його виконання, а після перевіримо через консоль powershell наявні права на створену теку.

```

Win-10-2023 [Running] - Oracle VM VirtualBox
Administrator: Windows PowerShell
PS C:\PowerShellScripts\FolderAccess>
PS C:\PowerShellScripts\FolderAccess>
PS C:\PowerShellScripts\FolderAccess>
PS C:\PowerShellScripts\FolderAccess>
PS C:\PowerShellScripts\FolderAccess> .\CreateTestFolder

Directory: C:\PowerShellScripts\FolderAccess\CreateTestFolder

Mode                LastWriteTime         Length Name
----                -
-a-----          5/20/2025   6:09 PM           6802 localhost.mof

PS C:\PowerShellScripts\FolderAccess> Start-DscConfiguration -Path .\CreateTestFolder -Wait -Verbose -Force
V>>
VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, 'methodName' =
SendConfigurationApply, 'className' = MSFT_DSCLocalConfigurationManager, 'namespaceName' =
root\Microsoft\Windows\DesiredStateConfiguration'.
VERBOSE: An LCM method call arrived from computer DESKTOP-GIK3LGS with user sid
S-1-5-21-3247501395-3078649286-117273642-500.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set ]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Resource ] [[File]TestFolder]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[File]TestFolder]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[File]TestFolder] The system cannot find the file specified.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[File]TestFolder] The related file/directory is:
C:\PowerShellScripts\FolderAccess\TestFolder.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Test ] [[File]TestFolder] in 0.0150 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set ] [[File]TestFolder]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[File]TestFolder] The system cannot find the file specified.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[File]TestFolder] The related file/directory is:
C:\PowerShellScripts\FolderAccess\TestFolder.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ] [[File]TestFolder] in 0.0160 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Resource ] [[File]TestFolder]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Resource ] [[Script]SetPermissions]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test ] [[Script]SetPermissions]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Test ] [[Script]SetPermissions] in 0.3910 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set ] [[Script]SetPermissions]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set ] [[Script]SetPermissions] Performing the operation
"Set-TargetResource" on target "Executing the SetScript with the user supplied credential".
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ] [[Script]SetPermissions] in 0.1090 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Resource ] [[Script]SetPermissions]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set ] in 2.5010 seconds.
VERBOSE: Operation 'Invoke CimMethod' complete.
VERBOSE: Time taken for configuration job to complete is 2.854 seconds
  
```

Рис. 58 Виконання сткрипта для створення теки з правами

```

Win-10-2023 [Running] - Oracle VM VirtualBox
Administrator: Windows PowerShell
PS C:\PowerShellScripts\FolderAccess> (Get-Acl -Path "C:\PowerShellScripts\FolderAccess\TestFolder").Access
>>
FileSystemRights : ReadAndExecute, Synchronize
AccessControlType : Allow
IdentityReference : HLUHENKIY\Folder-admin
IsInherited       : False
InheritanceFlags  : ContainerInherit, ObjectInherit
PropagationFlags  : None

PS C:\PowerShellScripts\FolderAccess>
  
```

Рис. 59 Перевірка наявних прав

Як видно на фото команда показує, що доступ до теки є лише у однієї категорії користувачів: групи Folder-Admin. Тепер підключимось до робочої станції за допомогою облікових даних користувача із групи Folder-Admin та перевіримо доступ до відповідної папки. Спочатку спробуємо відкрити теку з правами звичайного користувача.

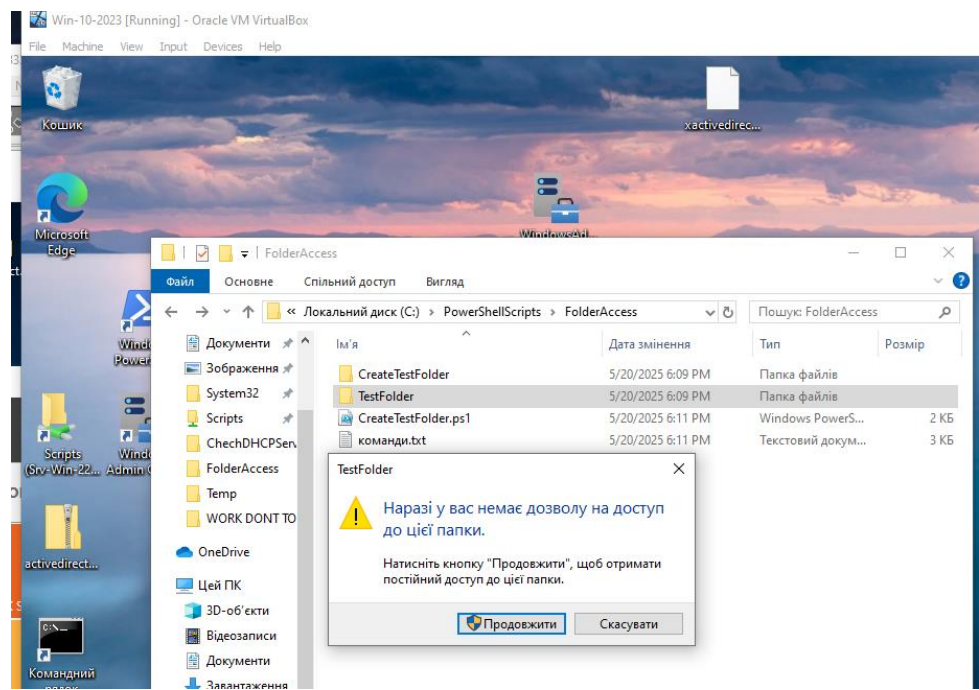


Рис. 60 Спроба підключитись до теки

Після неуспішної спроби відкрити папку підключимось за обліковими даними потрібного нам користувача та спробуємо іще раз.

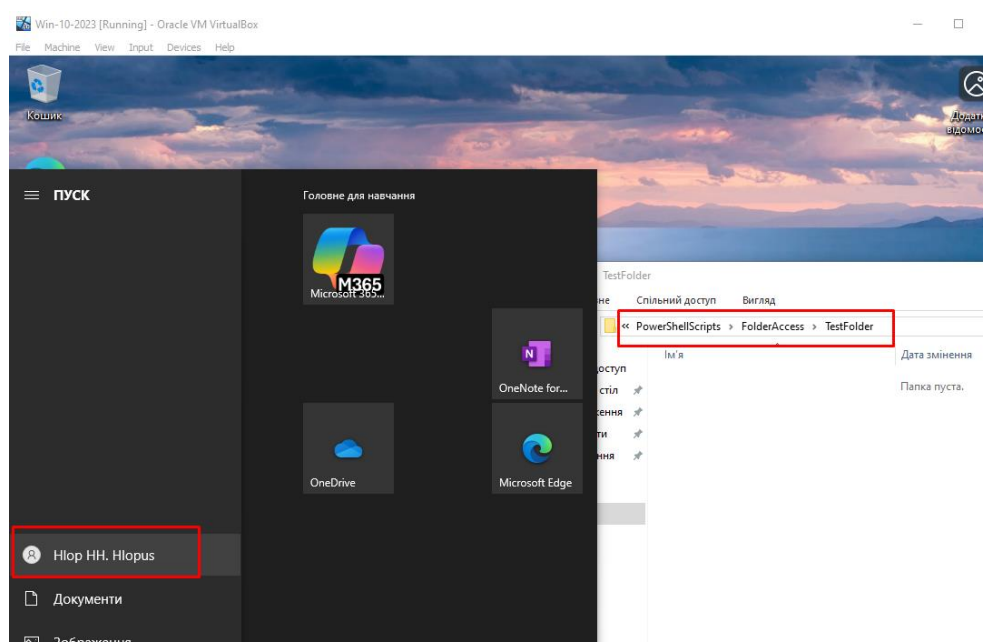


Рис. 61 Спроба підключитись до теки

Як бачимо, права були налаштовані правильно. Даний скрипт забезпечує автоматизоване створення захищеної папки з чітко визначеними правами, що є важливим для обмеження несанкціонованого доступу до мережі.

Наступним кроком оглянемо скрипт який забороняє використовувати usb-порти на робочих станціях. Даний скрипт може використовуватись для будь якої кількості машин, що значно полегшує роботу з доменом. Загальна ідея складає запобігання несанкціонованому підключенню до пристроїв.

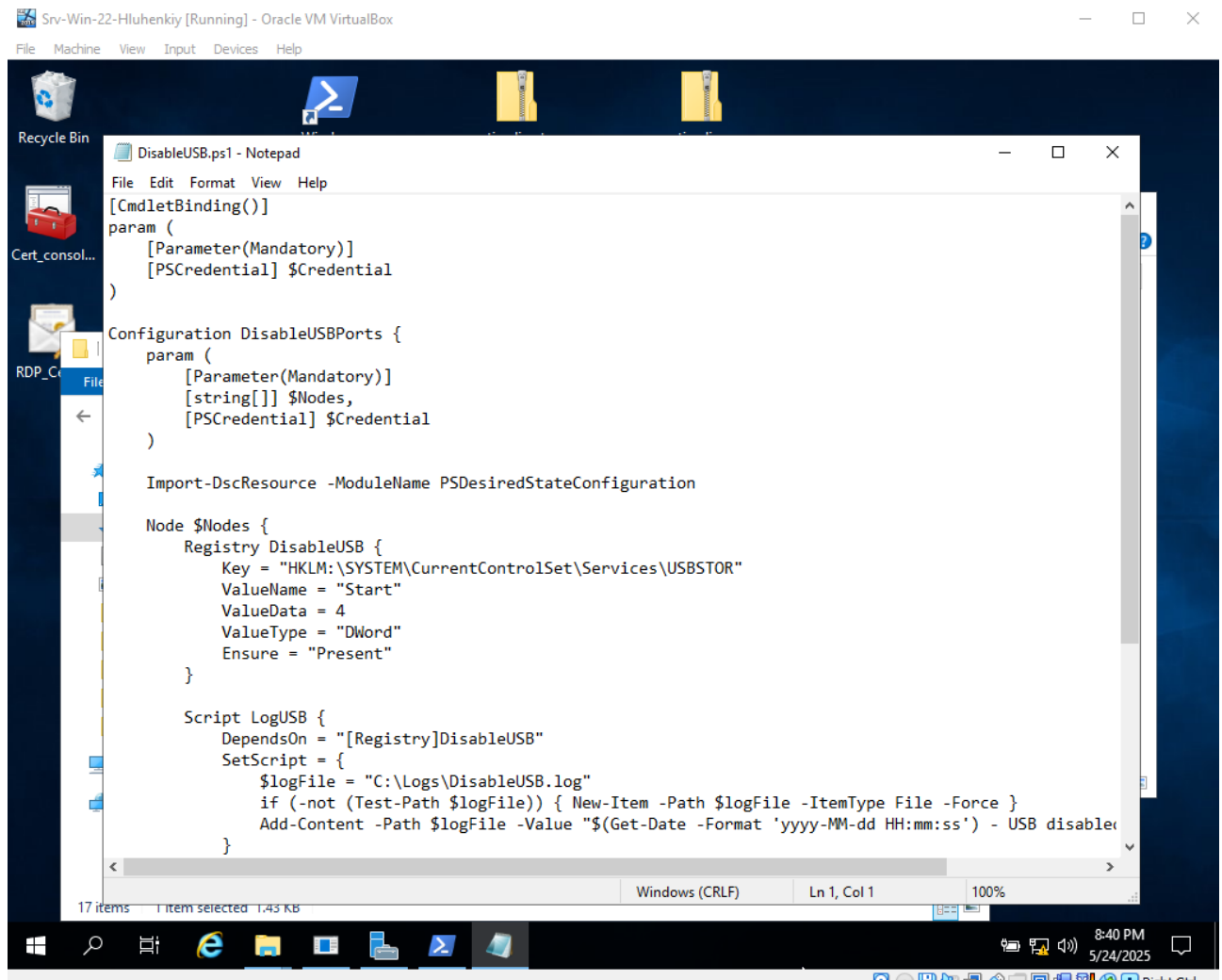


Рис. 62 Створений скрипт

Оглянемо основні елементи скрипту:

Параметри: Використовує CmdletBinding() для підтримки розширених функцій. Приймає обов'язковий об'єкт \$Credential для автентифікації на глобальному рівні та в конфігурації DisableUSBPorts, яка також приймає обов'язковий масив вузлів (\$Nodes) та необов'язковий \$Credential.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

Імпорт ресурсу: Імпортує модуль PSDesiredStateConfiguration для доступу до DSC-ресурсів.

Ресурс Registry:

DisableUSB: Налаштовує реєстр, змінюючи значення Start на 4 у ключі HKLM:\SYSTEM\CurrentControlSet\Services\USBSTOR, що вимикає службу USBSTOR, із параметром Ensure = "Present".

Ресурс Script:

LogUSB: Логує дію, залежну від Registry, створюючи файл C:\Logs\DisableUSB.log із датою та ім'ям вузла (SetScript), повертає статус логування (GetScript), а TestScript завжди повертає \$true для примусового виконання.

Виклик конфігурації: Створює директорію C:\Scripts\DisableUSBMOF, генерує MOF-файл із параметрами \$targetMachines і \$Credential, а потім застосовує конфігурацію через Start-DscConfiguration із параметрами -Wait, -Verbose і -Force.

Такий порядок дій забезпечує послідовність та покроковість виконання завдання. Спочатку скрипт аналізує наявні параметри конкретної робочої станції, а потім, якщо наявні розбіжності, виправляє стан usb порту на заданий.

Запустимо виконання скрипта та перевіримо правильність проведених налаштувань.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				99
Змн.	Арк.	№ докум.	Підпис	Дата		

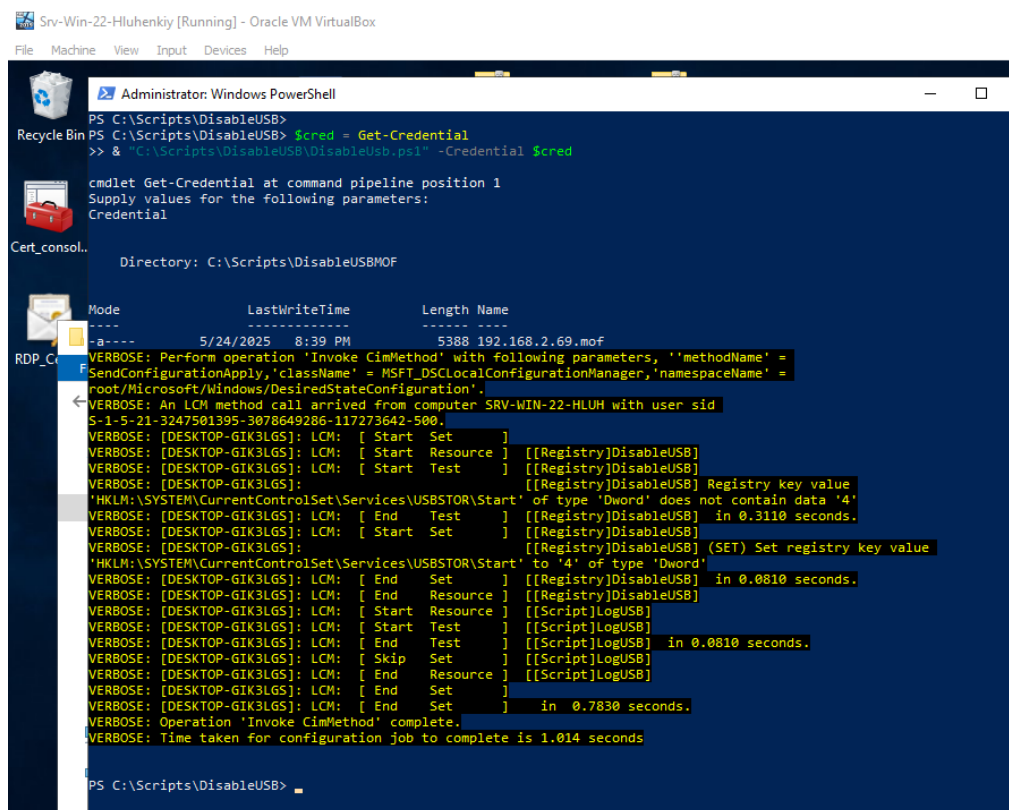


Рис. 63 Успішне виконання скрипта

Для перевірки проведених налаштувань необхідно переглянути значення ключа `HKLM:\SYSTEM\CurrentControlSet\Services\USBSTOR`. Статус «3» означає, що служба активна, та usb-порт готовий до використання. Якщо ж служба має статус «4», то скрипт спрацював правильно та вимкнув службу, блокуючи відповідні usb-порти.

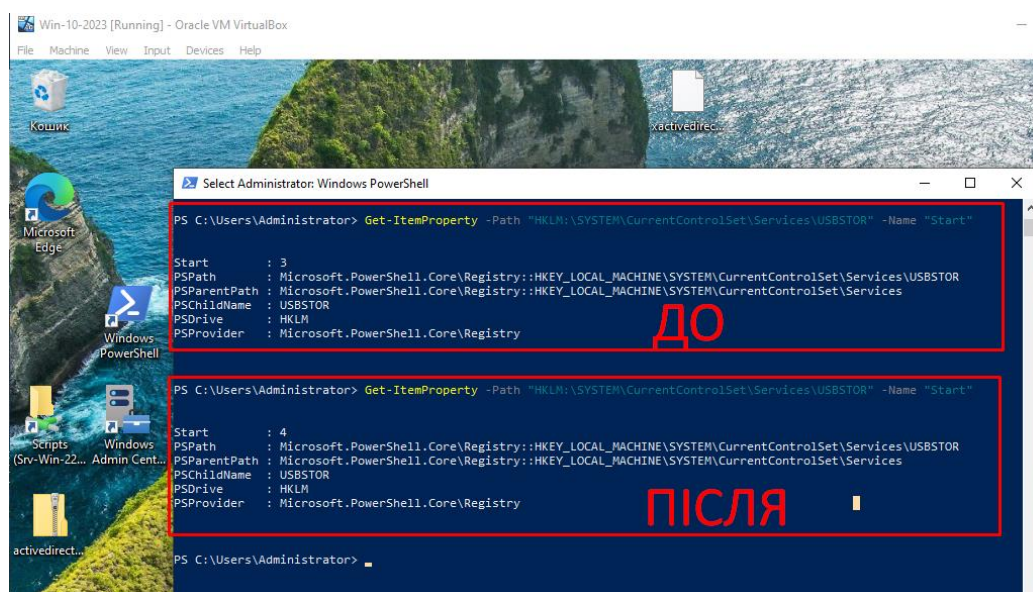


Рис. 64 Успішне виконання скрипта

		Глухенький Д.О				ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І					100
Змн.	Арк.	№ докум.	Підпис	Дата			

Успішне виконання скрипта дає змогу збільшити рівень безпеки мережі та перейти до виконання наступного кроку налаштування мережі.

Автоматизація адміністрування допомагає адміністратору краще налаштовувати ті чи інші аспекти мережі. Проте варто також звернути увагу на аномальний сценарій протидії вірусному програмному забезпеченню. Якщо вірусне ПО потрапить в домен, воно може отримати доступ до ключових сервісів мережі, таких як Active Directory. Це може нанести значні збитки компанії в мережу якої потрапив вірус. Саме тому було розроблено метод протидії таким ПО, а саме з сумнозвісним вірусом Petya. Для цього необхідно створити пастку, яка буде працювати «маркером успіху» для вірусного ПО, що дозволить зберегти важливі дані домену.

"Пастка" для вірусу Petya - це захисний механізм, запропонований аналітиками безпеки для запобігання виконанню шкідливого коду вірусів типу Petya/NotPetya. Вона полягає у створенні файлу з назвою perfc або perfc.dat у каталозі C:\Windows із атрибутом "лише для читання". Цей підхід використовує особливості поведінки вірусу, який перевіряє наявність таких файлів як маркер зараження, і припиняє свою дію, якщо вони вже існують і захищені від модифікації.

Віруси типу Petya/NotPetya під час запуску перевіряють наявність файлу perfc.dat у каталозі C:\Windows. Якщо файл відсутній, вірус розпочинає шифрування Master File Table (MFT) і відображає фальшивий екран відновлення, блокуючи доступ до системи. У разі наявності файлу з атрибутом "лише для читання", вірус інтерпретує це як ознаку попереднього зараження й зупиняє виконання, уникаючи подальшого шифрування. Таким чином, створення такого файлу дозволяє обдурити вірус і зупинити його активність на ранньому етапі.

Головна мета такої "пастки" - це знизити ризик успішного виконання Petya/NotPetya, захищаючи систему від шифрування даних і втрати доступу. Це особливо актуально для інфраструктур, де оновлення антивірусного програмного забезпечення чи патчів безпеки може бути затриманим, а також як додатковий шар захисту в умовах високого ризику кібератак.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				101
Змн.	Арк.	№ докум.	Підпис	Дата		

Для реалізації "пастки" можна модифікувати існуючий PowerShell DSC-скрипт, наприклад DisableUSB, додавши ресурс File для створення файлу perfc.dat. Нижче наведено приклад коду, який інтегрується в конфігурацію:

```
File PetyaTrap {  

  Ensure = "Present"  

  Type = "File"  

  DestinationPath = "C:\Windows\perfc.dat"  

  Attributes = "ReadOnly"  

  Contents = ""  

}
```

Цей код створює порожній файл perfc.dat у каталозі C:\Windows із атрибутом "лише для читання". Для виконання скрипту необхідні права адміністратора, оскільки системна папка C:\Windows захищена. Після виконання можна перевірити наявність файлу та його атрибути через провідник або командний рядок. Така практика є простим і ефективним методом захисту від відомих версій цього вірусу, дозволяючи зупинити його виконання без складних змін у системі. Проте її ефективність обмежена сучасними загрозами, які можуть ігнорувати такі маркери. Цей механізм доцільно використовувати як додатковий захисний захід у поєднанні з антивірусним ПЗ, регулярними оновленнями та іншими стратегіями безпеки.

3.4.4 Автоматизація моніторингу та оптимізації роботи системи

У попередніх підпунктах було розглянуто автоматизацію управління користувачами, налаштування мережевих служб і заходів безпеки через PowerShell Desired State Configuration (DSC), що створило надійну основу для функціонування інфраструктури. Даний підрозділ передбачає огляд двох powershellDSC скриптів: перевірка дискового простору та очищення тимчасових файлів. Скрипт з перевіркою дискового простору вже було розглянуто раніше у пункті [3.3.1](#), тому далі буде розглянуто лише другий скрипт.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				102
Змн.	Арк.	№ докум.	Підпис	Дата		

Скрипт для очищення тимчасових файлів є корисним, оскільки дозволяє автоматично видаляти накопичені невикористовувані дані, такі як кеш програм і тимчасові установчі файли, що зменшує завантаження дискового простору. Це сприяє підвищенню продуктивності робочих станцій і серверів, запобігає потенційним конфліктам через застарілі файли та оптимізує загальну стабільність системи. Автоматизація цього процесу через PowerShell скорочує час адміністрування та мінімізує ризик людської помилки.

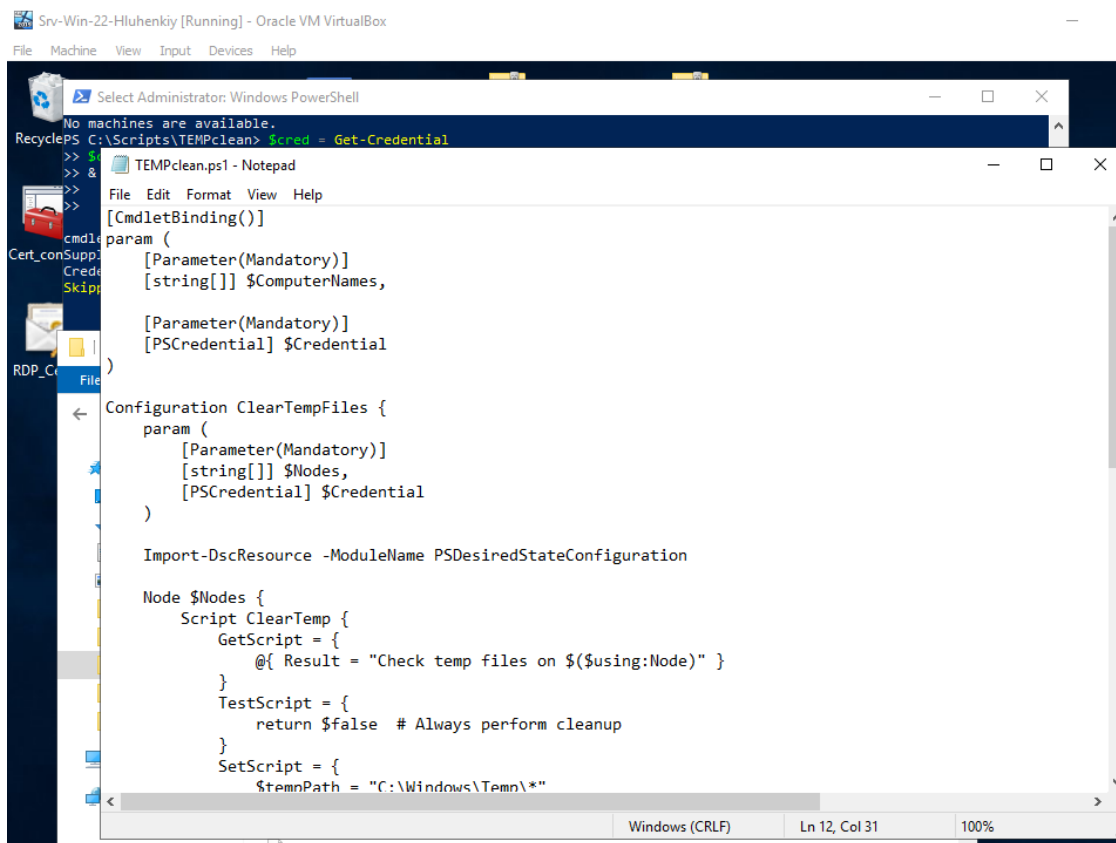


Рис. 65 Написання скрипта для очищення кешу

Оглянемо основні аспекти даного скрипта:

Параметри: Використовує CmdletBinding() для розширених функцій. Приймає обов'язковий масив імен комп'ютерів (\$ComputerNames) та об'єкт \$Credential для автентифікації на глобальному рівні, а конфігурація ClearTempFiles - масив вузлів (\$Nodes) та \$Credential.

Імпорт ресурсу: Імпортує модуль PSDesiredStateConfiguration для доступу до DSC-ресурсів.

Ресурс Script:

- **ClearTemp:** Виконує очищення тимчасових файлів.

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				103
Змн.	Арк.	№ докум.	Підпис	Дата		

- **GetScript:** Повертає статус перевірки тимчасових файлів на вузлі.
- **TestScript:** Повертає \$false, щоб завжди виконувати очищення.
- **SetScript:** Видаляє файли з C:\Windows\Temp* через Remove-Item, записуючи результат або помилки в лог-файл C:\Logs\ClearTemp.log.

Виклик конфігурації: Перевіряє доступність комп'ютерів із \$ComputerNames через Test-Connection, створює директорію C:\Scripts\TEMPcleanMOF, генерує MOF-файл для онлайн-вузлів і застосовує конфігурацію через Start-DscConfiguration із параметрами -Wait, -Verbose і -Force.

Виконаємо даний скрипт та перевіримо чи змінився розмір теки.

```

PS C:\Scripts\TEMPclean> $cred = Get-Credential
Recycle> $computers = @("192.168.2.69", "192.168.2.70")
>> & "C:\Scripts\TEMPclean\TEMPclean.ps1" -ComputerNames $computers -Credential $cred
>>
cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Cert_conCredential
Skipping 192.168.2.70 (offline or unreachable)

Directory: C:\Scripts\TEMPcleanMOF

RDP_C Mode          LastWriteTime          Length Name
-----
-a----- 5/24/2025  9:12 PM          5290 192.168.2.69.mof
VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, ''methodName' =
SendConfigurationApply,'className' = MSFT_DSCLocalConfigurationManager,'namespaceName' =
root/Microsoft/Windows/DesiredStateConfiguration'.
VERBOSE: An LCM method call arrived from computer SRV-WIN-22-HLUH with user sid
S-1-5-21-3247501395-3078649286-117273642-500.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set          ]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Resource ] [[Script]ClearTemp]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Test      ] [[Script]ClearTemp]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Test        ] [[Script]ClearTemp] in 0.0940 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set          ] [[Script]ClearTemp]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ Start Set          ] [[Script]ClearTemp] Performing the operation
"Set-TargetResource" on target "Executing the SetScript with the user supplied credential".
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set          ] [[Script]ClearTemp] in 0.0940 seconds.
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Resource ] [[Script]ClearTemp]
VERBOSE: [DESKTOP-GIK3LGS]: LCM: [ End Set          ]
VERBOSE: Operation 'Invoke CimMethod' complete.
VERBOSE: Time taken for configuration job to complete is 0.884 seconds

PS C:\Scripts\TEMPclean>

```

Рис. 66 Виконання скрипта для очищення теки

Як видно на фото вище, скрипт виконувався для двох робочих станцій: 192.168.2.69 та 192.168.2.70. Під час перевірки скрипт помітив, що одна із станцій не відповідає, адже на момент виконання скрипта вона була вимкнена. Через це скрипт написав в консоль повідомлення Skipping 192.168.2.70 (offline or

unreachable). Такий підхід дає змогу адміністратору викликати скрипт для будь якої кількості станцій, і навіть якщо якась із станцій не дасть відповідь, то скрипт продовжить свою роботу для активних машин.

Перевіримо розмір теки ДО та ПІСЛЯ очищення.

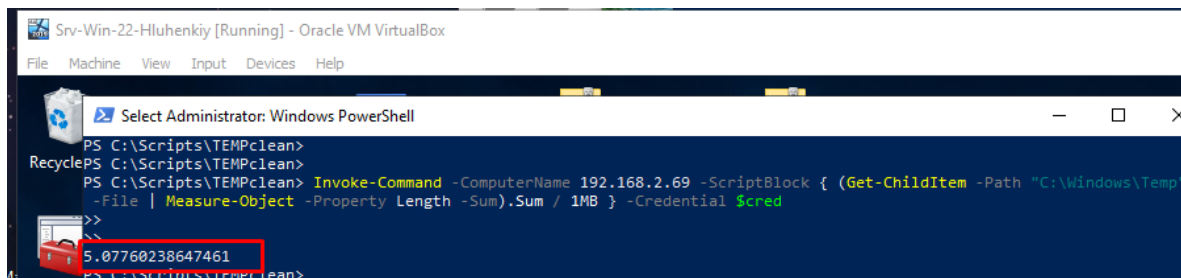


Рис. 67 Перевірка ДО виклику скрипта

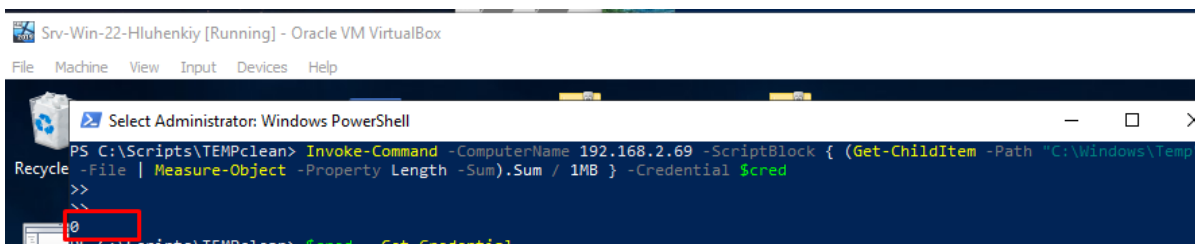


Рис. 68 Перевірка ПІСЛЯ виклику скрипта

Даний скрипт працює так, як було вказано в завданні. Такий скрипт забезпечує ефективне очищення тимчасових файлів із логуванням, оптимізуючи роботу системи на віддалених вузлах.

Завершення роботи з даним скриптом дозволяє нам перейти до заключного етапу виконання проєкту, а саме необхідно протестувати роботу створеної нами мережі, перевірити зв'язок між пристроями та відсутність будь яких помилок та проблем.

3.5 Тестування роботи побудованої мережі

Після виконання всіх налаштувань необхідно перевірити роботу створеної нами схеми. Для цього скористаємося низкою спеціальних моніторингових команд. Перевірку зробимо у такому ж порядку, у якому проводилось налаштування пристроїв. Почнемо з ключової робочої станції, на якій було встановлено Windows Admin Center:

```
Administrator: Windows PowerShell
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator> nslookup hluhenkiy.net
Server: Srv-Win-22-Hluh.hluhenkiy.net
Address: 192.168.2.67

Name: hluhenkiy.net
Addresses: 192.168.2.67
192.168.2.68

PS C:\Users\Administrator> ping 192.168.2.65

Pinging 192.168.2.65 with 32 bytes of data:
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.2.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator> ping 192.168.2.67

Pinging 192.168.2.67 with 32 bytes of data:
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator> ping 192.168.2.68

Pinging 192.168.2.68 with 32 bytes of data:
Reply from 192.168.2.68: bytes=32 time<1ms TTL=128
Reply from 192.168.2.68: bytes=32 time<1ms TTL=128
Reply from 192.168.2.68: bytes=32 time<1ms TTL=128
Reply from 192.168.2.68: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator>
```

Рис. 69 Перевірка зв'язку

Також перевіримо конфігурацію пристрою за допомогою команди ipconfig/all. Одразу виконаємо аналогічні дії на тестовій робочій станції.

```
Win-10-2023 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Select Administrator: Windows PowerShell
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator> ipconfig /all

Windows IP Configuration

Host Name . . . . . : DESKTOP-GIK3LGS
Primary Dns Suffix . . . . . : hluhenkiy.net
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : hluhenkiy.net

Ethernet adapter Ethernet 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
Physical Address. . . . . : 08-00-27-7C-EC-57
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : hluhenkiy.net
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-61-5A-0D
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::3227:6f84:4853:5dc6%9(Preferred)
IPv4 Address. . . . . : 192.168.2.69(Preferred)
Subnet Mask . . . . . : 255.255.255.240
Lease Obtained. . . . . : Saturday, May 24, 2025 8:07:39 PM
Lease Expires . . . . . : Thursday, June 5, 2025 10:24:09 PM
Default Gateway . . . . . : 192.168.2.65
DHCP Server . . . . . : 192.168.2.71
DHCPv6 IAID . . . . . : 101187623
DHCPv6 Client DUID. . . . . : 00-01-00-01-2C-87-03-08-00-27-61-5A-0D
DNS Servers . . . . . : 192.168.2.67
```

Рис. 70 Перевірка конфігурації

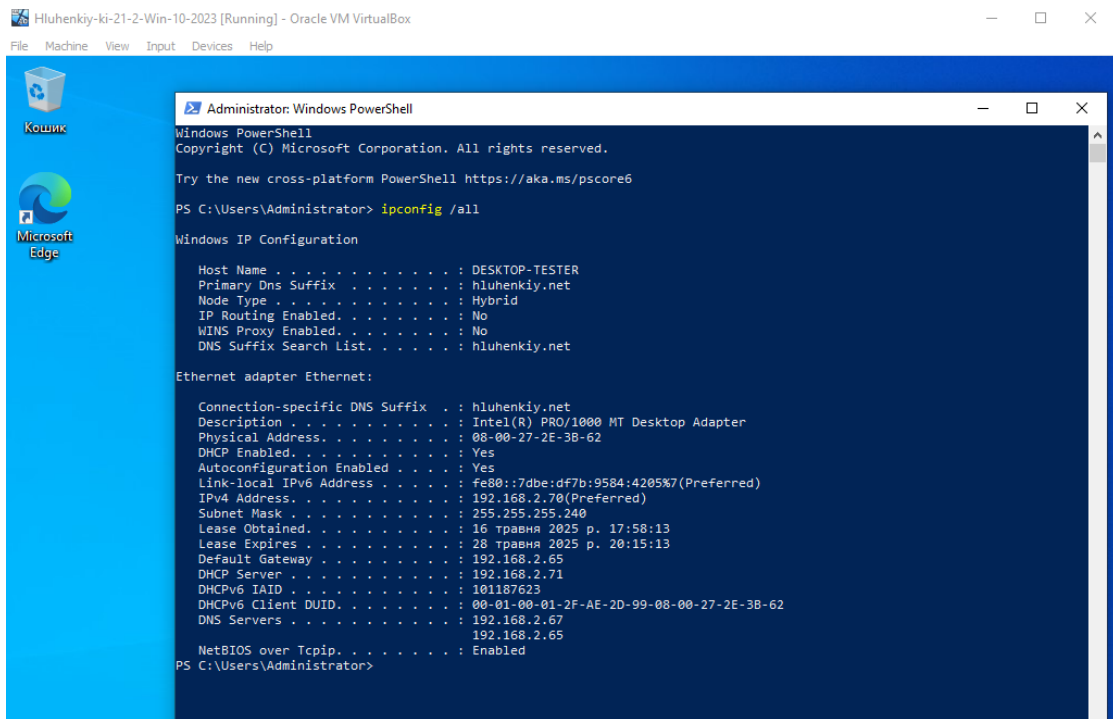


Рис. 71 Перевірка конфігурації

Перевіримо наявність під'єднаних віртуальних станцій в інструменті Windows Admin Center.

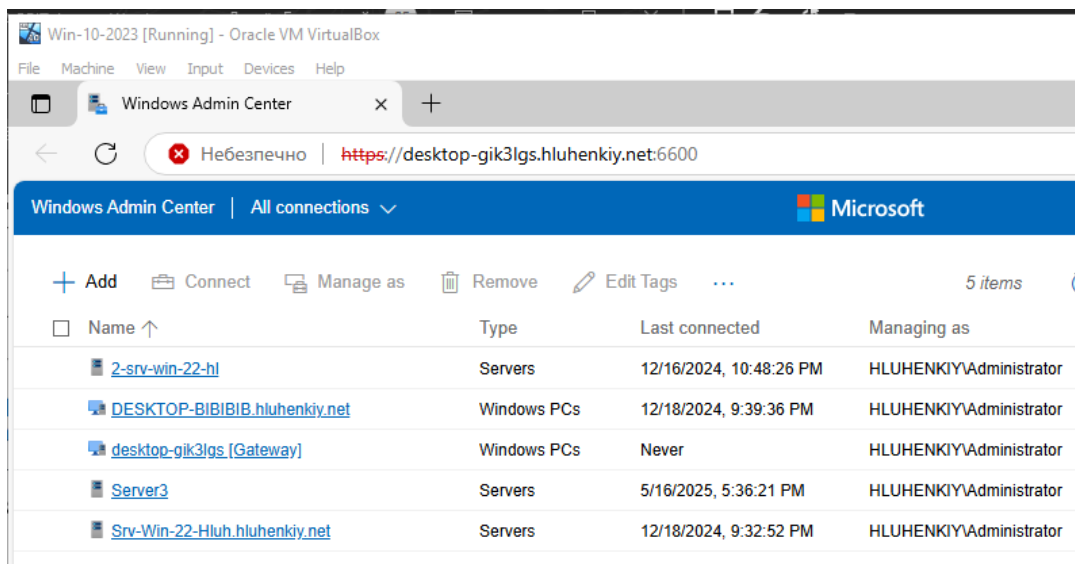


Рис. 72 Перевірка під'єднаних машин

Перейдемо до перевірки зв'язку на контролері домену. Оглянемо якість зв'язку зі станціями та правильність налаштування політик:

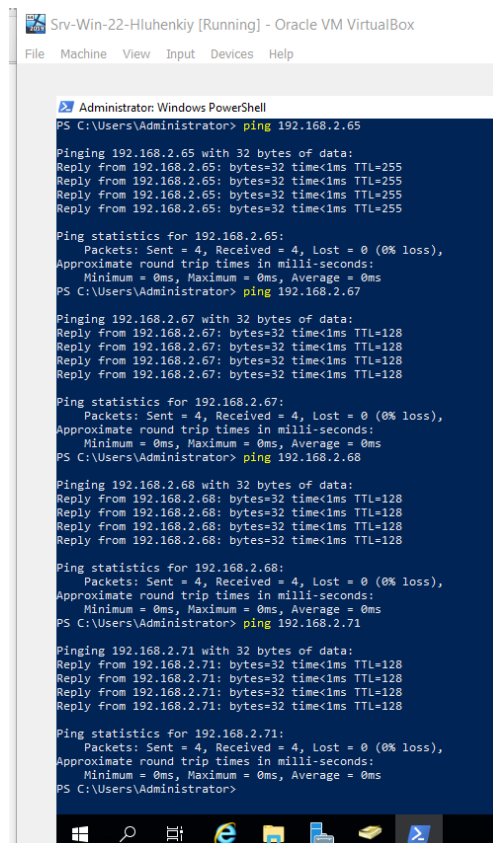


Рис. 73 Перевірка з'єднання

Оглянемо працездатність налаштованих політик та інструментів на контроллері домену:

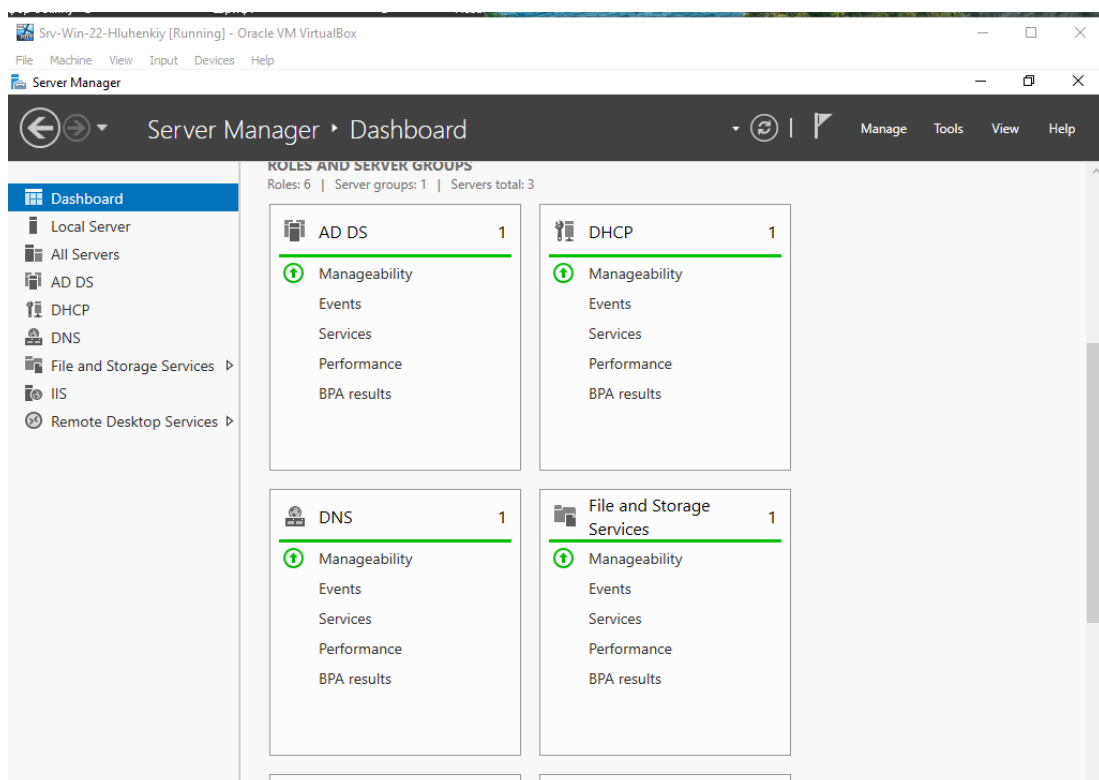


Рис. 74 Перевірка налаштувань політик

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				108
Змн.	Арк.	№ докум.	Підпис	Дата		

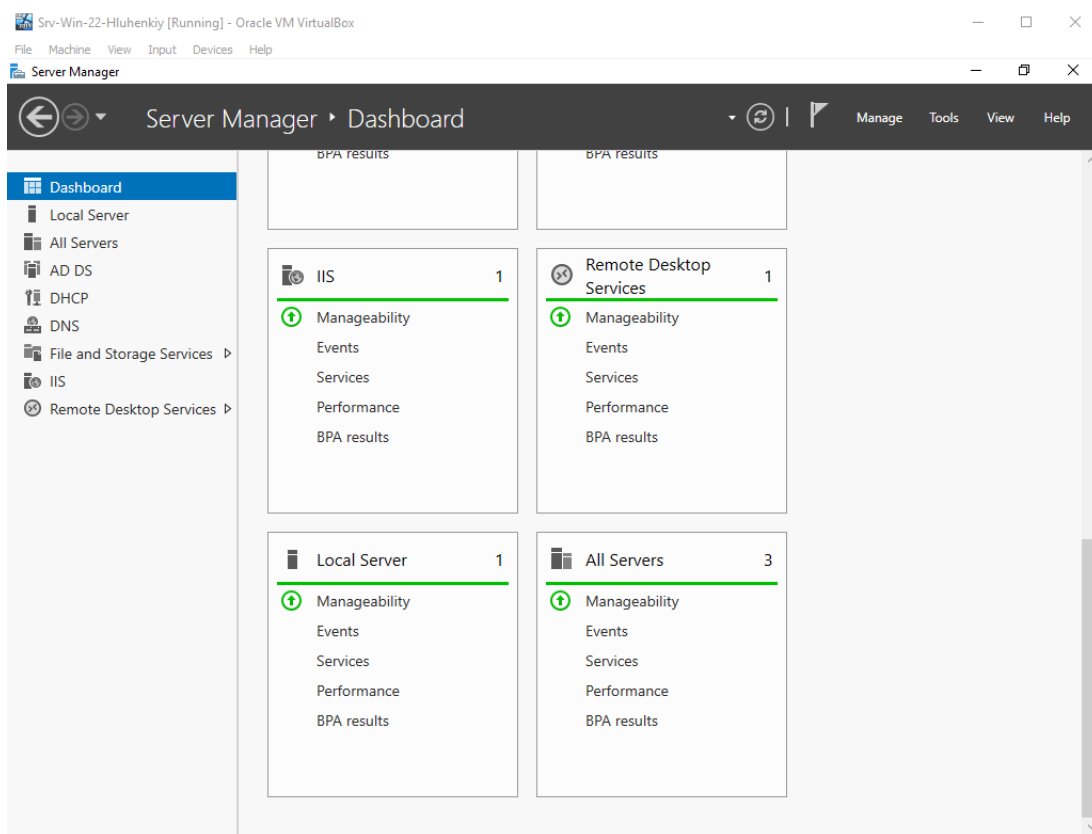


Рис. 75 Перевірка налаштувань політик

Оглянемо працездатність резервного сервера.

```

2-Srv-Win-22-Hluhenkiy [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Administrator: Windows PowerShell
PS C:\Users\Administrator> ping 192.168.2.65

Pinging 192.168.2.65 with 32 bytes of data:
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255
Reply from 192.168.2.65: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.2.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator> ping 192.168.2.67

Pinging 192.168.2.67 with 32 bytes of data:
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator> ping 192.168.2.71

Pinging 192.168.2.71 with 32 bytes of data:
Reply from 192.168.2.71: bytes=32 time<1ms TTL=128
Reply from 192.168.2.71: bytes=32 time<1ms TTL=128
Reply from 192.168.2.71: bytes=32 time<1ms TTL=128
Reply from 192.168.2.71: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.71:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\Administrator> ping Srv-Win-22-Hluh

Pinging Srv-Win-22-Hluh.hluhenkiy.net [192.168.2.67] with 32 bytes of data:
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128
Reply from 192.168.2.67: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

Рис. 76 Перевірка налаштувань з'єднання

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				109
Змн.	Арк.	№ докум.	Підпис	Дата		

3.5.1 Перевірка налаштувань DHCP через Windows Admin Center

В попередніх підрозділах було виконано перенесення ролі DHCP-серверу на окремий Сервер №3. Були оглянуті основні ознаки правильності проведеної роботи, проте в даному підрозділі буде проведений практичний дослід для закріплення результатів.

Підключимось до Windows Admin Center з обліковими даними адміністратора та підключимось до Сервера №3. Необхідно перевірити чи справді наявна встановлена роль DHCP та чи працюють потрібні сервіси.

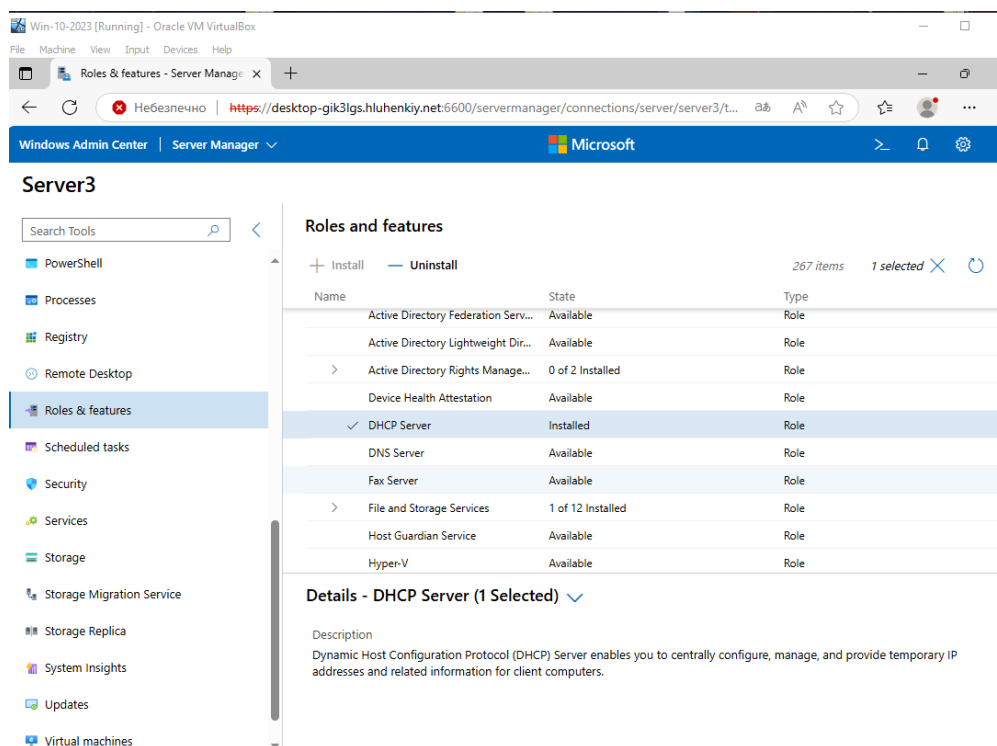


Рис. 77 Перевірка ролей на Сервері №3

Як бачимо з фото роль встановлена. Тепер перевіримо всі наявні служби і процеси на Сервері №3 та знайдемо там DHCP.

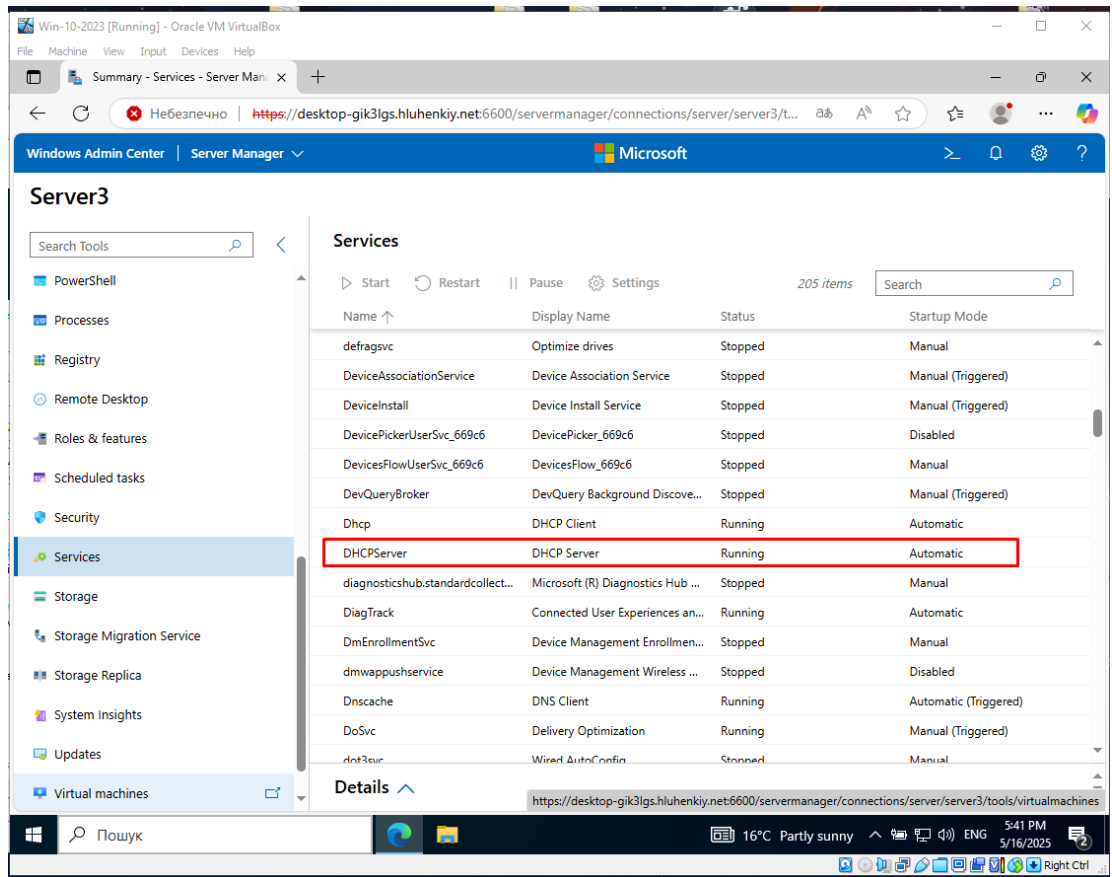


Рис. 78 Перевірка сервісів на Сервері №3

Сервіси та ролі були успішно встановлені та запущені, це значить, що налаштування були виконані правильно. Для закріплення результатів дослідження проведемо практичний експеримент: за допомогою WAC створимо віддалено на Сервері №3 новий діапазон DHCP-адрес, протестуємо його та відключимо.

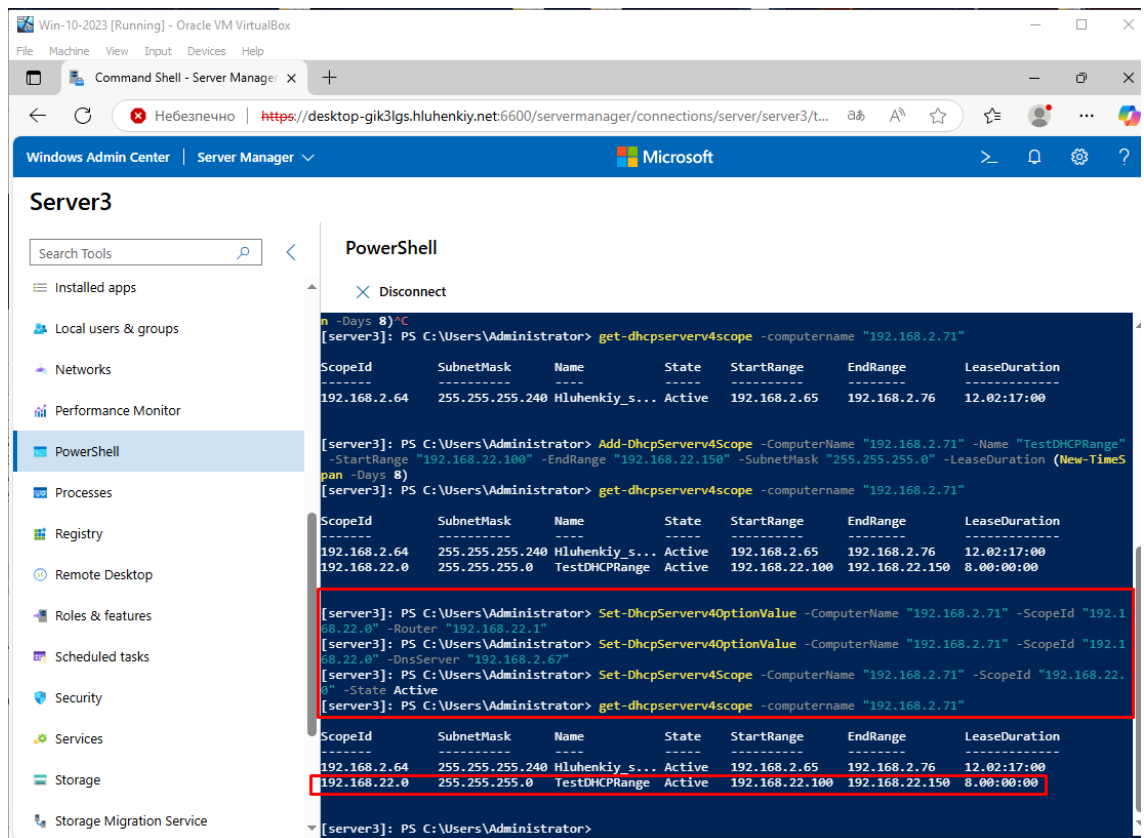


Рис. 79 Створення нового діапазону адрес

Одразу після створення діапазону було протестовано моніторингову команду для виведення всіх наявних діапазонів в мережі. Як показано на фото діапазон був створений, та знаходиться в списку поряд з основним діапазоном. Тепер спробуємо його відключити та перевірити чи вдалось це.

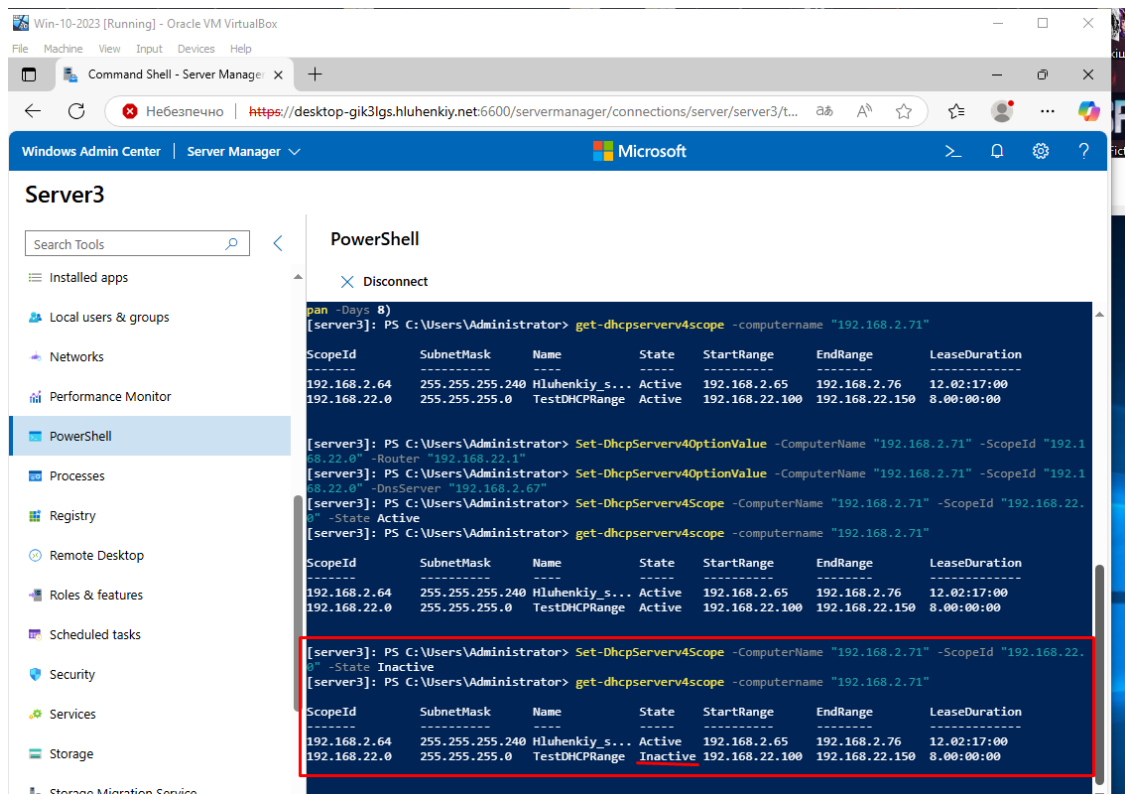


Рис. 80 Видалення діапазону адрес

Було успішно деактивовано діапазон DHCP-адрес. За потреби адміністратор домену може в будь який момент підключитись до Серверу №3 використовуючи WAC та активувати цей діапазон, або ж створити новий, якщо цього вимагає положення.

Отже проведені тести підтвердили коректну роботу спроектованої мережевої інфраструктури, включаючи налаштування DHCP, моніторинг через Windows Admin Center та автоматизацію адміністрування за допомогою PowerShell DSC. Аналіз результатів засвідчив стабільність і ефективність системи, а виявлені незначні відхилення успішно усунуто, що забезпечило відповідність вимогам проєкту та готовність мережі до подальшої експлуатації. Всі конфігураційні файли та додаткові скріншоти можна побачити у [додатку Б](#).

Висновок до третього розділу

В даному розділі було виконано комплексну реалізацію та оцінку спроектованої мережевої інфраструктури, що включала вибір обладнання, налаштування DHCP-сервера, конфігурацію Windows Admin Center, автоматизацію адміністрування через PowerShell Desired State Configuration та тестування функціональності мережі. Було проведено всі необхідні налаштування для підготовки схеми до роботи в реальному середовищі. Вибрано відповідне обладнання та образи віртуальних машин, а також успішно реалізовано всі завдання, вирішивши проблеми, підняті в попередніх розділах.

Проведене тестування та перевірка не виявили помилок; налаштовано віддалений доступ до пристроїв, перевірено внутрішній трафік схеми та створено кілька DSC-конфігурацій для автоматизації управління мережевою інфраструктурою. Результати засвідчили стійкість даної версії продукту до збоїв у роботі.

Крім того, тестування підтвердило, що мережа виконує всі функції, визначені в технічному завданні та описані в попередніх розділах, що забезпечує її готовність до практичного застосування.

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				114
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання даної роботи було проаналізовано основи прикладної галузі, такої як автоматизація адміністрування комп'ютерних мереж. Незважаючи на наявність великої кількості різних подібних схем стає питання доступності, зручності та безпеки їх використання.

У першому розділі було закладено теоретичну основу, проаналізовано ключові технології автоматизації, зокрема WAC і DSC, їхні переваги (зручність, декларативний підхід) та обмеження (масштабованість), а також визначено вимоги до проєкту, що стало основою для подальших етапів.

Другий розділ охопив проєктування віртуальної IPv4-мережі, включаючи розробку топологій, інтеграцію віртуалізації через VirtualBox і Windows Server 2019, налаштування DHCP, Active Directory та системи безпеки, що забезпечило міцну основу для практичної реалізації.

У третьому розділі проведено комплексну реалізацію спроектованої інфраструктури: виконано налаштування обладнання, створено DSC-конфігурації для автоматизації управління, організовано віддалений доступ через WAC і протестовано функціональність мережі, підтвердивши її стійкість до збоїв і відповідність технічному завданню. Під час тестування програма проходила різні непередбачені розробником випадки та ситуації.

У результаті дослідження створено функціональну, стабільну та автоматизовану мережеву інфраструктуру, яка готова до роботи в реальному середовищі. Отримані результати мають практичну цінність і можуть бути використані як основа для подальшого масштабування та вдосконалення мережевих рішень у корпоративному середовищі.

Таким чином було забезпечено автоматизацію адміністрування мережі за допомогою утиліти Windows Admin Center та PowerShell DSC. Для створення проєкта використовувалась платформа Graphical Network Simulator-3, та інструмент для віртуалізації VirtualBox.

Дана робота буде розвиватися та вдосконалюватися. Буде проведено низку тривалих тестувань задля покращення якості продукту.

		Глухенький Д.О			KKIKB.KP.B-123-03-P3	Арк.
		Фальковський І.І				115
Змн.	Арк.	№ докум.	Підпис	Дата		

Список використаних джерел

1. Навчальний курс: Адміністрування комп'ютерних систем та мереж (Лабораторна робота №8 «Основи використання PowerShell для адміністрування домену Active Directory») [Електронний ресурс] - Режим доступу до ресурсу:
https://learn.ztu.edu.ua/pluginfile.php/339766/mod_resource/content/2/2023-24_AKCM_Лаб_08.pdf
2. Навчальний курс: Системний та мережевий моніторинг [Електронний ресурс] - Режим доступу до ресурсу:
<https://learn.ztu.edu.ua/course/view.php?id=5902>
3. Навчальний курс: Learn about DSC and managing Windows Servers with configuration as code (Server Academy) [Електронний ресурс] - Режим доступу до ресурсу: <https://www.serveracademy.com>
4. Стаття: Understanding Desired State Configuration (DSC) Basics (Pluralsight) [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.pluralsight.com>
5. Навчальний курс: PowerShell DSC for Beginners [Електронний ресурс] - Режим доступу до ресурсу: <https://www.udemy.com>
6. Стаття: Get started with Desired State Configuration (DSC) for Windows (Microsoft Learn) [Електронний ресурс] - Режим доступу до ресурсу:
<https://learn.microsoft.com>
7. Стаття: PowerShell DSC: Best Practices and Common Pitfalls (PowerShell Magazine) [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.powershellmagazine.com>
8. Стаття: Introduction to PowerShell Desired State Configuration (DSC) (DevOps Institute) [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.devopsinstitute.com>
9. Стаття: Install Windows Admin Center (WAC) using DSC (PowerShell, Programming and DevOps) [Електронний ресурс] - Режим доступу до ресурсу: <https://dscottraynsford.wordpress.com>

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				116
Змн.	Арк.	№ докум.	Підпис	Дата		

- 10.Стаття: Windows Admin Center Overview and Deployment Guide (Tech Community Microsoft) [Електронний ресурс] - Режим доступу до ресурсу: <https://techcommunity.microsoft.com>
- 11.Книга: "Mastering Windows Admin Center" by Jordan Krause (O'Reilly Media) [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oreilly.com>
- 12.Стаття: Microsoft Windows Server 2019 Installation Step by Step Guide (Tech Encyclopedia) [Електронний ресурс] - Режим доступу до ресурсу: <https://techencyclopedia.wordpress.com>
- 13.Стаття: Windows Server Hardening with PowerShell DSC (NVISO Labs) [Електронний ресурс] - Режим доступу до ресурсу: <https://blog.nviso.eu>
- 14.Книга: "Windows PowerShell in Action" by Bruce Payette [Електронний ресурс] - Режим доступу до ресурсу: <https://www.doc-developpement-durable.org/file/Projets-informatiques/cours-&-manuels-informatiques/Windows/Windows%20PowerShell%20in%20Action,%202nd%20Edition.pdf>
- 15.Курс: "Automating Administration with Windows PowerShell" (Microsoft Learn) [Електронний ресурс] - Режим доступу до ресурсу: <https://learn.microsoft.com/training/paths/administer-infrastructure-windows-powershell>
- 16.PowerShell DSC-exsmpls GITHUB: [Електронний ресурс] - Режим доступу до ресурсу: <https://github.com/PowerShell/DSC-Samples>
- 17.Використання PowerShell для керування налаштуваннями Windows Admin Center: [Електронний ресурс] - Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/windows-server/manage/windows-admin-center/configure/use-powershell?>
- 18.Початок роботи з Desired State Configuration (DSC) для Windows: [Електронний ресурс] - Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/powershell/dsc/getting-started/wingettingstarted?view=dsc-1.1>

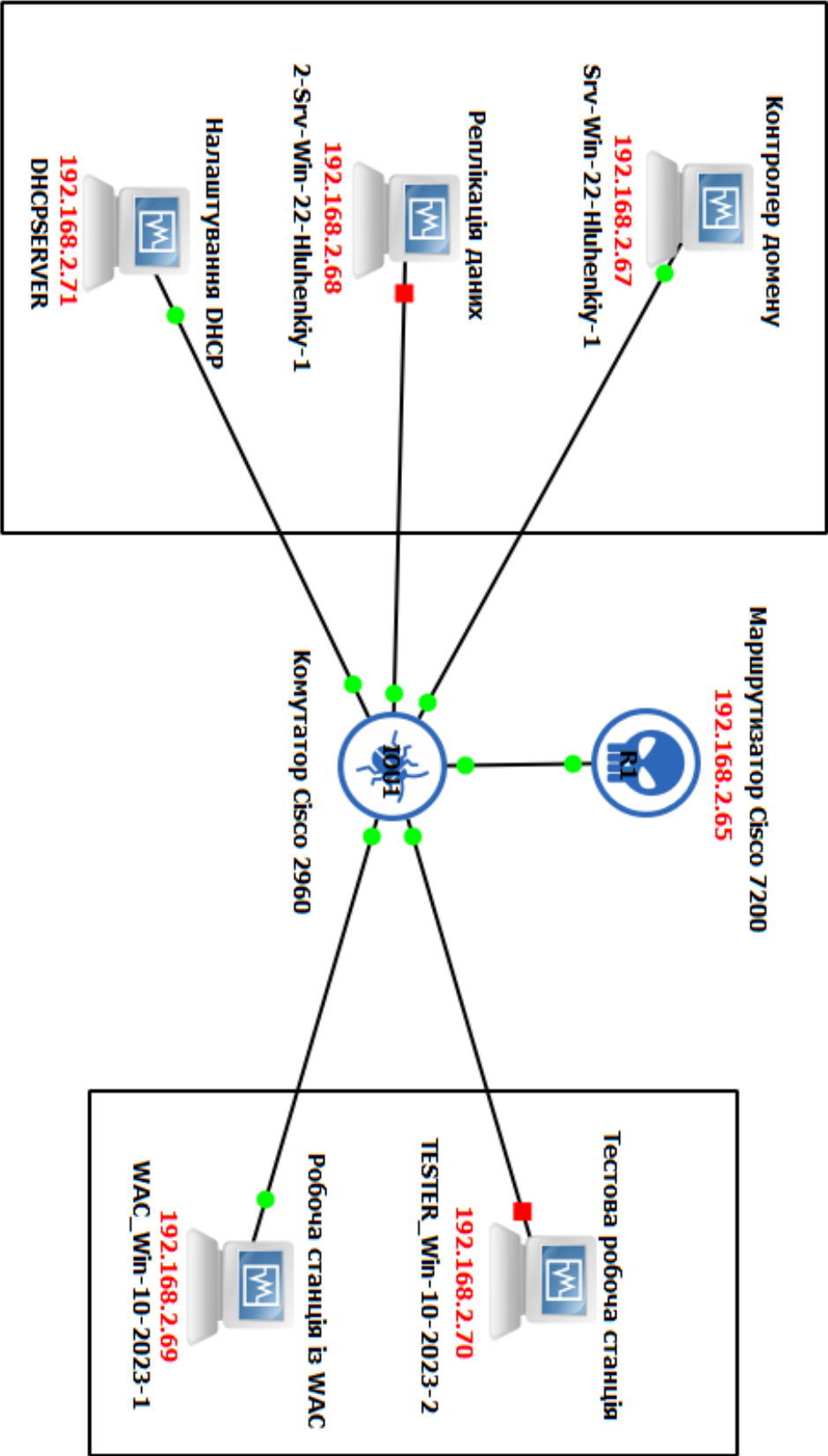
		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				117
Змн.	Арк.	№ докум.	Підпис	Дата		

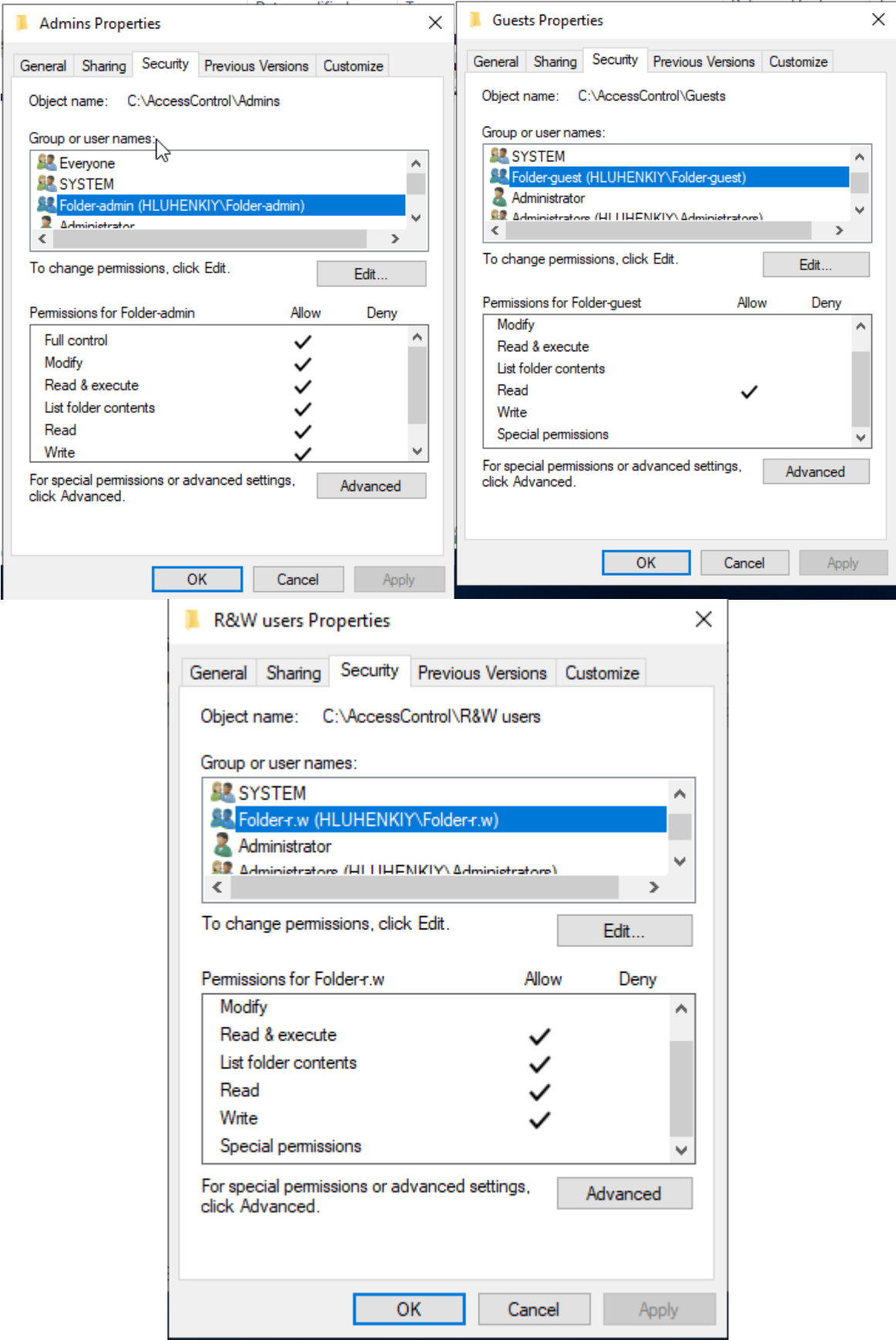
19. Навчальний курс: Network Manage [Електронний ресурс] - Режим доступу: www.netacad.com
20. Форум: PowerShell Community [Електронний ресурс] - Режим доступу: <https://powershell.org>
21. Курс: комп'ютерна грамотність: від «чайника» до професіонала [Електронний ресурс] - Режим доступу: https://balka-book.com/ua/samouchiteli_raboty_i_noutbuk_kompyuter_planshet-30
22. Стаття: Petya Ransomware Mitigation Strategies (Cisco Talos Intelligence) [Електронний ресурс] - Режим доступу: <https://www.talosintelligence.com>
23. Стаття: Understanding and Defending Against NotPetya (McAfee Blogs) [Електронний ресурс] - Режим доступу: <https://www.mcafee.com/blogs>

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				118
Змн.	Арк.	№ докум.	Підпис	Дата		

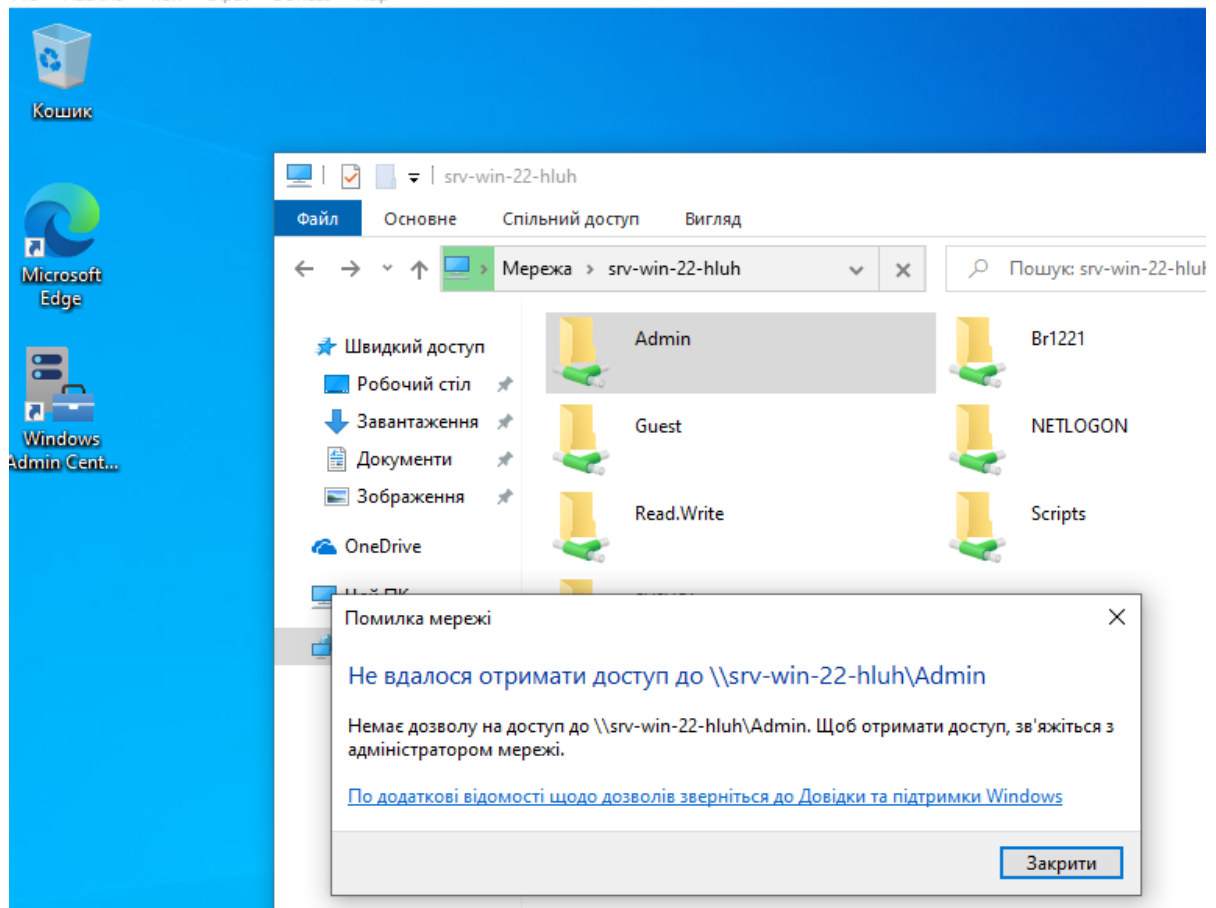
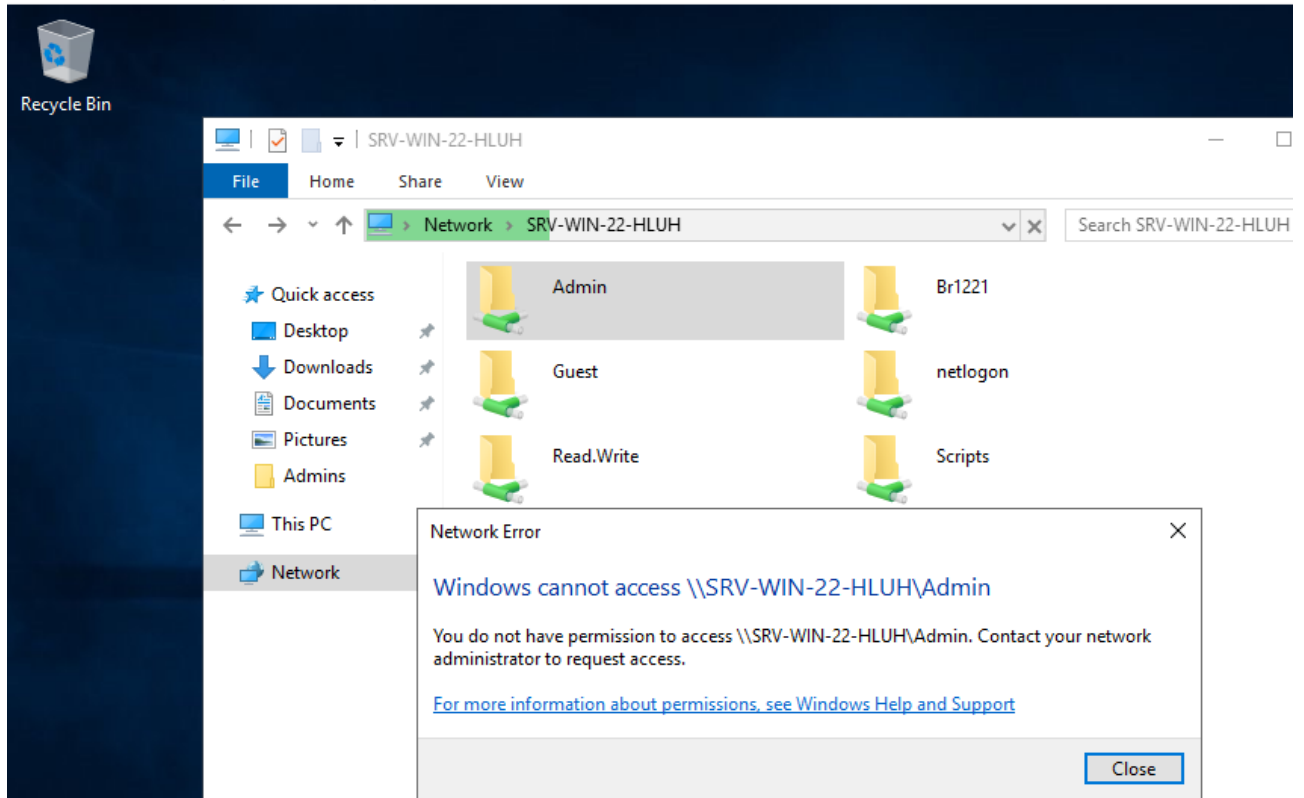
ДОДАТКИ

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				119
Змн.	Арк.	№ докум.	Підпис	Дата		



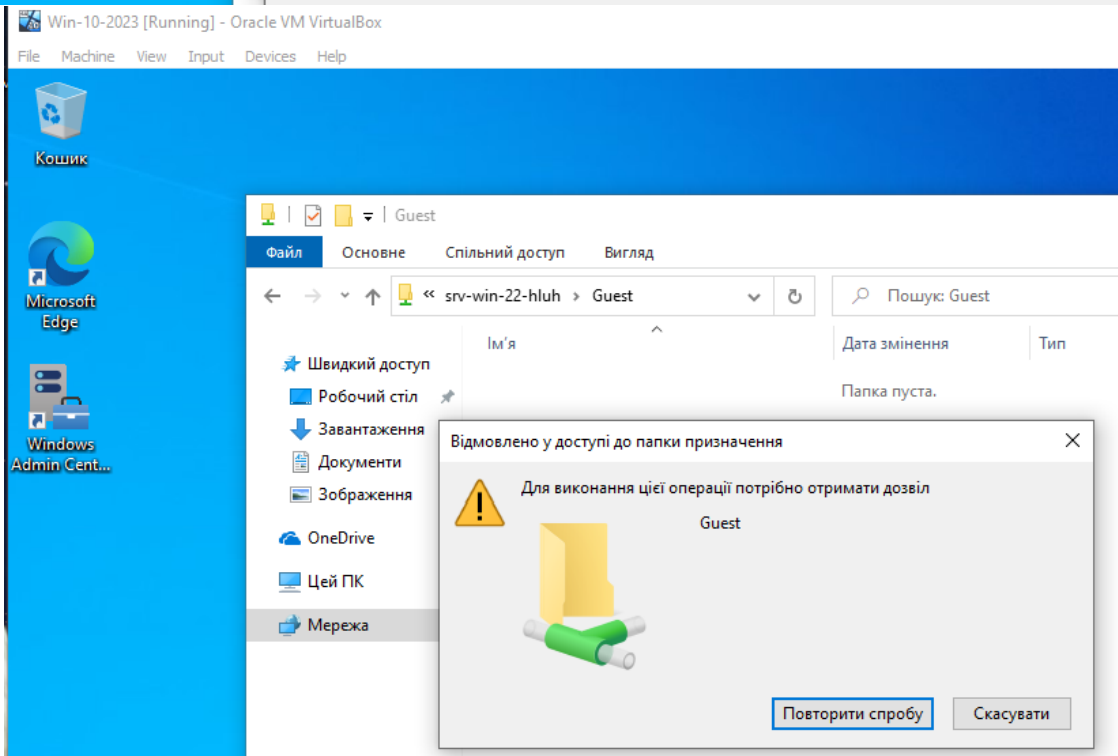
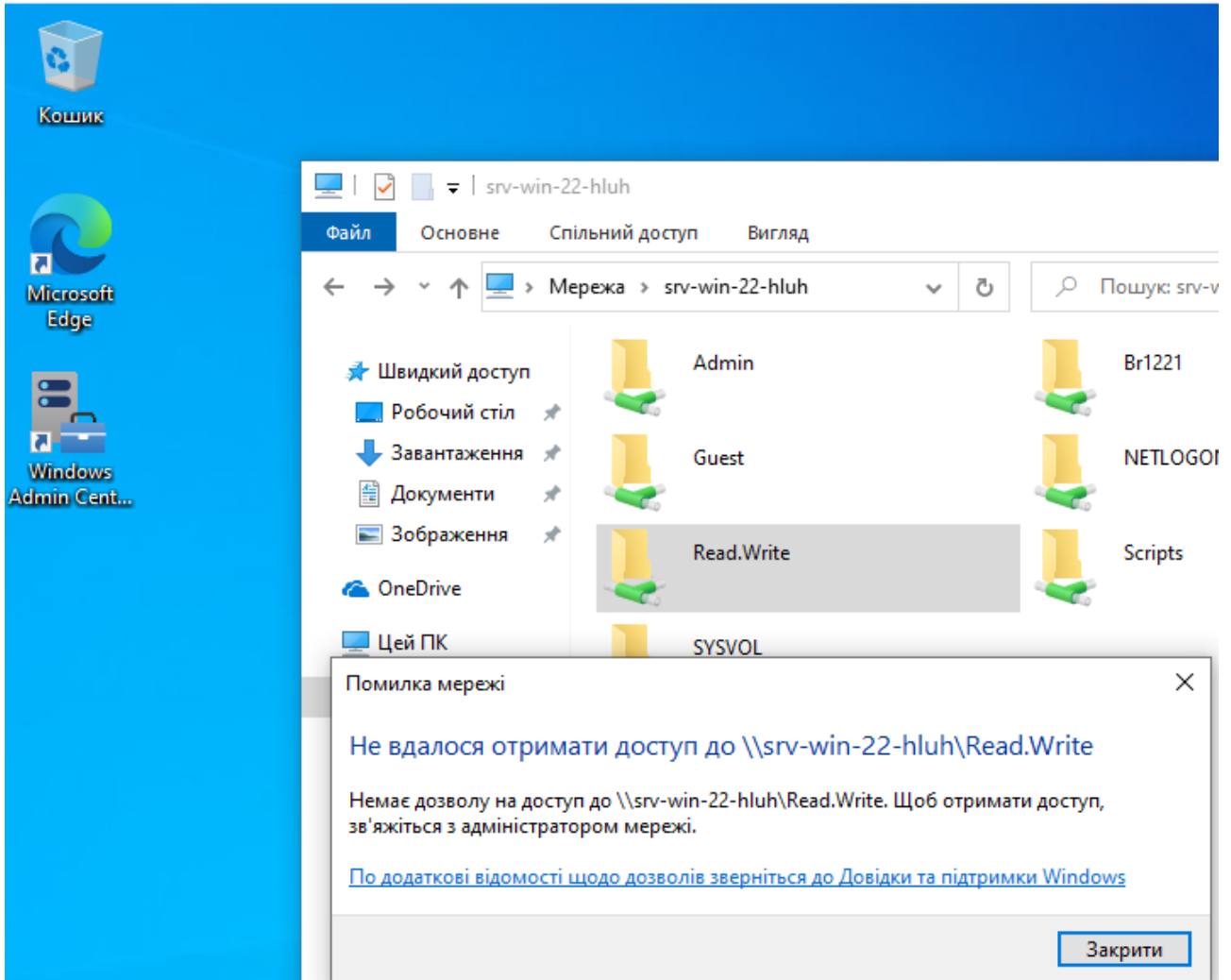


Налаштування прав доступу до теки



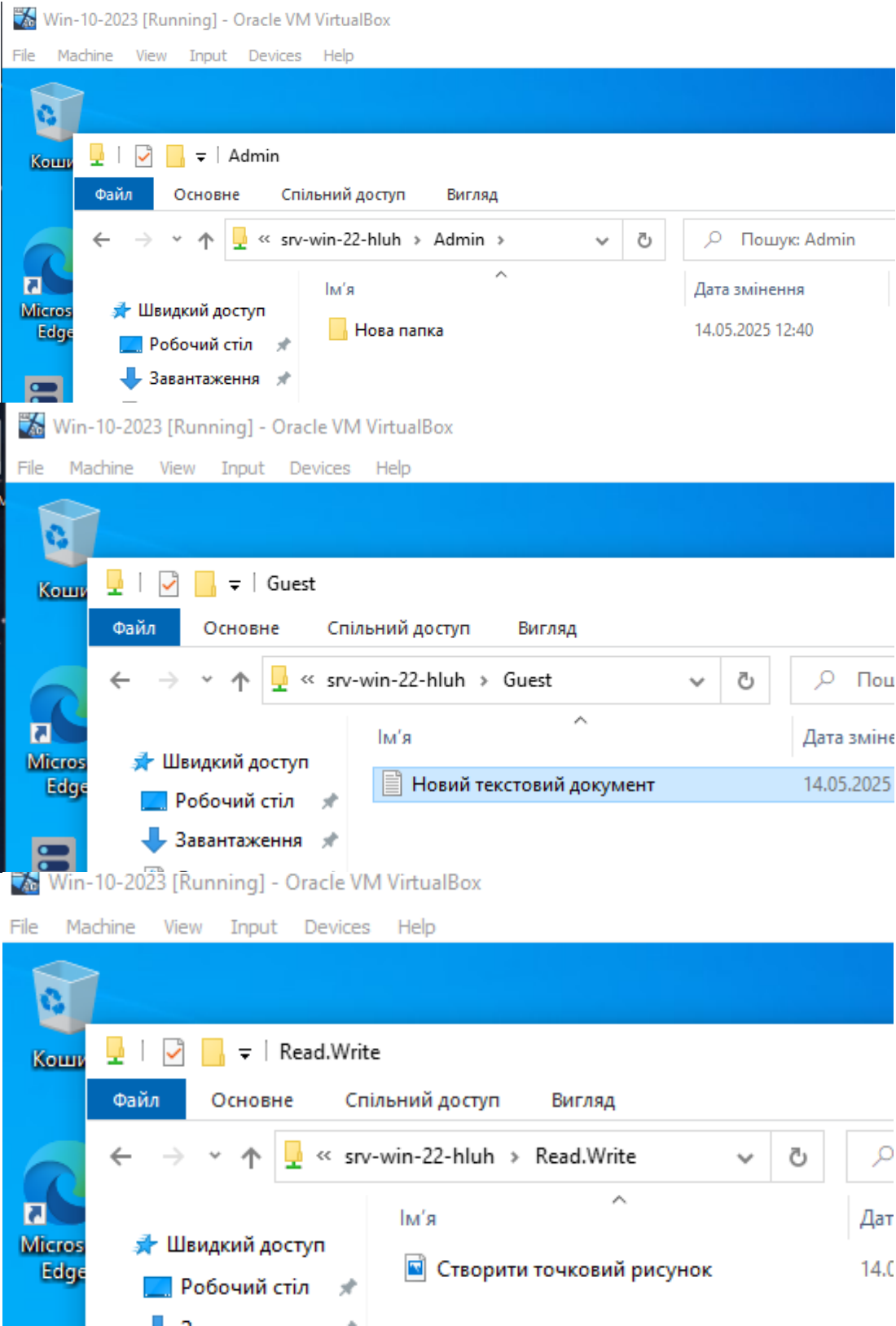
Перевірка прав доступу до теки

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				122
Змн.	Арк.	№ докум.	Підпис	Дата		



Перевірка прав доступу до теки

		Глухенький Д.О.			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				123
Змн.	Арк.	№ докум.	Підпис	Дата		



Перевірка прав доступу до теки

SpaceMonitor.ps1

```

Configuration DiskSpaceMonitor {
    param (
        [string]$DriveLetter = "C",
        [int]$MinFreeSpaceGB = 10
    )
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node localhost {
        Script CheckDiskSpace {
            SetScript = {
                $drive = Get-PSDrive -Name $using:DriveLetter -ErrorAction
                SilentlyContinue
                if ($null -ne $drive) {
                    $freeSpace = [math]::Round($drive.Free / 1GB, 2)
                    if ($freeSpace -lt $using:MinFreeSpaceGB) {
                        Write-Output "Warning: Free space on drive $($using:DriveLetter): is
below $($using:MinFreeSpaceGB) GB ($freeSpace GB available)."
                        Add-Content -Path "C:\Logs\DiskSpaceMonitor.log" -Value "[$(Get-
Date)] WARNING: Drive $($using:DriveLetter) has low disk space: $freeSpace GB
free."
                    }
                }
            }
            TestScript = {
                $drive = Get-PSDrive -Name $using:DriveLetter -ErrorAction
                SilentlyContinue
                if ($null -ne $drive) {
                    $freeSpace = [math]::Round($drive.Free / 1GB, 2)
                    return $freeSpace -ge $using:MinFreeSpaceGB
                }
                return $false
            }
            GetScript = {
                @{
                    DriveLetter = $using:DriveLetter
                    MinFreeSpaceGB = $using:MinFreeSpaceGB
                }
            }
        }
    }
}

```

DiskSpaceMonitor -DriveLetter "C" -MinFreeSpaceGB 50

		Глухенький Д.О			ККИКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				125
Змн.	Арк.	№ докум.	Підпис	Дата		

NewOU.ps1

```
Configuration CreateOU {  
    param (  
        [string]$OUName,  
        [string]$Domain  
    )  
}
```

```
Import-DscResource -ModuleName 'xActiveDirectory'
```

```
Node $Domain {  
    xADOrganizationalUnit CreateOU {  
        Name = $OUName  
        Path = "DC=hluhenkiy,DC=net"  
        Ensure = 'Present'  
    }  
}  
}
```

```
$OUName = Read-Host -Prompt "Введіть назву організаційної одиниці (OU)"  
$Domain = "srv-win-22-hluh.hluhenkiy.net"
```

```
CreateOU -OUName $OUName -Domain $Domain  
Start-DscConfiguration -Path .\CreateOU -Wait -Verbose
```

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				126
Змн.	Арк.	№ докум.	Підпис	Дата		

CreateGroup.ps1

```
Configuration CreateGroup {  
    param (  
        [string]$GroupName,  
        [string]$Domain  
    )  
  
    Import-DscResource -ModuleName 'xActiveDirectory'  
  
    Node $Domain {  
        xADGroup CreateADGroup {  
            GroupName = $GroupName  
            Ensure = 'Present'  
            Path = "DC=hluhenkiy,DC=net"  
            Description = "Група для тестових користувачів"  
        }  
    }  
}  
  
$Domain = "srv-win-22-hluh.hluhenkiy.net"  
CreateGroup -GroupName "TestGroup" -Domain $Domain  
  
Start-DscConfiguration -Path .\CreateGroup -Wait -Verbose
```

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				127
Змн.	Арк.	№ докум.	Підпис	Дата		

InstallDHCPserver.ps1

Configuration InstallDHCPServer {

param (

[string]\$ComputerName = "192.168.2.71"

)

Import-DscResource -ModuleName 'PSDesiredStateConfiguration'

Node \$ComputerName {

WindowsFeature DHCP {

Ensure = "Present"

Name = "DHCP"

IncludeAllSubFeature = \$true

}

Service DHCPServer {

Name = "DHCPServer"

State = "Running"

StartupType = "Automatic"

DependsOn = "[WindowsFeature]DHCP"

}

}

}

InstallDHCPServer -OutputPath .

Start-DscConfiguration -Path .\InstallDHCPServer -ComputerName 192.168.2.71 -

Wait -Verbose

		Глухенький Д.О			ККИКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				
Змн.	Арк.	№ докум.	Підпис	Дата		128

CreateTestFolder.ps1

```
Configuration CreateTestFolder {
    Import-DscResource -ModuleName 'PSDesiredStateConfiguration'
    Node "localhost" {
        File TestFolder {
            Ensure = "Present"
            Type = "Directory"
            DestinationPath = "C:\PowerShellScripts\FolderAccess\TestFolder"
        }

        Script SetPermissions {
            DependsOn = "[File]TestFolder"
            GetScript = {
                @{
                    Result = (Get-Acl -Path
"C:\PowerShellScripts\FolderAccess\TestFolder").Access | Out-String
                }
            }
            TestScript = {
                $acl = Get-Acl -Path "C:\PowerShellScripts\FolderAccess\TestFolder"

                $group = $acl.Access | Where-Object {
                    $_.IdentityReference -eq "hluhenkiy\Folder-admin" -and
                    $_.FileSystemRights -eq "ReadAndExecute"
                }
                return ($admin -ne $null -and $group -ne $null)
            }
            SetScript = {
                $path = "C:\PowerShellScripts\FolderAccess\TestFolder"
                $acl = Get-Acl -Path $path

                # Забороняємо наслідування та чистимо права
                $acl.SetAccessRuleProtection($true, $false)
                $acl.Access | ForEach-Object { $acl.RemoveAccessRule($_) }

                $groupRule = New-Object
System.Security.AccessControl.FileSystemAccessRule(
                    "hluhenkiy\Folder-admin", "ReadAndExecute",
                    "ContainerInherit,ObjectInherit", "None", "Allow")
                $acl.AddAccessRule($groupRule)

                Set-Acl -Path $path -AclObject $acl
                Write-Output "Permissions set."
            }
        }
    }
}
CreateTestFolder -OutputPath .\CreateTestFolder
```

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				129
Змн.	Арк.	№ докум.	Підпис	Дата		

SyncTime.ps1

```
param (
    [string[]]$servers = @("192.168.2.68", "192.168.2.71"),
    [int]$allowedDriftSeconds = 5,
    [System.Management.Automation.PSCredential]$cred = $null
)

$localTime = Get-Date
Write-Host "Local time on domain controller (this machine): $localTime`n"

foreach ($server in $servers) {
    try {
        if ($cred -ne $null) {
            $remoteTime = Invoke-Command -ComputerName $server -Credential $cred
        } else {
            $remoteTime = Invoke-Command -ComputerName $server -ScriptBlock {
                Get-Date } -ErrorAction Stop
        }

        if ($remoteTime -eq $null) {
            Write-Warning "[$server] Unable to retrieve remote time (null value).
            Skipping..."
            continue
        }

        # Обчислення різниці в секундах
        $delta = ($remoteTime - $localTime).TotalSeconds
        Write-Host "[$server] Current time: $remoteTime ( $\Delta$   $[(\text{math})::\text{Round}(\$delta, 2)]$  sec)"

        if ( $[(\text{math})::\text{Abs}(\$delta)]$  -gt $allowedDriftSeconds) {
            Write-Host "[$server] Time drift > $allowedDriftSeconds seconds.
            Attempting to sync..."

            if ($cred -ne $null) {
                Invoke-Command -ComputerName $server -Credential $cred -ScriptBlock {
                    w32tm /resync /nowait | Out-Null
                } -ErrorAction Stop
            } else {
                Invoke-Command -ComputerName $server -ScriptBlock {
                    w32tm /resync /nowait | Out-Null
                } -ErrorAction Stop
            }
        }
    }
}
```

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				130
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Write-Host "[$server] Time sync triggered."
    } else {
        Write-Host "[$server] Time is within allowed drift."
    }

} catch {
    Write-Warning "[$server] Error: $($_.Exception.Message)"
}

Write-Host ""
}

```

		Глухенький Д.О			ККІКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				131
Змн.	Арк.	№ докум.	Підпис	Дата		

DNSConfig.ps1

```
configuration SetCorrectDNS {
    param (
        [string[]]$NodeNames = @("192.168.2.69"),
        [string]$DnsAddress = "192.168.2.67"
    )
    Import-DscResource -ModuleName PSDesiredStateConfiguration

    Node $NodeNames {
        Script SetDNS {
            GetScript = {
                $interfaces = Get-DnsClientServerAddress -AddressFamily IPv4 | Where-Object { $_.InterfaceAlias -ne "Loopback Pseudo-Interface 1" }
                $currentDNS = ($interfaces | Select-Object -ExpandProperty ServerAddresses) -join ","
                return @{ Result = $currentDNS }
            }
            SetScript = {
                $interfaces = Get-NetAdapter | Where-Object { $_.Status -eq "Up" }
                foreach ($adapter in $interfaces) {
                    Set-DnsClientServerAddress -InterfaceAlias $adapter.Name -ServerAddresses $using:DnsAddress
                }
            }
            TestScript = {
                $interfaces = Get-DnsClientServerAddress -AddressFamily IPv4 | Where-Object { $_.InterfaceAlias -ne "Loopback Pseudo-Interface 1" }
                foreach ($entry in $interfaces) {
                    if ($entry.ServerAddresses -notcontains $using:DnsAddress) {
                        return $false
                    }
                }
                return $true
            }
        }
    }

    $cred = Get-Credential
    $nodes = @("192.168.2.69")
    $output = "C:\Scripts\DNSConfigMOF"
    SetCorrectDNS -NodeNames $nodes -OutputPath $output
    foreach ($node in $nodes) {
        Start-DscConfiguration -ComputerName $node -Credential $cred -Path $output -Wait -Force -Verbose
    }
}
```

		Глухенький Д.О			ККИКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				132
Змн.	Арк.	№ докум.	Підпис	Дата		

DisableUSB.ps1

[CmdletBinding()]

param (

[Parameter(Mandatory)]

[PSCredential] \$Credential

)

Configuration DisableUSBPorts {

param (

[Parameter(Mandatory)]

[string[]] \$Nodes,

[PSCredential] \$Credential

)

Import-DscResource -ModuleName PSDesiredStateConfiguration

Node \$Nodes {

Registry DisableUSB {

Key = "HKLM:\SYSTEM\CurrentControlSet\Services\USBSTOR"

ValueName = "Start"

ValueData = 4

ValueType = "DWord"

Ensure = "Present"

}

Script LogUSB {

DependsOn = "[Registry]DisableUSB"

SetScript = {

\$logFile = "C:\Logs\DisableUSB.log"

if (-not (Test-Path \$logFile)) { New-Item -Path \$logFile -ItemType File - Force }

Add-Content -Path \$logFile -Value "\$(Get-Date -Format 'yyyy-MM-dd HH:mm:ss') - USB disabled on \$(\$using:Node)"

}

TestScript = { \$true }

GetScript = { @{ Result = "Log created on \$(\$using:Node)" } } }

if (-not (Test-Path "C:\Scripts\DisableUSBMOF")) {

New-Item -Path "C:\Scripts\DisableUSBMOF" -ItemType Directory -Force}

\$targetMachines = @("192.168.2.69")

DisableUSBPorts -Nodes \$targetMachines -Credential \$Credential -OutputPath "C:\Scripts\DisableUSBMOF"

Start-DscConfiguration -Path "C:\Scripts\DisableUSBMOF" -Wait -Verbose - Credential \$Credential -Force

		Глухенький Д.О.			ККИБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І.				133
Змн.	Арк.	№ докум.	Підпис	Дата		

TEMPclean.ps1

```
[CmdletBinding()]
```

```
param (
```

```
    [Parameter(Mandatory)]
```

```
    [string[]] $ComputerNames,
```

```
    [Parameter(Mandatory)]
```

```
    [PSCredential] $Credential
```

```
)
```

```
Configuration ClearTempFiles {
```

```
    param (
```

```
        [Parameter(Mandatory)]
```

```
        [string[]] $Nodes,
```

```
        [PSCredential] $Credential
```

```
    )
```

```
    Import-DscResource -ModuleName PSDesiredStateConfiguration
```

```
    Node $Nodes {
```

```
        Script ClearTemp {
```

```
            GetScript = {
```

```
                @{ Result = "Check temp files on $($using:Node)" }
```

```
            }
```

```
            TestScript = {
```

```
                return $false # Always perform cleanup
```

```
            }
```

```
            SetScript = {
```

```
                $tempPath = "C:\Windows\Temp\*"
```

```
                $logFile = "C:\Logs\ClearTemp.log"
```

```
                if (-not (Test-Path $logFile)) { New-Item -Path $logFile -ItemType File -
```

```
Force }
```

```
                try {
```

```
                    Remove-Item -Path $tempPath -Recurse -Force -ErrorAction Stop
```

```
                    Add-Content -Path $logFile -Value "$(Get-Date -Format 'yyyy-MM-dd
```

```
HH:mm:ss') - Temp files cleared on $($using:Node)"
```

```
                } catch {
```

```
                    Add-Content -Path $logFile -Value "$(Get-Date -Format 'yyyy-MM-dd
```

```
HH:mm:ss') - Error clearing temp files on $($using:Node): $_"
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

		Глухенький Д.О			ККИКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				134
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# Check if MOF directory exists
if (-not (Test-Path "C:\Scripts\TEMPcleanMOF")) {
    New-Item -Path "C:\Scripts\TEMPcleanMOF" -ItemType Directory -Force
}

# Check online computers
$onlineComputers = foreach ($name in $ComputerNames) {
    if (Test-Connection -ComputerName $name -Count 1 -Quiet) {
        $name
    } else {
        Write-Host "Skipping $name (offline or unreachable)" -ForegroundColor Yellow
    }
}

if ($onlineComputers.Count -gt 0) {
    ClearTempFiles -Nodes $onlineComputers -Credential $Credential -OutputPath
"C:\Scripts\TEMPcleanMOF"
    Start-DscConfiguration -Path "C:\Scripts\TEMPcleanMOF" -Wait -Verbose -
Credential $Credential -Force
} else {
    Write-Host "No machines are available."
}
```

		Глухенький Д.О			ККИКБ.КР.Б-123-03-ПЗ	Арк.
		Фальковський І.І				135
Змн.	Арк.	№ докум.	Підпис	Дата		