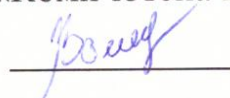


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми моніторингу гетерогенної мережі на основі Nagios
Core »

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

 Віталій ХОМЕНЮК

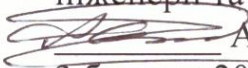
Керівник; доцент кафедри комп'ютерної інженерії та
кібербезпеки, кандидат технічних наук, доцент

 Юрій РОССІНСЬКИЙ

Рецензент; завідувач кафедри комп'ютерної інженерії та
кібербезпеки, кандидат технічних наук, доцент

 Андрій ЄФІМЕНКО

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 випускню кваліфікаційну роботу бакалавра

Студент: Хоменюк.Віталій Юрійович

Тема роботи: «Проект підсистеми моніторингу гетерогенної мережі на основі Nagios Core»

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

Термін здачі студентом закінченої роботи 27 червня 2025 р.

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

вихідні дані для роботи;

Обладнання та програмне забезпечення:

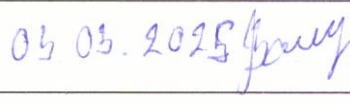
Обладнання:

1. Центральний процесор(CPU)-Rythen5 4600H
2. Оперативна пам'ять (RAM)-16 ГБ DDR4
3. Жорсткий диск (SSD/HDD)-1 Тб(HDD),126 Гб(SSD)
- 4.Відеокарта – AMD Radeon™ Graphics
- 5.Мережева плата – Gigabit Ethernet

Програмне забезпечення

Категорія	Назва / версія	Призначення
Операційна система	Windows 10 Pro	Основна ОС
Система віртуалізації	Oracle VM VirtualBox	Створення віртуального середовища

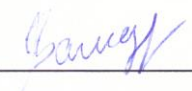
Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

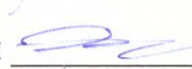
Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1,2,3	Олексій Шелуха	03.03.2025 	03.03.2025 

Студент  Віталій ХОМЕНЮК
Керівник  Юрій РОССІНСЬКИЙ

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	1 тиждень	
2	Аналіз даних	1 тиждень	
3	Формування технічних завдань	1 тиждень	
4	Проектування архітектури	1 тиждень	
5	Розгортання віртуального серидовища	1 тиждень	
6	Інсталяція та налаштування Nagios Core	1 тиждень	
7	Реалізація сповіщення та звітності	1 тиждень	
8	Проведення експериментів	1 тиждень	
9	Анліз результатів	1 тиждень	
10	Написання заключення	1 тиждень	
11	Рецензування та підготовка до захисту	1 тиждень	

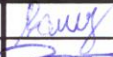
Студент  Віталій ХОМЕНЮК

Керівник  Юрій РОССІНСЬКИЙ

АНОТАЦІЯ

Дипломна робота присвячена проектуванню та практичній реалізації системи моніторингу мережевої інфраструктури на базі платформи Nagios Core. Розроблено методологію створення віртуального стенду для імітації гетерогенної корпоративної мережі з різнотиповими компонентами. Здійснено встановлення та конфігурацію Nagios Core з використанням модульного підходу до організації конфігураційних файлів. Впроваджено комплексний моніторинг різнотипових систем з використанням спеціалізованих агентів NSClient++, NRPE та NCPA. Реалізовано багаторівневий моніторинг критичних сервісів, включаючи поштовий сервер Postfix. Розроблено інтелектуальну систему аналізу мережевої топології через визначення залежностей між компонентами інфраструктури. Налаштовано ефективний механізм сповіщень для оперативного інформування про інциденти. Обґрунтовано переваги впровадженої системи та сформовано рекомендації щодо її подальшого розвитку. Результати роботи можуть бути використані для проектування та впровадження систем моніторингу в мережевих інфраструктурах різного масштабу та призначення.

Ключові слова: моніторинг мережі, Nagios Core, віртуалізація, NSClient++, NRPE, NCPA, поштовий сервер, мережева топологія, система сповіщень, гетерогенна інфраструктура.

					<i>KKIKB.KP.B – 123 – 13 – ПЗ</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Хоменук.В.Ю.		26.06.25	Проект системи моніторингу мережі на базі Nagios Core 4.x з функціональністю для різнотипових хостів та периферії		
Керівник		Росієнський Ю.М.		26.06.25			
Реценз.		Сфіменко А.А.		26.06.25			
Н. контр.		Сфіменко А.А.		26.06.25			
Затверд.		Сфіменко А.А.		26.06.25			
					Лім.	Арк.	Аркушів
						4	109
					ФІКТ, Гр. КІ-21-2		

ABSTRACT

The thesis is devoted to the design and practical implementation of a network infrastructure monitoring system based on the Nagios Core platform. A methodology for creating a virtual stand to simulate a heterogeneous corporate network with different types of components has been developed. Nagios Core was installed and configured using a modular approach to organizing configuration files. Comprehensive monitoring of different types of systems has been implemented using specialized agents: NSClient++, NRPE, and NCPA. Multi-level monitoring of critical services, including the Postfix mail server, has been implemented. An intelligent network topology analysis system has been developed by defining dependencies between infrastructure components. An effective notification mechanism has been configured for prompt incident reporting. The advantages of the implemented system are substantiated and recommendations for its further development are provided. The results of the work can be used for the design and implementation of monitoring systems in network infrastructures of various scales and purposes.

Keywords: network monitoring, Nagios Core, virtualization, NSClient++, NRPE, NCPA, mail server, network topology, notification system, heterogeneous infrastructure.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росі́нський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

DHCP – Dynamic Host Configuration Protocol (Протокол динамічної конфігурації хостів)

DNS – Domain Name System (Система доменних імен)

FQDN – Fully Qualified Domain Name (Повністю кваліфіковане доменне ім'я)

HTTP – Hypertext Transfer Protocol (Протокол передачі гіпертексту)

ICMP – Internet Control Message Protocol (Протокол контрольних повідомлень Інтернету)

IMAP – Internet Message Access Protocol (Протокол доступу до інтернет-повідомлень)

IP – Internet Protocol (Інтернет-протокол)

IT – Information Technology (Інформаційні технології)

LVM – Logical Volume Manager (Менеджер логічних томів)

MTA – Mail Transfer Agent (Агент передачі пошти)

NAT – Network Address Translation (Трансляція мережевих адрес)

NCPA – Nagios Cross-Platform Agent (Крос-платформний агент Nagios)

NRPE – Nagios Remote Plugin Executor (Виконавець віддалених плагінів Nagios)

NSClient++ – Nagios Service Client (Сервісний клієнт Nagios)

POP3 – Post Office Protocol version 3 (Протокол поштового відділення версії 3)

SASL – Simple Authentication and Security Layer (Простий шар автентифікації та безпеки)

SMTP – Simple Mail Transfer Protocol (Простий протокол передачі пошти)

SNMP – Simple Network Management Protocol (Простий протокол управління мережею)

SSH – Secure Shell (Безпечна оболонка)

SSL/TLS – Secure Sockets Layer/Transport Layer Security (Рівень захищених сокетів/Безпека транспортного рівня)

vCPU – Virtual Central Processing Unit (Віртуальний центральний процесор)

YAML – YAML Ain't Markup Language (YAML — не мова розмітки)

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ЗМІСТ

Вступ.....	8
Розділ 1. Проектування та Розгортання Тестового Середовища	10
1.1. Архітектура та IP-адресація Віртуальної Мережі	10
1.2. Інсталяція та Початкова Конфігурація Хостів.....	15
Розділ 2: Впровадження Системи Моніторингу Nagios Core.....	29
2.1. Встановлення Nagios Core та Основних Компонентів.....	29
2.2. Додавання Хостів та Базовий Моніторинг	46
Розділ 3. Налаштування Спеціалізованого Моніторингу та Звітності	75
3.1. Моніторинг Поштового Сервера Postfix	75
3.2. Моніторинг Віртуального Комутатора та Топології Мережі.....	86
3.3. Конфігурація Системи Сповіщень	92
Висновки	103
Список використаних джерел	106
Додатки.....	107

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Вступ

У сучасному світі інформаційні системи стають дедалі складнішими та включають велику кількість розподілених компонентів. Це створює підвищені вимоги до стабільної роботи мережевої інфраструктури. Навіть нетривалі збої в роботі окремих вузлів можуть спричинити значні фінансові втрати, порушення бізнес-процесів і втрату критичних даних. За наявними аналітичними звітами, одна година простою ключових ІТ-систем може коштувати організаціям десятки тисяч або навіть мільйони доларів залежно від галузі та масштабу. У зв'язку з цим актуальним є впровадження ефективних систем моніторингу, здатних заздалегідь виявляти загрози ще до того, як вони спричинять критичні порушення.

Системи моніторингу є невід'ємною складовою управління ІТ-інфраструктурою, забезпечуючи безперервний збір, аналіз і візуалізацію даних про стан мережевих ресурсів у реальному часі. Завдяки проактивному підходу до виявлення збоїв вдається скоротити час реагування, підвищити ефективність використання технічних ресурсів та загальну надійність системи. У протилежному випадку, за відсутності моніторингу, організація змушена діяти реактивно — реагувати лише після настання інцидентів, що вже вплинули на кінцевих користувачів.

Nagios Core є потужною платформою з відкритим кодом, яка забезпечує гнучке конфігурування та адаптацію до особливостей різних ІТ-інфраструктур. Завдяки широкому набору плагінів, ця система дозволяє контролювати широкий спектр об'єктів: від серверів і маршрутизаторів до спеціалізованих сервісів і бізнес-додатків. Вбудовані механізми раннього виявлення проблем, автоматичного інформування та ескалації інцидентів сприяють надійній роботі інфраструктури.

Метою цієї роботи є проектування та реалізація системи моніторингу мережевої інфраструктури на базі Nagios Core із застосуванням сучасних методів

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

забезпечення стабільності та захисту ІТ-систем. Практична частина проєкту передбачає створення тестового середовища з моделюванням типової корпоративної мережі, налаштування компонентів моніторингу та організацію ефективного контролю за всіма критичними елементами.

Для досягнення цієї мети необхідно виконати такі завдання:

- Розробити та змоделювати віртуальне середовище, що відтворює корпоративну мережеву інфраструктуру.
- Інсталювати та налаштувати базову конфігурацію системи Nagios Core.
- Реалізувати моніторинг ключових сервісів, зокрема поштового сервера.
- Налаштувати систему сповіщень для оперативного реагування на відмови та загрози.
- Впровадити моніторинг топології мережі з можливістю аналізу причин збоїв.
- Оцінити ефективність розгорнутої системи та надати рекомендації щодо її вдосконалення.

Практична цінність даної роботи полягає у створенні гнучкої методики впровадження системи моніторингу, що може бути адаптована під конкретні потреби різних організацій. Це сприятиме підвищенню стабільності ІТ-середовища, зниженню ризиків простою та більш ефективному управлінню інформаційними ресурсами. Отримані результати можуть стати основою для формування стратегій моніторингу, пристосованих до специфіки конкретних галузей і технологічних платформ.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

Розділ 1. Проектування та Розгортання Тестового Середовища

1.1. Архітектура та IP-адресація Віртуальної Мережі

Проектування та розгортання тестового середовища становить фундаментальну основу для реалізації системи моніторингу мережевої інфраструктури. Практична імплементація першого етапу включає створення контрольованої інфраструктури, яка дозволить провести всебічне тестування функціональних можливостей Nagios Core без впливу на робочі системи. Методологічний підхід до формування віртуального стенда відображає рівень компетенції в галузі мережевих технологій та системного адміністрування. Кожен етап проектування потребує ретельного аналізу та обґрунтування прийнятих рішень, оскільки архітектурні помилки на початковому етапі призводять до експоненціального зростання складності в подальшій імплементації. Методичність та послідовність дій забезпечують створення стабільного середовища для коректного функціонування всіх компонентів системи моніторингу від налаштування мережевої інфраструктури до конфігурації прикладного програмного забезпечення на віртуальних машинах.

Віртуалізація корпоративного середовища вимагає створення ізольованого мережевого простору з можливістю повного контролю над усіма параметрами. Гіпервізор Oracle VirtualBox надає широкий спектр можливостей для конфігурації віртуальних мереж різного рівня складності, забезпечуючи при цьому необхідний рівень ізоляції від фізичної інфраструктури. Віртуальний стенд, побудований на основі програмної віртуалізації, дозволяє експериментувати з різними конфігураціями системи моніторингу, здійснювати стрес-тестування та моделювати потенційні проблеми без негативного впливу на продуктивні системи.

Для створення віртуальної мережевої інфраструктури обрано режим NAT Network (Network Address Translation), який пропонує оптимальний баланс між ізоляцією та доступністю. Функціональні можливості віртуального

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

маршрутизатора, який автоматично створюється при конфігурації NAT Network, дозволяють організувати приватну підмережу з повною взаємною видимістю всіх віртуальних машин. Віртуальний маршрутизатор здійснює трансляцію мережевих адрес, забезпечуючи доступ віртуальних машин до зовнішніх ресурсів через мережевий інтерфейс хостової системи. Подібна конфігурація має суттєві переваги над іншими режимами віртуальних мереж, які пропонує VirtualBox. На відміну від режиму NAT, який забезпечує ізольований доступ до Інтернету кожній віртуальній машині окремо без можливості комунікації між ними, режим NAT Network створює повноцінне віртуальне середовище з можливістю міжхостової взаємодії. Режим Bridge networking, який підключає віртуальні машини безпосередньо до фізичної мережі, не відповідає вимогам ізоляції тестового середовища від продуктивної інфраструктури.

Створення віртуальної мережі у середовищі Oracle VirtualBox реалізується через інтерфейс мережевого менеджера. Процедура налаштування починається з вибору пункту «File» в головному меню гіпервізора, після чого обирається опція «Network Manager». У вікні, що відкривається, необхідно перейти на вкладку «NAT Networks» та натиснути кнопку «Create». За замовчуванням створюється мережа з іменем «NatNetwork» та адресним простором 10.0.2.0/24, які можуть бути змінені згідно з вимогами проекту.

Віртуальна мережа потребує детального планування адресного простору для забезпечення ефективного розподілу IP-адрес між серверними системами та клієнтськими пристроями. Відповідно до технічного завдання, для віртуальної інфраструктури виділено підмережу 192.168.13.0/26, яка належить до приватного адресного простору, визначеного стандартом RFC 1918. Маска підмережі /26 (255.255.255.192) визначає розмір адресного простору та обмежує кількість доступних IP-адрес.

Математичний розрахунок параметрів підмережі 192.168.13.0/26 демонструє наступні характеристики:

- Адреса мережі: 192.168.13.0
- Маска підмережі: 255.255.255.192

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

- Кількість бітів маски: 26
- Кількість бітів хостової частини: 6
- Максимальна кількість хостів: $2^6 - 2 = 62$
- Перша доступна адреса хоста: 192.168.13.1
- Остання доступна адреса хоста: 192.168.13.62
- Широкомовна адреса: 192.168.13.63

Перша доступна адреса в підмережі (192.168.13.1) автоматично резервується гіпервізором VirtualBox для віртуального шлюзу, який виконує функції маршрутизатора та DNS-форвардера. Віртуальний шлюз забезпечує взаємодію між внутрішньою мережею та зовнішніми ресурсами, що критично важливо для завантаження програмного забезпечення, оновлення операційних систем та інших завдань, які потребують доступу до Інтернету.

Для організації структурованого та передбачуваного розподілу мережесих адрес в межах виділеного адресного простору запропоновано логічне розділення на дві функціональні зони. Перша зона охоплює діапазон адрес від 192.168.13.2 до 192.168.13.31 та призначена для статичних призначень. Серверні системи, мережеве обладнання та інші критичні компоненти інфраструктури, які потребують стабільної адресації, отримують IP-адреси саме з цього діапазону. Статична адресація спрощує процес налаштування системи моніторингу, оскільки відпадає необхідність відслідковувати зміни адрес при кожному перезавантаженні віртуальних машин.

Таблиця 1.1 - Розподіл статичних IP-адрес між серверними компонентами віртуальної інфраструктури

IP-адреса	Призначення	Операційна система
192.168.13.2	Сервер моніторингу (Nagios Core)	Ubuntu Server 22.04 LTS
192.168.13.3	Контролер домену та DNS-сервер	Windows Server 2022
192.168.13.4	Сервер баз даних	Ubuntu Server 22.04 LTS
192.168.13.5	Веб-сервер	Ubuntu Server 22.04 LTS
192.168.13.6	Файловий сервер	Windows Server 2022
192.168.13.7	Поштовий сервер	Ubuntu Server 22.04 LTS
192.168.13.8	Сервер резервного копіювання	Ubuntu Server 22.04 LTS
192.168.13.9	Сервер аналізу логів	Ubuntu Server 22.04 LTS
192.168.13.10	Маршрутизатор	VyOS 1.4

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Друга зона охоплює діапазон адрес від 192.168.13.32 до 192.168.13.62 та призначена для динамічного розподілу через протокол DHCP (Dynamic Host Configuration Protocol). Клієнтські робочі станції, тимчасові пристрої та системи, які не потребують фіксованої адресації, отримують IP-адреси автоматично при підключенні до мережі. DHCP-сервер, розгорнутий на Windows Server 2022, керує процесом розподілу адрес, забезпечуючи ефективне використання адресного простору та спрощуючи процес підключення нових пристроїв.

Конфігурація DHCP-серверу передбачає створення області (scope) з діапазоном адрес від 192.168.13.32 до 192.168.13.62 та налаштування додаткових параметрів:

- Маска підмережі: 255.255.255.192
- Адреса шлюзу за замовчуванням: 192.168.13.1
- DNS-сервери: 192.168.13.3 (основний), 192.168.13.1 (альтернативний)
- Час оренди адреси: 7 днів (для тестового середовища)

Схематичне представлення архітектури мережі наведено на рисунку 1.1, який відображає логічну організацію віртуальної інфраструктури, включаючи сервери, робочі станції та мережеві з'єднання.

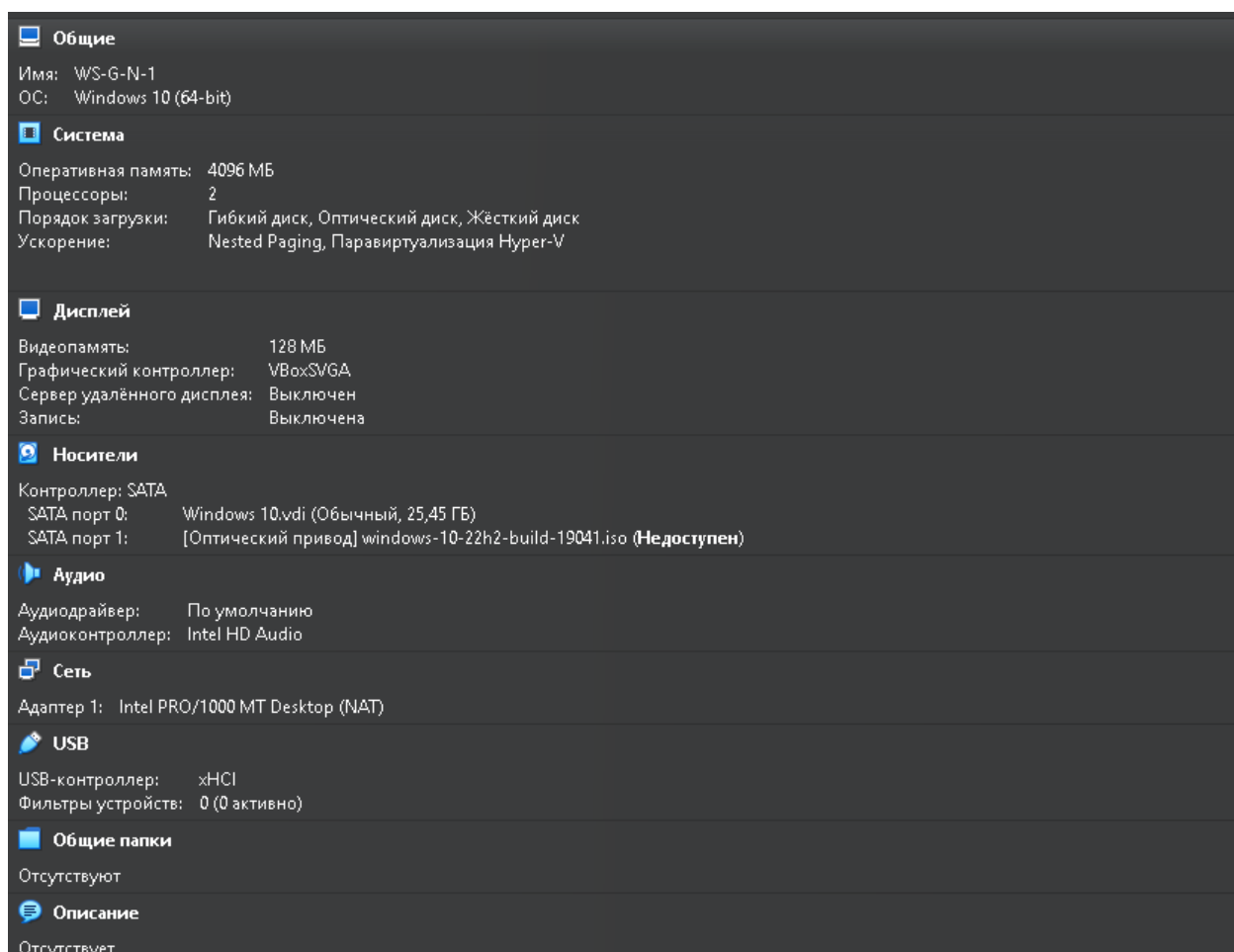
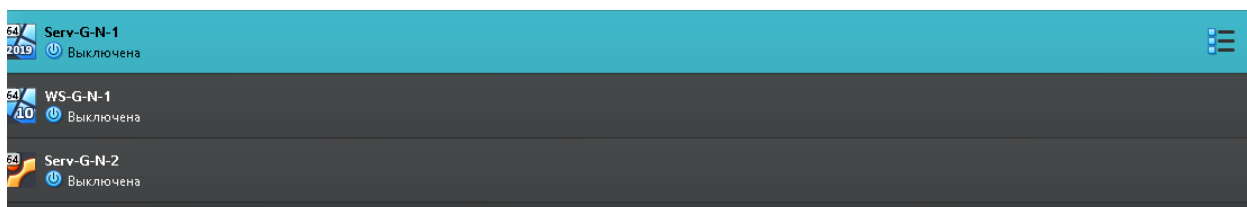
Запропонована архітектура віртуальної мережі та схема IP-адресації забезпечують необхідну гнучкість для розгортання та тестування системи моніторингу на базі Nagios Core. Розділення адресного простору на статичну та динамічну зони спрощує управління мережевою інфраструктурою та забезпечує передбачуваність при налаштуванні моніторингу. Віртуальне середовище, реалізоване за допомогою режиму NAT Network у VirtualBox, надає можливість повного контролю над усіма аспектами мережевої взаємодії, включаючи маршрутизацію, фільтрацію та перенаправлення портів.

Планування мережевої інфраструктури з використанням підмережі 192.168.13.0/26 демонструє розуміння принципів організації IP-мереж та практичні навички роботи з адресним простором обмеженого розміру. Детальний

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

розрахунок параметрів підмережі та обґрунтування розподілу адрес забезпечує технічну основу для розгортання всіх компонентів системи моніторингу в контрольованому віртуальному середовищі.

Після завершення налаштування мережевої інфраструктури наступним етапом стане встановлення та базова конфігурація операційних систем на віртуальних машинах, що буде розглянуто в наступному підрозділі. Кожна віртуальна машина отримає відповідну IP-адресу згідно з розробленою схемою адресації, що забезпечить підготовку платформи для подальшого розгортання Nagios Core та супутніх компонентів системи моніторингу.



		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

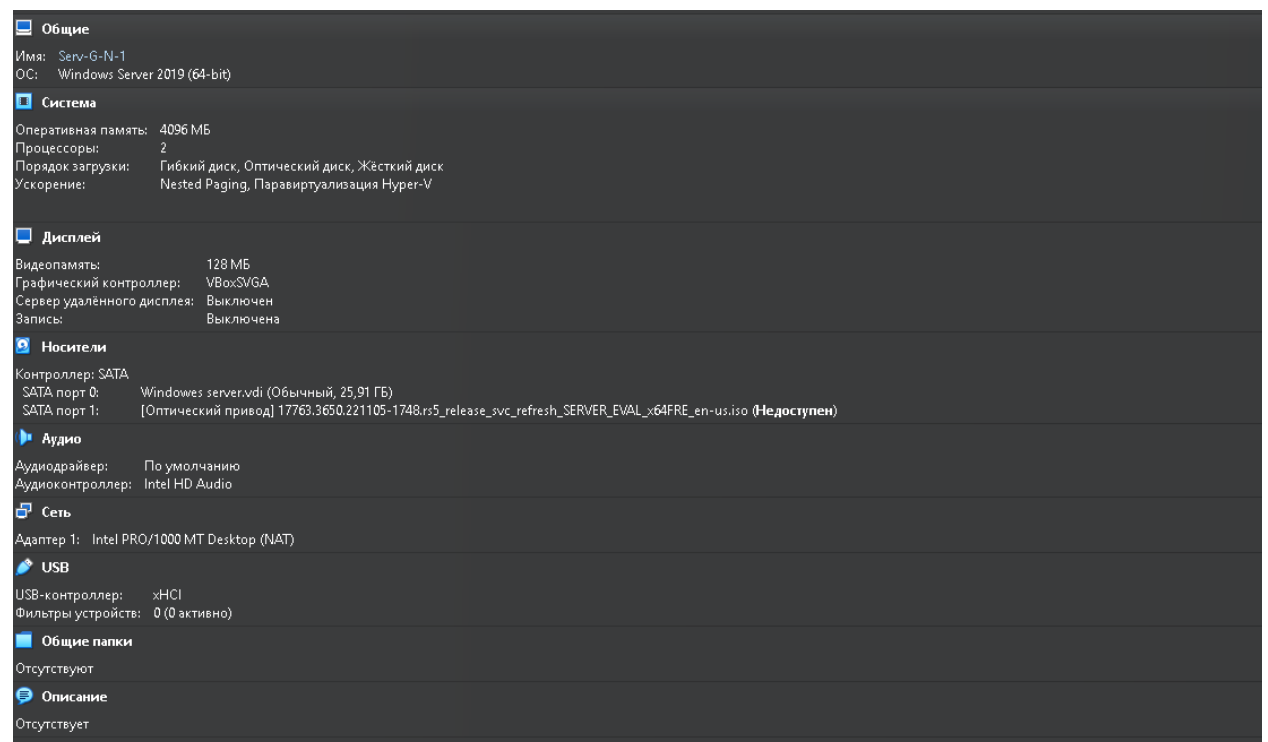
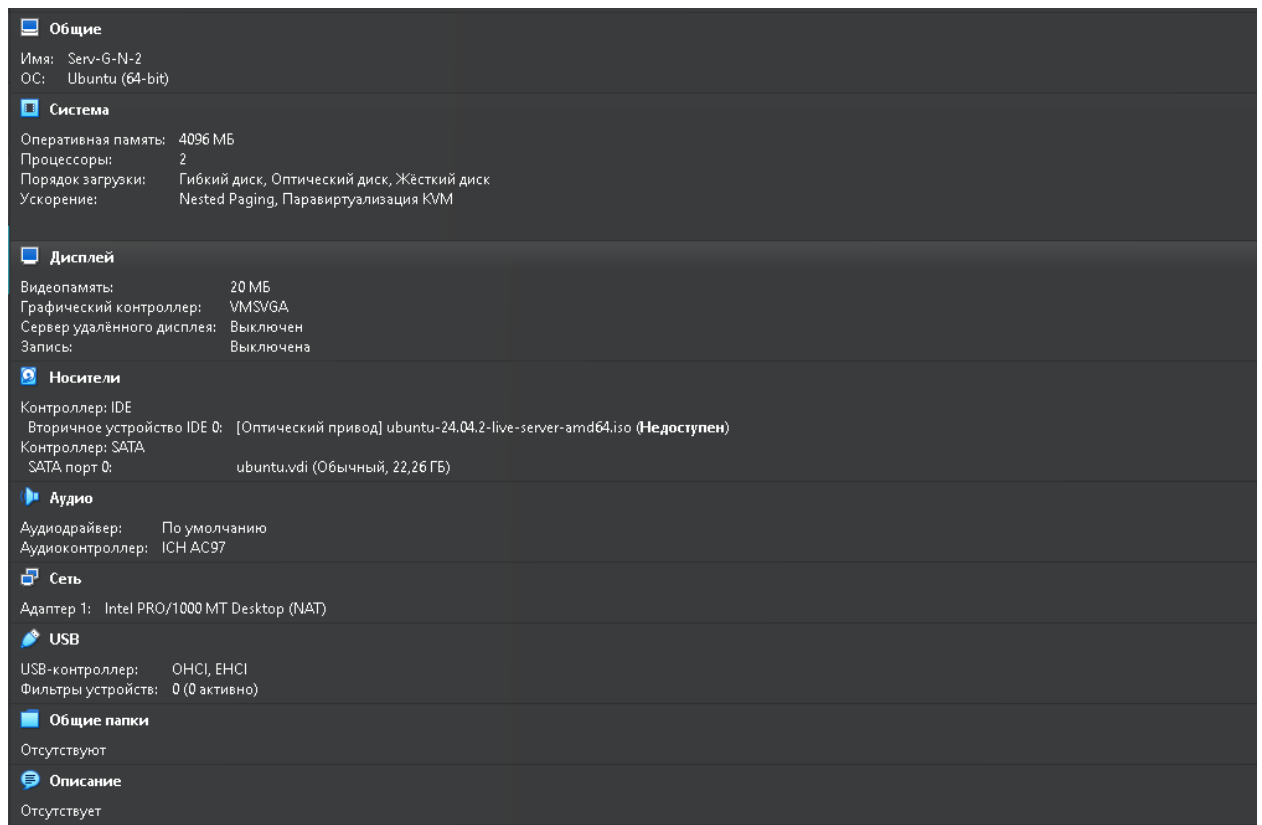


Рисунок 1.2. Параметри віртуальних машин у середовищі Oracle VirtualBox

1.2. Інсталяція та Початкова Конфігурація Хостів

Наступним етапом після проектування мережевої інфраструктури виступає розгортання віртуальних машин та їх базове налаштування. Методологія

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

створення віртуального середовища передбачає застосування сучасних підходів до автоматизації процесів інсталяції та конфігурації, використання командного рядка замість графічного інтерфейсу та документування кожного кроку для забезпечення відтворюваності результатів. Впровадження принципів "Infrastructure as Code" дозволяє значно прискорити процес розгортання середовища, мінімізувати вплив людського фактора та створити основу для майбутнього масштабування інфраструктури. Віртуальне середовище включає чотири ключові системи, кожна з яких виконує специфічну роль у загальній архітектурі моніторингу.

Процес створення віртуальних машин у Oracle VirtualBox розпочинається з визначення технічних характеристик для кожної системи, враховуючи функціональне призначення та прогнозоване навантаження. Застосування уніфікованого підходу до конфігурації апаратних ресурсів спрощує подальше управління інфраструктурою та забезпечує ефективне використання ресурсів хостової системи. Підбір оптимальних параметрів віртуальних машин базується на аналізі мінімальних системних вимог для програмного забезпечення, яке планується встановлювати, а також враховує потенційне зростання навантаження під час проведення тестів.

Центральним компонентом віртуальної інфраструктури виступає сервер моніторингу на базі Ubuntu Server 22.04 LTS, який забезпечуватиме функціонування Nagios Core. Вибір серверної версії Ubuntu обґрунтовується кількома факторами: тривалою підтримкою (Long Term Support) протягом п'яти років, високою стабільністю, оптимізованим використанням системних ресурсів через відсутність графічного інтерфейсу та широкою підтримкою спільноти. Ubuntu Server 22.04 LTS, випущений у квітні 2022 року, включає ядро Linux 5.15, яке забезпечує підтримку сучасного обладнання та містить вдосконалення в сфері безпеки та продуктивності.

Процес створення віртуальної машини починається з визначення технічних характеристик. Для серверу моніторингу оптимальною конфігурацією є: 2 віртуальних процесорних ядра (vCPU), 2 гігабайти оперативної пам'яті (RAM) та

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

25 гігабайт дискового простору. Така конфігурація забезпечує достатній запас продуктивності для функціонування Nagios Core та супутніх компонентів, включаючи веб-інтерфейс, бази даних та додаткові модулі. Виділення 2 vCPU дозволяє системі ефективно розподіляти навантаження між процесами моніторингу та іншими системними задачами, що особливо важливо при обробці даних від багатьох контрольованих хостів.

Створення віртуальної машини через інтерфейс VirtualBox виконується наступним чином: на головному екрані гіпервізора необхідно натиснути кнопку «New» або використати комбінацію клавіш Ctrl+N. У діалоговому вікні вказується ім'я віртуальної машини «Serv-Nagios», тип операційної системи «Linux» та версія «Ubuntu (64-bit)». На наступних етапах задаються параметри оперативної пам'яті (2048 MB), кількість процесорних ядер (2) та створюється віртуальний жорсткий диск типу VDI (VirtualBox Disk Image) з динамічним розподілом місця та розміром 25 GB. Після створення віртуальної машини необхідно налаштувати мережевий адаптер, вибравши режим «NAT Network» та вказавши ім'я віртуальної мережі, створеної на попередньому етапі.

Інсталяція Ubuntu Server 22.04 LTS виконується з ISO-образу, який можна завантажити з офіційного сайту Ubuntu (<https://ubuntu.com/download/server>). Процес встановлення розпочинається з вибору мови інтерфейсу (англійська для спрощення пошуку інформації в майбутньому), розкладки клавіатури та типу інсталяції. Оптимальним варіантом є вибір «Ubuntu Server» з мінімальним набором компонентів для зменшення поверхні атаки та оптимізації використання ресурсів. Під час розбиття диску рекомендується використовувати схему «Use Entire Disk» з LVM (Logical Volume Manager), що забезпечує гнучкість при подальшому керуванні дисковим простором. Впровадження LVM дозволяє динамічно змінювати розміри розділів та додавати нові диски без необхідності перерозбиття існуючих.

Важливим етапом інсталяції є налаштування облікового запису адміністратора. Рекомендується створити користувача з ім'ям, відмінним від стандартних (наприклад, «nagiosadmin» замість «admin»), та використовувати

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

складний пароль, який відповідає сучасним вимогам безпеки: довжина не менше 12 символів, включення символів різних регістрів, цифр та спеціальних символів. Для підвищення безпеки системи, доступ за паролем через SSH буде вимкнено на пізніших етапах, з переходом на автентифікацію за ключами.

Під час інсталяції критично важливо активувати опцію встановлення OpenSSH-сервера, яка доступна на екрані вибору додаткових компонентів. OpenSSH забезпечить віддалений доступ до сервера через захищений канал, що спростить подальше адміністрування системи. Інші компоненти, такі як Snaps або додаткові драйвери, можна пропустити на етапі інсталяції, оскільки вони можуть бути встановлені пізніше за необхідності.

По завершенню інсталяції та першого завантаження системи необхідно провести базову конфігурацію мережевих параметрів. Ubuntu Server 22.04 LTS використовує систему Netplan для налаштування мережі, яка базується на YAML-файлах конфігурації. Для налаштування статичної IP-адреси необхідно створити файл конфігурації `/etc/netplan/01-netcfg.yaml` з наступним вмістом:

```
yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.13.2/26]
      routes:
        - to: default
          via: 192.168.13.1
      nameservers:
        addresses: [192.168.13.3, 192.168.13.1]
```

Наведена конфігурація визначає статичну IP-адресу 192.168.13.2 для мережевого інтерфейсу enp0s3 (ім'я може відрізнятися залежно від версії VirtualBox та налаштувань віртуальної машини), встановлює маршрут за замовчуванням через шлюз 192.168.13.1 та налаштовує DNS-сервери. Застосування конфігурації виконується командою `sudo netplan apply`, після чого необхідно перевірити коректність налаштувань за допомогою команд `ip addr show` та `ping -c 4 8.8.8.8` для перевірки підключення до Інтернету.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

Після налаштування мережі необхідно виконати повне оновлення системи, щоб встановити останні патчі безпеки та оновлення пакетів. Процес оновлення виконується за допомогою наступних команд:

```
bash
sudo apt update
sudo apt upgrade -y
sudo apt dist-upgrade -y
sudo apt autoremove -y
sudo apt clean
```

Послідовність команд оновлює індекс пакетів, встановлює доступні оновлення, видаляє непотрібні пакети та очищає локальний кеш завантажених пакетів. Параметр -у автоматично підтверджує встановлення без додаткових запитів, що важливо при автоматизації процесу.

Додатково рекомендується встановити базовий набір утиліт, які спростять подальше адміністрування системи:

```
bash
sudo apt install -y vim htop iftop net-tools curl wget unzip git
```

Встановлені інструменти включають текстовий редактор vim, утиліти моніторингу системи htop та iftop, стандартні мережеві інструменти net-tools (ifconfig, netstat, тощо), інструменти для завантаження файлів curl та wget, а також систему контролю версій git, яка буде використовуватися для завантаження додаткових модулів Nagios.

Для підвищення безпеки системи рекомендується налаштувати брандмауер UFW (Uncomplicated Firewall), який входить до складу Ubuntu Server:

```
bash
sudo ufw allow OpenSSH
sudo ufw allow 80/tcp
sudo ufw enable
sudo ufw status
```

Наведені команди дозволяють вхідний трафік для SSH (порт 22) та HTTP (порт 80), активують брандмауер та відображають його поточний статус. Порт 80 буде використовуватися для доступу до веб-інтерфейсу Nagios Core, який буде встановлено на наступних етапах.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Налаштування часового поясу та синхронізації часу є критично важливим для системи моніторингу, оскільки коректна часова позначка подій забезпечує точність аналізу та звітності:

```
bash
sudo timedatectl set-timezone Europe/Kiev
sudo apt install -y chrony
sudo systemctl enable chrony
sudo systemctl start chrony
```

Chrony забезпечує більш точну синхронізацію часу порівняно з традиційним ntpd, особливо в віртуалізованих середовищах, де системний годинник може дрейфувати.

Другим компонентом віртуальної інфраструктури є контролер домену на базі Windows Server 2022, який забезпечуватиме централізоване управління обліковими записами, групову політику та DNS-сервіс для всього віртуального середовища. Windows Server 2022, випущений у вересні 2021 року, представляє найновішу версію серверної операційної системи Microsoft з розширеними можливостями в сфері безпеки, віртуалізації та управління.

Створення віртуальної машини для контролера домену виконується аналогічно процесу для сервера моніторингу, з деякими відмінностями в апаратній конфігурації. Для Windows Server 2022 рекомендується наступна конфігурація: 2 vCPU, 4 GB RAM та 50 GB дискового простору. Збільшення обсягу оперативної пам'яті обґрунтовується вищими вимогами Windows Server до системних ресурсів, особливо при роботі служб Active Directory та DNS. Більший розмір віртуального диску необхідний для розміщення файлів операційної системи, бази даних Active Directory та файлів журналів.

Інсталяція Windows Server 2022 виконується з ISO-образу, який можна отримати через програму Microsoft Evaluation Center для тестових та навчальних цілей. Під час інсталяції рекомендується вибрати варіант «Windows Server 2022 Standard Evaluation (Desktop Experience)», який включає графічний інтерфейс користувача, що спрощує початкове налаштування. На етапі розбиття диску оптимальним варіантом є використання всього доступного простору для системного розділу.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

Після завершення інсталяції та першого входу в систему необхідно налаштувати мережеві параметри. Налаштування статичної IP-адреси виконується через Панель керування -> Мережа та Інтернет -> Центр управління мережами та спільним доступом -> Зміна параметрів адаптера. Для мережевого адаптера налаштовуються наступні параметри:

- IP-адреса: 192.168.13.3
- Маска підмережі: 255.255.255.192
- Шлюз за замовчуванням: 192.168.13.1
- Переважний DNS-сервер: 127.0.0.1 (localhost, оскільки сервер буде виконувати роль DNS)
- Альтернативний DNS-сервер: 192.168.13.1

Після налаштування мережі рекомендується змінити ім'я комп'ютера на більш описове (наприклад, «Serv-Domain») через Панель керування -> Система та безпека -> Система -> Змінити параметри -> Змінити. Зміна імені комп'ютера вимагає перезавантаження системи.

Наступним кроком є встановлення ролей Active Directory Domain Services та DNS Server, які необхідні для функціонування контролера домену:

1. Відкрийте Server Manager (Диспетчер сервера)
2. Виберіть «Add roles and features» (Додати ролі та компоненти)
3. Перейдіть до розділу «Server Roles» (Ролі сервера)
4. Встановіть прапорці біля «Active Directory Domain Services» та «DNS Server»
5. Додайте необхідні компоненти за запитом системи
6. Завершіть інсталяцію та перезавантажте сервер за необхідності

Після встановлення ролей необхідно підвищити сервер до рівня контролера домену, створивши новий ліс Active Directory:

1. Відкрийте Server Manager
2. У верхньому правому куті буде повідомлення про необхідність конфігурації AD DS
3. Натисніть на повідомлення та виберіть «Promote this server to a domain controller» (Підвищити цей сервер до контролера домену)

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

4. Виберіть опцію «Add a new forest» (Додати новий ліс)
5. Введіть ім'я кореневого домену (наприклад, «nagioslab.local»)
6. Встановіть рівень функціональності домену та лісу як «Windows Server 2016»
7. Введіть пароль відновлення Directory Services Restore Mode (DSRM)
8. На наступних екранах використовуйте налаштування за замовчуванням
9. Перевірте конфігурацію та натисніть «Install» (Встановити)

Процес підвищення до контролера домену включає перевірку передумов, конфігурацію служб Active Directory та DNS, а також створення початкової бази даних Active Directory. По завершенню процесу система буде автоматично перезавантажена.

Після перезавантаження та входу в систему з обліковим записом адміністратора домену необхідно налаштувати DHCP-сервер для автоматичного розподілу IP-адрес клієнтським машинам:

- 1 Відкрийте Server Manager
- 2 Виберіть «Add roles and features»
- 3 Встановіть роль «DHCP Server»
- 4 Завершіть інсталяцію та перезавантажте сервер за необхідності
- 5 Після перезавантаження відкрийте «DHCP Manager» через Server Manager - > Tools
- 6 Розгорніть вузол сервера та створіть нову область (scope):
- 7 Ім'я: NagiosLab DHCP Scope
- 8 Діапазон IP-адрес: від 192.168.13.32 до 192.168.13.62
- 9 Маска підмережі: 255.255.255.192
- 10 Виключення: немає
- 11 Термін оренди: 7 днів
- 12 Параметри області:
- 13 Шлюз за замовчуванням: 192.168.13.1
- 14 DNS-сервери: 192.168.13.3, 192.168.13.1
- 15 Домен DNS: nagioslab.local

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Після створення області необхідно авторизувати DHCP-сервер в Active Directory, щоб запобігти конфліктам з іншими DHCP-серверами:

- 1 В DHCP Manager клацніть правою кнопкою миші на вузол сервера
- 2 Виберіть «Authorize» (Авторизувати)
- 3 Перезапустіть службу DHCP через Server Manager -> Tools -> Services

Таблиця 1.2 - Параметри DHCP-області для віртуальної мережі

Параметр	Значення
Ім'я області	NagiosLab DHCP Scope
Діапазон IP-адрес	192.168.13.32 - 192.168.13.62
Маска підмережі	255.255.255.192
Шлюз за замовчуванням	192.168.13.1
DNS-сервери	192.168.13.3, 192.168.13.1
Домен DNS	nagioslab.local
Термін оренди	7 днів

Завершальним етапом базової конфігурації контролера домену є налаштування політик безпеки та створення організаційної структури.

Створення організаційної структури Active Directory забезпечує основу для подальшого впровадження групових політик та управління доступом до ресурсів.

Третім компонентом віртуальної інфраструктури є клієнтська робоча станція на базі Windows 10 Pro, яка представлятиме типового користувача мережі та буде об'єктом моніторингу з боку Nagios Core. Windows 10 Pro обрано через широке розповсюдження в корпоративних середовищах та розширені можливості інтеграції з доменом Active Directory.

Для клієнтської робочої станції рекомендується наступна конфігурація: 2 vCPU, 4 GB RAM та 50 GB дискового простору. Параметри віртуальної машини відповідають рекомендованим вимогам для Windows 10 та забезпечують комфортну роботу користувача.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

Інсталяція Windows 10 Pro виконується з ISO-образу, який можна отримати через програму Microsoft Evaluation Center або використовувати наявну ліцензійну копію. Під час інсталяції рекомендується вибрати варіант «Вибіркова інсталяція» для більш точного контролю над процесом. На етапі розбиття диску оптимальним варіантом є використання всього доступного простору для системного розділу.

Після завершення інсталяції та першого входу в систему необхідно виконати початкове налаштування Windows 10, включаючи створення локального облікового запису (наприклад, «TestUser») та налаштування мережових параметрів. На відміну від серверів, для клієнтської робочої станції рекомендується використовувати динамічну IP-адресацію через DHCP, що відповідає типовій конфігурації в корпоративних мережах.

Для приєднання робочої станції до домену Active Directory виконайте наступні кроки:

1. Відкрийте Панель керування -> Система та безпека -> Система
2. Клацніть на «Змінити параметри» в розділі «Ім'я комп'ютера, домен та параметри робочої групи»
3. Натисніть кнопку «Змінити»
4. Виберіть опцію «Домен» та введіть ім'я домену (nagioslab.local)
5. Введіть облікові дані адміністратора домену за запитом системи
6. Перезавантажте комп'ютер для застосування змін

Після перезавантаження увійдіть в систему, використовуючи обліковий запис домену, та встановіть додаткове програмне забезпечення, яке буде моніторитися за допомогою Nagios Core. Для демонстрації можливостей моніторингу рекомендується встановити веб-сервер IIS (Internet Information Services):

1. Відкрийте Панель керування -> Програми -> Програми та компоненти
2. Клацніть на «Увімкнення або вимкнення компонентів Windows»
3. Встановіть прапорець біля «Internet Information Services» та розгорніть його

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

4. Виберіть базові компоненти, включаючи «Веб-сервер» та «Засоби керування веб-сервером»

5. Натисніть «ОК» для встановлення компонентів

Після встановлення IIS перевірте його функціонування, відкривши веб-браузер та перейшовши за адресою <http://localhost>. Повинна відобразитися стандартна сторінка вітання IIS.

Четвертим компонентом віртуальної інфраструктури є тестовий сервер на базі Ubuntu Desktop 22.04 LTS, який представлятиме альтернативну операційну систему для демонстрації кросплатформених можливостей моніторингу Nagios Core. Вибір десктопної версії Ubuntu обґрунтовується наявністю графічного інтерфейсу, що спрощує деякі операції налаштування та демонстрації.

Для тестового сервера рекомендується наступна конфігурація: 2 vCPU, 4 GB RAM та 25 GB дискового простору. Збільшення обсягу оперативної пам'яті порівняно з сервером моніторингу обґрунтовується наявністю графічного інтерфейсу, який споживає додаткові ресурси.

Інсталяція Ubuntu Desktop 22.04 LTS виконується з ISO-образу, який можна завантажити з офіційного сайту Ubuntu. Процес встановлення схожий на інсталяцію серверної версії, з додатковим вибором компонентів графічного інтерфейсу. Під час інсталяції рекомендується вибрати мінімальний набір додаткових компонентів для оптимізації використання ресурсів.

Після завершення інсталяції та першого входу в систему необхідно налаштувати статичну IP-адресу. В Ubuntu Desktop 22.04 LTS це можна зробити через графічний інтерфейс або використовуючи Netplan, аналогічно до серверної версії:

1. Створіть файл конфігурації `/etc/netplan/01-netcfg.yaml` з наступним вмістом:

```
yaml
network:
  version: 2
  ethernets:
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

```

enp0s3:
  dhcp4: no
  addresses: [192.168.13.4/26]
  routes:
    - to: default
      via: 192.168.13.1
  nameservers:
    addresses: [192.168.13.3, 192.168.13.1]

```

2. Застосуйте конфігурацію командою `sudo netplan apply`

Після налаштування мережі виконайте повне оновлення системи та встановіть додаткове програмне забезпечення, яке буде моніторитися за допомогою Nagios Core:

```

bash
sudo apt update
sudo apt upgrade -y
sudo apt install -y apache2 mysql-server php libapache2-mod-php

```

Встановлені пакети включають веб-сервер Apache, сервер баз даних MySQL та інтерпретатор PHP, що створює типове середовище для веб-додатків.

Під час встановлення phpMyAdmin виберіть веб-сервер Apache2 та налаштуйте базу даних відповідно до запитів інсталятора.

Для демонстрації можливостей моніторингу мережевих сервісів рекомендується також встановити FTP-сервер:

```

bash
sudo apt install -y vsftpd
sudo systemctl enable vsftpd
sudo systemctl start vsftpd

```

Після встановлення необхідно налаштувати базову конфігурацію vsftpd, відредагувавши файл `/etc/vsftpd.conf`:

```

bash
sudo nano /etc/vsftpd.conf

```

Змініть наступні параметри:

```

anonymous_enable=NO
local_enable=YES

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

```

write_enable=YES
chroot_local_user=YES
Перезапустіть службу для застосування змін:
bash
sudo systemctl restart vsftpd

```

Налаштування брандмауера є важливим аспектом безпеки, але для тестового середовища необхідно відкрити порти для сервісів, які будуть моніторитися:

```

bash
sudo ufw allow OpenSSH
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 3306/tcp
sudo ufw allow 21/tcp
sudo ufw enable
sudo ufw status

```

Наведені команди дозволяють вхідний трафік для SSH (порт 22), HTTP (порт 80), HTTPS (порт 443), MySQL (порт 3306) та FTP (порт 21).

Таблиця 1.3 - Зведена інформація про віртуальні машини в тестовому середовищі

Назва ВМ	ОС	IP-адреса	vCPU	RAM	Диск	Призначення
Serv- Nagios	Ubuntu Server 22.04	192.168.13.2	2	2 GB	25 GB	Сервер моніторингу
Serv- Domain	Windows Server 2022	192.168.13.3	2	4 GB	50 GB	Контролер домену та DNS
Serv- Linux	Ubuntu Desktop 22.04	192.168.13.4	2	4 GB	25 GB	Тестовий сервер
Client- Win	Windows 10 Pro	DHCP	2	4 GB	50 GB	Клієнтська робоча станція

Після завершення базової конфігурації всіх компонентів віртуальної інфраструктури необхідно перевірити взаємну видимість систем та доступність ключових сервісів. Для перевірки мережевої зв'язності використовуйте команду `ping` з кожної віртуальної машини до інших компонентів середовища. Для перевірки доступності веб-серверів використовуйте веб-браузер або команду `curl`.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

Ретельне проєктування та початкова конфігурація віртуальних машин створюють надійну основу для подальшого розгортання системи моніторингу на базі Nagios Core. Наступним етапом буде встановлення та конфігурація самого Nagios Core на сервері моніторингу, а також налаштування агентів моніторингу на інших компонентах віртуальної інфраструктури.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Розділ 2: Впровадження Системи Моніторингу Nagios Core

2.1. Встановлення Nagios Core та Основних Компонентів

Імплементація системи моніторингу становить ключовий етап розробки інфраструктури кібербезпеки, що дозволяє забезпечити безперервний контроль стану мережевих ресурсів, серверних систем та критичних сервісів. Nagios Core представляє собою високопродуктивне рішення з відкритим вихідним кодом, яке набуло широкого визнання серед фахівців з інформаційних технологій завдяки модульній архітектурі, гнучким можливостям налаштування та розширеному функціоналу оповіщення. Впровадження Nagios Core дозволяє реалізувати проактивний підхід до управління ІТ-інфраструктурою через ідентифікацію потенційних проблем до їх ескалації в критичні інциденти, що сприяє мінімізації часу простою систем та підвищенню рівня інформаційної безпеки. Перехід від підготовленої віртуальної інфраструктури до функціонуючої системи моніторингу вимагає послідовного виконання комплексу технічних заходів: встановлення базового програмного забезпечення, налаштування модулів розширення, конфігурацію агентів на контрольованих хостах, та імплементацію механізмів сповіщення адміністраторів.

Встановлення Nagios Core представляє собою багатоетапний процес, який вимагає глибокого розуміння принципів функціонування Linux-систем, процесів компіляції програмного забезпечення та механізмів взаємодії компонентів веб-серверів. Реалізація інсталяції на сервері Serv-Nagios під управлінням Ubuntu Server 22.04 LTS забезпечує оптимальну платформу для розгортання моніторингової системи завдяки тривалій підтримці дистрибутиву, високій стабільності та ефективному використанню системних ресурсів. Застосування методології встановлення з вихідних кодів, на відміну від використання пакетів з репозиторіїв, надає повний контроль над конфігурацією компонентів та версіями програмного забезпечення, що є критично важливим для забезпечення максимальної безпеки та функціональної гнучкості системи. Компіляція з

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

вихідних кодів також дозволяє отримати глибше розуміння внутрішньої структури Nagios Core, що спрощує подальше налаштування та адаптацію системи під специфічні вимоги моніторингу.

Архітектура Nagios Core базується на модульному принципі та включає кілька ключових компонентів: ядро системи, яке відповідає за планування та виконання перевірок, плагіни, що реалізують конкретні механізми моніторингу, та веб-інтерфейс для візуалізації стану інфраструктури. Модульність архітектури забезпечує високу масштабованість рішення та можливість інтеграції з різноманітними системами та сервісами. Механізм роботи Nagios Core полягає у періодичному виконанні визначених перевірок стану хостів та сервісів з подальшою обробкою результатів згідно з налаштованою логікою сповіщення. Важливою особливістю Nagios Core є можливість визначення залежностей між сервісами, що дозволяє уникнути каскадних сповіщень при відмові батьківського компонента.

Процес встановлення Nagios Core розпочинається з підготовки середовища шляхом інсталяції необхідних залежностей. Для забезпечення успішної компіляції та функціонування системи моніторингу необхідний широкий спектр програмних компонентів, включаючи компілятори, бібліотеки розробки, веб-сервер та інтерпретатор PHP. Базовий набір залежностей включає build-essential, який надає інструменти для компіляції програмного забезпечення з вихідних кодів, apache2 для забезпечення роботи веб-інтерфейсу, php та його модулі для обробки динамічного вмісту, бібліотеку libgd для генерації графіків, а також додаткові утиліти для роботи з мережевими протоколами та криптографічними функціями.

bash

```
sudo apt install -y wget build-essential apache2 php openssl perl make php-gd libgd-dev libapache2-mod-php libperl-dev libssl-dev daemon autoconf unzip
```

Комплексність набору залежностей відображає багатогранність функціоналу Nagios Core. Пакет build-essential включає компілятори GCC та G++, інструменти автоматизації збірки Make, та базові бібліотеки розробки, необхідні для трансляції вихідного коду в виконуваний бінарний формат. Веб-сервер

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

Apache2 забезпечує обробку HTTP-запитів та візуалізацію інтерфейсу користувача. Інтерпретатор PHP та його модулі надають можливість виконання динамічних сценаріїв, необхідних для генерації інтерактивних елементів інтерфейсу та обробки даних моніторингу. Бібліотека Graphics Draw (libgd) забезпечує функціональність створення графіків та візуалізації метрик у веб-інтерфейсі. Додаткові компоненти, такі як OpenSSL, надають криптографічні функції для захищеної передачі даних, а Perl використовується деякими плагінами моніторингу для обробки складних логічних конструкцій.

Наступним етапом встановлення є створення спеціалізованого користувача та групи для роботи сервісу Nagios. Імплементація принципу найменших привілеїв вимагає виконання процесів моніторингу від імені непривілейованого користувача, що мінімізує потенційний вплив на систему в разі компрометації сервісу. Створення користувача nagios та групи nagcmd, а також додавання до групи користувача веб-сервера www-data забезпечує коректний рівень ізоляції та необхідні права доступу для взаємодії компонентів системи.

```
bash
sudo useradd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd www-data
```

Розділення прав доступу між компонентами системи підвищує загальний рівень безпеки через ізоляцію процесів та мінімізацію поверхні атаки. Користувач nagios не має прав доступу до системних файлів, що обмежує потенційний збиток у разі компрометації сервісу. Додавання користувача веб-сервера www-data до групи nagcmd забезпечує можливість виконання команд перевірки через веб-інтерфейс, зберігаючи при цьому необхідний рівень ізоляції. Застосування групових політик доступу дозволяє гнучко керувати правами компонентів системи без необхідності надання надмірних привілеїв.

Третій етап процесу встановлення включає завантаження, розпакування, компіляцію та інсталяцію основних компонентів Nagios Core та Nagios Plugins.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

Вибір актуальних стабільних версій програмного забезпечення (Nagios Core 4.5.9 та Nagios Plugins 2.4.9) забезпечує оптимальний баланс між функціональністю та стабільністю. Процес компіляції та встановлення Nagios Core включає конфігурацію вихідного коду з відповідними параметрами, збірку бінарних файлів, встановлення компонентів в систему, налаштування автоматичного запуску при завантаженні системи, встановлення відповідних прав доступу та базової конфігурації.

```
bash
```

```
cd /tmp
```

```
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.9.tar.gz
```

```
wget https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
```

```
# Розпакування архівів
```

```
tar -xzf nagios-4.5.9.tar.gz
```

```
tar -xzf nagios-plugins-2.4.9.tar.gz
```

```
# Компіляція та встановлення Nagios Core
```

```
cd nagios-4.5.9
```

```
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

```
sudo make all
```

```
sudo make install
```

```
sudo make install-init
```

```
sudo make install-commandmode
```

```
sudo make install-config
```

```
sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf
```

Процес конфігурації вихідного коду Nagios Core (команда ./configure) виконує аналіз системного середовища, перевіряє наявність необхідних компонентів та генерує файли Makefile з оптимальними параметрами для конкретного оточення. Параметр --with-httpd-conf=/etc/apache2/sites-enabled вказує

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

шлях для встановлення конфігураційного файлу веб-сервера Apache2, що відповідає структурі каталогів в Ubuntu Server 22.04. Команда `make all` виконує власне компіляцію вихідного коду, результатом якої є набір бінарних файлів та бібліотек. Послідовність команд `make install`, `make install-init`, `make install-commandmode` та `make install-config` встановлює скомпільовані компоненти в систему, створює сервіс для автоматичного запуску Nagios, налаштовує права доступу для виконання команд та встановлює базову конфігурацію відповідно. Заклучна команда встановлює конфігураційний файл для веб-сервера Apache2, який забезпечує доступ до інтерфейсу Nagios через HTTP.

Після успішного встановлення Nagios Core необхідно встановити Nagios Plugins, який представляє собою набір скриптів та програм для виконання конкретних перевірок стану хостів та сервісів. Процес встановлення Nagios Plugins аналогічний до встановлення Nagios Core та включає конфігурацію вихідного коду, компіляцію та інсталяцію компонентів.

```
bash
cd /tmp/nagios-plugins-2.4.9
sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios
sudo make
sudo make install
```

Параметри конфігурації `--with-nagios-user=nagios --with-nagios-group=nagios` вказують користувача та групу, від імені яких будуть виконуватися плагіни, забезпечуючи узгодженість прав доступу між компонентами системи. Nagios Plugins включає широкий спектр інструментів для моніторингу різноманітних аспектів ІТ-інфраструктури: перевірка доступності хостів через ICMP, моніторинг мережевих сервісів (HTTP, SMTP, POP3, IMAP, FTP, SSH тощо), контроль системних ресурсів (навантаження процесора, використання пам'яті, дискового простору), моніторинг серверних додатків (баз даних, веб-серверів) та багато іншого. Модульна структура Nagios Plugins дозволяє легко розширювати функціональність системи через додавання нових плагінів або модифікацію існуючих.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Четвертий етап встановлення фокусується на налаштуванні веб-інтерфейсу Nagios, який забезпечує візуалізацію стану інфраструктури та надає інструменти для управління системою моніторингу. Процес налаштування включає активацію необхідних модулів Apache2, створення користувача для доступу до веб-інтерфейсу та перезапуск веб-сервера для застосування змін.

```
bash
```

```
# Вмикаємо необхідні модулі Apache
```

```
sudo a2enmod rewrite cgi
```

```
# Створюємо користувача для веб-доступу
```

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
# Перезапускаємо Apache
```

```
sudo systemctl restart apache2
```

Модуль rewrite забезпечує функціональність перенаправлення URL-запитів, що використовується для реалізації зручних шляхів доступу до різних розділів інтерфейсу. Модуль cgi (Common Gateway Interface) необхідний для виконання CGI-скриптів, які є основою динамічного функціоналу веб-інтерфейсу Nagios. Команда htpasswd створює файл аутентифікації з хешованими паролями для захисту доступу до веб-інтерфейсу. Параметр -c вказує на створення нового файлу, /usr/local/nagios/etc/htpasswd.users визначає шлях до файлу аутентифікації, а nagiosadmin є ім'ям користувача, для якого створюється запис. Після виконання команди система запропонує ввести та підтвердити пароль для користувача nagiosadmin. Завершальний перезапуск Apache2 забезпечує застосування всіх виконаних налаштувань.

П'ятий етап включає перевірку коректності конфігурації та запуск сервісу Nagios. Перед активацією системи моніторингу критично важливо переконатися у відсутності помилок в конфігураційних файлах, оскільки некоректні налаштування можуть призвести до неправильної роботи або повної відмови сервісу.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

```
bash
```

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Утиліта перевірки конфігурації аналізує всі конфігураційні файли, включаючи визначення хостів, сервісів, команд та інших об'єктів, та виводить детальний звіт про знайдені проблеми. Параметр `-v` вказує на режим перевірки, а `/usr/local/nagios/etc/nagios.cfg` є шляхом до головного конфігураційного файлу Nagios. Успішна перевірка (0 помилок, 0 попереджень) є необхідною умовою для запуску сервісу.

```
bash
```

```
sudo systemctl start nagios
```

Команда `systemctl start nagios` активує сервіс Nagios, який починає виконання запланованих перевірок згідно з налаштованою конфігурацією. Для забезпечення автоматичного запуску Nagios при завантаженні системи рекомендується додатково виконати команду `sudo systemctl enable nagios`.

Після успішного запуску сервісу веб-інтерфейс Nagios стає доступним за адресою <http://192.168.13.2/nagios> (або через налаштоване перенаправлення портів). Вхід до інтерфейсу вимагає введення облікових даних користувача `nagiosadmin`, створеного на попередньому етапі. Веб-інтерфейс надає доступ до тактичного огляду стану інфраструктури, детальної інформації про окремі хости та сервіси, журналу подій, сповіщень та інших функцій системи моніторингу.

Для забезпечення ефективного управління конфігурацією Nagios в умовах зростаючої інфраструктури необхідно впровадити структурований підхід до організації конфігураційних файлів. Стандартна інсталяція Nagios використовує монолітні конфігураційні файли, що ускладнює підтримку та масштабування системи при збільшенні кількості контрольованих хостів. Реалізація модульної структури конфігурації дозволяє логічно групувати об'єкти за типами пристроїв або функціональним призначенням, спрощуючи навігацію та мінімізуючи ризик помилок при внесенні змін.

```
bash
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

```
/usr/local/nagios/etc/objects/{windows,linux,workstation,network,templates}
```

Команда створює ієрархію каталогів для різних типів об'єктів моніторингу: windows для серверів під управлінням Windows, linux для Linux-систем, workstation для робочих станцій, network для мережевого обладнання та templates для шаблонів конфігурації. Розділення конфігураційних файлів за типами пристроїв спрощує управління великими інфраструктурами та забезпечує можливість делегування відповідальності різним адміністраторам.

Для активації модульної структури необхідно модифікувати головний конфігураційний файл Nagios, замінивши посилання на окремі файли (директиви `cfg_file`) на директиви сканування каталогів (`cfg_dir`):

```
bash
```

```
# Закоментуйте або видаліть ці рядки:
```

```
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

```
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
```

```
#cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
```

```
# Додайте ці рядки для сканування каталогів:
```

```
cfg_dir=/usr/local/nagios/etc/objects/windows
```

```
cfg_dir=/usr/local/nagios/etc/objects/linux
```

```
cfg_dir=/usr/local/nagios/etc/objects/workstation
```

```
cfg_dir=/usr/local/nagios/etc/objects/network
```

```
cfg_dir=/usr/local/nagios/etc/objects/templates
```

Директива `cfg_dir` вказує Nagios на необхідність сканування всіх файлів з розширенням `.cfg` у вказаному каталозі та включення їх до конфігурації. Такий підхід дозволяє додавати нові об'єкти моніторингу шляхом створення окремих файлів у відповідних каталогах без необхідності модифікації головного конфігураційного файлу. Впровадження модульної структури конфігурації від початку проекту спрощує подальше масштабування системи та знижує ризик помилок при внесенні змін.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

Додатково рекомендується впровадити систему контролю версій для відстеження змін у конфігураційних файлах. Використання Git або іншої системи контролю версій дозволяє документувати зміни, відслідковувати авторів модифікацій та, за необхідності, відкочувати конфігурацію до попереднього стану. Типовий процес управління конфігурацією включає створення локального Git-репозиторію в каталозі /usr/local/nagios/etc, комітування початкового стану конфігурації та регулярне комітування всіх подальших змін з детальними коментарями.

```
bash
cd /usr/local/nagios/etc
git init
git add .
git config --global user.email "admin@example.com"
git config --global user.name "Nagios Administrator"
git commit -m "Initial configuration"
```

Результатом встановлення та початкової конфігурації Nagios Core є функціонуюча система моніторингу з веб-інтерфейсом та структурованою організацією конфігураційних файлів. Впровадження модульної структури конфігурації та системи контролю версій забезпечує основу для подальшого масштабування та розвитку системи моніторингу відповідно до зростаючих потреб інфраструктури. Наступними етапами є налаштування моніторингу хостів та сервісів, впровадження механізмів сповіщення та інтеграція з додатковими системами та сервісами.

Таблиця 2.1 - Огляд основних компонентів Nagios Core та їх призначення

Компонент	Призначення
Ядро Nagios (nagios)	Основний процес, який планує та виконує перевірки, обробляє результати та генерує сповіщення
Плагіни Nagios (plugins)	Набір утиліт для виконання конкретних перевірок (ping, HTTP, SMTP, CPU, memory, тощо)
Веб-інтерфейс	Візуалізація стану інфраструктури, доступ до звітів, конфігурації та інших функцій
CGI-скрипти	Забезпечують динамічну функціональність веб-інтерфейсу
Конфігураційні файли	Визначають хости, сервіси, команди, контакти та інші об'єкти моніторингу

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Командний інтерфейс	Механізм для виконання зовнішніх команд через веб-інтерфейс або зовнішні системи
Механізм сповіщень	Відправлення повідомлень про проблеми через різні канали комунікації (email, SMS, месенджери)

Представлена методологія встановлення Nagios Core з вихідних кодів забезпечує повний контроль над процесом інсталяції та конфігурації, що є важливим аспектом для систем моніторингу в критичних інфраструктурах. Впровадження структурованого підходу до організації конфігураційних файлів та системи контролю версій створює основу для ефективного управління системою моніторингу в довгостроковій перспективі. Подальше налаштування Nagios Core для моніторингу конкретних хостів та сервісів буде розглянуто в наступних підрозділах.

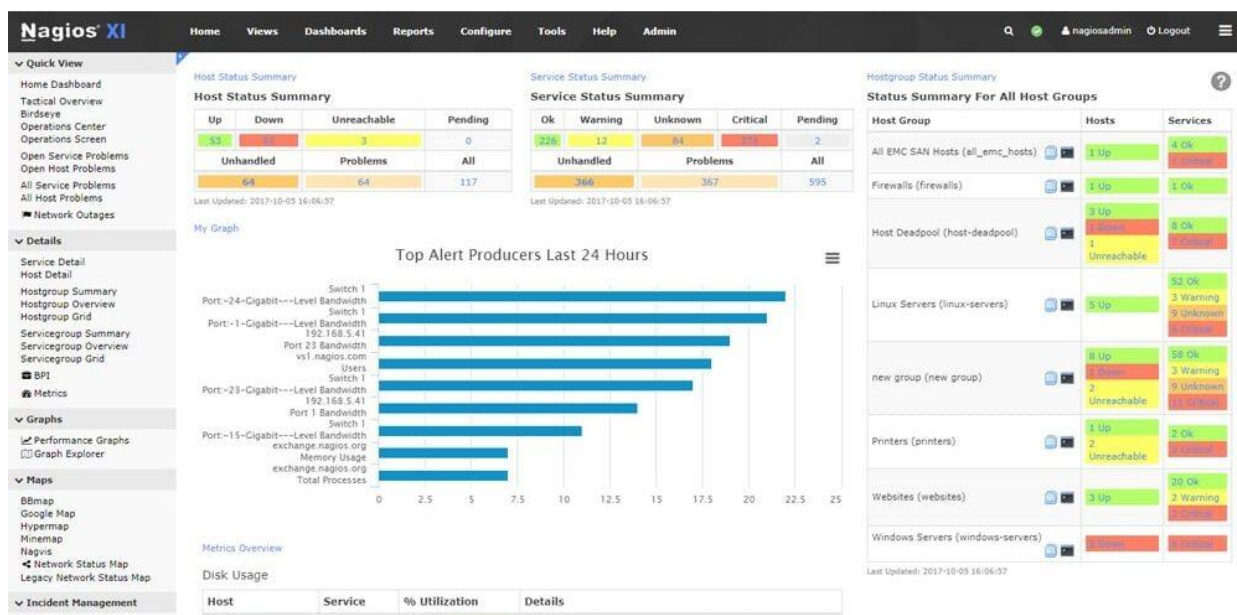


Рис 2.1 Система Serv-Nagios (Ubuntu Server 22.04)

Розгортання будемо проводити на сервері Serv-Nagios (Ubuntu Server 22.04). Встановлення з вихідних кодів, хоча й складніше, ніж з репозиторіїв, надає повний контроль над процесом та компонентами, що є важливою навичкою для спеціаліста з комп'ютерної інженерії.

Спочатку встановимо всі пакети, необхідні для компіляції та роботи Nagios та його веб-інтерфейсу.

```
sudo apt install -y wget build-essential apache2 php openssl perl make php-gd libgd-dev libapache2-mod-php libperl-dev libssl-dev daemon autoconf unzip
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

```
# sudo apt install -y wget build-essential apache2 php openssl perl make php-gd libgd-dev libapache2-mod-php libperl-dev libssl-dev daemon autoconf unzip

build-essential is already the newest version (12.12).
build-essential set to manually installed.
apache2 is already the newest version (2.4.63-1).
apache2 set to manually installed.
php is already the newest version (2:8.4+96).
php set to manually installed.
libapache2-mod-php is already the newest version (2:8.4+96).
libapache2-mod-php set to manually installed.
Upgrading:
fontconfig                libjpeg62-turbo           perl-modules-5.40
fontconfig-config         liblzma5                  php8.4
libaom3                   libperl5.40               php8.4-cli
libapache2-mod-php8.4     libpng16-16t64           php8.4-common
libavif16                 librav1e0.7              php8.4-mysql
libbrotli1                libssl3t64               php8.4-opcache
libdeflate0               libtiff6                  php8.4-readline
libfontconfig1            libturbojpeg0             python3-brotli
libgd3                    libvpx9                   python3-zstandard
libheif-plugin-aomenc     libzstd1                  unzip
libheif-plugin-dav1d      make                      wget
libheif-plugin-libde265  openssl                   xz-utils
libheif-plugin-x265       openssl-provider-legacy  zstd
```

Рис 2.2 Створення користувача та групи Nagios

Створимо спеціального користувача та групу, від імені яких працюватиме процес Nagios. Це є стандартною практикою безпеки для ізоляції процесів.

```
sudo useradd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd www-data
```

```
# id nagios
uid=1001(nagios) gid=1002(nagios) groups=1002(nagios),1001(nagcmd)
```

Рис 2.3 Завантаження та компіляція Nagios Core та Plugins

Завантажимо останні стабільні версії Nagios Core та Nagios Plugins. На момент написання цього звіту, це Nagios Core 4.5.9 та Nagios Plugins 2.4

```
cd /tmp
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.9.tar.gz
wget https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

```

# Розпакування архівів
tar -xzf nagios-4.5.9.tar.gz
tar -xzf nagios-plugins-2.4.9.tar.gz

# Компіляція та встановлення Nagios Core
cd nagios-4.5.9
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
sudo make all
sudo make install
sudo make install-init
sudo make install-commandmode
sudo make install-config
sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf

# Компіляція та встановлення Nagios Plugins
cd /tmp/nagios-plugins-2.4.9
sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios
sudo make
sudo make install
Налаштуємо Apache та створимо користувача для доступу до веб-інтерфейсу Nagios.

# Вмикаємо необхідні модулі Apache
sudo a2enmod rewrite cgi

# Створюємо користувача для веб-доступу
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

# Перезапускаємо Apache
sudo systemctl restart apache2

Перед першим запуском необхідно перевірити конфігурацію на наявність помилок.
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Якщо перевірка пройшла успішно (0 помилок, 0 попереджень), запускаємо сервіс Nagios.

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

```

Nagios Core 4.5.9
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-12-19
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

```

Рис 2.4. Результати перевірки конфігурації Nagios Core. Вивід команди демонструє успішне завершення процесу валідації конфігураційних файлів системи моніторингу без помилок та попереджень, що підтверджує коректність налаштувань перед запуском сервісу.

Архітектура конфігураційної системи Nagios Core базується на текстових файлах, організація яких суттєво впливає на ефективність адміністрування платформи моніторингу. Стандартна інсталяція Nagios Core пропонує монолітний підхід до зберігання конфігурацій, при якому визначення об'єктів розміщуються в невеликій кількості файлів значного обсягу. Зростання масштабів контрольованої інфраструктури при такому підході призводить до експоненціального збільшення складності управління, підвищення ризику помилок при внесенні змін та сповільнення процесу оновлення конфігурації. Модульна організація конфігураційних файлів трансформує процес адміністрування, забезпечуючи

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

логічне групування об'єктів, мінімізацію ризиків при модифікації параметрів та оптимізацію робочих процесів.

Імплементація структурованого підходу до організації конфігураційних файлів розпочинається зі створення спеціалізованої ієрархії каталогів, яка відображає логічну класифікацію об'єктів моніторингу за типами пристроїв або функціональним призначенням. Каталог `/usr/local/nagios/etc/objects` виступає базовим розташуванням для конфігураційних файлів об'єктів, а новостворена структура підкаталогів забезпечує сегментацію файлів за логічними категоріями. Створення каталогів виконується за допомогою команди `mkdir` з параметром `-p`, який забезпечує рекурсивне формування структури каталогів:

```
bash
sudo                                mkdir                                -p
/usr/local/nagios/etc/objects/{windows,linux,workstation,network,templates}
```

Синтаксис команди використовує розширення оболонки Bash для одночасного створення кількох підкаталогів: `windows` для зберігання конфігурацій Windows-серверів, `linux` для Linux-систем, `workstation` для робочих станцій, `network` для мережевого обладнання та `templates` для шаблонів об'єктів. Дана структура забезпечує логічне розділення конфігураційних файлів за типами систем, спрощуючи навігацію та пошук необхідних визначень.

Модифікація головного конфігураційного файлу Nagios Core `/usr/local/nagios/etc/nagios.cfg` становить наступний етап впровадження структурованого підходу. Стандартна конфігурація використовує директиви `cfg_file`, які вказують на окремі файли з визначеннями об'єктів. Трансформація підходу полягає у заміні цих директив на `cfg_dir`, які вказують Nagios на необхідність рекурсивного сканування вказаних каталогів та включення всіх файлів з розширенням `.cfg` до конфігурації:

```
# Закоментуйте або видаліть ці рядки:
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
#cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

Додайте ці рядки для сканування каталогів:

cfg_dir=/usr/local/nagios/etc/objects/windows

cfg_dir=/usr/local/nagios/etc/objects/linux

cfg_dir=/usr/local/nagios/etc/objects/workstation

cfg_dir=/usr/local/nagios/etc/objects/network

cfg_dir=/usr/local/nagios/etc/objects/templates

Процес модифікації передбачає коментування або видалення існуючих директив `cfg_file`, які вказують на стандартні конфігураційні файли, та додавання нових директив `cfg_dir` для кожного з створених підкаталогів. Символ решітки (#) на початку рядка перетворює директиву на коментар, вимикаючи її обробку системою. Важливо забезпечити збереження інших директив `cfg_file`, які вказують на критичні конфігураційні файли, такі як `commands.cfg`, `timeperiods.cfg`, `contacts.cfg` та інші, необхідні для базового функціонування системи.

Після внесення змін до головного конфігураційного файлу необхідно перевірити коректність конфігурації за допомогою команди:

bash

sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Успішна перевірка конфігурації (0 помилок, 0 попереджень) є необхідною умовою для перезапуску сервісу Nagios та застосування нової структури:

bash

sudo systemctl restart nagios

Впровадження модульної структури конфігураційних файлів надає низку суттєвих переваг для адміністрування системи моніторингу. Логічне групування об'єктів за типами систем спрощує процес навігації та пошуку необхідних визначень при внесенні змін, що особливо актуально для великих інфраструктур з сотнями хостів та тисячами сервісів. Ізоляція змін у межах окремих файлів мінімізує ризики при модифікації конфігурації, оскільки внесення змін до одного файлу не впливає на інші компоненти системи. Розподіл відповідальності між адміністраторами спрощується через можливість призначення окремих каталогів або файлів конкретним фахівцям, що відповідають за відповідні компоненти інфраструктури.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

Методологія розподілу конфігураційних файлів у створеній структурі каталогів передбачає використання окремих файлів для кожного хоста або логічної групи хостів. Наприклад, файл `/usr/local/nagios/etc/objects/windows/serv-domain.cfg` містить визначення хоста для контролера домену Serv-Domain та всіх пов'язаних з ним сервісів. Файл `/usr/local/nagios/etc/objects/templates/host-templates.cfg` містить шаблони хостів, які використовуються в усій конфігурації. Такий підхід забезпечує максимальну гнучкість та мінімізує конфлікти при одночасному редагуванні конфігурації кількома адміністраторами.

Додатковою перевагою модульної структури є спрощення процесу резервного копіювання та відновлення конфігурації. Можливість створення резервних копій окремих компонентів системи без необхідності копіювання всієї конфігурації спрощує процес управління змінами та мінімізує ризики при внесенні модифікацій. Інтеграція з системами контролю версій, такими як Git, також спрощується завдяки логічній сегментації файлів, що забезпечує більш детальне відстеження змін та можливість відкату до попередніх версій окремих компонентів конфігурації.

Автоматизація процесу управління конфігурацією через скрипти або системи управління конфігурацією, такі як Ansible, значно спрощується при використанні модульної структури. Можливість генерації окремих файлів для нових хостів або модифікації існуючих конфігурацій без необхідності обробки великих монолітних файлів підвищує ефективність автоматизації та мінімізує ризики помилок. Стандартизація іменування файлів та структури каталогів забезпечує передбачуваність та спрощує розробку скриптів автоматизації.

Впровадження структурованого підходу до організації конфігураційних файлів Nagios Core демонструє розуміння принципів ефективного адміністрування систем моніторингу та забезпечує основу для подальшого масштабування інфраструктури. Модульна архітектура конфігурації трансформує процес управління системою, підвищуючи гнучкість, мінімізуючи ризики та спрощуючи робочі процеси адміністрування.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

```

PRE-CACHED OBJECT FILE
This option determines the location of the precached object file.
If you run Nagios with the -p command line option, it will preprocess
your object configuration file(s) and write the cached config to this
file. You can then start Nagios with the -u option to have it read
object definitions from this precached file, rather than the standard
object configuration files (see the cfg_file and cfg_dir options above).
Using a precached object file can speed up the time needed to (re)start
the Nagios process if you've got a large and/or complex configuration.
Read the documentation section on optimizing Nagios to find out more
about how this feature works.

precached_object_file=/usr/local/nagios/var/objects.precache

Nagios Core 4.4.0 (Debian) Server at 192.168.1.9 Port 80

RESOURCE FILE
This is an optional resource file that contains $USERx$ macro
definitions. Multiple resource files can be specified by using
multiple resource_file definitions. The CGIs will not attempt to
read the contents of resource files, so information that is
considered to be sensitive (usernames, passwords, etc) can be
defined as macros in this file and restrictive permissions (600)
can be placed on this file.

resource_file=/usr/local/nagios/etc/resource.cfg

STATUS FILE
This is where the current status of all monitored services and
hosts is stored. Its contents are read and processed by the CGIs.
The contents of the status file are deleted every time Nagios
restarts.

status_file=/usr/local/nagios/var/status.dat

cfg_dir=/usr/local/nagios/etc/objects/windows
cfg_dir=/usr/local/nagios/etc/objects/linux
cfg_dir=/usr/local/nagios/etc/objects/workstation
cfg_dir=/usr/local/nagios/etc/objects/network

STATUS FILE UPDATE INTERVAL
This option determines the frequency (in seconds) that
Nagios will periodically dump program, host, and
service status data.

status_update_interval=10

NAGIOS USER
This determines the effective user that Nagios should run as.
You can either supply a username or a UID.

nagios_user=nagios

NAGIOS GROUP
This determines the effective group that Nagios should run as.
You can either supply a group name or a GID.

nagios_group=nagios

EXTERNAL COMMAND OPTION
This option allows you to specify whether or not Nagios should check
for external commands (in the command file defined below).

```

Рис 2.5 Фрагмент головного конфігураційного файлу Nagios Core (nagios.cfg) зі структурованою організацією директив. На зображенні представлено налаштування ключових параметрів системи моніторингу, включаючи шляхи до файлів статусу, ресурсів та директорії з об'єктами моніторингу, організовані за типами пристроїв.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

Цей підхід дозволить у майбутньому легко додавати нові хости, просто створюючи новий .cfg файл у відповідному каталозі, без необхідності редагувати головний конфігураційний файл.\

2.2. Додавання Хостів та Базовий Моніторинг

Налаштування моніторингу хостів представляє фундаментальний етап впровадження системи Nagios Core, який визначає фактичний обсяг контрольованої інфраструктури та глибину моніторингу кожного компонента. Основна філософія Nagios базується на концепції об'єктно-орієнтованої моделі конфігурації, де кожен контрольований елемент інфраструктури визначається як об'єкт певного типу з набором атрибутів, які описують його характеристики та параметри перевірок. Архітектура моніторингу в Nagios Core реалізується через ієрархічну систему об'єктів: контактні групи, які визначають отримувачів сповіщень; часові періоди, що встановлюють проміжки часу для виконання перевірок; команди, які описують механізми виконання перевірок; хости, що представляють фізичні або віртуальні пристрої; сервіси, які відповідають конкретним аспектам функціонування хостів; та залежності, які визначають логічні взаємозв'язки між компонентами. Інтеграція різноманітних хостів у єдину систему моніторингу реалізується через відповідні агенти моніторингу, які встановлюються на контрольованих системах та забезпечують механізм збору та передачі інформації про стан компонентів.

Додавання хостів до системи моніторингу вимагає розуміння специфіки різних операційних систем та протоколів, що використовуються для збору інформації. Nagios Core підтримує декілька основних підходів до моніторингу хостів: активні перевірки через мережеві протоколи (ICMP, SNMP, HTTP, SSH тощо) без встановлення агентів; моніторинг через спеціалізовані агенти, які встановлюються на контрольованих системах та надають розширені можливості збору інформації; та пасивний моніторинг, при якому хости самостійно відправляють дані про свій стан до сервера Nagios. Вибір оптимального підходу

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

залежить від типу системи, доступних механізмів комунікації та специфічних вимог до моніторингу.

Впровадження моніторингу серверів під управлінням операційної системи Windows вимагає використання спеціалізованих агентів, оскільки стандартні інструменти Nagios розраховані переважно на Unix-подібні системи. NSClient++ представляє собою потужний та гнучкий агент моніторингу для Windows-систем, який забезпечує широкий спектр можливостей: від базових перевірок стану системних ресурсів до складного моніторингу служб, подій журналу та виконання скриптів. NSClient++ підтримує декілька протоколів комунікації з сервером Nagios, включаючи NRPE (Nagios Remote Plugin Executor), NSClient (власний протокол) та NSCA (Nagios Service Check Acceptor), що забезпечує гнучкість при інтеграції з різними архітектурами моніторингу.

Процес встановлення NSClient++ на контролері домену Serv-Domain розпочинається з завантаження інсталяційного пакету з офіційного сайту проєкту. Версія 0.5.2.39 рекомендована для використання з плагіном check_nt в Nagios Core через стабільність та повну сумісність з існуючими командами перевірки. Інсталяційний пакет NSCP-0.5.2.39-x64.msi надається у форматі Windows Installer, що спрощує процес встановлення та подальшого оновлення компонента.

Інсталяція NSClient++ виконується через графічний інтерфейс з наступними параметрами:

1. На сторінці "Installation mode" обирається тип "Generic", який забезпечує встановлення базових компонентів без специфічних налаштувань для конкретних систем моніторингу.

2. На сторінці "Configuration" у полі "Allowed hosts" вказується IP-адреса сервера Nagios (192.168.13.2), що обмежує доступ до агента лише з авторизованих хостів та підвищує безпеку системи. У полі "Password" встановлюється пароль для автентифікації запитів від сервера Nagios (наприклад, "P@ssword2023"). Стійкий пароль є критично важливим для запобігання несанкціонованому доступу до агента.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

3. На сторінці "Modules" активуються необхідні модулі: CheckSystem для моніторингу системних ресурсів, CheckDisk для контролю дискового простору, NRPEServer для забезпечення NRPE-протоколу та NSClientServer для підтримки протоколу NSClient. Додатково рекомендується включити модуль CheckEventLog для моніторингу журналу подій Windows, який є важливим джерелом інформації про стан системи.

Після завершення інсталяції рекомендується перевірити та, за необхідності, модифікувати конфігураційний файл NSClient++ для забезпечення оптимальної функціональності. Файл розташований за шляхом C:\Program Files\NSClient++\nscclient.ini та містить всі налаштування агента, організовані в секції за функціональним призначенням. Критично важливо переконатися, що необхідні модулі активовані в секції [/modules] шляхом встановлення параметра enabled=true для кожного з них:

```
ini
[/modules]
CheckSystem = enabled
CheckDisk = enabled
CheckEventLog = enabled
NRPEServer = enabled
NSClientServer = enabled
```

Додатково можна налаштувати параметри окремих модулів для оптимізації їх роботи. Наприклад, для модуля CheckDisk можна вказати конкретні диски для моніторингу та встановити порогові значення для сповіщень:

```
ini
[/settings/disk]
default drive = c:\ d:\
default filter = free < 10%
```

Для модуля CheckEventLog можна визначити конкретні журнали для моніторингу та фільтри для відбору релевантних подій:

```
ini
[/settings/eventlog/real-time]
enabled = true
log = Application, System, Security
maximum age = 60m
level = critical, error, warning
```

Після внесення змін до конфігураційного файлу необхідно перезапустити службу NSClient++ для застосування нових налаштувань. Перезапуск можна

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

виконати через інтерфейс "Служби" в Windows або за допомогою командного рядка:

```
cmd
net stop nscp
net start nscp
```

Для перевірки коректності роботи агента можна використати вбудовану консоль NSClient++, яка дозволяє виконувати тестові перевірки безпосередньо на контрольованому хості:

```
cmd
cd "C:\Program Files\NSClient++"
nscp client -q check_cpu
```

Після успішного встановлення та налаштування агента NSClient++ на контролері домену необхідно створити відповідну конфігурацію на сервері Nagios для включення хоста до системи моніторингу. Конфігурація створюється у вигляді окремого файлу в каталозі /usr/local/nagios/etc/objects/windows/ для дотримання структурованого підходу до організації конфігураційних файлів.

Створіть файл /usr/local/nagios/etc/objects/windows/serv-domain.cfg з наступним вмістом:

```
define host{
    use                windows-server
    host_name          Serv-Domain
    alias              Windows Domain Controller
    address            192.168.13.3
}

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description Uptime
    check_command       check_nt!UPTIME -s P@ssword2023
}

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description CPU Load
    check_command       check_nt!CPULOAD!-1 5,80,90 -s P@ssword2023
}
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description Memory Usage
    check_command       check_nt!MEMUSE!-w 80 -c 90 -s P@ssword2023
}

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description C:\ Drive Space
    check_command       check_nt!USEDISKSPACE!-l c -w 80 -c 90 -s
P@ssword2023
}

```

Наведена конфігурація включає визначення хоста Serv-Domain та чотирьох базових сервісів для моніторингу: час безперервної роботи системи (Uptime), навантаження процесора (CPU Load), використання оперативної пам'яті (Memory Usage) та вільний простір на системному диску (C:\ Drive Space). Кожен сервіс використовує команду check_nt з відповідними параметрами для виконання перевірки через протокол NSClient.

Додатково рекомендується налаштувати моніторинг критичних служб Windows, які забезпечують функціонування контролера домену та DNS-сервера:

```

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description DNS Service
    check_command       check_nt!SERVICESTATE!-d SHOWALL -l DNS -s
P@ssword2023
}

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description Active Directory Service
    check_command       check_nt!SERVICESTATE!-d SHOWALL -l NTDS -s
P@ssword2023
}

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description DHCP Service
    check_command       check_nt!SERVICESTATE!-d SHOWALL -l DHCPService -
s P@ssword2023
}

```

Моніторинг критичних служб дозволяє вчасно виявляти проблеми з функціонуванням ключових компонентів інфраструктури та запобігати каскадним відмовам, які можуть виникнути при збої базових сервісів.

Для забезпечення комплексного моніторингу контролера домену також рекомендується налаштувати перевірку журналу подій Windows, який містить критичну інформацію про стан системи та потенційні проблеми:

```

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description System Event Log
    check_command       check_nt!EVENTLOG!-l System -s P@ssword2023
}

define service{
    use                generic-service
    host_name          Serv-Domain
    service_description Application Event Log
    check_command       check_nt!EVENTLOG!-l Application -s P@ssword2023
}

```

Команда `check_nt!EVENTLOG` дозволяє відстежувати нові записи в журналі подій Windows та генерувати сповіщення при виявленні подій з певним рівнем критичності (за замовчуванням перевіряються події типу Error та Warning).

Для моніторингу Linux-систем Nagios надає декілька підходів, включаючи прямі перевірки через SSH та використання агента NRPE (Nagios Remote Plugin Executor). NRPE забезпечує безпечний механізм виконання плагінів Nagios безпосередньо на контрольованому хості, що дозволяє збирати детальну інформацію про стан системи, яка недоступна через стандартні мережеві протоколи. Встановлення NRPE на сервері Serv-Linux (Ubuntu Desktop 22.04 LTS) забезпечить можливість глибокого моніторингу системних ресурсів та сервісів.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

Процес встановлення NRPE на Ubuntu включає інсталяцію пакетів nagios-nrpe-server та моніторингових плагінів, які будуть виконуватися на контрольованому хості:

```
bash
sudo apt update
sudo apt install -y nagios-nrpe-server nagios-plugins
```

Після встановлення необхідно налаштувати NRPE, відредагувавши файл конфігурації `/etc/nagios/nrpe.cfg`:

```
bash
sudo nano /etc/nagios/nrpe.cfg
```

Основні параметри, які потребують налаштування:

1. `server_address`: IP-адреса, на якій NRPE буде прослуховувати вхідні з'єднання. За замовчуванням використовується 127.0.0.1, що обмежує доступ лише з локального хоста. Для забезпечення доступу з сервера Nagios необхідно вказати IP-адресу мережевого інтерфейсу (192.168.13.4).

2. `server_port`: Порт, на якому NRPE буде прослуховувати вхідні з'єднання. Стандартний порт для NRPE - 5666.

3. `allowed_hosts`: Список IP-адрес, з яких дозволено підключення до NRPE. Для забезпечення безпеки необхідно вказати лише IP-адресу сервера Nagios (192.168.13.2).

4. `dont_blame_nrpe`: Параметр, який дозволяє або забороняє передачу аргументів з сервера Nagios до плагінів на контрольованому хості. Для базового моніторингу рекомендується встановити значення 0 (заборонено), але для деяких сценаріїв може знадобитися значення 1 (дозволено).

5. `command[...]`: Визначення команд, які можуть бути виконані через NRPE. За замовчуванням включені базові перевірки, такі як контроль навантаження процесора, використання пам'яті та дискового простору.

Приклад налаштування основних параметрів:

```
server_address=192.168.13.4
server_port=5666
allowed_hosts=127.0.0.1,192.168.13.2
dont_blame_nrpe=0
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

Успішне виконання команди свідчить про коректну роботу NRPE на контрольованому хості.

Після налаштування NRPE на Serv-Linux необхідно створити відповідну конфігурацію на сервері Nagios. Створіть файл /usr/local/nagios/etc/objects/linux/serv-linux.cfg з наступним вмістом:

```
define host{
    use          linux-server
    host_name    Serv-Linux
    alias        Linux Test Server
    address      192.168.13.4
}

define service{
    use          generic-service
    host_name    Serv-Linux
    service_description  PING
    check_command    check_ping!100.0,20%!500.0,60%
}

define service{
    use          generic-service
    host_name    Serv-Linux
    service_description  SSH
    check_command    check_ssh
}

define service{
    use          generic-service
    host_name    Serv-Linux
    service_description  Current Load
    check_command    check_nrpe!check_load
}

define service{
    use          generic-service
    host_name    Serv-Linux
    service_description  Current Users
    check_command    check_nrpe!check_users
}
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Disk Space
    check_command      check_nrpe!check_disk
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Total Processes
    check_command      check_nrpe!check_total_procs
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Apache Status
    check_command      check_nrpe!check_apache
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description MySQL Status
    check_command      check_nrpe!check_mysql
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description HTTP
    check_command      check_nrpe!check_http
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Apt Updates
    check_command      check_nrpe!check_apt
}

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Наведена конфігурація включає визначення хоста Serv-Linux та набору сервісів для комплексного моніторингу: базові перевірки доступності (PING, SSH), контроль системних ресурсів (навантаження процесора, використання пам'яті, дисковий простір, кількість процесів), моніторинг сервісів (Apache, MySQL, HTTP) та перевірка наявності доступних оновлень (Apt Updates).

Моніторинг клієнтської робочої станції під управлінням Windows 10 Pro здійснюється аналогічно до моніторингу Windows Server, з використанням агента NSClient++. Основна відмінність полягає в наборі перевірок, які фокусуються на аспектах, релевантних для робочої станції, та з урахуванням потенційно нижчої критичності цього компонента інфраструктури порівняно з серверними системами.

Процес встановлення NSClient++ на Client-Win ідентичний до описаного раніше для Serv-Domain, з аналогічними параметрами конфігурації. Після успішного встановлення та налаштування агента створіть файл конфігурації /usr/local/nagios/etc/objects/workstation/client-win.cfg на сервері Nagios з наступним вмістом:

```
define host{
    use      windows-workstation
    host_name Client-Win
    alias    Windows 10 Workstation
    address  192.168.13.32
}

define service{
    use      generic-service
    host_name Client-Win
    service_description Uptime
    check_command check_nt!UPTIME -s P@ssword2023
}

define service{
    use      generic-service
    host_name Client-Win
    service_description CPU Load
    check_command check_nt!CPULOAD!-l 5,80,90 -s P@ssword2023
}

define service{
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

use          generic-service
host_name    Client-Win
service_description  Memory Usage
check_command    check_nt!MEMUSE!-w 80 -c 90 -s P@ssword2023
}

define service{
use          generic-service
host_name    Client-Win
service_description  C:\ Drive Space
check_command    check_nt!USEDISKSPACE!-l c -w 80 -c 90 -s P@ssword2023
}

define service{
use          generic-service
host_name    Client-Win
service_description  IIS Service
check_command    check_nt!SERVICESTATE!-d SHOWALL -l W3SVC -s P@ssword2023
}

```

Наведена конфігурація включає визначення хоста Client-Win та базових сервісів для моніторингу робочої станції: час безперервної роботи, навантаження процесора, використання оперативної пам'яті, вільний простір на системному диску та статус служби IIS (Internet Information Services), яка була встановлена на етапі підготовки віртуального середовища.

Для забезпечення комплексності системи моніторингу необхідно налаштувати моніторинг самого сервера Nagios, що дозволить відстежувати його стан та запобігати відмовам системи моніторингу. Налаштування моніторингу локального хоста не потребує встановлення додаткових агентів, оскільки плагіни Nagios можуть виконуватися безпосередньо на сервері.

Створіть файл конфігурації /usr/local/nagios/etc/objects/linux/serv-nagios.cfg з наступним вмістом:

```

define host{
use          linux-server
host_name    Serv-Nagios
alias        Nagios Monitoring Server
address      127.0.0.1
}

define service{

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

```

use          local-service
host_name    Serv-Nagios
service_description  PING
check_command    check_ping!100.0,20%!500.0,60%
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  Root Partition
    check_command    check_local_disk!20%!10%!/
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  Current Users
    check_command    check_local_users!20!50
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  Total Processes
    check_command    check_local_procs!250!400!RSZDT
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  Current Load
    check_command    check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  Swap Usage
    check_command    check_local_swap!20!10
}

```

```

define service{

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

```

use          local-service
host_name    Serv-Nagios
service_description  SSH
check_command    check_ssh
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  HTTP
    check_command    check_http
}

```

```

define service{
    use          local-service
    host_name    Serv-Nagios
    service_description  Nagios Process
    check_command    check_nagios!/usr/local/nagios/var/status.dat!5!10
}

```

Наведена конфігурація включає визначення хоста Serv-Nagios та набору сервісів для комплексного моніторингу сервера: перевірка доступності (PING), контроль системних ресурсів (дисковий простір, кількість користувачів, процесів, навантаження процесора, використання swar), моніторинг сервісів (SSH, HTTP) та перевірка статусу самого процесу Nagios.

Для спрощення конфігурації та забезпечення уніфікованого підходу до моніторингу різних типів хостів рекомендується створити спеціалізовані шаблони, які визначають базові параметри моніторингу для кожного типу систем. Шаблони дозволяють уникнути дублювання конфігурації та забезпечують єдиний підхід до налаштування перевірок.

Створіть файл /usr/local/nagios/etc/objects/templates/host-templates.cfg з наступним вмістом:

```

define host{
    name          generic-host
    notifications_enabled  1
    event_handler_enabled  1
    flap_detection_enabled  1
    process_perf_data      1
}

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

retain_status_information 1
retain_nonstatus_information 1
check_command      check-host-alive
max_check_attempts  10
notification_interval 120
notification_period 24x7
notification_options d,u,r
contact_groups      admins
register            0
}

```

```

define host{
    name          windows-server
    use            generic-host
    check_period   24x7
    check_interval 5
    retry_interval 1
    max_check_attempts 10
    check_command  check-host-alive
    notification_period workhours
    notification_interval 30
    notification_options d,u,r
    contact_groups  admins
    register        0
}

```

```

define host{
    name          windows-workstation
    use            generic-host
    check_period   24x7
    check_interval 5
    retry_interval 1
    max_check_attempts 10
    check_command  check-host-alive
    notification_period workhours
    notification_interval 30
    notification_options d,u,r
    contact_groups  admins
    register        0
}

```

```

define host{

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

```

name          linux-server
use           generic-host
check_period   24x7
check_interval 5
retry_interval 1
max_check_attempts 10
check_command  check-host-alive
notification_period 24x7
notification_interval 30
notification_options d,u,r
contact_groups admins
register       0
}

```

Також створіть файл /usr/local/nagios/etc/objects/templates/service-templates.cfg для визначення шаблонів сервісів:

```

define service{
    name          generic-service
    active_checks_enabled 1
    passive_checks_enabled 1
    parallelize_check 1
    obsess_over_service 1
    check_freshness 0
    notifications_enabled 1
    event_handler_enabled 1
    flap_detection_enabled 1
    process_perf_data 1
    retain_status_information 1
    retain_nonstatus_information 1
    is_volatile 0
    check_period 24x7
    max_check_attempts 3
    normal_check_interval 10
    retry_check_interval 2
    contact_groups admins
    notification_options w,u,c,r
    notification_interval 60
    notification_period 24x7
    register 0
}

```

```

define service{
    name          local-service
    use           generic-service

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

```

max_check_attempts 4
normal_check_interval 5
retry_check_interval 1
register 0
}

```

```

define service{
    name windows-service
    use generic-service
    max_check_attempts 4
    normal_check_interval 5
    retry_check_interval 1
    notification_period workhours
    register 0
}

```

```

define service{
    name linux-service
    use generic-service
    max_check_attempts 4
    normal_check_interval 5
    retry_check_interval 1
    register 0
}

```

Створені шаблони хостів та сервісів забезпечують базову конфігурацію для різних типів систем, яка може бути розширена або модифікована відповідно до специфічних вимог моніторингу. Використання шаблонів спрощує додавання нових хостів до системи моніторингу та забезпечує консистентний підхід до конфігурації.

Для забезпечення функціональності моніторингу необхідно визначити команди, які використовуються для виконання перевірок. Команди в Nagios представляють собою шаблони виклику плагінів з відповідними параметрами, які можуть бути використані для моніторингу різних аспектів системи.

Створіть файл `/usr/local/nagios/etc/objects/templates/commands.cfg` з наступним вмістом:

```

define command{
    command_name check_nt
    command_line $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -v $ARG1$ $ARG2$
}

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

}

define command{

command_name check_nrpe

command_line \$USER1\$/check_nrpe -H \$HOSTADDRESS\$ -c \$ARG1\$

}

define command{

command_name check_local_disk

command_line \$USER1\$/check_disk -w \$ARG1\$ -c \$ARG2\$ -p \$ARG3\$

}

define command{

command_name check_local_load

command_line \$USER1\$/check_load -w \$ARG1\$ -c \$ARG2\$

}

define command{

command_name check_local_procs

command_line \$USER1\$/check_procs -w \$ARG1\$ -c \$ARG2\$ -s \$ARG3\$

}

define command{

command_name check_local_users

command_line \$USER1\$/check_users -w \$ARG1\$ -c \$ARG2\$

}

define command{

command_name check_local_swap

command_line \$USER1\$/check_swap -w \$ARG1\$ -c \$ARG2\$

}

define command{

command_name check_http

command_line \$USER1\$/check_http -I \$HOSTADDRESS\$ \$ARG1\$

}

define command{

command_name check_ssh

command_line \$USER1\$/check_ssh \$ARG1\$ \$HOSTADDRESS\$

}

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

```
define command{
    command_name check_nagios
    command_line $USER1$/check_nagios -F $ARG1$ -C $ARG2$ -W $ARG3$
}
```

Наведені команди забезпечують функціональність моніторингу різних аспектів системи: check_nt для моніторингу Windows-систем через NSClient++, check_nrpe для виконання перевірок на Linux-системах через NRPE, а також команди для локальних перевірок на сервері Nagios та моніторингу стандартних сервісів, таких як HTTP та SSH.

Після створення всіх необхідних конфігураційних файлів необхідно перевірити їх коректність та активувати нову конфігурацію. Перевірка конфігурації виконується за допомогою команди:

```
bash
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Утиліта перевірки аналізує всі конфігураційні файли та виводить звіт про знайдені помилки та попередження. Успішна перевірка (0 помилок, 0 попереджень) є необхідною умовою для активації нової конфігурації.

Після успішної перевірки необхідно перезапустити сервіс Nagios для застосування змін:

```
bash
sudo systemctl restart nagios
```

Після перезапуску сервісу нова конфігурація активується, і Nagios починає моніторинг всіх визначених хостів та сервісів. Статус моніторингу можна переглянути через веб-інтерфейс Nagios за адресою <http://192.168.13.2/nagios>.

Таблиця 2.2 - Порівняння агентів моніторингу для різних операційних систем

Агент	ОС	Протокол	Переваги	Недоліки
NSClient++	Windows	NSClient, NRPE	Широкий функціонал, гнучка конфігурація	Складніша інсталяція та налаштування
NRPE	Linux, Unix	NRPE	Безпечний	Вимагає

			механізм, стандарт для Linux	додаткового відкриття портів
check_by_ssh	Linux, Unix	SSH	Не потребує додаткового ПЗ	Менша ефективність, більше навантаження
SNMP	Мережеве обладнання	SNMP	Універсальність, підтримка різних пристроїв	Обмежений набір даних, менша безпека

Налаштування базового моніторингу для всіх хостів віртуальної інфраструктури забезпечує фундамент для подальшого розширення системи моніторингу та впровадження додаткових функцій, таких як сповіщення, візуалізація даних, аналіз тенденцій та автоматизація реагування на інциденти. Структурований підхід до організації конфігураційних файлів та використання шаблонів спрощує підтримку та масштабування системи моніторингу відповідно до зростаючих потреб інфраструктури.

1.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

```
GNU nano 8.3 /etc/nagios/conf.d/windows-server.conf
define host{
    use generic-service
    host_name Serv-WinDC
    alias Windows Domain Controller
    address 192.168.13.11
}

define service{
    use generic-service
    host_name Serv-WinDC
    service_description Uptime
    check_command check_nt!UPTIME -s P@ssword2023
}

define service{
    use generic-service
    host_name Serv-WinDC
    service_description CPU Load
    check_command check_nt!CPULOAD!-l 5,80,90 -s P@ssword2023
}

define service{
    use generic-service
    host_name Serv-WinDC
    service_description Memory Usage
    check_command check_nt!MEMUSE!-w 80 -c 90 -s P@ssword2023
}

define service{
    use generic-service
    host_name Serv-WinDC
    service_description C:\ Drive Space
    check_command check_nt!USEDISKSPACE!-l c -w 80 -c 90 -s P@ssword2023
}

define service{
    use generic-service
    host_name Serv-WinDC
    service_description C:\ Drive Space
    check_command check_nt!USEDISKSPACE!-l c -w 80 -c 90 -s P@ssword2023
}
```

Рисунок 2.6. Приклад конфігураційного файлу для моніторингу Windows Server у Nagios Core. Зображення демонструє визначення хоста контролера домену (Serv-WinDC) та набору базових сервісів для моніторингу системних ресурсів з використанням агента NSClient++ через команду check_nt.

Різноманітність агентів моніторингу в екосистемі Nagios забезпечує можливість вибору оптимального рішення для конкретних сценаріїв використання та типів контрольованих систем. Nagios Cross-Platform Agent (NCRA) представляє собою сучасне мультиплатформне рішення нового покоління, яке забезпечує уніфікований підхід до моніторингу різних операційних систем. Архітектура NCRA базується на принципі агностичності до платформи, що дозволяє використовувати єдиний механізм моніторингу для Windows, Linux, macOS та інших операційних систем без необхідності адаптації конфігурації для

кожної платформи окремо. Ключовими перевагами NCPA порівняно з традиційними агентами (NSClient++, NRPE) є вбудований веб-інтерфейс для локального перегляду метрик та налаштування агента, інтегрований API для програмного доступу до даних, модульна система розширень та вбудована підтримка захищеного з'єднання через TLS.

Архітектура NCPA складається з трьох основних компонентів: Listener, який забезпечує доступ до метрик через API та веб-інтерфейс; Passive, який відправляє дані на сервер Nagios через NSCA або NRDP; та Graphing Engine, який забезпечує візуалізацію історичних даних безпосередньо в інтерфейсі агента. Модульний дизайн дозволяє активувати лише необхідні компоненти, оптимізуючи використання системних ресурсів на контрольованих хостах. Впровадження NCPA для моніторингу клієнтських систем демонструє розуміння сучасних тенденцій у сфері моніторингу та здатність інтегрувати різноманітні інструменти в єдину систему контролю.

Процес встановлення NCPA на Windows 10 (Client-Win) розпочинається з завантаження інсталяційного пакету з офіційного сайту Nagios. Актуальна стабільна версія 3.1.3 забезпечує оптимальний баланс між функціональністю та стабільністю роботи. Інсталяційний пакет ncpa-3.1.3.exe доступний у форматі Windows Installer, що спрощує процес встановлення та подальшого оновлення компонента.

Процедура інсталяції NCPA на Windows 10 виконується через графічний інтерфейс з наступними кроками та параметрами:

1. Запустіть інсталятор ncpa-3.1.3.exe з правами адміністратора.
2. На екрані привітання натисніть "Next" для продовження.
3. Прийміть ліцензійну угоду та натисніть "Next".
4. Виберіть компоненти для встановлення. Рекомендується залишити всі

компоненти активними для забезпечення повної функціональності: Listener для обробки запитів від сервера Nagios, Passive для відправки даних на сервер, та Windows Service для автоматичного запуску агента при завантаженні системи.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

5. На екрані "Listener Configuration" введіть токен автентифікації (наприклад, "mytoken"), який буде використовуватися для захисту доступу до агента. Залиште параметр "Bind IP" як "0.0.0.0" для прослуховування всіх мережевих інтерфейсів, або вкажіть конкретну IP-адресу для підвищення безпеки.

6. На екрані "Passive Configuration" вкажіть параметри для пасивних перевірок, якщо планується їх використання. В базовій конфігурації можна залишити значення за замовчуванням.

7. Виберіть каталог для встановлення та натисніть "Next".

8. Підтвердіть налаштування та натисніть "Install" для початку процесу інсталяції.

9. По завершенню встановлення натисніть "Finish".

Після успішного встановлення NCPA на Windows 10 рекомендується перевірити коректність роботи агента, відкривши веб-інтерфейс за адресою <http://localhost:5693> та вказавши токен автентифікації. Веб-інтерфейс надає доступ до детальної інформації про стан системи, включаючи використання процесора, пам'яті, дискового простору, мережеву активність та інші метрики.

Встановлення NCPA на Ubuntu Desktop (Serv-Linux) відрізняється від процесу для Windows та виконується через командний рядок з використанням пакетного менеджера. Процедура інсталяції включає наступні кроки:

1. Завантаження пакету NCPA для Debian/Ubuntu:

```
bash
```

```
cd /tmp
```

```
wget https://assets.nagios.com/downloads/ncpa/ncpa-latest.deb
```

2. Встановлення пакету за допомогою dpkg:

```
bash
```

```
sudo dpkg -i ncpa-latest.deb
```

При виникненні помилок залежностей використовуйте apt для їх вирішення:

```
bash
```

```
sudo apt -f install
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

3. Налаштування токена автентифікації шляхом редагування файлу конфігурації:

```
bash
```

```
sudo nano /usr/local/ncpa/etc/ncpa.cfg
```

У секції [api] змініть параметр community_string на обраний токен:

```
ini
```

```
[api]
```

```
community_string = mytoken
```

4. Перезапуск служби NCPA Listener для застосування змін:

```
bash
```

```
sudo systemctl restart ncpa-listener
```

5. Перевірка статусу служби для підтвердження успішного запуску:

```
bash
```

```
sudo systemctl status ncpa-listener
```

Для забезпечення автоматичного запуску NCPA при завантаженні системи активуйте службу:

```
bash
```

```
sudo systemctl enable ncpa-listener
```

Аналогічно до Windows-версії, після встановлення рекомендується перевірити коректність роботи агента через веб-інтерфейс за адресою <http://localhost:5693>, використовуючи налаштований токен автентифікації.

Для інтеграції NCPA з Nagios Core необхідно встановити плагін check_ncpa.py на сервері моніторингу, який забезпечить взаємодію з API агента та отримання метрик. Процес налаштування включає наступні кроки:

1. Завантаження та налаштування плагіну check_ncpa.py:

```
bash
```

```
sudo wget -O /usr/local/nagios/libexec/check_ncpa.py
```

```
https://raw.githubusercontent.com/NagiosEnterprises/ncpa/master/client/check_ncpa.py
```

```
sudo chmod +x /usr/local/nagios/libexec/check_ncpa.py
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

2. Додавання визначення команди `check_ncpa` до конфігурації Nagios. Відредагуйте файл `/usr/local/nagios/etc/objects/templates/commands.cfg` та додайте наступне визначення:

```
define command{
    command_name    check_ncpa
    command_line     $USER1$/check_ncpa.py -H $HOSTADDRESS$ -t
'$ARG1$' -M '$ARG2$' $ARG3$
}
```

Параметри команди `check_ncpa` включають: `-H` для вказання хоста, `-t` для токена автентифікації, `-M` для шляху до метрики в API NCPA, та додаткові аргументи для порогових значень та опцій.

Після налаштування плагіну та команди необхідно створити конфігурації для моніторингу хостів з використанням NCPA. Для Windows 10 (Client-Win) створіть файл `/usr/local/nagios/etc/objects/workstation/client-win-ncpa.cfg` з наступним вмістом:

```
define host{
    use                windows-workstation
    host_name          Client-Win-NCPA
    alias              Windows 10 Workstation (NCPA)
    address            192.168.13.32
}

define service{
    use                generic-service
    host_name          Client-Win-NCPA
    service_description CPU Usage
    check_command      check_ncpa!mytoken!'cpu/percent'!-w 80 -c 95
}

define service{
    use                generic-service
    host_name          Client-Win-NCPA
    service_description Memory Usage
    check_command      check_ncpa!mytoken!'memory/virtual'!-w 80 -c 90
}
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

```

define service{
    use                generic-service
    host_name           Client-Win-NCPA
    service_description C:\ Drive Space
    check_command
check_ncpa!mytoken!'disk/logical/C:|/used_percent'!-w 80 -c 90
}

define service{
    use                generic-service
    host_name           Client-Win-NCPA
    service_description System Uptime
    check_command       check_ncpa!mytoken!'system/uptime'
}

define service{
    use                generic-service
    host_name           Client-Win-NCPA
    service_description Active Processes
    check_command       check_ncpa!mytoken!'processes'!--warning=150 --
critical=200
}

define service{
    use                generic-service
    host_name           Client-Win-NCPA
    service_description Network Interface Status
    check_command       check_ncpa!mytoken!'interface'
}

```

Для Ubuntu Desktop (Serv-Linux) створіть файл
/usr/local/nagios/etc/objects/linux/serv-linux-ncpa.cfg з наступним вмістом:

```

define host{
    use                linux-server
    host_name           Serv-Linux-NCPA
    alias               Linux Test Server (NCPA)
    address             192.168.13.4
}

define service{
    use                generic-service
    host_name           Serv-Linux-NCPA
    service_description CPU Usage
    check_command       check_ncpa!mytoken!'cpu/percent'!-w 80 -c 95

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		70

```

}

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description Memory Usage
    check_command       check_ncpa!mytoken!'memory/virtual'!-w 80 -c 90
}

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description Root Partition
    check_command
check_ncpa!mytoken!'disk/logical/|/used_percent'!-w 80 -c 90
}

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description System Uptime
    check_command       check_ncpa!mytoken!'system/uptime'
}

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description Active Processes
    check_command       check_ncpa!mytoken!'processes'!--warning=150 --
critical=200
}

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description Network Interface Status
    check_command       check_ncpa!mytoken!'interface'
}

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description Apache Status

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росі́нський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

```

        check_command

check_ncpa!mytoken! 'processes|/count/name=apache2'! --warning=0:1 --critical=0:0
    }

define service{
    use                generic-service
    host_name          Serv-Linux-NCPA
    service_description MySQL Status
    check_command

check_ncpa!mytoken! 'processes|/count/name=mysqlld'! --warning=0:1 --critical=0:0
    }

```

Конфігурації для обох хостів включають перевірки основних системних ресурсів (процесор, пам'ять, дисковий простір), а також специфічні перевірки для відповідних операційних систем. Шлях до метрики в параметрі -М відповідає структурі API NCRA, де компоненти розділені символом '/' або '|' для вказання шляху до конкретного значення.

Після створення конфігурацій необхідно перевірити їх коректність та активувати нову конфігурацію:

```

bash

sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

sudo systemctl restart nagios

```

Використання NCRA для моніторингу клієнтських систем демонструє гнучкість системи Nagios Core та можливість інтеграції з сучасними інструментами моніторингу. Порівняно з традиційними агентами (NSClient++, NRPE), NCRA забезпечує більш уніфікований підхід до моніторингу різних операційних систем, спрощує процес конфігурації та надає додаткові можливості через веб-інтерфейс та API.

Таблиця 2.3 - Порівняння агентів моніторингу NCRA та традиційних рішень

Характеристика	NCRA	NSClient++	NRPE
Підтримувані ОС	Windows, Linux, macOS, Unix	Переважно Windows	Linux, Unix
Веб-інтерфейс	Вбудований	Відсутній	Відсутній

API	RESTful API з автентифікацією	Обмежений	Відсутній
Конфігурація	Єдиний формат для всіх платформ	Специфічний для Windows	Специфічний для Linux
Безпека	Токен автентифікації, підтримка TLS	Пароль, підтримка SSL	Обмежена (IP-фільтрація)
Графіки	Вбудовані	Відсутні	Відсутні
Пасивні перевірки	Вбудована підтримка	Обмежена підтримка	Відсутня підтримка
Розширюваність	Модульна система плагінів	Через скрипти	Через скрипти

Впровадження комбінованого підходу до моніторингу з використанням різних агентів для різних типів систем демонструє глибоке розуміння екосистеми Nagios та здатність обирати оптимальні інструменти для конкретних сценаріїв використання. NSClient++ забезпечує стабільний моніторинг серверних систем Windows з багаторічною історією розвитку та широкою підтримкою спільноти. NRPE представляє собою стандартне рішення для моніторингу Linux-серверів з глибокою інтеграцією з основними дистрибутивами. NCPA забезпечує сучасний підхід до моніторингу клієнтських систем з розширеними можливостями візуалізації та єдиним механізмом конфігурації для різних платформ.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

Розділ 3. Налаштування Спеціалізованого Моніторингу та Звітності

3.1. Моніторинг Поштового Сервера Postfix

Розширення функціональності системи моніторингу для забезпечення контролю над специфічними компонентами інфраструктури вимагає глибокого розуміння принципів роботи відповідних сервісів та протоколів взаємодії. Спеціалізований моніторинг фокусується на контролі критичних бізнес-сервісів, забезпечуючи не лише перевірку доступності, але й контроль функціональності та продуктивності. Моніторинг поштового сервера та віртуального комутатора представляє собою комплексну задачу, яка вимагає розуміння специфіки відповідних технологій та методів їх контролю. Імплементація механізмів сповіщення забезпечує оперативне реагування на інциденти, мінімізуючи час простою систем та потенційні втрати.

Архітектура спеціалізованого моніторингу в Nagios Core базується на принципі розширення базової функціональності через додаткові плагіни, адаптовані для контролю конкретних сервісів. Кожен сервіс має специфічні параметри та метрики, які потребують індивідуального підходу до моніторингу. Інтеграція спеціалізованих інструментів контролю в єдину систему моніторингу забезпечує комплексний підхід до управління ІТ-інфраструктурою, дозволяючи відстежувати взаємозв'язки між компонентами та виявляти потенційні проблеми до їх ескалації в критичні інциденти. Імплементація ефективного моніторингу поштового сервера представляє собою комплексну технічну задачу, яка виходить далеко за межі простої перевірки стану процесів. Електронна пошта залишається критичним комунікаційним каналом для більшості організацій, тому забезпечення стабільного функціонування поштової інфраструктури потребує багаторівневого

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

підходу до моніторингу. Сучасна методологія контролю поштових систем передбачає аналіз як зовнішньої доступності сервісів через стандартні протоколи (SMTP, IMAP, POP3), так і внутрішнього стану серверного програмного забезпечення, включаючи моніторинг поштових черг, швидкості обробки повідомлень та загальної продуктивності системи. Nagios Core надає широкий набір інструментів для реалізації комплексного моніторингу поштової інфраструктури з можливістю раннього виявлення потенційних проблем та автоматизації процесу сповіщення адміністраторів.

Розгортання поштового сервера на базі Serv-Linux (Ubuntu Desktop 22.04 LTS) розпочинається з встановлення необхідного програмного забезпечення. Postfix представляє собою високопродуктивний агент передачі пошти (Mail Transfer Agent, MTA), який забезпечує маршрутизацію та доставку електронних повідомлень. Dovecot доповнює функціональність Postfix, надаючи серверну реалізацію протоколів IMAP та POP3 для забезпечення доступу клієнтів до поштових скриньок. Mailutils надає набір утиліт командного рядка для роботи з електронною поштою, які спрощують процес тестування та адміністрування поштової системи.

Інсталяція необхідних компонентів виконується через систему управління пакетами APT:

```
bash
sudo apt install -y postfix dovecot-core dovecot-imapd dovecot-pop3d mailutils
```

Під час інсталяції пакету Postfix система запропонує вибрати тип конфігурації. Для корпоративної поштової системи оптимальним варіантом є "Internet Site", який налаштовує Postfix для прийому та відправки пошти безпосередньо через протокол SMTP. На запит "System mail name" необхідно вказати доменне ім'я, яке буде використовуватися для електронної пошти - в контексті віртуального середовища можна використовувати "khomenyuk.net". Вказаний домен визначає формат електронних адрес користувачів системи (наприклад, user@khomenyuk.net).

Після завершення базової інсталяції необхідно виконати додаткове налаштування Postfix для роботи в локальній мережі. Конфігурація Postfix

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

зберігається у файлі `/etc/postfix/main.cf`, який потребує модифікації наступних параметрів:

```
# Базові налаштування доменів та мережевих параметрів
myhostname = mail.khomenyk.net
mydomain = khomenyk.net
myorigin = $mydomain
inet_interfaces = all
inet_protocols = all

# Налаштування приймання пошти
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.13.0/26

# Формат зберігання пошти
home_mailbox = Maildir/
```

Параметр `inet_interfaces = all` дозволяє Postfix прослуховувати всі мережеві інтерфейси, що необхідно для приймання пошти з різних сегментів мережі. Директива `mydestination` визначає домени, для яких Postfix виступає кінцевим пунктом доставки пошти. Параметр `mynetworks` вказує мережі, з яких дозволено відправляти пошту без додаткової автентифікації, що критично важливо для запобігання використанню сервера як відкритого релея для спаму. Налаштування `home_mailbox = Maildir/` визначає формат зберігання пошти у вигляді структури каталогів Maildir, який забезпечує більшу надійність та ефективність порівняно з традиційним форматом `mbox`.

Конфігурація Dovecot вимагає модифікації декількох файлів для забезпечення коректної роботи з форматом Maildir та налаштування механізмів автентифікації. Файл `/etc/dovecot/conf.d/10-mail.conf` визначає параметри зберігання та доступу до поштових скриньок:

```
mail_location = maildir:~/Maildir
# Файл /etc/dovecot/conf.d/10-auth.conf налаштовує механізми автентифікації користувачів:
disable_plaintext_auth = no
auth_mechanisms = plain login
```

Параметр `disable_plaintext_auth = no` дозволяє використання автентифікації відкритим текстом, що спрощує тестування в локальному середовищі. У продуктивному середовищі рекомендується забезпечити шифрування з'єднань

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

через SSL/TLS та заборонити автентифікацію відкритим текстом для підвищення безпеки.

Файл `/etc/dovecot/conf.d/10-master.conf` визначає параметри взаємодії з іншими компонентами системи, включаючи Postfix:

```
service auth {  
    unix_listener /var/spool/postfix/private/auth {  
        mode = 0660  
        user = postfix  
        group = postfix  
    }  
}
```

Наведена конфігурація створює Unix-сокет для взаємодії між Dovecot та Postfix, що дозволяє використовувати механізми автентифікації Dovecot при відправці пошти через SMTP.

Після внесення змін до конфігураційних файлів необхідно перезапустити відповідні сервіси для застосування налаштувань:

```
bash  
sudo systemctl restart postfix  
sudo systemctl restart dovecot
```

Для перевірки коректності налаштувань поштової системи рекомендується створити тестового користувача та виконати відправку тестового повідомлення:

```
bash  
# Створення тестового користувача  
sudo adduser testuser  
  
# Відправка тестового повідомлення  
echo "Test message content" | mail -s "Test Subject" testuser@khomenyk.net  
  
# Перевірка отримання повідомлення  
sudo su - testuser -c "ls -la Maildir/new/"
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

Успішне отримання тестового повідомлення підтверджує коректність базової конфігурації поштової системи та готовність до налаштування моніторингу.

Базовий рівень моніторингу поштового сервера включає перевірку доступності ключових протоколів: SMTP для відправки та отримання пошти, IMAP та POP3 для доступу клієнтів до поштових скриньок. Методологія "чорної скриньки" передбачає виконання перевірок з перспективи зовнішнього користувача, без прямого доступу до внутрішньої структури системи, що дозволяє оцінити реальний рівень доступності сервісів для кінцевих користувачів.

Nagios Core надає стандартні плагіни для перевірки поштових протоколів: check_smtp для контролю SMTP-сервера, check_imap для IMAP та check_pop для POP3. Кожен з плагінів встановлює з'єднання з відповідним сервісом, виконує базовий обмін даними згідно з протоколом та вимірює час відгуку, що дозволяє не лише перевірити факт функціонування сервісу, але й оцінити його продуктивність.

Для налаштування моніторингу доступності поштових сервісів необхідно додати відповідні визначення до конфігураційного файлу /usr/local/nagios/etc/objects/linux/serv-linux.cfg:

```
define service{
    use                generic-service
    host_name          Serv-Linux
    service_description SMTP Service
    check_command       check_smtp!-p 25 -w 5 -c 10
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description IMAP Service
    check_command       check_imap!-p 143 -w 5 -c 10
}

define service{
    use                generic-service
    host_name          Serv-Linux
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

```

service_description POP3 Service
check_command      check_pop!-p 110 -w 5 -c 10
}

```

Кожне визначення сервісу включає посилання на шаблон generic-service, який визначає базові параметри моніторингу; ім'я хоста Serv-Linux, на якому розгорнуто поштовий сервер; опис сервісу, який відображається в інтерфейсі Nagios; та команду перевірки з відповідними параметрами. Параметр -p вказує порт для перевірки (25 для SMTP, 143 для IMAP, 110 для POP3), а параметри -w та -c визначають порогові значення часу відгуку в секундах для генерації попереджень та критичних сповіщень відповідно.

Для більш комплексної перевірки функціональності SMTP-сервера рекомендується додатково налаштувати моніторинг повного циклу відправки та отримання пошти. Плагін check_smtp_send дозволяє не лише перевірити доступність порту, але й виконати реальну відправку тестового повідомлення, перевіряючи коректність роботи всього ланцюга обробки пошти:

```

define command{
    command_name      check_smtp_send
    command_line      $USER1$/check_smtp_send -H $HOSTADDRESS$ -f
nagios@khomenyk.net -r testuser@khomenyk.net -S "Test from Nagios" -m "This is a
test message from Nagios" -w 10 -c 20
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description SMTP Mail Delivery
    check_command      check_smtp_send
    check_interval     30
    retry_interval     5
}

```

Параметри команди check_smtp_send включають: адресу хоста (-H), адресу відправника (-f), адресу отримувача (-r), тему повідомлення (-S), текст повідомлення (-m) та порогові значення часу відправки для генерації попереджень та критичних сповіщень (-w та -c). Збільшені значення параметрів check_interval

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		80

та `retry_interval` оптимізують частоту перевірок для мінімізації навантаження на поштову систему.

Глибинний моніторинг внутрішнього стану поштового сервера вимагає доступу до системи для виконання спеціалізованих команд діагностики. Ключовим аспектом моніторингу Postfix є стан поштових черг, який відображає ефективність процесу доставки повідомлень. Накопичення повідомлень у чергах `active`, `deferred` або `hold` може свідчити про різноманітні проблеми: мережеві збої, неправильну конфігурацію, блокування з боку віддалених серверів або спам-атаку.

Nagios Remote Plugin Executor (NRPE) забезпечує механізм віддаленого виконання команд на контрольованому хості з передачею результатів на сервер Nagios для аналізу та генерації сповіщень. Впровадження NRPE для моніторингу поштових черг дозволяє отримувати детальну інформацію про внутрішній стан Postfix без необхідності надання прямого доступу до системи з боку сервера моніторингу.

Процес налаштування моніторингу поштових черг включає встановлення та конфігурацію NRPE на поштовому сервері, створення спеціалізованої команди для перевірки стану черг та налаштування відповідного сервісу в Nagios.

Інсталяція NRPE та необхідних плагінів на Serv-Linux виконується через систему управління пакетами:

```
bash
```

```
sudo apt install -y nagios-nrpe-server nagios-plugins
```

Пакет `nagios-nrpe-server` встановлює демон NRPE, який прослуховує вхідні з'єднання від сервера Nagios та виконує запитані перевірки, а `nagios-plugins` надає набір стандартних плагінів для моніторингу різних аспектів системи, включаючи плагін `check_mailq` для аналізу поштових черг.

Конфігурація NRPE вимагає модифікації файлу `/etc/nagios/nrpe.cfg` для визначення дозволених хостів та команд перевірки:

```
# Базові параметри NRPE
server_address=192.168.13.4
server_port=5666
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		81

```

allowed_hosts=127.0.0.1,::1,192.168.13.2

# Параметри безпеки
dont_blame_nrpe=0
allow_bash_command_substitution=0

# Команди перевірки
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c
30,25,20
command[check_disk]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /
command[check_procs]=/usr/lib/nagios/plugins/check_procs -w 250 -c 400
command[check_mailq]=/usr/lib/nagios/plugins/check_mailq -w 10 -c 50

```

Параметр `allowed_hosts` визначає IP-адреси, з яких дозволено підключення до NRPE-сервера, що є критично важливим для забезпечення безпеки системи. Значення `dont_blame_nrpe=0` забороняє передачу аргументів з сервера Nagios до плагінів, що запобігає потенційним атакам через ін'єкцію команд. Команда `check_mailq` використовує стандартний плагін для перевірки поштових черг з пороговими значеннями 10 повідомлень для генерації попередження та 50 повідомлень для критичного сповіщення. Плагін `check_mailq` автоматично визначає тип поштового сервера (Postfix, Sendmail, Qmail) та використовує відповідні команди для отримання інформації про стан черг.

Після внесення змін до конфігурації необхідно перезапустити NRPE-сервер для застосування налаштувань:

```

bash
sudo systemctl restart nagios-nrpe-server

```

Для перевірки коректності конфігурації NRPE можна використати команду `check_nrpe` на сервері Nagios:

```

bash
/usr/local/nagios/libexec/check_nrpe -H 192.168.13.4 -c check_mailq

```

Після успішної перевірки функціональності NRPE необхідно налаштувати відповідний сервіс в Nagios, додавши наступне визначення до файлу `/usr/local/nagios/etc/objects/linux/serv-linux.cfg`:

```

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Postfix Mail Queue
    check_command      check_nrpe!check_mailq
}

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

Команда `check_nrpe!check_mailq` виконує запит до NRPE-сервера на хості Serv-Linux з вказівкою виконати команду `check_mailq`, визначену в конфігурації NRPE. Результат виконання команди аналізується Nagios для визначення стану сервісу та генерації відповідних сповіщень при перевищенні порогових значень.

Комплексний моніторинг поштового сервера включає не лише перевірку доступності сервісів та стану черг, але й аналіз продуктивності системи та моніторинг журналів подій для виявлення потенційних проблем. Високе навантаження на процесор, недостатня кількість оперативної пам'яті або вільного місця на диску можуть суттєво впливати на продуктивність поштового сервера, призводячи до затримок у доставці повідомлень або відмов у обслуговуванні.

Для моніторингу системних ресурсів доцільно використовувати стандартні плагіни NRPE, додавши відповідні сервіси до конфігурації Nagios:

```
define service{
    use                generic-service
    host_name          Serv-Linux
    service_description CPU Load
    check_command       check_nrpe!check_load
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Memory Usage
    check_command       check_nrpe!check_mem!80!90
}

define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Disk Space
    check_command       check_nrpe!check_disk
}
```

Для моніторингу використання оперативної пам'яті необхідно додати відповідну команду до конфігурації NRPE на поштовому сервері:

```
command[check_mem]=/usr/lib/nagios/plugins/check_memory.sh -w $ARG1$ -c $ARG2$
```

Та створити скрипт `/usr/lib/nagios/plugins/check_memory.sh` з наступним вмістом:

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

bash
#!/bin/bash
# Nagios plugin to check memory usage

while getopts 'w:c:' OPT; do
    case $OPT in
        w) WARN=$OPTARG ;;
        c) CRIT=$OPTARG ;;
    esac
done

if [ -z "$WARN" ] || [ -z "$CRIT" ]; then
    echo "UNKNOWN: Missing thresholds"
    exit 3
fi

MEM_TOTAL=$(free -m | grep 'Mem:' | awk '{print $2}')
MEM_FREE=$(free -m | grep 'Mem:' | awk '{print $4}')
MEM_CACHED=$(free -m | grep 'Mem:' | awk '{print $6}')
MEM_BUFFERS=$(free -m | grep 'Mem:' | awk '{print $5}')
MEM_USED=$((MEM_TOTAL - MEM_FREE - MEM_CACHED - MEM_BUFFERS))
MEM_PERCENT=$((MEM_USED * 100 / MEM_TOTAL))

OUTPUT="Memory      usage:      $MEM_PERCENT%      ($MEM_USED      MB      of      $MEM_TOTAL      MB)      |
'usage'=$MEM_PERCENT%;$WARN;$CRIT;0;100"

if [ $MEM_PERCENT -ge $CRIT ]; then
    echo "CRITICAL: $OUTPUT"
    exit 2
elif [ $MEM_PERCENT -ge $WARN ]; then
    echo "WARNING: $OUTPUT"
    exit 1
else
    echo "OK: $OUTPUT"
    exit 0
fi

```

Не менш важливим аспектом моніторингу поштового сервера є аналіз журналів подій для виявлення помилок, спроб несанкціонованого доступу або інших аномалій. Для реалізації моніторингу логів можна використовувати плагін check_log, який аналізує журнали на наявність специфічних паттернів:

```

command[check_mail_log]=/usr/lib/nagios/plugins/check_log      -F
/var/log/mail.log -O /tmp/mail.log.old -q fatal -q error -q disconnect
define service{
    use                generic-service
    host_name          Serv-Linux
    service_description Mail Log Errors

```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

```

    check_command      check_nrpe!check_mail_log
    check_interval     10
    retry_interval     2
}

```

Плагін check_log порівнює поточний стан файлу журналу з попередньо збереженою копією та виявляє нові записи, які містять вказані ключові слова. У наведеному прикладі перевіряється файл /var/log/mail.log на наявність записів з рівнем критичності "fatal", "error" або записів, пов'язаних з роз'єднанням клієнтів.

Таблиця 3.1 - Порівняння методів моніторингу поштового сервера

Метод моніторингу	Переваги	Недоліки	Реалізація в Nagios
Перевірка доступності портів	Простота реалізації, мінімальне навантаження	Обмежена інформативність, виявлення лише повних відмов	check_smtp, check_imap, check_pop
Функціональне тестування	Перевірка реальної функціональності сервісів	Більше навантаження на систему, складніша реалізація	check_smtp_send
Моніторинг черг	Виявлення проблем з доставкою до появи скарг	Потребує прямого доступу до системи	check_mailq через NRPE
Аналіз продуктивності	Виявлення причин зниження швидкодії	Складність встановлення порогових значень	check_load, check_mem через NRPE
Моніторинг логів	Виявлення прихованих проблем та аномалій	Великий обсяг даних для аналізу	check_log через NRPE

Впровадження комплексного підходу до моніторингу поштового сервера, який поєднує перевірку доступності сервісів, функціональне тестування, контроль поштових черг, моніторинг продуктивності системи та аналіз журналів подій, забезпечує раннє виявлення потенційних проблем та мінімізацію впливу збоїв на користувачів. Інтеграція різноманітних методів моніторингу в єдину систему Nagios Core дозволяє централізовано контролювати стан поштової інфраструктури та автоматизувати процес сповіщення адміністраторів про виникнення нестандартних ситуацій.

3.2. Моніторинг Віртуального Комутатора та Топології Мережі

Імплементація ефективного моніторингу мережевої інфраструктури в віртуалізованому середовищі вимагає нестандартних підходів, оскільки традиційні методи контролю, застосовні до фізичного обладнання, виявляються неефективними при роботі з віртуальними комутаторами. Віртуальний комутатор NAT Network у середовищі Oracle VirtualBox представляє собою програмну абстракцію, яка не підтримує стандартні протоколи управління мережевим обладнанням, такі як SNMP (Simple Network Management Protocol), що суттєво ускладнює процес моніторингу. Методологія прямого контролю стану віртуального комутатора трансформується в концепцію моніторингу зв'язності та логічної топології мережі, що забезпечує значно більшу інформативність та практичну цінність у контексті віртуального середовища.

Концептуальним фундаментом цього підходу виступає трансформація системи моніторингу від простої колекції незалежних перевірок до інтелектуальної платформи, яка "розуміє" архітектурні взаємозв'язки між компонентами інфраструктури. Визначення ієрархічних залежностей між мережевими елементами дозволяє системі моніторингу моделювати поведінку мережевої інфраструктури при виникненні збоїв, забезпечуючи точне визначення кореневої причини проблеми замість генерації численних сповіщень про вторинні ефекти. У контексті Nagios Core це реалізується через механізм "батьківства"

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		86

(parent-child relationships), який визначає логічні залежності між хостами та впливає на процес визначення стану пристроїв при виникненні проблем зв'язності.

Крок 1: Створення хоста для шлюзу

Першим етапом впровадження моніторингу топології мережі є визначення центрального елемента інфраструктури — шлюзу NAT Network, який забезпечує з'єднання віртуальної мережі з зовнішнім світом. Віртуальний шлюз, створений VirtualBox при налаштуванні NAT Network, має фіксовану IP-адресу 192.168.13.1 та виконує функції маршрутизатора і DNS-форвардера для всіх пристроїв у віртуальній мережі. Незважаючи на те, що цей шлюз не підтримує протоколи управління, він доступний для базових перевірок зв'язності через ICMP (ping), що дозволяє відстежувати його статус.

Для включення шлюзу до системи моніторингу необхідно створити конфігураційний файл /usr/local/nagios/etc/objects/network/snm-gw.cfg з наступним вмістом:

```
define host{
    use                generic-switch      ; Використовуємо шаблон для мережевих
пристроїв
    host_name          SNM-GW
    alias              VirtualBox NAT Network Gateway
    address             192.168.13.1
    check_command       check_host_alive
    hostgroups          switches           ; Додаємо до групи мережевих
пристроїв
}

define service{
    use                generic-service
    host_name          SNM-GW
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}
```

Використання шаблону generic-switch для визначення хоста забезпечує застосування оптимальних параметрів моніторингу для мережевого обладнання, включаючи частоту перевірок та механізми сповіщення. Команда check_host_alive виконує базову перевірку доступності пристрою через ICMP, а сервіс PING

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		87

забезпечує детальний моніторинг часу відгуку та втрати пакетів з відповідними пороговими значеннями для генерації попереджень та критичних сповіщень.

Для групування мережевих пристроїв та спрощення управління конфігурацією необхідно визначити групу хостів switches. Якщо файл `/usr/local/nagios/etc/objects/hostgroups.cfg` не існує, його необхідно створити з наступним вмістом:

```
define hostgroup{
    hostgroup_name switches
    alias          Network Switches
}
```

Групування хостів за функціональним призначенням спрощує навігацію в інтерфейсі Nagios та забезпечує можливість виконання масових операцій над групами пристроїв, таких як планове відключення сповіщень під час технічного обслуговування.

Ключовим аспектом моделювання мережевої топології в Nagios Core є визначення ієрархічних залежностей між хостами, які відображають реальні шляхи комунікації в мережевій інфраструктурі. Механізм "батьківства" (parent-child relationships) дозволяє системі моніторингу "розуміти", що доступність певного хоста залежить від доступності іншого хоста, через який проходить маршрут комунікації. У контексті віртуальної інфраструктури VirtualBox всі віртуальні машини, підключені до NAT Network, залежать від доступності віртуального шлюзу, оскільки він забезпечує єдиний шлях комунікації між пристроями.

Для визначення залежностей необхідно модифікувати конфігурації всіх хостів, додавши директиву `parents` з вказанням імені батьківського хоста. Модифікація виконується для всіх файлів конфігурації хостів:

У файлі `/usr/local/nagios/etc/objects/windows/serv-domain.cfg` до секції `define host` додайте:

```
parents    SNM-GW
```

У файлі `/usr/local/nagios/etc/objects/workstation/client-win.cfg` до секції `define host` додайте:

```
parents    SNM-GW
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		88

У файлі /usr/local/nagios/etc/objects/linux/serv-linux.cfg до секції define host додайте:

```
parents      SNM-GW
```

У файлі /usr/local/nagios/etc/objects/linux/serv-nagios.cfg до секції define host додайте:

```
parents      SNM-GW
```

Визначення залежностей для всіх хостів інфраструктури забезпечує коректне відображення мережевої топології та механізм інтелектуального визначення стану пристроїв при виникненні проблем зв'язності. При втраті зв'язку з віртуальним шлюзом SNM-GW система моніторингу змінить стан залежних хостів на UNREACHABLE (недосяжний) замість DOWN (відмова), що відображає реальну ситуацію: хост не може бути перевірений через недоступність проміжного пристрою, а не через власну відмову.

Механізм визначення стану хостів в Nagios Core базується на алгоритмі, який враховує результати перевірок та визначені залежності. Якщо хост має статус UP, але його батьківський хост має статус DOWN або UNREACHABLE, Nagios встановлює статус хоста як UNREACHABLE, відображаючи неможливість достовірної перевірки його стану. Це забезпечує точне визначення кореневої причини проблеми та запобігає генерації надмірної кількості сповіщень при відмові критичних компонентів інфраструктури.

Після налаштування залежностей між хостами необхідно перевірити коректність конфігурації та перезапустити Nagios для застосування змін:

```
bash
```

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
sudo systemctl restart nagios
```

Для візуалізації визначеної топології мережі Nagios Core надає інструмент "Map" (в новіших версіях "Legacy Map"), доступний через веб-інтерфейс у розділі "Maps". Цей інструмент генерує графічне представлення хостів та їх взаємозв'язків, базуючись на визначених залежностях. У центрі відображається

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		89

хост SNM-GW, а всі залежні хости розташовуються навколо нього, з'єднані лініями, які відображають відносини "батько-нащадок".

Візуалізація топології не лише забезпечує зручне представлення архітектури мережі, але й відображає поточний стан пристроїв, використовуючи кольорове кодування: зелений для UP, червоний для DOWN, помаранчевий для UNREACHABLE, жовтий для PENDING. При виникненні проблем зв'язності ця візуалізація дозволяє швидко ідентифікувати кореневу причину збою та визначити впливи на залежні системи.

Для перевірки коректності налаштування залежностей та демонстрації інтелектуального визначення кореневої причини проблеми можна провести тестове відключення віртуального шлюзу. Це можна реалізувати через тимчасове вимкнення NAT Network у налаштуваннях VirtualBox або через імітацію недоступності шлюзу за допомогою брандмауера на сервері Nagios:

```
bash
```

```
sudo iptables -A OUTPUT -d 192.168.13.1 -j DROP
```

Після імітації відмови шлюзу система моніторингу відобразить статус DOWN для хоста SNM-GW та статус UNREACHABLE для всіх залежних хостів. Важливим аспектом є те, що Nagios згенерує сповіщення лише для зміни стану шлюзу, не створюючи інформаційного шуму через численні сповіщення про недосяжність залежних систем. Після відновлення зв'язку з шлюзом (видалення правила брандмауера або відновлення NAT Network) система моніторингу автоматично відновить статуси всіх хостів до UP при успішному проходженні перевірок.

```
bash
```

```
sudo iptables -D OUTPUT -d 192.168.13.1 -j DROP
```

Для забезпечення комплексного моніторингу мережевої інфраструктури рекомендується доповнити базову перевірку доступності шлюзу додатковими перевітками, які забезпечують контроль ключових аспектів функціонування мережі: маршрутизації, DNS-резолвінгу та пропускну здатності. Моніторинг маршрутизації можна реалізувати за допомогою трасування маршруту до

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		90

зовнішніх ресурсів, що дозволяє виявляти проблеми на рівні провайдера або зовнішніх шлюзів:

```
define command{
    command_name    check_traceroute

    command_line    $USER1$/check_traceroute -H $ARG1$ -w $ARG2$ -c $ARG3$
}

define service{
    use              generic-service

    host_name        SNM-GW

    service_description Traceroute to Google DNS

    check_command    check_traceroute!8.8.8.8!10!15
}
```

Моніторинг DNS-резолвінгу забезпечує контроль функціональності системи розв'язання імен, яка є критичним компонентом для більшості мережесервісів:

```
define command{
    command_name    check_dns

    command_line    $USER1$/check_dns -H $ARG1$ -s $HOSTADDRESS$
}

define service{
    use              generic-service

    host_name        SNM-GW

    service_description DNS Resolution

    check_command    check_dns!google.com
}
```

Моніторинг пропускну́ї здатності та затримок можна реалізувати через періодичне вимірювання часу відгуку до ключових зовнішніх ресурсів:

```
define service{
    use              generic-service

    host_name        SNM-GW

    service_description Latency to Google

    check_command    check_ping!100.0,20%!500.0,60%!google.com
}
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		91

Таблиця 3.2 - Порівняння методів моніторингу мережевої інфраструктури

Метод моніторингу	Переваги	Недоліки	Застосування
Прямий моніторинг через SNMP	Детальна інформація про стан пристрою	Неможливий для віртуальних комутаторів	Фізичні мережеві пристрої
Моніторинг доступності	Простота реалізації, мінімальні вимоги	Обмежена інформативність	Базовий контроль стану
Моніторинг топології	Визначення кореневої причини, зменшення шуму	Вимагає точного моделювання залежностей	Комплексні мережеві інфраструктури
Моніторинг продуктивності	Виявлення деградації до повної відмови	Складність встановлення порогових значень	Критичні мережеві з'єднання
Моніторинг сервісів	Функціональна перевірка реальних сценаріїв	Більше навантаження на систему	Критичні бізнес-процеси

Впровадження моніторингу віртуального комутатора та топології мережі демонструє трансформацію підходу до контролю інфраструктури від простої перевірки доступності до інтелектуального аналізу взаємозв'язків та кореневих причин проблем. Визначення ієрархічних залежностей між компонентами інфраструктури забезпечує не лише точну ідентифікацію джерела збоїв, але й оптимізацію процесу сповіщення, запобігаючи інформаційному перевантаженню адміністраторів при виникненні масштабних проблем. Ця методологія моніторингу формує основу для проактивного управління ІТ-інфраструктурою, дозволяючи фокусуватися на кореневих причинах проблем замість боротьби з їх наслідками.

3.3. Конфігурація Системи Сповіщень

Система сповіщень представляє критичний компонент інфраструктури моніторингу, який трансформує пасивний збір даних у активний механізм реагування на інциденти. Функціональна ефективність моніторингової платформи визначається не лише точністю ідентифікації проблем, але й швидкістю передачі інформації відповідальним фахівцям для оперативного вирішення виявлених збоїв. Nagios Core надає гнучкий механізм сповіщень, який підтримує різноманітні канали комунікації, включаючи електронну пошту, SMS, месенджери та інтеграцію з системами управління інцидентами. Конфігурація

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		92

системи сповіщень потребує комплексного підходу, який охоплює налаштування транспортного механізму доставки повідомлень, визначення отримувачів сповіщень, встановлення критеріїв генерації повідомлень та формування змістовних шаблонів, які містять актуальну інформацію про інцидент.

Електронна пошта залишається найбільш універсальним та надійним каналом доставки сповіщень, який забезпечує документування історії інцидентів та можливість інтеграції з різноманітними системами обробки повідомлень. Для забезпечення відправки електронних сповіщень Nagios Core потребує доступу до SMTP-сервера, який може приймати та маршрутизувати вихідні повідомлення. Оптимальним рішенням у контексті віртуального середовища є встановлення локального поштового релея на сервері моніторингу, який забезпечить передачу повідомлень до зовнішніх поштових систем.

Postfix представляє собою високопродуктивний та безпечний агент передачі пошти (Mail Transfer Agent, MTA), який оптимально відповідає вимогам системи сповіщень Nagios Core. Встановлення та базове налаштування Postfix на сервері Serv-Nagios виконується через систему управління пакетами APT:

```
bash
sudo apt install -y postfix mailutils
```

Під час інсталяції пакету Postfix система запропонує вибрати тип конфігурації. Для сервера моніторингу, який виступає джерелом вихідних повідомлень, оптимальним варіантом є "Internet Site", який налаштовує Postfix для прямої відправки пошти через протокол SMTP. На запит "System mail name" необхідно вказати повне доменне ім'я (FQDN) сервера, наприклад, "nagios.khomenyuk.net". Вказаний домен визначає значення поля "From" у вихідних повідомленнях та впливає на успішність доставки пошти через зовнішні сервери.

Параметр `inet_interfaces = loopback-only` обмежує прослуховування SMTP-порту лише локальним інтерфейсом, що підвищує безпеку системи, запобігаючи прийому пошти з зовнішніх джерел. Директива `relayhost` вказує SMTP-сервер, через який будуть відправлятися всі вихідні повідомлення. У реальному середовищі це може бути корпоративний поштовий сервер або зовнішній

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		93

провайдер, такий як Gmail або Office 365. Параметри `smtp_sasl_auth_enable` та `smtp_sasl_password_maps` налаштовують автентифікацію на релейному сервері, що необхідно для більшості сучасних поштових систем.

Успішне отримання тестового повідомлення підтверджує коректність конфігурації поштового релея та готовність до інтеграції з системою сповіщень Nagios Core.

Механізм сповіщень в Nagios Core базується на концепції контактів та груп контактів, які визначають отримувачів повідомлень про інциденти. Контакт представляє собою об'єкт, який містить інформацію про конкретного отримувача: ім'я, адресу електронної пошти, періоди доступності та типи сповіщень, які повинні бути відправлені. Групи контактів об'єднують кілька контактів для спрощення управління та забезпечення масової розсилки сповіщень.

Визначення контакту `nagiosadmin` включає базові параметри сповіщень: періоди доступності (`service_notification_period` та `host_notification_period`), які визначають проміжки часу, коли сповіщення можуть бути відправлені; опції сповіщень (`service_notification_options` та `host_notification_options`), які вказують типи подій, про які слід сповіщати; та команди сповіщень (`service_notification_commands` та `host_notification_commands`), які визначають механізм доставки повідомлень.

Опції сповіщень для сервісів включають: `w` (warning) - попередження, `u` (unknown) - невідомий стан, `c` (critical) - критична помилка, `r` (recovery) - відновлення, `f` (flapping) - нестабільний стан. Опції сповіщень для хостів включають: `d` (down) - недоступність, `u` (unreachable) - недосяжність, `r` (recovery) - відновлення, `f` (flapping) - нестабільний стан.

Директива `use generic-contact` включає посилання на шаблон контакту, який визначає базові параметри сповіщень. Шаблон `generic-contact` зазвичай визначений у файлі `/usr/local/nagios/etc/objects/templates.cfg` та може бути модифікований для встановлення глобальних параметрів сповіщень.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		94

Для більш складних сценаріїв сповіщень можна створити додаткові контакти та групи контактів, які відображають організаційну структуру підприємства та розподіл відповідальності за різні компоненти інфраструктури:

```
define contact{
    contact_name      network-admin
    use                generic-contact
    alias              Network Administrator
    email              network@example.com
    service_notification_period    workhours
    host_notification_period       workhours
    service_notification_options   w,u,c,r
    host_notification_options      d,u,r
    service_notification_commands  notify-service-by-email
    host_notification_commands     notify-host-by-email
}
```

```
define contact{
    contact_name      server-admin
    use                generic-contact
    alias              Server Administrator
    email              server@example.com
    service_notification_period    24x7
    host_notification_period       24x7
    service_notification_options   w,u,c,r
    host_notification_options      d,u,r
    service_notification_commands  notify-service-by-email
    host_notification_commands     notify-host-by-email
}
```

```
define contactgroup{
    contactgroup_name    network-team
    alias                Network Administration Team
    members               nagiosadmin,network-admin
}
```

```
define contactgroup{
    contactgroup_name    server-team
    alias                Server Administration Team
    members               nagiosadmin,server-admin
}
```

Наведена конфігурація визначає спеціалізовані контакти для адміністраторів мережі та серверних систем з різними періодами доступності: мережеві адміністратори отримують сповіщення лише в робочий час (workhours), тоді як серверні адміністратори доступні цілодобово (24x7). Групи контактів network-team та server-team об'єднують відповідних фахівців для призначення відповідальності за конкретні компоненти інфраструктури.

Після визначення контактів та груп контактів необхідно призначити їх хостам та сервісам для вказання, хто повинен отримувати сповіщення про конкретні компоненти інфраструктури. Призначення контактів може виконуватися на рівні індивідуальних об'єктів або через шаблони для забезпечення уніфікованого підходу до великої кількості однотипних систем.

Для призначення групи контактів admins до хоста Serv-Domain необхідно додати директиву contact_groups до його визначення у файлі /usr/local/nagios/etc/objects/windows/serv-domain.cfg:

```
define host{
    use                windows-server
    host_name          Serv-Domain
    alias              Windows Domain Controller
    address            192.168.13.3
    parents            SNM-GW
    contact_groups     admins
}
```

Аналогічно, для призначення групи контактів до конкретного сервісу, який потребує особливої уваги, необхідно додати директиву contact_groups до його визначення:

```
define service{
    use                generic-service
    host_name          Serv-Domain
    service_description DNS Service
    check_command      check_nt!SERVICESTATE!-d SHOWALL -l DNS -s
P@ssword2023
    contact_groups     admins,network-team
}
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		96

Для забезпечення ефективного розподілу відповідальності та мінімізації інформаційного шуму рекомендується структурувати призначення контактів відповідно до організаційної ієрархії та функціональних обов'язків:

- Базові контакти (nagiosadmin) призначаються через шаблони до всіх об'єктів моніторингу для забезпечення загального контролю
- Спеціалізовані групи контактів (network-team, server-team) призначаються до відповідних категорій систем
- Критичні сервіси отримують додаткові контакти для забезпечення швидкого реагування на інциденти

Для автоматизації призначення контактів у масштабних інфраструктурах доцільно модифікувати базові шаблони хостів та сервісів, додавши директиву contact_groups з відповідними значеннями. Наприклад, модифікація шаблону windows-server у файлі /usr/local/nagios/etc/objects/templates/host-templates.cfg:

```
define host{
    name                windows-server
    use                  generic-host
    check_period         24x7
    check_interval       5
    retry_interval       1
    max_check_attempts   10
    check_command         check-host-alive
    notification_period   workhours
    notification_interval 30
    notification_options  d,u,r
    contact_groups        admins,server-team
    register              0
}
```

Така конфігурація забезпечить автоматичне призначення груп контактів admins та server-team до всіх хостів, які використовують шаблон windows-server, без необхідності модифікації індивідуальних визначень.

Після завершення конфігурації системи сповіщень необхідно виконати комплексне тестування для перевірки коректності роботи всіх компонентів: генерації подій, визначення стану об'єктів, формування сповіщень та доставки повідомлень відповідним контактам. Тестування дозволяє виявити потенційні

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		97

проблеми та оптимізувати конфігурацію до виникнення реальних інцидентів у продуктивному середовищі.

Успішне завершення всіх етапів тестування підтверджує коректність конфігурації системи сповіщень та готовність до функціонування в продуктивному режимі.

Стандартні шаблони сповіщень Nagios Core надають базову інформацію про інциденти, але можуть бути оптимізовані для підвищення інформативності та зручності використання. Налаштування шаблонів дозволяє включити додаткову інформацію про об'єкт, контекст проблеми та рекомендації щодо вирішення, що спрощує процес діагностики та зменшує час реагування на інциденти.

Шаблони сповіщень визначаються через команди сповіщень у файлі `/usr/local/nagios/etc/objects/commands.cfg`. Стандартні команди `notify-host-by-email` та `notify-service-by-email` можуть бути модифіковані для включення додаткової інформації:

```
define command{
    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios Monitor Alert
*****\n\nType: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n\nNotes:
$HOSTNOTES$\nNotes URL: $HOSTNOTESURL$\n\nView in Nagios:
http://nagios.khomenyk.net/nagios/cgi-bin/status.cgi?host=$HOSTNAME$" |
/usr/bin/mail -s "*** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ *"
$CONTACTEMAIL$
}
```

```
define command{
    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios Monitor Alert
*****\n\nType: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nService:
$SERVICEDESC$\nState: $SERVICESTATE$\nAddress: $HOSTADDRESS$\nInfo:
$SERVICEOUTPUT$\n\nDate/Time: $LONGDATETIME$\n\nNotes: $SERVICENOTES$\nNotes URL:
$SERVICENOTESURL$\n\nView in Nagios: http://nagios.khomenyk.net/nagios/cgi-
bin/extinfo.cgi?type=2&host=$HOSTNAME$&service=$SERVICEDESC$" | /usr/bin/mail -s
*** $NOTIFICATIONTYPE$ Service Alert: $HOSTNAME$/$SERVICEDESC$ is $SERVICESTATE$
***" $CONTACTEMAIL$
}
```

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		98

Модифіковані шаблони включають додаткові поля: Notes та Notes URL для відображення приміток та посилань, які можуть бути визначені для хостів та сервісів; прямі посилання на відповідні сторінки в інтерфейсі Nagios для швидкого доступу до детальної інформації; а також форматування повідомлення для підвищення читабельності. Додатково можна включити історію попередніх станів, графіки продуктивності та інші дані, які допомагають в діагностиці проблеми.

Для використання модифікованих шаблонів з усіма хостами та сервісами необхідно переконатися, що відповідні команди сповіщень вказані у визначеннях контактів або шаблонах контактів.

Таблиця 3.3 - Порівняння методів сповіщення в Nagios Core

Метод сповіщення	Переваги	Недоліки	Застосування
Електронна пошта	Універсальність, документування, легка інтеграція	Можливі затримки, потенційні проблеми доставки	Базовий метод для всіх типів сповіщень
SMS (через шлюзи)	Висока ймовірність швидкого отримання	Обмежений обсяг інформації, додаткові витрати	Критичні сповіщення
Месенджери (через API)	Оперативність, можливість групових сповіщень	Складність інтеграції, залежність від сервісів	Оперативні сповіщення для команд
Системи тікетів (JIRA, RT)	Автоматизація процесу обробки інцидентів	Складна інтеграція, додаткові системи	Корпоративні середовища з ITSM

Впровадження ефективної системи сповіщень трансформує Nagios Core з пасивного інструменту моніторингу в активну систему раннього виявлення та

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		99

реагування на інциденти. Комбінація гнучкої конфігурації контактів, оптимізованих шаблонів повідомлень та надійного транспортного механізму забезпечує своєчасне інформування відповідальних фахівців про проблеми в інфраструктурі, що мінімізує час простою систем та потенційні втрати бізнесу.

Імплементація комплексної системи моніторингу на базі Nagios Core у віртуалізованому середовищі демонструє потужність та гнучкість відкритих рішень для забезпечення контролю над гетерогенною інфраструктурою. Розроблена система охоплює всі ключові аспекти моніторингу: від базового контролю доступності до спеціалізованого моніторингу критичних сервісів, інтелектуального аналізу мережевої топології та ефективного механізму сповіщень. Методологічний підхід до проєктування, встановлення та конфігурації компонентів забезпечує стабільність функціонування системи та можливість подальшого масштабування відповідно до зростаючих потреб інфраструктури.

Практична реалізація проєкту охоплювала комплекс взаємопов'язаних технічних завдань, кожне з яких вносило вклад у формування цілісної системи моніторингу. Методичний підхід до виконання проєкту забезпечив логічну послідовність етапів, де кожен наступний крок базувався на результатах попереднього, утворюючи завершений технологічний цикл.

Створення віртуального стенду в середовищі Oracle VirtualBox з імітацією корпоративної мережевої інфраструктури заклало фундамент для подальшого розгортання системи моніторингу. Ретельне проєктування мережевої архітектури, включаючи схему IP-адресації, налаштування NAT Network та конфігурацію віртуальних машин, забезпечило стабільну платформу для тестування та демонстрації функціональності Nagios Core. Віртуальний стенд включив різноманітні компоненти: сервер моніторингу на базі Ubuntu Server 22.04 LTS, контролер домену на Windows Server 2022, робочу станцію Windows 10 Pro та тестовий сервер на Ubuntu Desktop 22.04 LTS, що відображає типову гетерогенну інфраструктуру сучасного підприємства.

Встановлення та налаштування Nagios Core у версії 4.5.9 продемонструвало методологію розгортання моніторингової платформи з вихідних кодів, що

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		100

забезпечує повний контроль над процесом інсталяції та оптимізацію конфігурації відповідно до конкретних вимог середовища. Структурований підхід до організації конфігураційних файлів через систему каталогів та шаблонів створив основу для ефективного управління великою кількістю об'єктів моніторингу, спрощуючи процес додавання нових хостів та сервісів без необхідності модифікації глобальних налаштувань.

Імплементація моніторингу різнотипових хостів з використанням спеціалізованих агентів для кожної операційної системи продемонструвала глибоке розуміння екосистеми Nagios та методологій інтеграції з різноманітними платформами. NSClient++ забезпечив стабільний моніторинг Windows-серверів з багаторічною історією розвитку та підтримкою. NRPE представив стандартне рішення для моніторингу Linux-систем з глибокою інтеграцією з основними дистрибутивами. NCPA забезпечив сучасний підхід до моніторингу клієнтських систем з розширеними можливостями візуалізації та уніфікованим механізмом конфігурації для різних операційних систем.

Розгортання спеціалізованого моніторингу поштового сервера Postfix продемонструвало комплексний підхід до контролю критичних бізнес-сервісів. Комбінація "чорноскринькових" перевірок доступності протоколів (SMTP, IMAP, POP3) та аналізу внутрішнього стану через моніторинг поштових черг забезпечила всебічний контроль функціональності поштової системи. Впровадження моніторингу журналів подій та продуктивності системи доповнило картину, створюючи повний профіль стану поштового сервера.

Інноваційний підхід до моніторингу віртуального комутатора через аналіз топології мережі та визначення залежностей між хостами трансформувал пасивний моніторинг у інтелектуальну систему аналізу кореневих причин збоїв. Визначення ієрархічних відносин "батько-нащадок" дозволило системі моніторингу розуміти взаємозв'язки між компонентами інфраструктури та мінімізувати інформаційний шум при виникненні масштабних проблем, фокусуючи увагу адміністраторів на джерелі збою, а не його наслідках.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		101

Конфігурація системи сповіщень через електронну пошту з використанням Postfix як поштового релея забезпечила механізм оперативного інформування відповідальних фахівців про інциденти. Гнучке визначення контактів та груп контактів з диференційованими періодами доступності та типами сповіщень дозволило адаптувати систему до організаційної структури підприємства та розподілу відповідальності. Оптимізація шаблонів повідомлень підвищила інформативність сповіщень, включивши контекстну інформацію та посилання на детальні дані в інтерфейсі Nagios.

Розроблена система моніторингу демонструє баланс між функціональністю, гнучкістю та складністю, забезпечуючи ефективний контроль над різноманітними компонентами інфраструктури без надмірного навантаження на ресурси системи або адміністративного персоналу. Аналіз результатів проєкту дозволяє виділити ключові переваги та обмеження впровадженого рішення, а також визначити потенційні напрямки подальшого розвитку.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		102

Висновки

Проектування та практична реалізація системи моніторингу мережевої інфраструктури на базі Nagios Core демонструє ефективність та гнучкість відкритих програмних рішень для забезпечення контролю над гетерогенними ІТ-середовищами. Комплексний підхід до впровадження моніторингової платформи, який включає всі етапи від проектування віртуального стенду до налаштування спеціалізованих перевірок та механізмів сповіщення, забезпечив створення повнофункціональної системи для раннього виявлення та запобігання інцидентам у мережевій інфраструктурі.

У результаті виконання роботи досягнуто наступних ключових результатів:

1. Розроблено структуровану методологію проектування тестового середовища, яка включає обґрунтований вибір віртуалізаційної платформи, проектування схеми IP-адресації та налаштування різнотипових операційних систем. Створений віртуальний стенд забезпечує адекватну імітацію корпоративної мережевої інфраструктури для всебічного тестування системи моніторингу.

2. Впроваджено модульний підхід до організації конфігураційних файлів Nagios Core, який забезпечує масштабованість системи, мінімізацію ризиків при внесенні змін та оптимізацію процесів адміністрування. Застосування структурованої ієрархії каталогів для розподілу конфігурацій за типами систем спрощує управління великими інфраструктурами.

3. Продемонстровано гнучкість системи моніторингу через інтеграцію різноманітних агентів та методів контролю, адаптованих для різних операційних систем та типів пристроїв. Використання NSClient++, NRPE та NCPA для моніторингу Windows та Linux систем забезпечує оптимальний баланс між функціональністю та ефективністю.

4. Реалізовано спеціалізований моніторинг критичних сервісів, включаючи поштовий сервер Postfix, який поєднує перевірку доступності протоколів, контроль внутрішнього стану черг та аналіз журналів подій.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		103

Багаторівневий підхід до моніторингу забезпечує всебічний контроль функціональності та продуктивності сервісів.

5. Впроваджено інтелектуальний підхід до моніторингу мережевої топології через визначення залежностей між компонентами інфраструктури, що дозволяє аналізувати кореневі причини збоїв та мінімізувати інформаційний шум при виникненні масштабних проблем.

6. Налаштовано ефективну систему сповіщень, яка забезпечує оперативне інформування відповідальних фахівців про інциденти з використанням електронної пошти як основного каналу комунікації. Оптимізовані шаблони повідомлень підвищують інформативність сповіщень та спрощують діагностику проблем.

Практична значимість виконаної роботи полягає у створенні комплексної методології впровадження системи моніторингу, яка може бути адаптована для різноманітних мережевих інфраструктур. Документований процес розгортання та конфігурації компонентів забезпечує можливість репродукування результатів та впровадження аналогічних систем у реальних виробничих середовищах. Модульний підхід до організації конфігурації забезпечує масштабованість рішення та можливість адаптації до специфічних вимог конкретних організацій.

Перспективи подальшого розвитку системи моніторингу включають:

1. Інтеграцію з системами візуалізації даних, такими як Grafana, для створення інформативних панелей моніторингу та аналізу трендів продуктивності інфраструктури.

2. Впровадження автоматизації конфігурації через інструменти управління конфігурацією, такі як Ansible, для спрощення процесу налаштування та оновлення компонентів системи.

3. Розширення системи моніторингу для контролю хмарних ресурсів та контейнеризованих додатків, що особливо актуально в умовах сучасних гібридних інфраструктур.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		104

4. Підвищення безпеки системи моніторингу через впровадження шифрування комунікацій, багатофакторної автентифікації та механізмів аудиту доступу.

5. Інтеграцію з системами управління інцидентами для автоматизації процесу обробки інцидентів та забезпечення відстеження їх життєвого циклу.

6. Впровадження механізмів предиктивного аналізу на основі історичних даних моніторингу для виявлення потенційних проблем до їх ескалації в критичні інциденти.

Впроваджена система моніторингу на базі Nagios Core демонструє ефективність відкритих рішень для забезпечення контролю над сучасними мережевими інфраструктурами. Гнучкість конфігурації, широкий спектр підтримуваних платформ та можливості інтеграції з додатковими інструментами створюють потужну основу для побудови комплексних систем моніторингу, адаптованих до конкретних потреб організацій різного масштабу та галузевої спрямованості.

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Россінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		105

Список використаних джерел

1. Releases · NagiosEnterprises/nagioscore - GitHub, <https://github.com/NagiosEnterprises/nagioscore/releases>
2. Nagios Core 4.x Version History, <https://www.nagios.org/projects/nagios-core/4x/>
3. Downloads - <https://nagios-plugins.org/downloads/>
4. Index of /downloads/nagiosxi/agents/NSClient++, <https://assets.nagios.com/downloads/nagiosxi/agents/NSClient++/>
5. NSClient - <https://exchange.nagios.org/directory/Addons/Monitoring-Agents/NSClient++/details>
6. Nagios Cross-Platform Agent (NCPA) | Download and Monitor Systems with Ease, <https://www.nagios.org/ncpa/>
7. Releases · NagiosEnterprises/ncpa - GitHub, 17, 2025, <https://github.com/NagiosEnterprises/ncpa/releases>
8. Nagios Stuff - Holger Weiß, <https://www.jhweiss.de/software/nagios.html>
9. Nagios check_smtp Plugin, https://nagios-plugins.org/doc/man/check_smtp.html
10. Manual Pages - Nagios Plugins, <https://nagios-plugins.org/doc/man/>
11. Checking Postfix queue size with Nagios - monitoring - Server Fault <https://serverfault.com/questions/531972/checking-postfix-queue-size-with-nagios>
12. SNMP Monitoring 2025, <https://www.nagios.com/solutions/snmp-monitoring/>
13. SNMP Interface - 2025, <https://exchange.nagios.org/directory/Plugins/Network-Protocols/SNMP/SNMP-Interface/details>

Додатки

Nagios®

General

Home
Documentation

Current Status

Tactical Overview
Map
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages

Quick Search:

Reports

Availability
Trends
Alerts
History
Summary
Histogram
Notifications
Event Log

System

Comments
Downtime
Process Info
Performance Info

Current Network Status

Last Updated: Fri Oct 17 18:51:18 UTC 2014
Updated every 90 seconds
Nagios® Core™ 4.0.8 - www.nagios.org
Logged in as nagiosadmin

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up Down Unreachable Pending
11 0 0 0

All Problems All Types
0 11

Service Status Totals

Ok Warning Unknown Critical Pending
33 1 1 4 0

All Problems All Types
6 39

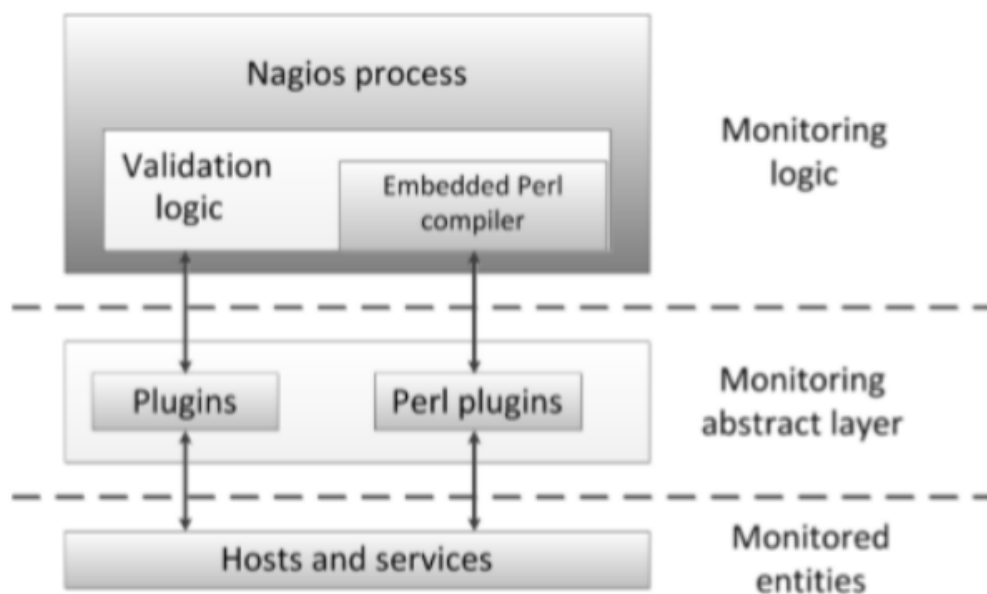
Service Status Details For All Hosts

Limit Results: 100

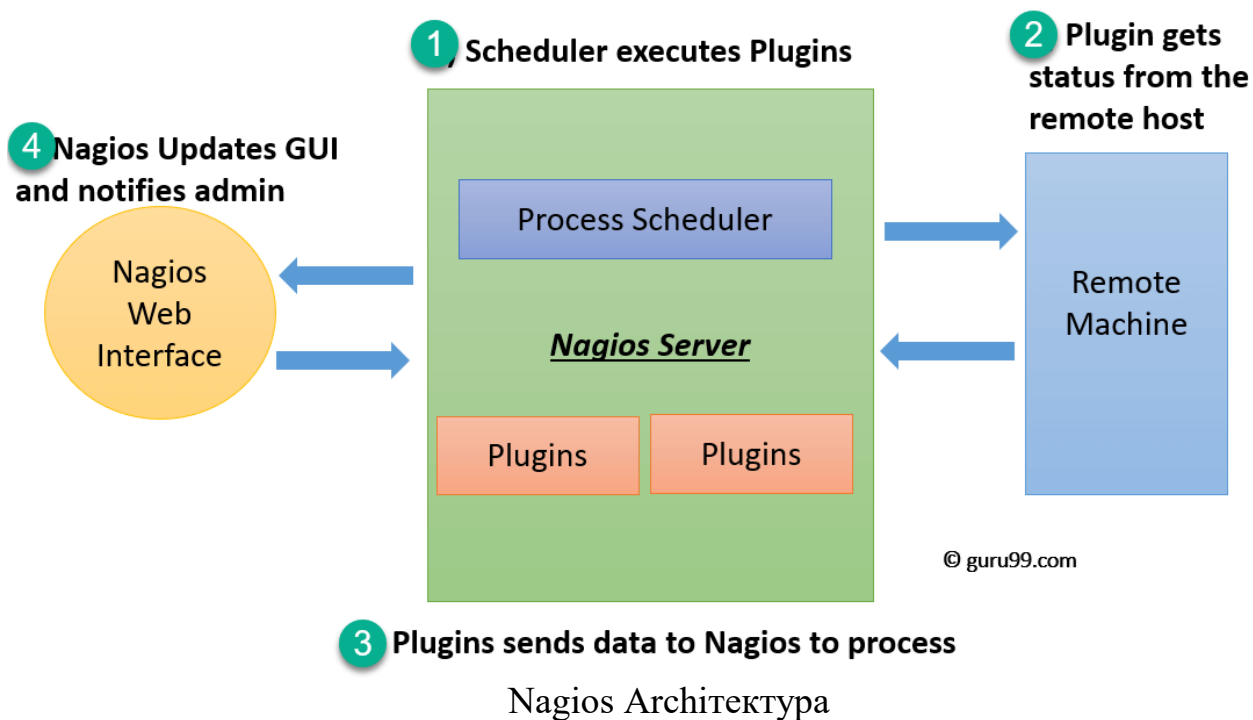
Host	Service	Status	Last Check	Duration	Attempt	Status Information
NOAA	Auroral Activity	OK	10-17-2014 18:51:09	535d 4h 28m 6s	1/3	Aurora OK: Activity level is 0.
	Weather Carteret North Carolina	WARNING	10-17-2014 18:43:15	0d 0h 46m 57s	3/3	Weather Warning: Beach High
	Weather King Washington	OK	10-17-2014 18:45:25	737d 1h 52m 46s	1/3	Weather OK: No watches in area.
	Weather Ramsey Minnesota	OK	10-17-2014 18:46:45	59d 20h 47m 12s	1/3	Weather OK: No watches in area.
	Weather San Bernardino California	OK	10-17-2014 18:41:45	0d 0h 48m 40s	1/3	Weather OK: No watches in area.
	Weather Strafford New Hampshire	OK	10-17-2014 18:43:45	0d 0h 46m 51s	1/3	Weather OK: No watches in area.
	Weather Tulsa Oklahoma	OK	10-17-2014 18:45:53	737d 1h 53m 51s	1/3	Weather OK: No watches in area.
	Current Load	OK	10-17-2014 18:49:08	0d 0h 46m 9s	1/4	OK - load average: 0.29, 0.29, 0.29
localhost	Current Users	OK	10-17-2014 18:51:02	1710d 15h 36m 24s	1/4	USERS OK - 0 users currently logged in
	HTTP	OK	10-17-2014 18:48:25	1019d 2h 7m 58s	1/4	HTTP OK: HTTP/1.1 200 OK response time
	PING	OK	10-17-2014 18:50:20	1710d 15h 35m 9s	1/4	PING OK - Packet loss = 0%
	Root Partition	OK	10-17-2014 18:48:32	938d 2h 32m 35s	1/4	DISK OK - free space: 7.2GB
	SSM	OK	10-17-2014 18:48:38	1705d 7h 35m 15s	1/4	SSM OK - QueueSize: 0

Огляд системи моніторингу Nagios Core

		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росінський Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		107



Логіка моніторинга



		Хоменюк В.Ю.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Росіінський Ю.М.				109
Змн.	Арк.	№ докум.	Підпис	Дата		