ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система автоматизованого збору даних про конфігурації серверного обладнання»

на здобуття освітнього ступеня магістра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

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

_____ Михайло БУРАКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти Михайло БУРАКОВ
група КНДМ-63

Керівник: Юрій КАТКОВ
науковий ступінь, вчене звання д.т.н., професор

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

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Буракову Михайлу Михайловичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система автоматизованого збору даних про конфігурації серверного обладнання

керівник кваліфікаційної роботи Юрій КАТКОВ д.т.н., професор,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. №320

2. Строк подання кваліфікаційної роботи «15» грудня 2024р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з питань, пов'язаних моніторингом серверного обладнання, технологія Nagios.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Аналіз сучасних інструментів для моніторінгу та управління в системі автоматизованого збору даних про конфігурації серверного обладнання.
4.2 Програмна реалізація системи моніторінгу конфігурації серверного обладнання на базі інструменту NAGIOS

4.3 Реалізація системи моніторінгу конфігурації серверного обладнання на базі інструменту NAGIOS

5. Перелік графічного матеріалу: *презентація*
   1) Тема дипломної роботи.
   2) Мета роботи. Об'єкт дослідження. Предмет дослідження.
   3) Завдання дипломної роботи.
   4) Порівняння інструментів моніторингу.
   5) Етапи методики тестування.
   6) Інтеграція Windows-системи в моніторингове середовище на базі Nagios.
   7) Тестування системи та аналіз результатів.
   8) Сконфігурований сервер.
   9) Висновки.

6. Дата видачі завдання«16» вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури.	16.09-27.09.24	Виконано
2	Аналіз предметної області та проектування web-сервісу моніторингу конфігурації серверного обладнання.	30.09-11.10.24	Виконано
3	Порівняльний аналіз технологій моніторингу конфігурацій серверного обладнання.	14.10-25.10.24	Виконано
4	Програмна реалізація системи моніторингу конфігурації серверного обладнання на базі системи Nagios.	28.10-08.11.24	Виконано
5	Розробка основних розділів.	11.11-22.12.24	Виконано
6	Розробка обов'язкових матеріалів.	25.11-29.11.24	Виконано
7	Оформлення роботи: вступ, висновки, реферат.	02.12-06.12.24	Виконано
8	Розробка демонстраційних матеріалів.	09.12-13.12.24	Виконано

Здобувач вищої освіти _____ Михайло БУРАКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник

кваліфікаційної роботи _____ Юрій КАТКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 147 стор., 3 табл., 71 рис., 46 джерел.

Наукове завдання – обґрунтування доцільності застосування інструменту Nagios для моніторингу конфігурації серверного обладнання.

Мета роботи – підвищити ефективність застосування системи автоматизованого збору даних моніторингу та управління про конфігурацію серверного обладнання на базі інструменту Nagios.

Об'єкт дослідження – процес використання Nagios для моніторингу та управління конфігурацією серверного обладнання.

Предмет дослідження – інструмент автоматизованого збору даних Nagios для моніторингу та управління про конфігурацію серверного обладнання.

Метод дослідження – заснований на відомих методах системного підходу, а саме: аналітично-розрахунковий, теоретичного аналізу з використанням моделювання, реконструювання і типологізації на основі аналізу емпіричних даних.

Короткий зміст роботи: Проведено аналіз сучасних інструментів для моніторінгу та управління в системі автоматизованого збору даних на базі інструменту Nagios про конфігурації серверного обладнання. Виконане дослідження програмної реалізації системи моніторінгу конфігурації серверного обладнання на базі інструменту NAGIOS. Запропанована реалізація системи моніторінгу конфігурації серверного обладнання на базі інструменту NAGIOS.

КЛЮЧОВІ СЛОВА: інструмент NAGIOS, адміністрування сервірів, моніторінг

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 147 pages, 3 tables, 71 figures, 46 sources.

Scientific task – is to justify the feasibility of using the Nagios tool for monitoring the configuration of server equipment.

Goal of the work – is to increase the efficiency of using an automated data collection and management system for monitoring and managing the configuration of server equipment based on the Nagios tool.

Object of research – is the process of using Nagios during monitoring and managing the configuration of server equipment.

Subject of research – is the automated data collection tool Nagios for monitoring and managing the configuration of server equipment.

Research methods – is based on well-known methods of the systems approach, namely: analytical-calculation, theoretical analysis using modeling, reconstruction, and typologization based on the analysis of empirical data.

Summary of the work: An analysis of modern monitoring and management tools in an automated data collection system based on the Nagios tool for server equipment configurations was conducted. A study of the software implementation of the server equipment configuration monitoring system based on the NAGIOS tool was conducted. The implementation of the server equipment configuration monitoring system based on the NAGIOS tool was proposed.

Keywords: NAGIOS tool, server administration, monitoring

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTTP	-	HyperText Transfer Protocol	Протокол передавання гіпертексту
ELK	-	Elasticsearch, Logstash, Kibana	Набір інструментів для роботи з логами та пошуком
YAML	-	YAML Ain't Markup Language	Мова серіалізації даних для конфігурацій
SNMP	-	Simple Network Management Protocol	Простий протокол керування мережею
PHP	-	PHP: Hypertext Preprocessor	Мова програмування для веб-розробки
NRDP	-	Nagios Remote Data Processor	Інструмент для передавання даних до Nagios
RRD	-	Round Robin Database	Кільцевий формат баз даних для збереження часових рядів
SSH	-	Secure Shell	Протокол для безпечного доступу до віддалених систем
NRPE	-	Nagios Remote Plugin Executor	Інструмент для виконання плагінів на віддалених хостах у Nagios
DNS	-	Domain Name System	Система доменних імен
CPU	-	Central Processing Unit	Центральний процесор
RAM	-	Random Access Memory	Оперативна пам'ять

ВСТУП

У сучасному світі IT-інфраструктури є важливою складовою для стабільного функціонування бізнесу, урядових установ та організацій різного масштабу. Складність IT-систем та серверного обладнання постійно зростає, що вимагає від інженерів більшого контролю за станом технічних ресурсів, їх налаштуванням та оновленням. Один із ключових аспектів ефективного управління IT-інфраструктурою – це наявність актуальних і достовірних даних про конфігурацію серверного обладнання.

Актуальність дослідження. З розвитком цифрових технологій та стрімким зростанням обсягу інформаційних ресурсів, управління IT-інфраструктурою стає критично важливим елементом стабільної роботи організацій. У багатьох компаніях IT-системи складаються з великої кількості серверів різних конфігурацій, що створює необхідність ефективного контролю за їхнім станом, налаштуванням та продуктивністю. Ручний збір та оновлення інформації про конфігурації серверного обладнання є трудомістким процесом, який вимагає багато часу і людських ресурсів та може призвести до помилок і неактуальних даних. Це, у свою чергу, впливає на швидкість реагування на збої, правильність інвентаризації та ефективність прийняття управлінських рішень.

Автоматизація процесів збору інформації про серверне обладнання допомагає значно спростити адміністрування IT-інфраструктури, підвищити її безпеку та знизити витрати на обслуговування. Вона дає можливість в реальному часі отримувати актуальні дані про конфігурацію, стан і продуктивність серверів, що є важливим для забезпечення безперервної роботи критично важливих сервісів та програмного забезпечення.

Таким чином, дослідження системи автоматизованого збору даних моніторингу та управління конфігурацією серверного обладнання є актуальним завданням, оскільки дозволяє розв'язувати проблеми ефективного управління ресурсами IT-інфраструктури, забезпечуючи її стабільну та безперебійну роботу.

Мета роботи – підвищити ефективність застосування системи автоматизованого збору даних моніторингу та управління про конфігурацію серверного обладнання на базі інструменту Nagios.

Об'єкт дослідження – процес використання Nagios для моніторингу та управління конфігурацією серверного обладнання.

Предмет дослідження – інструмент автоматизованого збору даних Nagios для моніторингу та управління про конфігурацію серверного обладнання.

Наукове завдання – обґрунтування доцільності застосування інструменту Nagios для моніторингу конфігурації серверного обладнання.

Метод дослідження – заснований на відомих методах системного підходу, а саме: аналітично-розрахунковий, теоретичного аналізу з використанням моделювання, реконструювання і типологізації на основі аналізу емпіричних даних.

Завдання роботи:

1. Провести аналіз сучасних інструментів для моніторингу та управління в системі автоматизованого збору даних про конфігурації серверного обладнання.

2. Провести обґрунтування доцільності застосування інструменту Nagios для моніторингу конфігурації серверного обладнання.

3. Реалізація системи моніторингу конфігурацій серверного обладнання на базі інструменту Nagios

Апробація результатів надається в 1 статті, 2 тезисах в рецензованих наукових журналах і виданнях.

Стаття: Бураков М. М., Критичні аспекти у системах автоматизованого збору даних про конфігурації серверного обладнання// Бураков М. М., Катков Ю.І.// Наукові записки Державного університету телекомунікацій №1, 2025, Подано до друку.

https://journals.dut.edu.ua/index.php/sciencenotes/issue/archive

Тезах:

Бураков М. М., Катков Ю. І. Проблеми безпеки систем автоматизованого збору даних про конфігурації серверного обладнання // науково-практична

конференція «Актуальні проблеми безпеки інформаційно-телекомунікаційних систем» Збірник тез. – К.: ДУІКТ, 2024. 03 листопада 2024, С.180-182. https://duikt.edu.ua/uploads/p_2661_93669247.pdf

Бураков М. М., Катков Ю. І. Особливості систем автоматизованого збору даних про конфігурації серверного обладнання Hewlett Packard Enterprise // науково-практична конференція «Сучасні досягнення компанії Hewlett Packard Enterprise в галузі іт та нові можливості їх вивчення і застосування» - подане до збірнику тез. – К.: ДУІКТ, 2024. 14 грудня 2024, С. https://duikt.edu.ua/ua/2661-konferencii-2024-roku-konferencii-ta-seminari

1 АНАЛІЗ СУЧАСНИХ ІНСТРУМЕНТІВ ДЛЯ МОНІТОРИНГУ ТА УПРАВЛІННЯ В СИСТЕМІ АВТОМАТИЗОВАНОГО ЗБОРУ ДАНИХ ПРО КОНФІГУРАЦІЇ СЕРВЕРНОГО ОБЛАДНАННЯ

1.1 Цілі та завдання сучасних інструментів моніторингу та управління для спостереження за конфігурацією серверного обладнання

Сучасні інструменти моніторингу та управління серверним обладнанням спрямовані на забезпечення високого рівня надійності, доступності та ефективності IT-інфраструктури. Оскільки сервери є основою більшості корпоративних мереж і зберігають важливі для бізнесу дані та додатки, ефективне управління їхньою конфігурацією має ключове значення для запобігання простою, втраті даних та зниженню продуктивності.

Основні цілі та завдання інструментів моніторингу [1, 3, 5]:

1. Системи моніторингу дозволяють безперервно контролювати статус серверів і швидко виявляти проблеми, такі як перевантаження процесорів, недостатність пам’яті чи ресурсів зберігання даних, завдання полягає в підтриманні постійної доступності та стабільності серверів. Ціль полягає у своєчасному реагуванні на відхилення, що може запобігти виникненню критичних збоїв у роботі систем.

2. Оптимізація продуктивності серверного обладнання – завдання полягає в аналізі даних про завантаження серверів та їхню продуктивність для виявлення можливих точок покращення. Це може включати балансування навантаження між різними серверами, перерозподіл ресурсів та оптимізацію конфігураційних параметрів. Сучасні інструменти можуть автоматично масштабувати ресурси відповідно до потреб для забезпечення стабільної роботи додатків.

3. Інструменти моніторингу часто мають функції автоматизації для спрощення завдань, таких як оновлення конфігурацій, управління користувачами,

встановлення патчів та безпекових оновлень. Це знижує ймовірність людських помилок, підвищує ефективність роботи адміністраторів та забезпечує безперервність обслуговування.

4. Забезпечення безпеки та дотримання стандартів – інструменти моніторингу відстежують зміни конфігурації, контролюють доступ до серверів і забезпечують відповідність нормативним вимогам, таким як ISO 27001 або PCI DSS. Це дозволяє мінімізувати ризики несанкціонованого доступу та забезпечити захист даних.

5. Використання механізмів прогнозування, які допомагають визначити тенденції використання ресурсів і прогнозувати потреби в них. Це дозволяє планувати розширення інфраструктури, уникаючи перевантаження обладнання та забезпечуючи безперебійне функціонування в періоди пікового навантаження [2].

Існує широкий спектр інструментів для моніторингу та управління конфігурацією серверного обладнання. Вони забезпечують автоматизацію збору даних, контроль стану серверів, а також управління їх конфігурацією. Нижче наведено популярні інструменти, розділені на категорії [1, 2, 3, 5]:

1. Інструменти моніторингу серверів. Ці інструменти забезпечують збір інформації про продуктивність, стан та доступність серверного обладнання, наприклад, Zabbix, Nagios, Prometheus, Solar Winds Server & Application Monitor (SAM), Checkmk та інші.

2. Інструменти управління конфігурацією. Ці інструменти автоматизують процес конфігурації та управління серверним обладнанням, наприклад, Ansible, Puppet, Chef, SaltStack, CFEngine.

3. Інтегровані системи моніторингу та управління. Поєднують моніторинг і управління конфігурацією в одному рішенні, наприклад, ManageEngine Applications Manager, BMC TrueSight Operations Management, Dynatrace.

4. Інструменти для візуалізації та аналітики. Забезпечують зручний інтерфейс для аналізу зібраних даних, наприклад, Grafana, Kibana.

5. Інструменти для аудиту та відстеження змін. Ці інструменти дозволяють відстежувати історію змін у конфігурації серверів, наприклад, Netwrix Auditor, Microsoft System Center Configuration Manager (SCCM).

6. Спеціалізовані інструменти для апаратного моніторингу. Розроблені виробниками серверів для збору даних про стан їхнього обладнання, например, Dell OpenManage, HP/HPE Integrated Lights-Out (iLO), Lenovo XClarity Administrator.

Кожен з цих інструментів має свої особливості, переваги та обмеження. Вибір залежить від вимог інфраструктури, бюджету та технічної підготовки команди. Для забезпечення реалізації зазначених загальних цілей і завдань, сучасні інструменти моніторингу та управління пропонують різноманітні можливості, адаптовані до специфічних потреб і особливостей серверного обладнання. Кожен із цих інструментів володіє унікальним функціоналом, що відповідає його призначенню, і вирішує окремі завдання, відстежуючи конфігурацію та стан інфраструктури.

Zabbix – це система з відкритим вихідним кодом для моніторингу і управління IT-інфраструктурою, яка дозволяє здійснювати контроль за серверним обладнанням, мережевими пристроями, додатками, базами даних і хмарними сервісами. Цей інструмент збирає, аналізує і відображає дані про роботу різних компонентів IT-середовища в реальному часі, що дозволяє вчасно виявляти та усувати потенційні проблеми. Zabbix складається з трьох базових компонентів[4]:

– сервера для координації виконання перевірок, формування перевірочних запитів та накопичення статистики;

– агентів для здійснення перевірок на стороні зовнішніх хостів;

– фронтенда для організації управління системою.

На рисунку 1.1 наведено приклад зручності інтерфейсу користувача Zabbix.

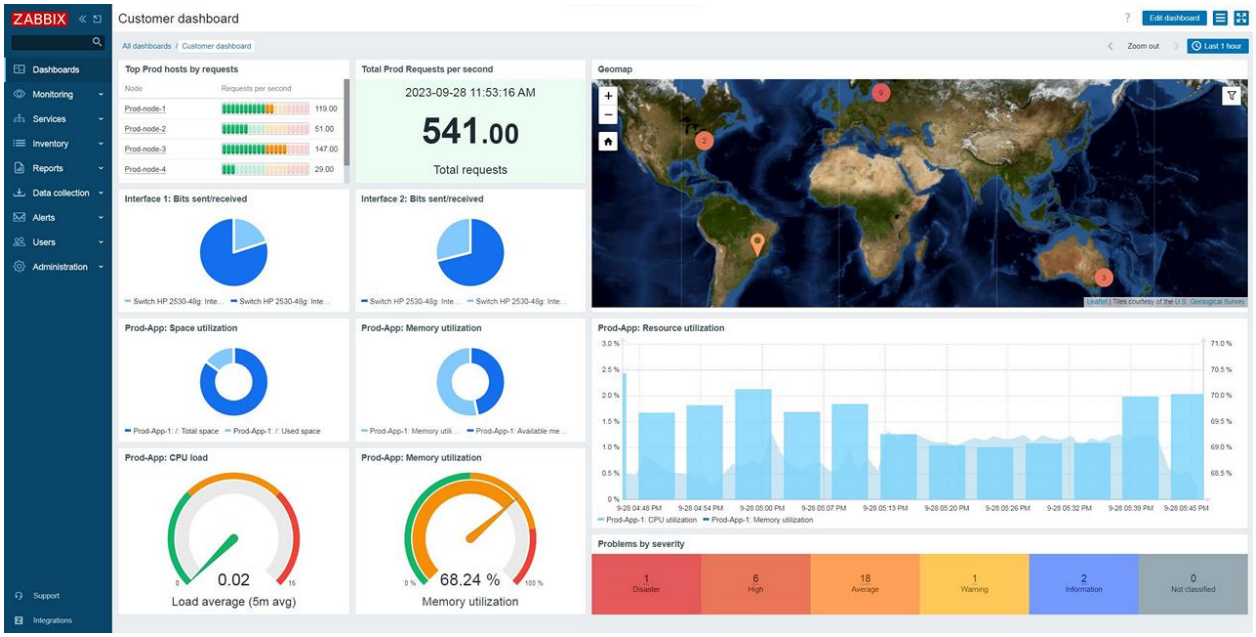


Рисунок 1.1 – Приклад зручності інтерфейсу користувача Zabbix [4]

Основні завдання Zabbix:

– своєчасне виявлення та попередження про збої або відхилення у роботі обладнання та додатків.

– оптимізація роботи IT-інфраструктури за рахунок постійного контролю за продуктивністю.

– підвищення ефективності адміністрування завдяки автоматизації моніторингу та оповіщень.

– створення звітності для аналізу стану інфраструктури та прийняття управлінських рішень.

Однією з основних функцій Zabbix є можливість візуалізації зібраних даних у вигляді графіків і дашбордів, що полегшує аналіз продуктивності IT-інфраструктури. Система також підтримує автоматичні оповіщення та налаштування тригерів для сповіщення адміністраторів у разі перевищення критичних значень або виникнення збоїв. Оповіщення можуть надсилатися через електронну пошту, SMS або інші канали зв'язку, що дозволяє швидко реагувати на будь-які інциденти.

Prometheus – це система для моніторингу та збору метрик, яка створена спеціально для роботи з сучасними розподіленими середовищами, такими як

хмарні інфраструктури, мікросервіси та контейнери [5]. На рисунку 1.2 наведено приклад зручності інтерфейсу користувача Prometheus.

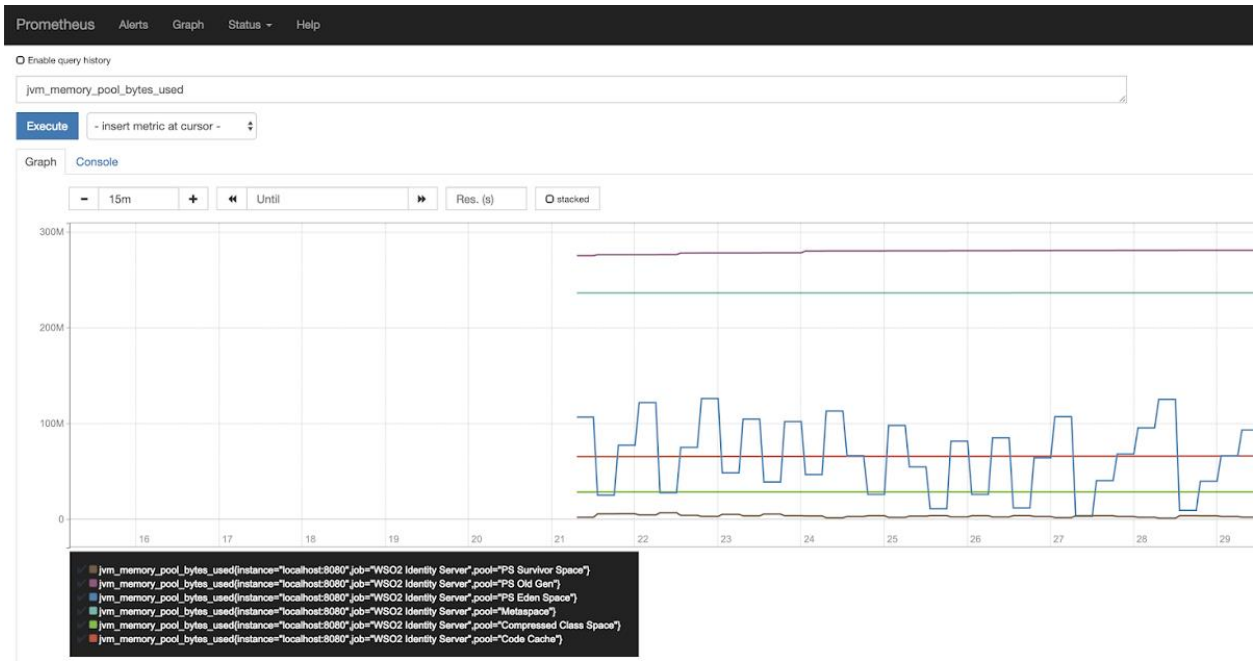


Рисунок 1.2 – Приклад зручності інтерфейсу користувача Prometheus [5]

Prometheus збирає метрики про продуктивність і стан системи через HTTP-запити, дозволяючи кожному компоненту інфраструктури виступати як окремий вузол, що відправляє свої дані. Дані збираються в часових рядах, що дає змогу відстежувати їх зміни з часом і здійснювати глибокий аналіз продуктивності. Інструмент використовує мову запитів PromQL, яка дозволяє будувати складні запити для аналізу зібраних метрик і створювати дашборди та графіки для візуалізації результатів.

Для виявляти проблеми в роботі сервісів та швидкого реагувати на можливі збої чи підвищене навантаження Prometheus підтримує механізм оповіщень, завдяки якому адміністратори можуть отримувати сповіщення про відхилення показників від норми.

Система чудово інтегрується з такими інструментами, як Grafana, яка використовується для побудови візуальних дашбордів. Також Prometheus підтримує експортери, спеціальні модулі, що збирають метрики з різних джерел (наприклад, операційних систем, баз даних, веб-сервісів).

Prometheus використовується для моніторингу метрик у динамічних інфраструктурах, де потрібно відстежувати стан великої кількості компонентів, що можуть змінюватися у реальному часі, як у випадку з контейнеризацією та оркестрацією (наприклад, Kubernetes). Інструмент зручний для комплексного аналізу продуктивності, планування ресурсів і управління стабільністю розподілених систем. На рисунку 1.3 показана концепція архітектури Prometheus.

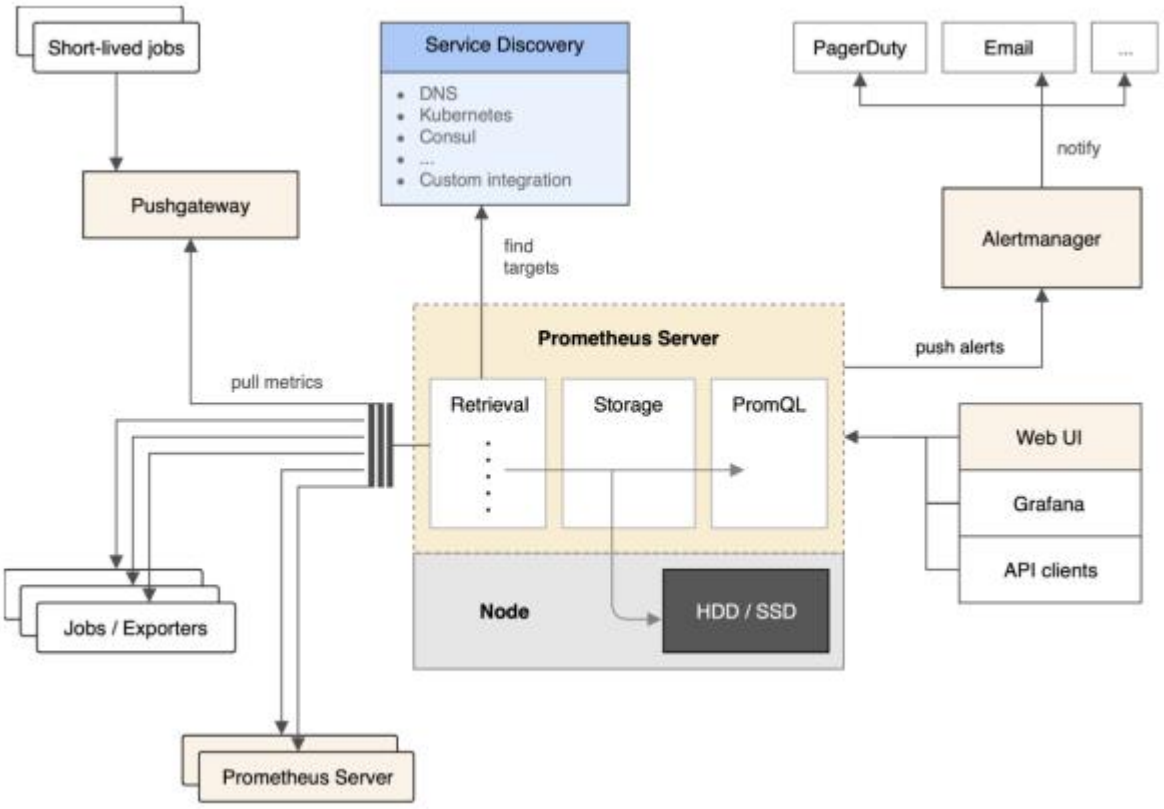


Рисунок 1.3 – Архітектура інструменту Prometheus [5]

Grafana – це інструмент для створення інтерактивних дашбордів та візуалізації даних, який використовується для моніторингу та аналітики IT-інфраструктури. Grafana підтримує інтеграцію з різними джерелами даних [6]. На рисунку 1.4 наведено приклад зручності інтерфейсу користувача.

Рисунок 1.4 – Приклад зручності інтерфейсу користувача Grafana [6]

Grafana надає широкі можливості для створення індивідуальних візуалізацій і дашбордів. Користувачі можуть налаштовувати графіки, таблиці, гістограми, теплові карти та інші візуальні елементи, що дає змогу гнучко відстежувати стан інфраструктури та робити аналітичні висновки. Grafana підтримує інтуїтивний інтерфейс для налаштування параметрів відображення даних, а також дозволяє створювати інтерактивні панелі, що оновлюються в реальному часі.

Однією з ключових особливостей Grafana є можливість створювати «алерти» або оповіщення, які надсилаються при перевищенні критичних показників. Це дозволяє вчасно реагувати на відхилення у роботі системи. Оповіщення можуть бути інтегровані з електронною поштою, Slack, PagerDuty, щоб оперативно повідомляти команди про проблеми.

Grafana використовується для централізованого контролю та аналізу даних, зібраних з різних компонентів IT-інфраструктури. Вона ідеально підходить для DevOps-команд, які працюють з динамічними і розподіленими середовищами. Вона надає інструмент для створення дашбордів і моніторингу складних систем. Завдяки своїй гнучкості, Grafana дозволяє компаніям швидко адаптувати

візуалізацію під специфічні потреби бізнесу та об’єднувати важливі метрики у зрозумілі інтерфейси, що сприяє ухваленню обґрунтованих рішень.

Grafana часто використовується в поєднанні з іншими системами моніторингу, такими як Prometheus, Zabbix або Elasticsearch, для забезпечення надійного та зручного моніторингу IT-інфраструктури.

ELK Stack — це набір програмних інструментів для збору та аналізу великих обсягів даних, особливо логів. На рисунку 1.5 наведено приклад зручності інтерфейсу користувача.

Рисунок 1.5 – Приклад зручності інтерфейсу користувача ELK Stack [7]

ELK — це абревіатура, що складається з трьох основних компонентів:

1. Elasticsearch – це розподілена пошукова та аналітична система, яка забезпечує швидкий і масштабований пошук та аналіз даних. Elasticsearch дозволяє зберігати дані у форматі JSON та здійснювати запити до них в реальному часі.

2. Logstash – це інструмент для збору, обробки та передачі даних. Він може інтегрувати дані з файлів журналів, баз даних, API тощо, обробляти їх (наприклад, фільтрувати або трансформувати) та відправляти в Elasticsearch для подальшого зберігання та аналізу.

3. *Kibana* – це інтерфейс візуалізації, що дозволяє користувачам створювати графічні матеріали, діаграми та дашборди на основі даних, збережених в

Elasticsearch. Kibana надає зручні інструменти для аналізу даних та візуалізації їх у реальному часі [7] (рис. 1.6).

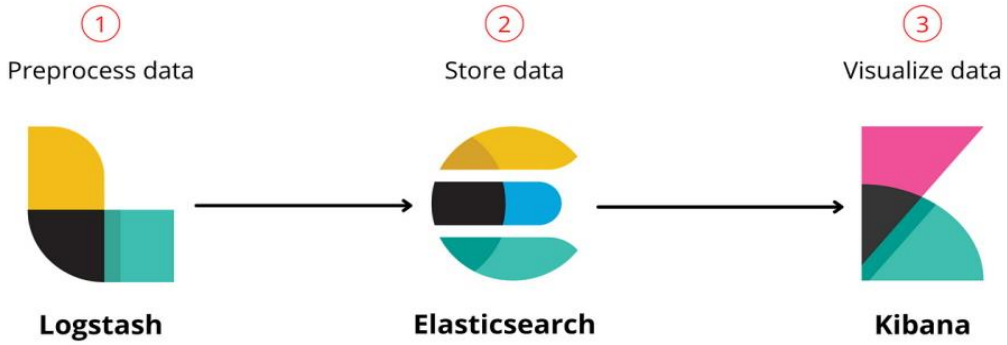


Рисунок 1.6 – Компоненти з яких складається стек ELK [7]

Разом ці компоненти дозволяють організаціям ефективно працювати з даними, отримувати аналітику в реальному часі та здійснювати моніторинг систем і додатків. ELK Stack часто використовується для аналізу логів, моніторингу продуктивності, виявлення аномалій та інших сценаріїв, пов'язаних з обробкою даних.

Ansible – це інструмент *infrastructure as a code* для автоматизації завдань з підготовки та конфігурування інфраструктури. Він дозволяє адміністраторам систем і DevOps-інженерам автоматизувати різноманітні завдання, такі як:

1. Управління конфігурацією – Ansible дозволяє визначати та підтримувати конфігурацію серверів і додатків, щоб забезпечити їх узгодженість і стандартність.

2. Розгортання додатків, завдяки Ansible можна швидко і безпечно розгортати нові версії програмного забезпечення на кількох серверах.

3. Оркестрація в Ansible дозволяє координувати виконання складних завдань, які можуть охоплювати кілька етапів або різні системи, гарантуючи, що всі компоненти працюють узгоджено та ефективно (рис. 1.7).

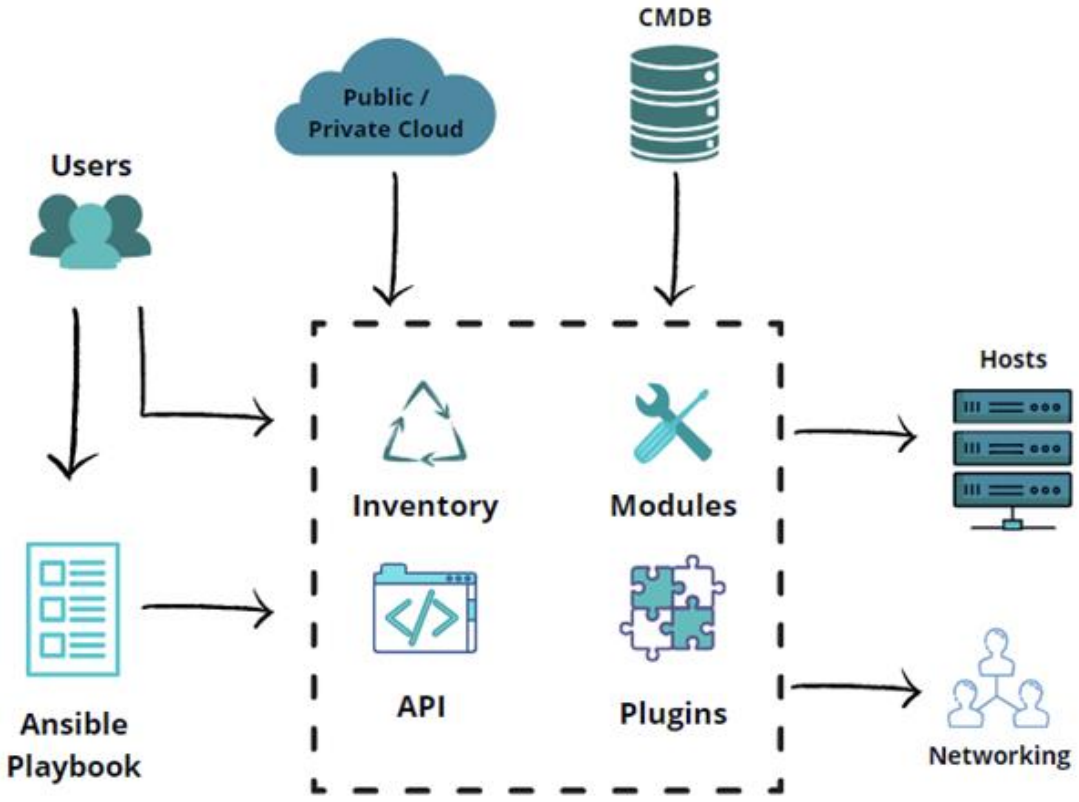


Рисунок 1.7 – Архітектура інструменту Ansible [8]

Основні особливості Ansible:

– Простота використання, ansible використовує просту синтаксис у форматі YAML (файли конфігурацій називаються плейбуками), що робить його легким для читання і написання.

– Безагентна архітектура, ansible не потребує установки агентів на управлювані вузли. Він використовує стандартні протоколи, такі як SSH для Linux або WinRM для Windows, для виконання завдань.

– Масштабованість, ansible здатен управляти тисячами серверів одночасно, що робить його придатним для великих інфраструктур.

– Модульність, ansible має велику бібліотеку модулів, які виконують різні функції (наприклад, управління пакетами, користувачами, файлами тощо). Користувачі також можуть створювати свої власні модулі.

Ansible часто використовується в поєднанні з іншими інструментами DevOps, такими як Docker, Kubernetes і Jenkins, для створення ефективних та автоматизованих робочих процесів [8].

SNMP (Simple Network Management Protocol, протокол простого управління мережею) – це мережевий протокол, який використовується для управління та моніторингу мережевих пристроїв, таких як маршрутизатори, комутатори, сервери, принтери та інші пристрої, що підтримують підключення до мережі. Цей протокол забезпечує адміністраторів можливістю отримувати інформацію про стан пристроїв, діагностувати проблеми та керувати мережевими ресурсами в реальному часі. На рисунку 1.8 наведена архітектура SNMP, а саме:

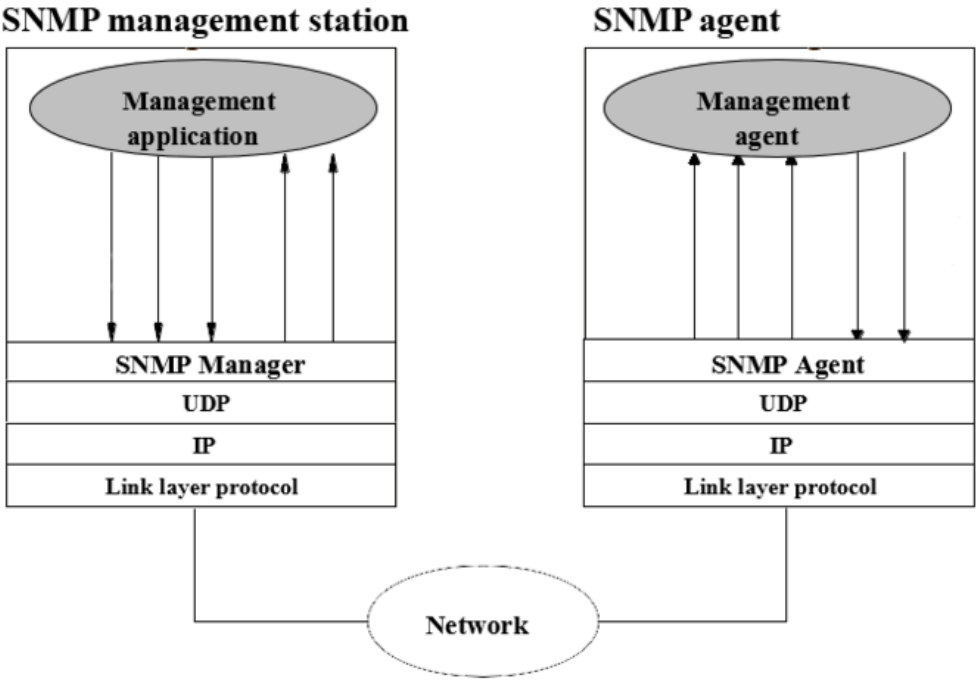


Рисунок 1.8 – Архітектура інструменту SNMP [9]

SNMP Management Station, або станція управління SNMP, є центральною ланкою в системі моніторингу та управління мережею. Це програмно-апаратний комплекс, який виконує функції збору інформації, аналізу, моніторингу та управління мережею. Станція управління використовує SNMP-протокол для запитів до мережевих пристроїв, отримуючи від них дані про стан та продуктивність. Основні компоненти станції включають Management Application – прикладну програму для обробки даних і візуалізації мережевих показників, та SNMP Manager – модуль, який безпосередньо взаємодіє з агентами. Менеджер посилає команди на пристрої, наприклад, для зміни налаштувань або отримання даних, і отримує відповіді на запити. SNMP Management Station є критично

важливим для адміністраторів, оскільки дозволяє централізовано керувати та контролювати стан всіх мережевих пристроїв.

SNMP Agent – це програмне забезпечення, що встановлюється на кожному керованому пристрої в мережі, наприклад, на маршрутизаторах, комутаторах, принтерах чи серверах. Агент відповідає за збір інформації про стан пристрою та його ресурси, зокрема, про навантаження на процесор, використання пам'яті, стан інтерфейсів тощо. Ці дані зберігаються в базі MIB (Management Information Base), структурованій у вигляді дерева об'єктів з унікальними ідентифікаторами. Агент також може надсилати Trap-повідомлення станції управління про критичні події, наприклад, якщо пристрій вийшов з ладу або виявлено збій. Таким чином, SNMP Agent є інструментом для збору, зберігання та передачі даних у відповідь на запити від станції управління або при виникненні важливих подій, забезпечуючи адміністраторам можливість своєчасного реагування на неполадки.

Завдяки поділу на SNMP Management Station і SNMP Agent, мережеві адміністратори отримують централізований контроль над всіма пристроями, можливість швидкого виявлення та усунення збоїв, а також аналізу продуктивності мережі. Протокол SNMP спрощує процеси діагностики та підтримки мережі, дозволяючи своєчасно реагувати на критичні події і запобігати можливим проблемам [9].

Nagios – система моніторингу мережевих ресурсів, серверів, додатків і всієї IT-інфраструктури. Вона забезпечує можливість виявлення і попередження про збої, а також контролю продуктивності, що дозволяє адміністраторам мережі своєчасно реагувати на проблеми і підтримувати стабільну роботу сервісів [10].

Веб-інтерфейс Nagios (рис. 1.9) забезпечує зручний моніторинг, налаштування сповіщень, доступ до звітів і керування правами користувачів для ефективного контролю інфраструктури.

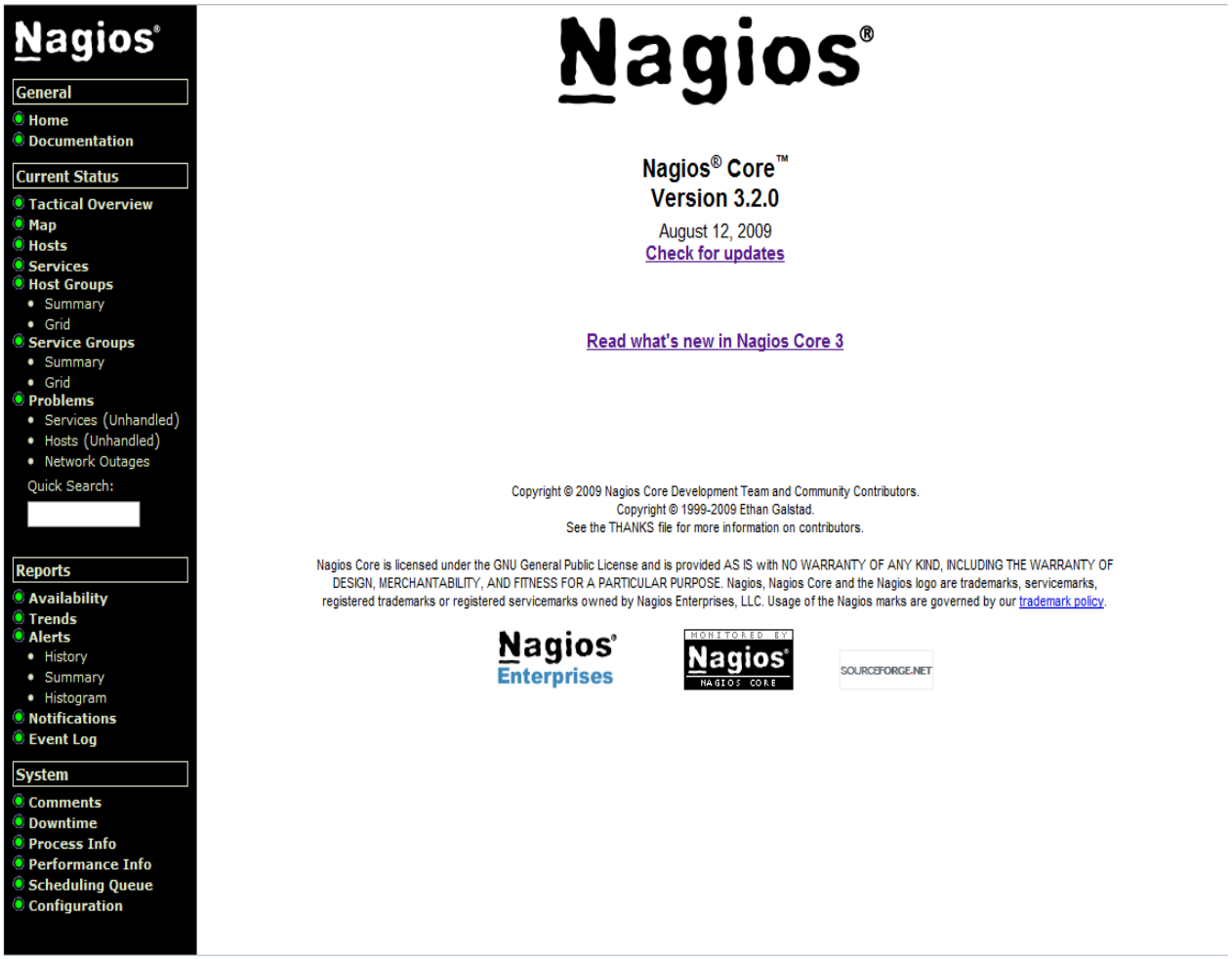


Рисунок 1.9 – Веб-інтерфейс системи моніторингу Nagios [10]

Це рішення є надзвичайно гнучким і потужним, оскільки ядро системи моніторингу можна доповнювати плагінами для розширення його функціональних можливостей.

На рисунку 1.10 зображено архітектуру моніторингу в Nagios, яка включає три ключові компоненти: об'єкти моніторингу, центральний сервер Nagios і методи відображення статусу системи.

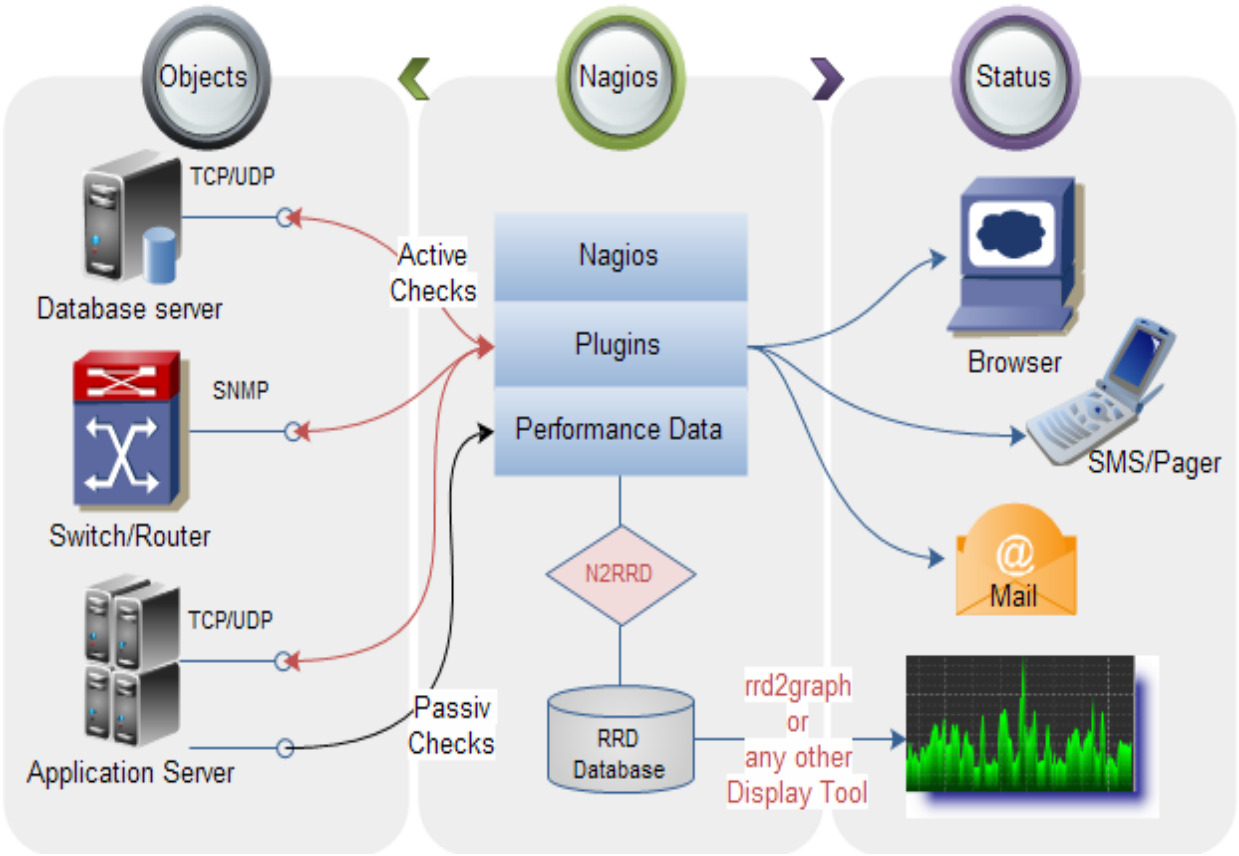


Рисунок 1.10 – Архітектура моніторингу Nagios [11]

Перша частина діаграми представляє об'єкти, які моніторяться за допомогою Nagios, такі як сервери баз даних, мережеві пристрої (маршрутизатори, комутатори) та прикладні сервери. Взаємодія з цими об'єктами може відбуватися двома способами. По-перше, є активні перевірки, які ініціює сам Nagios, здійснюючи запити на сервери та пристрої для збору інформації про їхній поточний стан. По-друге, є пасивні перевірки, коли об'єкти самостійно надсилають дані в Nagios, зазвичай у випадках, коли виникають критичні ситуації або коли необхідно зберігати ресурси.

Центральна частина діаграми відображає основний сервер Nagios, який обробляє всю інформацію, отриману від об'єктів моніторингу. Сервер використовує плагіни, які виконують конкретні завдання з моніторингу різних ресурсів, збираючи дані про стан та продуктивність систем. Зібрані дані зберігаються як "Performance Data" (дані продуктивності), які можна аналізувати для оцінки продуктивності і здоров'я системи. Для передачі пасивних даних

використовується протокол NRDP, який дозволяє ефективно обмінюватися інформацією між Nagios і віддаленими об'єктами. Історичні дані зберігаються у базі даних RRD, що дозволяє Nagios зберігати інформацію про продуктивність та створювати графіки для довгострокового аналізу.

Третя частина діаграми демонструє відображення статусу та результатів моніторингу. Статус усіх об'єктів можна переглядати через веб-інтерфейс у браузері, що забезпечує зручний і наочний доступ до інформації про стан мережі та серверів. Окрім цього, система може надсилати сповіщення адміністраторам через SMS, пейджер або електронну пошту, що дає змогу оперативно реагувати на виниклі проблеми. Дані продуктивності можуть бути візуалізовані у вигляді графіків за допомогою таких інструментів, як mrtg2graph або інших засобів для відображення, що дозволяє проводити детальний аналіз тенденцій у використанні ресурсів.

1.2 Порівняння інструментів моніторингу та управління конфігурацією серверного обладнання

Для порівняння інструментів моніторингу слід дослідити рівень відповідності кожній з вимог для різних інструментів дослідження. Для цього зібрано дані з документації різних інструментів, а також з їхніх спеціальних форумів і репозиторіїв, щоб знайти альтернативне та функціональне рішення. Результати порівняння наведені в таблиці 1.1

Таблиця 1.1 - Порівняння інструментів моніторингу

Критерій	Zabbix	Prometheus	Grafana	ELK Stack	Ansible	SNMP	Nagios
Архітектура	Централізована	Розподілена, оптимізована для контейнерів	Візуалізація даних	Розподілена, з компонентами для збору даних	Декларативна	Розподілена	Централізована
Масштабованість	Добра, але з обмеженнями на великі середовища	Висока, адаптована для великих кластерів	Залежить від джерела даних	Висока, але вимоги до ресурсів зростають	Добра, але залежить від складності	Обмежена, в основному для простих мереж	Добра, але може бути складно для налаштування
Налаштування та конфігурація	Складне, багато параметрів	Середнє, прості конфігураційні файли	Легке, з інтуїтивним інтерфейсом	Складне, вимагає знань з Elastic search	Середнє, проста синтаксична структура	Легке, але вимагає налаштування на кожному пристрої	Складне, багато налаштувань
Типи метрик	Часові ряди, статуси, події	Часові ряди, метрики системи	Візуалізація будь-яких даних	Логи, метрики, події	Конфігурації, але не специфічно для метрик	Статуси, продуктивність	Часові ряди, статуси, продуктивність
Алертинг	Потужна система алертингу	Гнучка система алертингу	Підтримує алерти через Prometheus	Відсутня в Kibana, потрібно налаштовувати окремо	Підтримує різні способи сповіщення	Просте сповіщення	Гнучка система сповіщень
Інтеграції	Різноманітні плагіни та API	Підтримує багато сторонніх інтеграцій	Висока, з підтримкою численних джерел	Висока, інтеграція з багатьма джерелами	Широкі можливості інтеграцій	Обмежені, в основному для мережевих пристроїв	Велика кількість плагінів

Продовження таблиці 1.1 - Порівняння інструментів моніторингу

Критерій	Zabbix	Prometheus	Grafana	ELK Stack	Ansible	SNMP	Nagios
Документація та спільнота	Гарна документація, активна спільнота	Висока якість документації, активна спільнота	Гарна документація, активна спільнота	Висока якість документації, активна спільнота	Хороша документація, активна спільнота	Обмежена, але доступна	Гарна документація, активна спільнота
Безпека	Підтримує аутентифікацію та шифрування	Підтримує шифрування та аутентифікацію	Залежить від джерела даних	Підтри-мує безпеку через TLS/SSL	Відсутні вбудовані механі-зми безпеки	Основні механізми безпеки	Основні механізми безпеки
Вартість	Безкоштовний, з можливістю комерційної підтримки	Безкоштовний, з можливістю комерційної підтримки	Безкоштовний, з можливістю комерційної підтримки	Безкоштовний, з можливістю комерційної підтрим-ки	Безкоштовний, відкритий вихідний код	Безкоштовний, відкритий вихідний код	Безкоштовний, відкритий вихідний код
Кроспла-тформність	Підтримує Linux, Windows	Підтримує Linux, Windows	Підтримує Linux, Windows, MacOS	Підтри-мує Linux, Windows	Підтри-мує різні ОС	Підтримує всі основні ОС	Підтримує Linux, Windows
Гнучкість та кастомізація	Висока, через плагіни та API	Висока, через експорт метрик	Висока, через створення власних панелей	Висока, з можливістю налаштування	Дуже висока, можна налаштувати під будь-які потреби	Обмежена, стандартні параметри	Висока, можна налаштувати під потреби
Інтерфейс користувача	Інтуїтивно зрозумілий, але застарілий	Сучасний, простий у використанні	Інтуїтивно зрозумілий, сучасний	Інтуїтив-но зрозумі-лий, але може бути складним	Командний рядок, немає GUI	Простий, але застарілий	Простий, але застарілий
Системні вимоги	Середні, залежить від кількості даних	Низькі, але зростають з кількістю даних	Низькі, залежить від джерела даних	Високі, потребує багато ресурсів	Низькі, залежить від виконуваних задач	Низькі, залежить від пристроїв	Середні, залежить від кількості даних

Функціональні вимоги:

Архітектура є критично важливим аспектом при виборі системи моніторингу або управління конфігурацією, оскільки вона визначає, як інструмент взаємодіє з іншими компонентами системи. Централізовані рішення, такі як Zabbix і Nagios, можуть бути простішими для налаштування в невеликих або середніх середовищах, тоді як розподілені архітектури, такі як Prometheus, забезпечують більшу гнучкість та масштабованість в умовах, де обробляється велика кількість даних, наприклад, у контейнеризованих середовищах. Вибір архітектури впливає на продуктивність, швидкість збору даних і загальну ефективність моніторингу [12].

Масштабованість, дозволяє системі адаптуватися до змінюваних потреб бізнесу. З ростом обсягу даних і кількості об'єктів для моніторингу, здатність системи обробляти ці дані стає критично важливою. Nagios демонструє вражаючу масштабованість завдяки своїй архітектурі, яка дозволяє легко додавати нові плагіни та розширення. На відміну від деяких конкурентів, Nagios може адаптуватися до змін у розмірах і вимогах інфраструктури без значних зусиль. Це робить його надійним вибором для компаній, які планують зростати та розширювати свої системи.

Налаштування та конфігурація системи безпосередньо впливають на її доступність і ефективність використання. Інструменти з простими і зрозумілими процесами налаштування, такі як Grafana, дозволяють швидко інтегруватися з різними джерелами даних, що є важливим фактором для швидкого реагування на зміну умов роботи. У той же час, складні конфігурації в Zabbix або Nagios можуть забезпечити глибший контроль і можливість налаштування під специфічні потреби організації. Це важливо для оптимізації моніторингу та забезпечення точності даних.

Вибір типів метрик, які можуть бути зібрані, визначає, наскільки всебічно інструмент може моніторити системи. Інструменти, такі як Prometheus і Zabbix, можуть обробляти як часові ряди, так і події, що дозволяє отримати детальну інформацію про продуктивність системи. Nagios підтримує широкий спектр

метрик, включаючи часові ряди та статуси, що дозволяє отримати всебічну картину продуктивності системи. Його гнучкість у зборі метрик забезпечує можливість моніторингу різних компонентів інфраструктури, чого не завжди можна досягти з іншими системами. Це робить Nagios відмінним вибором для підприємств, які потребують детальної аналітики.

Алертинг є критичним компонентом системи моніторингу, оскільки він дозволяє виявляти проблеми в реальному часі. Nagios пропонує потужну систему сповіщень, що дозволяє налаштувати різні методи сповіщення в залежності від типу проблеми. Це забезпечує ефективне управління інцидентами, чого може не вистачати іншим системам, які можуть пропонувати менш гнучкі або прості рішення. Завдяки своїй здатності швидко реагувати на проблеми, Nagios залишається одним з кращих виборів для управління інфраструктурою [13].

Інтеграції з іншими системами та сервісами можуть значно підвищити ефективність роботи інструментів моніторингу. Наприклад, можливість інтеграції з CI/CD, системами управління конфігурацією та іншими платформами дозволяє створити комплексне середовище для автоматизації та моніторингу. Інструменти з широкими можливостями інтеграції, такі як ELK Stack і Ansible, забезпечують високу продуктивність завдяки спільному використанню даних і функцій.

Документація та спільнота є важливими факторами, що впливають на впровадження та використання інструментів. Якісна документація полегшує навчання користувачів і швидке вирішення проблем. Активні спільноти, такі як у Prometheus та Zabbix, забезпечують додаткову підтримку та обмін досвідом, що може бути надзвичайно корисним у процесі впровадження та використання.

Безпека системи моніторингу є критично важливою, оскільки вона зберігає конфіденційну інформацію про інфраструктуру. Nagios пропонує вбудовані механізми безпеки, які забезпечують надійний захист даних. Це особливо важливо для підприємств, які повинні дотримуватися стандартів безпеки. У порівнянні з іншими системами, Nagios забезпечує гнучкість в налаштуванні рівнів доступу, що дозволяє контролювати, хто може отримувати доступ до певних даних [14].

Вартість інструмента може суттєво впливати на рішення організації. Безкоштовні або відкриті рішення, такі як Zabbix, Prometheus, Grafana та Ansible, можуть бути привабливими для малих і середніх підприємств. Nagios дозволяє організаціям отримати потужні можливості моніторингу без великих фінансових витрат. Проте важливо враховувати не лише початкову вартість, але й витрати на підтримку, налаштування та масштабування системи в майбутньому.

Кросплатформність є важливим аспектом, оскільки сучасні інфраструктури часто складаються з різноманітних систем і операційних середовищ. Інструменти, які підтримують роботу на різних ОС, такі як Linux, Windows і MacOS, можуть легше інтегруватися в існуючі системи та забезпечити безперебійну роботу.

Гнучкість та кастомізація дозволяють організаціям адаптувати інструменти під свої специфічні вимоги. Інструменти з високою гнучкістю, такі як Ansible, пропонують можливості кастомізації, які можуть суттєво покращити ефективність управління інфраструктурою. Це важливо для адаптації до змін у бізнес-потребах та технологічному середовищі.

Інтерфейс користувача впливає на зручність роботи з інструментами моніторингу. Інтуїтивно зрозумілий інтерфейс, як у Grafana, може значно полегшити навчання нових користувачів і пришвидшити процес впровадження. Nagios має класичний інтерфейс, він забезпечує простоту навігації та доступ до всіх необхідних функцій. В порівнянні з більш сучасними інтерфейсами інших інструментів, Nagios може здаватися застарілим, проте його функціональність і надійність компенсують це.

Системні вимоги також важливі для вибору інструменту. Nagios має помірні вимоги до ресурсів, що робить його доступним для впровадження в більшості існуючих інфраструктур. У порівнянні з системами, які потребують великих апаратних ресурсів, Nagios дозволяє організаціям ефективно використовувати наявні ресурси.

Після порівняння систем моніторингу та управління конфігурацією, можна з упевненістю стверджувати, що Nagios є одним з найкращих варіантів для

організацій, які шукають надійне і гнучке рішення. Його централізована архітектура, що спрощує налаштування та управління, робить його ідеальним вибором для підприємств різного розміру. Також Nagios демонструє масштабованість, дозволяючи легко адаптувати систему до зростаючих потреб бізнесу. Його гнучка система алертингу забезпечує своєчасне сповіщення про потенційні проблеми, що є критично важливим для підтримки безперервної роботи. Завдяки широким можливостям інтеграції з іншими системами, Nagios легко вписується в існуючі технологічні стеки, що надає йому додаткові переваги. Крім того, активна спільнота користувачів і доступна документація роблять інструмент привабливим вибором для тих, хто потребує підтримки та ресурсів під час впровадження та налаштування системи. Його баланс між простотою використання і потужними можливостями кастомізації дозволяє організаціям адаптувати систему під свої специфічні вимоги. У порівнянні з іншими інструментами, Nagios виграє завдяки своїй надійності, простоті в налаштуванні та високій гнучкості. Ці характеристики роблять його відмінним вибором для організацій, які прагнуть створити ефективну систему моніторингу та управління інфраструктурою, що відповідає їхнім потребам у змінюваному бізнес-середовищі [15].

1.3 Визначення загальних вимог та етапів реалізації системи автоматизованого збору даних про конфігурацію серверного обладнання

Загальні вимоги.

1. Стабільність і надійність: Система повинна працювати цілодобово та бути здатною виявляти і реєструвати збої або проблеми в конфігурації.

2. Масштабованість: Система повинна підтримувати додавання нових серверів і пристроїв без значних змін в основній конфігурації.

3. Інтуїтивний інтерфейс: Наявність веб-інтерфейсу для управління та перегляду інформації, що забезпечується Nagios.

4. Безпека: Система повинна підтримувати багаторівневу аутентифікацію користувачів і шифрування даних.

Етапи реалізації системи.

Етап 1. Планування і підготовка.

1.1 Оцінка інфраструктури: Провести аналіз поточного стану серверної інфраструктури, визначити кількість серверів, специфікацію обладнання та типи сервісів, що потребують моніторингу.

1.2 Вибір плагінів: Визначити, які стандартні плагіни Nagios будуть використані для моніторингу основних параметрів (ЦП, пам'ять, дисковий простір), та розробити перелік необхідних кастомних плагінів для моніторингу специфічних аспектів конфігурації.

1.3 Розробка конфігураційного плану: Створити план конфігурації, який включає структуру хостів і служб, правила для сповіщень та обмеження доступу.

Етап 2. Встановлення та базове налаштування Nagios.

2.1 Встановлення Nagios: Встановити Nagios Core на виділений сервер відповідно до вимог до системи.

2.2 Налаштування основних компонентів: Налаштувати основні конфігураційні файли Nagios (`nagios.cfg`, `hosts.cfg`, `services.cfg`) для моніторингу обраних серверів.

2.3 Додавання хостів і служб: Додати конфігурації для кожного сервера, що потребує моніторингу, вказуючи IP-адреси, порти та перелік сервісів.

2.4 Тестування базових налаштувань: Перевірити коректність роботи системи після первинного налаштування. Забезпечити працездатність моніторингу основних параметрів (пінг, ЦП, пам'ять, диски) [16].

Етап 3. Налаштування плагінів для розширеного моніторингу.

3.1 Вибір та інтеграція стандартних плагінів: Підключити плагіни для моніторингу основних аспектів, таких як навантаження на ЦП, використання пам'яті, дискового простору, температуру серверів тощо.

3.2 Розробка кастомних плагінів: Створити додаткові плагіни для моніторингу специфічних аспектів конфігурації, наприклад, статус RAID-масивів, моніторинг оновлень ПЗ, параметрів безпеки (наприклад, файрволи).

3.3 Тестування та налаштування плагінів: Провести тестування роботи плагінів для перевірки коректного зчитування параметрів та відображення інформації в Nagios.

Етап 4. Налаштування політик сповіщень.

4.1 Конфігурація правил сповіщень: Налаштувати правила, що визначають, коли і кому повинні надходити сповіщення про збої або відхилення від конфігураційних норм.

4.2 Налаштування рівнів критичності: Визначити порогові значення для сповіщень (критичні, попереджувальні) для кожного параметра конфігурації.

4.3 Сповіщення через різні канали: Включити сповіщення через електронну пошту, SMS або інші канали в разі виявлення проблеми. Перевірити правильність роботи сповіщень.

Етап 5. Забезпечення безпеки та розмежування доступу.

5.1 Налаштування рівнів доступу: Визначити ролі користувачів (адміністратор, оператор, користувач) і налаштувати відповідний доступ до системи.

5.2 Налаштування аутентифікації: Включити базову аутентифікацію через веб-сервер (Apache) або LDAP для контролю доступу до інтерфейсу Nagios.

5.3 Шифрування даних: Встановити SSL-сертифікати для забезпечення безпеки з’єднання між сервером Nagios і користувачами, а також між Nagios і моніторинговими серверами.

Етап 6. Налаштування періодичного контролю конфігурації та візуалізації даних.

6.1 Періодичний моніторинг: Налаштувати регулярний моніторинг конфігурації серверів для перевірки їхньої актуальності.

6.2 Візуалізація даних: Інтегрувати додаткові інструменти в Nagios (наприклад, NagiosGraph або PNP4Nagios) для створення графіків і звітів, що відображають динаміку змін конфігурацій.

6.3 Створення періодичних звітів: Налаштувати автоматичне формування звітів для подальшого аналізу змін у конфігурації та стану серверного обладнання.

Етап 7. Тестування та введення системи в експлуатацію.

7.1 Підготовка до тестування: Розробити план тестування системи, включаючи функціональні, навантажувальні та негативні тести.

7.2 Проведення тестування: Перевірити працездатність системи моніторингу на різних сценаріях, включаючи можливість моніторингу в умовах підвищеного навантаження.

7.3 Аналіз результатів тестування: Оцінити результати тестування, внести необхідні корективи в налаштування та забезпечити стабільну роботу системи.

7.4 Введення в експлуатацію: Після успішного тестування, система вводиться в експлуатацію для постійного моніторингу конфігурацій серверів.

Етап 8. Підтримка та документування системи.

8.1 Генерація звітів і журналів: Налаштувати автоматичне формування звітів і ведення журналів для аналізу історії змін у конфігураціях.

8.2 Підтримка та оновлення: Забезпечити регулярні оновлення Nagios та плагінів, а також періодичне налаштування параметрів моніторингу згідно з новими вимогами.

8.3 Навчання користувачів: Надати користувачам інструкції для роботи з системою, щоб забезпечити їхнє розуміння основних функцій і правил безпеки.

Таким чином відповідно цілей та завдань сучасних інструментів моніторингу та управління для спостереження за конфігурацією серверного обладнання, а також на основі порівняння інструментів моніторингу та управління конфігураціями серверного обладнання визначено загальні вимоги та етапи реалізації системи автоматизованого збору даних про конфігурацію серверного обладнання. Це допоможе провести дослідження програмної реалізації моніторингу серверного обладнання на базі інструменту Nagios.

2 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ЗАСТОСУВАННЯ ІНСТРУМЕНТУ NAGIOS ДЛЯ МОНІТОРИНГУ КОНФІГУРАЦІЇ СЕРВЕРНОГО ОБЛАДНАННЯ

2.1 Визначення виду методики для дослідження програмної реалізації могіторінгу серверного обладнання на базі інструменту Nagios

2.1.1 Види методик для дослідження програмної реалізації моніторингу серверного обладнання на базі інструменту Nagios. Дослідження програмної реалізації системи моніторингу конфігурації серверного обладнання на базі інструменту Nagios може бути проведено за допомогою різних методик. Вони залежать від мети дослідження (оптимізація, впровадження, аналіз продуктивності тощо) і обраного підходу. Ось основні методики:

1. Методика аналізу та проєктування системи моніторингу. Ця методика включає етапи, які допомагають зрозуміти вимоги та спроєктувати архітектуру системи моніторингу:

- Збір вимог: визначення компонентів системи (сервери, мережеві пристрої, сервіси); визначення метрик моніторингу (завантаженість CPU, пам'яті, дискового простору, доступність мережевих ресурсів тощо).
- Проєктування архітектури: побудова схеми взаємодії компонентів моніторингу; вибір між централізованою та розподіленою моделлю моніторингу.

2. Емпірична методика. Передбачає проведення практичного дослідження роботи системи:

- Розгортання та налаштування Nagios: інсталяція Nagios Core, плагінів і необхідного додаткового ПЗ, налаштування конфігураційних файлів для моніторингу обраних компонентів.

- Тестування функціональності: перевірка коректності отриманих даних за допомогою плагінів; симуляція аварійних ситуацій (наприклад, зупинка служби або втрата доступу до сервера) для оцінки роботи системи.
- Аналіз результатів: збір логів і створення звітів для аналізу.

3. Методика оцінки продуктивності. Ця методика спрямована на дослідження ефективності системи моніторингу:

- Вимірювання навантаження: моніторинг продуктивності сервера, на якому розгорнуто Nagios; визначення впливу Nagios на серверні ресурси.
- Аналіз затримок: оцінка часу реакції системи на зміни стану обладнання; аналіз ефективності роботи плагінів.
- Оптимізація: налаштування частоти перевірок; використання більш ефективних плагінів або налаштувань.

4. Методика порівняльного аналізу. Передбачає зіставлення Nagios із іншими інструментами моніторингу:

- Визначення критеріїв: функціональність, масштабованість, простота налаштування, продуктивність.
- Проведення тестів: використання тестових сценаріїв для оцінки Nagios і конкурентів (Zabbix, PRTG, Prometheus).
- Звітування: опис переваг і недоліків Nagios у порівнянні з альтернативами.

5. Методика інтеграції та кастомізації. Цей підхід корисний для адаптації Nagios до специфічних потреб:

- Інтеграція: з'єднання Nagios з іншими системами (наприклад, Grafana, Prometheus) для розширення функціоналу.
- Розробка кастомних плагінів: написання скриптів для специфічних метрик моніторингу.
- Автоматизація процесів: використання Ansible, Puppet або Chef для автоматичного розгортання та налаштування Nagios.

6. Методика моделювання.

- Створення тестового середовища: моделювання мережі й обладнання в ізольованому середовищі; тестування роботи Nagios на цій моделі.
- Симуляція навантаження: перевірка роботи системи за великого обсягу даних і багатьох хостів.

7. Документування та навчання.

- Створення інструкцій: документування процесів встановлення, налаштування, використання та масштабування Nagios.
- Навчання користувачів: підготовка адміністраторів до роботи з системою.

Ці методики можуть використовуватися окремо або в поєднанні для всебічного дослідження й адаптації системи моніторингу на базі Nagios. Пропонується створити комбіновану методику для дослідження програмної реалізації моніторінгу серверного обладнання на базі інструменту Nagios.

2.1.2 Характеристика інструменту Nagios. Nagios — це популярний інструмент для моніторингу IT-інфраструктури, який дозволяє забезпечити контроль за станом серверів, мережевих пристроїв, додатків та інших компонентів системи. Його гнучкість, масштабованість і розширюваність зробили його стандартом у сфері моніторингу [17]. Основні характеристики наступні Nagios:

1. Функціонал моніторингу

- Сервери та мережі: Моніторинг стану серверів, робочих станцій, мережевих пристроїв (маршрутизатори, комутатори) та протоколів (HTTP, FTP, SMTP, SNMP тощо).
- Системні ресурси: Спостереження за завантаженістю процесора, використанням пам'яті, дискового простору, станом процесів і сервісів.
- Додатки: Моніторинг роботи додатків, баз даних (MySQL, PostgreSQL), веб-серверів (Apache, Nginx) тощо.

2. Система сповіщень

- Сповіщення через електронну пошту, SMS або месенджери.

- Налаштування порогів, які визначають, коли відправляти сповіщення (наприклад, при досягненні певного завантаження процесора або збої в роботі служби).

3. Гнучкість і модульність

- Плагіни: Nagios підтримує тисячі сторонніх плагінів, які дозволяють розширювати його функціонал.
- Можливість розробки власних плагінів: Використовуючи будь-яку мову програмування, наприклад Python, Bash чи Perl.

4. Веб-інтерфейс

- Інтуїтивно зрозумілий інтерфейс для перегляду стану моніторингу, створення звітів і налаштування сповіщень.
- Графічне відображення інцидентів і проблем.

5. Масштабованість

- Підходить як для малих мереж, так і для великих розподілених систем.
- Підтримує моніторинг віддалених об'єктів через агентські або безагентські методи.

6. Система управління подіями

- Запускає заздалегідь визначені сценарії у разі виявлення проблем (наприклад, автоматичний рестарт служби).

Переваги Nagios:

1. Відкрите програмне забезпечення: Базова версія Nagios (Nagios Core) є безкоштовною та відкритою.
2. Гнучке налаштування: Можливість адаптації до будь-яких потреб моніторингу.
3. Велика спільнота: Наявність великої кількості навчальних матеріалів, плагінів і підтримки від спільноти.
4. Підтримка розширень: Наприклад, Nagios XI (платна версія з розширеним функціоналом та підтримкою).

Недоліки Nagios:

1. Складність налаштування: Початкове налаштування може бути складним для новачків.

2. Обмежений інтерфейс у базовій версії: Nagios Core має простий, але менш інтерактивний інтерфейс порівняно з сучасними інструментами.

3. Масштабованість у великих мережах: У великих інфраструктурах необхідно налаштовувати спеціальні методи для ефективного використання.

Типові сценарії використання:

- Моніторинг серверів у дата-центрі.
- Контроль за доступністю веб-сервісів та баз даних.
- Спостереження за мережевими пристроями в розподілених системах.
- Автоматизація реагування на збої.

Nagios ідеально підходить для компаній, які шукають універсальний інструмент моніторингу, здатний адаптуватися до індивідуальних потреб, з можливістю інтеграції з іншими рішеннями через плагіни чи API.

2.1.3 Завдання дослідження програмної реалізації системи моніторингу конфігурації серверного обладнання на базі інструменту Nagios. Обгрунтування доцільності застосування інструменту Nagios для моніторингу серверного обладнання — це структурований підхід до вивчення, розробки та впровадження системи моніторингу, яка дозволяє відстежувати стан серверів, мережевого обладнання, служб і додатків, а також дозволяють проаналізувати ефективність, функціональність і адаптивність цього інструменту для специфічних потреб моніторингу. Вона передбачає наступні завдання:

1. Аналіз, розробка та впровадження системи моніторингу на базі інструменту Nagios для збору, аналізу та управління даними про конфігурацію серверного обладнання.

2. Аналіз можливостей інструменту Nagios у контексті моніторингу серверної конфігурації: апаратні параметри, стан сервісів, продуктивність.

3. Аналіз масштабованості системи для великих серверних інфраструктур.

4. Налаштування та оптимізації для конкретних потреб організації.

5. Інтеграція із зовнішніми інструментами та розширення функціоналу шляхом використання плагінів чи створення власних модулів.

Методика дослідження програмної реалізації моніторингу серверного обладнання на базі інструменту Nagios передбачає такі основні етапи дослідження:

1. Аналіз вимог до моніторингу конфігурації передбачає:

- Визначення цілей і завдань моніторингу.
- Перелік серверів, мережевого обладнання та служб, які потрібно контролювати.
- Визначення метрик (наприклад, завантаженість CPU, RAM, доступність мережі, дисковий простір тощо).
- Визначення, які параметри серверного обладнання будуть відслідковуватися:
  - Апаратні характеристики (процесор, пам'ять, диски).
  - Налаштування мережі.
  - Встановлене програмне забезпечення та його версії.
  - Стан і конфігурація операційної системи.
- Оцінка частоти збору даних, допустимого навантаження на сервери, обсягу збережених даних.

2. Вивчення можливостей Nagios передбачає:

- Огляд функціоналу Nagios, включаючи підтримку плагінів, налаштування сповіщень, інтеграцію з іншими інструментами.
- Аналіз версій (Nagios Core, Nagios XI) і вибір відповідної версії.

3. Планування архітектури моніторингової системи передбачає:

- Вибір підходу до розгортання (локальний сервер, хмарне рішення, розподілена система).

- Створення схеми мережі з позначенням усіх елементів, що потребують моніторингу.
- Розробка топології моніторингу (централізована чи розподілена).

4. Розгортання Nagios передбачає:

- Інсталяція Nagios Core або Nagios XI на сервер.
- Встановлення Nagios Core у локальному або хмарному середовищі.
- Налаштування конфігураційних файлів (наприклад, **nagios.cfg**). та налаштування веб-інтерфейсу для відображення стану серверів і конфігурацій.
- Встановлення та налаштування плагінів для збору даних.
- Інтеграція з віддаленими серверами через агенти (NRPE, NSClient++) або безагентські підходи (SNMP, SSH).

3. Налаштування моніторингу конфігурації передбачає:

- Додавання хостів (серверів) і сервісів до конфігурації.
- Визначення параметрів перевірки (наприклад, інтервали між перевірками, тайм-аути).
- Налаштування сповіщень для адміністраторів (e-mail, SMS, інтеграція з месенджерами).
- Використання стандартних плагінів для збору даних про сервери.
- Моніторинг доступності серверів (ping).
- Збір інформації про використання ресурсів (процесор, пам'ять, дисковий простір).
- Розробка власних плагінів для відстеження специфічних конфігурацій (написання скриптів для перевірки версій програмного забезпечення; визначення відповідності конфігурації корпоративним стандартам).

4. Аналіз продуктивності (тестування системи) передбачає:

- Тестування навантаження на сервери від системи моніторингу.
- Аналіз затримок при зборі та оновленні даних.
- Перевірка коректності зібраних даних.

- Аналіз роботи системи за різних умов (навантаження, відмова серверів тощо).
- Виявлення та усунення помилок у конфігурації.
- Інструкції для адміністраторів щодо підтримки та розширення системи моніторингу.

5. Масштабування передбачає:

- Вивчення можливостей Nagios для роботи у великих розподілених мережах (налаштування Nagios Distributed Monitoring для моніторингу великих систем; інтеграція з іншими інструментами, такими як Grafana, для більш зручної візуалізації даних).

6. Інтеграція з іншими інструментами передбачає:

- Використання зовнішніх систем збору даних або автоматизації (Ansible, Puppet) для розширення функціональності.
- Експорт даних у зовнішні бази даних для аналізу.

7. Документування методики передбачає:

- Опис налаштувань, використаних плагінів і параметрів.

8. Оцінка, оптимізація та підтримка передбачає:

- Аналіз ефективності системи моніторингу.
- Оптимізація налаштувань для зниження використання ресурсів і підвищення точності моніторингу.
- Регулярне оновлення Nagios і плагінів.
- Додавання нових хостів і сервісів у міру їх появи

Завдяки наданій методиці можна отримати наступні очікувані результати:

1. Реалізація функціональної системи моніторингу конфігурації серверного обладнання.
2. Визначення сильних та слабких сторін Nagios у сфері моніторингу серверів.
3. Розробка рекомендацій для впровадження Nagios у різних типах інфраструктури.

4. Створення власних модулів та плагінів для розширення функціоналу.

Ця методика є особливо важливою для підприємств, що прагнуть мінімізувати простої серверного обладнання та гарантувати стабільність IT-інфраструктури. Крім того має науков значущість тому, що дослідження надасть практичні рекомендації щодо використання Nagios для моніторингу серверного обладнання, а також сприятиме розвитку систем автоматизації управління IT-інфраструктурою.

2.1.4 Особливості встановлення Nagios Core 4.3.4 і Nagios Plugin 2.2.1. Для встановлення Nagios Core 4.3.4 і Nagios Plugin 2.2.1 треба виконати наступні дії (програмний код див. Додаток А):

1. Попередня підготовка системи:

- Оновіть систему
- Встановіть необхідні пакети
- Додайте користувача Nagios

2. Завантаження вихідного коду:

- Завантажте Nagios Core 4.3.4
- Завантажте Nagios Plugins 2.2.1

3. Встановлення Nagios Core:

- Скомпілюйте та встановіть Nagios Core
- Встановіть Apache веб-сервер і налаштуйте доступ
- Запустіть Nagios

4. Встановлення Nagios Plugins:

- Перейдіть до каталогу з плагінами та скомпілюйте
- Перевірте плагіни

5. Перевірка системи:

- Перевірте конфігурацію Nagios
- Доступ до веб-інтерфейсу Nagios

6. Налаштування моніторингу:

- Додайте нові хости, служби та параметри моніторингу, редагуючи файли конфігурації в каталозі
- Перезапустіть Nagios після внесення змін

7. Тестування:

- Упевніться, що всі хости та сервіси відображаються на веб-інтерфейсі
- Використовуйте вбудовані плагіни, такі як `check_ping`, `check_http`, для перевірки стану систем

Примітка: Ця інструкція базується на стандартних умовах. Якщо виникають проблеми або використовуєте специфічну ОС/середовище, уточніть це для додаткової допомоги.

Таким чином:

– Nagios і його плагіни буде встановлено в каталозі /usr/local/nagios.

– Nagios на RHEL буде налаштований для моніторингу служб вашого локального комп'ютера (використання диска, завантаження процесора, поточні користувачі, загальна кількість процесів тощо).

– Веб-інтерфейс Nagios буде доступний за адресою http://localhost/nagios.

Порядок підготовки Nagios до роботи наданий у Додатку Б, він має наступні кроки:

1. Встановлення необхідних залежностей.
2. Створення користувача і групи Nagios на RHEL.
3. Завантаження Nagios Core 4.3.4 і Nagios Plugin 2.2.1.
4. Налаштування Nagios Core.
5. Налаштування конфігурації Nagios на RHEL.
6. Встановлення та налаштування веб-інтерфейсу для Nagios.
7. Компілювання та встановлення плагіну Nagios.
8. Перевірка файлів конфігурації Nagios.

Вхід у веб-інтерфейс Nagios. На рисунку 2.1 та 2.2 показана панель керування Nagios Core.

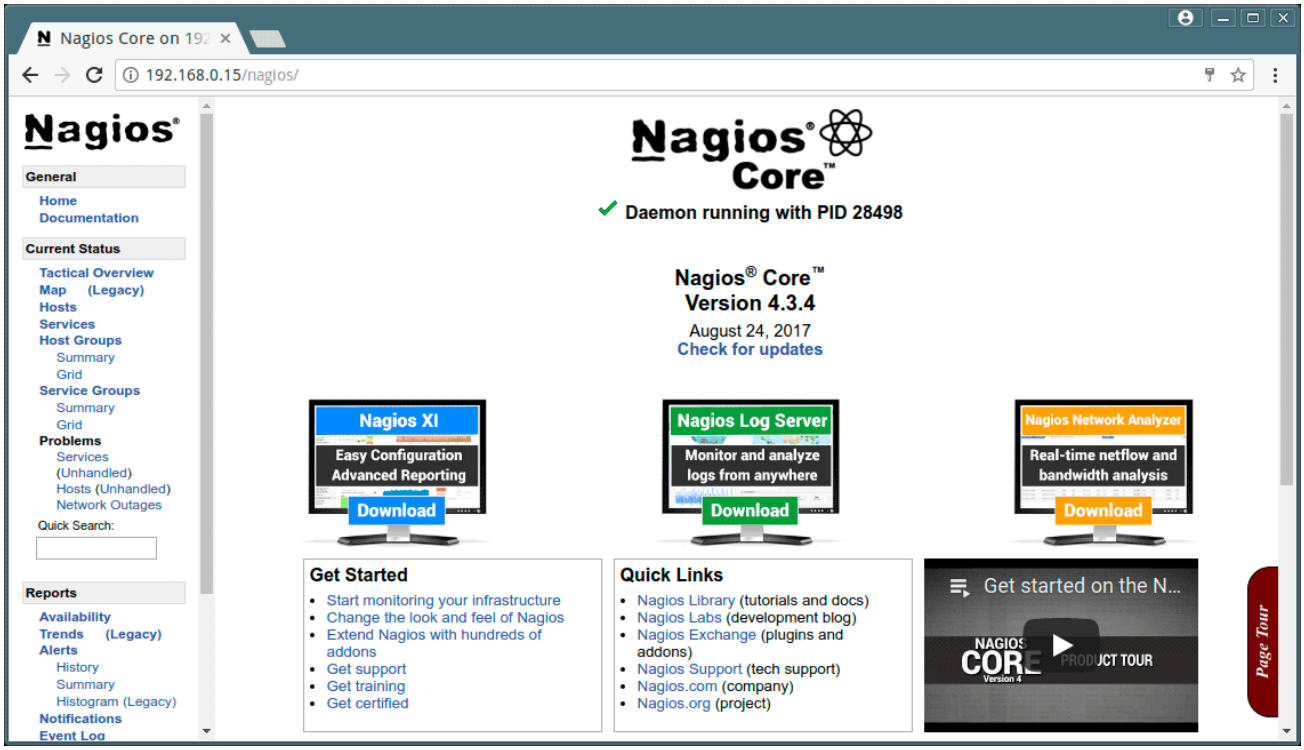


Рисунок 2.1 – Панель керування Nagios Core [18]

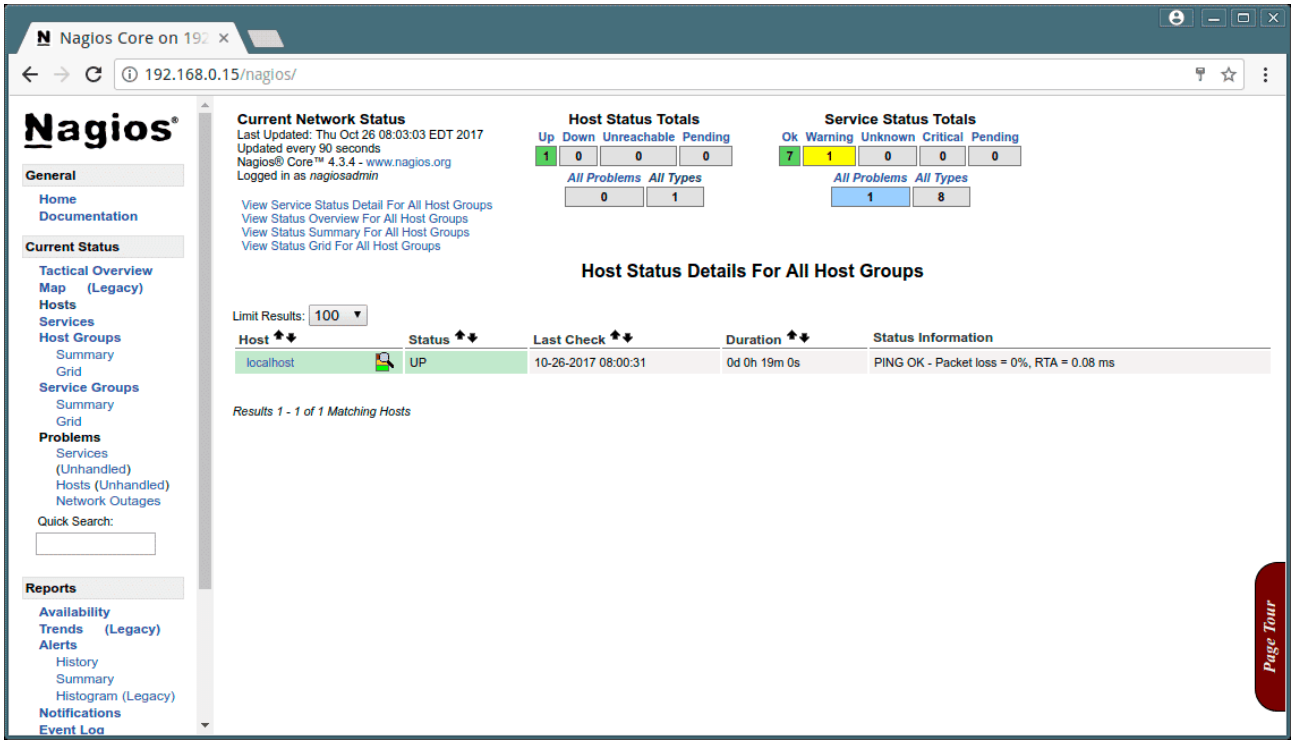


Рисунок 2.2 – Перегляд існуючих хостів [18]

На рисунку 2.3 продемонстровано огляд інтерфейсу Nagios.

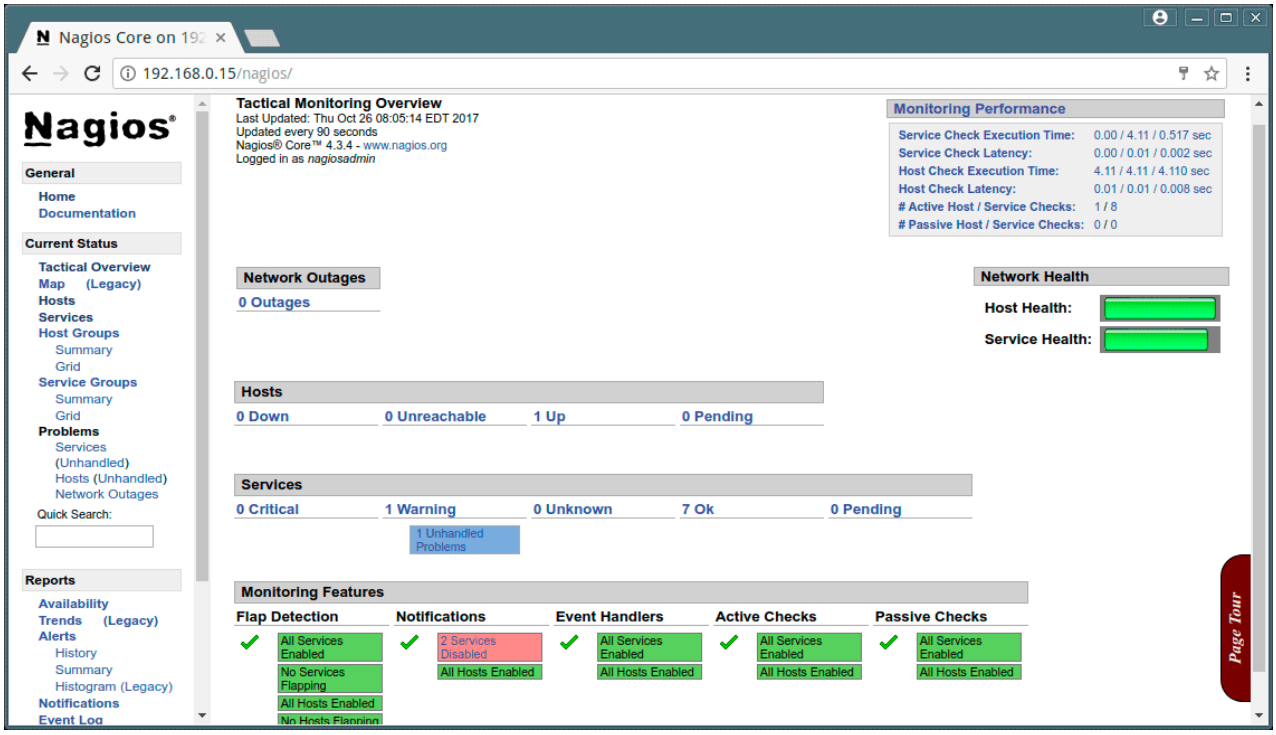


Рисунок 2.3 – Огляд інтерфейсу [18]

На рисунку 2.4 наведено список запущених служб.

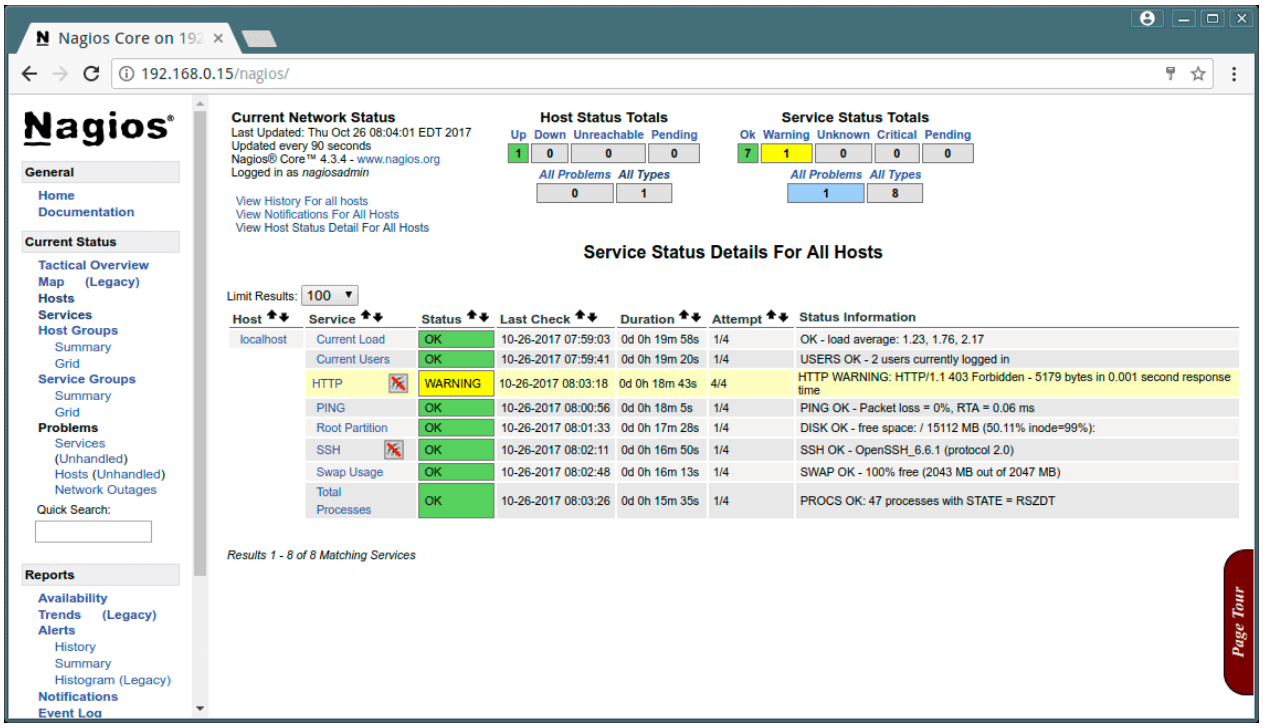


Рисунок 2.4 – Перегляд служб [18]

На рисунку 2.5 показано список запущених процесів.

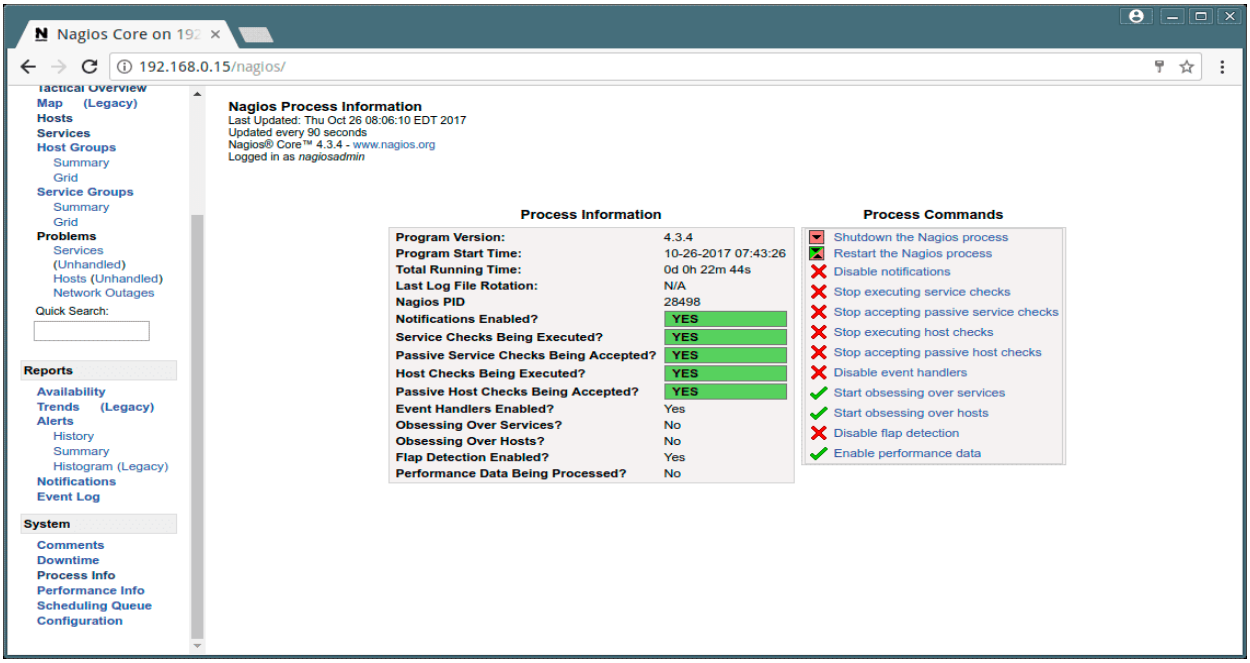


Рисунок 2.5 – Перегляд процесів [18]

2.2 Особливості налаштування плагінів для моніторингу

Використання існуючих плагінів Nagios Nagios пропонує широкий спектр готових плагінів, які покривають базові потреби моніторингу.

Найбільш популярні плагіни здатні відстежувати показники які показані на таблиці 2.1:

Таблиця 2.1 - Плагіни Nagios та їх параметри моніторингу

Назва плагіну	Опис	Параметри моніторингу
check_cpu	Моніторинг навантаження на центральний процесор	Завантаження ЦП, середнє використання ЦП
check_mem	Моніторинг оперативної пам'яті	Загальний обсяг пам'яті, вільна та зайнята пам'ять
check_disk	Моніторинг дискового простору	Наявність вільного місця на дисках
check_temp	Моніторинг температури	Температура основних компонентів
check_network	Моніторинг мережевої активності	Пропускна здатність, затримки, втрата пакетів
check_raid	Моніторинг RAID-масиву	Статус RAID, помилки дисків

Ці плагіни можуть бути налаштовані для виконання періодичних перевірок, а також для інформування адміністратора при виникненні критичних ситуацій.

Налаштування кастомних плагінів.

Кастомні плагіни необхідні для моніторингу специфічних параметрів обладнання, що не входять до базового набору Nagios. Їх створюють, коли готові плагіни не покривають потрібний функціонал або якщо серверне обладнання потребує нестандартного підходу. Плагіни можуть бути написані на мовах Python, Bash, Perl та інших скриптових мовах.

Основні етапи розробки кастомного плагіна:

1. Визначення параметрів моніторингу: обрати параметри, які необхідно контролювати, наприклад, стан певних служб, конфігурації або специфічні апаратні параметри.

2. Написання скрипту: скрипт повинен приймати параметри конфігурації, збирати потрібні дані та виводити їх у форматі, зрозумілому Nagios (зазвичай це коди стану: 0 – OK, 1 – Warning, 2 – Critical, 3 – Unknown).

3. Тестування та налагодження: запуск і тестування на сервері для перевірки точності роботи.

4. Інтеграція в Nagios: додавання плагіна в конфігурацію Nagios для автоматичного запуску, визначення періодичності перевірок і повідомлень [19].

Приклад створення кастомного плагіна на Bash для моніторингу конкретної служби показаний на рисунку 2.6.

```bash
#!/bin/bash
SERVICE_NAME="my_custom_service"

if systemctl is-active --quiet "$SERVICE_NAME"; then
    echo "OK - $SERVICE_NAME is running"
    exit 0
else
    echo "CRITICAL - $SERVICE_NAME is not running"
    exit 2
fi
```

Рисунок 2.6 – Приклад інтегрування кастомного плагіна[18]

Налаштування параметрів моніторингу.

Для всіх плагінів важливо налаштувати оптимальні пороги, які дозволять уникнути хибних спрацьовувань або, навпаки, пропуску важливих подій. Налаштування для основних показників показано в таблиці 2.2:

Таблиця 2.2 - Основні показники плагінів веб-серверу

Показник	Поріг Warning	Поріг Critical	Коментар
CPU Load	70%	90%	Залежить від типу сервера та навантаження
Memory Usage	80%	95%	Чим більше пам'яті, тим вищий поріг
Disk Space	15% вільного	5% вільного	Для основних розділів налаштування строгіше
Temperature (°C)	70	85	Критичні значення можуть варіюватися
Network Latency	100ms	200ms	Для внутрішніх і зовнішніх мереж по-різному

Інтеграція плагінів з оповіщенням.

Налаштування оповіщень є важливим елементом моніторингу. Nagios дозволяє налаштувати надсилання повідомлень за допомогою email, SMS або месенджерів при досягненні критичних порогів. Оповіщення можна кастомізувати за такими параметрами, як час отримання (наприклад, тільки робочі години), група отримувачів та кількість повторів.

Переваги використання плагінів Nagios:

– Гнучкість: можливість вибору готових або кастомних плагінів для повного покриття вимог.

– Масштабованість: плагіни дозволяють масштабувати моніторинг на нові сервери або додаткові параметри.

– Оповіщення: зручна система сповіщень для швидкої реакції на проблеми.

Таке налаштування забезпечує ефективний моніторинг серверного обладнання та дозволяє знизити ризики простоїв чи втрати даних завдяки своєчасному виявленню проблем [20].

2.3 Конфігурація моніторингу серверів

Щоб налаштувати ефективний моніторинг у Nagios, важливо пройти кілька ключових етапів, починаючи з визначення всіх серверів і пристроїв, які необхідно моніторити, та закінчуючи точним налаштуванням хостів і служб у конфігураційних файлах. Наведемо детальний процес налаштування з прикладами.

Визначення серверів та пристроїв для моніторингу.

На початковому етапі потрібно скласти повний перелік серверів та інших критичних пристроїв, таких як маршрутизатори, комутатори, сховища даних, щоб їх включити в систему моніторингу. Для кожного пристрою збирається така інформація:

– IP-адреса або DNS-ім’я;

– Тип пристрою (сервер, маршрутизатор, сховище);

– Параметри моніторингу (які служби або ресурси потрібно контролювати, наприклад, ЦП, оперативна пам’ять, дисковий простір, мережеві порти тощо);

– Частота перевірок і пороги оповіщення для різних служб (нормальний рівень, попередження, критичний).

Налаштування хостів у конфігураційних файлах Nagios.

Конфігурація Nagios для моніторингу серверів базується на додаванні хостів та служб до конфігураційних файлів. Після визначення списку пристроїв та серверів можна додати їх у файл конфігурації. У Nagios, зазвичай, використовуються файли:

– nagios.cfg — головний файл конфігурації;

– hosts.cfg — файл для налаштування хостів;

– services.cfg — файл для налаштування служб.

У Додатку В наведено приклад додавання сервера у файл hosts.cfg. У цьому прикладі додаємо сервер бази даних з IP-адресою 192.168.1.10.

Налаштування служб у конфігураційних файлах.

Наступним кроком є додавання конкретних служб, які необхідно контролювати на кожному хості. Це може бути моніторинг доступності портів, використання ЦП або оперативної пам’яті, дискового простору тощо.

У Додатку Г надається приклад конфігурації служби для моніторингу доступності бази даних (порт 3306).

Налаштування груп серверів.

Для великих мереж рекомендується використовувати групи хостів, щоб зручно керувати серверними групами (наприклад, веб-сервери, сервери баз даних, поштові сервери). Приклад налаштування групи хостів у Nagios надається у Додатку Ґ.

Встановлення порогів і налаштування сповіщень.

Nagios дозволяє задавати пороги (наприклад, критичні значення використання ресурсів) для кожної служби. Приклад налаштування порогів для моніторингу використання дискового простору надається у Додатку Ґ.

Перевірка та перезапуск Nagios.

Після внесення змін до конфігураційних файлів необхідно перевірити конфігурацію та перезапустити Nagios. Приклад перевірки та перезапуску Nagios надається у Додатку Ґ.

2.4 Створення параметрів і політик сповіщень

Ефективне налаштування сповіщень у Nagios дозволяє адміністраторам оперативно реагувати на проблеми з конфігурацією серверного обладнання. Налаштування включає вибір каналів оповіщення, таких як електронна пошта, SMS тощо, а також встановлення рівнів критичності сповіщень залежно від параметрів конфігурації серверів [21].

Налаштування сповіщень про проблеми з конфігурацією серверного обладнання.

У Nagios можна налаштувати сповіщення через різні канали, залежно від доступності ресурсів та вимог компанії:

– Електронна пошта: Найпоширеніший спосіб, що дає можливість відправляти детальні повідомлення з описом проблеми. Налаштовується за допомогою команд у Nagios.

– SMS: Для критичних сповіщень, які потребують швидкої реакції. Потребує використання шлюзу SMS або інтеграції з сервісом сторонніх повідомлень, що можна налаштувати через спеціальні плагіни.

Приклад налаштування сповіщення через електронну пошту надається у Додатку Ґ.

Прив’язка сповіщень до хостів та служб: вказуємо, які хости та служби відправлятимуть повідомлення в разі збоїв.

Встановлення політик критичних і попереджувальних рівнів сповіщень.

Рівні сповіщень визначаються, щоб інформувати адміністраторів про проблеми на різних стадіях.

Встановлення політик попереджувальних (Warning) і критичних (Critical) сповіщень дозволяє оптимізувати реакцію на збої:

– Попереджувальний рівень (Warning): Вказує на потенційні проблеми (наприклад, 80% використання ЦП). Дає змогу адміністратору вжити заходів до настання критичної ситуації.

– Критичний рівень (Critical): Вказує на критичний стан (наприклад, 95% використання ЦП), який потребує негайної реакції.

Приклад політики для моніторингу дискового простору надається у Додатку Ґ.

2.5 Налаштування періодичного контролю конфігурації

Періодичний моніторинг у Nagios виконується шляхом налаштування інтервалів перевірок для різних хостів та служб. Це дозволяє регулярно отримувати дані про стан серверів і актуальність їх конфігурацій, наприклад,

навантаження ЦП, дисковий простір, час роботи, доступність служб тощо [22]. Параметри періодичності задаються для кожного ресурсу відповідно до вимог. Приклад надається у Додатку Ґ.

Завдяки цьому сервер буде періодично перевірятися на перевантаження процесора, що дозволяє вчасно виявити критичні зміни.

Для повного аналізу стану серверів корисно додати візуалізацію даних у вигляді графіків і звітів. Це дозволяє швидко оцінити динаміку змін, такі як збільшення навантаження або використання ресурсів, і прийняти відповідні дії.

Nagios підтримує інструменти візуалізації, такі як NagiosGraph, PNP4Nagios, або інтеграцію з Grafana для відображення динамічних графіків. Ці інструменти дають змогу автоматично формувати графіки для основних показників:

– CPU, RAM, Disk Usage: відображення зміни навантаження за обраний період.

– Температура і статус обладнання: графіки з температурними змінами для прогнозування потенційних перегрівів.

– Доступність мережі та служб: звіти про доступність, які допомагають виявити тренди, пов'язані з простоєм.

Приклад інтеграції з PNP4Nagios для візуалізації:

1. Встановлення PNP4Nagios для збору і зберігання даних у базі.

2. Налаштування шаблонів візуалізації для різних типів перевірок (наприклад, використання CPU, пам’яті).

3. Автоматична генерація графіків: створення динамічних графіків, які можна переглядати безпосередньо в інтерфейсі Nagios.

Переваги періодичного моніторингу та візуалізації даних.

1. Прогнозування і попередження проблем: регулярний моніторинг та графіки допомагають виявити тренди (наприклад, стабільне збільшення використання диска) і запобігти проблемам.

2. Швидкий аналіз: завдяки графікам і звітам адміністратор може швидко оцінити зміни й ухвалити рішення про оптимізацію налаштувань.

3. Історія змін конфігурацій: історичні дані дозволяють переглянути динаміку стану сервера і виявити фактори, що впливають на продуктивність.

Використання періодичного моніторингу разом із візуалізацією дає можливість ефективно відслідковувати стан серверів та забезпечує надійний інструмент для управління інфраструктурою.

2.6 Звітування та документування

Звітування та документування є важливими елементами моніторингу серверного обладнання, що дозволяють не лише відстежувати стан систем, а й аналізувати тенденції роботи серверів та історію змін у конфігураціях. Розглянемо підходи до генерації звітів та ведення журналів за допомогою розширень Nagios [23].

Генерація звітів для аналізу роботи серверного обладнання.

Nagios пропонує кілька варіантів звітності, щоб допомогти адміністраторам аналізувати стан серверів та серверного обладнання, відстежувати їхню продуктивність та швидко виявляти потенційні проблеми. Основні види звітів включають:

– Звіти про доступність: Відображають, який відсоток часу сервери та служби були доступні протягом певного періоду. Це допомагає зрозуміти, наскільки стабільною є інфраструктура та чи існують сервери або служби з частими збоями.

– Звіти про продуктивність: Оцінюють, наскільки оптимально використовуються ресурси, такі як ЦП, оперативна пам'ять та дисковий простір. На основі цих даних можна побачити, коли сервери починають перевантажуватися і потребують оптимізації.

– Звіти про тренди: Допомагають побачити тенденції у використанні ресурсів, що дозволяє прогнозувати можливі майбутні проблеми, як-от недостатність дискового простору чи пам'яті.

Приклад генерації звітів.

Для зручності аналізу звіти можна налаштувати на автоматичну генерацію в кінці робочого дня або тижня. Наприклад, за допомогою інструмента Nagios XI можна створювати регулярні звіти, що містять інформацію про продуктивність серверів і статус їхніх служб за обраний період.

Процес виглядає наступним чином:

1. Вибір параметрів звіту, таких як період, тип звіту та тип обладнання.

2. Автоматичне формування звітів за обраним графіком (щоденно, щотижнево, щомісячно).

3. Надсилання звітів на електронну пошту для оперативного доступу до результатів моніторингу.

Такі звіти дозволяють адміністраторам бачити актуальні дані про стан серверів без необхідності входу в систему та ручного збору інформації.

Використання розширень Nagios для створення журналів змін у конфігураціях серверів.

Для більш глибокого аналізу і точного відстеження історії змін у конфігураціях Nagios можна налаштувати збирання журналів за допомогою розширень, таких як NagiLog Server або Log Server, а також зберігати та структурувати дані у базах даних (наприклад, MySQL). Журнали змін є корисним інструментом для вивчення причин виникнення проблем і їх вирішення [24].

Журнали в Nagios дозволяють зберігати дані про:

– Зміни конфігурації: Кожна зміна у налаштуваннях серверів та служб фіксується у журналі, що допомагає бачити, коли і ким була внесена зміна.

– Збої та аварійні ситуації: Інформація про всі збитки в роботі сервера і помилки служб записується у журнал, що полегшує аналіз післявідмовних подій.

– Відновлення та дії: Фіксуються дані про те, які дії були вжиті для вирішення проблеми, що є важливим для зворотного зв’язку і покращення процедур реагування.

Приклад використання розширення для журналів.

Одним з найпоширеніших розширень є Nagios Log Server. Він дозволяє централізовано зберігати журнали з різних серверів, що спрощує доступ до них і надає можливості для пошуку та аналізу. Ось як налаштовується робота з логами:

1. Налаштування зберігання журналів: логи зберігаються на спеціальному сервері, де з ними можна працювати централізовано.

2. Автоматична фільтрація та організація: Log Server дозволяє розділяти логи за категоріями, зокрема змін конфігурації, помилок і повідомлень відновлення.

3. Моніторинг та аналіз логів: за допомогою вбудованих інструментів адміністратори можуть знаходити специфічні зміни або збої, аналізувати причини і зберігати шаблони проблем для швидшого вирішення в майбутньому.

Журнали забезпечують надійну базу для аудиту та відстеження змін, що допомагає не лише у випадку збою, але й для оптимізації роботи серверів та створення довгострокових стратегій їхньої підтримки.

Переваги звітності та ведення журналів у Nagios.

1. Вчасне виявлення та вирішення проблем: звіти дозволяють швидко виявляти проблеми, що впливають на продуктивність серверів, і вживати заходів до їхнього загострення.

2. Прозорість та відстежуваність змін: журнали надають адміністраторам повну інформацію про всі зміни конфігурацій, що підвищує рівень контролю та обізнаності.

3. Історичні дані для аналізу: на основі зібраної інформації можна не лише вивчити причини поточних проблем, а й прогнозувати потреби в ресурсах.

Налаштування звітності та документування за допомогою Nagios забезпечує детальне розуміння процесів у серверному середовищі та дозволяє швидше реагувати на зміни [25].

2.7 Забезпечення безпеки та розмежування доступу

Забезпечення розмежування прав доступу, шифрування та аутентифікація користувачів – це основні компоненти безпечного середовища. Розглянемо ці компоненти докладніше.

Впровадження правил доступу та безпеки для адміністраторів і користувачів системи моніторингу.

Приклад надається у Додатку Д. Nagios дозволяє створювати індивідуальні облікові записи для різних ролей користувачів, наприклад, адміністраторів, технічних фахівців та інших співробітників. Цей підхід дозволяє налаштувати права доступу залежно від обов’язків кожного користувача:

Адміністратори: отримують повний доступ до всіх функцій системи моніторингу, включаючи налаштування хостів, служб, політик сповіщення та керування обліковими записами.

Оператори: мають обмежений доступ для моніторингу певних хостів і служб, з можливістю перегляду, але без права змінювати налаштування системи.

Користувачі: можуть лише переглядати інформацію про стан серверів, які вони контролюють, без можливості редагування чи впливу на інші служби.

Для захисту даних, що передаються між Nagios і серверними пристроями, необхідно налаштувати шифрування каналів зв’язку за допомогою SSL/TLS. Це забезпечує захищений зв'язок, зменшуючи ризик перехоплення даних під час їх передавання через мережу.

Налаштування шифрування SSL/TLS.

1. Встановлення SSL-сертифіката: Генерується SSL-сертифікат і ключ для сервера, на якому розгорнуто Nagios.

2. Конфігурація веб-сервера (наприклад, Apache) для використання HTTPS. У конфігурації Apache налаштовуються порти для SSL-з’єднання та шляхи до сертифіката і ключа.

3. Переналаштування Nagios для роботи через HTTPS: Після налаштування веб-сервера користувачі отримають доступ до інтерфейсу Nagios через захищене HTTPS-з'єднання.

Аутентифікація користувачів Nagios підтримує базову аутентифікацію користувачів через веб-сервер або за допомогою LDAP, якщо потрібно інтегруватися з корпоративним каталогом користувачів. Це дозволяє централізовано керувати обліковими записами, спрощуючи доступ і захищаючи систему, а саме: створення облікових записів з паролями, налаштування аутентифікації у веб-сервері [26].

Переваги забезпечення безпеки та розмежування доступу:

Захист даних. Шифрування та аутентифікація знижують ризики несанкціонованого доступу.

Контроль доступу. Чітке розмежування прав дозволяє контролювати, хто має доступ до конфіденційної інформації та команд управління.

Гнучкість. Індивідуальні налаштування для різних рівнів користувачів дозволяють зберігати високу ефективність роботи, забезпечуючи водночас захист системи.

Таким чином було проведено дослідження програмної реалізації моніторингу серверного обладнання на базі інструменту Nagios.

3 РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ КОНФІГУРАЦІЇ СЕРВЕРНОГО ОБЛАДНАННЯ НА БАЗІ ІНСТРУМЕНТУ NAGIOS

3.1 Розробка методики тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios

3.1.1 Зміст методики тестування системи моніторингу конфігурацій серверного обладнання на базі інструменту Nagios. Системні адміністратори, відповідальні за підтримку критично важливих серверів і додатків, які забезпечують роботу близько 100 пристроїв на базі Windows і Linux. Їх завдання – гарантувати стабільність і безперебійність роботи систем із доступністю на рівні 99,8% протягом року.

Під час експлуатації деякі системи виходять з ладу. Відчуття розгубленості та розчарування через неможливість швидко виявити та усунути проблему може мати серйозні наслідки. Саме такі виклики спонукали до створення Nagios Core – потужного інструменту моніторингу з відкритим кодом.

Nagios Core дозволяє віддалено контролювати стан систем і сервісів, таких як завантаження процесора, використання диска й пам’яті, а також роботу критичних протоколів і служб (SMTP, HTTP, POP3, SNMP, ICMP, FTP, SSH тощо). Інструмент забезпечує своєчасні сповіщення у разі збоїв, що дає змогу оперативно реагувати та мінімізувати вплив на бізнес-процеси [27].

Тестування системи моніторингу на базі Nagios передбачає оцінку її функціональності, продуктивності, надійності та зручності використання.

Методика тестування системи моніторингу конфігурації серверного обладнання на базі Nagios включає кілька послідовних етапів, які допомагають перевірити функціональність, продуктивність, масштабованість та надійність

системи. Для цього потрібно виконати кілька послідовних етапів, кожен із яких має свою мету, сценарії тестування та критерії оцінки результатів [28].

Етапи методики тестування.

Етапи методики тестування наступні:

1. Планування тестування.

- *Визначення цілей тестування*, що передбачає: перевірку правильності збору даних про конфігурації серверів; оцінку швидкості оновлення даних; тестування роботи системи в умовах навантаження; тестування точності сповіщень та їх своєчасності; аналіз стабільності роботи системи у тривалих сесіях.
- *Розробка тестових сценаріїв*, що передбачає: тестування базових функцій (доступність серверів, моніторинг ресурсів); перевірка користувацьких плагінів для збору специфічних даних; аналіз поведінки системи під час помилок (збій сервера, недоступність мережі).
- *Визначення критеріїв успішності*, що передбачає: час реагування на зміни; відсоток коректно отриманих даних; мінімальний вплив системи моніторингу на продуктивність серверів.

2. Підготовчі дії.

- *Розгортання тестового середовища*, що передбачає: налаштування серверів різної конфігурації (операційні системи, апаратні параметри); встановлення Nagios Core та необхідних плагінів для моніторингу серверів; налаштування віддалених серверів через NRPE, SSH або SNMP.
- *Визначення набору тестів, що передбачає*: збір базових метрик (стан процесора, пам'яті, дисків); моніторинг доступності серверів (ping, мережеві порти); тестування збору специфічних конфігурацій (версії ОС, програмного забезпечення).
- *Налаштування сповіщень*, що передбачає: налаштування електронної пошти, SMS або месенджерів для отримання сповіщень про збої;
- *Інструменти для аналізу*, що передбачає: використання логів Nagios для вивчення помилок.

- *Підключення зовнішніх інструментів* (Grafana, Kibana), що передбачає аналіз продуктивності.

3. Налаштування середовища тестування.

- *Інфраструктура для тестування,* що передбачає**:** встановлення серверу Nagios (основного вузла моніторингу); налаштування кількох серверів для моніторингу (фізичних, віртуальних); інтеграція з віддаленими вузлами через NRPE, SNMP або SSH.

- *Налаштування моніторингу,* що передбачає**:** впровадження стандартних плагінів для моніторингу загальних параметрів; розробку користувацьких плагінів для перевірки конфігурації (наприклад, версій ОС, встановлених програм).

- *Симуляція навантаження,* що передбачає**:** імітацію високих навантажень на сервери; генерацію помилкових умов (відключення мережі, несправність дисків).

4. Виконання тестів (проведення різних видів тестування).

- *Функціональне тестування,* що передбачає**:** перевірку кожного моніторингового сценарію, а саме: чи правильно система збирає та обробляє дані; тестування плагінів, а саме: чи повертають вони очікувані результати; перевірку точності сповіщень у разі порушення параметрів (наприклад, при перевищенні використання CPU на 80%).

- *Тестування продуктивності,* що передбачає*:* впровадження навантажувальних тестів шляхом додавання великої кількості хостів до системи (наприклад, 1000 серверів) чи збільшення частоти збору даних та оцінка часу відповіді системи під час роботи в умовах пікового навантаження.

- *Тестування відмовостійкості,* що передбачає**:** імітацію збоїв (відключення мережі, збій служби); визначення швидкості реагування системи на збої.

- *Тестування масштабованості,* що передбачає**:** додавання нових серверів для моніторингу в реальному часі; перевірка роботи у розподіленій архітектурі (з використанням Nagios Distributed Monitoring).

Види сценарії тестування.

Тестування проводиться за кількома сценаріями зміст яких наступній:

- *Функціональне тестування***:** перевірка коректності збору базових метрик (завантаженість CPU, RAM, стан дискових систем); тестування збору специфічних параметрів конфігурації (наприклад, версії ядра ОС, стану мережевих інтерфейсів).
- *Тестування продуктивності***:** вимірювання часу від спрацювання тригера до оновлення даних у веб-інтерфейсі; аналіз швидкості опитування серверів у великій інфраструктурі.
- *Тестування під навантаженням***:** моніторинг сотень серверів з різними параметрами конфігурації; аналіз стабільності роботи системи за умов високої кількості запитів.
- *Тестування надійності:* перевірка роботи Nagios у разі збою підключення до віддалених вузлів.
- *Тестування механізмів відновлення після помилок*.
- *Тестування безпеки*: оцінка захисту даних при передачі через мережу (використання SSL/TLS); тестування доступу до інтерфейсу моніторингу для авторизованих користувачів.

Види критеріїв оцінки результатів тестування.

Після завершення тестування результати аналізуються за кількома критеріями:

- *Коректність роботи системи*: відсоток успішно зібраних даних; випадки помилок і способи їх усунення.
- *Швидкість роботи*: середній час реагування на зміни; час генерації сповіщень.
- *Стабільність*: поведінка системи під час високого навантаження; аналіз журналів помилок Nagios.
- *Вплив на сервери*: рівень споживання ресурсів (CPU, RAM) системою моніторингу.

- *Відповідність вимогам*: Чи задовольняє система поставлені завдання (моніторинг потрібних конфігурацій).

Оцінка результатів.

Оцінка результатів здійснюється на основі наступних дій або оцінок:

- *Успішність тестування*: якщо 95% і більше тестів виконано без помилок, систему можна вважати придатною для впровадження.
- *Продуктивність*: час реакції системи не перевищує 5-10 секунд у реальному середовищі.
- *Стабільність*: система не припиняє роботу під час симуляції помилок або великого навантаження.
- *Вплив на сервери*: система моніторингу використовує не більше 5-10% ресурсів моніторингового сервера.

Висновки за результатами тестування.

Висновки за результатами тестування здійснюються на осонові аналізу:

1. Оцінки відповідності функціональності Nagios специфічним вимогам моніторингу конфігурації серверного обладнання.
2. Ідентифікації потенційних вузьких місць (наприклад, потреба в оптимізації налаштувань опитування).
3. Відповідності рекомендації щодо використання Nagios у виробничих середовищах (застосування плагінів, підходи до масштабування).

Рекомендації на основі результатів комплексного тестування.

На основі результатів комплексного тестування розробляються наступні рекомендації щодо:

- *Оптимізації конфігурації Nagios*: налаштування частоти перевірок і порогів для кожного параметра; оптимізації бази даних для зберігання результатів перевірок.
- Покращення масштабованості: впровадження розподіленого моніторингу; використання проксі-серверів для зменшення навантаження на основний сервер Nagios.

- Інтеграції з іншими системами: використання зовнішніх інструментів для візуалізації (Grafana) чи аналізу даних (ELK-стек).
- Автоматизація реагування: налаштування автоматичного запуску скриптів у разі виникнення проблем.

Таким чином запропонована методика дозволяє отримати комплексне уявлення про можливості Nagios та забезпечити його ефективну інтеграцію в інфраструктуру. Така методика тестування забезпечує детальну оцінку роботи системи моніторингу конфігурації серверного обладнання на базі Nagios, дозволяючи адаптувати її до потреб конкретної IT-інфраструктури [29].

3.1.2 Опис виконання методики тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios. Методика тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios наступна (Рис. 3.1).

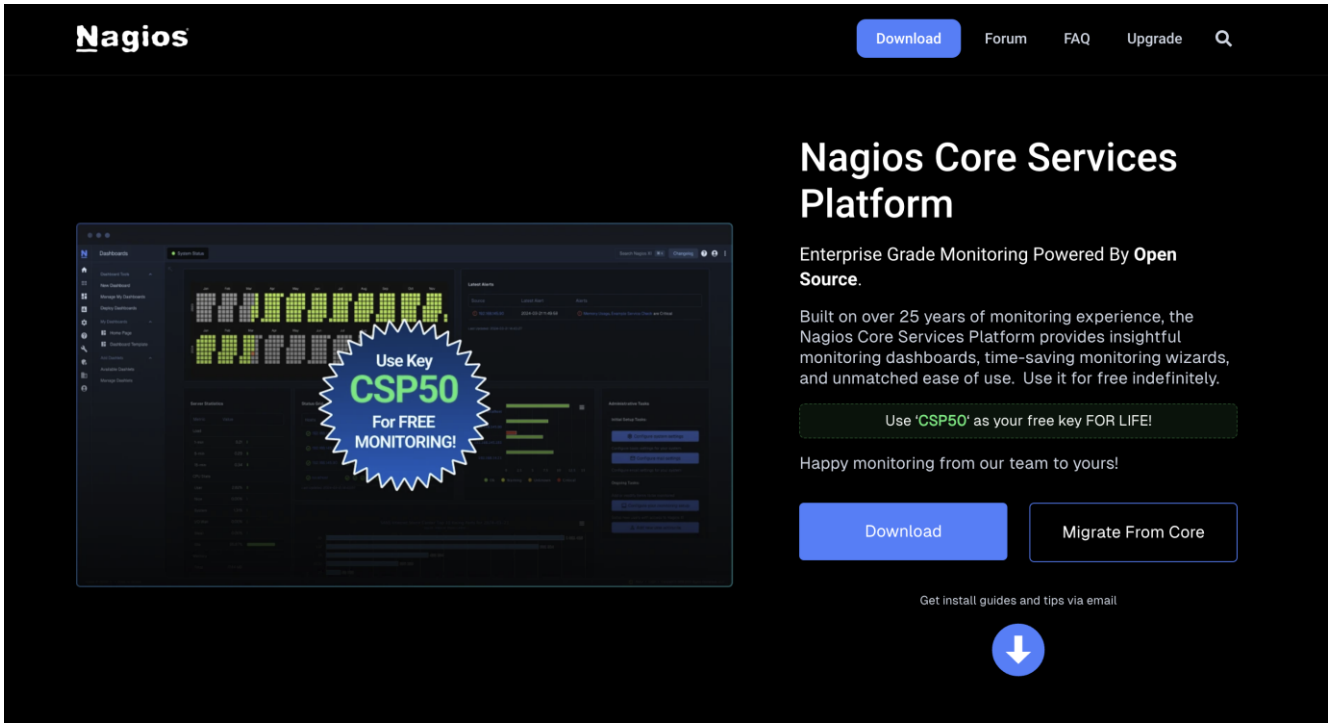


Рисунок 3.1 – Ядро Nagios [30]

У додатку Е надається розгорнутий опис методики тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios. Нижче надається її стислий опис.

Крок 1. Спершу потрібно виконати вхід до серверного середовища Oracle Linux 7 під користувачем із правами root. Це дозволить виконувати всі необхідні команди для налаштування системи. Налаштуємо брандмауер для доступу до веб-інтерфейсу Nagios Щоб зробити веб-інтерфейс Nagios доступним із зовнішніх машин, виконаємо відповідну команду в терміналі.

Крок 2. Потрібно вимкнути ***SELINUX***, відредагувавши файл конфігурації, розташований за адресою: */etc/selinux/* і змінивши ***SELINUX*** із примусового на вимкнений.

Крок 3. Потрібно встановити залежні пакети nagios.

Крок 4. Завантажимо Nagios Core Source.

Крок 5. Створимо користувачів Nagios і apache, створимо групу та додамо їх до цієї групи.

Крок 6. Розархівуємо файл завантаження з кроку 4 та змінимо каталог на папку, що містить завантажений файл.

Крок 7. Скомпілюємо Nagios із вихідного коду. Виконаємо заміну каталогу на папку, що містить розархівований файл із кроку 6. Потрібно зробити інсталяцію.

Крок 8. Все ще в розархівованій папці (/nagios-4.4.3/), скопіюємо файли Evenhandler до /usr/local/nagios/libexec/

Крок 9. Змінимо дозволи обробників подій.

Крок 10. Створимо користувача nagios (для входу в Інтернет) і згенеруємо його пароль.

Крок 11. Запустимо веб-службу, увімкнення її автоматичного запуску після перезавантаження сервера. Запустимо веб-сервіс та увімкнення автоматичного запуску після перезавантаження. Потрібно перевірити стан веб-служби, щоб переконатися, що вона працює.

Крок 12. Запуск служби Nagios. Запустимо сервери Nagios, увімкнення автоматичного запуску після перезавантаження. Перевіримо статус служби nagios, щоб переконатися, що вона працює.

Крок 13. Завантажимо плагіни Nagios.

Крок 14. Розпакуймо завантажені плагіни Nagios. змінимо каталог на папку, що містить завантажений файл.

Крок 15. Встановимо плагіни Nagios. Змінимо каталог на папку, що містить розархівовані файли з кроку 14.

Крок 16. Завантажимо файл NRPE, в папку tmp.

Nagios успішно зконфігуровано та налаштовано. Увійдемо до веб-інтерфейсу nagios за допомогою ip-адреси сервера nagios, а потім /nagios: http://192.168.20.131/nagios/. Буде запропоновано ввести облікові дані для входу, використовуйте облікові дані, створені на кроці 10.

3.2 Інтеграція Windows-системи в моніторингове середовище на базі інструменту Nagios

3.2.1 Зміст інтеграції Windows-системи в моніторингове середовище на базі інструменту Nagios. Інтеграція Windows-системи в моніторингове середовище на базі Nagios включає налаштування відповідного середовища для збору даних із Windows-серверів, використання плагінів або агентів, а також проведення тестування та аналізу отриманих результатів. Нижче представлено покроковий план інтеграції та тестування.

У додатку Ж надається розгорнутий опис покрокового плану інтеграції та тестування. Нижче наведено стислий його опис.

Крок 1. **Підготовка Windows-системи для інтеграції.** Він передбачає:

1.1. Встановлення агента моніторингу. Nagios використовує агенти для збору даних із Windows-систем. Найпоширеніші агенти:

- **NSClient++**: Забезпечує моніторинг ресурсів Windows, таких як CPU, пам'ять, дисковий простір, служби, події.

- **WMI (Windows Management Instrumentation)**: Може використовуватися для безагентського моніторингу через плагіни Nagios.

Дії:

1. Завантажити та встановити NSClient++ на Windows-сервер.
2. Налаштувати файл конфігурації nsclient.ini для доступу з Nagios-сервера (активувати потрібні модулі (CheckDisk, CheckSystem, NRPE тощо)).
3. Дозволити підключення через порт (зазвичай 5666) у брандмауері Windows.

1.2. Налаштування дозволів. Для безагентського моніторингу через WMI треба:

- Забезпечити доступ до WMI-інтерфейсу на Windows-сервері.
- Налаштувати обліковий запис із правами на читання конфігурацій через WMI.

Крок 2. **Налаштування Nagios для моніторингу Windows.** Він передбачає:

2.1. Додавання хоста. Додати інформацію про Windows-систему в конфігураційний файл Nagios.

2.2. Додавання перевірок (services). Додати служби для моніторингу.

2.3. Перевірка плагінів. Переконатися, що плагіни для перевірки Windows-систем встановлені на сервері Nagios:

- *check_nt*: Використовується з NSClient++.
- *check_wmi_plus*: Дозволяє збирати дані через WMI.

Крок 3. **Тестування системи моніторингу.** Він передбачає:

3.1. Проведення тестових перевірок. Перевірити, чи відповідає агент і чи отримуються дані.

3.2. Створення сценаріїв навантаження. Створити штучні навантаження на сервер (наприклад, запустити ресурсоємні процеси); перевірити, як система моніторингу фіксує зміни.

Крок 4. **Аналіз результатів тестування.** Він передбачає:

4.1. Збір даних. Зібрати метрики, що відображають продуктивність та конфігурацію Windows-систем: середнє навантаження CPU; використання пам'яті;

обсяг доступного дискового простору; стан ключових служб (IIS, SQL Server тощо).

4.2. Оцінка точності та ефективності. Оцінка точності та ефективності виконується через оцінку: точності (порівняти дані, зібрані Nagios, із реальними показниками на сервері); час відгуку (оцінити затримки між виявленням проблеми і створенням сповіщення); ресурсомісткість (аналіз навантаження на Windows-сервер і сервер Nagios під час моніторингу).

4.3. Оцінка стабільності. Оцінка стабільності шляхом перевірки - чи система моніторингу залишається стабільною при тривалому спостереженні та великій кількості запитів.

Крок 5. **Оптимізація моніторингу.** Він передбачає:

- **Зниження навантаження:** Встановити оптимальну частоту перевірок для зменшення використання ресурсів.
- **Визначенн фільтра подій:** Виключити несуттєві перевірки, щоб уникнути надмірної кількості сповіщень.
- **Інтеграцію з Grafana:** Підключити Grafana для візуалізації даних і покращення аналізу.

Крок 6. **Висновки за результатами.** Він передбачає:

1. **Функціональність Nagios:** Оцінити, наскільки інструмент відповідає вимогам моніторингу Windows-систем.
2. **Продуктивність:** Визначити, чи система справляється з великим навантаженням і чи надає актуальні дані.
3. **Придатність до масштабування:** Перевірити можливість інтеграції з іншими інструментами та додатковими хостами.

Такий підхід забезпечує повноцінну інтеграцію Windows-систем у моніторингове середовище, а також дає змогу оцінити ефективність і надійність Nagios у роботі з гетерогенними системами.

3.2.2 Рекомендації щодо інтеграції Windows-системи в моніторингове середовище на базі інструменту Nagios. Рекомендації щодо інтеграції Windows-системи в моніторингове середовище на базі Nagios потрібні для забезпечення ефективного, стабільного та повного контролю за станом Windows-систем у корпоративній чи іншій IT-інфраструктурі. Такі рекомендації допомагають налаштувати моніторинг, уникнути типових помилок і оптимізувати процеси адміністрування. Основні види таких рекомендацій:

1. Забезпечення моніторингу критичних метрик. Ця рекомендація допомагає визначити, які саме метрики важливі для бізнесу, і як їх налаштувати, а саме: моніторинг Windows-систем із Nagios дозволяє контролювати: використання процесора, пам’яті, дискового простору; стан служб (Windows Services) і процесів; стан мережевих підключень (порти, затримки, доступність); події в системному журналі (Windows Event Log); стан оновлень безпеки та статус антивірусних програм.

2. Інтеграція зі специфічними Windows-технологіями. Ця рекомендація вказує, як налаштувати моніторинг цих служб за допомогою плагінів і спеціальних скриптів тому, що Windows-системи часто використовуються з такими технологіями: **Active Directory:** моніторинг стану контролерів домену; **IIS:** контроль за станом веб-сервера; **SQL Server:** перевірка продуктивності бази даних.

3. Вибір і налаштування протоколу збору даних. Така рекомендація уточнює, який протокол краще використовувати для конкретних цілей. Мова йде про те що для інтеграції Windows-систем із Nagios зазвичай використовуються: NSClient++(агент для збору даних із Windows-систем); WMI (Windows Management Instrumentation) (моніторинг без агента через мережеві запити); NRPE (Nagios Remote Plugin Executor) (для виконання плагінів на віддаленому сервері); SNMP (Simple Network Management Protocol) (для отримання базової інформації про стан системи).

4. Оптимізація налаштувань і продуктивності. Така рекомендація дозволяє: налаштувати частоту перевірок для мінімізації навантаження на сервер;

визначити критичні пороги для сповіщень (warning, critical); забезпечити стабільність роботи агента (наприклад, NSClient++).

5. Безпека інтеграції. Windows-системи можуть містити конфіденційну інформацію, тому важливо: використовувати захищені протоколи (наприклад, TLS/SSL для NRPE або HTTPS для NSClient++); обмежувати доступ до моніторингових портів; забезпечувати відповідність політикам безпеки підприємства.

6. Автоматизація. Ця рекомендація охоплює: використання автоматизованих скриптів або інструментів (Ansible, PowerShell) для масового розгортання та налаштування агентів; налаштування шаблонів конфігурацій для типових Windows-систем.

7. Документування процесів. Ця рекомендація передбачає правильну інтеграцію: документування всіх налаштувань (списки серверів, метрики, конфігураційні файли); надання інструкцій для подальшої підтримки та масштабування.

Виконання вказаних рекомендацій надає таки переваги:

1. Зменшення часу на інтеграцію Windows-систем.
2. Гарантія стабільності та точності зібраних даних.
3. Спрощення процесу діагностики та усунення проблем.
4. Підвищення загальної ефективності моніторингового середовища.

Таким чином рекомендації особливо важливі для компаній, де одночасно використовується велика кількість Windows-систем і де критично важливо забезпечити безперервність роботи.

У додатку И надається розгорнутий опис рекомендацій щодо інтеграції Windows-системи в моніторингове середовище на базі Nagios. Нижче надається їх стислий опис. Приклад завантаження компонентів та встановлення NSCP NG наведено на (рис. 3.2).

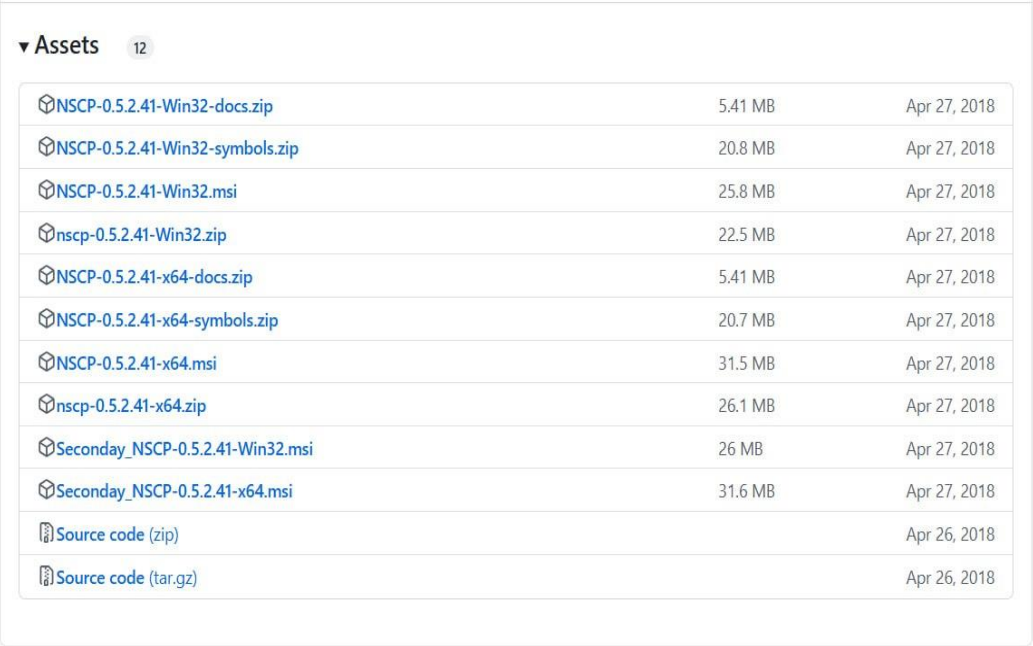


Рисунок 3.2 – Встановлення агента NSCP NG [32]

Ці рекомендаціїї передбачають наступні кроки:

Крок 1. *Завантаження та встановлення NSClient++.*

Розпочнемо з завантаження інструменту NSClient++, який необхідний для інтеграції Windows-систем у середовище моніторингу Nagios. Перейдімо за посиланням https://nsclient.org/download/0.5.2/#0.5.2.39 та завантажимо останню доступну версію (0.5.2.39). Після завершення завантаження запустимо інсталятор і виконаємо покрокову установку, враховуючи налаштування, які забезпечать сумісність із Nagios (рис. 3.3-3.5). Це дозволить підключити Windows-систему до моніторингового середовища та почати відстеження її стану й параметрів.

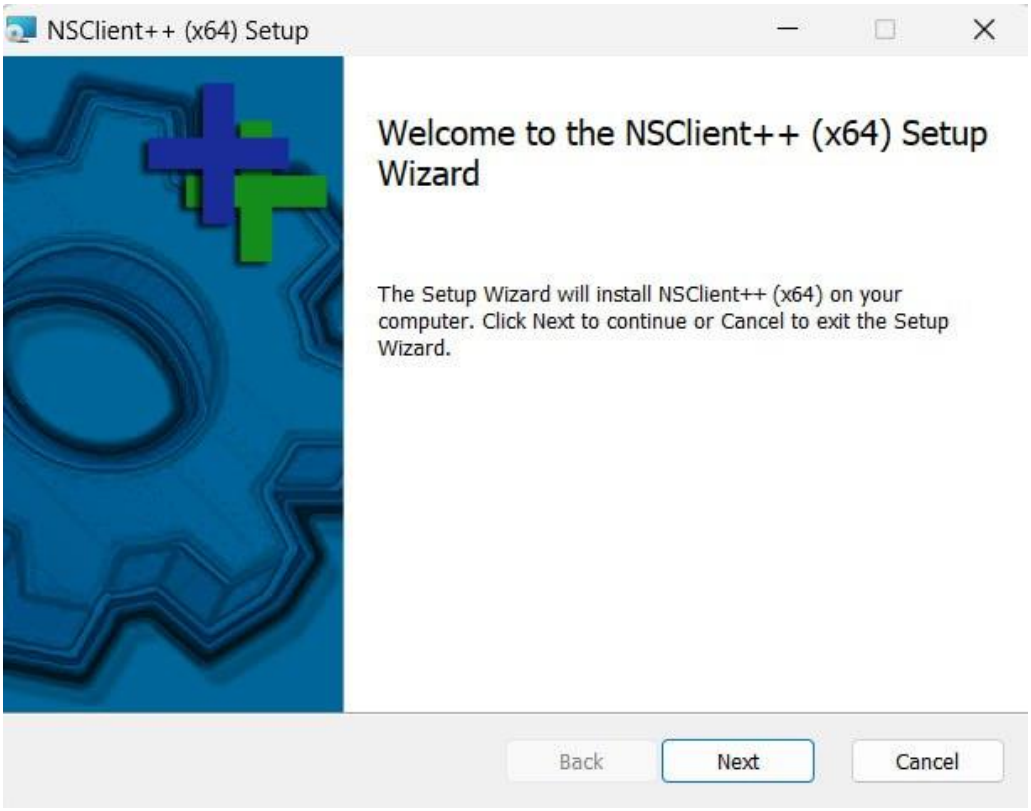


Рисунок 3.3 – Пакет NSClient.msi[31]

Введемо IP-адресу сервера Nagios Core у полі «Дозволені хости» та пароль у полі «Пароль», виберемо стандартний запропонований системою (рис. 3.4).

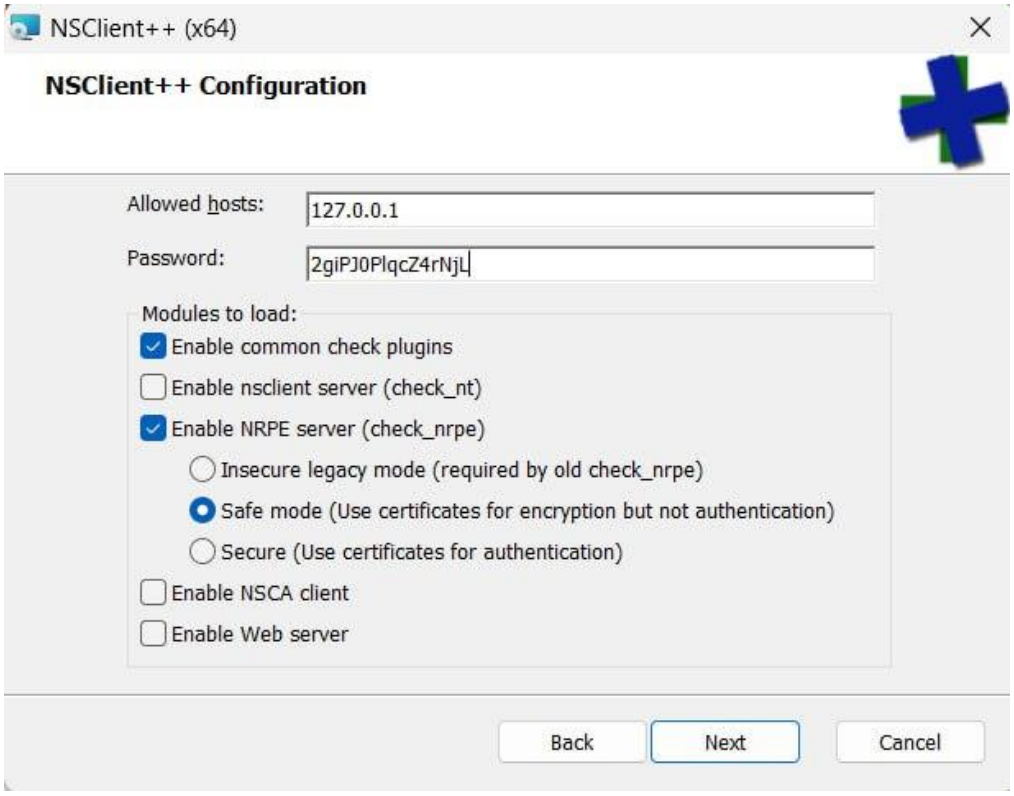


Рисунок 3.4 – Конфігурація агента [31]

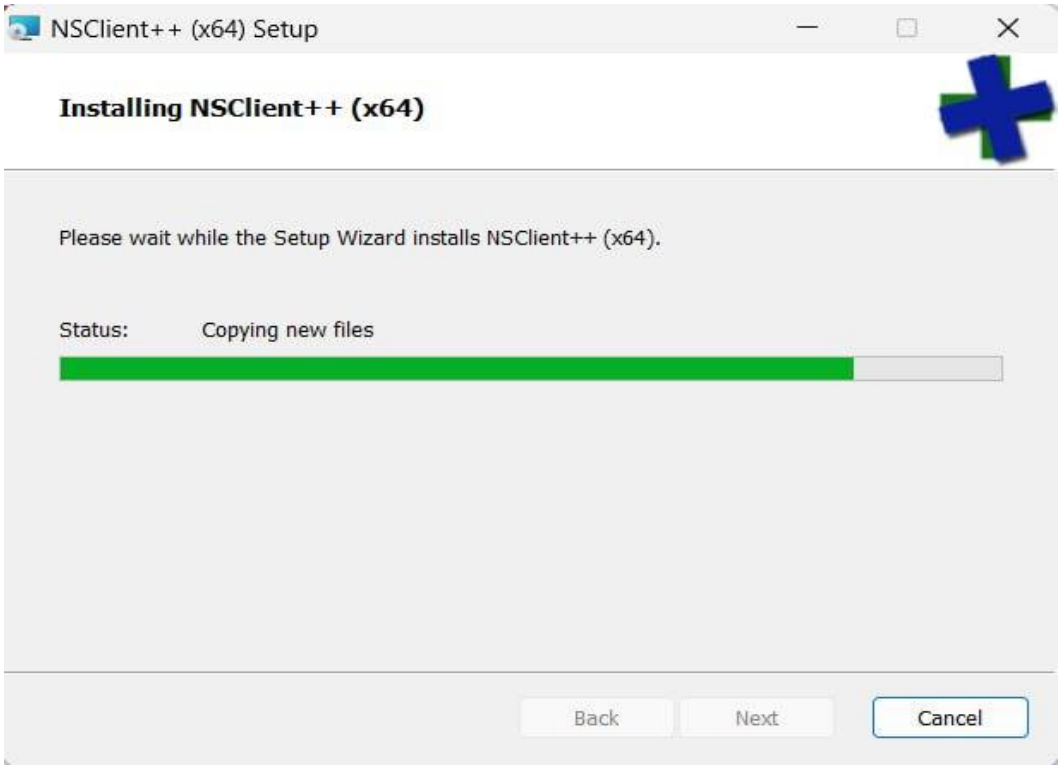


Рисунок 3.5 – Процес встановлення[31]

Крок 2. *Додамо вікно Windows*, додавши запис у файл windows.cfg, розташований за адресою (на сервері Nagios): /usr/local/nagios/etc/objects.

За замовчуванням у Nagios увімкнено наступні служби Windows (/usr/local/nagios/etc/objects/windows.cfg), також можемо смостійно додати будь-яку службу, яку бажаємо контролювати.

Крок 3. Далі, на сервері Nagios, відкриємо nagios.cfg, розташований за адресою: /usr/local/nagios/etc/.

Крок 4. Змінимо файл nsclient.ini, розташований у C:\Program Files\NSClient++.

Крок 5. Дозволимо порт 12489 через брандмауер Windows (рис. 3.6).

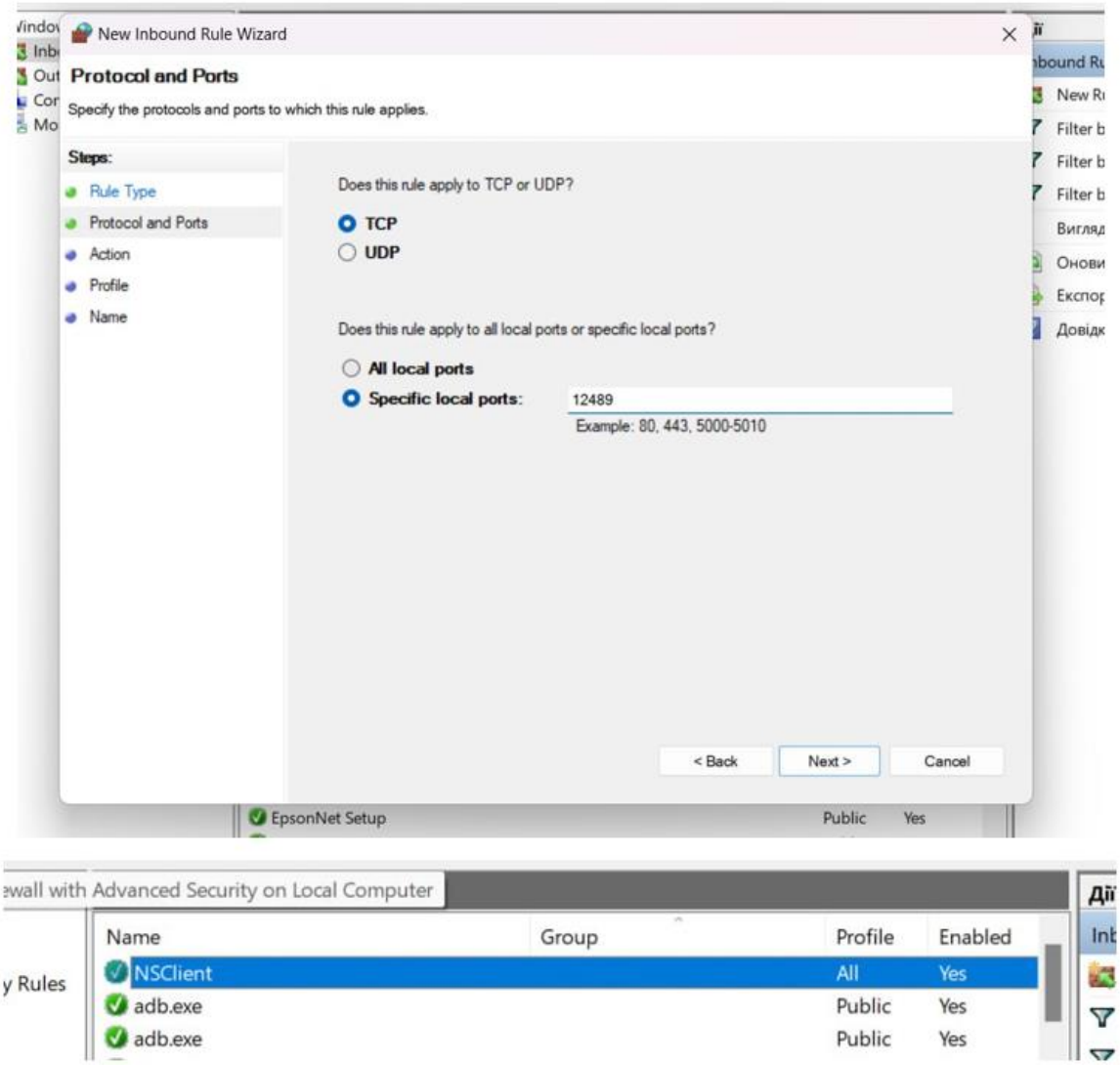


Рисунок 3.6 – Включене правило на відкриття порту 12489[32]

Крок 6. Перезапустимо службу агента моніторингу NSClient++ (рис. 3.7).

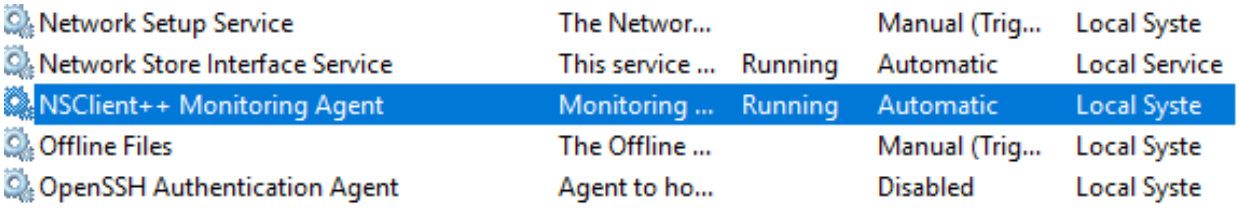


Рисунок 3.7 – Перезапуск служби[32]

Вікно Windows для моніторингу сервером Nagios успішно налаштовано. Увійдемо до веб-інтерфейсу nagios за допомогою ip-адреси сервера nagios, а потім http://192.168.20.131/nagios/. Повиннен відобразитися статус сервера Windows на Nagios (рис. 3.8-3.9).

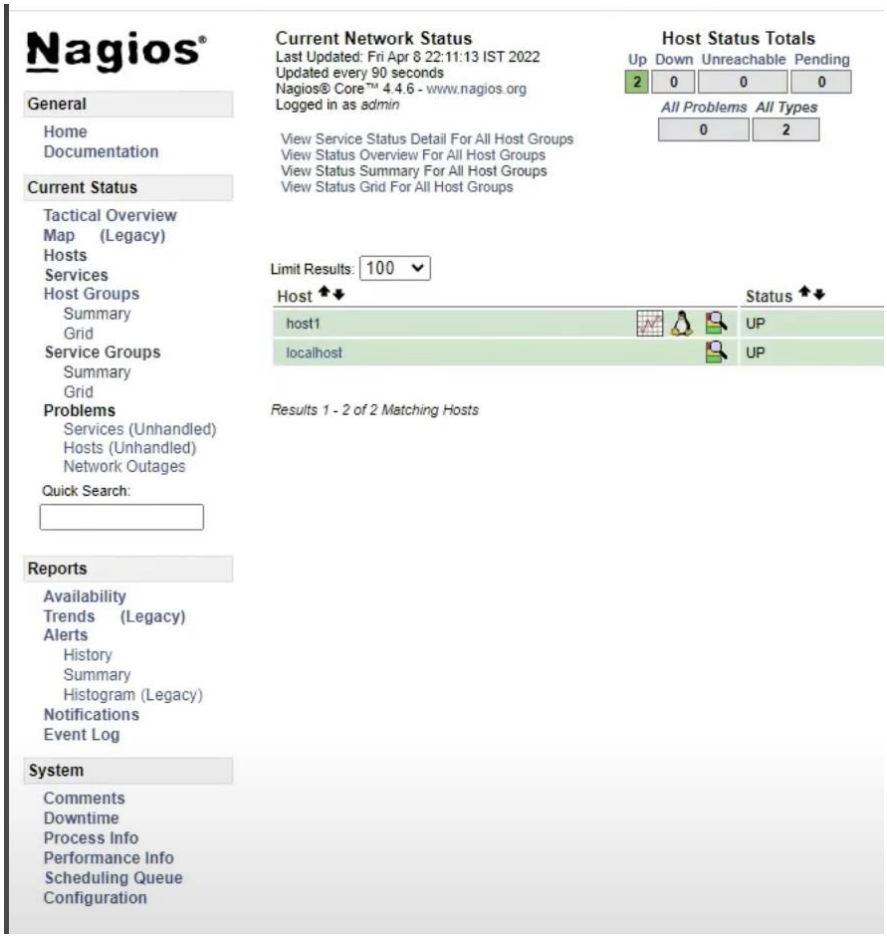


Рисунок 3.8 – Веб-консоль моніторингу конфігурації та активності серверу[33]

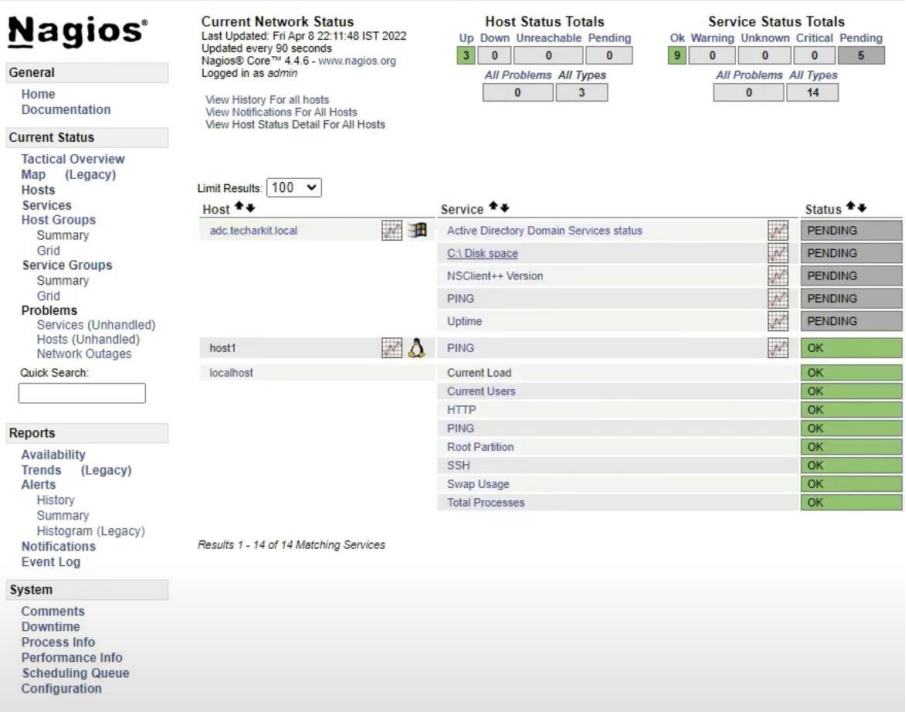


Рисунок 3.9 – Продовження веб-консоль моніторингу конфігурації та активності серверу[33]

Таким чином виконано завдання розробки процесу реалізації системи моніторінгу конфігурації серверного обладнання на базі інструменту NAGIOS. Розроблена методика тестування системи, виконано аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios. Для цього наведено зміст методики тестування системи моніторингу конфігурацій серверного обладнання на базі Nagios, опис виконання методики тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios. Запропонована процедура інтеграції Windows-системи в моніторингове середовище на базі Nagios. Для цього розглянуто зміст інтеграції Windows-системи в моніторингове середовище на базі Nagios та рекомендації щодо інтеграції Windows-системи в моніторингове середовище на базі Nagios.

ВИСНОВКИ

У сучасному світі IT-інфраструктури є важливою складовою для стабільного функціонування бізнесу, урядових установ та організацій різного масштабу. Складність IT-систем та серверного обладнання постійно зростає, що вимагає від інженерів більшого контролю за станом технічних ресурсів, їх налаштуванням та оновленням. Один із ключових аспектів ефективного управління IT-інфраструктурою – це наявність актуальних і достовірних даних про конфігурацію серверного обладнання.

Відповідно наукового завдання, а саме: оцінка доцільності та ефективності застосування удосконалення Nagios для моніторингу конфігурації серверного обладнання були виконані наступні завдання:

1. Проведен аналіз сучасних інструментів для моніторингу та управління в системі автоматизованого збору даних про конфігурації серверного обладнання.

2. Проведено дослідження програмної реалізації системи моніторингу конфігурації серверного обладнання на базі інструменту Nagios.

3. Розроблена методика тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios.

Це дозволяє застосувати сучасні інструменти для моніторингу та управління конфігурацією серверного обладнання. Особливу увагу в дослідженні приділено їхнім можливостям у контексті автоматизованого збору даних і забезпечення безперебійної роботи систем. Для цього проведене порівняння, яке виявило, що Nagios є оптимальним рішенням завдяки своїй гнучкості, широкому набору плагінів і можливості інтеграції з різними платформами.

Розглянуто методику встановлення та налаштування Nagios для забезпечення моніторингу серверного обладнання. Зокрема, докладно описано процес підключення Windows- і Linux-систем, налаштування плагінів для відстеження ключових параметрів, а також конфігурацію сповіщень.

Ці дії дозволили забезпечити повноцінний контроль за станом систем і швидке реагування на можливі збої.

Розроблено комплексну методику тестування функціональності системи моніторингу. Результати тестування підтвердили, що Nagios здатний ефективно виконувати поставлені завдання, забезпечуючи моніторинг таких параметрів, як завантаження процесора, використання пам’яті та дискового простору, а також стан мережевих сервісів. Важливим етапом стала інтеграція системи в багатоплатформне середовище, що демонструє її універсальність і надійність.

Продемонстровано важливість налаштування політик сповіщень та автоматичного звітування. Ці функції дозволяють не лише своєчасно інформувати про можливі проблеми, а й забезпечують докладну документацію для аналізу продуктивності та планування подальших дій.

Реалізовано заходи для забезпечення безпеки та розмежування доступу до системи моніторингу, що є критично важливим для захисту даних і запобігання несанкціонованим діям.

ПЕРЕЛІК ПОСИЛАНЬ

1. В. Іванченко, “Системи моніторингу серверного обладнання: аналіз та особливості впровадження,” Науковий журнал "Інформаційні системи", 2020; doi:10.32782/2221-8688.2020.2.45.

2. Бурак О. М. Інструменти бізнес-аналітики в регіональному аналізі / О. М. Бурак // Економічний аналіз. - 2015. - Т. 19(1). - С. 29-35.

3. Грабарєв А. В. Застосування сучасних інформаційних систем бізнес-аналітики у менеджменті / А. В. Грабарєв, В. М. Домрачев // Вісник Східноєвропейського університету економіки і менеджменту. Серія : Економіка і менеджмент. - 2015. - № 1. - С. 139-145.

4. Офіційний сайт Zabbix [Electronic resource]. – URL: https://www.zabbix.com/ru/features (дата звернення: 10.10.2024).

5. Моніторинг Prometheus: варіанти використання, показники та найкращі практики [Electronic resource]. – URL: https://itedu.center/ua/blog/review/monitorynh-prometheus-varianty-vykorystannia-pokaznyky-ta-naikrashchi-praktyky/?srsltid=AfmBOoqfy-oZiA9nI8joulDoazOMtueV95WmSPFuH8bKxTFOJgB3vXBP (дата звернення: 15.10.2024).

6. Офіційний сайт Grafana [Electronic resource]. – URL: https://grafana.com/ (дата звернення: 25.10.2023).

7. Elastic Stack Monitoring Dashboard [Electronic resource]. – URL: https://elastic-content-share.eu/downloads/elastic-stack-monitoring-dashboard/ (дата звернення: 26.10.2024).

8. Ansible [Electronic resource]. – URL: https://kangaroot.net/solutions/ansible (дата звернення: 26.10.2024).

9. Simple Network Management Protocol [Electronic resource]. – URL: https://uk.wikipedia.org/wiki/SNMP (дата звернення: 30.10.2024).

10. Nagios – система моніторингу серверів, служб, мережевих протоколів, мережевої інфраструктури [Electronic resource]. – URL: https://idealsoft.com.ua/vendory/vendors-more/nagios/ (дата звернення: 20.10.2024).

11. О. Сидоренко, “Nagios: можливості застосування для корпоративних мереж,” Праці Одеського національного університету, 2019.

12. Л. Ткачук, “Моніторинг серверного обладнання у хмарних середовищах,” Наукові записки НУК, 2021; doi:10.33108/2221.2021.04.50.

13. Nagios Log Server [Electronic resource]. – URL: https://www.nagios.com/products/nagios-log-server/ (дата звернення: 30.10.2024).

14. R. Amin et al., “A Framework for Server Monitoring Using Nagios,” Int’l J. of Computer Science and Information Technologies, 2020.

15. M. Dworak et al., “Linux Server Monitoring Using Open Source Tools,” Proc. IEEE Int’l Conf. on Information Systems, 2019; doi:10.1109/ICIS.2019.8955487.

16. J. Smith, “Cloud-Based Monitoring Solutions for Enterprise Systems,” Computer Science Review, 2021.

17. H. Anderson, “Integrating Nagios for Cross-Platform Server Monitoring,” Int’l J. of Advanced Networking and Applications, 2022.

18. Базове налаштування Nagios [Electronic resource]. – URL: https://blog.g3.1oodhoster.net/uk/bazovaya-nastrojka-nagios/ (дата звернення: 05.11.2024).

19. E. White et al., “Scalable Server Monitoring Using Open-Source Tools,” Proc. 11th Int’l Workshop on Scalable Computing, 2021; doi:10.1109/SC2021.8896589.

20. F. Taylor, “A Study of Monitoring Frameworks for Linux Servers,” IEEE Transactions on Cloud Computing, vol. 12, no. 1, 2020, pp. 30–45.

21. J. Adams, “Real-Time Alerts in Server Monitoring Systems,” Int’l J. of Systems Management, 2021.

22. T. Brown, “Using SNMP and Nagios for Enhanced Network Monitoring,” Proc. IEEE Networking Conf., 2020; doi:10.1109/NWC.2020.9123456.

23. M. Green, “Comprehensive Tools for Monitoring Server Infrastructure,”

European J. of Information Technology, 2021.

24. L. Fernandez, “Optimizing Open-Source Monitoring Solutions,” Proc. ACM Int’l Conf. on Cloud Computing, 2020; doi:10.1145/3341228.3343433.

25. Johnson, “Best Practices for Implementing Server Monitoring with Nagios,” Proc. IEEE Conf. on IT Systems, 2019; doi:10.1109/ITS.2019.8876541.

26. Wilson, “Server Monitoring in Multi-Platform Environments,” Int’l J. of Systems Engineering, 2021.

27. K. Schmidt, “Monitoring Tools for Hybrid Server Systems,” European Conf. on Systems Management, 2022.

28. P. Anderson et al., “Effective Data Collection Using Open-Source Monitoring Tools,” ACM Transactions on Network Management, 2021.

29. L. Rosen, “Configuring Nagios for Secure Server Monitoring,” Proc. IEEE Security Conf., 2020; doi:10.1109/SEC.2020.8903458.

30. Making the Switch to a Better Monitoring Solution [Electronic resource]. – URL: https://www.nagios.org/switch/ (дата звернення: 25.11.2024).

31. T. Huang et al., “Cross-Platform Monitoring Using Open-Source Tools,” Proc. Int’l Conf. on Advanced Computing, 2021; doi:10.1109/AC.2021.8884763.

32. Extending / Download NSClient++ (nscp) [Electronic resource]. – URL: https://github.com/mickem/nscp/releases/tag/0.5.3.3 (дата звернення: 03.11.2024).

33. How to configure a firewall on Linux with firewalld [Electronic resource]. – URL: https://www.redhat.com/en/blog/firewalld-linux-firewall (дата звернення: 03.11.2024).

34. SELinux Configuration [Electronic resource]. – URL: https://wiki.centos.org/HowTos(2f)SELinux.html (дата звернення: 04.11.2024).

35. Nagios Core Quickstart Installation Guides [Electronic resource]. – URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart.html (дата звернення: 04.11.2024).

36. System Management Commands [Electronic resource]. – URL: https://www.linux.org/docs/man8/useradd.html (дата звернення: 04.11.2024).

37. GNU Autoconf (version 2.72) [Electronic resource]. – URL:

https://www.gnu.org/savannah-checkouts/gnu/autoconf/manual/autoconf-2.72/autoconf.html (дата звернення: 04.11.2024).

38. GNU tar: an archiver tool [Electronic resource]. – URL: https://www.gnu.org/software/tar/manual/tar.html (дата звернення: 04.11.2024).

39. GNU make utility [Electronic resource]. – URL: https://www.gnu.org/software/make/manual/make.html (дата звернення: 04.11.2024).

40. Nagios Core Event handlers [Electronic resource]. – URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/eventhandlers.html (дата звернення: 04.11.2024).

41. How to allow a user in Nagios [Electronic resource]. – URL: https://serverfault.com/questions/111030/how-to-allow-a-user-in-nagios-to-view-the-status-of-some-servers-but-not-disabl (дата звернення: 04.11.2024).

42. systemctl — Control the systemd system and service manager [Electronic resource]. – URL: https://www.freedesktop.org/software/systemd/man/latest/systemctl.html (дата звернення: 04.11.2024).

43. The official Nagios Plugins are distributed with the Nagios Core Service Platform (CSP) [Electronic resource]. – URL: https://nagios-plugins.org/ (дата звернення: 04.11.2024).

44. Nagios Log Server - Manual Installation Instructions [Electronic resource]. – URL: https://support.nagios.com/kb/article/nagios-log-server-manual-installation-instructions-101.html (дата звернення: 04.11.2024).

45. Nagios Core Monitoring Windows Machines [Electronic resource]. – URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-windows.html (дата звернення: 04.11.2024).

46. How to configure Nagios in Linux [Electronic resource]. – URL: https://gurukgyan.blogspot.com/2017/11/q-how-to-configure-nagios-in-linux_13.html (дата звернення: 27.10.2024).

Додаток А

Основні кроки встановлення Nagios Core 4.3.4 і Nagios Plugin 2.2.1 на сервер з ОС Linux (наприклад, Ubuntu чи CentOS)

1. Попередня підготовка системи

- Оновіть систему:

```
bash
sudo apt update && sudo apt upgrade -y   # Для Ubuntu/Debian
sudo yum update -y                       # Для CentOS/RHEL
```

- Встановіть необхідні пакети:

```
bash
sudo apt install -y build-essential apache2 php gcc libc6
libgd-dev libmcrypt-dev make wget unzip # Ubuntu/Debian
sudo yum install -y gcc glibc glibc-common wget unzip httpd
php gd gd-devel perl make net-snmp # CentOS/RHEL
```

- Додайте користувача Nagios:

```
bash
sudo useradd nagios
sudo groupadd nagcmd
sudo usermod -aG nagcmd nagios
sudo usermod -aG nagcmd www-data         # Ubuntu/Debian
sudo usermod -aG nagcmd apache           # CentOS/RHEL
```

2. Завантаження вихідного коду

- Завантажте Nagios Core 4.3.4:

```
bash
wget
https://assets.nagios.com/downloads/nagioscore/releases/nagio
s-4.3.4.tar.gz
tar -xvf nagios-4.3.4.tar.gz
cd nagios-4.3.4
```

- Завантажте Nagios Plugins 2.2.1:

```
bash
wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz
tar -xvf nagios-plugins-2.2.1.tar.gz
```

3. Встановлення Nagios Core

- Скомпілюйте та встановіть Nagios Core:

```
bash
./configure --with-command-group=nagcmd
make all
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
sudo make install-webconf
```

- Встановіть Apache веб-сервер і налаштуйте доступ:

```
bash
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
systemctl restart apache2                # Ubuntu/Debian
systemctl restart httpd                  # CentOS/RHEL
```

- Запустіть Nagios:

```
bash
sudo systemctl start nagios
sudo systemctl enable nagios
```

4. Встановлення Nagios Plugins

- Перейдіть до каталогу з плагінами та скомпілюйте:

```
bash
cd ../nagios-plugins-2.2.1
./configure --with-nagios-user=nagios --with-nagios-group=nagcmd
```

```
make
sudo make install
```

- Перевірте плагіни:

  Плагіни встановлюються в каталог /usr/local/nagios/libexec.

  Використовуйте їх для моніторингу різних метрик.

5. Перевірка системи

- Перевірте конфігурацію Nagios:

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

- Доступ до веб-інтерфейсу Nagios:

```
1. Відкрийте браузер і перейдіть за адресою:
http://<ваша_IP_адреса>/nagios
2. Увійдіть за допомогою користувача nagiosadmin і пароля,
створеного вище.
```

6. Налаштування моніторингу

- Додайте нові хости, служби та параметри моніторингу, редагуючи файли конфігурації в каталозі:

  `/usr/local/nagios/etc/objects/.`

- Перезапустіть Nagios після внесення змін:

```
bash
sudo systemctl restart nagios
```

7. Тестування

- Упевніться, що всі хости та сервіси відображаються на веб-інтерфейсі.
- Використовуйте вбудовані плагіни, такі як `check_ping`, `check_http`, для перевірки стану систем.

Примітка: Ця інструкція базується на стандартних умовах. Якщо у вас виникають проблеми або ви використовуєте специфічну ОС/середовище, уточніть це для додаткової допомоги.

Додаток Б

Порядок підготовки Nagios до роботи

1. Встановлення необхідних залежностей.

Перед встановленням Nagios 4.3.4 необхідно встановити Apache, PHP і деякі бібліотеки, такі як gcc, glibc, glibc-common, GD-бібліотеки та бібліотеки розробника. Для цього можемо використовувати установник пакетів за замовчуванням yum:

```
[root@tecmint]# yum install -y httpd httpd-tools php gcc
glibc glibc-common gd gd-devel make net-snmp
-------------- On Fedora 22+ Onwards --------------
[root@tecmint]# dnf install -y httpd httpd-tools php gcc
glibc glibc-common gd gd-devel make net-snmp
```

2. Створення користувача і групи Nagios на RHEL.

Створимо новий обліковий запис користувача nagios і nagcmd і встановіть пароль:

```
[root@tecmint]# useradd nagios
[root@tecmint]# groupadd nagcmd
```

Потім додамо користувача nagios і користувача apache до групи nagcmd:

```
[root@tecmint]# usermod -G nagcmd nagios
[root@tecmint]# usermod -G nagcmd apache
```

3. Завантажимо Nagios Core 4.3.4 і Nagios Plugin 2.2.1.

Створимо каталог для встановлення Nagios і всіх його майбутніх завантажень:

```
[root@tecmint]# mkdir /root/nagios
[root@tecmint]# cd /root/nagios
```

Тепер завантажимо останні плагіни Nagios Core 4.3.4 і Nagios 2.2.1 за допомогою команди wget:

```
[root@tecmint nagios~]# wget
https://assets.nagios.com/downloads/nagioscore/releases/nagio
s-4.3.4.tar.gz
[root@tecmint nagios~]# wget https://nagios-
plugins.org/download/nagios-plugins-2.2.1.tar.gz
```

Витягнемо Nagios Core і його плагіни.

Потрібно витягти завантажений пакет за допомогою команди tar, як показано нижче:

```
[root@tecmint nagios~]# tar -xvf nagios-4.3.4.tar.gz
[root@tecmint nagios~]# tar -xvf nagios-plugins-2.2.1.tar.gz
```

Якщо витягти ці tarball, у цьому каталозі з'являться дві нові папки:

```
[root@tecmint nagios ~]# ls -l
total 13520
drwxrwxr-x 18 root root 4096 Aug 24 17:43 nagios-4.3.4
-rw-r--r-- 1 root root 11101966 Aug 24 17:48 nagios-
4.3.4.tar.gz
drwxr-xr-x 15 root root 4096 Apr 19 12:04 nagios-plugins-
2.2.1
-rw-r--r-- 1 root root 2728818 Apr 19 12:04 nagios-plugins-
2.2.1.tar.gz
```

4. Налаштування Nagios Core.

Виконаємо конфігурування Nagios Core. Для цього потрібно перейти до каталогу Nagios на RHEL і запустити файл конфігурації, і якщо все виконано правильно, він покаже вам такий результат, як у прикладі виводу, наведеному нижче.

```
[root@tecmint nagios~]# cd nagios-4.3.4/
[root@tecmint nagios-4.3.4 ]# ./configure --with-command-
group=nagcmd
Creating sample config files in sample-config/ ...
*** Configuration summary for nagios 4.3.4 2017-08-24 ***:
General Options:
-------------------------
```

```
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: /run/nagios.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory: /etc/rc.d/init.d
Apache conf.d directory: /etc/httpd/conf.d
Mail program: /usr/bin/mail
Host OS: linux-gnu
IOBroker Method: epoll
Web Interface Options:
------------------------
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/bin/traceroute
Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.
```

Після налаштування необхідно скомпілювати і встановити всі бінарні файли за допомогою make all і make install, ці команди встановлять всі необхідні бібліотеки на комп'ютері, що дасть змогу продовжити роботу:

```
[root@tecmint nagios-4.3.4 ]# make all
[root@tecmint nagios-4.3.4 ]# make install
```

Наступна команда встановить скрипти init для Nagios:

```
[root@tecmint nagios-4.3.4 ]# make install-init
```

Щоб змусити nagios працювати прямо з командного рядка, потрібно встановити command-mode:

```
[root@tecmint nagios-4.3.4 ]# make install-commandmode
```

Потім, запустимо наступну команду, для встановлення зразків файлів nagios:

```
[bahs][root@tecmint nagios-4.3.4 ]# make install-config[/bash].
```

```
    /usr/bin/install -c -m 775 -o nagios -g nagios -d
    /usr/local/nagios/etc
    /usr/bin/install -c -m 775 -o nagios -g nagios -d
    /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
    /usr/bin/install -c -b -m 660 -o nagios -g nagios sample-
    config/resource.cfg /usr/local/nagios/etc/resource.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/templates.cfg
    /usr/local/nagios/etc/objects/templates.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/commands.cfg
    /usr/local/nagios/etc/objects/commands.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/contacts.cfg
    /usr/local/nagios/etc/objects/contacts.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/timeperiods.cfg
    /usr/local/nagios/etc/objects/timeperiods.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/localhost.cfg
    /usr/local/nagios/etc/objects/localhost.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/windows.cfg
    /usr/local/nagios/etc/objects/windows.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/printer.cfg
    /usr/local/nagios/etc/objects/printer.cfg
    /usr/bin/install -c -b -m 664 -o nagios -g nagios sample-
    config/template-object/switch.cfg
    /usr/local/nagios/etc/objects/switch.cfg
```

- **Налаштування конфігурації Nagios на RHEL.**

Відкрийте файл «contacts.cfg», використовуючи будь-який ваш улюблений редактор, і встановіть адресу електронної пошти, пов'язану з контактом nagiosadmin, для отримання повідомлень електронною поштою:

```
# vi /usr/local/nagios/etc/objects/contacts.cfg
```

- **Встановлення та налаштування веб-інтерфейсу для Nagios.**

Вся конфігурація в бекенді виконана, тепер необхідно налаштувати веб-інтерфейс для Nagios на RHEL. Наступна команда налаштує веб-інтерфейс для Nagios, і буде створено користувача веб-адміністратор («nagiosadmin»):

```
[root@tecmint nagios-4.3.4 ]# make install-webconf
```

На цьому етапі створимо пароль для «nagiosadmin». Після виконання цієї команди двічі вкажіть пароль і збережемо його, тому що цей пароль буде використовуватися під час входу в веб-інтерфейс Nagios:

```
[root@tecmint nagios-4.3.4]# htpasswd -s -c
/usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

Перезапустимо Apache, щоб нові налаштування набули чинності.

```
[root@tecmint ]# service httpd start [On RHEL/CentOS 6/5 and
Fedora]
[root@tecmint ]# systemctl start httpd.service [On
RHEL/CentOS 7 and Fedora 19 Onwards]
```

- **Компілювання та встановлення плагіну Nagios.**

Плагіни nagios завантажено в /root/nagios, перейдіть туди і виконайте налаштування, що вказані нижче:

```
[root@tecmint nagios]# cd /root/nagios
[root@tecmint nagios]# cd nagios-plugins-2.2.1/
[root@tecmint nagios]# ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
[root@tecmint nagios]# make
[root@tecmint nagios]# make install
```

- **Перевірка файлів конфігурації Nagios.**

Тепер вся необхідна конфігурація Nagios виконана. І настав час перевірити його роботу.

```
[root@tecmint nagios]# /usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
Nagios Core 4.3.4
Copyright (c) 2009-present Nagios Core Development Team and
Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2017-08-24
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
```

Додавання служби Nagios до запуску системи.

Щоб змусити Nagios працювати після перезавантаження, потрібно додати nagios і httpd до запуску системи за допомогою команд chkconfig і systemctl.

```
[root@tecmint ]# chkconfig --add nagios
[root@tecmint ]# chkconfig --level 35 nagios on
[root@tecmint ]# chkconfig --add httpd
[root@tecmint ]# chkconfig --level 35 httpd on
```

Перезапустимо Nagios, щоб нові налаштування набули чинності.

Nagios готовий до роботи, відкриємо його у своєму браузері за допомогою «http://192.168.0.15 /nagios».

Додаток В

Приклад додавання сервера у файл hosts.cfg

У цьому прикладі додаємо сервер бази даних з IP-адресою 192.168.1.10:

```
define host {
    use             linux-server
    host_name       database_server
    alias           Main Database Server
    address         192.168.1.10
    max_check_attempts 5
    check_period    24x7
    notification_interval 30
    notification_period   24x7
}
```

– use: вказує на тип хоста (у цьому випадку linux-server — шаблон, який містить базові налаштування для серверів на Linux).

– host_name: унікальне ім’я хоста.

– alias: зрозуміле ім’я хоста для ідентифікації.

– address: IP-адреса хоста.

– max_check_attempts: максимальна кількість спроб перед тим, як буде вказано, що хост не відповідає.

– check_period: період перевірок (наприклад, 24x7 для постійного моніторингу). notification_interval: інтервал повторних сповіщень у хвилинах.

– notification_period: період часу, коли надсилатимуться сповіщення (також 24x7).

Додаток Г

Приклад конфігурації служби для моніторингу доступності бази даних (порт 3306)

Додаємо службу для контролю порту 3306 (типовий порт MySQL) у файл services.cfg:

```
define service {
    use                     generic-service
    host_name               database_server
    service_description     MySQL Port Check
    check_command           check_tcp!3306
    max_check_attempts      5
    normal_check_interval   1
    retry_check_interval    1
    notification_interval   60
    notification_period     24x7
    notification_options    w,u,c,r
}
```

– use: тип служби (у цьому випадку generic-service, який містить загальні параметри для служб).

– host_name: назва хоста, до якого належить ця служба (database_server). service_description: опис служби (у нашому прикладі це перевірка доступності порту MySQL).

– check_command: команда для перевірки (у цьому випадку check_tcp з параметром порту 3306).

– max_check_attempts: кількість спроб перевірки перед відміткою "Критичний". normal_check_interval: інтервал звичайної перевірки у хвилинах. retry_check_interval: інтервал повторної перевірки після виявлення проблеми. notification_interval: інтервал повторних сповіщень у хвилинах.

– notification_period: період часу для сповіщень (24x7).

– notification_options: види сповіщень (w — Warning, u — Unknown, c — Critical, r — Recovery).

Додаток Ґ

Приклади налаштування Nagios

Налаштування груп серверів. Для великих мереж рекомендується використовувати групи хостів, щоб зручно керувати серверними групами (наприклад, веб-сервери, сервери баз даних, поштові сервери). Приклад налаштування групи хостів у Nagios:

```
define hostgroup {
    hostgroup_name  web_servers
    alias           Web Servers Group
    members         web_server_1, web_server_2, web_server_3
}
```

У даному прикладі створено групу web_servers, що включає три сервери: web_server_1, web_server_2 та web_server_3. Тепер можна застосувати один набір правил моніторингу для всієї групи, що спрощує налаштування.

Встановлення порогів і налаштування сповіщень. Nagios дозволяє задавати пороги (наприклад, критичні значення використання ресурсів) для кожної служби. Ось приклад налаштування порогів для моніторингу використання дискового простору:

```
define service {
    use                     generic-service
    host_name               database_server
    service_description     Disk Space Check
    check_command           check_disk!20%!10%
    max_check_attempts      3
    normal_check_interval   5
    retry_check_interval    1
    notification_interval   120
    notification_period     24x7
}
```

У цьому прикладі:

`check_command check_disk!20%!10%` задає поріг для Warning, коли дисковий простір досягає 20%, і для Critical — 10%.

Перевірка та перезапуск Nagios. Після внесення змін до конфігураційних файлів необхідно перевірити конфігурацію та перезапустити Nagios:

Перевірка конфігурації

```
nagios -v /etc/nagios/nagios.cfg
```

Перезапуск служби

```
Nagios systemctl restart nagios
```

Налаштування сповіщень про проблеми з конфігурацією серверного обладнання. У Nagios можна налаштувати сповіщення через різні канали, залежно від доступності ресурсів та вимог компанії:

– Електронна пошта: Найпоширеніший спосіб, що дає можливість відправляти детальні повідомлення з описом проблеми. Налаштовується за допомогою команд у Nagios.

– SMS: Для критичних сповіщень, які потребують швидкої реакції. Потребує використання шлюзу SMS або інтеграції з сервісом сторонніх повідомлень, що можна налаштувати через спеціальні плагіни.

Приклад налаштування сповіщення через електронну пошту. Створення команди для сповіщення в конфігураційному файлі *commands.cfg*:

```
define command {
 command_name    notify-host-by-email
 command_line          /usr/bin/printf   "%b"   "Host   Alert:
$HOSTALIAS$ is $HOSTSTATE$\n" | /usr/bin/mail -s "Host Alert:
$HOSTALIAS$ is $HOSTSTATE$" admin@example.com
}
define command {
 command_name    notify-service-by-email
 command_line         /usr/bin/printf  "%b"  "Service  Alert:
$SERVICEDESC$   on   $HOSTALIAS$   is   $SERVICESTATE$\n"   |
/usr/bin/mail -s "Service Alert: $SERVICEDESC$ on $HOSTALIAS$
is $SERVICESTATE$" admin@example.com
```

```
}
```

Налаштування користувачів для сповіщення в contacts.cfg:

```
define contact {
    contact_name            admin
    use                             generic-contact
    alias                           Admin
    email                           admin@example.com
}
```

Прив’язка сповіщень до хостів та служб: Вказуємо, які хости та служби відправлятимуть повідомлення в разі збоїв.

Встановлення політик критичних і попереджувальних рівнів сповіщень Рівні сповіщень визначаються, щоб інформувати адміністраторів про проблеми на різних стадіях.

Встановлення політик попереджувальних (Warning) і критичних (Critical) сповіщень дозволяє оптимізувати реакцію на збої:

– Попереджувальний рівень (Warning): Вказує на потенційні проблеми (наприклад, 80% використання ЦП). Дає змогу адміністратору вжити заходів до настання критичної ситуації.

– Критичний рівень (Critical): Вказує на критичний стан (наприклад, 95% використання ЦП), який потребує негайної реакції.

– ***Приклад політики для моніторингу дискового простору.***

```
define service {
    use                     generic-service
    host_name               web_server
    service_description     Disk Space
    check_command           check_disk!20%!10%
    notification_options    w,u,c,r
}
```

– `check_command check_disk!20%!10%`: налаштовує пороги, де 20% — попереджувальний рівень, 10% — критичний рівень.

– `notification_options`: задає типи сповіщень (w — Warning, u — Unknown, c — Critical, r — Recovery).

Налаштування періодичного контролю конфігурації. Періодичний моніторинг у Nagios виконується шляхом налаштування інтервалів перевірок для різних хостів та служб. Це дозволяє регулярно отримувати дані про стан серверів і актуальність їх конфігурацій, наприклад, навантаження ЦП, дисковий простір, час роботи, доступність служб тощо [22]. Параметри періодичності задаються для кожного ресурсу відповідно до вимог.

```
define service {
    use                     generic-service
    host_name               app_server
    service_description     CPU Load
    check_command           check_cpu
    normal_check_interval   5
    retry_check_interval    1
}
```

- `normal_check_interval`: задає інтервал перевірки у хвилинах (у цьому випадку кожні 5 хвилин).
- `retry_check_interval`: інтервал повторної перевірки у разі проблеми (1 хвилина).

Завдяки цьому сервер буде періодично перевірятися на перевантаження процесора, що дозволяє вчасно виявити критичні зміни.

Додаток Д

Впровадження правил доступу та безпеки для адміністраторів і користувачів системи моніторингу

Nagios дозволяє створювати індивідуальні облікові записи для різних ролей користувачів, наприклад, адміністраторів, технічних фахівців та інших співробітників. Цей підхід дозволяє налаштувати права доступу залежно від обов'язків кожного користувача:

Адміністратори: отримують повний доступ до всіх функцій системи моніторингу, включаючи налаштування хостів, служб, політик сповіщення та керування обліковими записами.

Оператори: мають обмежений доступ для моніторингу певних хостів і служб, з можливістю перегляду, але без права змінювати налаштування системи.

Користувачі: можуть лише переглядати інформацію про стан серверів, які вони контролюють, без можливості редагування чи впливу на інші служби.

У конфігурації cgi.cfg можна задати різні рівні доступу для користувачів:

```
authorized_for_system_information=admin
authorized_for_configuration_information=admin,operator
authorized_for_all_services=admin,operator
authorized_for_all_hosts=admin,operator
authorized_for_all_service_commands=admin
authorized_for_all_host_commands=admin
```

Тут можна побачити, що:

admin має доступ до всіх команд і налаштувань.

operator може переглядати і керувати окремими службами та хостами без права змінювати загальну конфігурацію.

Для захисту даних, що передаються між Nagios і серверними пристроями, необхідно налаштувати шифрування каналів зв'язку за допомогою SSL/TLS. Це

забезпечує захищений зв'язок, зменшуючи ризик перехоплення даних під час їх передавання через мережу.

Налаштування шифрування SSL/TLS.

1. Встановлення SSL-сертифіката: Генерується SSL-сертифікат і ключ для сервера, на якому розгорнуто Nagios.

2. Конфігурація веб-сервера (наприклад, Apache) для використання HTTPS. У конфігурації Apache налаштовуються порти для SSL-з’єднання та шляхи до сертифіката і ключа.

Налаштування SSL для Apache:

```
<VirtualHost *:443>
    DocumentRoot "/usr/local/nagios/share"
    ServerName nagios.example.com
    SSLEngine on
    SSLCertificateFile /path/to/server.crt
    SSLCertificateKeyFile /path/to/server.key
</VirtualHost>
```

3. Переналаштування Nagios для роботи через HTTPS: Після налаштування веб-сервера користувачі отримають доступ до інтерфейсу Nagios через захищене HTTPS-з'єднання.

Аутентифікація користувачів Nagios підтримує базову аутентифікацію користувачів через веб-сервер або за допомогою LDAP, якщо потрібно інтегруватися з корпоративним каталогом користувачів. Це дозволяє централізовано керувати обліковими записами, спрощуючи доступ і захищаючи систему [26].

4. Створення облікових записів з паролями: Наприклад, у Apache для кожного користувача задається пароль.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users admin
```

5. Налаштування аутентифікації у веб-сервері: Додаємо вимогу авторизації в конфігурації Nagios.

```
<Directory "/usr/local/nagios/share">
    AuthType Basic
```

```
        AuthName "Nagios Access"
        AuthUserFile /usr/local/nagios/etc/htpasswd.users
        Require valid-user
    </Directory>
```

Переваги забезпечення безпеки та розмежування доступу:

Захист даних. Шифрування та аутентифікація знижують ризики несанкціонованого доступу.

Контроль доступу. Чітке розмежування прав дозволяє контролювати, хто має доступ до конфіденційної інформації та команд управління.

Гнучкість. Індивідуальні налаштування для різних рівнів користувачів дозволяють зберігати високу ефективність роботи, забезпечуючи водночас захист системи.

Додаток Е

Методика тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios

Крок 1. Увійдемо до середовища Oracle Linux 7 як користувач root Спершу потрібно виконати вхід до серверного середовища Oracle Linux 7 під користувачем із правами root. Це дозволить виконувати всі необхідні команди для налаштування системи.

Налаштуємо брандмауер для доступу до веб-інтерфейсу Nagios Щоб зробити веб-інтерфейс Nagios доступним із зовнішніх машин, виконаємо наступну команду в терміналі (Рис. Е1–Е2):

sudo firewall-cmd --permanent --add-port=80/tcp

Рисунок Е1 – Налаштування Firewall [33]

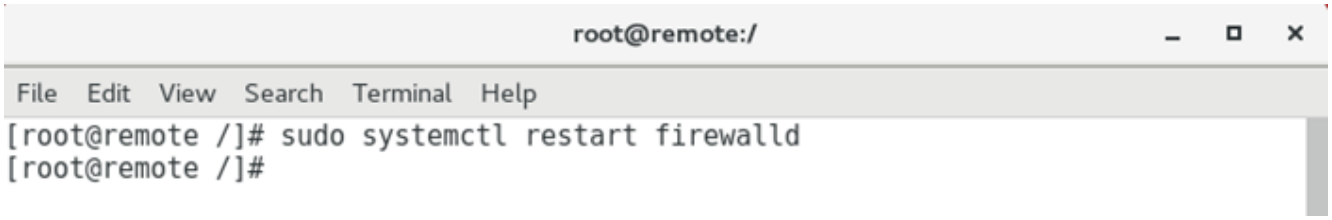


Рисунок Е2 – Перезапуск демона Firewall [33]

Крок 2. потрібно вимкнути ***SELINUX***, відредагувавши файл конфігурації, розташований за адресою: */etc/selinux/* і змінивши ***SELINUX*** із примусового на вимкнений (рис. Е3–Е4):

vi /etc/selinux/config

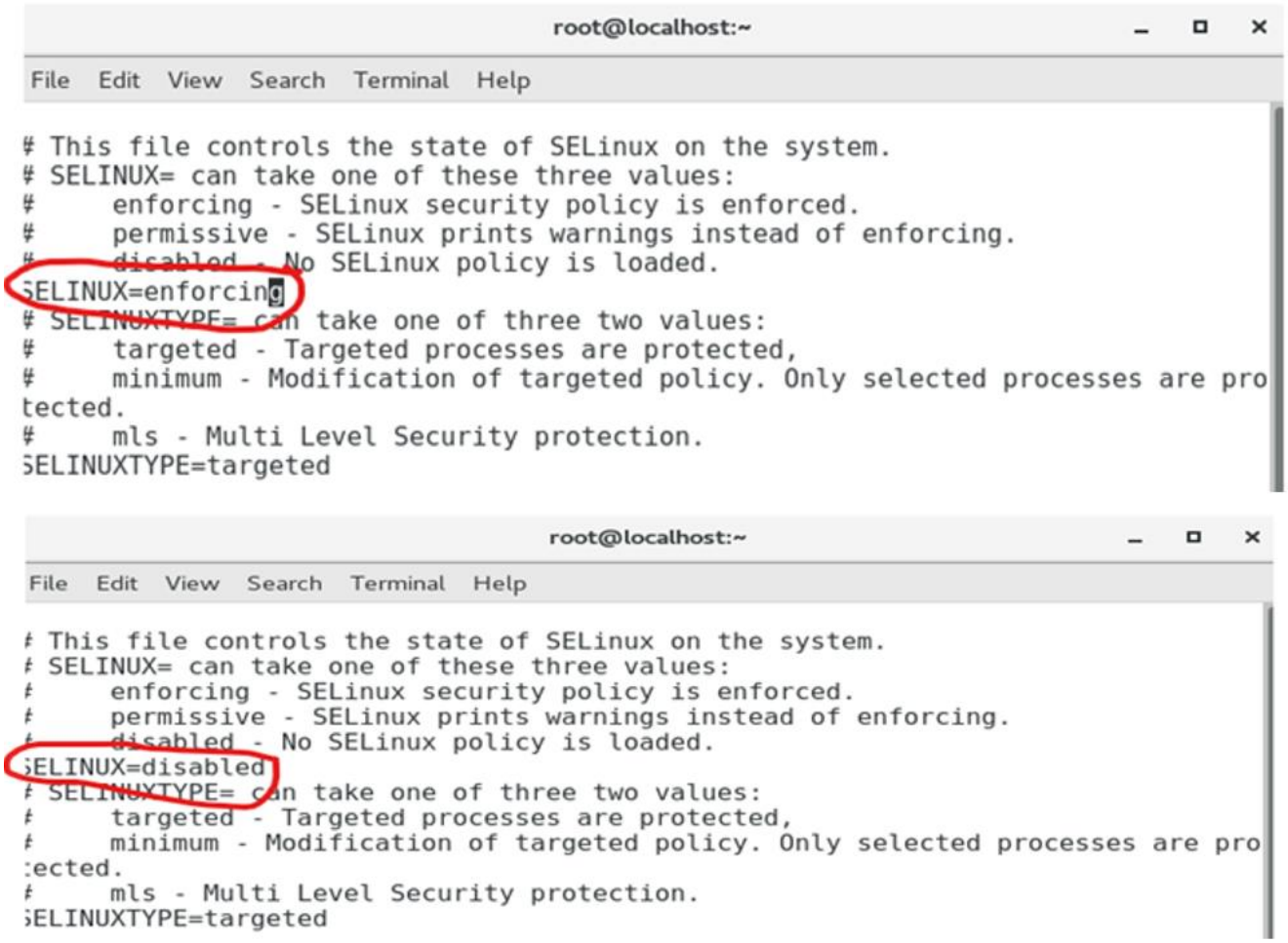


Рисунок Е3 – Вимкнення параметру SELINUX [34]

Крок 3. Потрібно встановити залежні пакети nagios.

yum install -y wget httpd php gcc glibc glibc-common gd gd-devel make net-snmp perl perl-devel openssl openssl-devel

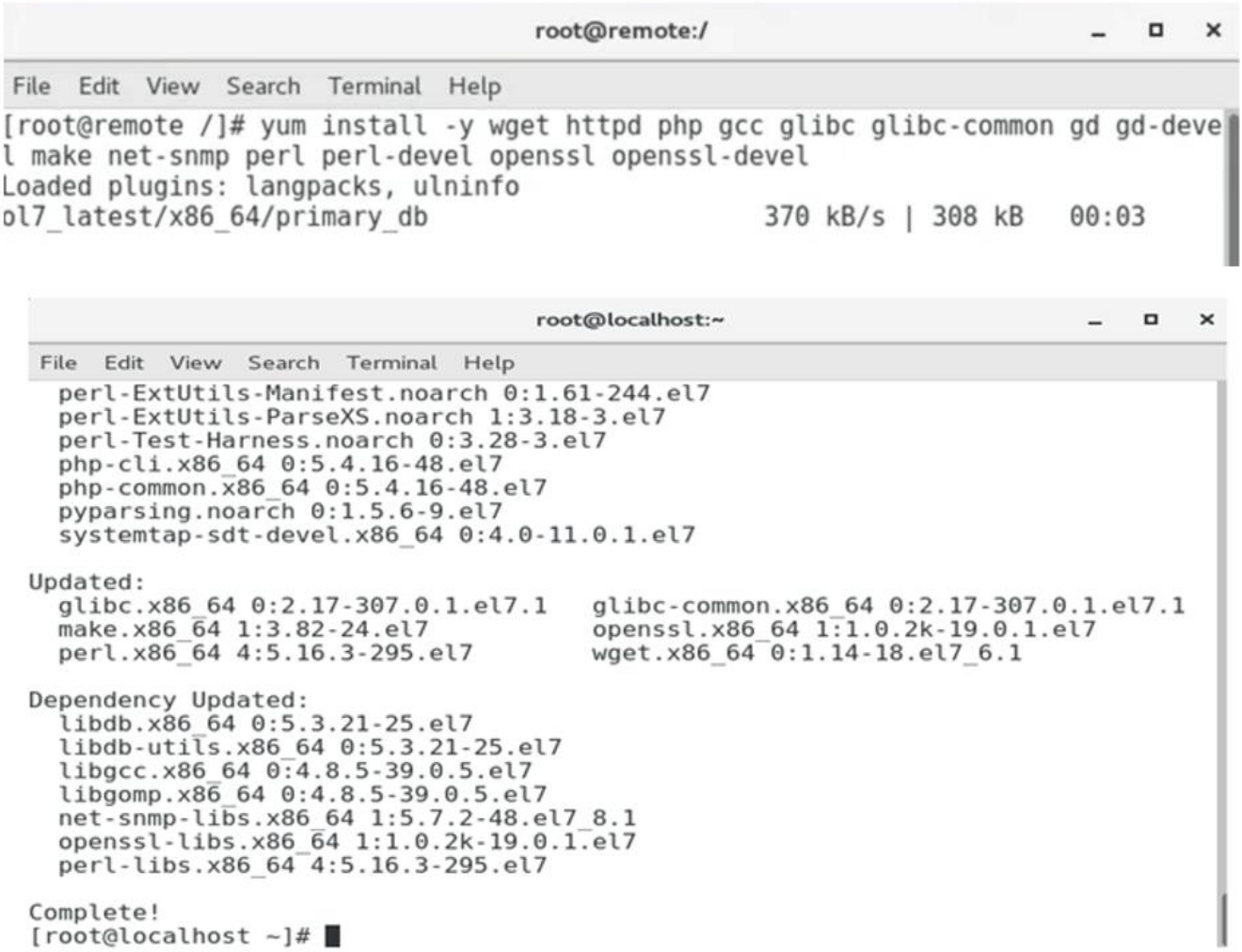


Рисунок Е4 – Завантаження залежностей з репозиторію [34, 35]

Крок 4. Завантажимо Nagios Core Source (рис. Е5).

wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.3.tar.gz

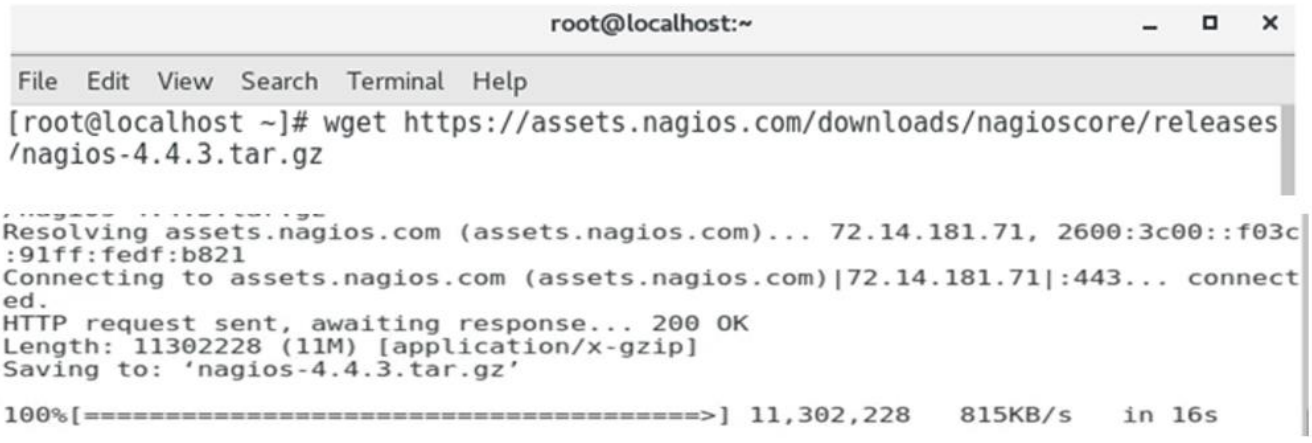


Рисунок Е5 – Завантаження Nagios Core Source [35]

Крок 5. Створимо користувачів Nagios і apache, створимо групу та додамо їх до цієї групи (рис. Е6).

useradd Nagios

groupadd naggrp

usermod -a -G naggrp Nagios

usermod -a -G naggrp apache

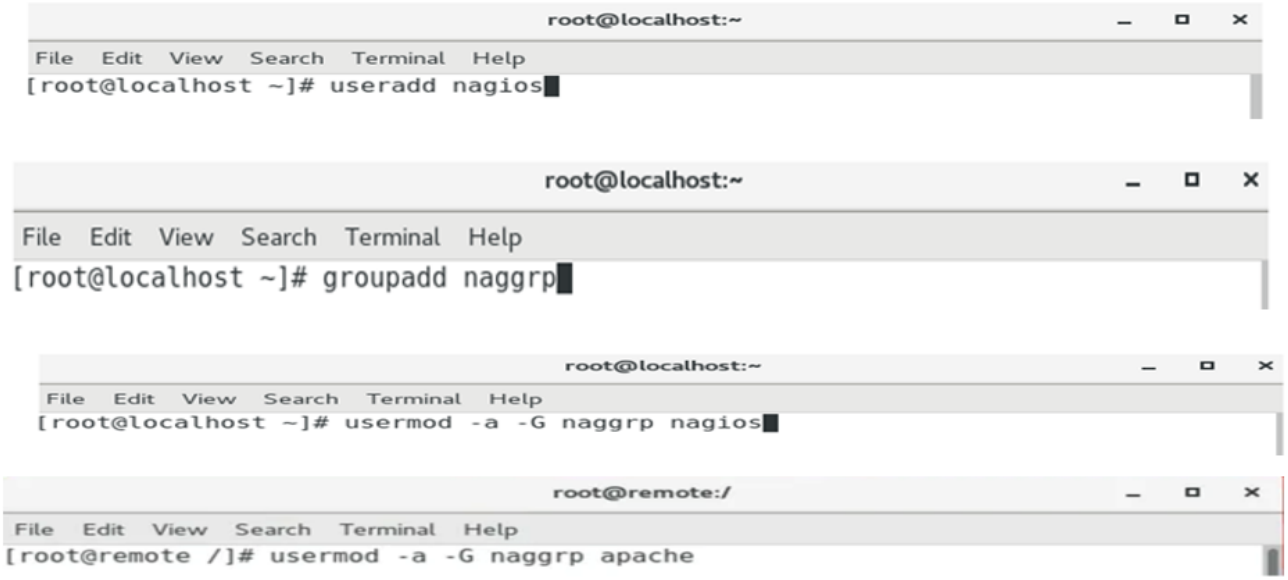


Рисунок Е6 – Створення груп для користувачів Nagios і Apache [36]

Крок 6. Розархівуйте файл (рис. Е7) завантаження з кроку 4, змініть каталог на папку, що містить завантажений файл, і виконайте цю команду:

tar -xvf nagios-4.4.3.tar.gz

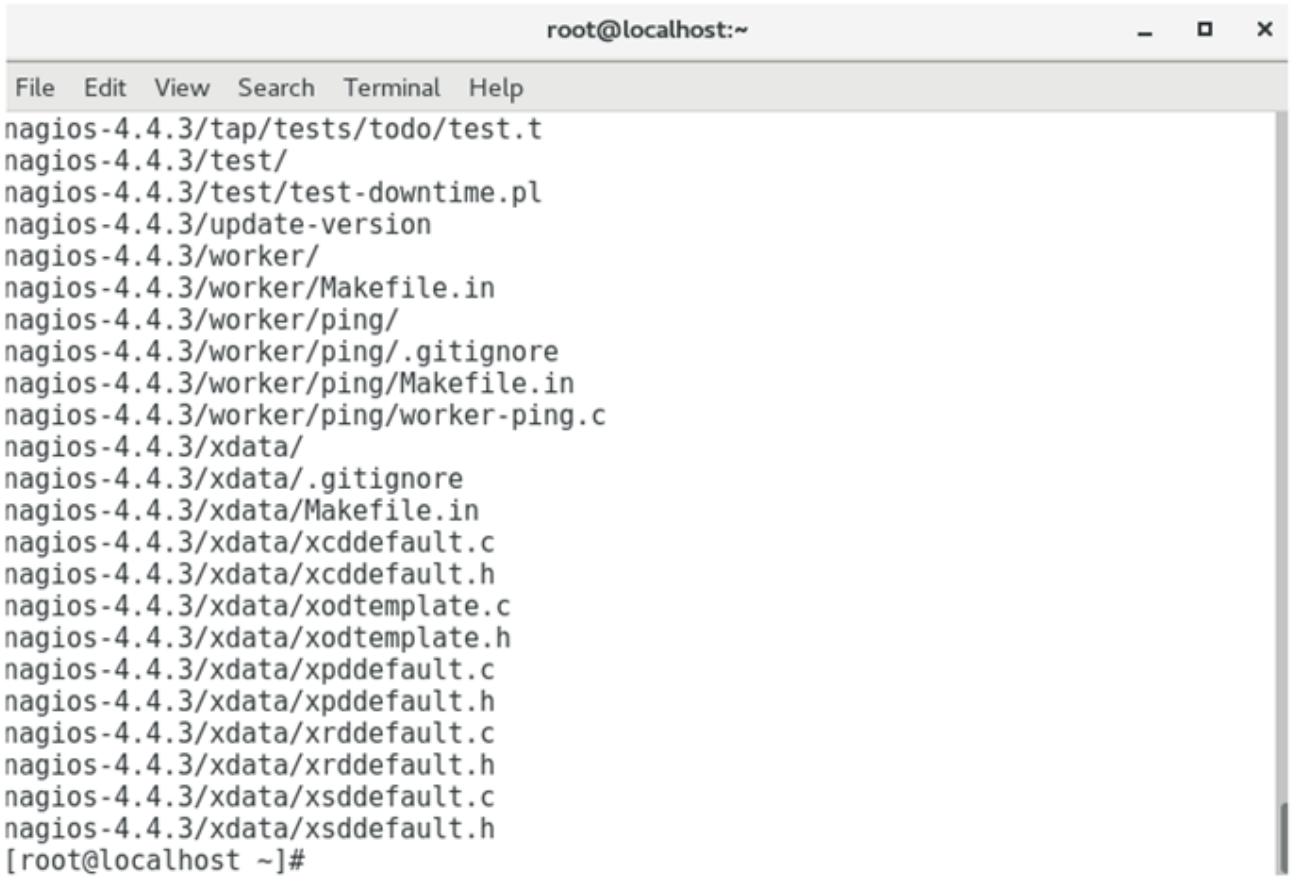


Рисунок Е7 – Розархівований файл (ключова версія nagios-4.4.3) [38]

Крок 7. Скомпілюйте Nagios із вихідного коду. Змініть каталог на папку, що містить розархівований файл із кроку 6 (рис. Е8–Е9).

cd nagios-4.4.3

./configure --with-command-group=naggrp

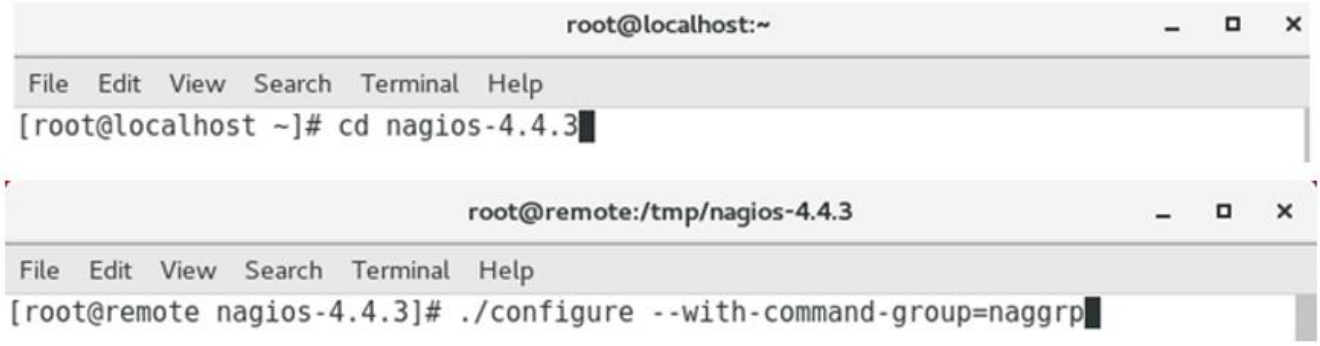


Рисунок Е8 – Конфігурація файлу в групу Nagios [37]

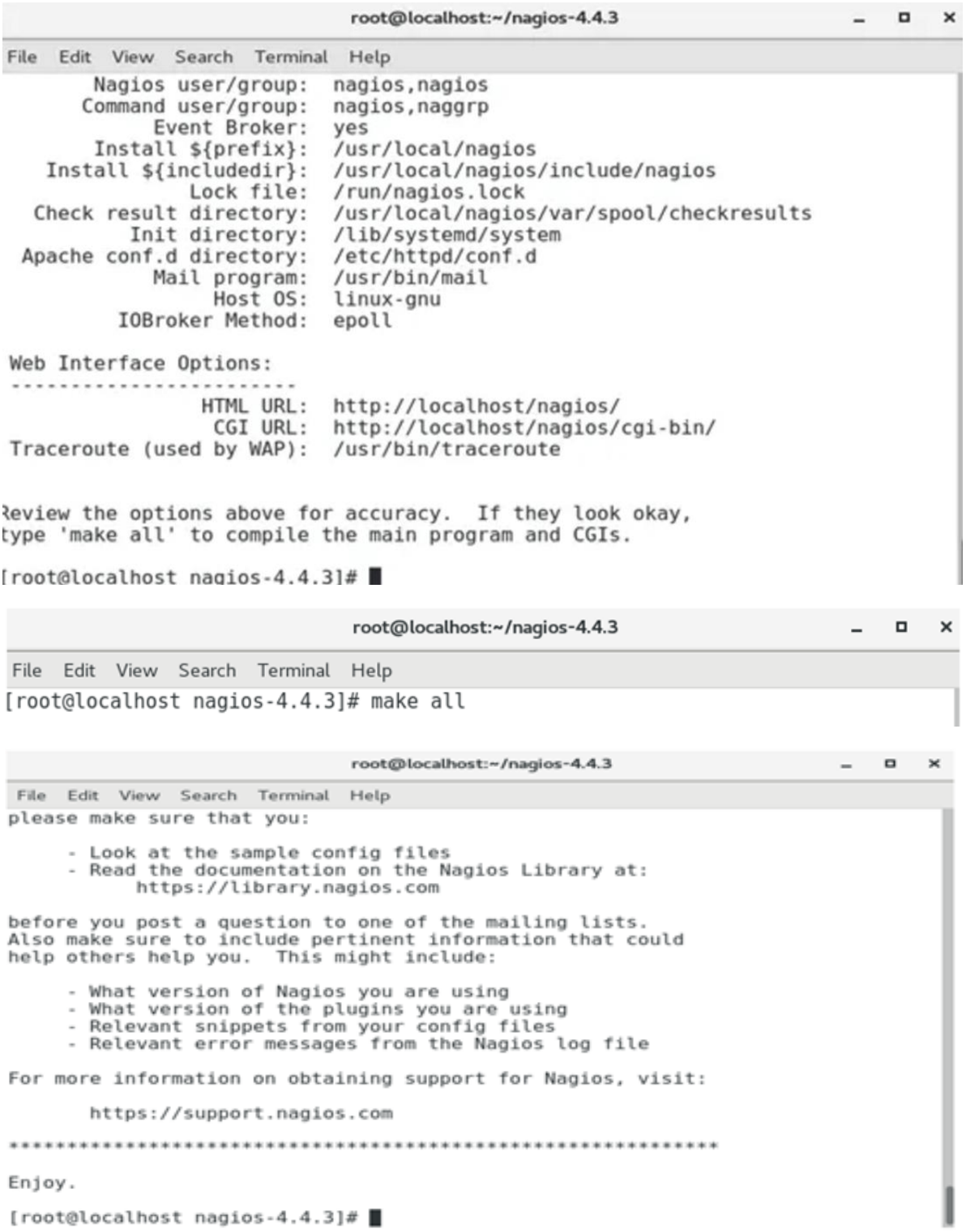


Рисунок Е9 – Продовження конфігурації файлу в групу Nagios [37]

Потрібно зробити інсталяцію (рис. Е10–Е15).

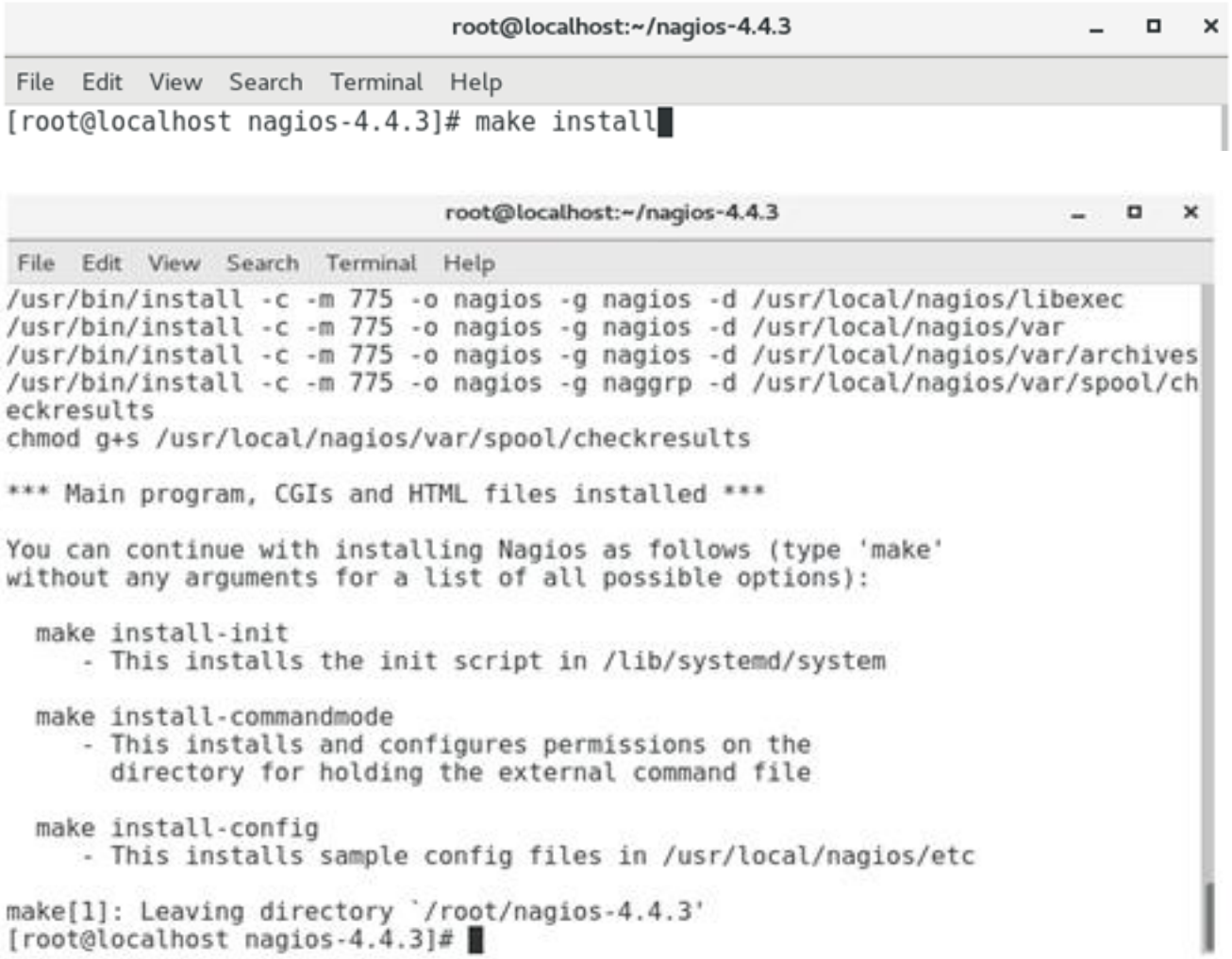


Рисунок Е10 – Make install [39]

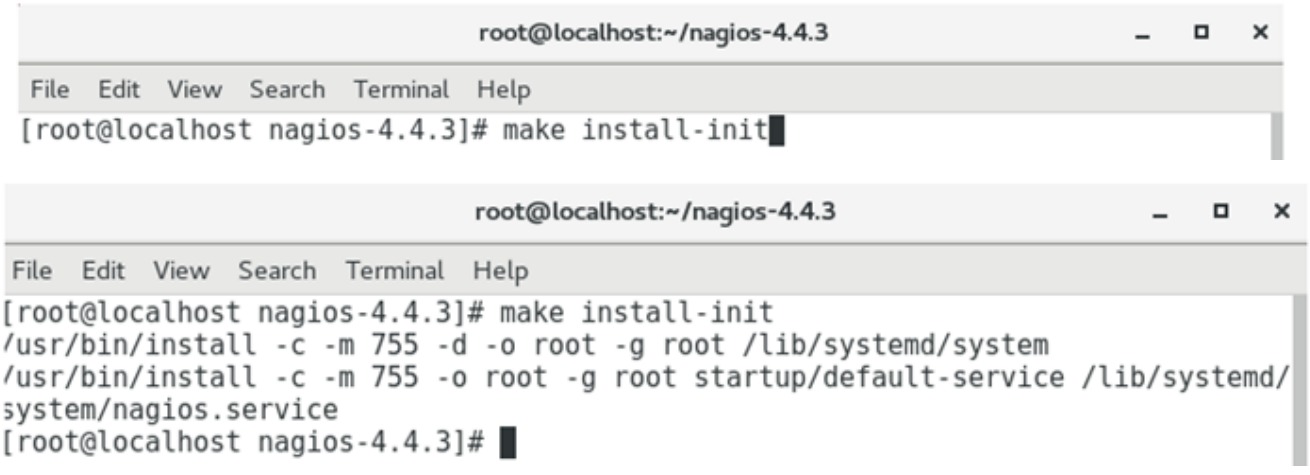


Рисунок Е11 – Make install-init [39]

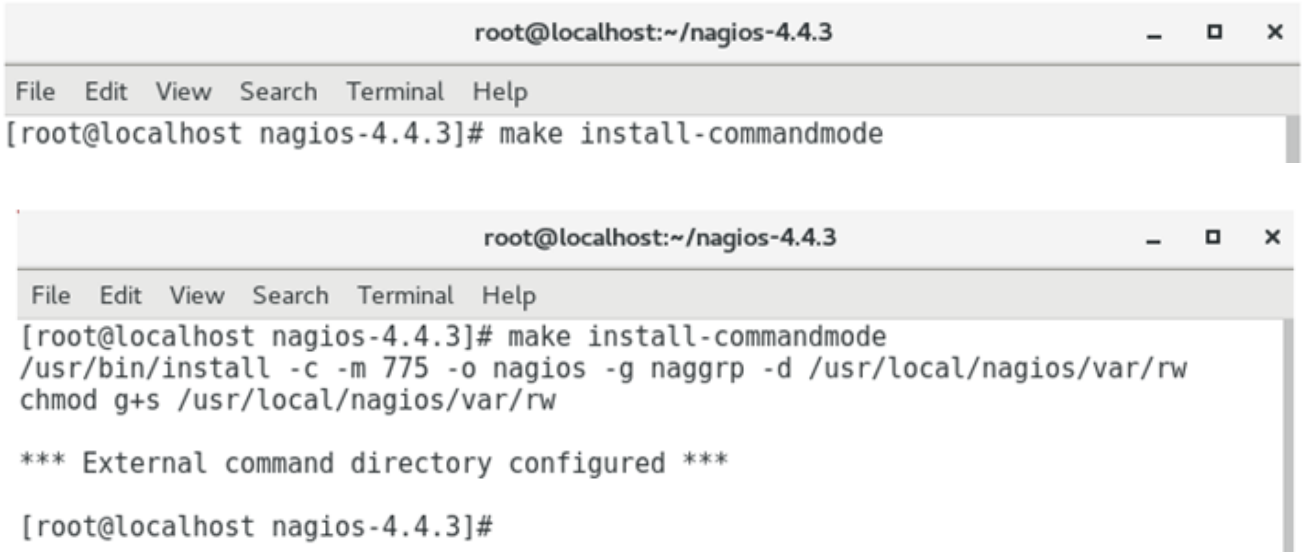


Рисунок Е12 – make install-commandmode [39]

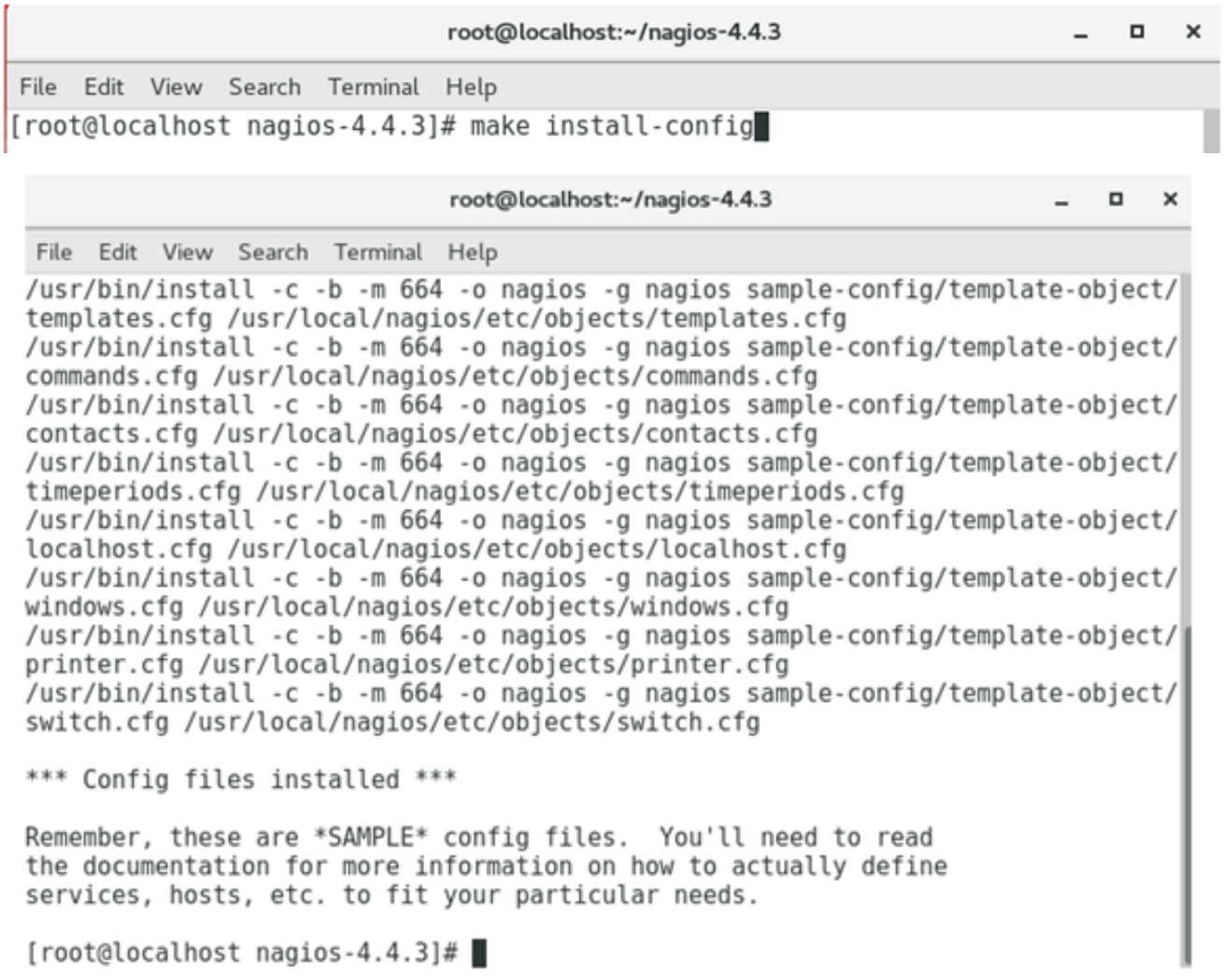


Рисунок Е13 – Make install-config [39]

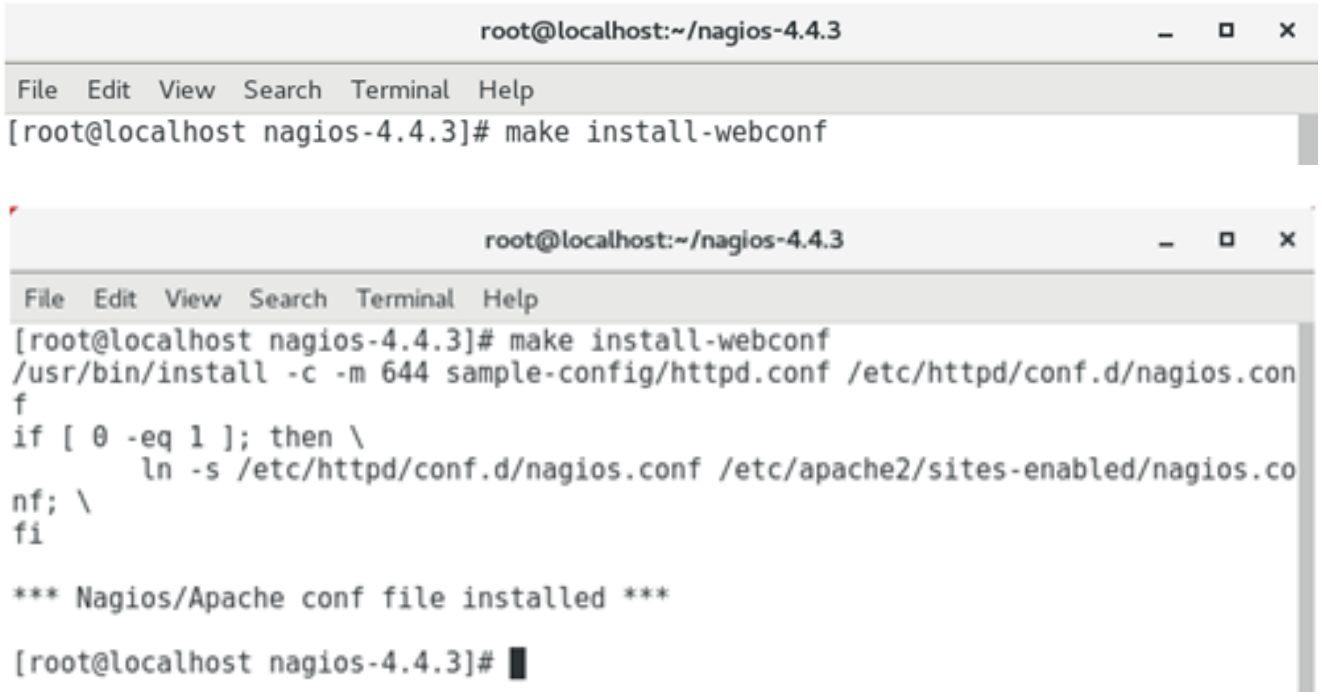


Рисунок Е14 – Make install-webconfig [39]

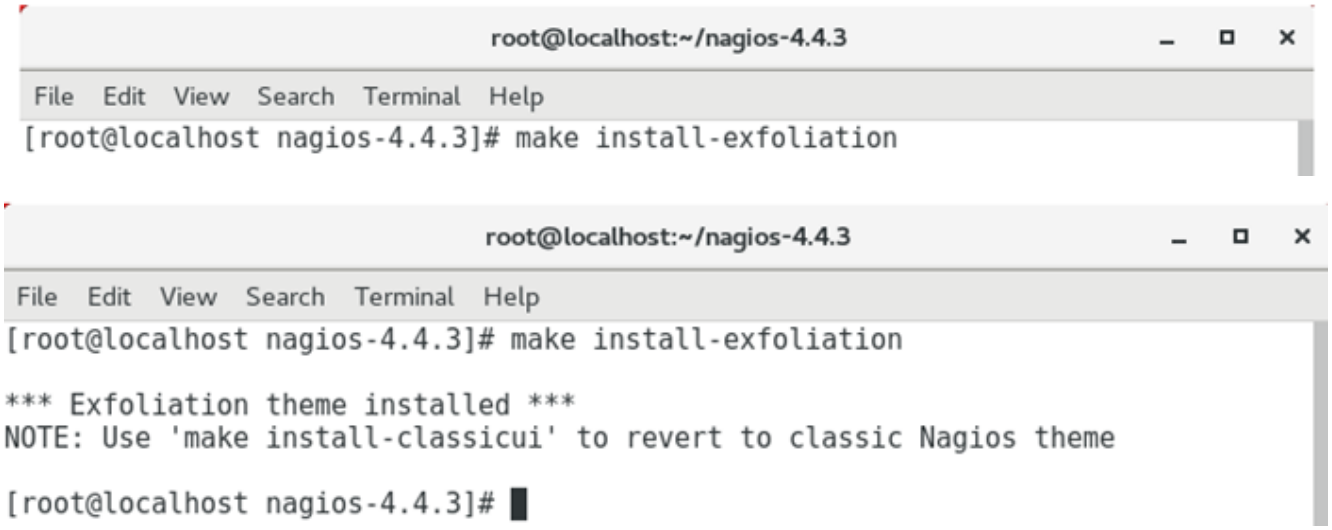


Рисунок Е15 – Make install-exfoliation [39]

Крок 8. все ще в розархівованій папці (/nagios-4.4.3/), скопіюємо файли Evenhandler до /usr/local/nagios/libexec/ (рис. Е16).

cp -rvf contrib/eventhandlers/ /usr/local/nagios/libexec/

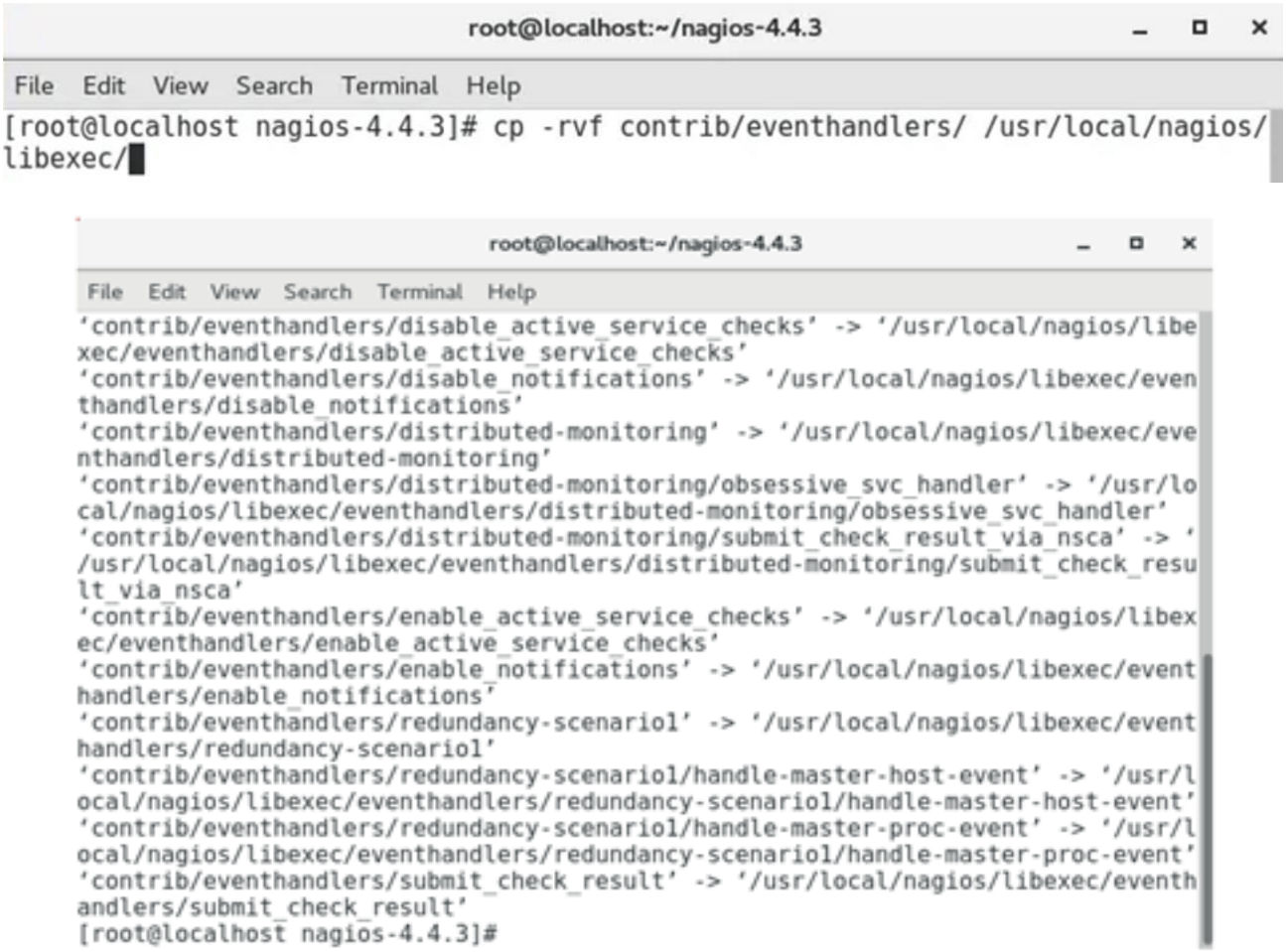


Рисунок Е16 – Копіювання файлів [40]

Крок 9. Змінимо дозволи обробників подій (рис. Е17).

chown -R nagios:naggrp /usr/local/nagios/libexec/eventhandlers

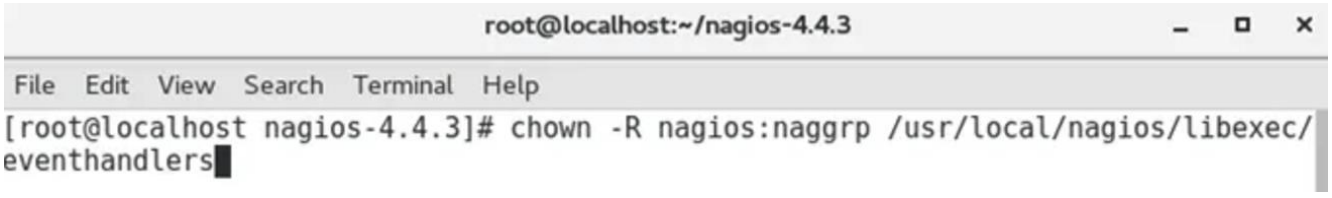


Рисунок Е17 – Зміна прав на обробник подій [40]

Крок 10. Створення користувача nagios (для входу в Інтернет) і генерація його пароля (рис. Е18).

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

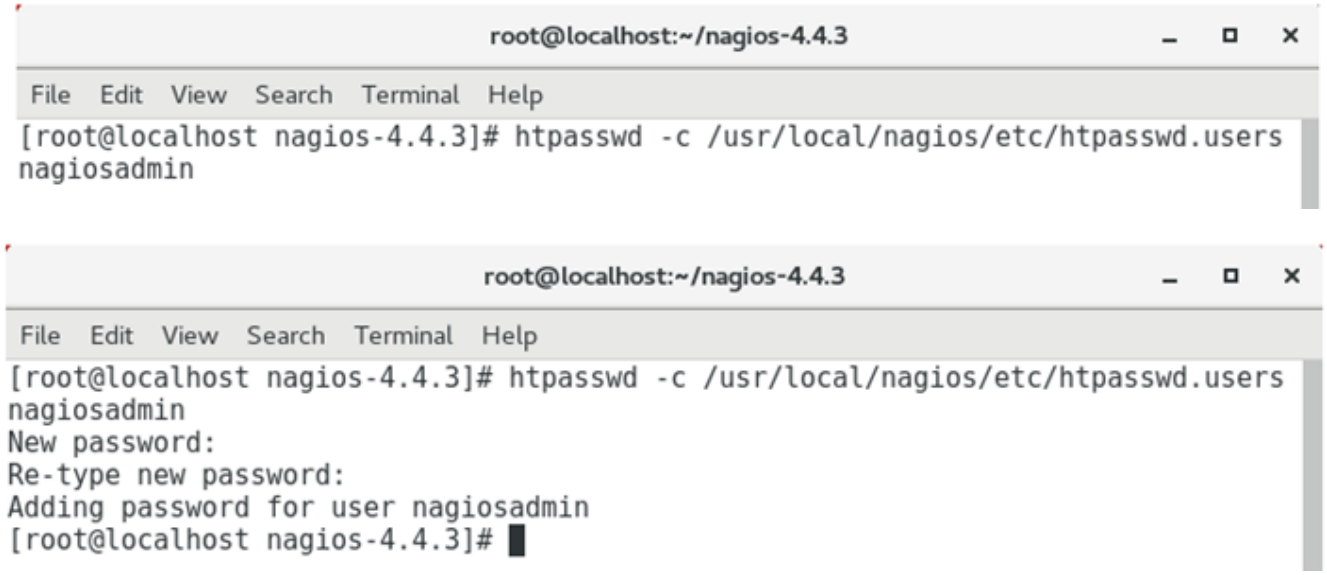


Рисунок Е18 – Створення користувача nagios [41]

Крок 11. Запуск веб-служби, увімкнення її автоматичного запуску після перезавантаження сервера. Запустимо веб-сервіс:

systemctl start httpd.service

увімкнення автоматичного запуску після перезавантаження (рис. Е19):

systemctl enable httpd.service

Рисунок Е19 – Налаштування веб-сервісу на автоматичний запуск [42]

Потрібно перевірити стан веб-служби, щоб переконатися, що вона працює (рис. Е20):

systemctl status httpd.service

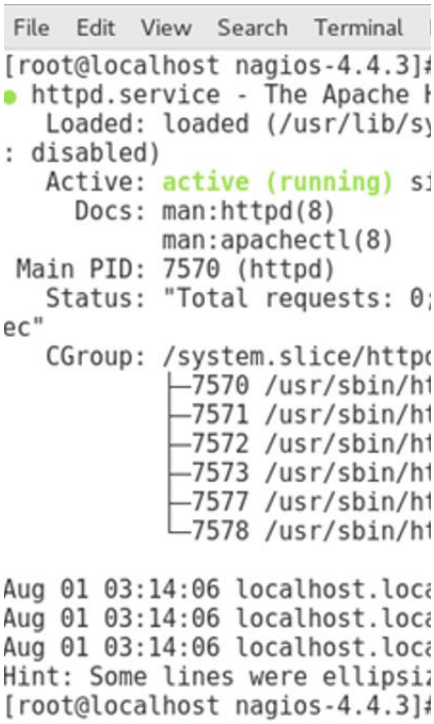


Рисунок Е20 – Статус роботи демону httpd.service [42]

Крок 12. Запуск служби Nagios. Запустимо сервери nagios.

systemctl start nagios.service

Увімкнення автоматичного запуску після перезавантаження (рис. Е21):

systemctl enable nagios.service

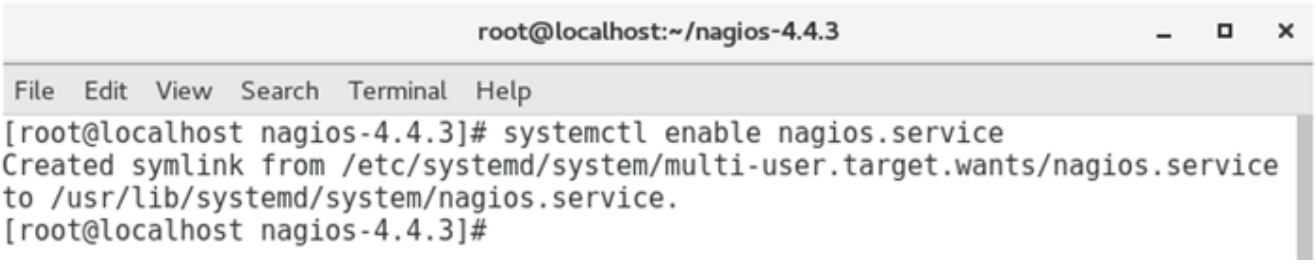


Рисунок Е21 – Налаштування служби Nagios на автоматичний запуск [42]

Перевіримо статус служби nagios, щоб переконатися, що вона працює (рис. Е22):

systemctl status nagios.service

Рисунок Е22 – Статус роботи демону nagios.service [42]

Крок 13. Завантажимо плагіни Nagios (рис. Е23).

wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz

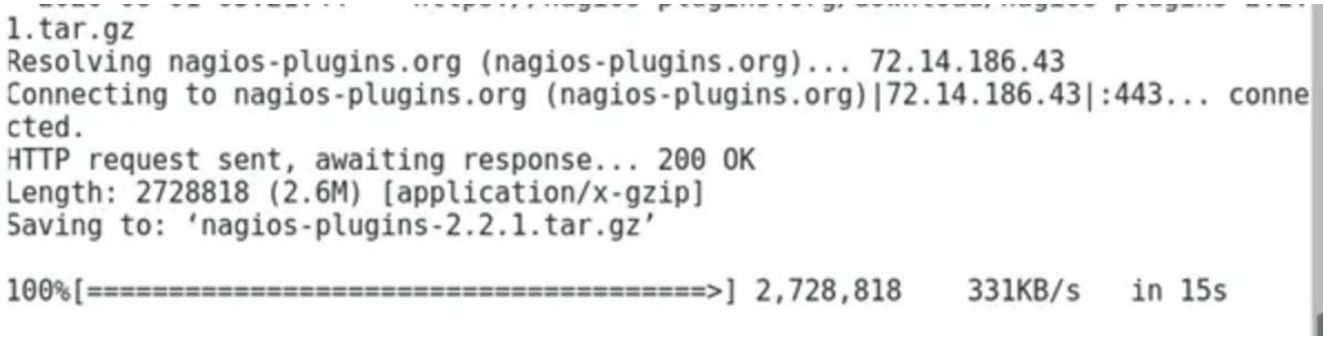


Рисунок Е23 – Завантаження плагінів Nagios [43]

Крок 14. Розпакуймо завантажені плагіни Nagios. змінимо каталог на папку, що містить завантажений файл (рис. Е24). Виконаємо цю команду:

tar -xvf nagios-plugins-2.2.1.tar.gz

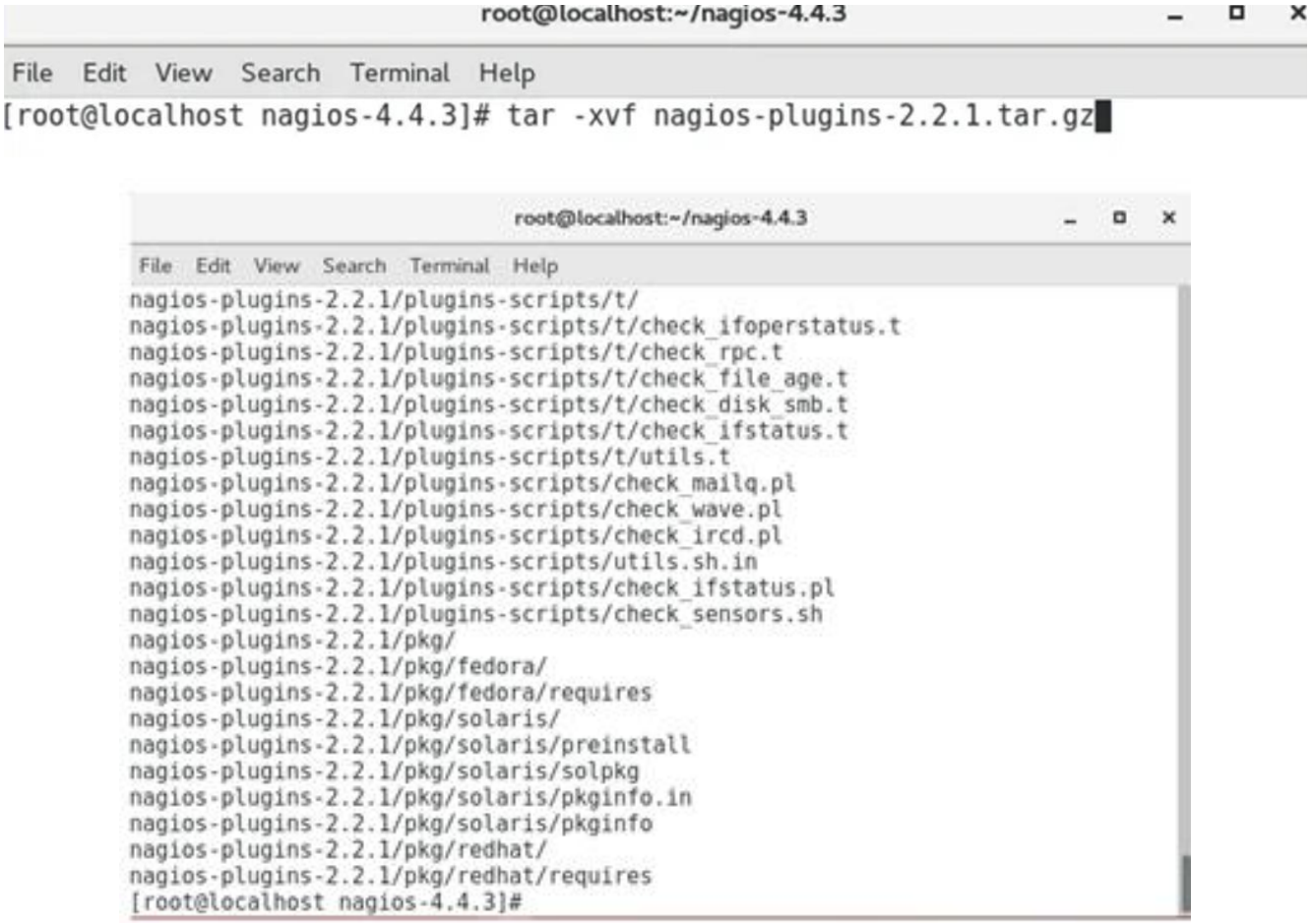
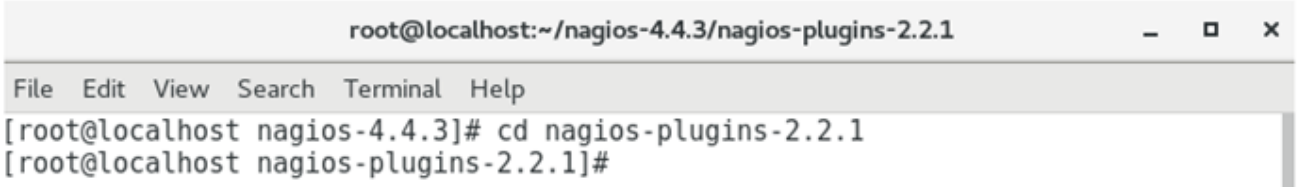


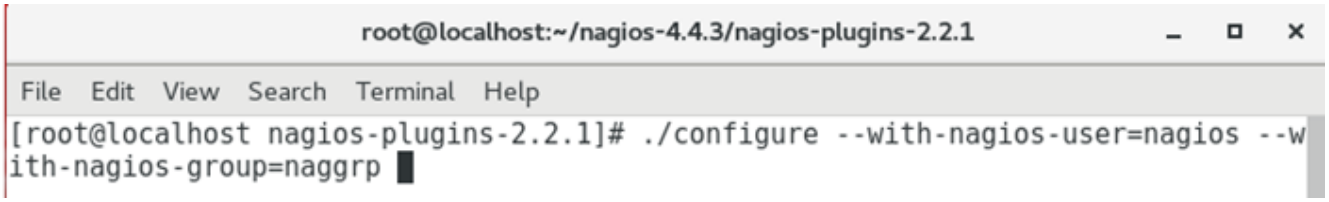
Рисунок Е24 – Деархівація плагіну [43]

Крок 15. Встановимо плагіни Nagios (рис. Е25–Е27). Змінимо каталог на папку, що містить розархівовані файли з кроку 14.

cd nagios-plugins-2.2.1/



./configure — with-nagios-user=nagios — with-nagios-group=naggrp



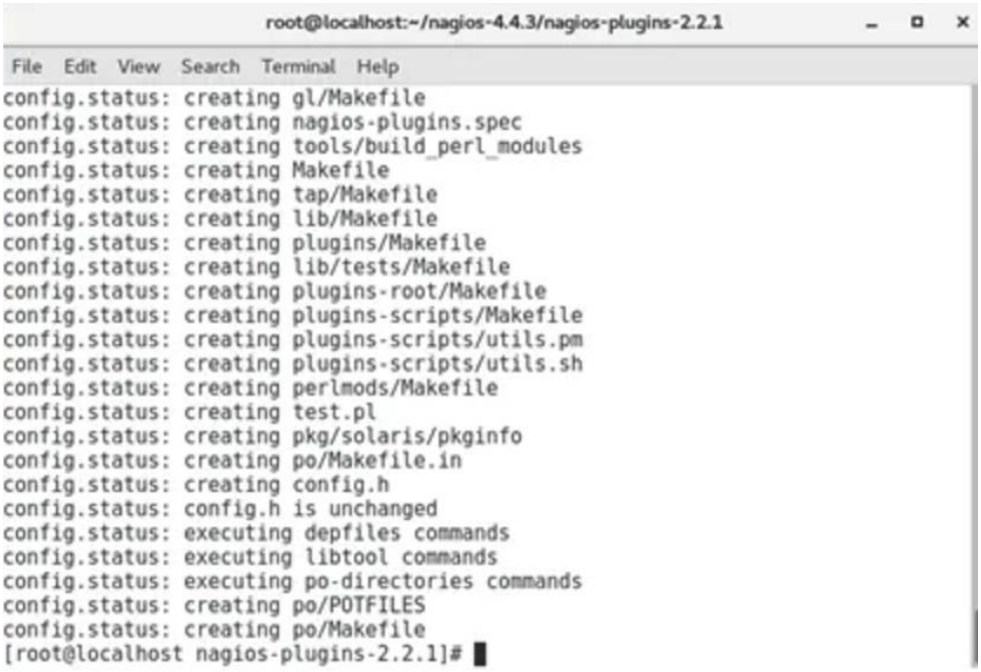


Рисунок Е25 – Створення залежностей для плагіну [43]

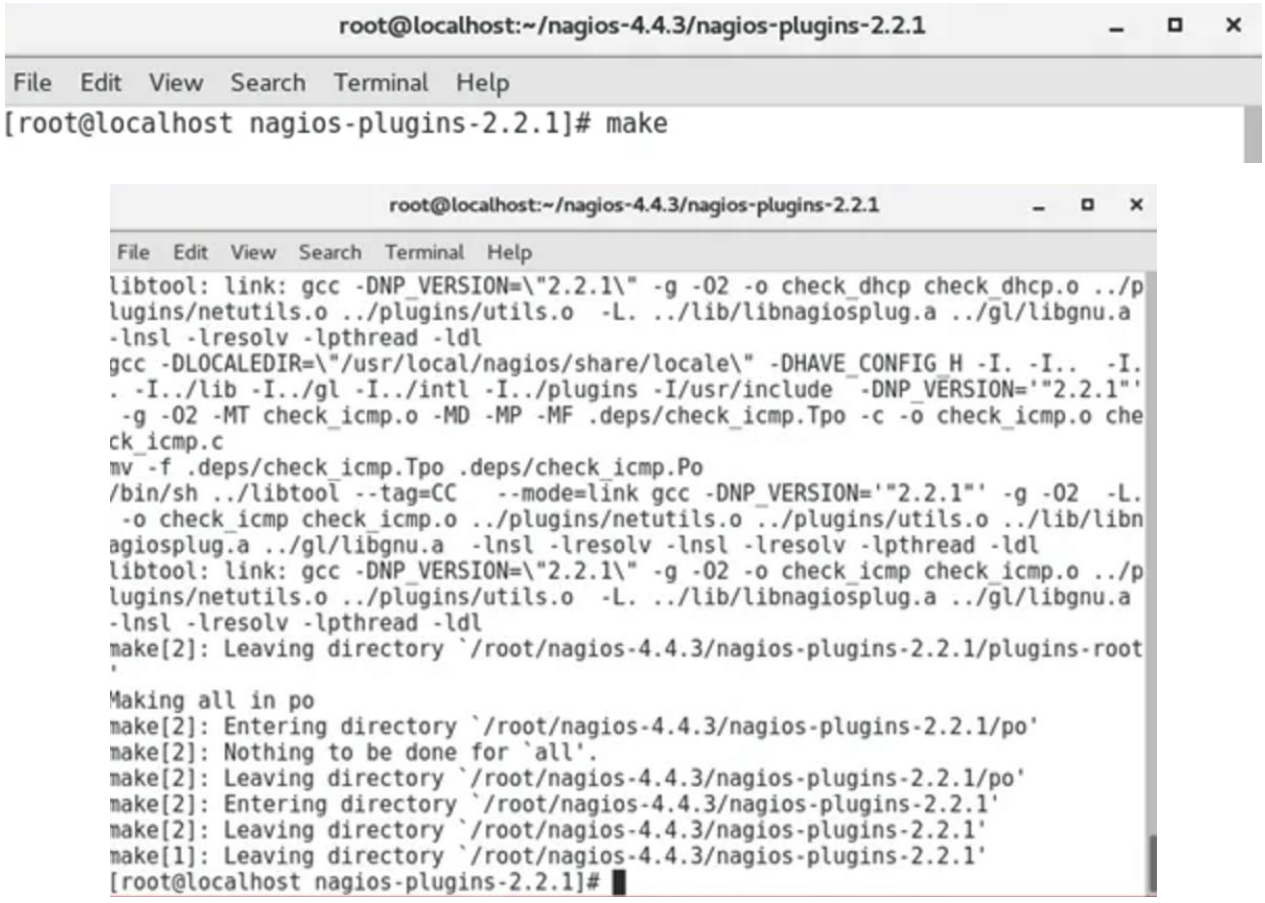


Рисунок Е26 – Збірка конфігурації [39]

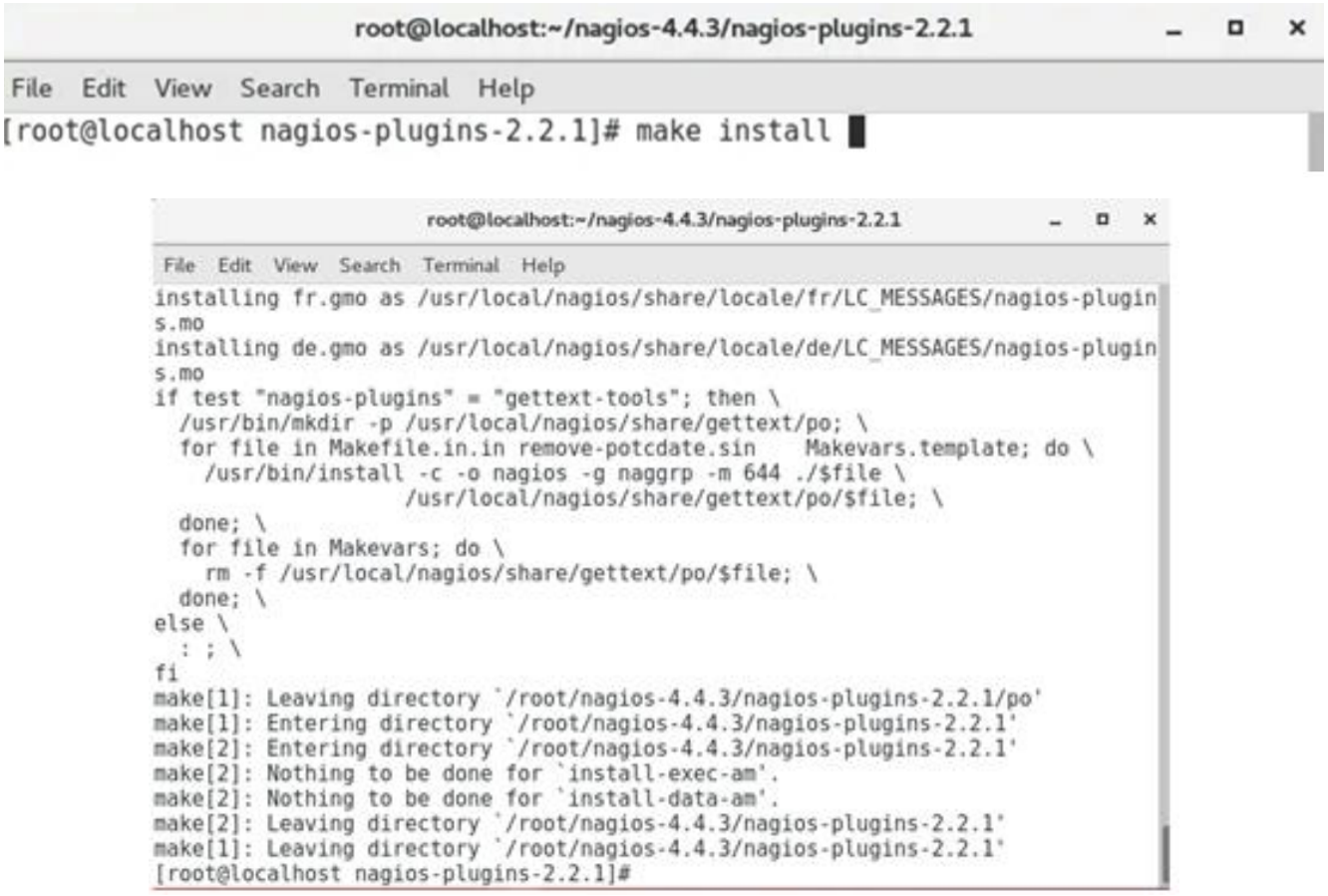


Рисунок Е27 – Інсталюємо збірку [39]

Крок 16. Завантажимо файл NRPE, в папку tmp (рис. Е28).

wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-3.2.1/nrpe-3.2.1.tar.gz -P /tmp/

Рисунок Е28 – Завантаження файлу NRPE [44]

Крок 17. Розпакуймо вихідний файл NRPE (рис. Е29–Е30).

tar zxvf nrpe-3.2.1.tar.gz

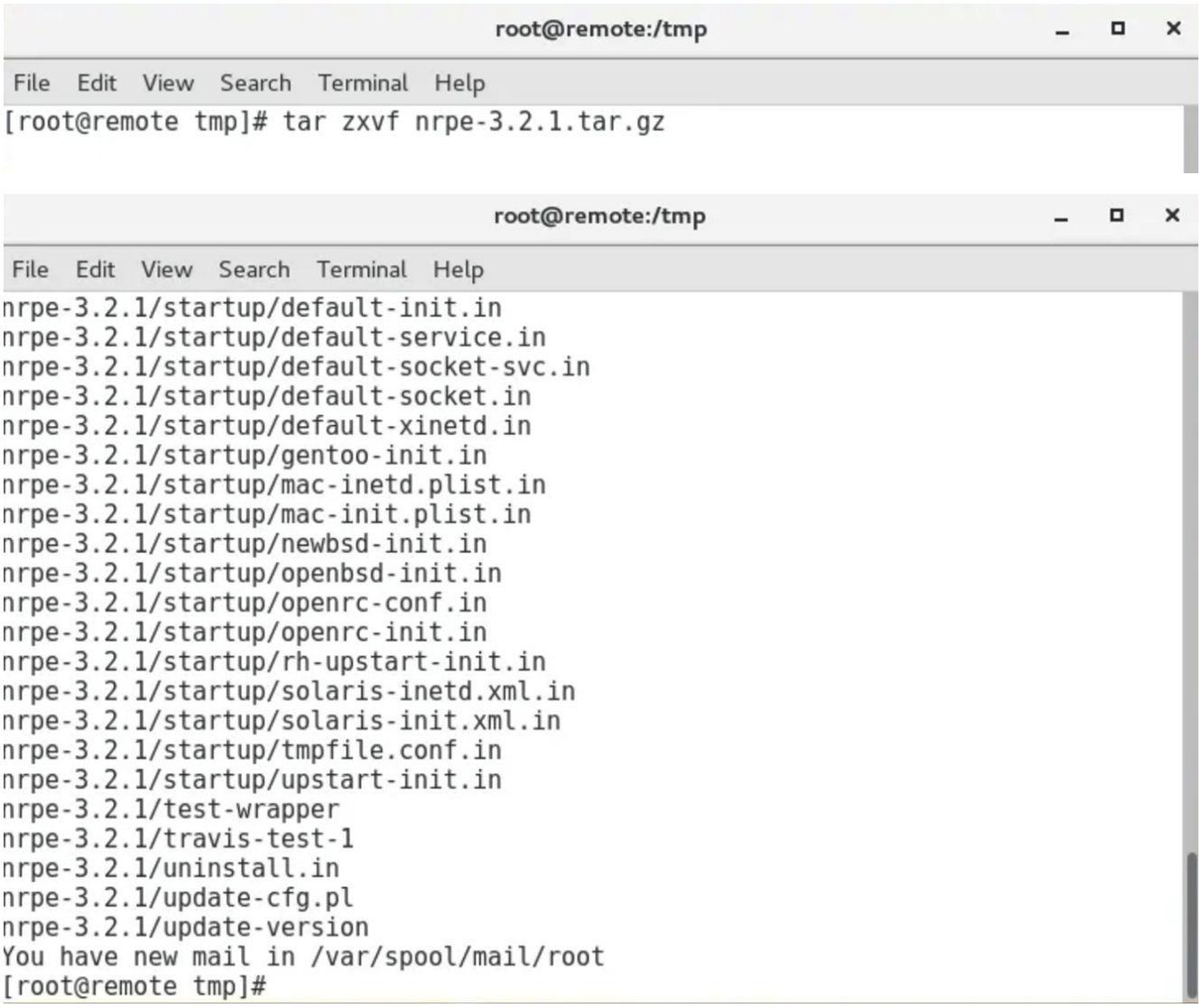
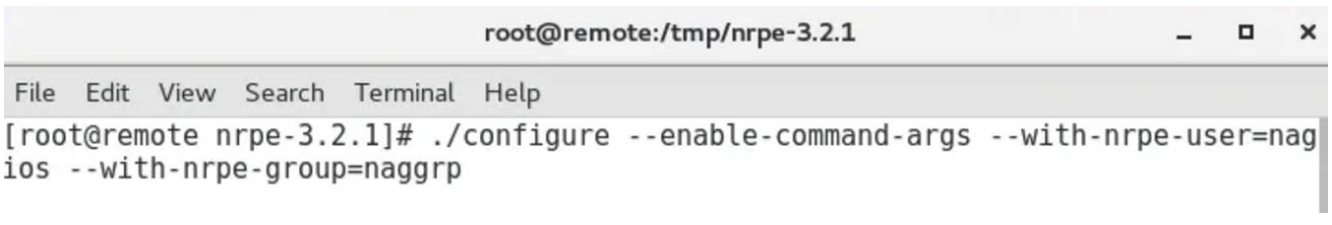


Рисунок Е29 – Розпаковка файлу [44]

Налаштуємо NRPE.

./configure --enable-command-args --with-nrpe-user=nagios --with-nrpe-group=naggrp



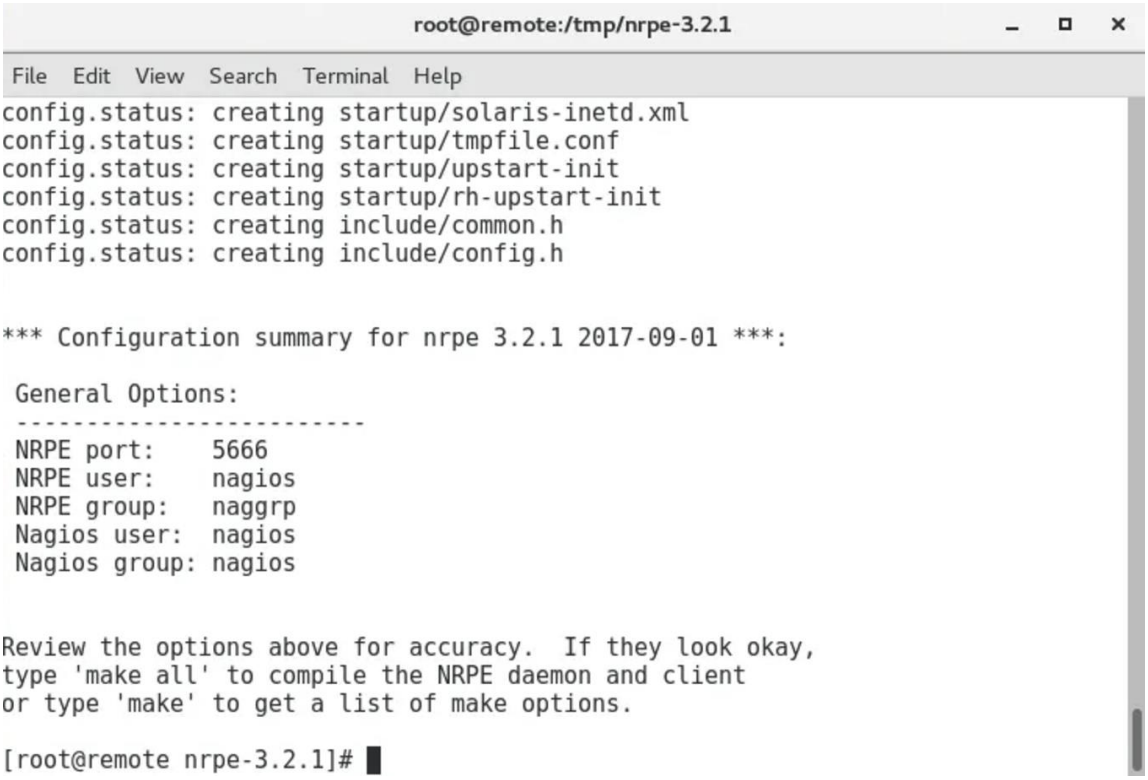


Рисунок Е30 – Збірка конфігурації [44]

Перебуваючи в каталозі /nrpe-3.2.1/, виконаймо такі команди (рис. Е31–Е34):

Make all

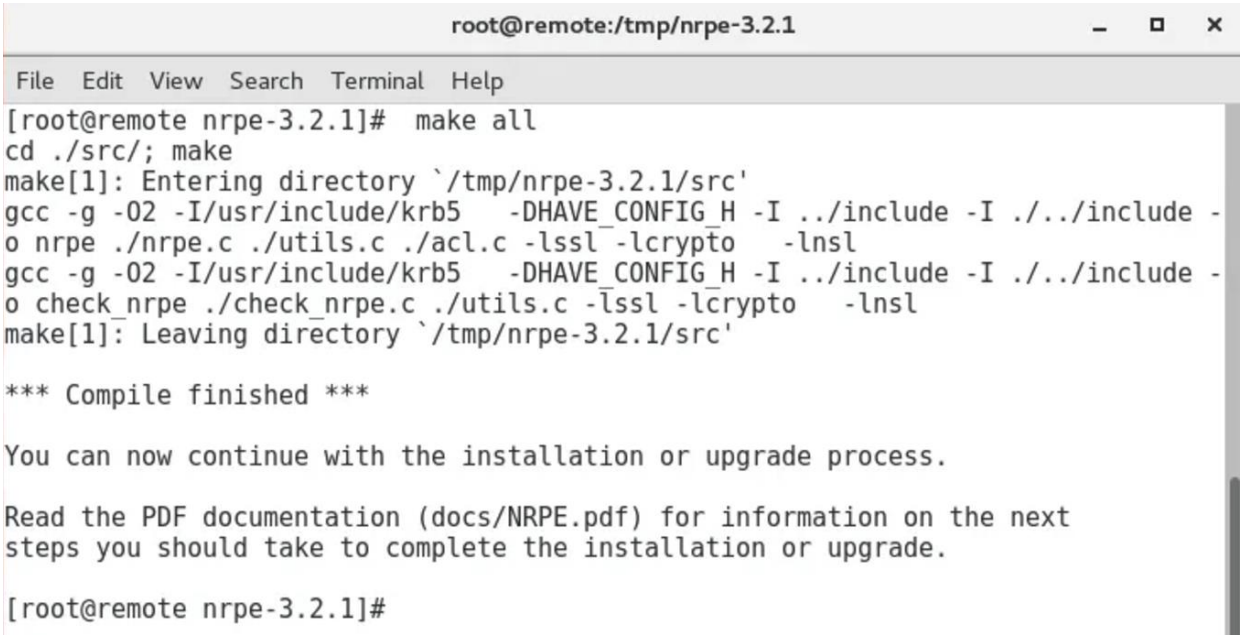


Рисунок Е31 – Make all [39]

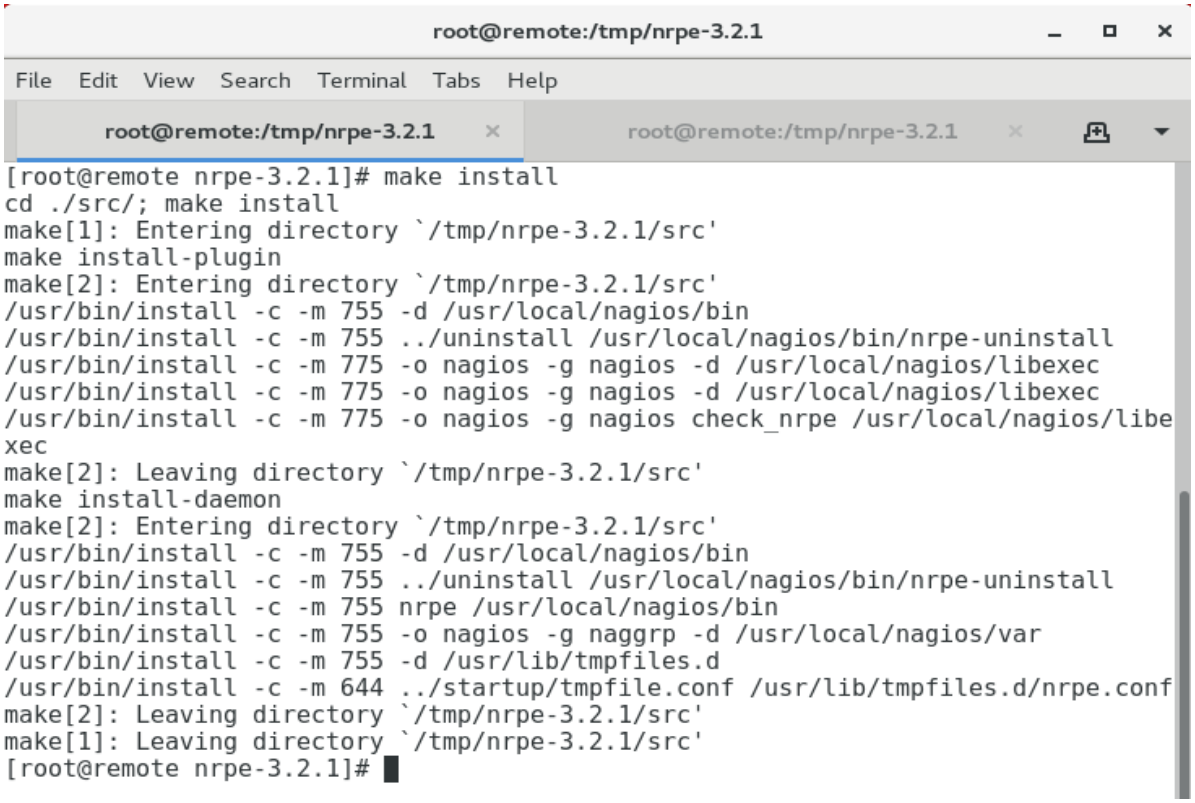


Рисунок Е32 – Make install [39]

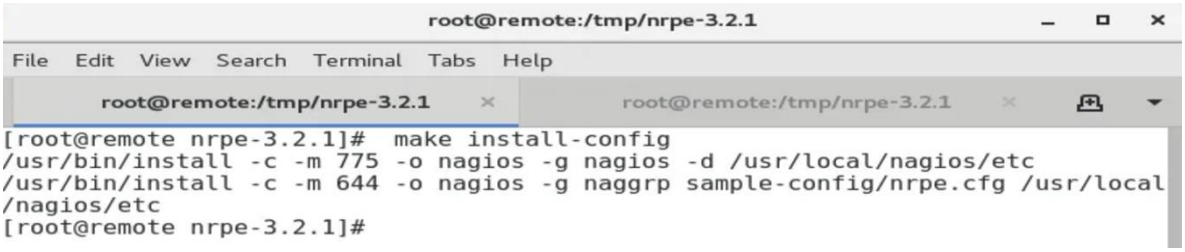


Рисунок Е33 – Make install-config [39]

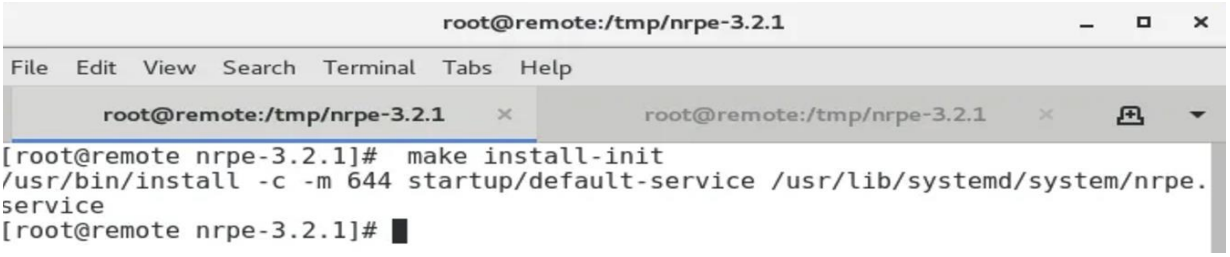


Рисунок Е34 – Make install-init [39]

Nagios успішно зконфігуровано та налаштовано. Увійдемо до веб-інтерфейсу nagios за допомогою ip-адреси сервера nagios, а потім /nagios: http://192.168.20.131/nagios/. Буде запропоновано ввести облікові дані для входу,

використовуйте облікові дані, створені на кроці 10, тобто nagiosadmin і згенерований пароль (рис. Е35).

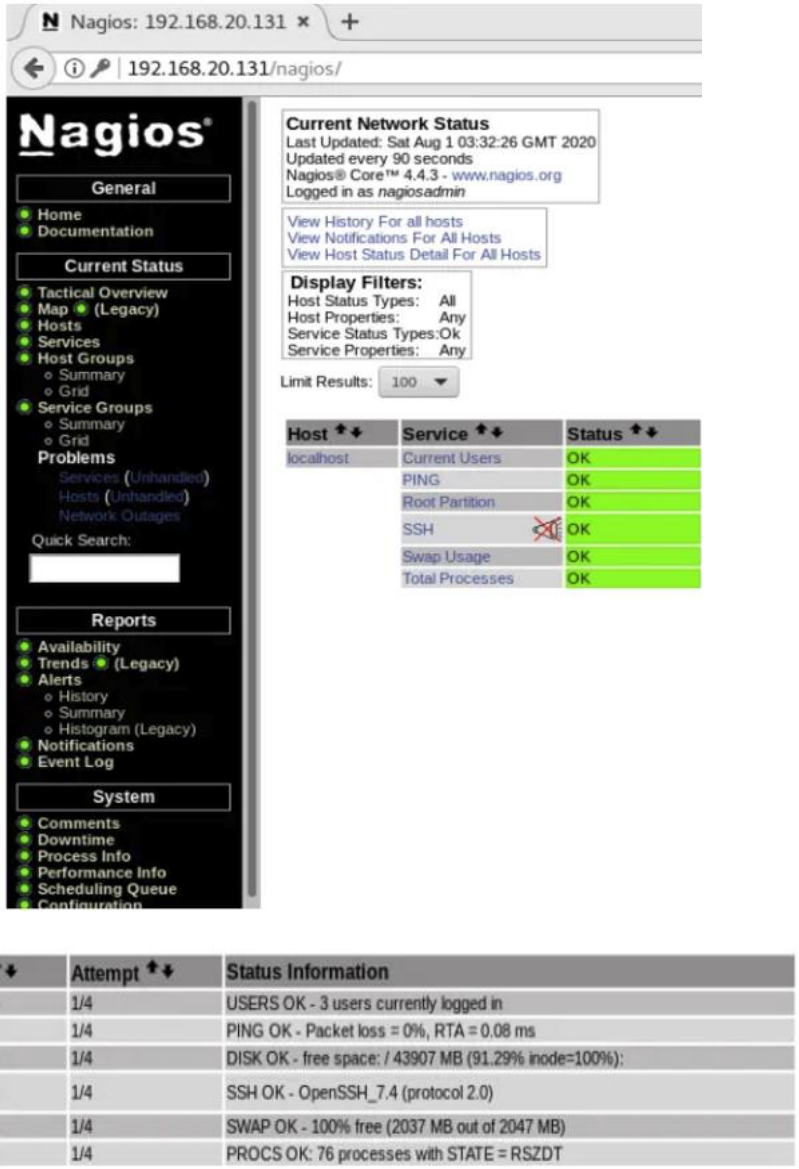


Рисунок Е35 – Моніторинг серверу Nagios Core

Додаток Ж

Покроковий план інтеграції та тестування

Крок 1. Підготовка Windows-системи для інтеграції. Він передбачає:

1.1. Встановлення агента моніторингу. Nagios використовує агенти для збору даних із Windows-систем. Найпоширеніші агенти:

- **NSClient++**: Забезпечує моніторинг ресурсів Windows, таких як CPU, пам'ять, дисковий простір, служби, події.
- **WMI (Windows Management Instrumentation)**: Може використовуватися для безагентського моніторингу через плагіни Nagios.

Дії:

1. Завантажити та встановити NSClient++ на Windows-сервер.
2. Налаштувати файл конфігурації nsclient.ini для доступу з Nagios-сервера (активувати необхідні модулі (CheckDisk, CheckSystem, NRPE тощо)).
3. Дозволити підключення через порт (зазвичай 5666) у брандмауері Windows.

1.2. Налаштування дозволів. Для безагентського моніторингу через WMI треба:

- Забезпечити доступ до WMI-інтерфейсу на Windows-сервері.
- Налаштувати обліковий запис із правами на читання конфігурацій через WMI.

Крок 2. Налаштування Nagios для моніторингу Windows. Він передбачає:

2.1. Додавання хоста. Додати інформацію про Windows-систему в конфігураційний файл Nagios:

```
cfg
define host {
    use         windows-server  ; Шаблон хоста для Windows
    host_name       win-server1     ; Ім'я сервера
```

```
    alias           Windows Server  ; Опис
    address         192.168.1.100   ; IP-адреса
}
```

2.2. Додавання перевірок (services). Додати служби для моніторингу:

```
cfg
Copy code
define service {
    use                     generic-service
    host_name               win-server1
    service_description     CPU Usage
    check_command           check_nt!CPULOAD!-w 80 -c 90
}
define service {
    use                     generic-service
    host_name               win-server1
    service_description     Disk Usage
    check_command           check_nt!USEDDISKSPACE!-l C -w 80
-c 90
}
```

2.3. Перевірка плагінів. Переконатися, що плагіни для перевірки Windows-систем встановлені на сервері Nagios:

- ***check_nt***: Використовується з NSClient++.
- ***check_wmi_plus***: Дозволяє збирати дані через WMI.

Крок 3. Тестування системи моніторингу. Він передбачає:

3.1. Проведення тестових перевірок. Використовувати командний рядок для тестування плагінів:

```
bash
/usr/local/nagios/libexec/check_nt -H 192.168.1.100 -p 5666 -
v CPULOAD -w 80 -c 90
```

Перевірити, чи відповідає агент і чи отримуються дані.

3.2. Створення сценаріїв навантаження:

- Створити штучні навантаження на сервер (наприклад, запустити ресурсоємні процеси).

- Перевірити, як система моніторингу фіксує зміни.

Крок 4. Аналіз результатів тестування. Він передбачає:

4.1. Збір даних. Зібрати метрики, що відображають продуктивність та конфігурацію Windows-систем: середнє навантаження CPU; використання пам'яті; обсяг доступного дискового простору; стан ключових служб (IIS, SQL Server тощо).

4.2. Оцінка точності та ефективності. Оцінка точності та ефективності виконується через оцінку: точності (порівняти дані, зібрані Nagios, із реальними показниками на сервері); час відгуку (оцінити затримки між виявленням проблеми і створенням сповіщення); ресурсомісткість (аналіз навантаження на Windows-сервер і сервер Nagios під час моніторингу).

4.3. Оцінка стабільності. Оцінка стабільності шляхом перевірки - чи система моніторингу залишається стабільною при тривалому спостереженні та великій кількості запитів.

Крок 5. Оптимізація моніторингу. Він передбачає:

- **Зниження навантаження:** Встановити оптимальну частоту перевірок для зменшення використання ресурсів.
- **Визначенн фільтра подій:** Виключити несуттєві перевірки, щоб уникнути надмірної кількості сповіщень.
- **Інтеграцію з Grafana:** Підключити Grafana для візуалізації даних і покращення аналізу.

Крок 6. Висновки за результатами. Він передбачає:

1. **Функціональність Nagios:** Оцінити, наскільки інструмент відповідає вимогам моніторингу Windows-систем.
2. **Продуктивність:** Визначити, чи система справляється з великим навантаженням і чи надає актуальні дані.

3. **Придатність до масштабування:** Перевірити можливість інтеграції з іншими інструментами та додатковими хостами.

Такий підхід забезпечує повноцінну інтеграцію Windows-систем у моніторингове середовище, а також дає змогу оцінити ефективність і надійність Nagios у роботі з гетерогенними системами.

Додаток И

Розгорнутий опис рекомендацій щодо інтеграції Windows-системи в моніторингове середовище на базі Nagios

Крок 1. *Завантаження та встановлення NSClient++.* Розпочнемо з завантаження інструменту NSClient++, який необхідний для інтеграції Windows-систем у середовище моніторингу Nagios. Перейдімо за посиланням https://nsclient.org/download/0.5.2/#0.5.2.39 та завантажимо останню доступну версію (0.5.2.39). Після завершення завантаження запустимо інсталятор і виконаємо покрокову установку, враховуючи налаштування, які забезпечать сумісність із Nagios (рис. И1–И3). Це дозволить підключити Windows-систему до моніторингового середовища та почати відстеження її стану й параметрів.

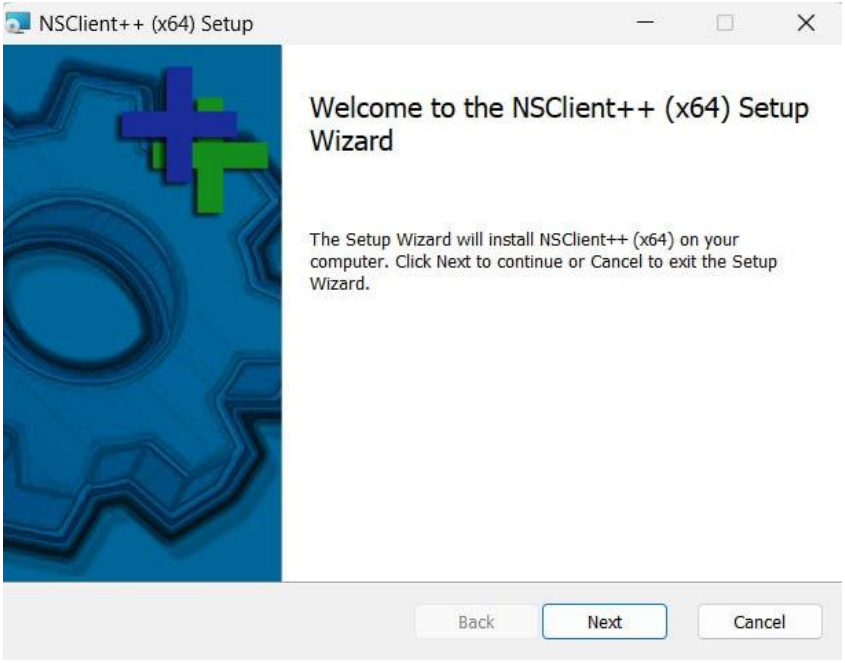


Рисунок И1 – Пакет NSClient.msi

Введемо IP-адресу сервера Nagios Core у полі «Дозволені хости» та пароль у полі «Пароль», виберемо стандартний запропонований системою (рис. И2).

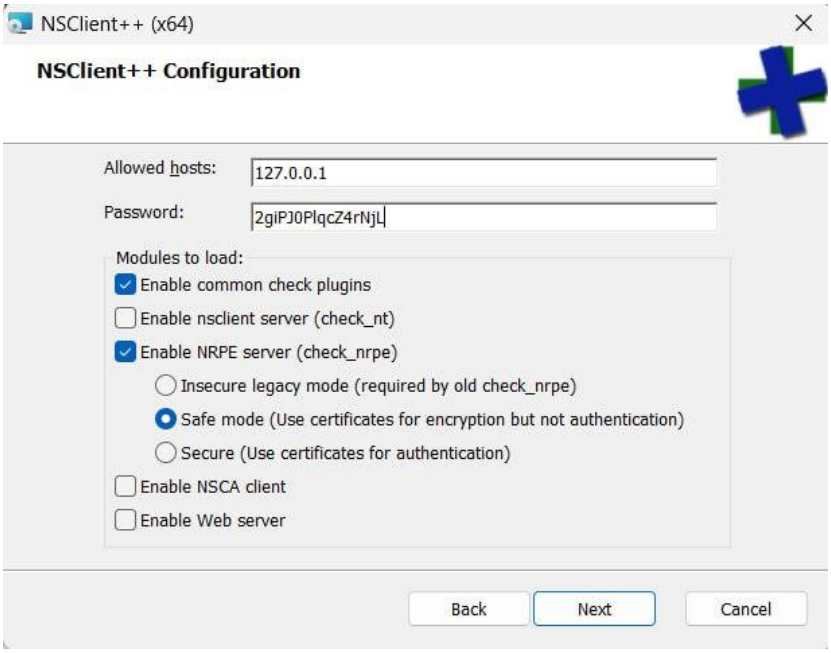


Рисунок И2 – Конфігурація агента

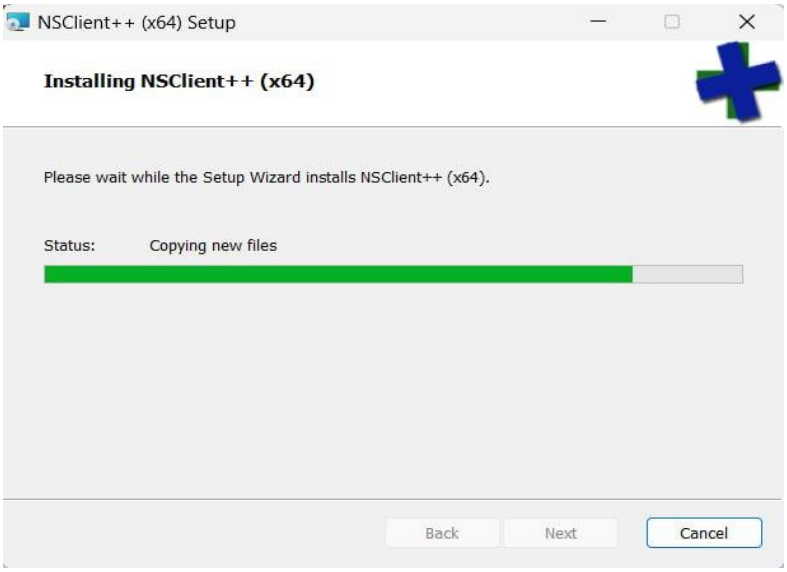


Рисунок И3 – Процес встановлення

Крок 2. *Додамо вікно Windows*, додавши запис у файл windows.cfg, розташований за адресою (на сервері Nagios): /usr/local/nagios/etc/objects (рис. И4).

```
vi /usr/local/nagios/etc/objects/windows.cfg
```

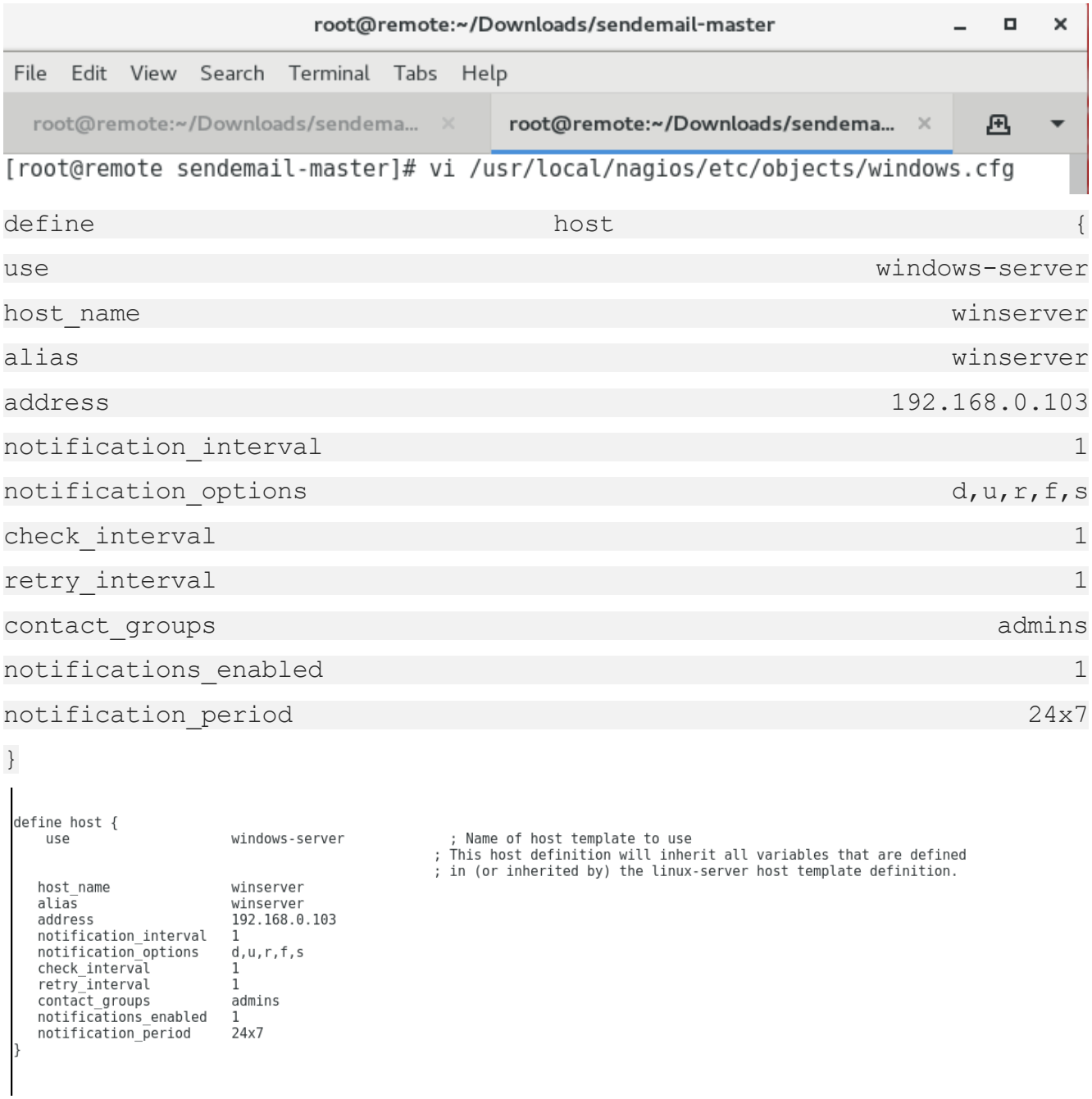


Рисунок И4 – Конфігураційний файл клієнта [45]

За замовчуванням у Nagios увімкнено наступні служби Windows (/usr/local/nagios/etc/objects/windows.cfg), також можемо смостійно додати будь-яку службу, яку бажаємо контролювати (рис. И5):

```
define service {
use generic-service
host_name winserver
service_description NSClient++ Version
```

```
check_command check_nt!CLIENTVERSION
}
define service {
use generic-service
host_name winserver
service_description Uptime
check_command check_nt!UPTIME
}
define service {
use generic-service
host_name winserver
service_description CPU Load
check_command check_nt!CPULOAD!-l 5,80,90
}
define service {
use generic-service
host_name winserver
service_description Memory Usage
check_command check_nt!MEMUSE!-w 80 -c 90
}
```

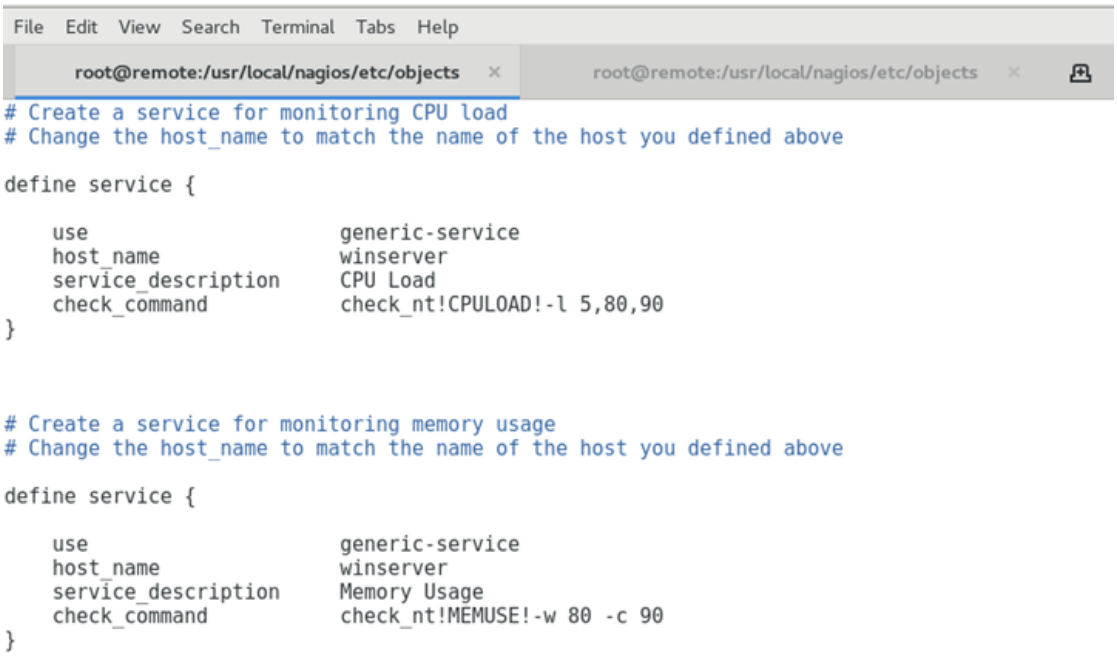


Рисунок И5 – vi /usr/local/nagios/etc/objects/windows.cfg [45]

Пояснення різних параметрів:

host_name = Коротке ім'я, яке використовується для ідентифікації хоста.

notification_interval = кількість «одиниць часу» очікування перед повторним сповіщенням контакту про те, що ця служба все ще не працює або недоступна. Це значення в хвилинах.

notification_options = Використовується для визначення часу надсилання сповіщень для хосту. Дійсні параметри – це комбінація одного або кількох із наведеного нижче: d = надсилати сповіщення у стані НЕМАЄ, u = надсилати сповіщення про стан НЕДОСТУПНО, r = надсилати сповіщення про відновлення (стан OK), f = надсилати сповіщення, коли хост запускається і припиняє махати, і s = надсилати сповіщення про початок і завершення запланованого простою [31].

check_interval = Використовується для визначення кількості «одиниць часу», які потрібно чекати перед плануванням наступного «звичайна» перевірка сервісу. «Звичайні» перевірки – це перевірки, які відбуваються, коли служба перебуває в нормальному стані або коли служба перебуває в стані не в порядку, але вже була перевірена повторно max_check_attempts кількість разів.

retry_interval = використовується для визначення кількості «одиниць часу» для зачекайте, перш ніж запланувати повторну перевірку служби.

contact_groups = Це список коротких імен груп контактів, які слід сповіщати, коли виникають проблеми (або відновлення) з цією послугою. Кілька груп контактів слід розділяти комами.

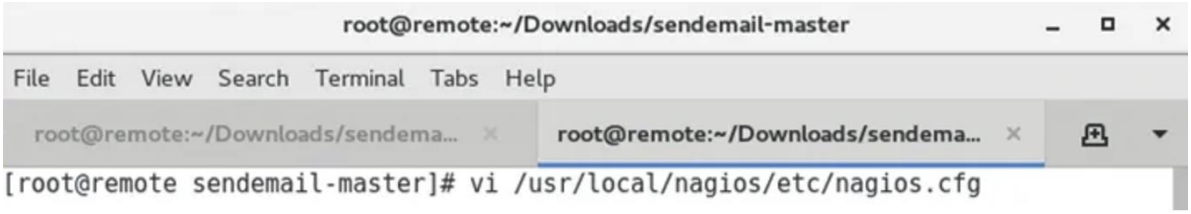
notifications_enabled = використовується, щоб визначити, чи ввімкнено сповіщення для хосту чи служби. Значення: 0 = вимкнути сповіщення хоста, 1 = увімкнути сповіщення хоста.

notification_period = використовується для вказівки короткої назви періоду часу, протягом якого сповіщення про події для цієї служби можуть надсилатися контактам. Жодні службові сповіщення не надсилатимуться протягом періоду часу, який не охоплюється періодом часу

Після завершення збережемо зміни та вийдемо.

Крок 3. Далі, на сервері Nagios, відкриємо nagios.cfg, розташований за адресою: /usr/local/nagios/etc/.

```
vi /usr/local/nagios/etc/nagios.cfg
```



Розкоментуємо рядок нижче, видаливши знак #, і збережемо зміни (рис. И6).

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

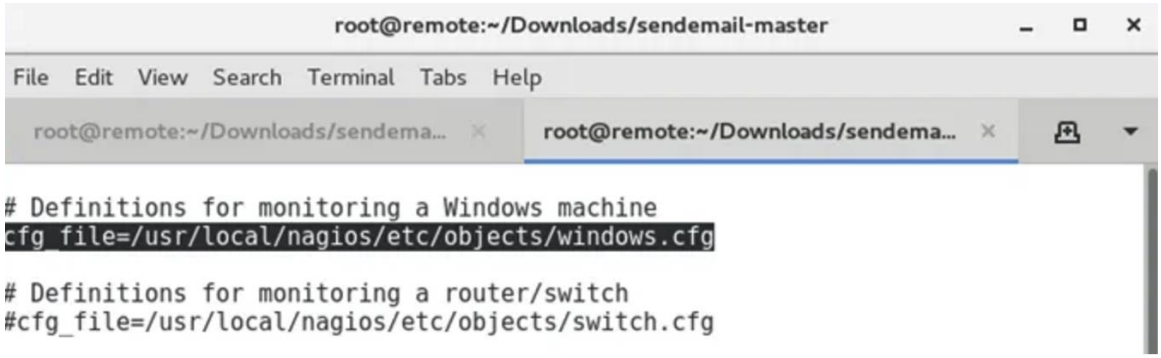


Рисунок И6 – Коригування файлу інтеграції [45]

Перезапустимо службу Nagios (рис. И7).

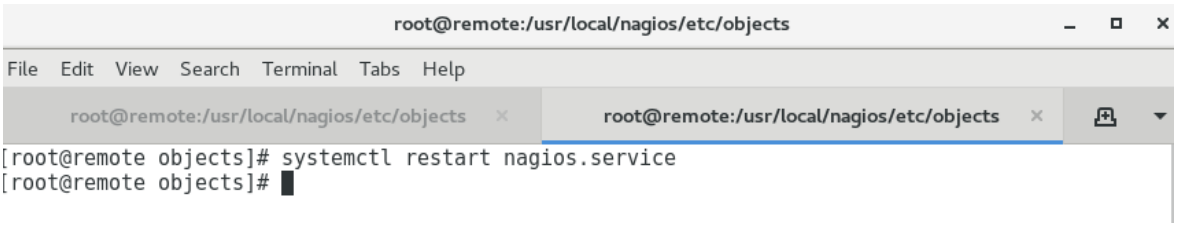


Рисунок И7 – Перезапуск служби Nagios [45]

Крок 5. Змінимо файл nsclient.ini, розташований у C:\Program Files\NSClient++.

Недокументований розділ [/settings/default];

ПАРОЛЬ - Пароль, який використовується для автентифікації на сервері

; пароль = час;

ДОЗВОЛЕНІ ХОСТИ - Розділений список дозволених хостів. Для створення діапазонів можна використовувати мережеві маски (синтаксис /) або *. Дозволені хости = 192.168.0.11

порт = 12489;

Недокументований розділ [/modules];

CheckSystem – різноманітні системні перевірки, такі як завантаження ЦП, стан процесу, використання пам’яті стану служби та лічильники PDH.

CheckSystem = 1;

NSClientServer - сервер, який прослуховує вхідне підключення check_nt і обробляє вхідні запити.

NSClientServer = 1;

CheckExternalScripts - Виконання зовнішніх сценаріїв

CheckExternalScripts = 1;

CheckHelpers – різноманітні допоміжні функції для розширення інших перевірок.

CheckHelpers = 1;

CheckEventLog - перевірте наявність помилок і попереджень у журналі подій.

CheckEventLog = 1;

CheckDisk - CheckDisk може перевіряти різні файли та диски.

CheckDisk = 1;

CheckNSCP - Використовуйте цей модуль, щоб перевірити працездатність і статус самого NSClient++CheckNSCP = 1;

NRPEServer – сервер, який прослуховує вхідне з’єднання NRPE і обробляє вхідні запити.

NRPEServer = 1;

Недокументований розділ [/modules];

Недокументований ключ CheckExternalScripts = 1;

Недокументований ключ CheckHelpers = 1;

Недокументований ключ CheckEventLog = 1;

Недокументований ключ CheckNSCP = 1;

Недокументований ключCheckDisk = 1;

Недокументований ключCheckSystem = 1;

Недокументований ключNSClientServer = 1;

Недокументований ключNRPEServer = 1

Крок 6. дозвольте порт 12489 через брандмауер Windows (рис. И8).

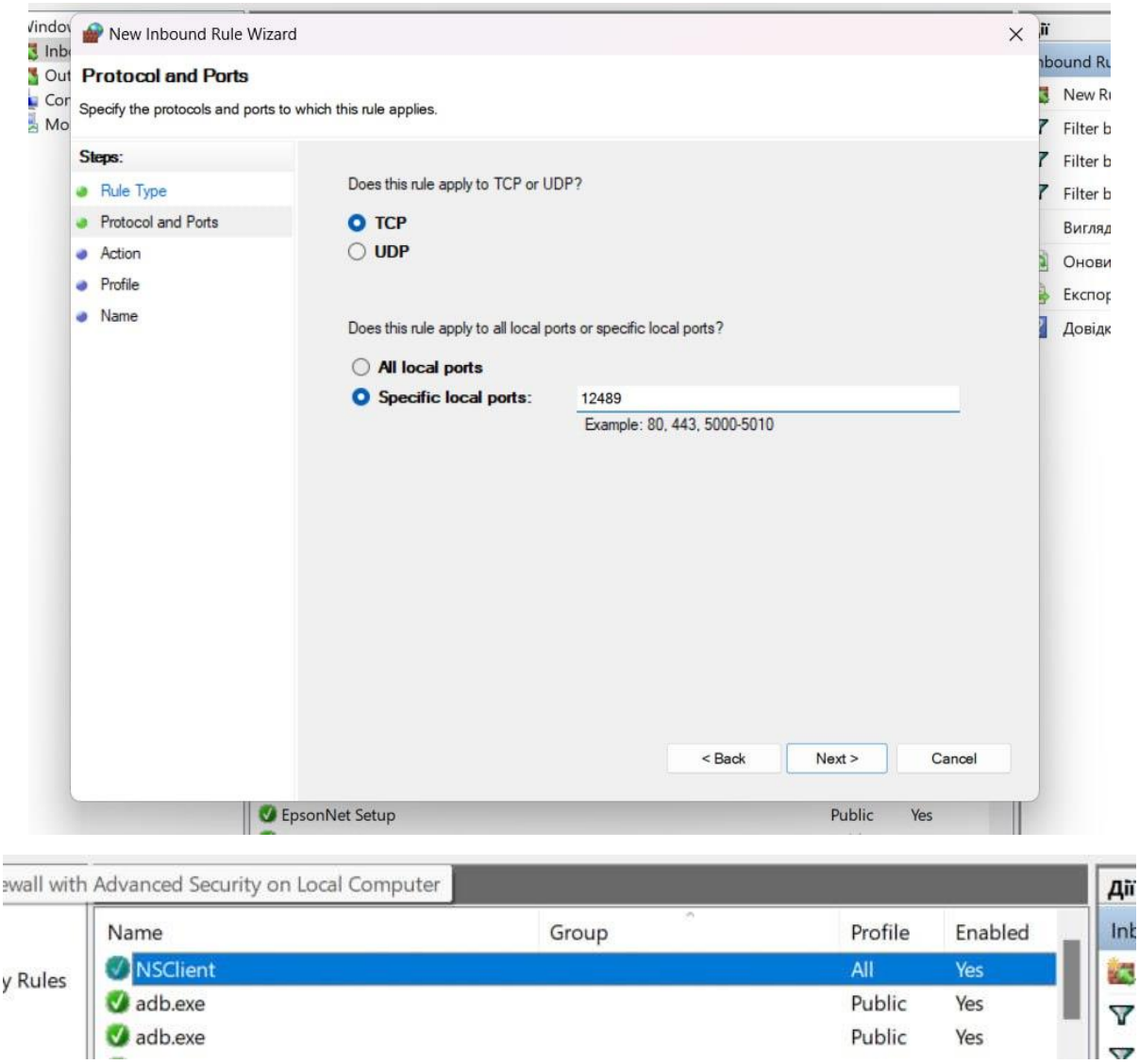


Рисунок И8 – Включене правило на відкриття порту 12489

Крок 7. перезапустіть службу агента моніторингу NSClient++ (рис. И9).

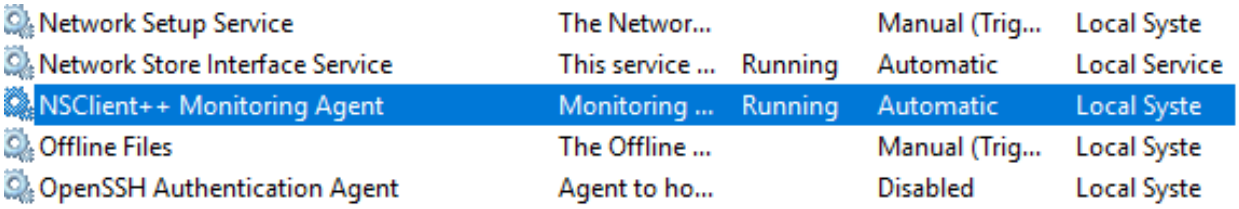


Рисунок И9 – Перезапуск служби

Вікно Windows для моніторингу сервером Nagios успішно налаштовано.

Увійдемо до веб-інтерфейсу nagios за допомогою ip-адреси сервера nagios, а потім http://192.168.20.131/nagios/. Повиннен відобразитися статус сервера Windows на Nagios (рис. И10–И11).

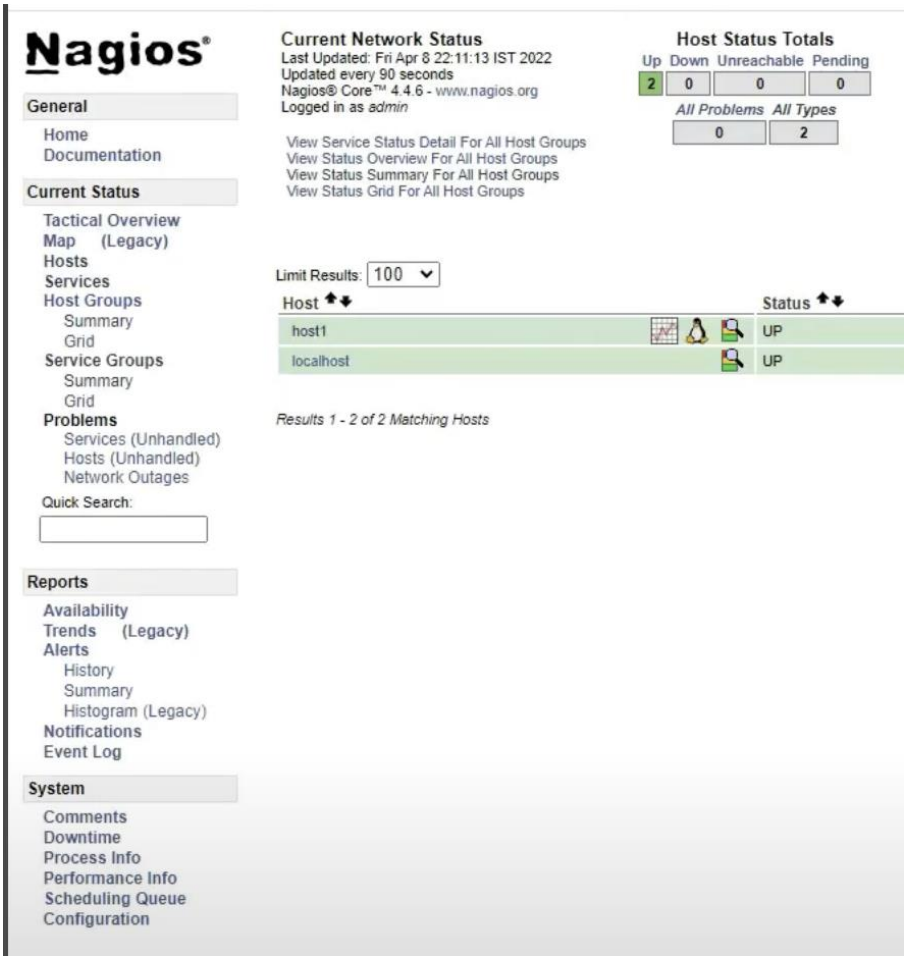
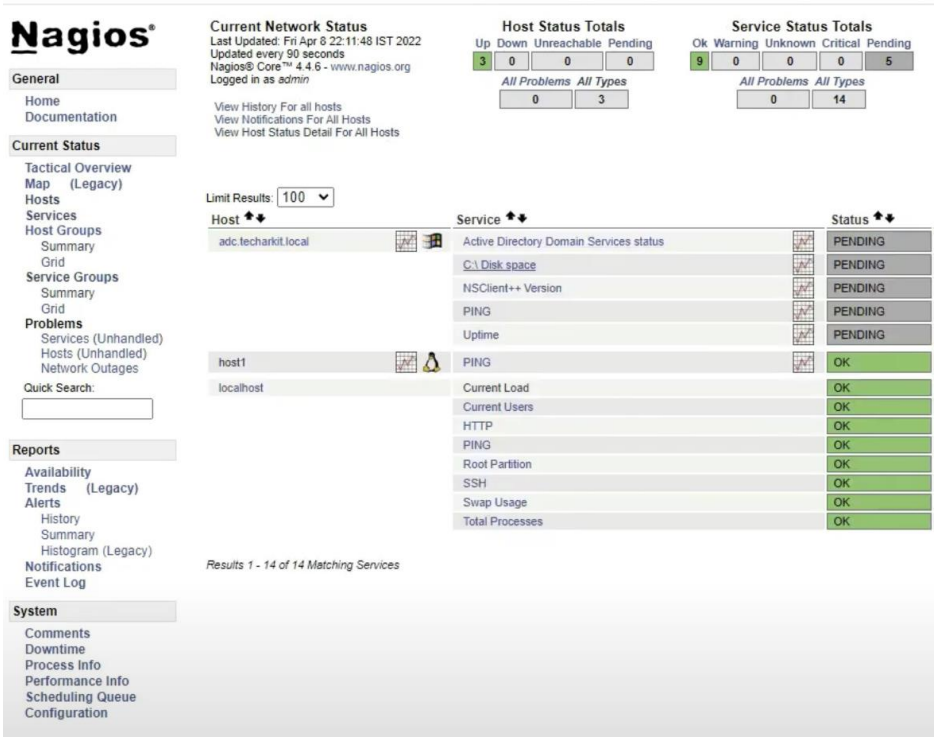


Рисунок И10 – Веб-консоль моніторингу конфігурації та активності серверу



3d 14h 22m 44s	1/4	OK - load average: 0.04, 0.03, 0.05
3d 14h 22m 6s	1/4	USERS OK - 1 users currently logged in
0d 0h 41m 5s	1/4	HTTP OK: HTTP/1.1 301 Moved Permanently - 456 bytes in 0.001 second response time
3d 14h 20m 51s	1/4	PING OK - Packet loss = 0%, RTA = 0.06 ms
3d 14h 20m 14s	1/4	DISK OK - free space: / 14602 MiB (95.18% inode=100%):
3d 14h 19m 36s	1/4	SSH OK - OpenSSH_7.4 (protocol 2.0)
3d 14h 18m 59s	1/4	SWAP OK - 100% free (8191 MB out of 8191 MB)
3d 14h 23m 21s	1/4	PROCS OK: 56 processes with STATE = RSZDT

Рисунок И11 – Продовження Веб-консоль моніторингу конфігурації та активності серверу

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Дослідження системи автоматизованого збору даних про конфігурації серверного обладнання

Виконав студент 6 курсу

Групи КНДМ-63
Бураков Михайло
Керівник роботи

д.т.н, професор кафедри ННІІТ Юрій Катков

Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи – підвищити ефективність застосування системи автоматизованого збору даних моніторингу та управління про конфігурацію серверного обладнання на базі інструменту Nagios.

Об'єкт дослідження – процес використання Nagios для моніторингу та управління конфігурацією серверного обладнання.

Предмет дослідження – інструмент автоматизованого збору даних моніторингу та управління про конфігурацію серверного обладнання.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз сучасних інструментів для моніторингу та управління в системі автоматизованого збору даних про конфігурації серверного обладнання.
2. Провести дослідження програмної реалізації системи моніторингу конфігурації серверного обладнання на базі інструменту Nagios.
3. Розробити методику тестування системи та аналіз результатів тестування моніторингу конфігурацій серверного обладнання на базі інструменту Nagios.

ПОРІВНЯННЯ ІНСТРУМЕНТІВ МОНІТОРИНГУ

Таблиця 1.1 Порівняння інструментів моніторингу

Критерій	Zabbix	Prometheus	Grafana	ELK Stack	Ansible	SNMP	Nagios
Архітекту-ра	Централі-зована	Розподіле-на, оптимі-зована для контейне-рів	Візуалі-зація даних	Розподіле-на, з ком-понентами для збору даних	Деклара-тивна	Розподілена	Централі-зована
Масштабо-ваність	Добра, але з обмеже-ннями на великі середо-вища	Висока, адаптована для великих кластерів	Залежить від джерела даних	Висока, але вимоги до ресурсів зростають	Добра, але залежить від складно-сті	Обмежена, в основному для простих мереж	Добра, але може бути складно для налашту-вання
Налашту-вання та конфігу-рація	Складне, багато параме-трів	Середнє, прості конфігу-раційні файли	Легке, з інтуї-тивним інтерфей-сом	Складне, вимагає знань з Elastic search	Середнє, проста синтакси-чна структура	Легке, але вимагає налаштування на кожному пристрої	Складне, багато налашту-вань
Типи метрик	Часові ряди, статуси, події	Часові ряди, метрики системи	Візуалі-зація будь-яких даних	Логи, метрики, події	Конфігу-рації, але не специфі-чно для метрик	Статуси, продуктив-ність	Часові ряди, статуси, проду-ктивність
Алертинг	Потужна система алертингу	Гнучка система алертингу	Підтри-мує алерти через Prometheus	Відсутня в Kibana, потрібно налашто-вувати окремо	Підтри-мує різні способи спові-щення	Просте сповіщення	Гнучка система сповіщень
Інтеграції	Різнома-нітні плагіни та API	Підтримує багато сторонніх інтеграцій	Висока, з підтрим-кою числе-нних джерел	Висока, інтеграція з багатьма джере-лами	Широкі можли-вості інтегра-цій	Обмежені, в основному для мережевих пристроїв	Велика кількість плагінів

Критерій	Zabbix	Prometheus	Grafana	ELK Stack	Ansible	SNMP	Nagios
Документація та спільнота	Гарна документ-тація, активна спільнота	Висока якість докумен-тації, активна спільнота	Гарна документ-тація, активна спільнота	Висока якість докумен-тації, активна спільнота	Хороша докумен-тація, активна спільнота	Обмежена, але доступна	Гарна докумен-тація, активна спільнота
Безпека	Підтри-мує аутенти-фікацію та шифру-вання	Підтримує шифру-вання та аутенти-фікацію	Залежить від джерела даних	Підтри-мує безпеку через TLS/SSL	Відсутні вбудовані механі-зми безпеки	Основні механізми безпеки	Основні механізми безпеки
Вартість	Безко-штовний, з можли-вістю комерцій-ної підтримки	Безко-штовний, з можли-вістю комерцій-ної підтримки	Безко-штовний, з можли-вістю комер-ційної підтрим-ки	Безко-штовний, з можли-вістю комер-ційної підтрим-ки	Безко-штовний, відкритий вихідний код	Безкоштовний, відкритий вихідний код	Безко-штовний, відкритий вихідний код
Кроспла-тформність	Підтри-мує Linux, Windows	Підтримує Linux, Windows	Підтри-мує Linux, Windows, MacOS	Підтри-мує Linux, Windows	Підтри-мує різні ОС	Підтримує всі основні ОС	Підтри-мує Linux, Windows
Гнучкість та кастомі-зація	Висока, через плагіни та API	Висока, через експорт метрик	Висока, через створення власних панелей	Висока, з можли-вістю налашту-вання	Дуже висока, можна налашту-вати під будь-які потреби	Обмежена, стандартні параметри	Висока, можна налашту-вати під потреби
Інтерфейс користу-вача	Інтуїтив-но зрозумі-лий, але застарі-лий	Сучасний, простий у викори-станні	Інтуїтив-но зрозумі-лий, сучасний	Інтуїтив-но зрозумі-лий, але може бути складним	Команд-ний рядок, немає GUI	Простий, але застарілий	Простий, але застарілий
Системні вимоги	Середні, залежить від кількості даних	Низькі, але зростають з кількістю даних	Низькі, залежить від джерела даних	Високі, потребує багато ресурсів	Низькі, залежить від виконуваних задач	Низькі, залежить від пристроїв	Середні, залежить від кількості даних

ЕТАПИ МЕТОДИКИ ТЕСТУВАННЯ

1. Планування тестування.

Визначення цілей тестування, що передбачає: перевірку правильності збору даних про конфігурації серверів; оцінку швидкості оновлення даних; тестування роботи системи в умовах навантаження; тестування точності сповіщень та їх своєчасності; аналіз стабільності роботи системи у тривалих сесіях.

Розробка тестових сценаріїв, що передбачає: тестування базових функцій (доступність серверів, моніторинг ресурсів); перевірка користувацьких плагінів для збору специфічних даних; аналіз поведінки системи під час помилок (збій сервера, недоступність мережі).

Визначення критеріїв успішності, що передбачає: час реагування на зміни; відсоток коректно отриманих даних; мінімальний вплив системи моніторингу на продуктивність серверів.

2. Підготовчі дії.

Розгортання тестового середовища, що передбачає: налаштування серверів різної конфігурації (операційні системи, апаратні параметри); встановлення Nagios Core та необхідних плагінів для моніторингу серверів; налаштування віддалених серверів через NRPE, SSH або SNMP.

Визначення набору тестів, що передбачає: збір базових метрик (стан процесора, пам'яті, дисків); моніторинг доступності серверів (ping, мережеві порти); тестування збору специфічних конфігурацій (версії ОС, програмного забезпечення).

Налаштування сповіщень, що передбачає: налаштування електронної пошти, SMS або месенджерів для отримання сповіщень про збої;

Інструменти для аналізу, що передбачає: використання логів Nagios для вивчення помилок.

Підключення зовнішніх інструментів (Grafana, Kibana), що передбачає аналіз продуктивності.

3. Налаштування середовища тестування.

Інфраструктура для тестування, що передбачає: встановлення серверу Nagios (основного вузла моніторингу); налаштування кількох серверів для моніторингу (фізичних, віртуальних); інтеграція з віддаленими вузлами через NRPE, SNMP або SSH.

Налаштування моніторингу, що передбачає: впровадження стандартних плагінів для моніторингу загальних параметрів; розробку користувацьких плагінів для перевірки конфігурації (наприклад, версій ОС, встановлених програм).

ЕТАПИ МЕТОДИКИ ТЕСТУВАННЯ

4. Виконання тестів (проведення різних видів тестування).

Функціональне тестування, що передбачає: перевірку кожного моніторингового сценарію, а саме: чи правильно система збирає та обробляє дані; тестування плагінів, а саме:: чи повертають вони очікувані результати; перевірку точності сповіщень у разі порушення параметрів (наприклад, при перевищенні використання CPU на 80%).

Тестування продуктивності, що передбачає: впровадження навантажувальних тестів шляхом додавання великої кількості хостів до системи (наприклад, 1000 серверів) чи збільшення частоти збору даних та оцінка часу відповіді системи під час роботи в умовах пікового навантаження.

Тестування відмовостійкості, що передбачає: імітацію збоїв (відключення мережі, збій служби); визначення швидкості реагування системи на збої.

Тестування масштабованості, що передбачає: додавання нових серверів для моніторингу в реальному часі; перевірка роботи у розподіленій архітектурі (з використанням Nagios Distributed Monitoring).

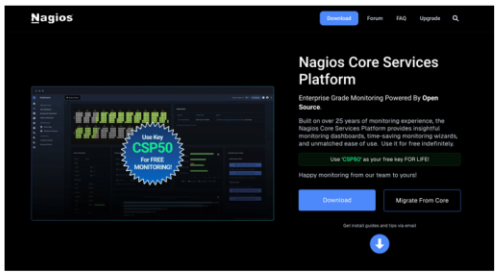


Рисунок 1 – Ядро Nagios

ІНТЕГРАЦІЯ WINDOWS-СИСТЕМИ В МОНІТОРИНГОВЕ СЕРЕДОВИЩЕ НА БАЗІ NAGIOS.

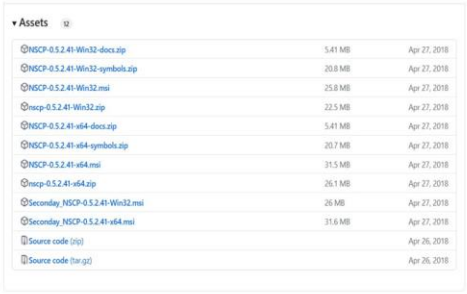


Рисунок 2 – Встановлення агента NSCP NG

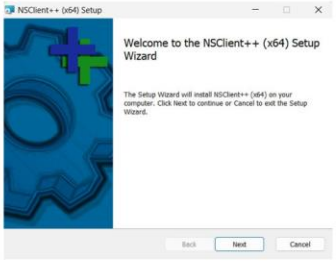


Рисунок 3 – Пакет NSClient.msi

Рисунок 4 – Конфігурація агента

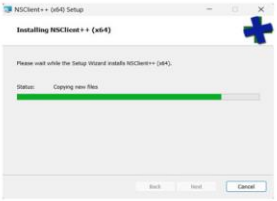


Рисунок 5 – Процес встановлення

Рисунок 6 – Включене правило на відкриття порту 12489

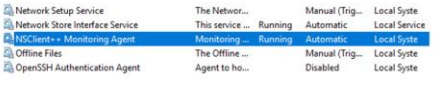


Рисунок 7 – Перезапуск служби

Рисунок 8 – Веб-консоль моніторингу конфігурації та активності серверу

3d 14h 22m 44s	1/4	OK - load average: 0.04, 0.03, 0.05
3d 14h 22m 6s	1/4	USERS OK - 1 users currently logged in
0d 0h 41m 5s	1/4	HTTP OK: HTTP/1.1 301 Moved Permanently - 456 bytes in 0.001 second response time
3d 14h 20m 51s	1/4	PING OK - Packet loss = 0%, RTA = 0.06 ms
3d 14h 20m 14s	1/4	DISK OK - free space: / 14602 MiB (95.18% inode=100%):
3d 14h 19m 36s	1/4	SSH OK - OpenSSH_7.4 (protocol 2.0)
3d 14h 18m 59s	1/4	SWAP OK - 100% free (8191 MB out of 8191 MB)
3d 14h 23m 21s	1/4	PROCS OK: 56 processes with STATE = RSZDT

Рисунок 9 – Продовження Веб-консоль моніторингу конфігурації та активності серверу

ТЕСТУВАННЯ СИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Рисунок 10 – Налаштування Firewall

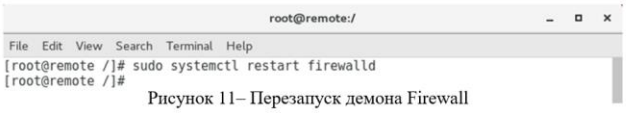


Рисунок 11– Перезапуск демона Firewall

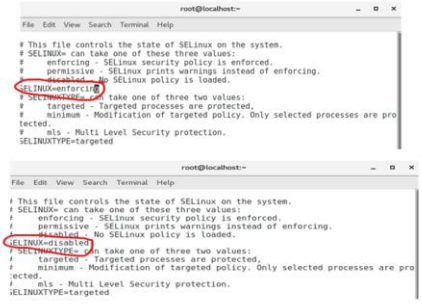


Рисунок 12 – Вимкнення параметру SELINUX

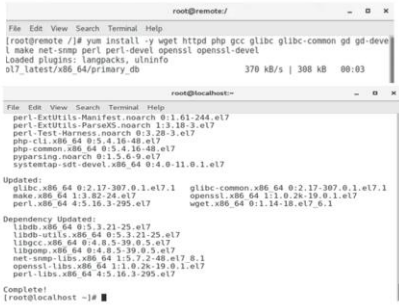


Рисунок 13– Завантаження залежностей з репозиторію

СКОНФІГУРОВАНИЙ СЕРВЕР

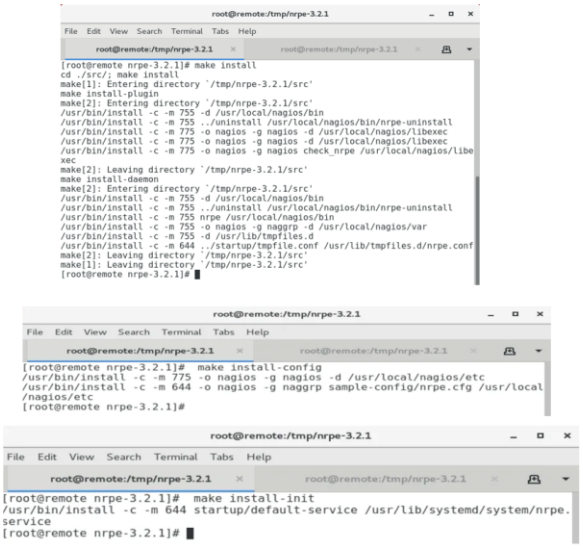


Рисунок 14 – Ініціалізація серверу

Рисунок 15 – Моніторинг серверу Nagios Core

ВИСНОВКИ

1. Проведено аналіз сучасних інструментів моніторингу серверного обладнання, що підтвердив ефективність Nagios.
2. Розроблено методику тестування системи на базі Nagios з акцентом на її багатоплатформність.
3. Встановлено, що Nagios забезпечує гнучке налаштування та інтеграцію з Windows- і Linux-системами.
4. Реалізовано систему сповіщень і звітування для оперативного реагування та планування.
5. Продемонстровано можливість автоматизованого збору даних і моніторингу ключових параметрів систем.
6. Забезпечено безпеку та розмежування доступу для захисту даних.

Апробація результатів надається в 1 статті, 2 тезисах в рецензованих наукових журналах і виданнях.

Стаття: Бураков М. М., Катков Ю. І. Критичні аспекти у системах автоматизованого збору даних про конфігурації серверного обладнання// Бураков М. М., Катков Ю.І.// Наукові записки Державного університету телекомунікацій №1, 2025, Подано до друку.
https://journals.dut.edu.ua/index.php/sciencenotes/issue/archive

Тезах:

Бураков М. М., Катков Ю. І. Проблеми безпеки систем автоматизованого збору даних про конфігурації серверного обладнання // науково-практична конференція «Актуальні проблеми безпеки інформаційно-телекомунікаційних систем» Збірник тез. – К.: ДУІКТ, 2024. 03 листопада 2024, С.180-182. https://duikt.edu.ua/uploads/p_2661_93669247.pdf

Бураков М. М., Катков Ю. І. Особливості систем автоматизованого збору даних про конфігурації серверного обладнання Hewlett Packard Enterprise // науково-практична конференція «Сучасні досягнення компанії Hewlett Packard Enterprise в галузі іт та нові можливості їх вивчення і застосування» - подане до збірнику тез. – К.: ДУІКТ, 2024. 14 грудня 2024, С.
https://duikt.edu.ua/ua/2661-konferencii-2024-roku-konferencii-ta-seminari