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

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «СИСТЕМА ВІДДАЛЕНОГО МОНІТОРИНГУ СТАНУ КОМП'ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМСТВА»

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

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

_____ *Роман КОНДРАТЮК*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач(ка) вищої освіти гр. КІД-42
Роман КОНДРАТЮК
Ім'я, ПРІЗВИЩЕ

Керівник: Анастасія ВЄЧЕРКОВСЬКА
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

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

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кондратюк Роман Юрійович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Система віддаленого моніторингу стану комп'ютерної мережі підприємства»
керівник кваліфікаційної роботи: Анастасія ВЄЧЕРКОВСЬКА, кандидат технічних наук, доцент,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. №56
2. Строк подання кваліфікаційної роботи «05» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, стандарти та протоколи моніторингу мереж (SNMP, ICMP, NetFlow), архітектура та компоненти корпоративних мереж, методи збору й візуалізації телеметричних даних, системи сповіщення про збої, порівняльний аналіз систем моніторингу, інструменти тестування навантаження та працездатності.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
    1. Теоретичні основи моніторингу комп'ютерних мереж
    2. Практична реалізація системи моніторингу
    3. Перспективи вдосконалення системи моніторингу
5. Перелік ілюстративного матеріалу: *презентація*
    1. Мета, об'єкт, предмет дослідження
    2. Актуальність роботи
    3. Класифікація та задачі систем моніторингу
    4. Архітектура системи віддаленого моніторингу
    5. Практична реалізація системи моніторингу у VMware
    6. Забезпечення віддаленого доступу до моніторингу
    7. Власний дашборд для перевірки ключових метрик
    8. Перевірка системи на виявлення аномалій (відхилень)
    9. Методи масштабування та інтеграції
    10. Порівняльний аналіз швидкості розгортання існуючих рішень

11. Кошторис реалізації та безкоштовне рішення
12. Висновки
13. Публікації та апробація роботи

6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-19.03.2025	
2	Обробка матеріалу	20.03-02.04.2025	
3	Розробка теоретичної частини	03.04-07.04.2025	
5	Практична реалізація системи моніторингу	08.04-27.04.2025	
6	Перспективи вдосконалення системи моніторингу	28.04-11.05.2025	
7	Висновки за результатами аналізу	12.05-15.05.2025	
8	Розробка презентації	16.05-26.05.2025	
9	Передзахист	27.05-29.05.2025	
10	Здача в деканат	30.05-05.06.2025	

Здобувач(ка) вищої освіти _____ Роман КОНДРАТЮК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Анастасія ВЄЧЕРКОВСЬКА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 96 стор., 46 рис., 7 табл., 32 джерел.

Мета роботи – спрощення розгортання та підвищення ефективності системи віддаленого моніторингу стану комп'ютерної мережі підприємства за рахунок використання відкритих інструментів, централізованої архітектури та модулів візуалізації.

Об'єкт дослідження – процеси контролю та забезпечення стабільної роботи комп'ютерної мережі підприємства.

Предмет дослідження – технології моніторингу, аналітики та оповіщення, що використовуються для побудови простих у впровадженні систем моніторингу на базі відкритого ПЗ.

Короткий зміст роботи: У роботі розглянуто принципи та підходи до побудови систем моніторингу для локальних і корпоративних мереж, зокрема використання інструмента Netdata на сервері з Ubuntu Server. Підкреслено переваги прямого розгортання без Docker для швидкої демонстрації можливостей.

Реалізовано базову систему моніторингу з візуалізацією навантаження CPU, мережі, дисків тощо, а також розглянуто підключення агентів для масштабування та зовнішніх баз даних для довготривалого зберігання метрик. Запропоновано способи підвищення захищеності доступу (TLS/SSL), а також механізми покращення ефективності шляхом додавання модулів оповіщення. Результати демонструють ефективність використання Netdata у малих і середніх IT-інфраструктурах.

КЛЮЧОВІ СЛОВА: МЕРЕЖА ПІДПРИЄМСТВА, МОНІТОРИНГ, NETDATA, ІНФРАСТРУКТУРА, ТЕЛЕМЕТРІЯ, ЗБІР МЕТРИК, ВІЗУАЛІЗАЦІЯ, SPoG, ВЕБ-ІНТЕРФЕЙС, ЦЕНТРАЛІЗАЦІЯ, СПОВІЩЕННЯ, TLS, OPEN SOURCE.

ABSTRACT

Text part of the qualification work for obtaining a bachelor's degree: 96 pages, 46 figures, 7 tables, 32 sources.

The purpose of the work is to simplify the deployment and increase the efficiency of a remote monitoring system for the state of an enterprise's computer network through the use of open tools, centralized architecture and visualization modules.

The object of the study is the processes of controlling and ensuring the stable operation of an enterprise's computer network.

The subject of the study is is monitoring, analytics and alerting technologies used to build easy-to-implement monitoring systems based on open software.

Summary of the work: The paper considers the principles and approaches to building monitoring systems for local and corporate networks, in particular, using the Netdata tool on a server with Ubuntu Server. The advantages of direct deployment without Docker for a quick demonstration of capabilities are emphasized.

A basic monitoring system with visualization of the CPU, network, disk load, etc. is implemented, and the connection of agents for scaling and external databases for long-term storage of metrics is also considered. Methods for increasing access security (TLS/SSL), as well as mechanisms for improving efficiency by adding alerting modules, are proposed. The results demonstrate the effectiveness of Netdata in small and medium IT infrastructures.

KEYWORDS: ENTERPRISE NETWORK, MONITORING, NETDATA, INFRASTRUCTURE, TELEMETRY, METRIC COLLECTION, VISUALIZATION, SPoG, WEB INTERFACE, CENTRALIZATION, NOTIFICATIONS, TLS, OPEN SOURCE.

ЗМІСТ

ВСТУП..... 9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ КОМП'ЮТЕРНИХ МЕРЕЖ 10
1.1 Поняття моніторингу в комп'ютерних мережах..... 10
1.2 Принципи побудови систем моніторингу..... 16
1.3 Інструменти та платформи для моніторингу мереж..... 25
1.4 Метрики та параметри, що підлягають моніторингу 34
1.5 Технічні вимоги до побудови системи моніторингу. 42
РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ. 51
2.1 Архітектура впроваджуваної системи 51
2.2 Побудова та конфігурування моніторингового середовища..... 56
2.3 Алгоритми обробки та відображення даних 64
2.4 Реалізація системи на прикладі тестового середовища 73
РОЗДІЛ 3. ПЕРСПЕКТИВИ ВДОСКОНАЛЕННЯ СИСТЕМИ МОНІТОРИНГУ. 81
3.1 Методи покращення ефективності системи..... 81
3.2 Порівняння обраного рішення з альтернативами 89
3.3 Можливості розширення у майбутньому 97
ВИСНОВКИ..... 104
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ……...…….105
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)…….....108

ВСТУП

Сучасні комп’ютерні мережі є критичним елементом інфраструктури будь-якого підприємства, оскільки забезпечують обмін даними між пристроями, доступ до сервісів та ресурсів, а також безперервну роботу корпоративних інформаційних систем. Зі зростанням обсягів переданої інформації та кількості підключених пристроїв виникає потреба в постійному контролі стану мережевих елементів для вчасного реагування на відхилення в роботі. Втрата доступності або зниження швидкодії мережі може призвести до простоїв, фінансових втрат або компрометації даних.

Моніторинг стану мережі дозволяє здійснювати збір, обробку та аналіз технічних параметрів мережевого обладнання й серверної інфраструктури в реальному часі. Завдяки цьому можна своєчасно виявляти збої, прогнозувати навантаження та оптимізувати використання ресурсів. У сучасних системах моніторингу використовуються принципи телеметрії, централізованого збору метрик, побудови дашбордів і інтеграції з модулями сповіщення, що забезпечує прозорість роботи мережі та підвищує її керованість.

У практиці корпоративного IT-управління поширеним є використання відкритих інструментів, які забезпечують широкий функціонал без ліцензійних витрат. Одним із таких рішень є система Netdata, що дозволяє розгорнути легкий та зручний у користуванні веб-інтерфейс моніторингу з можливістю масштабування. Водночас для ефективного впровадження подібних систем необхідно враховувати технічні вимоги, можливості захисту доступу, способи інтеграції з іншими інструментами, а також особливості інфраструктури підприємства.

З огляду на вищезазначене, постає задача побудови гнучкої та простої у розгортанні системи моніторингу, яка дозволяє не лише візуалізувати стан комп’ютерної мережі підприємства, але й забезпечує основу для подальшого масштабування, централізації та інтеграції з інструментами аналітики.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ КОМП'ЮТЕРНИХ МЕРЕЖ

1.1 Поняття моніторингу в комп'ютерних мережах

Моніторинг комп'ютерних мереж являє собою процес безперервного спостереження за їхнім станом, продуктивністю, навантаженням та активністю усіх елементів інфраструктури з метою вчасного виявлення збоїв, аномалій чи потенційних загроз, які можуть негативно вплинути на роботу системи в цілому. В умовах сучасних підприємств, де ефективність функціонування інформаційних систем є основою стабільності бізнес-процесів, наявність дієвої системи моніторингу вже не є опціональною, а виступає обов'язковим компонентом безпечної та надійної експлуатації мережевої інфраструктури. З огляду на те, що інформаційні мережі пронизують майже всі процеси сучасної організації, навіть короткочасний збій здатен призвести до серйозних наслідків, тому завдання моніторингу полягає не лише в зборі даних, а й у забезпеченні можливості для оперативного реагування на будь-які відхилення від штатної роботи. Своєчасне повідомлення адміністратора про наявність несправностей, зниження продуктивності чи підвищене навантаження дає змогу мінімізувати простої, запобігти втраті даних, а також попередити загострення інцидентів до критичного рівня. У рамках моніторингу можуть фіксуватися такі параметри як:

- перевантаження мережі;
- втрата пакетів;
- дефіцит ресурсів (оперативної пам'яті, процесорного часу, або виникнення позаштатних ситуацій, пов'язаних зі збоями обладнання чи спробами несанкціонованого доступу).

Дані, які збираються в процесі моніторингу, не лише відображають поточний стан системи, а й стають джерелом для аналітики, на основі якої можна формувати

прогнози щодо поведінки мережі в майбутньому, виявляти «вузькі місця», а також планувати масштабування або оновлення окремих вузлів. Таким чином, моніторинг трансформується із засобу реакції на проблеми у стратегічний інструмент управління інфраструктурою, що дає змогу приймати обґрунтовані рішення щодо розвитку IT-середовища, що особливо актуально в умовах постійного зростання складності мереж, збільшення кількості підключених пристроїв та необхідності дотримання високих стандартів безперервності бізнесу [1-4].

Класифікація типів моніторингу. З практичної точки зору, адміністрування комп'ютерних мереж передбачає використання різних типів моніторингу, які можна класифікувати за способом збору даних. Активний моніторинг базується на ініціюванні запитів до пристроїв або сервісів, щоб отримати інформацію про їхній стан, доступність та швидкодію, що можуть бути ICMP-перевірки (ping), HTTP-звернення, DNS-запити або SNMP-опитування, які дозволяють швидко виявляти втрату доступу, затримки або інші ознаки деградації сервісу. Основним недоліком цього підходу може бути додаткове навантаження на мережу, особливо у великих інфраструктурах з частими циклами опитування. Натомість пасивний моніторинг ґрунтується на аналізі вже наявного трафіку та подій, що фіксуються у логах, або через механізми на кшталт mirror-портів комутаторів. Він не створює додаткового навантаження, але може не виявити проблему, якщо сам пристрій не генерує повідомлень про несправності [1-2].

Існує також проміжний підхід — гібридний моніторинг, який поєднує активні й пасивні методи збору інформації, дозволяючи досягти балансу між повнотою спостереження та мінімізацією впливу на мережу (див. рис. 1.1).

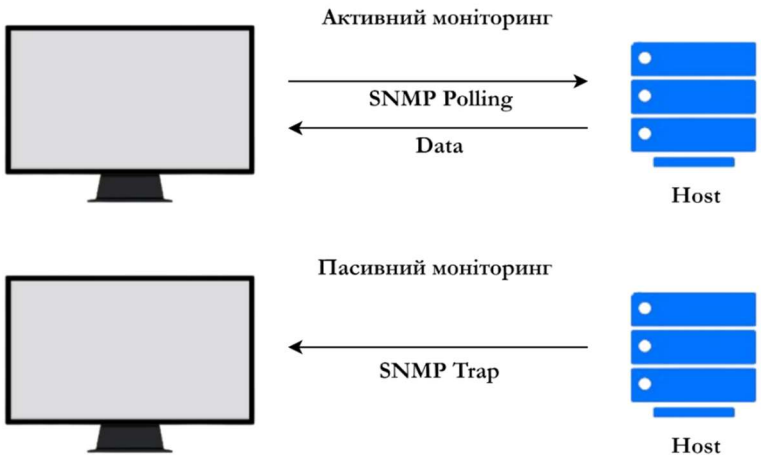


Рисунок 1.1 – Класифікація типів моніторингу в комп'ютерних мережах

У більшості сучасних рішень реалізується саме такий підхід, що забезпечує достатній рівень інформованості адміністратора при раціональному використанні ресурсів системи.

Вибір між цими типами моніторингу залежить від:

— особливостей конкретної інфраструктури;

— рівня її критичності;

— типів пристроїв;

— програмного забезпечення.

У деяких випадках пріоритетом буде швидке виявлення збоїв, в інших — безперервний ненав'язливий контроль, тому оптимальна конфігурація визначається лише в контексті реального середовища [1-2].

Призначення та основні задачі систем моніторингу. Системи моніторингу в контексті сучасних комп'ютерних мереж мають чітко визначене призначення, яке полягає у забезпеченні постійного спостереження за станом інфраструктури з метою її стабільної, передбачуваної та ефективної роботи. У межах такого підходу формується низка завдань, що реалізуються поетапно і охоплюють усі аспекти функціонування мережевого середовища. Передусім важливим є оперативне виявлення несправностей, коли за допомогою автоматизованих механізмів фіксуються відмови обладнання, збої у роботі серверів, маршрутизаторів, точок

доступу, падіння сервісів, розриви каналів зв’язку чи порушення функціонування окремих інтерфейсів, і саме це дозволяє вчасно локалізувати проблему та не допустити її подальшого поширення (див. рис. 1.2) [1, 3].



Рисунок 1.2 – Основні задачі систем моніторингу

Наступним логічним кроком є аналіз продуктивності, який передбачає оцінку навантаження на окремі компоненти мережі, включно з каналами зв’язку, процесорами, оперативною пам’яттю чи дисковими підсистемами, що дозволяє визначити необхідність масштабування інфраструктури або усунути вузькі місця (так звані "bottlenecks"), які можуть обмежувати загальну ефективність. Якщо використовується застаріле чи недостатньо потужне обладнання, то без аналізу продуктивності існує ризик виникнення проблем, які в подальшому унеможливлять нормальну роботу всієї мережі. На основі зібраних метрик формується звітність та накопичуються історичні дані, що дає змогу:

— відстежувати тренди;
— перевіряти ефективність прийнятих технічних рішень;
— оцінювати динаміку навантаження;
— ухвалювати обґрунтовані рішення щодо подальших змін у структурі або конфігурації систем.

До наступного етапу відноситься реалізація сповіщень та реагування, коли у разі виявлення аномалій або досягнення критичних значень показників система автоматично надсилає повідомлення через різноманітні канали зв’язку, включаючи

електронну пошту, месенджери або SMS, що скорочує час реагування та зменшує необхідність постійного ручного моніторингу. Додатково система може підтримувати контроль відповідності умовам, зафіксованим у SLA (service level agreement), тобто угодах про рівень обслуговування, зокрема стосовно гарантій доступності або швидкості відновлення сервісів, що дозволяє забезпечувати надання послуг на належному рівні. На завершальному етапі варто виокремити функції, пов’язані з безпекою, зокрема виявлення загроз та вторгнень, що реалізується шляхом фіксації нетипової активності в мережі, наприклад спроб сканування чи проникнення, і може інтегруватися із засобами SIEM-класу, які обробляють дані для подальшого аналізу в контексті інформаційної безпеки. Усе це разом формує систему, яка не просто демонструє поточний стан ресурсів, а слугує інструментом як для оперативного управління, так і для стратегічного планування та розвитку IT-інфраструктури [1, 3].

Сфери застосування в корпоративному середовищі. Застосування систем моніторингу є актуальним для широкого кола організацій, зокрема в корпоративному секторі, де стабільність, безперервність і безпека цифрових ресурсів мають критичне значення. Одним з найбільш типових прикладів є внутрішні IT-відділи підприємств, де за допомогою моніторингу здійснюється контроль за станом серверів, робочих станцій, принтерів, мережевого обладнання, що дозволяє зменшити навантаження на технічний персонал і звести до мінімуму можливі простої. У центрах обробки даних (ЦОД) моніторинг є невід’ємною складовою керування великою кількістю серверів, комутаторів, енергетичних модулів і систем охолодження, адже навіть незначний збій у такому середовищі може мати серйозні наслідки для бізнесу чи державного сектора. Інтернет-провайдери також покладаються на системи моніторингу для забезпечення стабільності магістральних ліній, обладнання доступу та підключень клієнтів, що особливо важливо для забезпечення високої якості послуг, недопущення перевантаження каналів і швидкого реагування на інциденти (див. рис. 1.3) [3-4].



Рисунок 1.3 – Сфери застосування моніторингу в корпоративних мережах

У хмарних платформах і сервісах типу SaaS чи IaaS використання моніторингу дозволяє гарантувати доступність інстанцій, API, баз даних, контейнерів, а також ефективно масштабувати інфраструктуру в умовах зростання навантаження, запобігаючи падінню сервісів. У сфері промисловості, особливо з огляду на розвиток концепції Industry 4.0, моніторинг охоплює не тільки IT-інфраструктуру, а й виробниче обладнання — контролери, сенсори, системи SCADA та PLC, де безперервне спостереження дозволяє підтримувати стабільність і безпеку виробничих процесів. В освітніх і медичних установах системи моніторингу забезпечують контроль за доступом до критичних сервісів, навчальних платформ або медичних баз даних, де збої можуть мати значні наслідки, попри менші масштаби інфраструктури порівняно з провайдерами чи дата-центрами [3-4].

У підсумку, у всіх перерахованих випадках моніторинг виконує функцію діагностичного та аналітичного центру, який дає змогу підтримувати стабільну роботу систем, попереджати проблеми на ранніх етапах та забезпечувати відповідність вимогам як внутрішніх політик, так і зовнішніх стандартів. У разі

відсутності подібних рішень у корпоративному середовищі зростає ризик виникнення помилок, перенавантаження, втрат продуктивності та зниження якості обслуговування, що зрештою впливає як на ефективність функціонування підприємства, так і на його конкурентоспроможність.

1.2 Принципи побудови систем моніторингу

Побудова ефективної системи моніторингу комп'ютерної мережі вимагає:

— глибокого усвідомлення архітектурних рішень;

— протоколів обміну даними;

— вимог до безпеки.

Усе це повинно бути продумано та реалізовано на всіх рівнях. Кожен окремий компонент у такій системі впливає на її масштабованість, стабільність, зручність користування та подальший розвиток, оскільки невдале технічне рішення навіть на одному з етапів здатне нівелювати ефективність усього комплексу. Реалізація усіх ланок у чіткій взаємодії одна з одною дозволяє системі не просто фіксувати стан інфраструктури, а й повністю відповідати своїм функціональним завданням, виконуючи саме ту роль, заради якої вона впроваджується, адже інакше така система втрачає сенс свого існування [5-8].

Основні елементи систем моніторингу. Моніторингова система складається з низки взаємопов'язаних апаратних та програмних компонентів, які разом забезпечують:

— збір;

— збереження;

— обробку;

— стан мережі та її окремих вузлів.

Одним із перших елементів у такій структурі виступає агент моніторингу — програмний модуль, встановлений безпосередньо на цільовому обладнанні, такому

як сервери, маршрутизатори чи робочі станції. Його завданням є збір локальної інформації: завантаження процесора, обсяг спожитої оперативної пам’яті, активність мережевих інтерфейсів, події в системному журналі, загальний стан пристрою. Існують активні агенти, що самостійно передають дані на сервер, та пасивні, які лише відповідають на запити — найпоширенішими серед них є такі рішення як Zabbix Agent, Netdata Agent, Telegraf, Collectd [5].

Наступним логічним компонентом є сервер збору або колектор — центральний вузол, який приймає потоки даних від агентів або безпосередньо з мережі, обробляє отриману інформацію, реагує на задані умови, активує тригери й оповіщення, розраховує агреговані метрики. Він може виступати або як процесор даних, або лише як маршрутизатор, який перенаправляє зібрану телеметрію до бази даних. Саме сервер формує зв’язок між сирими технічними показниками й тими сигналами чи подіями, що зумовлюють дії оператора або автоматизованої системи реагування [5].

Уся отримана інформація надалі зберігається в базах даних метрик, які спеціалізуються на обробці часових рядів — даних із позначкою часу, що мають високий рівень оновлення та значний обсяг. До таких баз належать InfluxDB, Prometheus, TimescaleDB, RRD, VictoriaMetrics та інші. Вони оптимізовані для запису, агрегації, стиснення та оперативного доступу до великого обсягу числових значень, що надходять у реальному часі, і без подібного сховища моніторингова система не здатна ані аналізувати тенденції, ані формувати звітність чи забезпечувати історичну ретроспективу [5].

Завершальною ланкою в цьому ланцюгу є засоби візуалізації, які витягують дані з баз метрик і перетворюють їх у зручні для сприйняття інтерфейси — графіки, гістограми, дашборди, веб-панелі, що дозволяють аналітику чи адміністратору швидко оцінити поточний стан інфраструктури. Саме вони забезпечують найзрозуміліший контакт між системою та людиною, оскільки наочне представлення SLA, сповіщень і загальних характеристик дозволяє не лише контролювати стан мережі, а й швидко виявляти аномалії чи критичні події. Найбільш відомими серед таких засобів є Grafana, Kibana, Zabix Frontend, Netdata

Dashboard, кожен із яких дозволяє налаштувати персоналізовані панелі відповідно до конкретних потреб та особливостей інфраструктури (див. рис. 1.4) [5].

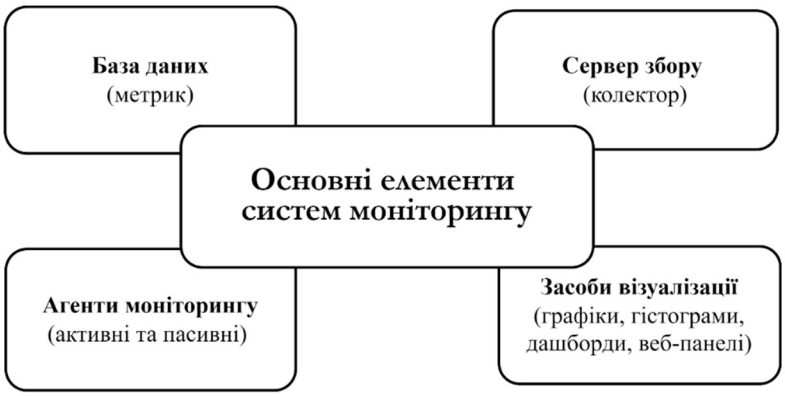


Рисунок 1.4 – Компоненти типової системи моніторингу

Кожен елемент системи, від агента на віддаленому хості до засобу візуалізації в центрі обробки даних, тісно пов’язаний з іншими компонентами, утворюючи цілісну екосистему моніторингу, у якій втрата або відсутність хоча б одного компонента знецінює ефективність усього комплексу. Без агента немає даних, без сервера — логіки, без бази — історії, без інтерфейсу — взаємодії. Саме тому повна реалізація усіх основних ланок дозволяє говорити про наявність справжньої системи моніторингу, яка не лише фіксує, але й аналізує, попереджає, забезпечує стабільність та передбачуваність у роботі всієї мережевої інфраструктури.

Архітектурні підходи – централізовані та розподілені моделі. Під час побудови будь-якої системи моніторингу важливо одразу визначити відповідну архітектуру, яка залежить від розміру підприємства, кількості вузлів, географічного розташування та рівня навантаження на інфраструктуру, оскільки саме ці фактори ще на етапі проєктування впливають на потенціал і стабільність подальшої роботи системи. Серед основних архітектурних підходів виділяють:

— централізовану модель;

— розподілену модель.

У централізованій архітектурі всі агенти передають дані на єдиний сервер збору, який відповідає за зберігання інформації у базі даних і забезпечення інтерфейсу доступу, при цьому перевагами такої моделі є:

— простота реалізації;

— легкість супроводу;

— зручність аналізу в одному центрі;

— нижчі витрати на обладнання.

Однак у такій конфігурації виникають критичні недоліки: наявність єдиної точки відмови, обмежене масштабування й перевантаження центрального вузла при зростанні кількості під’єднаних пристроїв (див. рис. 1.5).

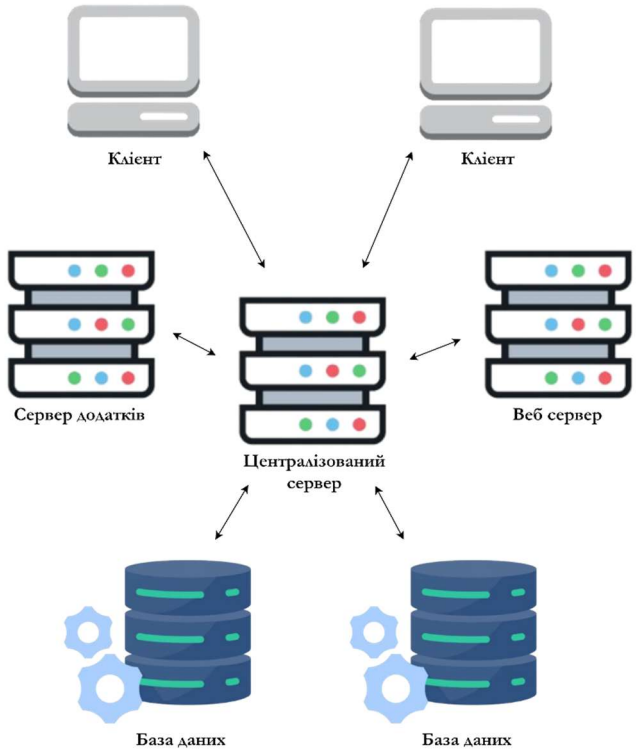


Рисунок 1.5 – Централізована модель моніторингу

У противагу цьому, розподілений підхід ґрунтується на використанні кількох серверів збору — так званих локальних хабів, які обробляють трафік на місцевому рівні й передають узагальнені або агреговані дані до центрального сховища, що дозволяє знизити міжрегіональний трафік, зменшити затримки й забезпечити

стабільну роботу навіть у випадку виходу з ладу окремих елементів. До переваг розподіленої моделі належать:

— краща масштабованість;

— вища відмовостійкість;

— можливість реалізації локального моніторингу.

Хоча ця архітектура значно складніша у налаштуванні й вимагає постійної синхронізації між серверами, що ускладнює підтримку (див. рис. 1.6).

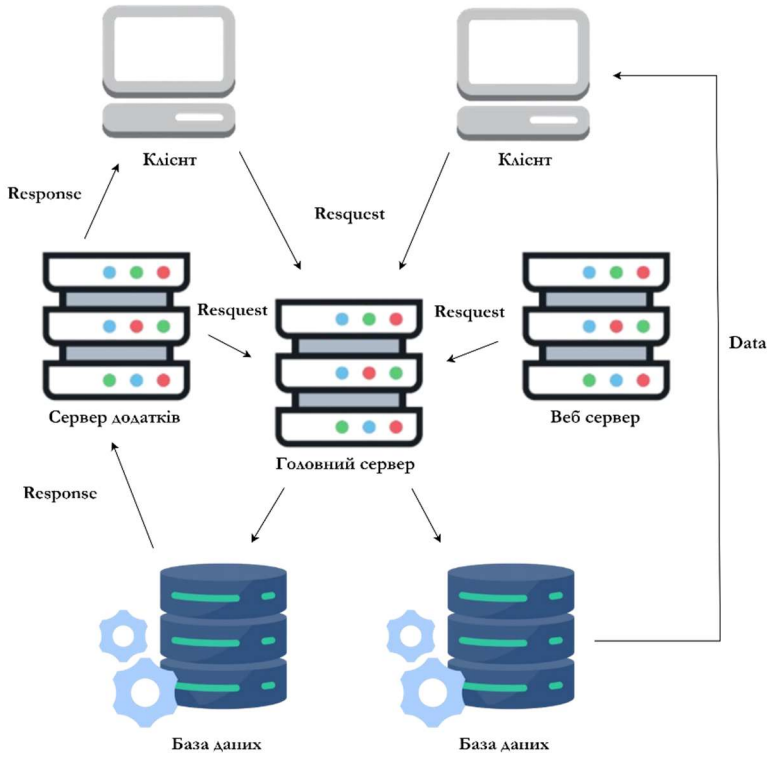


Рисунок 1.6 – Розподілена модель моніторингу

На практиці великі організації рідко використовують лише один з підходів, натомість застосовується гібридна модель, яка поєднує централізований аналіз з локальним збором і резервуванням даних — такий симбіоз дозволяє об’єднати зручність централізованого доступу з надійністю й гнучкістю розподіленої інфраструктури [6].

Використання протоколів у моніторингу. Окрім архітектури, важливим технічним аспектом для реалізації системи моніторингу є вибір протоколів, що

забезпечують комунікацію між пристроями, агентами і серверами збору даних. Одним із найстаріших і водночас найпоширеніших є SNMP (Simple Network Management Protocol), який підтримується практично всіма мережевими пристроями — маршрутизаторами, комутаторами, точками доступу — і дозволяє зчитувати інтерфейсні метрики, трафік, температуру та інші показники, а також надсилати сповіщення (SNMP traps) чи навіть частково змінювати налаштування обладнання. Протокол має декілька версій: версії 1 та 2c вважаються простими, але вразливими, натомість версія 3 забезпечує шифрування, автентифікацію та вищий рівень безпеки, що робить її кращим вибором у сучасних мережах. Іншим важливим протоколом є ICMP (Internet Control Message Protocol), що використовується для визначення доступності вузлів — наприклад, через команду ping можна швидко перевірити, чи пристрій працює і який рівень затримки, проте можливості ICMP обмежені, а в деяких випадках його пакети можуть блокуватися фаєрволами, що знижує ефективність. Щодо моніторингу трафіку використовуються протоколи NetFlow, sFlow та IPFIX, які дозволяють фіксувати потоки даних через маршрутизатори й детально аналізувати мережеву активність за IP-адресами, портами та протоколами, що, зокрема, корисно для виявлення аномалій, перевантажень або кібератак. Ці протоколи підтримуються багатьма пристроями, а sFlow, наприклад, є менш прив’язаним до конкретного виробника, на відміну від класичного NetFlow, який був розроблений Cisco (див. рис. 1.7) [7].

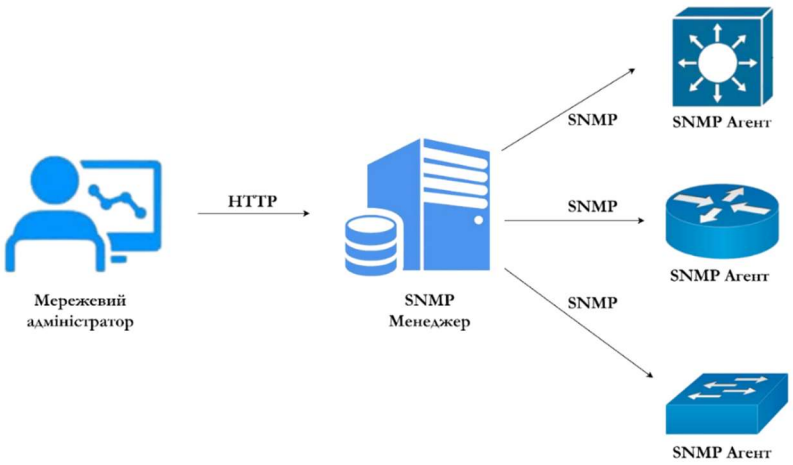


Рисунок 1.7 – Схема використання SNMP у мережевому моніторингу

Для роботи з сенсорними пристроями або у середовищах з низьким споживанням ресурсів доцільно використовувати MQTT (Message Queuing Telemetry Transport) — легкий протокол, який оптимізовано для передавання даних у реальному часі в умовах нестабільного з'єднання, що робить його ідеальним для IoT-систем, де сенсори надсилають короткі повідомлення безперервно або за подіями. У сучасних системах моніторингу також все частіше використовуються веб-протоколи — HTTP, REST API, WebSocket — які дозволяють інтегрувати системи збору з іншими сервісами або фронтендами, забезпечуючи гнучкість і можливість взаємодії за допомогою стандартних веб-запитів. Наприклад, Netdata або Prometheus можуть взаємодіяти з іншими інструментами через HTTP API для експорту або отримання даних. Вибір конкретних протоколів визначається типом обладнання, вимогами до безпеки та обраною архітектурною моделлю, тому у більшості випадків доцільно комбінувати кілька протоколів одночасно, щоб досягти комплексного моніторингу, забезпечити надійність збору й зменшити сліпі зони в аналітиці (див. рис. 1.8) [7].

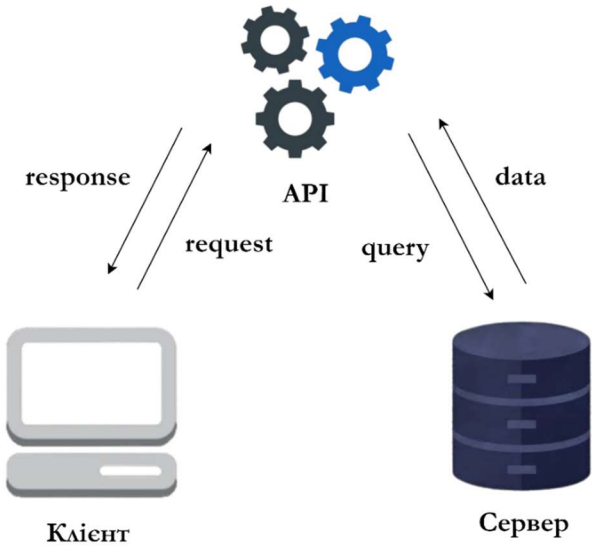


Рисунок 1.8 – Взаємодія агент-сервер через HTTP API

Безпека систем моніторингу. Безпека систем моніторингу є надзвичайно важливою складовою їх побудови, адже такі системи мають безпосередній доступ до конфіденційної інформації, даних про стан мережі та інтерфейсів керування

інфраструктурою підприємства. Серед найбільш критичних загроз, які можуть виникнути в таких умовах, можна відзначити:

- несанкціонований доступ до панелі керування;
- можливість перехоплення мережевого трафіку (особливо SNMPv2c);
- підміну агентів або даних;
- компрометацію вузлів;
- DDoS-атаки (на сервери збору інформації).

Всі ці загрози вказують на необхідність впровадження надійного захисту для зменшення ризиків і підтримання стабільної роботи системи, адже за відсутності належного захисту навіть незначна вразливість може бути використана зловмисником для проникнення в інфраструктуру та виведення з ладу всієї мережі [8].

Зменшення або повне усунення вказаних ризиків досягається за допомогою комплексного впровадження низки базових заходів безпеки, серед яких першочергове значення мають:

- автентифікація та авторизація;
- шифрування трафіку;
- сегментація мережі;
- журналювання і аудит;
- своєчасне оновлення;
- технічна підтримка.

Забезпечення автентифікації та авторизації користувачів передбачає обов’язкове використання захищених механізмів входу, таких як LDAP, багатофакторна автентифікація, а також реалізацію принципу мінімальних привілеїв, за якого користувачеві надається лише той обсяг доступу, що є дійсно необхідним для виконання його обов’язків. Надмірні права, наприклад адміністратора чи root-доступ, якщо вони не виправдані, можуть призвести до критичних порушень у роботі системи внаслідок людської помилки або зловмисної дії, тому мінімізація привілеїв — це фундаментальне правило кібербезпеки [8].

Не менш важливою складовою захисту є шифрування переданих даних, яке забезпечується, зокрема, шляхом використання TLS для HTTP-запитів, переходом на SNMPv3 замість небезпечних попередніх версій, а також впровадженням VPN або захищених тунелів між агентами та сервером збору даних. Такий підхід унеможливлює перехоплення трафіку і гарантує цілісність та конфіденційність інформації, що передається всередині системи моніторингу. Наступним кроком у підвищенні рівня безпеки є сегментація мережі, яка передбачає фізичне або логічне розділення інфраструктури на окремі VLAN або виділення серверів моніторингу у DMZ-зони, що дозволяє значно зменшити площу атаки. Обмеження доступу до портів лише з певних IP-адрес також допомагає запобігти несанкціонованим підключенням та забезпечує додатковий рівень контролю над потоками інформації у межах мережі [8].

Журналювання та аудит становлять ще одну критичну функцію, що забезпечує прозорість дій користувачів і виявлення підозрілої активності, оскільки фіксуються всі спроби доступу, зміни в конфігурації, неочікувані дії адміністраторів чи системних компонентів. Завдяки інтеграції з системами SIEM або іншими засобами безпеки з’являється можливість не лише реагувати на інциденти, але й проактивно виявляти аномальні патерни поведінки, що можуть свідчити про початок атаки. При цьому слід враховувати, що хоча деякі системи моніторингу автоматично включають функції журналювання, у багатьох випадках потрібне ручне налаштування, що вимагає глибшого розуміння інструментів і постійного контролю [8].

Завершальним, але не менш значущим заходом є регулярне оновлення як агентів, так і серверів моніторингу, що дозволяє вчасно усувати відомі вразливості та підтримувати систему в актуальному стані. Ігнорування цього аспекту рівносильне використанню застарілої операційної системи, яка вже не підтримується, що створює численні ризики та загрози. Незалежно від того, чи здається система стабільною на перший погляд, без своєчасних оновлень вона неминуче перетворюється на слабку ланку всієї інфраструктури. Тому оновлення та підтримка не є елементами, на яких можна зекономити чи тимчасово

відмовитися, — це обов’язкова частина будь-якої безпечної архітектури (див. рис. 1.9) [8].



Рисунок 1.9 – Механізми захисту інформації в системах моніторингу

У підсумку, реалізація комплексних заходів кібербезпеки в системах моніторингу не просто забезпечує їх захищеність, але й підвищує надійність і довговічність самої мережевої інфраструктури. Недооцінка ризиків або відсутність належної реалізації хоча б одного з описаних елементів може перетворити інструмент моніторингу з засобу захисту на потенційний вектор атаки, що є абсолютно неприйнятним у корпоративному середовищі.

1.3 Інструменти та платформи для моніторингу мереж

У сучасній комп’ютерній мережі підприємства критично важливим є забезпечення:

— постійного нагляду за станом її компонентів;

— вчасний аналіз продуктивності;

— оперативне реагування на відхилення.

Через зростання складності інфраструктури й необхідність високої надійності, дедалі частіше впроваджуються готові спеціалізовані платформи моніторингу, які забезпечують автоматизований:

— збір метрик;
— сповіщення про помилки;
— візуалізацію даних у реальному часі;
— можливість історичного аналізу.

Саме тому виникає необхідність детального аналізу найпопулярніших інструментів, які використовуються в корпоративному і приватному секторах, із фокусом на їхні функціональні можливості, підходи до впровадження, вимоги до інфраструктури та доцільність використання локальних або хмарних рішень. Особливу увагу в цьому контексті варто зосередити на Netdata як на легкому для розгортання інструменті, який підходить для малого та середнього бізнесу завдяки поєднанню простоти конфігурації, мінімальних вимог до системних ресурсів і широких можливостей візуального моніторингу [9-12].

Огляд популярних систем моніторингу. Ринок систем моніторингу вже давно сформував набір рішень, які стали стандартом і активно використовуються у різних масштабах — від невеликих офісних мереж до розподілених інфраструктур великих підприємств. Серед таких інструментів найчастіше згадуються:

— Zabbix;
— Nagios;
— Prometheus;
— Grafana;
— Netdata.

Кожен з них має свої підходи до архітектури та конфігурації, відтак і специфіку застосування залежно від цілей, ресурсів і масштабів мережі. Zabbix представляє собою комплексну централізовану систему моніторингу, яка здатна відслідковувати як мережеві пристрої через SNMP та IPMI, так і сервери або сервіси за допомогою агентів (див. рис. 1.10).



Рисунок 1.10 – Зовнішній вигляд дашборду Zabbix

Характеризується гнучкою системою тригерів і шаблонів, що дозволяє формувати складні логічні умови для сповіщення, однак потребує налаштування окремого сервера бази даних, як-от MySQL чи PostgreSQL, і доволі детального конфігурування, яке може виявитися непростим для початкового впровадження. У той же час Nagios — один із найстаріших представників цього класу — вирізняється гнучкістю та розгалуженою екосистемою плагінів, які дозволяють розширювати функціональність системи практично без обмежень, проте її конфігурація переважно відбувається через ручне редагування текстових файлів, що суттєво ускладнює масштабування та підтримку великої кількості вузлів [9].

Prometheus орієнтований на новітні інфраструктури, зокрема ті, що побудовані на базі Kubernetes, і збирає метрики за моделлю pull — тобто самостійно опитує визначені джерела. Завдяки вбудованій мові запитів PromQL та тісній інтеграції з Grafana, система забезпечує гнучкий підхід до збору, аналізу та візуалізації даних, що особливо актуально для DevOps-середовищ і мікросервісної архітектури. Варто зазначити, що Grafana не є самостійним інструментом моніторингу, а виступає потужною платформою для візуалізації, яка може використовуватись у зв’язці з такими джерелами, як Prometheus, InfluxDB, Zabbix

та іншими. Вона дозволяє створювати кастомізовані панелі управління, налаштовувати тривоги, а також інтерактивно аналізувати метрики, що робить її незамінною складовою комплексного моніторингу [9].

У цьому контексті особливої уваги заслуговує Netdata — сучасний інструмент для моніторингу в реальному часі, який орієнтований на простоту встановлення, автоматичну конфігурацію та зручний веб-інтерфейс. Його можливості охоплюють збір тисячі метрик щосекунди з мінімальним впливом на систему, підтримку широкого спектра служб і пристроїв, а також можливість роботи як у локальному середовищі, так і з централізованим хмарним сервісом Netdata Cloud для агрегації даних з багатьох вузлів. Така універсальність у поєднанні з детальною візуалізацією та простим налаштуванням робить його особливо доцільним вибором для організацій, які прагнуть оперативно впровадити ефективний моніторинг без складної інфраструктури чи потреби у значних витратах часу на конфігурування. Враховуючи порівняльний аналіз розглянутих інструментів, кожен з них має свої переваги та обмеження, що дозволяє формувати доцільні рекомендації для вибору рішення, яке найкраще відповідатиме технічним можливостям і вимогам конкретного підприємства (див. рис. 1.11) [9].



Рисунок 1.11 – Інтерфейс платформи Netdata

Критерії вибору систем моніторингу. Після огляду основних систем моніторингу виникає потреба переходу до усвідомленого вибору конкретної платформи, яка відповідатиме реальним умовам та вимогам підприємства, тому доцільно враховувати не лише технічні характеристики, а й практичні аспекти впровадження, подальшої підтримки, масштабування і ефективності використання в межах наявної інфраструктури. На першому місці при цьому має стояти простота встановлення та налаштування, адже інструмент повинен бути інтуїтивно зрозумілим і не вимагати тривалого часу для конфігурації, що є особливо актуальним для невеликих IT-команд із обмеженими ресурсами; наприклад, система Netdata здатна автоматично збирати метрики вже після виконання однієї команди в терміналі, завдяки чому її можна застосовувати у випадках, де час критично обмежений. Водночас функціональність і масштабованість залишаються основними чинниками, які визначають, наскільки платформа зможе охопити не лише мережеві пристрої типу маршрутизаторів, комутаторів і точок доступу, а й сервери, хостинг-сервіси, прикладні служби і все те, що забезпечує бізнес-процеси компанії; для цього найпридатнішими часто виявляються Zabbix або Prometheus, оскільки мають розширені механізми масштабування та гнучку архітектуру [10].

Не менш важливим критерієм є можливості візуалізації даних і доступу до аналітики, бо недостатньо лише збирати метрики – необхідно мати інструменти, які дозволяють швидко інтерпретувати інформацію через графіки, панелі стану та історичні звіти, які можуть бути реалізовані як за допомогою зовнішніх платформ на зразок Grafana, так і у вбудованому інтерфейсі самого моніторингового рішення, як це реалізовано в Netdata, де дашборди вже готові до використання без необхідності додаткового налаштування. Наступним критичним параметром виступає підтримка сповіщень та інтеграцій, адже без системи алертів, яка буде оперативно інформувати про перевищення допустимих навантажень, падіння сервісів чи відхилення від нормальної роботи, система моніторингу втрачає свій сенс як інструмент превентивного управління; завдяки підтримці таких каналів як email, Slack, Telegram або вебхуки, можлива гнучка інтеграція з внутрішніми процесами компанії, що суттєво підвищує швидкість реагування [10].

Окрему увагу також необхідно звернути на ресурсоємність обраного рішення, оскільки сама система моніторингу не повинна створювати критичного навантаження на інфраструктуру, яку вона має контролювати; з цієї точки зору, Netdata, особливо у своїй локальній конфігурації, має суттєву перевагу, адже потребує мінімум системних ресурсів при досить широкому функціоналі. Сукупність усіх цих критеріїв дозволяє не просто оцінити кожну з доступних систем, а класифікувати їх відповідно до реальних потреб підприємства, що стає основою для обґрунтованого вибору.

Порівняння локальних і хмарних рішень. У цьому контексті доречно звернутися до порівняння локальних і хмарних рішень, оскільки обидва підходи мають не лише переваги, а й потенційні обмеження. Локальні системи, такі як Zabbix, Prometheus або локальна інсталяція Netdata, забезпечують повний контроль над усіма даними та дозволяють повноцінно працювати навіть за відсутності інтернет-з’єднання, водночас дозволяючи максимально адаптуватися до політик безпеки внутрішньої мережі підприємства, однак потребують виділених ресурсів для обслуговування, регулярного резервного копіювання та постійної підтримки працездатності, що не завжди є можливим або ефективним для невеликих компаній. Хмарні рішення, як-от Netdata Cloud, Grafana Cloud чи Datadog, навпаки, надають централізований доступ до аналітики та дашбордів із будь-якої точки світу, зменшують навантаження на внутрішні сервери та мають більш гнучку систему оновлень, хоча і накладають певні фінансові витрати через необхідність передплати, а також залежать від стабільного інтернет-з’єднання та мають обмеження щодо зберігання історичних даних [11].

Нетрадиційним і водночас ефективним варіантом є гібридна модель, яку пропонує Netdata, оскільки дозволяє інсталювати агент локально, зберігаючи всі переваги on-premise інфраструктури, але водночас підключити його до хмарного середовища Netdata Cloud, де можливо об’єднати кілька вузлів в один інтегрований дашборд із централізованим керуванням, без необхідності відкривати порти, прокладати VPN або реалізовувати складні схеми маршрутизації; завдяки цьому стає можливою реалізація нагляду за розподіленими мережами в простий і

безпечний спосіб, що особливо цінно в умовах обмежених технічних і часових ресурсів (див. рис. 1.12) [11].

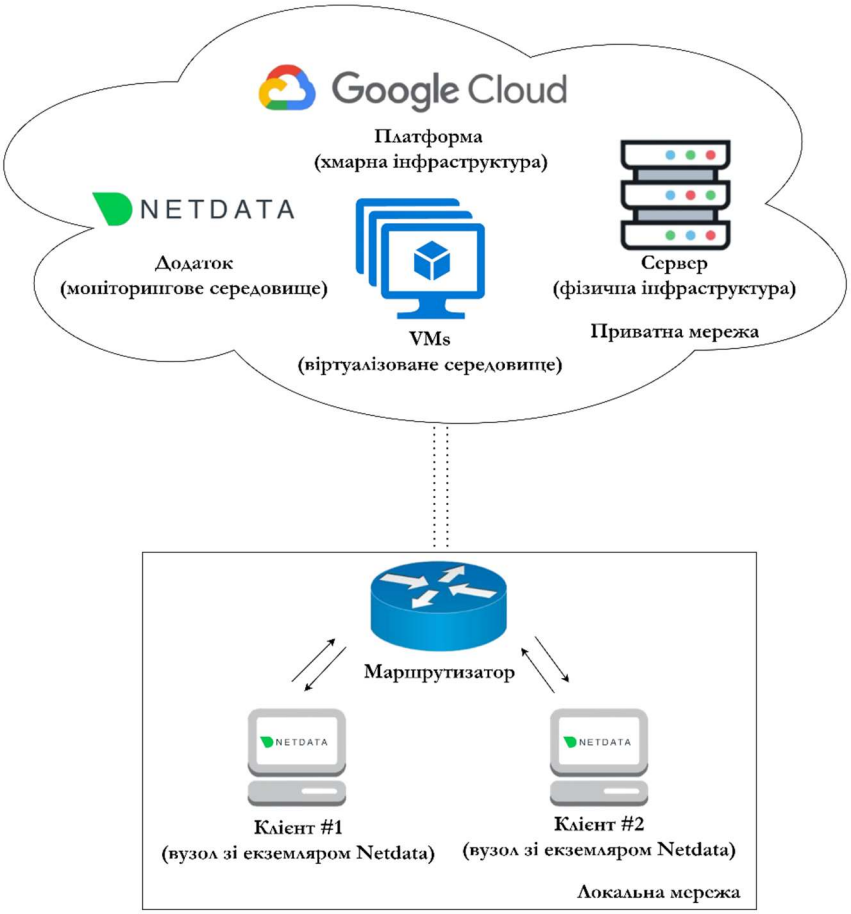


Рисунок 1.12 – Взаємодія хмарного рішення з локальними вузлами

Netdata — сучасне рішення для моніторингу в реальному часі. Netdata є сучасним рішенням для моніторингу в реальному часі, яке вирізняється серед інших платформ завдяки філософії «нульового налаштування», що дозволяє розпочати моніторинг одразу після встановлення без потреби у складній попередній конфігурації. Такий підхід робить систему придатною як для початківців, так і для досвідчених адміністраторів, оскільки забезпечується швидкий доступ до системних метрик через вбудований веб-інтерфейс одразу після інсталяції, яка займає лише кілька хвилин. Платформа автоматично виявляє й візуалізує ключові параметри, включно з використанням процесора, оперативної пам’яті, мережевого трафіку та дисків, у вигляді зручних графіків, що оновлюються

з мінімальною затримкою, меншою за одну секунду, що особливо важливо при оперативному реагуванні на зміни в системі (див. рис. 1.13) [12].



Рисунок 1.13 – Інтерфейс реального часу в Netdata

Крім базового функціоналу, Netdata підтримує понад двісті різних типів служб, зокрема Docker, Samba, UPS, веб-сервери та бази даних, що дозволяє централізовано спостерігати за роботою різноманітного програмного забезпечення з єдиного інтерфейсу. Кожен вузол можна оснастити агентом, який споживає вкрай мало системних ресурсів, зокрема лише кілька мегабайт оперативної пам’яті, що робить систему майже непомітною для основної інфраструктури. Такий підхід дозволяє впроваджувати моніторинг навіть на слабших або обмежених за ресурсами пристроях без шкоди для загальної продуктивності [12].

Окрему увагу варто звернути на можливість використання Netdata Cloud — хмарної частини платформи, яка забезпечує централізований перегляд великої кількості вузлів без необхідності налаштування VPN, пробросу портів чи інших складних дій, при цьому даючи змогу контролювати стан інфраструктури з будь-якої точки світу. У випадку невеликих чи середніх підприємств, де важливе значення має простота розгортання, швидке виявлення проблемних ділянок та мінімальні витрати на обслуговування, таке поєднання локального агента з можливістю хмарного підключення стає практичним і логічним вибором [12].

Таблиця 1.1 – Порівняння популярних інструментів моніторингу мережі

Назва	Тип архітектури	Простота встановлення	Візуалізація	Сповіщення	Ресурсоємність	Особливості
Zabbix	Централізована	Середня	Вбудована	Потужна	Висока	Глибоке налаштування, масштабування
Nagios	Централізована	Низька	Обмежена	Гнучка	Середня	Вимагає ручної конфігурації
Prometheus	Розподілена	Середня	Через Grafana	Обмежена	Середня	Орієнтований на метрики, PromQL
Grafana	Візуалізатор	Висока	Потужна	Так	Низька	Вимагає джерела метрик
Netdata	Агентна (гібрид)	Висока	Вбудована	Базова	Низька	Моніторинг у реальному часі, Cloud-інтеграція

Незважаючи на численні переваги (див. табл. 1.1), певні обмеження системи також слід враховувати під час впровадження. До таких обмежень можна віднести меншу гнучкість щодо налаштування сповіщень у порівнянні з деякими більш

складними системами, зокрема Zabbix, де можливості тригерів і повідомлень є значно ширшими. Також, за замовченням Netdata не зберігає історичні дані на тривалий період, що означає автоматичне очищення метрик при досягненні певного ліміту пам’яті або часу, якщо не використовується зовнішнє централізоване сховище або Netdata Cloud, що може створити складнощі при потребі довгострокового аналізу продуктивності або підготовки звітності, якщо ці аспекти є критично важливими [12].

У контексті розгортання моніторингу на підприємстві важливо враховувати як масштаби інфраструктури, так і технічні можливості самої платформи. Залежно від обсягів даних, кількості сервісів і серверів, а також вимог до зберігання й візуалізації інформації, обране рішення повинне не тільки ефективно функціонувати у поточних умовах, а й мати потенціал до масштабування. У випадку, коли необхідно швидко запровадити базовий моніторинг без додаткових складнощів із налаштуванням, навчанням персоналу та надмірним навантаженням на систему, саме Netdata демонструє свою ефективність, простоту й логічність впровадження. Тому у проєктах, орієнтованих на оперативність, компактність та мінімізацію технічного боргу на етапі запуску, обраний варіант заслуговує на розгляд як базове та перспективне рішення для сучасної інфраструктури.

1.4 Метрики та параметри, що підлягають моніторингу

Системи віддаленого моніторингу комп’ютерних мереж дозволяють отримати повну та всебічну картину технічного стану інфраструктури підприємства, включно з аспектами, які не завжди є очевидними при звичайному використанні. Збір і аналіз метрик, тобто кількісних параметрів, що відображають поточний стан обладнання, програмного забезпечення та мережевих підключень, забезпечує можливість завчасного виявлення потенційних проблем ще до того, як вони переростуть у критичні збої чи повну втрату доступу до важливих ресурсів.

Саме тому метрики, що підлягають моніторингу, потребують глибокого аналізу з чітким розумінням значення кожного показника та його впливу на загальну стабільність роботи інфраструктури. На прикладі практичного рішення, такого як Netdata, можна прослідкувати, як реалізується детальне спостереження за різними компонентами системи [13-16].

Основні категорії метрик. Метрики умовно поділяються на кілька основних категорій відповідно до елементів системи, за якими здійснюється нагляд. До ключових належать показники:

— центрального процесора;
— оперативної пам’яті;
— дискової підсистеми;
— мережевих інтерфейсів.

З огляду на центральний процесор, він є критичним елементом будь-якого обчислювального вузла, і його перевантаження здатне призводити до затримок у роботі прикладного ПЗ, втрати запитів і загального зниження продуктивності. Основними метриками тут виступають завантаження CPU у відсотках, що вказує на ступінь використання кожного ядра, середнє навантаження (load average) за останні 1, 5 та 15 хвилин, кількість активних процесів у черзі на обробку, а також розподіл часу процесора між користувацькими задачами, системними функціями та простоєм. Аналіз подібних показників дає змогу виявити вузли, що працюють на межі своїх можливостей, або ситуації, коли розподіл ресурсів відбувається неефективно [13].

Крім процесора, вагоме значення має оперативна пам’ять, яка виступає тимчасовим сховищем даних та коду, що виконується. Її дефіцит часто призводить до активації механізму свопінгу, коли дані тимчасово записуються на диск, що значно знижує продуктивність. Відповідні метрики включають обсяг використаної та вільної пам’яті (у мегабайтах або у відсотках), обсяг кешу та буферів, що свідчить про ефективність використання RAM для пришвидшення доступу до даних, а також показники свопу, які відображають обсяг даних, переміщених на диск унаслідок нестачі фізичної пам’яті. За цими показниками визначається

необхідність розширення ОЗП або оптимізації програмного середовища, що споживає ресурси [13].

Не менш важливим є моніторинг дискової підсистеми, яка зберігає постійні дані і може ставати вузьким місцем у разі надмірного навантаження чи зношення обладнання. Основні метрики тут охоплюють використання файлової системи — обсяг зайнятого та доступного простору, кількість операцій введення/виведення за секунду (IOPS), середній час затримки обробки запитів, а також кількість помилок читання або запису. Всі ці показники дозволяють виявити проблеми, пов’язані з нестабільністю сховища, зниженням його швидкодії чи ризиками втрати доступу до важливих даних [13].

Ще однією критичною категорією є метрики мережевих інтерфейсів, особливо актуальні для інфраструктур, які базуються на хмарних технологіях або мають значну частку віддаленого доступу. Основні показники тут включають пропускну здатність у мегабітах за секунду, яка показує фактичний обсяг даних, що передаються мережею, а також кількість прийнятих (RX) та переданих (TX) пакетів. Додатково варто враховувати втрати пакетів, які зазвичай сигналізують про фізичні або логічні проблеми в з’єднанні, помилки CRC, колізії, дублікати — усі ці показники можуть свідчити про несправність мережевого обладнання, погану конфігурацію або перевантаження ліній зв’язку. Своєчасне реагування на відхилення цих метрик дозволяє уникати деградації якості зв’язку та запобігати аварійним ситуаціям у передачі даних (див. рис. 1.14) [13].

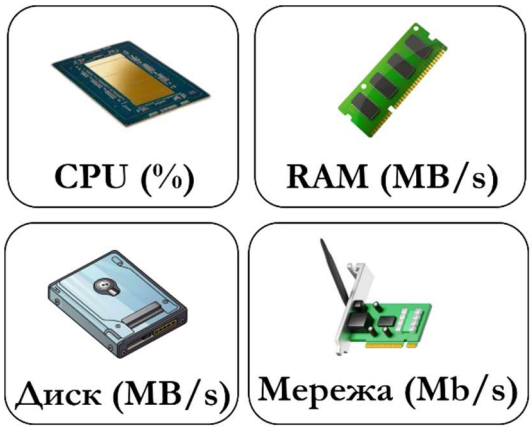


Рисунок 1.14 – Категорії метрик у системі моніторингу

Усі ці категорії метрик формують основу для побудови ефективної системи моніторингу, яка має не просто фіксувати поточний стан, але й дозволяти робити прогностичні висновки щодо майбутньої поведінки системи, реагувати на відхилення ще до того, як вони переростуть у критичні інциденти, що створює передумови для стабільної та надійної роботи всієї IT-інфраструктури.

Стан мережевих сервісів і підключень. Контроль стану мережевих сервісів і підключень є важливим аспектом у забезпеченні стабільної роботи підприємства, оскільки недостатньо лише відстежувати апаратні показники, необхідно також перевіряти доступність і працездатність ключових сервісів, таких як:

- веб-сервери;
- бази даних;
- служби автентифікації;
- DNS сервер;
- поштові системи;
- інші критичні елементи інфраструктури.

У цьому контексті основним критерієм перевірки є наявність відкритого порту на відповідному хості, що свідчить про активність сервісу. Окрім цього, важливим є підтвердження відповіді на мережеві запити, як-от ping чи HTTP, які дозволяють визначити не лише наявність відповіді, а й виміряти затримку при зверненні до конкретного сервісу. Значенням, що заслуговує окремої уваги, є також час відгуку, або response time, оскільки він демонструє, наскільки швидко сервіс обробляє запит і відповідає на нього, що прямо впливає на користувацький досвід і загальну ефективність взаємодії з мережею. Ще одним критичним аспектом є моніторинг терміну дії SSL/TLS сертифікатів, які використовуються для забезпечення безпечного з’єднання через HTTPS, адже у разі їхнього прострочення доступ до домену може бути повністю втрачений, що потенційно блокує функціонування цілих систем або сервісів. Таким чином, навіть за умови відсутності апаратних збоїв, можна виявити непрацездатність тієї чи іншої служби і вчасно втрутитися до того, як вона стане недоступною для користувачів (див. рис. 1.15) [13-14].

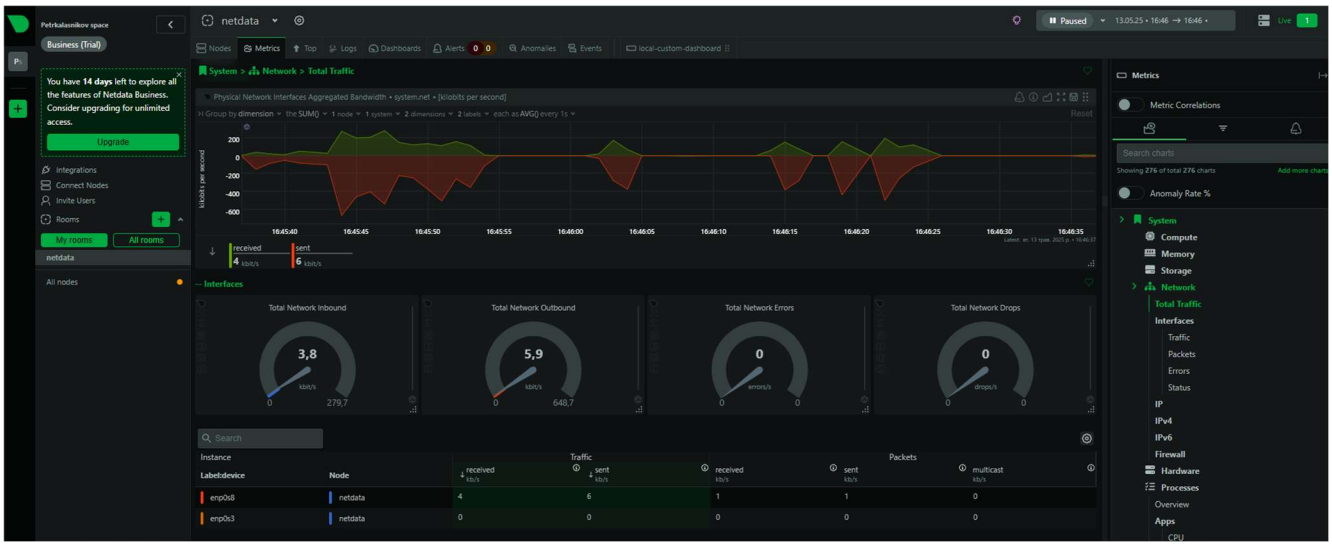


Рисунок 1.15 – Приклад моніторингу підключень до мережі

Події та сповіщення. Однак, важливим доповненням до процесу моніторингу є не лише пасивне збирання показників, а й активне реагування на зміни в системі шляхом генерації подій та сповіщень, що надає змогу автоматизувати процеси оповіщення відповідальних осіб або запуску скриптів у разі порушення стандартного режиму роботи. Серед типів подій, що можуть генеруватися системою, виділяють:

— критичні;

— попереджувальні;

— інформаційні.

Критичні події сигналізують про серйозні відхилення — як-от повна недоступність сервера чи сервісу, вичерпання вільного місця на диску, або перегрів обладнання, що може спричинити відмову системи; подібні інциденти потребують негайного втручання з боку адміністратора. Попереджувальні події інформують про досягнення визначених порогових значень, наприклад перевищення завантаження CPU понад 80% чи заповнення дискового простору понад певний відсоток, даючи змогу заздалегідь реагувати до того, як ситуація перетвориться на критичну. Інформаційні події, у свою чергу, фіксують загальні процеси, які регулярно відбуваються в системі, зокрема запуск чи завершення сервісів, виконання оновлень, створення резервних копій тощо — вони не вказують на проблеми, але

забезпечують можливість простеження історії змін і аналізу причин можливих відхилень у роботі. Важливо, що всі ці події можуть бути прив’язані до конкретних реакцій системи: надсилання повідомлень через електронну пошту, SMS, Telegram, Slack або інші канали зв’язку, що дає змогу оперативно інформувати відповідальних осіб про події, які відбуваються в інфраструктурі [15].

Окрім надсилання сповіщень, реакція системи може також включати автоматичне виконання дій, наприклад перезапуск проблемного сервісу, очищення тимчасових файлів для звільнення дискового простору або інші сценарії, що зменшують потребу в постійному втручанні людини у типові ситуації. У випадках, коли подія не потребує негайної дії, або коли її вплив не очевидний з першого погляду, вона може бути зафіксована у логах для подальшого аналізу або проведення аудиту, що дозволяє виявляти причинно-наслідкові зв’язки між подіями та потенційними збоями в роботі мережі (див. рис. 1.16).

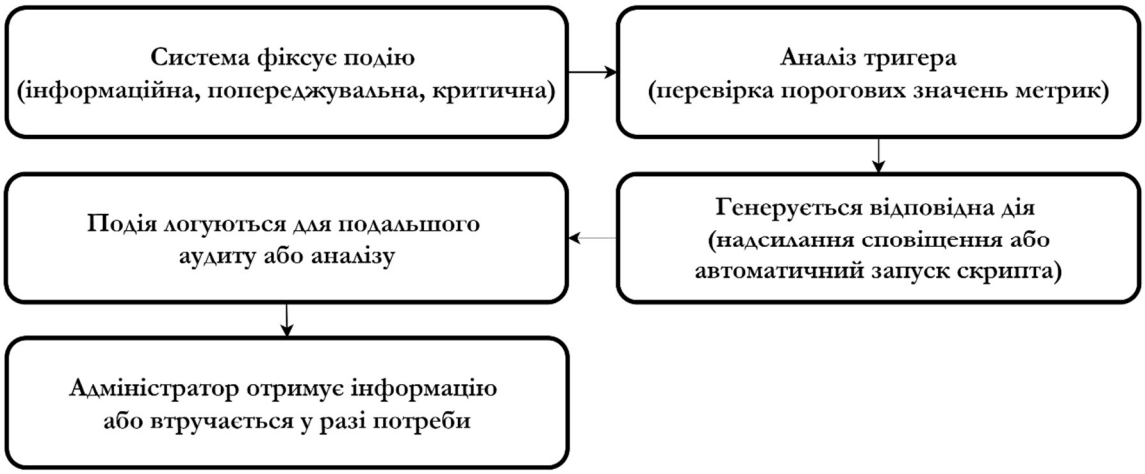


Рисунок 1.16 – Процес обробки подій і надсилання сповіщень

Таким чином, інтеграція системи подій і сповіщень у моніторингову інфраструктуру забезпечує гнучкий та надійний механізм управління інцидентами, завдяки якому вдається не лише отримувати повну картину про стан системи в реальному часі, але й своєчасно реагувати на загрози або попереджати їх появу [15].

Підтримка візуалізації часу простоїв і пікових навантажень. Розглядаючи поняття візуалізації у системах моніторингу більш детально, необхідно відзначити,

що це один із найефективніших способів аналізу великої кількості метрик, оскільки сучасні засоби спостереження за станом інформаційної інфраструктури надають широкі можливості побудови графіків, дашбордів і звітів, які значно полегшують процес виявлення критичних періодів, таких як часи простою окремих пристроїв або сервісів, що були недоступні, чи моменти пікових навантажень, коли споживання ресурсів досягало максимуму в окремі години чи дні. Завдяки візуалізації також можливо виявляти повторювані закономірності, наприклад, стабільне перенавантаження систем у певний час доби або поступове зростання навантаження на ресурси, що свідчить про необхідність масштабування інфраструктури [16].

Основними інструментами такої візуалізації найчастіше є графіки часових рядів (time series), що застосовуються для відображення динаміки завантаження процесора, оперативної пам’яті, дисків та мережевої активності, тоді як теплові карти дозволяють оцінити частоту виникнення певних подій у межах доби або тижня, демонструючи найбільш навантажені періоди (див. рис. 1.17).

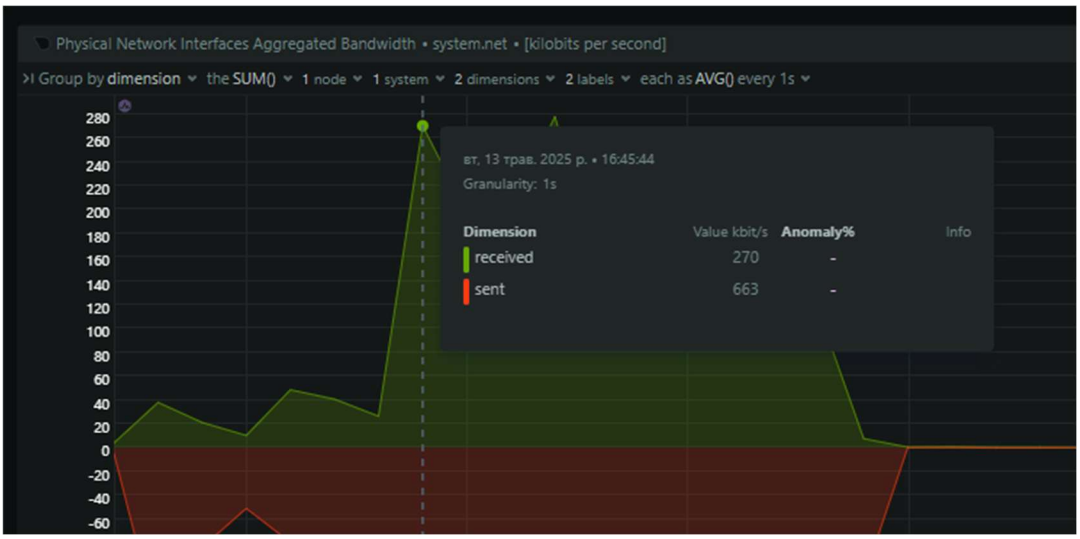


Рисунок 1.17 – Візуалізація пікових навантажень у дашборді

Інтерактивні панелі керування, або ж дашборди, також залишаються центральним елементом будь-якої системи моніторингу, оскільки вони забезпечують об’єднання десятків або сотень метричних даних у зручному візуальному середовищі, що

адаптується під конкретного користувача і дозволяє швидко ідентифікувати відхилення від встановлених нормативів або звичних значень [16].

Візуалізація не обмежується лише спрощенням аналітичного процесу для користувачів, які не мають глибоких технічних знань, оскільки навіть досвідчені спеціалісти, які звикли працювати через термінал або командний рядок, визнають, що при наявності сотень або тисяч метрик ручний аналіз за допомогою команд на зразок "grep" чи "awk" стає надзвичайно трудомістким і непродуктивним. У цьому контексті рішення на зразок Grafana чи аналогічних платформ забезпечують не лише оглядовість, а й можливість виявляти кореляції між різними подіями, порівнювати історичні дані, накладати одні метрики на інші, візуалізуючи всі ці аспекти в єдиному просторі. Без такої структури системи моніторингу ризикують перетворитися на склад масивів сирих чисел, які важко інтерпретувати, і які втрачають свою цінність без відповідного інтерфейсу представлення [16].

Візуалізація не є допоміжним або другорядним компонентом, а навпаки — визначальною умовою того, що процес моніторингу взагалі здатен виконувати свою функцію в реальному часі, роблячи результати доступними для всіх рівнів фахівців, зокрема менеджерів, яким потрібен швидкий огляд поточної ситуації без заглиблення у технічні подробиці, що забезпечує не лише оперативне реагування на виявлені збої, а й формування стратегічних рішень щодо розвитку мережевої інфраструктури, її оптимізації та підготовки до майбутніх навантажень (див. табл. 1.2) [16].

Таблиця 1.2 – Основні метрики для моніторингу компонентів комп’ютерної мережі

Категорія	**Ключові метрики**
Процесор (CPU)	Завантаження CPU (%), Load average, час користувача/ядра/простоїв
Оперативна пам’ять	Використана/вільна пам’ять, кеш/буфери, обсяг swap

Продовження таблиці 1.2 – Основні метрики для моніторингу компонентів комп’ютерної мережі

Категорія	**Ключові метрики**
Дискова система	Використання простору, IOPS, затримки, помилки читання/запису
Мережеві інтерфейси	Пропускна здатність, Rx/Tx пакети, втрати пакетів, CRC-помилки, колізії
Мережеві сервіси	Порт-чек, ping/HTTP-відповідь, час відгуку, стан сертифікатів
Події та сповіщення	Критичні/попереджувальні/інформаційні події, Email/SMS/скрипти, логування
Візуалізація	Графіки часу, теплові карти, інтерактивні дашборди

У підсумку моніторинг метрик у комп’ютерній мережі підприємства не обмежується лише контролем технічного стану окремих вузлів, а перетворюється на фундамент стабільності, передбачуваності та довгострокової ефективності всієї інформаційної системи, оскільки лише завдяки категорійному поділу метрик, автоматизованому збору інформації та її грамотному візуальному поданню можливо створити умови для прийняття зважених управлінських рішень, запобігання аваріям і забезпечення безперервної роботи IT-інфраструктури.

1.5 Технічні вимоги до побудови системи моніторингу

Побудова системи віддаленого моніторингу комп’ютерної мережі потребує врахування низки технічних вимог, які безпосередньо впливають як на ефективність функціонування програмних компонентів, так і на загальну стабільність всієї системи в умовах заданих обмежень. Незалежно від масштабу або складності конкретного середовища впровадження, будь-яке рішення з

моніторингу повинне мати мінімальний набір апаратних і програмних передумов, які дозволяють забезпечити безперебійну роботу сервісів збору, обробки та візуалізації даних, що є основою інформаційної надійності таких систем. У випадку реалізації рішення на базі системи Netdata, яка буде основним інструментом для відстеження стану мережевої інфраструктури в даному контексті, ці вимоги можуть бути адаптовані відповідно до специфіки обраної архітектури та мети розгортання [17-20].

Мінімальні системні ресурси для сервера моніторингу. Варто зазначити, що системи моніторингу не завжди потребують високопродуктивного обладнання, оскільки їх основна функція полягає у спостереженні, а не у виконанні ресурсоємних обчислень. У багатьох випадках достатньо базових апаратних конфігурацій, що особливо актуально для малих та середніх підприємств, де ресурси обмежені, але потреба у контролі за станом мережі є постійною. Разом з тим навіть мінімальне середовище повинно відповідати певним вимогам, щоб гарантувати коректну роботу обраного рішення. Так, процесор може бути одно- або двоядерним із архітектурою "x86_64" або "ARM", що дозволяє запускати систему як у віртуалізованому середовищі, так і на фізичному обладнанні. У реалізованому варіанті передбачено використання VirtualBox як основної платформи для розгортання, оскільки вона забезпечує швидке та зручне створення ізольованого середовища без додаткових технічних ускладнень [17].

Обсяг оперативної пам’яті для базового розгортання може починатися від 512 МБ, однак для стабільної роботи оптимальною буде конфігурація з 1 ГБ і більше, що пов’язано з тим, що система використовує лише необхідні служби для збору метрик та їх візуалізації, без зайвих графічних оболонок або непотрібних фонових процесів, як у випадку десктопних операційних систем. Обсяг дискового простору для інсталяції також не є критичним — для базової установки разом із логами і тимчасовими файлами достатньо 200–500 МБ, однак доцільно резервувати щонайменше 20–30 ГБ для забезпечення роботи всіх можливих сервісів та уникнення обмежень у майбутньому. При цьому слід зважати на те, чи система працюватиме у контейнеризованому середовищі чи віртуальних машинах, оскільки

від цього залежатиме розділення сервісів і можливі конфлікти, зокрема при використанні одних і тих самих мережевих портів.

Тип накопичувача також має значення: бажано використовувати SSD, який забезпечує швидший доступ до кешованих даних, хоча для невеликих обсягів інформації, особливо в системах зі спрощеним навантаженням, допустимим є і використання HDD. У випадку базового розгортання Netdata можна застосовувати навіть слабке або енергоефективне обладнання, включаючи одноплатні комп'ютери на кшталт Raspberry Pi 3 або 4, або міні-ПК із міні-ITX платою і мінімальним обсягом оперативної пам'яті, що відкриває можливість гнучкого використання системи як у локальних, так і у віддалених сценаріях без значних витрат.

Однак зі збільшенням масштабів — коли кількість вузлів зростає, або збір даних відбувається з високою частотою в реальному часі — навантаження на систему зростає пропорційно. Якщо в середовищі працює понад 50 активних сервісів або серверів, які постійно передають телеметрію, система повинна обробляти велику кількість одночасних запитів, зберігати історичні дані і надавати доступ до агрегованих результатів. У таких випадках необхідно оновити технічну базу:

— процесор з чотирма ядрами;
— оперативна пам'ять не менше 4 ГБ;
— дисковий простір не менше 60 ГБ.

Усе це необхідно для того, щоб система моніторингу була здатна зберігати значні обсяги історичних логів та метрик. Особливо це актуально при інтеграції із зовнішніми бекендами або додатковими інструментами агрегації даних, де затримка доступу чи нестача ресурсів може негативно вплинути на загальну стабільність системи [17].

Вимоги до операційної системи – Ubuntu Server. При створенні системи для віддаленого моніторингу стану комп'ютерної мережі підприємства важливим аспектом є вибір операційної системи, що слугуватиме основою для всієї моніторингової інфраструктури. Саме на цьому сервері розміщується відповідне

програмне забезпечення, зокрема й Netdata, тому стабільність, сумісність і ресурсна ефективність обраної системи прямо впливають на надійність і продуктивність усієї системи. Хоча в теорії можливо використовувати різноманітні дистрибутиви Linux, включно з надлегкими варіантами для максимальної економії системних ресурсів, у рамках цього проєкту було обрано Ubuntu Server як одне з найстабільніших, найпоширеніших і найкраще підтримуваних середовищ для серверного використання. Такий вибір пояснюється як практичними міркуваннями — простотою розгортання, широкою документаційною базою, доступністю спільноти — так і технічними перевагами, зокрема сумісністю з великим числом серверних рішень і зокрема з Netdata, що доступна як через офіційні пакети в "APT", так і через ручне встановлення .deb-файлів, що дозволяє скоротити час на налаштування, уникнути помилок при ручному конфігуруванні, а також забезпечити швидкий запуск і перевірку працездатності системи навіть при мінімальному досвіді роботи з Linux [18].

Ubuntu Server LTS-релізів надає стабільність завдяки п’ятирічному офіційному циклу підтримки, регулярним оновленням безпеки й надійності, а також дозволяє уникнути непередбачених збоїв при довготривалому використанні. До того ж, відсутність графічного інтерфейсу в типовій конфігурації Ubuntu Server забезпечує мінімальне споживання оперативної пам’яті та процесорного часу, що особливо важливо у випадках, коли проєкт реалізується на обмежених апаратних ресурсах. Згідно з документацією, для запуску Ubuntu Server 22.04 LTS мінімальні вимоги складають:

— один процесорний потік (CPU);
— 512 МБ оперативної пам’яті (RAM);
— приблизно 2 ГБ вільного дискового простору.

Однак у практичних цілях рекомендовано орієнтуватися на об’єм від 8 до 20 ГБ в залежності від планованого використання та наявності додаткових служб. Якщо система використовується для контейнеризації або розширюється іншими компонентами, обсяг ресурсів має бути збільшено з урахуванням майбутнього навантаження (див. рис. 1.18).

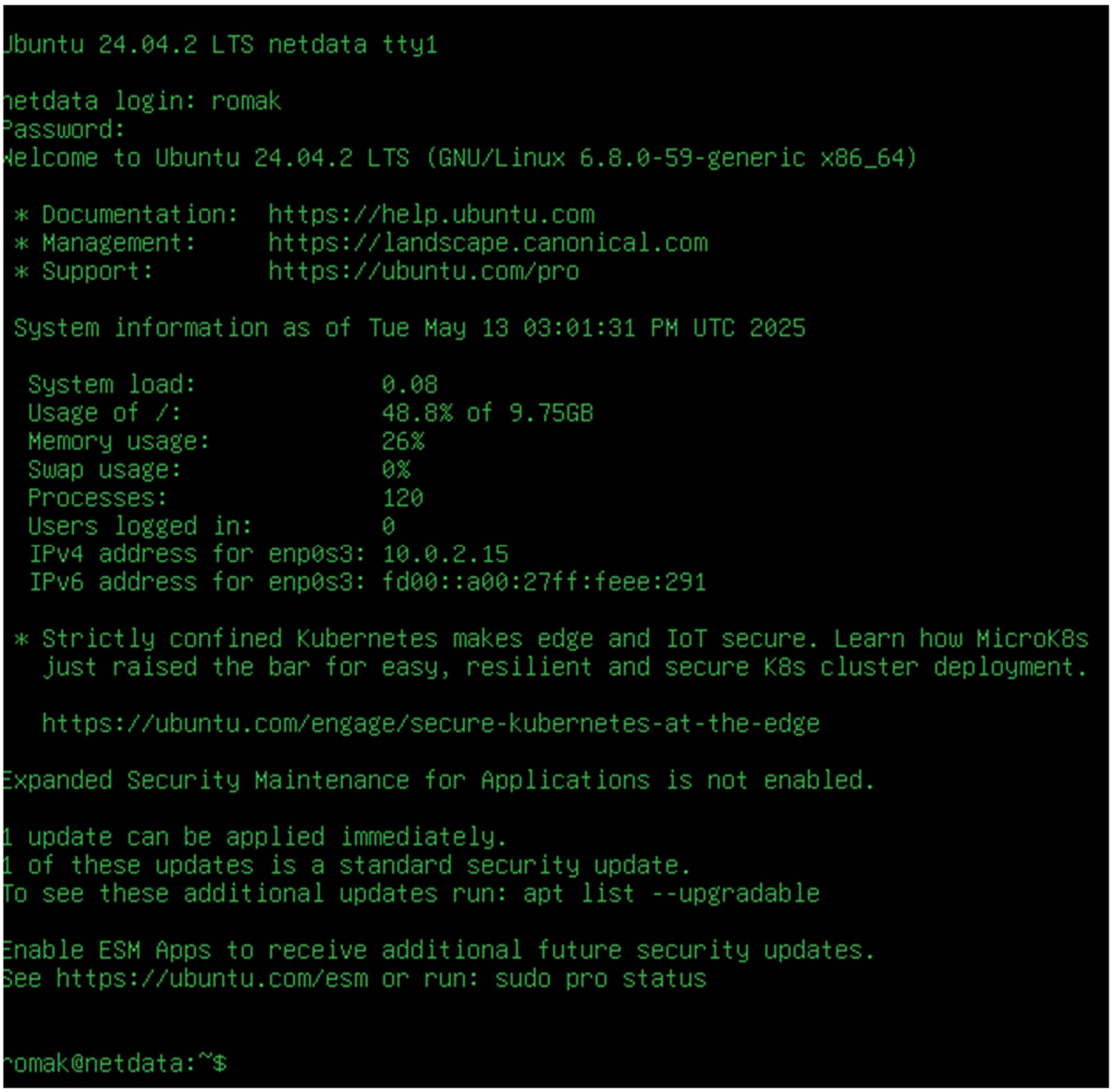


Рисунок 1.18 – Середовище Ubuntu Server для системи моніторингу

У підсумку, Ubuntu Server ідеально підходить для реалізації демонстраційних або навчальних проєктів, а також для продуктивного застосування в невеликих організаціях, де надійність і простота адміністрування мають вирішальне значення [18].

Мережеві порти та доступи. Поряд із вибором операційної системи необхідно також коректно налаштувати мережеві порти й забезпечити доступ до сервера, оскільки без відповідного мережевого оточення жодна система моніторингу не зможе повноцінно функціонувати. Netdata працює у вигляді

автономного веб-сервісу, доступ до якого відбувається через браузер. Основним портом для доступу до веб-інтерфейсу Netdata є "19999" TCP, що за замовчуванням відкриває панель моніторингу за адресою "http://IP:19999". Цей порт є ключовим для перегляду даних, зібраних агентом Netdata у режимі реального часу. Додатково для забезпечення адміністрування сервера використовується SSH-порт, традиційно "22" TCP, однак він може бути змінений на інший, наприклад "222" TCP, з міркувань безпеки. SSH забезпечує можливість віддаленого доступу до сервера, що особливо критично у випадку необхідності діагностики або втручання в роботу системи без фізичного доступу [19].

Крім того, можуть бути використані стандартні порти "80" та "443" TCP у випадках, коли до сервісу здійснюється доступ через зворотний проксі або налаштований HTTPS. Таке рішення реалізується за допомогою nginx або Apache і дозволяє прив’язати сервіс до доменного імені, додати TLS-захист та інтегрувати систему в існуючу інфраструктуру підприємства. Однак з міркувань інформаційної безпеки не рекомендується залишати порт "19999" відкритим для глобального доступу, натомість його слід обмежити лише для локальної мережі або захистити через VPN, а при необхідності зовнішнього доступу — реалізувати тунелювання або захищене з’єднання через TLS. Обов’язковим є також впровадження хоча б базових механізмів контролю доступу: авторизація через логін/пароль або обмеження за IP-адресою, аби запобігти несанкціонованому доступу до системи моніторингу ззовні. У кінці, правильно налаштована мережева інфраструктура дозволяє поєднати зручність використання з базовим рівнем безпеки, який є прийнятним для невеликих підприємств і внутрішніх систем контролю [19].

Обґрунтування вибору інсталяції без використання Docker. Netdata підтримує кілька способів встановлення включаючи:

— використання Docker-контейнера;

— інсталяцію безпосередньо на серверну операційну систему;

— агентське встановлення з автозапуском через Kickstart.

У межах реалізації даного проєкту було свідомо обрано пряме встановлення на операційну систему без використання Docker, що обумовлюється низкою

раціональних технічних причин. Передусім у рамках демонстраційного середовища відсутність проміжного шару контейнеризації дозволяє сконцентруватися на самому процесі моніторингу, не витрачаючи ресурси на налаштування та обслуговування контейнерного середовища. Docker доцільно використовувати у випадках, коли необхідно ізолювати велику кількість сервісів або керувати масштабованими розгортаннями, однак у простих реалізаціях пряме встановлення виявляється доцільнішим через свою прозорість, зменшення складності та кращу інтеграцію з базовим середовищем [20].

Інсталяція без Docker забезпечує безпосередній доступ до служб, портів та конфігураційних файлів без додаткових шарів абстракції, таких як bridge-мережа чи внутрішні правила контейнера, що часто ускладнюють діагностику та налагодження. Усе керування конфігурацією відбувається через системні каталоги, наприклад "/etc/netdata" або "/var/lib/netdata", що спрощує резервне копіювання, відновлення та загальне адміністрування. Така прозорість дозволяє краще зрозуміти механізми роботи самої системи моніторингу та її зв’язки з сервісами на рівні операційної системи, що особливо корисно в навчальних чи демонстраційних середовищах. Крім того, Netdata має вбудовані засоби автоматичного оновлення навіть при прямій інсталяції, що знижує потребу в регулярному ручному втручанні [20].

Пряме встановлення також зменшує накладні витрати, які все ж таки супроводжують запуск контейнеризованих сервісів. Незважаючи на уявну легкість Docker-контейнерів, у залежності від складності й навантаження конкретного сервісу, вони можуть створювати помітне споживання ресурсів CPU та пам’яті. У випадку старішого обладнання або систем із обмеженими ресурсами, як-от віртуальні машини з 1 ГБ оперативної пам’яті, навіть незначне зниження навантаження стає критично важливим. Саме тому розгортання без Docker дозволяє використовувати Netdata навіть на мінімальних конфігураціях, зберігаючи при цьому повноцінну функціональність моніторингу (див. табл. 1.3).

Таблиця 1.3 – Порівняння варіантів розгортання Netdata

Критерій	**Пряма інсталяція**	**Docker-контейнер**
Простота запуску	Вища прозорість, класичний підхід	Швидкий старт за допомогою одного рядка
Доступ до хост-системи	Повний доступ до системних ресурсів	Обмежений, потребує додаткових прав
Продуктивність	Вища, оскільки працює напряму на системі	Можливі накладні витрати контейнеризації
Оновлення та підтримка	Легко оновлювати через скрипт/репозиторій	Потребує оновлення образу та перезапуску
Гнучкість конфігурації	Прямий доступ до конфіг-файлів	Обмежений доступ, складніший мапінг томів
Безпека	Залежить від хост-системи	Краще ізольований від основної системи
Масштабованість	Підходить для стаціонарних вузлів	Зручно в DevOps-сценаріях і CI/CD
Залежності та сумісність	Потребує сумісної ОС та бібліотек	Контейнер вирішує залежності самостійно

Зручність налаштування, простота встановлення, прозорість обслуговування та гнучкість у керуванні роблять таку модель привабливою для невеликих офісів, навчальних закладів або обмежених у ресурсах локальних середовищ. Вона не вимагає спеціалізованих знань у галузі контейнеризації та дозволяє розгорнути повноцінну систему моніторингу у найкоротші терміни. Таким чином, у контексті побудови системи віддаленого моніторингу комп’ютерної мережі підприємства вибір прямої інсталяції Netdata на Ubuntu Server без використання Docker забезпечує оптимальний баланс між функціональністю, простотою розгортання та ефективним використанням ресурсів.

З урахуванням технічних обмежень, які виникають на етапі проєктування подібних систем, саме такий підхід дозволяє реалізувати моніторинг навіть у

найбільш скромних умовах, при цьому не втрачаючи жодного з ключових компонентів функціонування. Відсутність складних залежностей, мінімальні вимоги до апаратної частини, гнучкість у налаштуванні доступу та обслуговування — усе це робить пряме встановлення зручним і ефективним для впровадження у широкому спектрі сценаріїв, від локального тестування до стабільної експлуатації в середовищах малого масштабу (див. табл. 1.4) [20].

Таблиця 1.4 – Технічні вимоги до розгортання системи моніторингу на базі Netdata

Компонент	**Мінімальні вимоги**	**Рекомендації при масштабуванні**
Процесор (CPU)	1–2 ядра (x86-64 або ARM)	4 ядра і більше
Оперативна пам’ять (RAM)	512 МБ (оптимально – 1 ГБ)	4 ГБ і більше
Дисковий простір	200–500 МБ	додаткове місце для історичних даних
Тип диска	HDD або SSD (SSD бажаний)	SSD для швидшої обробки логів
Операційна система	Ubuntu Server 22.04 LTS	LTS-релізи з актуальними оновленнями
Порти для доступу	19999/tcp (Netdata), 22/tcp (SSH)	80/443 (опційно для HTTPS-проксі)
Спосіб інсталяції	Пряма (без Docker)	Рекомендується для прозорості та простоти
Мережевий доступ	Локальний або через VPN	HTTPS та проксі-сервер для публічного доступу

РОЗДІЛ 2 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ

2.1 Архітектура впроваджуваної системи

Для впровадження системи віддаленого моніторингу комп'ютерної мережі підприємства необхідно реалізувати просту, але водночас гнучку й ефективну архітектуру, яка забезпечує безперервне спостереження за станом основних серверів, сервісів та мережевих вузлів у режимі реального часу, при цьому не створюючи надмірного навантаження на інфраструктуру. У центрі такої архітектури лежить використання Netdata як локального агента, інстальованого безпосередньо на серверну операційну систему Ubuntu Server без використання контейнеризації або будь-якої додаткової віртуалізації, що дає змогу уникнути зайвих ускладнень у розгортанні й обслуговуванні (див. рис. 2.1).

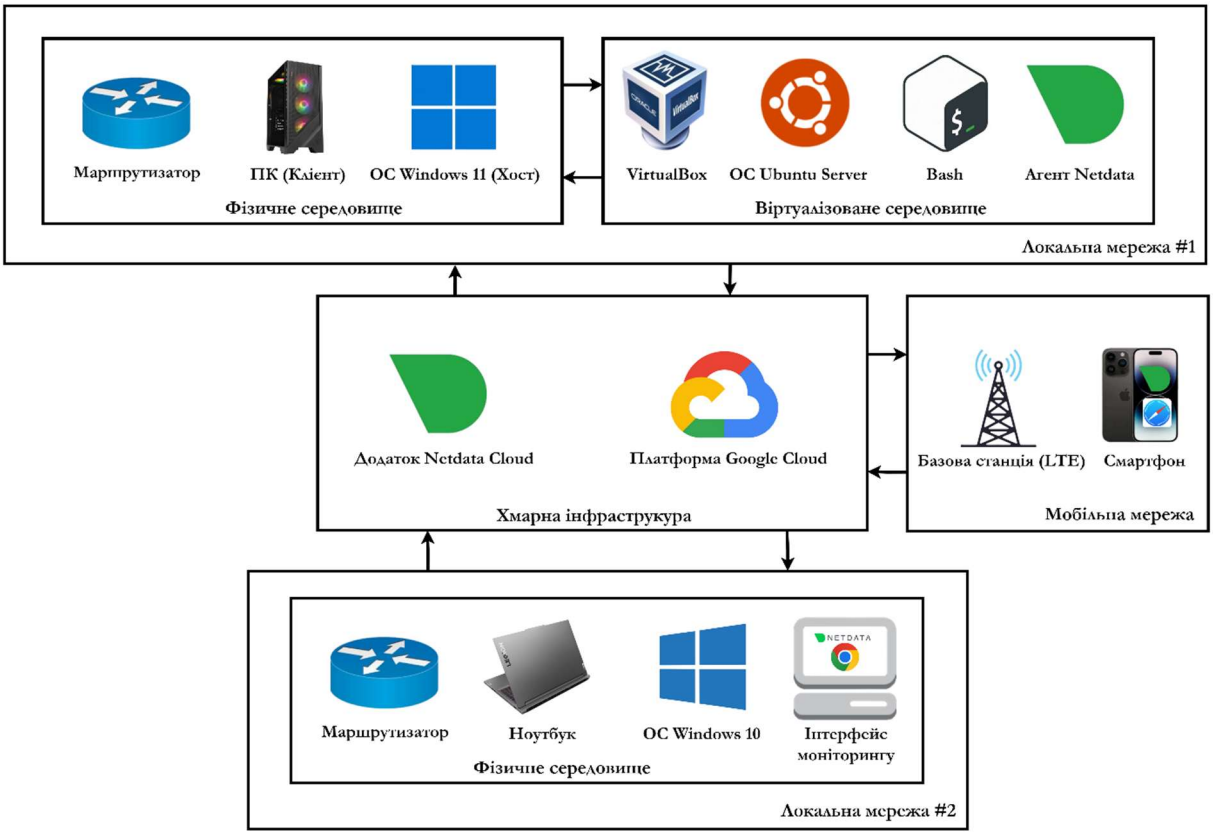


Рисунок 2.1 – Архітектура системи віддаленого моніторингу комп'ютерних мереж

Такий підхід дозволяє досягти високої прозорості, мінімального рівня абстракції між системою моніторингу та самою ОС, а також забезпечує гнучке масштабування та спрощене налагодження з боку адміністратора, що особливо важливо для малих та середніх підприємств [20-23, 31].

Побудова системи на базі VirtualBox та Ubuntu Server. Сама архітектура в цьому випадку реалізована на базі віртуальної машини VirtualBox, що виконується на хост-системі з Windows 11, хоча ця інформація радше допоміжна, оскільки ключовим компонентом виступає саме інсталяція Ubuntu Server у якості системи, на якій функціонуватиме агент моніторингу. Вибір Ubuntu Server як базової платформи обумовлений низкою переваг, серед яких стабільність, широке співтовариство, регулярні оновлення та загальна сумісність з більшістю сучасних інструментів моніторингу. Крім того, серверна версія Ubuntu не містить графічного середовища, що дозволяє суттєво знизити споживання оперативної пам’яті та інших системних ресурсів, а також пришвидшує час запуску й реагування системи в умовах обмеженого обчислювального середовища [21-22].

Локальний агент Netdata — встановлення на сервері підприємства. У межах побудови архітектури системи моніторингу комп’ютерної мережі підприємства доцільним стало використання простого, але функціонального підходу з орієнтацією на ефективність розгортання, прозорість процесів і швидкий доступ до основних показників систем. Така архітектура передбачає встановлення агента моніторингу Netdata безпосередньо на сервер підприємства на базі Ubuntu Server, який працює у середовищі віртуалізації VirtualBox під керуванням операційної системи Windows 11. Відмова від використання контейнеризації або додаткових хмарних компонентів дозволяє уникнути складної інфраструктурної надбудови, мінімізувати кількість точок відмови та спростити керування, що особливо важливо для підприємств з обмеженими ресурсами або початковими етапами впровадження систем моніторингу [22].

На етапі побудови системи серверна частина реалізується на базі Ubuntu Server, вибір якого обґрунтований поєднанням стабільності, широкої підтримки спільноти, мінімальних системних вимог і зручності конфігурації. Серверна версія

Ubuntu не містить графічного середовища, що знижує споживання ресурсів і дозволяє сконцентрувати доступні обчислювальні потужності на обробці й передачі моніторингових даних. Окрім того, вже після інсталяції система надає доступ до таких базових інструментів, як командна оболонка, менеджер пакетів apt та утиліти для налаштування мережі та служб. Після завершення встановлення операційної системи виконується базова конфігурація:

— налаштовується мережеве з’єднання;
— забезпечується SSH-доступ;
— виконується оновлення системи.

У цій архітектурі віртуальна машина або сервер виконує роль єдиного вузла, який приймає та обробляє всі метрики, що надходять локально.

Реалізація інструменту моніторингу ґрунтується на використанні локального агента Netdata, який не вимагає окремого серверного бекенду чи централізованої бази даних, тому його встановлення зводиться до запуску офіційного інсталяційного скрипта після попереднього оновлення системи. Сам процес встановлення триває декілька хвилин, після чого Netdata автоматично запускається як системний сервіс, що стартує при кожному перезавантаженні. Автоматичне оновлення за бажанням користувача може бути ввімкнено або вимкнено, а мінімальне використання системних ресурсів — приблизно 100 МБ оперативної пам’яті — дозволяє розгортати рішення навіть на пристроях із обмеженим обсягом ресурсів. Відсутність зовнішніх залежностей, таких як бази даних, брокери повідомлень чи хмарні API, забезпечує незалежність агента та зменшує кількість потенційних точок збоїв. Вже після встановлення Netdata негайно починає збір та візуалізацію системних показників, зокрема:

— завантаження процесора;
— використання пам’яті;
— активності дискової підсистеми;
— мережевого трафіку;
— запущених процесів.

Доступ до інтерфейсу через браузер. Доступ до даних моніторингу здійснюється через вбудований веб-інтерфейс, що автоматично активується разом з агентом і доступний за замовчуванням через порт "19999". Для входу до системи достатньо вказати IP-адресу сервера у будь-якому сучасному браузері, що робить рішення зручним для використання без потреби у додатковому програмному забезпеченні (див. рис. 2.2).

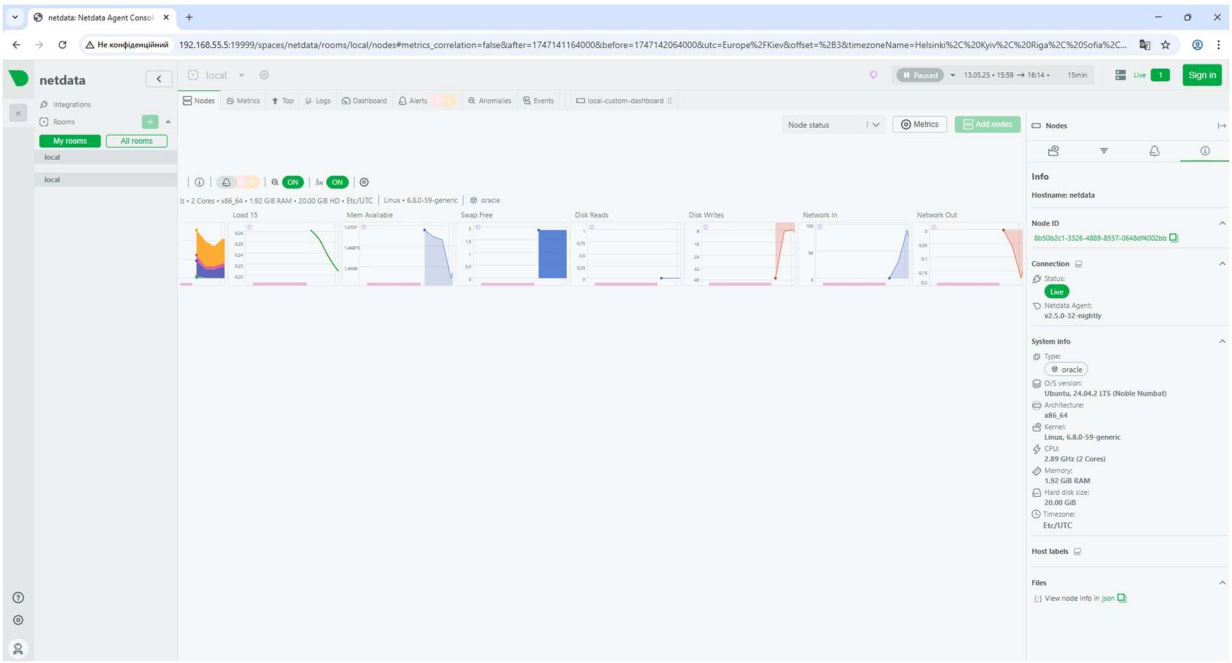


Рисунок 2.2 – Веб-інтерфейс Netdata у браузері

Інтерфейс оновлює дані щосекунди, що дозволяє отримувати актуальні метрики в реальному часі у вигляді динамічних графіків. Присутні інтерактивні можливості — вибір проміжків часу, навігаційне меню, пошук за метриками — що спрощує аналіз даних навіть для нетехнічних користувачів. Доступ до інтерфейсу може бути організований як у межах локальної мережі підприємства, так і віддалено через VPN, SSH-тунелювання або, за потреби, шляхом налаштування HTTPS Reverse Proxy з маршрутизацією через публічний домен. Така гнучкість забезпечує повноцінну можливість інтегрувати систему моніторингу до існуючої IT-інфраструктури без істотних змін або витрат на адаптацію, надаючи зручний інструмент контролю за станом сервера без необхідності взаємодії з терміналом чи

складними моніторинговими стеком на кшталт Prometheus та Grafana, які потребують більше зусиль на встановлення й конфігурацію [23].

Відсутність контейнеризації — пряме розгортання через shell-скрипт. У межах реалізації цього проєкту було свідомо відмовлено від використання контейнеризації, включаючи Docker, Snap, LXC та інші подібні форми ізоляції, на користь прямого встановлення через shell-скрипт, що дозволило одразу зосередитися на спрощенні процесу розгортання без потреби створення докерфайлів, окремих мережевих шарів, зберігання образів чи додаткових конфігурацій. Такий підхід є елементарним у реалізації, оскільки встановлення обмежується лише завантаженням і виконанням офіційного скрипта Netdata, який інсталює всі необхідні компоненти безпосередньо у систему, що дозволяє відразу розпочати роботу без додаткових дій або залежностей [20].

У процесі прямого встановлення з shell-скрипта Netdata працює як системна служба "systemd" і використовує лише стандартні бібліотеки операційної системи, що вже наявні в базовій інсталяції Ubuntu Server, тому відсутня потреба в додаткових інструментах на кшталт Docker Engine чи Podman, що знижує складність підтримки, що особливо актуально у середовищах із обмеженими ресурсами, таких як віртуальні машини або малопотужні фізичні сервери, де критично важливо мінімізувати навантаження на CPU та оперативну пам’ять. Крім того, відсутність контейнерного шару спрощує доступ до конфігураційних файлів, які розміщуються у звичних директоріях на кшталт "/etc/netdata", і це дозволяє легко редагувати параметри, здійснювати резервне копіювання або інтегрувати Netdata з системними механізмами логування через "journalctl" (див. рис. 2.3) [20, 22, 31].

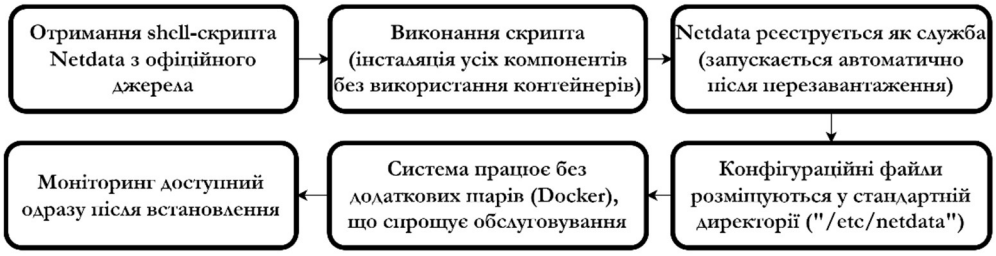


Рисунок 2.3 – Принцип розгортання системи без Docker-контейнера

Такий підхід також виявився зручним у навчальному контексті, оскільки дає змогу наочно продемонструвати роботу служби моніторингу, пояснити її структуру, відслідковувати місцезнаходження конфігурацій, зрозуміти процес оновлення, запуску та взаємодії з системою, що додає прозорості в адмініструванні та дозволяє точно контролювати всі аспекти функціонування агента без прихованих шарів чи абстракцій, властивих контейнеризованим рішенням.

У підсумку, обрана модель реалізації передбачала найпростішу форму інсталяції, орієнтовану на максимальний контроль і доступність. Віртуальна машина на базі VirtualBox була використана як середовище розгортання, де встановлено Ubuntu Server, на який безпосередньо за допомогою офіційного скрипта був розгорнутий агент Netdata. Усі необхідні функції моніторингу стали доступними відразу після завершення інсталяції, а веб-інтерфейс відкривався у браузері без потреби в будь-яких додаткових інструментах. Такий підхід дав змогу забезпечити стабільну роботу системи моніторингу з мінімальними витратами часу, ресурсів та технічної підтримки, що робить його доцільним рішенням для малих підприємств, освітніх установ і будь-яких середовищ, де важливо зменшити технічне навантаження при збереженні функціональності та керованості [31].

2.2 Побудова та конфігурування моніторингового середовища

Побудова моніторингового середовища для спостереження за комп’ютерною мережею починається з налаштування програмного середовища, яке забезпечує:

- контролювати стан системних ресурсів;
- фіксувати навантаження на сервер;
- виявляти зміни;
- реагувати на критичні ситуації.

Усе це задля мінімізації будь-яких ризиків нестабільності роботи інфраструктури. У межах цієї роботи для реалізації системи моніторингу було обрано програму

Netdata, яка встановлюється безпосередньо на сервер або віртуальну машину, як у даному випадку, через офіційний shell-інсталятор. Внаслідок цього створюється стабільне середовище автоматизованого збору метрик, що відображаються у вигляді динамічного веб-дашборду, доступного через браузер без необхідності встановлення додаткових клієнтських програм або середовищ візуалізації.

Першим кроком у розгортанні стало створення віртуальної машини у VirtualBox, для чого спочатку необхідно було встановити сам VirtualBox, якщо він не був попередньо інстальований у системі. Після цього було створено нову віртуальну машину з відповідною конфігурацією, що відповідає мінімальним вимогам для роботи Ubuntu Server та Netdata. У цьому випадку було обрано конфігурацію з двома процесорними ядрами, 1 ГБ оперативної пам’яті та 20 ГБ дискового простору, чого цілком достатньо для демонстраційного або дослідницького розгортання. Після створення машини був завантажений образ операційної системи Ubuntu Server 22.04 LTS, яка відзначається стабільністю та сумісністю з більшістю сучасних серверних рішень, а також широкою підтримкою спільноти (див. рис. 2.4).

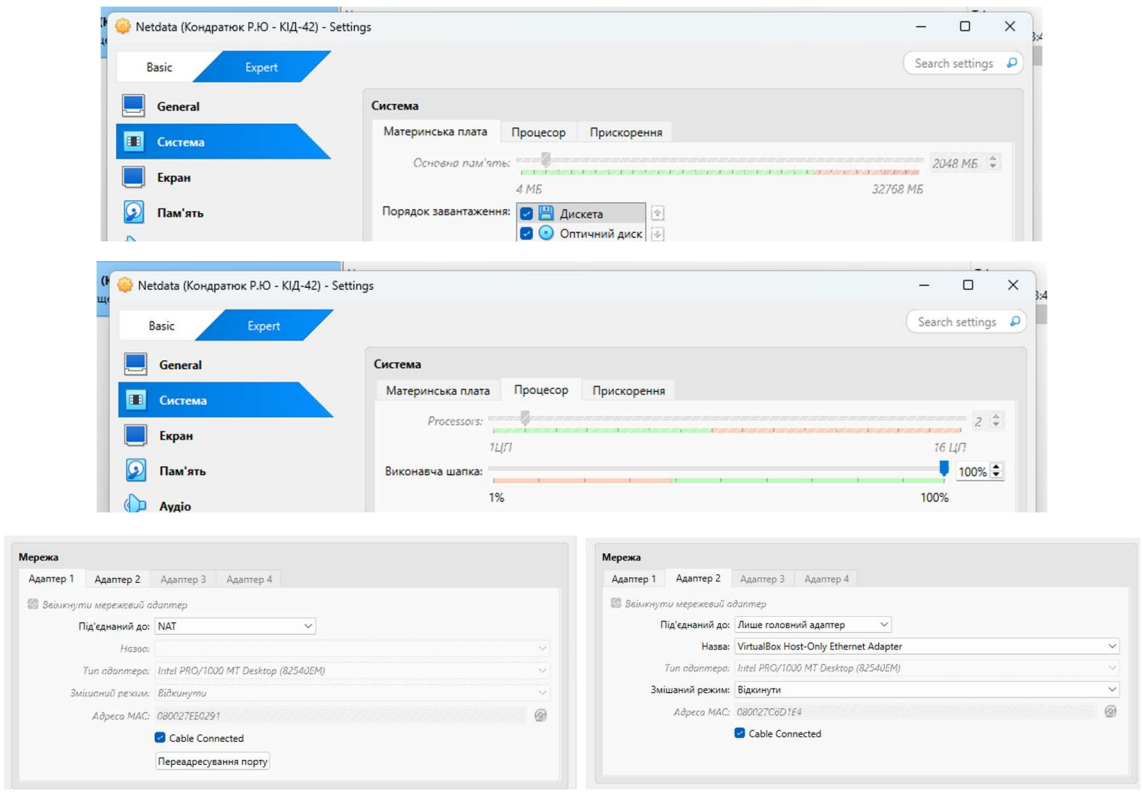


Рисунок 2.4 – Базова конфігурація віртуального середовища системи

Встановлення Netdata за допомогою офіційного bash-інсталятора. Netdata надає спрощений та водночас потужний спосіб інсталяції, який реалізується через офіційний bash-інсталятор, що повністю автоматизує процес розгортання без потреби у ручному встановленні залежностей, компіляції коду або редагуванні конфігураційних файлів, завдяки чому користувач отримує повністю налаштовану систему з мінімальними зусиллями. Для успішного запуску інсталятора потрібно попередньо підготувати систему, що передбачає оновлення наявних пакетів операційної системи та перевірку стабільності інтернет-з'єднання, оскільки всі компоненти завантажуються з офіційного джерела, а сама підготовка реалізується простою командою "sudo apt update && sudo apt upgrade -y", яка оновлює список пакетів та виконує оновлення системи до актуального стану (див. рис. 2.5) [20].

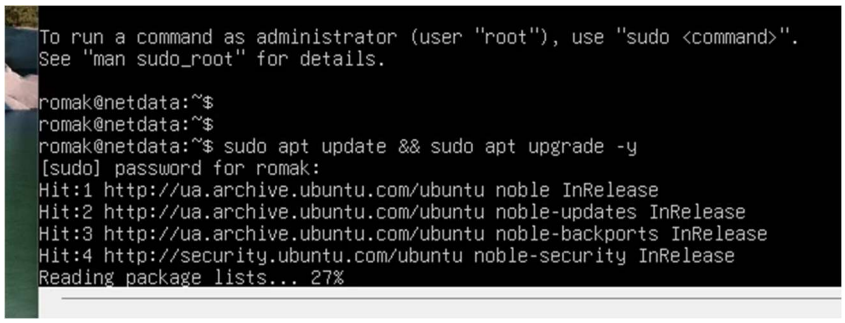


Рисунок 2.5 – Оновлення пакетів Ubuntu Server

Після завершення цього етапу можна одразу переходити до запуску самого інсталятора, який викликається однією командою "bash <(curl -Ss https://my-netdata.io/kickstart.sh)" і не потребує попереднього збереження файлів чи будь-яких ручних дій. Під час виконання скрипта система самостійно визначає поточну операційну систему та архітектуру, після чого виконується завантаження останньої стабільної версії Netdata з офіційного репозиторію, а також встановлення всіх необхідних залежностей, зокрема бібліотек, утиліт та компонентів, потрібних для запуску основного демона (див. рис. 2.6).

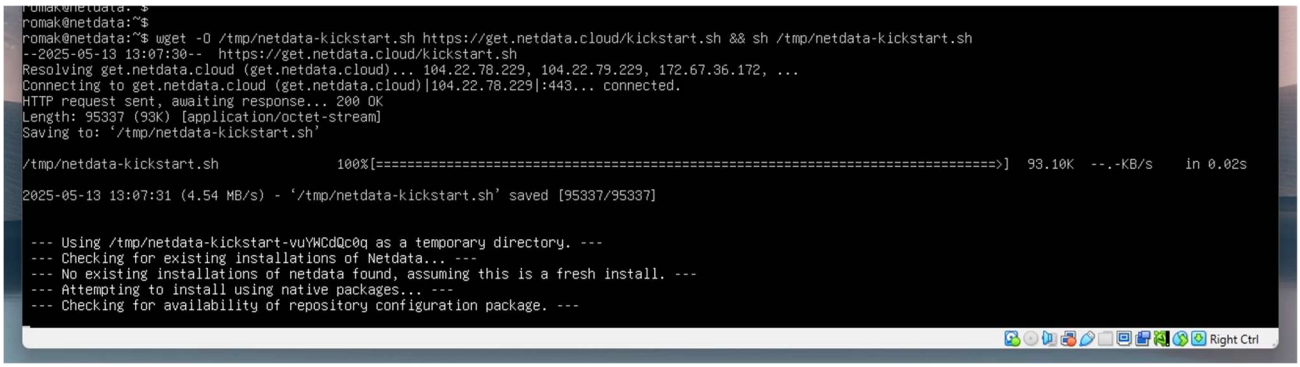


Рисунок 2.6 – Інсталяція Netdata через офіційний bash-інсталятор

Інсталятор також автоматично реєструє Netdata як системну службу через "systemd", що гарантує автозапуск після кожного перезавантаження, і відразу після цього створюється локальний веб-інтерфейс для доступу до зібраних метрик, який запускається на стандартному порту "19999" (див. рис. 2.7) [20]

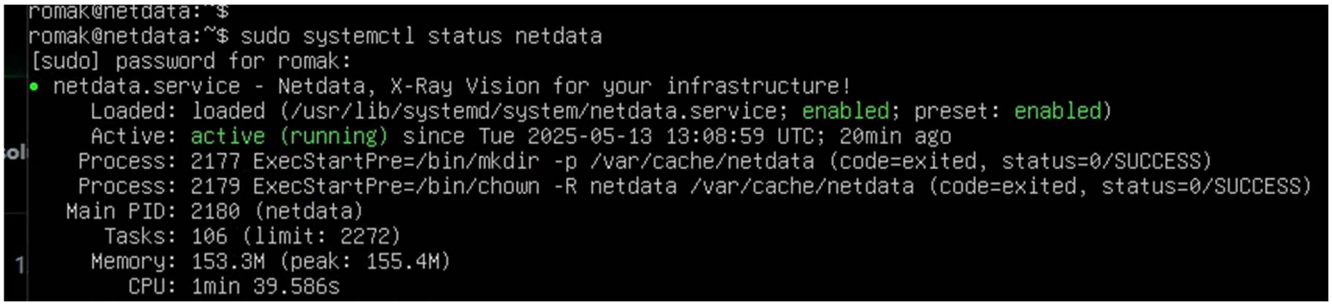


Рисунок 2.7 – Перевірка роботи Netdata як системної служби

У підсумку після завершення інсталяції Netdata одразу починає працювати як системна служба, самостійно активується та розпочинає моніторинг усіх доступних системних компонентів, включно з процесором, оперативною пам’яттю, файлами, мережею, дисками, службами та окремими процесами. Весь процес відбувається у фоновому режимі без потреби додаткового втручання, проте у разі потреби можна перевірити статус запущеної служби командою "sudo systemctl status netdata", що дозволяє переконатися у її коректному функціонуванні або виявити можливі помилки при запуску, якщо вони виникли. Такий спосіб інсталяції забезпечує максимально швидке та стабільне розгортання системи моніторингу, що одразу готова до використання без необхідності налаштування з нуля.

Автоматичне виявлення та моніторинг системних ресурсів. Однією з основних переваг Netdata є вбудований механізм автоматичного виявлення системних ресурсів, за станом яких здійснюється моніторинг без потреби у ручному конфігуруванні чи попередньому визначенні параметрів. Після першого запуску система одразу розпочинає процес збору метрик у реальному часі, охоплюючи ключові компоненти операційного середовища. Зокрема, серед першочергових категорій відображаються параметри центрального процесора, де демонструється загальне навантаження, розподіл завантаження по ядрах, частоти роботи кожного ядра, кількість контекстних перемикань, час очікування на операції введення-виведення, а також IO Wait, що дозволяє швидко оцінити, чи перебуває процесор у стані перевантаження, і чи існують фонові процеси, які споживають надмірну кількість обчислювальних ресурсів (див. рис. 2.8) [22].



Рисунок 2.8 – Приклад автоматичного виявлення ресурсів

Також автоматично виявляється і стан оперативної пам’яті з деталізацією по таких показниках, як обсяг зайнятої та вільної пам’яті, кількість кешованих і буферизованих даних, ступінь використання swap-розділу, сторінкові помилки та інші супутні параметри, що дозволяє робити висновки про витоки пам’яті або

нестачу ресурсів у пікові моменти навантаження. На додачу до цього система відображає метрики мережевих інтерфейсів, серед яких можна знайти швидкість прийому (RX) і передачі (TX) трафіку, активність по кожному фізичному або віртуальному інтерфейсу, кількість оброблених пакетів, виявлені помилки, а також загублені або пошкоджені пакети, що критично важливо для серверів, які функціонують у ролі маршрутизаторів чи фаєрволів із високим навантаженням на мережу.

Крім цього, окремим блоком реалізовано аналіз процесів із виведенням списку найресурсоємніших за споживанням CPU, їхніх ідентифікаторів, статусу та навантаження у розрізі користувачів, що дозволяє виявляти аномальні або завислі процеси без необхідності запускати додаткові утиліти на кшталт "top" чи "htop". У веб-інтерфейсі також відображаються дані про файлову підсистему: завантаження дисків, доступність точок монтування, час доступу до носіїв, рівень "IO latency", а у випадку підтримки апаратного моніторингу — температура накопичувачів через "lm-sensors". Додатково збирається інформація про середнє системне навантаження (load average), системні події на кшталт "interrupt" та "context switch", які вказують на інтенсивність взаємодії процесів з ядром операційної системи.

Усі зазначені дані збираються шляхом регулярного опитування відповідних API, зокрема з директорій "/proc" та "/sys", після чого вони обробляються та виводяться у вигляді динамічних графіків у зручному інтерфейсі, що дозволяє оперативно реагувати на зміни стану системи, виявляти проблеми продуктивності та оцінювати вплив окремих компонентів на загальну стабільність серверного середовища.

Перший запуск та доступ до дашборду. Після завершення встановлення та переконання в коректності роботи Netdata необхідно перевірити доступність веб-інтерфейсу, який запускається автоматично після інсталяції і функціонує через вбудований веб-сервер на порту "19999", тому доступ до дашборду здійснюється безпосередньо через браузер за IP-адресою сервера, що забезпечує зручність перегляду навіть без додаткових налаштувань. У разі успішного запуску відкривається головна сторінка з інтерфейсом Netdata, де ліворуч розташоване

меню з категоріями метрик, які охоплюють процесор, оперативну пам’ять, дискову підсистему, мережеві інтерфейси, активні процеси та інші системні компоненти, а центральна частина дашборду заповнена динамічними графіками, що автоматично оновлюються щосекунди й дозволяють оперативно реагувати на зміни стану ресурсів (див. рис. 2.9) [23].

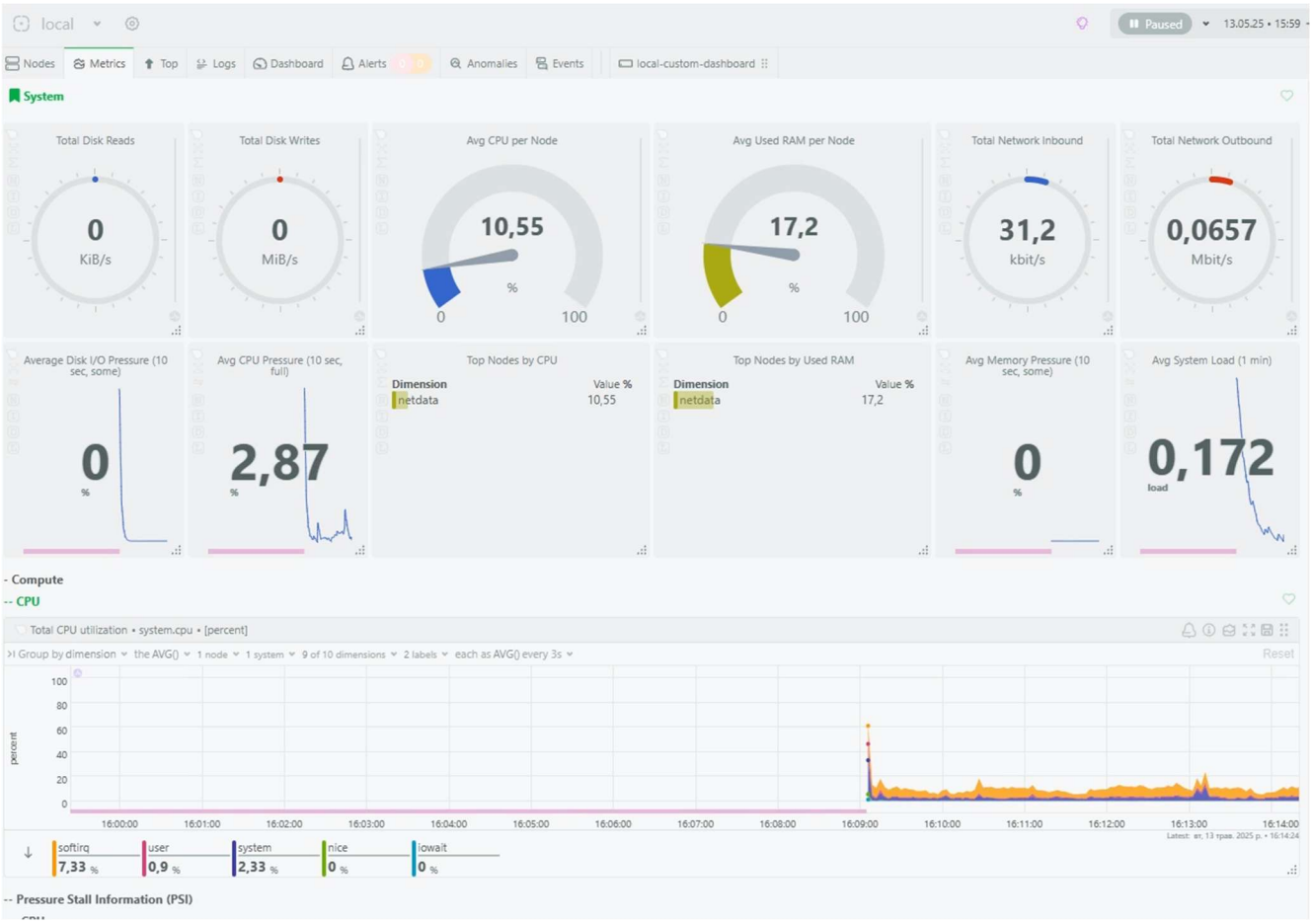


Рисунок 2.9 – Дашборд Netdata зі метриками після першого запуску

Передбачена функція пошуку значно полегшує навігацію по великій кількості графіків і дозволяє швидко знайти необхідний параметр чи процес, а колірна індикація рівнів навантаження на системні ресурси забезпечує візуальне розрізнення стану: зелений для нормального, жовтий для підвищеного і червоний для критичного рівня. Інтерфейс також дозволяє змінювати часовий масштаб графіків, розширюючи його від останніх кількох хвилин до цілої години, а також зберігати окремі графіки у вигляді зображень або завантажувати дані у форматі

JSON через IP. У разі використання кількох вузлів можлива навігація між ними, що реалізується через централізований доступ до інформації з усіх хостів, проте на даному етапі ця функція не активна через наявність лише одного вузла у тестовому середовищі [23].

З точки зору безпеки доступ до веб-інтерфейсу Netdata за замовчуванням відкритий лише в межах локальної мережі, і зовнішні підключення заборонені, якщо не вжито спеціальних заходів, оскільки віртуальна машина, на якій розгорнуто систему, використовує адаптер у режимі Host-Only, що додатково обмежує мережеву видимість. Для організації віддаленого доступу до моніторингового інтерфейсу рекомендовано використовувати VPN, SSH-тунелі або проксирування через зворотний проксі-сервер, що забезпечить як безпеку з’єднання, так і можливість централізованого спостереження за мережею підприємства з віддаленого розташування. Додатково можна обмежити доступ до порту Netdata за допомогою налаштувань файрволу, механізмів на основі IP або активувати вбудовану авторизацію у випадку потреби більш детального контролю за доступом [23].

Таким чином, побудова середовища моніторингу на основі Netdata дозволяє швидко й без надмірних зусиль реалізувати ефективну систему спостереження як за окремим сервером, так і в перспективі — за розподіленими інфраструктурними вузлами, причому всі необхідні етапи встановлення та запуску були автоматизовані офіційним інсталятором, завдяки чому одразу після інсталяції відкривається доступ до візуалізації сотень метрик, включно з процесорними, дисковими, мережевими та операційними даними. У межах продемонстрованого прикладу було реалізовано лише базову інсталяцію без глибоких інтеграцій з іншими сервісами, проте вже цього достатньо, щоб продемонструвати, як на основі одного повноцінного інструмента побудувати стабільну, інформативну систему моніторингу, яка за потреби може бути масштабована й доповнена новими функціональними можливостями відповідно до зростаючих вимог інфраструктури.

2.3 Алгоритми обробки та відображення даних

Ефективне функціонування системи моніторингу інфраструктури передбачає не лише її побудову та розгортання, але й постійний процес збирання, обробки, аналізу та візуалізації інформації, яка надходить від об’єктів спостереження в реальному часі. У такому підході важливим є забезпечення не тільки фіксації змін, але й створення умов для їх оперативного інтерпретування та прийняття відповідних рішень. Система Netdata у цьому контексті забезпечує повний цикл роботи з даними — від моменту їх отримання до відображення на інтерактивних панелях, включаючи буферизацію, збереження та можливість інтеграції з механізмами виявлення аномалій, що дозволяє значно підвищити якість контролю за станом комп’ютерної мережі [12, 15-16, 24-25, 32].

Принцип роботи Netdata – збирання, буферизація та візуалізація. Сама архітектура Netdata передбачає роботу у вигляді агента, який встановлюється безпосередньо на вузол системи моніторингу. У рамках цієї роботи використано сервер на базі Ubuntu у середовищі віртуалізації VirtualBox, де Netdata функціонує як локальний агент, який у реальному часі здійснює моніторинг ключових параметрів, таких як завантаження процесора, використання оперативної пам’яті, активність дискових підсистем, мережевий трафік, стан процесів і доступність служб. Частота оновлення даних у середньому становить одну секунду, при цьому сама система практично не створює істотного навантаження на ресурси, завдяки чому забезпечується стабільна робота основних служб на сервері [12, 32].

Наступним етапом є буферизація даних, яка реалізується через внутрішню систему збереження тимчасової інформації під назвою Time Series Database Engine. На відміну від традиційних рішень, де передбачається зберігання даних у зовнішніх СУБД, Netdata використовує власну високошвидкісну архітектуру збереження у оперативній пам’яті, що дозволяє забезпечити миттєвий доступ до історії параметрів без затримок. Уся інформація структурується за принципом кругової ротації (Round Robin), що означає автоматичне стискання та агрегацію даних при

досягненні певного об’єму, зменшуючи таким чином потребу у великих обсягах постійної пам’яті й зберігаючи при цьому необхідну точність для довготривалого аналізу [12, 24, 32].

Фінальним етапом є візуалізація, яка реалізується через інтерактивну веб-панель, доступну з будь-якого браузера у локальній мережі. Завдяки цьому користувач отримує динамічний інтерфейс, у якому графіки оновлюються щосекунди, демонструючи поточний стан системи з високим рівнем деталізації. Така панель дозволяє не тільки спостерігати за окремими метриками, але й порівнювати показники між собою, змінювати інтервал часу, масштабувати відображення та зберігати візуалізації у вигляді зображень (див. рис. 2.10).

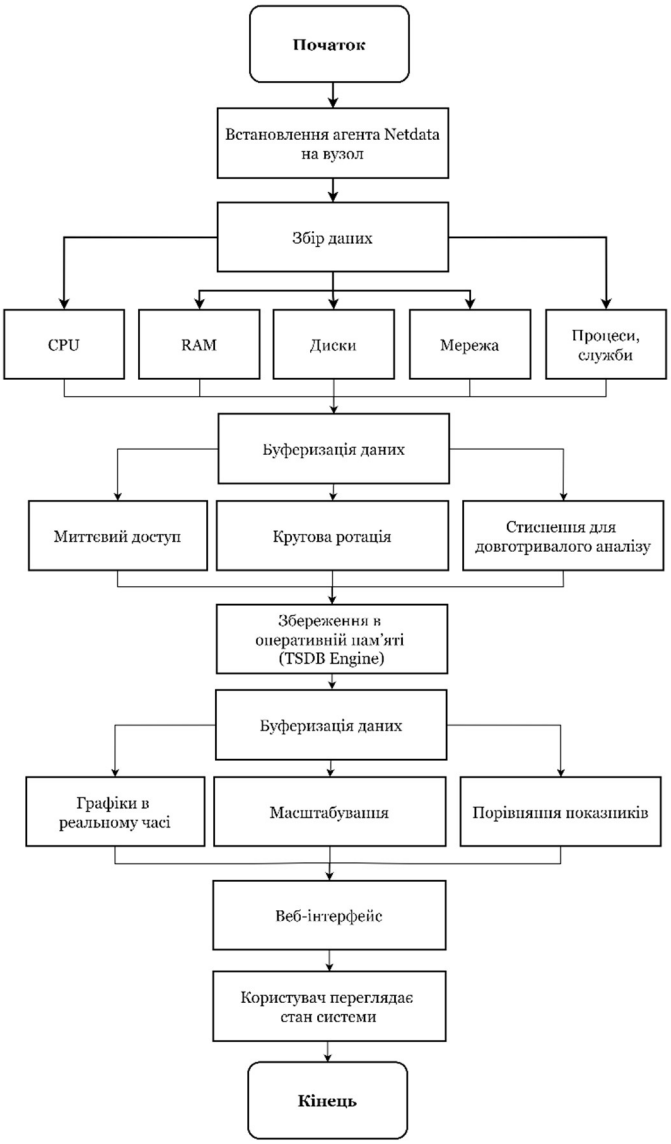


Рисунок 2.10 – Загальний принцип роботи системи Netdata

Усе це формує цілісну модель алгоритмів роботи Netdata, яка заснована на послідовності етапів збору, буферизації та подання даних, що забезпечує просту та зрозумілу архітектуру аналітичного інструменту для моніторингу комп'ютерних мереж у віддаленому або локальному режимі [32].

Виявлення аномалій – вбудовані механізми Netdata. Розглядаючи загальну архітектуру та функціональні можливості системи моніторингу Netdata, важливо звернути увагу на вбудовані механізми виявлення аномалій, які реалізовано на основі простих, але достатньо ефективних алгоритмів, що не потребують складного налаштування або зовнішніх аналітичних модулів. У цьому підході застосовується концепція "health checks", що передбачає використання набору правил контролю за станом системи, які описуються у вигляді граничних значень для кожної з доступних метрик. Ці правила задаються у спеціальних YAML-конфігураційних файлах, що зберігаються у відповідному каталозі "health.d" в структурі Netdata, де кожен файл містить опис окремих умов, за яких значення показника вважається нормальним, попереджувальним або критичним [25].

Виконання перевірок відбувається автоматично в реальному часі з частотою раз на секунду, завдяки чому система здатна оперативно фіксувати порушення у поведінці того чи іншого компонента інфраструктури. Кожне правило має пов'язаний із ним статус:

1) Green відповідає нормальному стану.
2) Yellow сигналізує про потенційне відхилення.
3) Red означає критичну ситуацію, що потребує негайної реакції.

Ці стани також відображаються у веб-інтерфейсі у вигляді індикаторів, що змінюють колір відповідно до активного статусу конкретної метрики або групи параметрів, завдяки чому користувач може одразу візуально оцінити загальний стан системи та локалізувати джерело проблеми (див. рис. 2.11) [25].

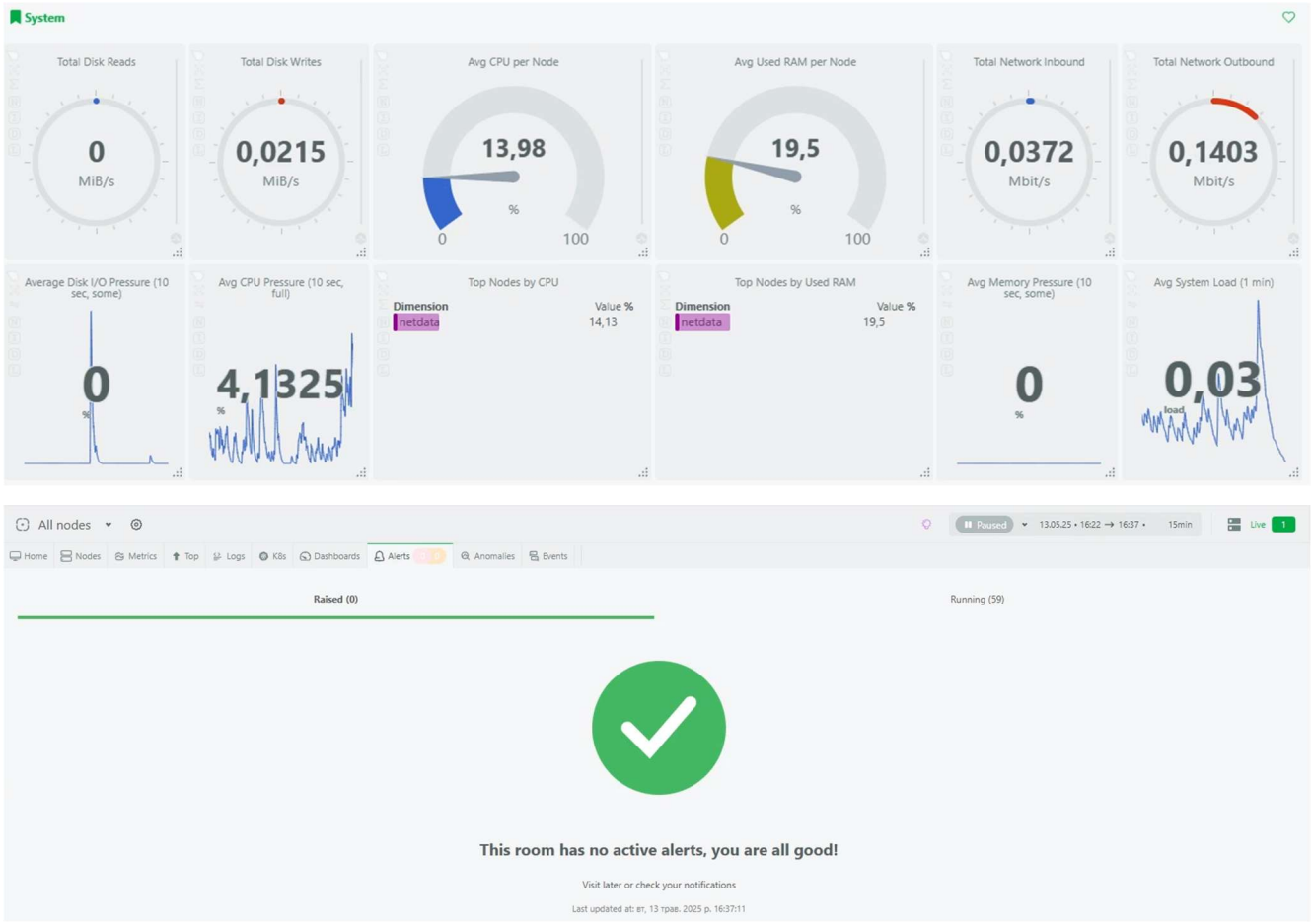


Рисунок 2.11 – Механізм виявлення аномалій у реальному часі

Механізм правил, закладених у Netdata, може бути як дуже загальним, так і точково адаптованим під специфіку конкретної інфраструктури. Наприклад, перевищення завантаження процесора понад 90% протягом десяти секунд може розглядатися як критичне навантаження, що запускає відповідне сповіщення. У той же час можливе використання більш тонких підходів до оцінки стабільності — зокрема, за допомогою методу під назвою "statistical anomaly detection", який оцінює відхилення поточних значень від базових трендів, виявлених на основі історичних даних. Цей метод дозволяє виявляти більш складні аномалії, не пов’язані з різким перевищенням порогів, а, наприклад, з поступовим, але систематичним зростанням споживання ресурсів або зміною характеру навантаження, яке може свідчити про витік пам’яті, неефективне використання дискових операцій чи збої у фонових процесах (див. табл. 2.1) [25].

Таблиця 2.1 – Порівняння механізмів виявлення аномалій у системах моніторингу

Характеристика	**Netdata**	**Zabbix**	**Prometheus (з Alertmanager)**
Тип виявлення	Пороговий та статистичний	Пороговий, складні вирази	Правила (PromQL), зовнішній аналіз
Частота перевірок	Щосекунди	Вказується вручну (зазвичай ≥1 хв)	Залежить від scrape-інтервалу (зазвичай 15–60 с)
Формат правил	YAML-файли (/etc/netdata/health.d/)	Веб-інтерфейс або XML-конфігурації	YAML-файли для Alertmanager
Рівні сповіщення	Green / Yellow / Red	OK / Warning / High / Disaster	Warning / Critical або кастомні рівні
Реакція у реальному часі	Графіки та сповіщення	Залежить від частоти перевірок	Залежить від scrape/alert інтервалів
Налаштування сповіщень	Просте, централізоване (health_alarm_notify.conf)	Гнучке, але складне	Через Alertmanager
Підтримка самонавчання/трендів	Пілотне statistical anomaly detection)	Відсутня	Відсутня, можна додати з зовнішніми інструментами

Алгоритми сповіщення – гнучке налаштування оповіщень. Розглядаючи далі архітектуру системи Netdata, особливу увагу заслуговує механізм оповіщень, який забезпечує гнучке реагування на проблеми без необхідності постійного перегляду веб-інтерфейсу. Така система побудована на централізованій логіці, яка передбачає взаємодію з широким спектром комунікаційних каналів, серед яких підтримуються електронна пошта через SMTP, месенджери Discord, Slack, Telegram, Microsoft Teams, мобільні системи сповіщень, такі як Gotify, а також інші сервіси, що підтримують вебхуки. Завдяки такій багатоканальній підтримці можлива інтеграція у вже існуючі середовища сповіщення, які використовуються у корпоративному або локальному масштабі, що дозволяє не дублювати інструменти, а доповнювати вже наявні рішення [15].

Алгоритм, що лежить в основі логіки сповіщення, умовно розбивається на кілька послідовних кроків. Спершу система фіксує тригер події, яка виявляється через механізм "health check" відповідно до заздалегідь визначених правил; далі формується шаблонізоване повідомлення, що містить ключову інформацію про проблему, включаючи назву вузла, тип інциденту, точний час виявлення, опис метрики, що спрацювала, та за наявності — рекомендацію щодо подальших дій. Після цього повідомлення надсилається через визначений канал зв’язку, а в разі збереження критичного стану воно може бути повторно доставлене через певні інтервали часу, доки інцидент не буде вирішено або не змінить свій статус [15, 25]. Усе конфігурування здійснюється за допомогою спеціального конфігураційного файлу "health_alarm_notify.conf", який знаходиться у каталозі налаштувань Netdata. У цьому файлі можна увімкнути або вимкнути конкретні канали, налаштувати параметри для кожного з них, включно з URL вебхука, іменем відправника, типом подій, що активують повідомлення, і навіть деталізацією метрик. У межах цієї конфігурації допускається встановлення окремих каналів оповіщення для різних категорій подій — наприклад, можна налаштувати надсилання повідомлень про інциденти з CPU до одного месенджера, повідомлень про стан мережі — до іншого, а сповіщення про дискові операції — ще до третього.

На прикладі умовного сценарію, у якому завантаження процесора перевищує 90% упродовж 30 секунд, система автоматично згенерує повідомлення про інцидент, де буде вказано назву вузла, тип спрацювання (наприклад, CPU utilization critical), час виявлення та інші технічні деталі, після чого передасть його, скажімо, до каналу Discord, якщо такий шлях визначено у відповідній секції конфігурації. У разі наявності детальніших налаштувань шаблонів повідомлення також може містити інструкції або посилання на документацію з усунення типових причин перевантаження (див. рис. 2.12).

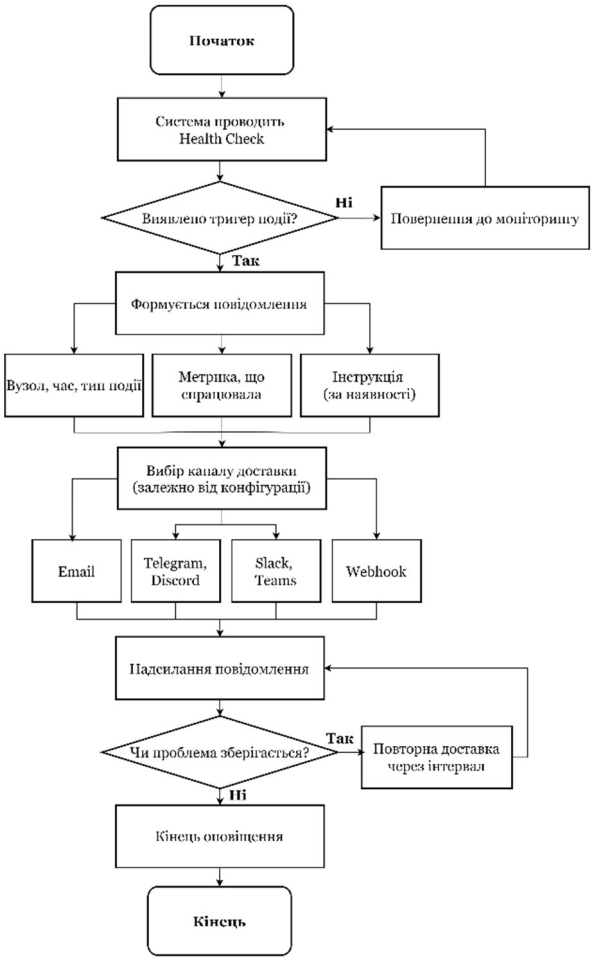


Рисунок 2.12 – Алгоритм оповіщення про події

Такий підхід дозволяє мінімізувати час реагування на інциденти навіть у ситуаціях, коли немає безпосереднього доступу до системи моніторингу або коли потрібно делегувати контроль за певними показниками окремим відповідальним особам, які отримуватимуть лише релевантні їм сповіщення через зручні для них канали.

Побудова графіків навантаження – аналіз мережевих піків. Окремий аспект моніторингу в системі Netdata пов’язаний з аналізом навантаження на мережеві інтерфейси, що реалізується через високоточні графіки для кожного мережевого каналу, таких як "usernet0", "veth", "ens" та інші, і ці графіки відображають завантаженість у відсотковому співвідношенні. Унікальність такого підходу полягає в тому, що завдяки оновленню даних у режимі реального часу щосекунди вдається миттєво зафіксувати піки навантаження, не втрачаючи критичних моментів для діагностики. При цьому плавність кривих на графіках забезпечується інтерполяцією, яка дозволяє спростити візуальне сприйняття без спотворення суті даних, і разом з можливістю вибору діапазону часу для перегляду — від хвилини до доби — утворює зручний інструмент аналітики. У межах одного екрану можна одночасно виводити кілька показників, таких як завантаження CPU, обсяг споживаної оперативної пам’яті та стан мережі, що суттєво полегшує комплексну оцінку продуктивності системи або окремого вузла (див. рис. 2.13) [16].

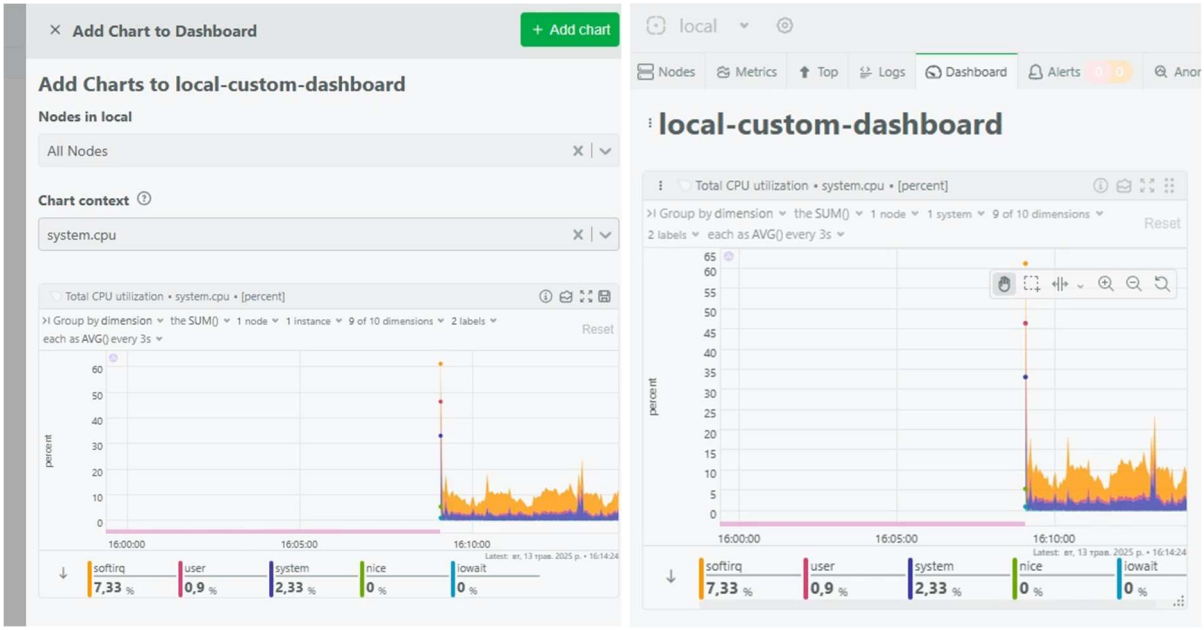


Рисунок 2.13 – Побудова графіків навантаження процесора

У процесі аналізу таких графіків часто виявляються періоди підвищеної активності, які пов’язані, наприклад, з виконанням резервного копіювання, завантаженням оновлень, сплесками запитів, потенційними атаками або ж із

надмірним використанням ресурсів окремими пристроями в мережі. Саме таке візуальне відображення дозволяє майже миттєво локалізувати проблему, яка в інших умовах залишилася б непоміченою, особливо коли йдеться про короткочасні, але критичні піки, що впливають на стабільність усієї інфраструктури. З огляду на такі функції графічного аналізу можна говорити про його ефективність як інструмента для діагностики, прогнозування і навіть оптимізації поведінки мережевих сервісів.

У підсумку розгляд можливостей Netdata дозволяє сформувати уявлення про неї як про комплексне рішення для збору, обробки й візуалізації інформації щодо системних і мережевих параметрів (див. табл. 2.2).

Таблиця 2.2 – Етапи обробки та відображення даних у Netdata

Компонент	**Опис**
Збирання метрик	Агент отримує дані з системних інтерфейсів і сервісів щосекунди
Буферизація	Зберігання у пам’яті з використанням Time Series DB (власний движок)
Візуалізація	Інтерактивні графіки у Web UI, оновлення в реальному часі
Виявлення аномалій	Health checks з YAML-правилами, автоматичне спрацювання
Статистичний аналіз	Оцінка трендів і відхилень для раннього виявлення проблем
Система сповіщень	Notification Engine з підтримкою Email, Discord, Telegram тощо
Мережеві графіки	Трафік, помилки, піки з можливістю масштабування

Завдяки інтелектуальному буферизованому збору даних, алгоритмам виявлення аномалій, гнучким механізмам сповіщення та деталізованим графікам навантаження забезпечується цілісний і одночасно адаптивний підхід до моніторингу. Реалізація в реальному часі дозволяє досягати максимальної точності

у відстеженні змін, що особливо важливо для оперативного реагування в умовах сучасних IT-інфраструктур, де будь-яка затримка або нестабільність може безпосередньо впливати на безперервність бізнес-процесів. У контексті підприємств, які опираються на цифрову інфраструктуру, така система є необхідним інструментом постійного контролю і стабільної роботи сервісів.

2.4 Реалізація системи на прикладі тестового середовища

Загальна реалізація системи моніторингу була представлена на прикладі побудови тестового середовища, яке дозволяло наочно дослідити основні етапи впровадження таких рішень у типових умовах [26-27].

Створення тестового середовища. Для цього було створено віртуальне серверне середовище з використанням платформи VirtualBox, що дало змогу змоделювати інфраструктуру, наближену до конфігурацій, притаманних малим підприємствам, без потреби в додаткових апаратних витратах. У рамках цієї моделі в якості основи було обрано операційну систему Ubuntu Server 22.04 LTS, якій було виділено 2 віртуальні процесорні ядра, 1 ГБ оперативної пам’яті та 20 ГБ дискового простору, чого достатньо для демонстрації базових функцій моніторингу та налаштування агента [26].

Мережеве підключення було реалізовано в режимі NAT із пробросом портів, що забезпечило можливість доступу до веб-інтерфейсу Netdata з хостової операційної системи Windows 11 через стандартний браузер, використовуючи IP-адресу віртуальної машини разом із портом "19999". Після завершення встановлення операційної системи, оновлення системних пакетів і встановлення моніторингового агента Netdata за допомогою офіційного скрипта-інсталятора було отримано повністю функціональне середовище, готове до роботи. У цьому середовищі з’явилася можливість одразу після запуску отримати доступ до візуалізації системних і мережевих метрик у реальному часі, спостерігати за станом

процесора, пам’яті, файлової системи, мережевих інтерфейсів, а також тестувати сповіщення та інші функції без необхідності в складному конфігуруванні.

Доступ до системи забезпечувався через звичайний браузер, а налаштування не вимагали глибоких технічних знань, що робить таку модель зручною не лише для тестування, а й як прототип реального розгортання в середовищі локальної мережі підприємства. Такий підхід дозволяє оцінити функціональність системи ще до її фактичної інтеграції у виробниче середовище, даючи змогу адаптувати конфігурації, підготувати сценарії сповіщень, переглянути поведінку системи в умовах навантаження або симульованих проблем, а також напрацювати початкові навички адміністрування. Таким чином, тестове середовище на базі VirtualBox і Ubuntu Server дозволяє не лише реалізувати практичну частину, але й перевірити життєздатність обраного рішення на початковому етапі, перш ніж здійснювати повноцінне впровадження в інфраструктуру реального підприємства.

Демонстрація віддаленого доступу до Netdata. Netdata, орієнтована на швидке розгортання та легкий віддалений моніторинг, не вимагає складного налаштування для доступу до свого веб-інтерфейсу, оскільки його структура передбачає відкриття у будь-якому сучасному браузері на кшталт Chrome, Firefox чи будь-якого іншого сумісного клієнта, що дозволяє максимально спростити процес взаємодії з даними про стан системи. Інтерфейс представляє собою вертикальне меню з категоріями моніторингу, кожна з яких веде до докладних графіків окремих системних або мережевих метрик, де кожна область має позначення допустимих та критичних значень, що дозволяє візуально та миттєво орієнтуватися у відхиленнях або потенційних проблемах, а сама побудова графіків дозволяє легко простежити як загальний стан, так і конкретні піки навантаження в динаміці. У реалізації використовувався Netdata Cloud, де був отриманий та переданий ID для прив’язки зі Google Cloud через акаунт Google та платформу Netdata Cloud, після чого був наданий безкоштовний період на 14 днів, що дозволяє віддалено отримувати дані про моніторинг мережі та інших показників протягом цього періоду. Після можна визначитися зі підпискою або ж використати інший варіант (див. рис. 2.14).

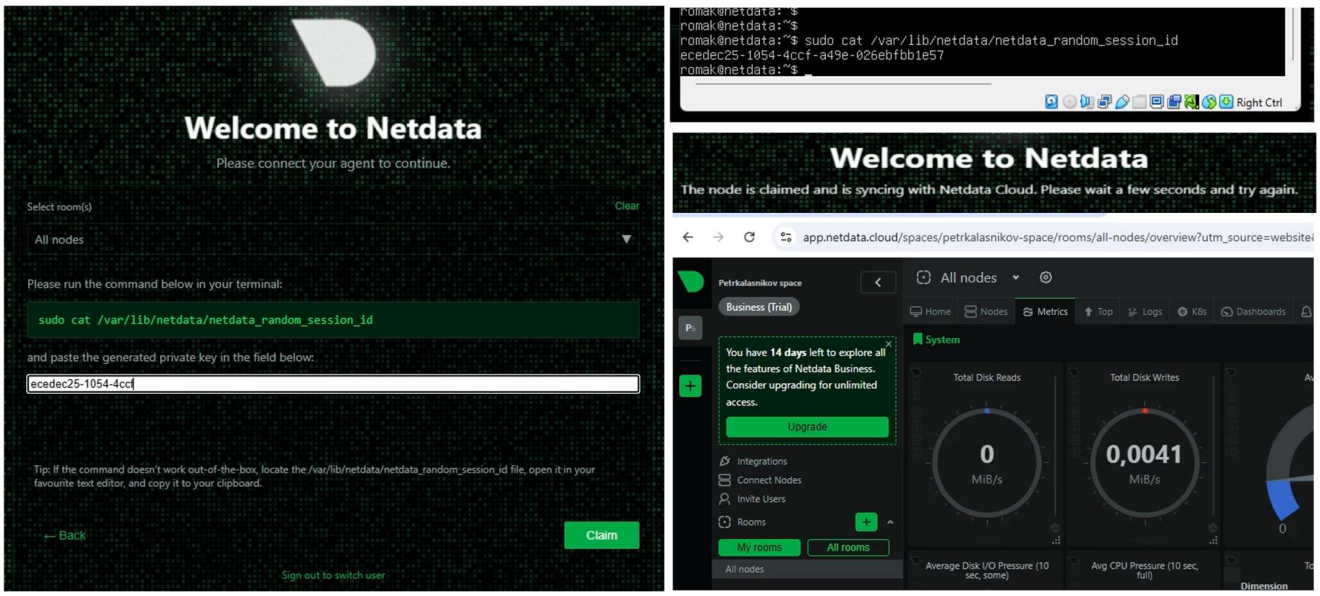


Рисунок 2.14 – Приклад віддаленого доступу до моніторингу

Таке рішення, навіть у межах простого тестового середовища, демонструє здатність швидко та ефективно надавати доступ до живих показників без потреби в додаткових клієнтах або складному розгортанні, оскільки вже з моменту запуску можна працювати з даними практично в реальному часі. Водночас якщо розглядати цю реалізацію крізь призму інформаційної безпеки, то хоча в поточному прикладі доступ не обмежувався, у виробничому середовищі доцільно було б впровадити додаткові механізми захисту, зокрема шифрування трафіку за допомогою HTTPS на основі безплатних TLS-сертифікатів, таких як "Let's Encrypt", що забезпечило б базовий рівень конфіденційності та цілісності переданих даних (див. рис. 2.15) [27].

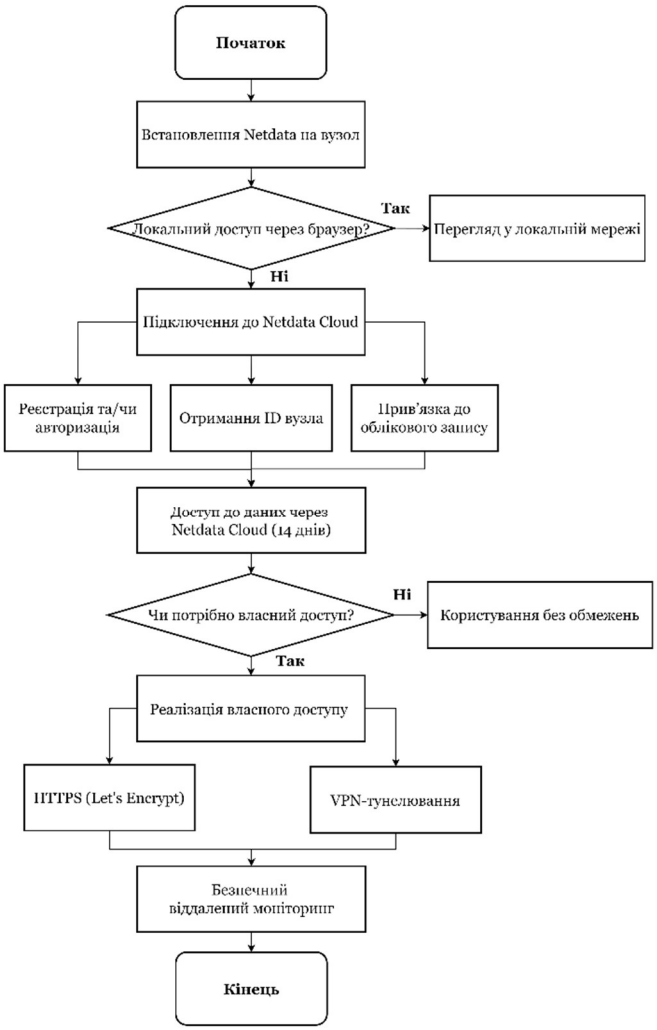


Рисунок 2.15 – Алгоритм впровадження віддаленого доступу до моніторингу

Окрім цього, при розгортанні в межах локальної мережі підприємства або при потребі забезпечити доступ ззовні, можливо реалізувати тунелювання через VPN для захисту каналу зв'язку, що дозволить значно знизити ризики несанкціонованого втручання в систему моніторингу. Таке поєднання відкритості у користуванні та гнучкості щодо варіантів безпечного доступу підкреслює здатність системи не лише наочно відображати технічний стан інфраструктури, а й бути адаптованою до конкретних вимог безпеки підприємства, що робить її придатною як для експериментального тестування, так і для практичного впровадження у виробниче середовище з усіма відповідними вимогами до захисту.

Аналіз роботи системи у різні періоди часу. Завдяки функціоналу, який надає Netdata, можливо ефективно здійснювати аналіз поведінки системи як у режимі реального часу, так і з урахуванням історичних даних, що дозволяє виявляти

характерні зміни навантаження та взаємозв’язки між різними показниками впродовж доби або тривалішого періоду. Під час тестування у межах штучно змодельованого офісного середовища були створені типові сценарії діяльності, зокрема:

— процеси оновлення програмного забезпечення;

— запуск операційної системи;

— активна робота з файлами.

Усе це часто спостерігається у годину пік або в ранковий та вечірній періоди, включаючи також фонову нічну активність.

У таких умовах система демонструє здатність фіксувати короткочасні сплески навантаження, наприклад різке зростання використання процесора під час ініціалізації певного ресурсоємного процесу або суттєве навантаження на дискову підсистему у разі великої кількості операцій читання-запису. Подібні зміни також стосуються інтенсивного мережевого трафіку, коли в години підвищеної активності виявляється помітне збільшення як вхідного, так і вихідного трафіку, що дозволяє оперативно оцінити поточну ситуацію, виявити потенційні вузькі місця в інфраструктурі та передбачити необхідність масштабування ресурсів або оптимізації окремих процесів (див. рис. 2.16).

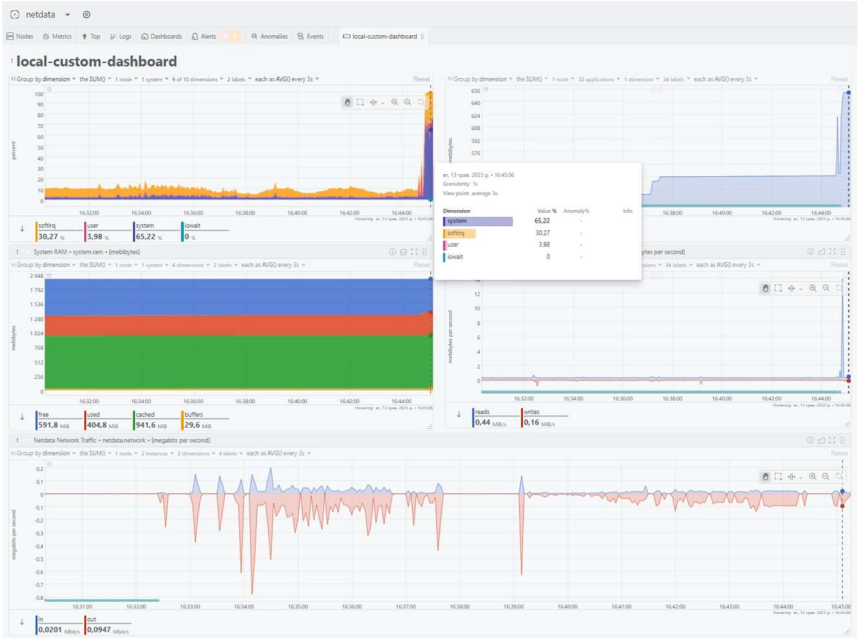


Рисунок 2.16 – Знімок дашборду у період пікового навантаження

Завдяки гнучкому налаштуванню часових діапазонів аналіз можливий як на рівні останніх хвилин, так і в межах більш тривалих відрізків — однієї години, доби, тижня або місяця — що у поєднанні зі збереженням історії моніторингових даних дозволяє виявляти довготривалі закономірності, відстежувати циклічні навантаження, порівнювати поведінку системи в різні дні та адаптувати рішення щодо управління ресурсами відповідно до фактичних даних. Таке спостереження за динамікою дає змогу вчасно виявити неочевидні проблеми, які можуть залишатися прихованими при виключно точковому або короткотерміновому моніторингу, і водночас підтримує прийняття обґрунтованих технічних рішень на основі наочних та легко доступних аналітичних даних.

Приклади метрик та їх інтерпретація. Netdata за своєю архітектурою автоматично збирає понад тисячу різноманітних метрик, серед яких окрему увагу варто звернути на ті, що найчастіше застосовуються в практиці моніторингу мережевих та серверних середовищ. Однією з ключових метрик виступає "CPU usage", яка відображає завантаженість процесора за типами використання: системні процеси, користувацькі додатки та очікування завершення операцій введення-виведення. Такий поділ дозволяє визначати джерела навантаження та оцінювати ефективність використання обчислювальних ресурсів, при цьому різкі стрибки навантаження понад 80% можуть сигналізувати про проблеми з оптимізацією процесів, зокрема про нераціональний розподіл задач або фонові процеси, що споживають надмірно багато часу центрального процесора. Подібне значення мають метрики оперативної пам’яті, серед яких "RAM usage" візуалізує не лише обсяг фактично зайнятої пам’яті, а й частку, відведену під кеш або залишену доступною для нових процесів (див. рис. 2.17).

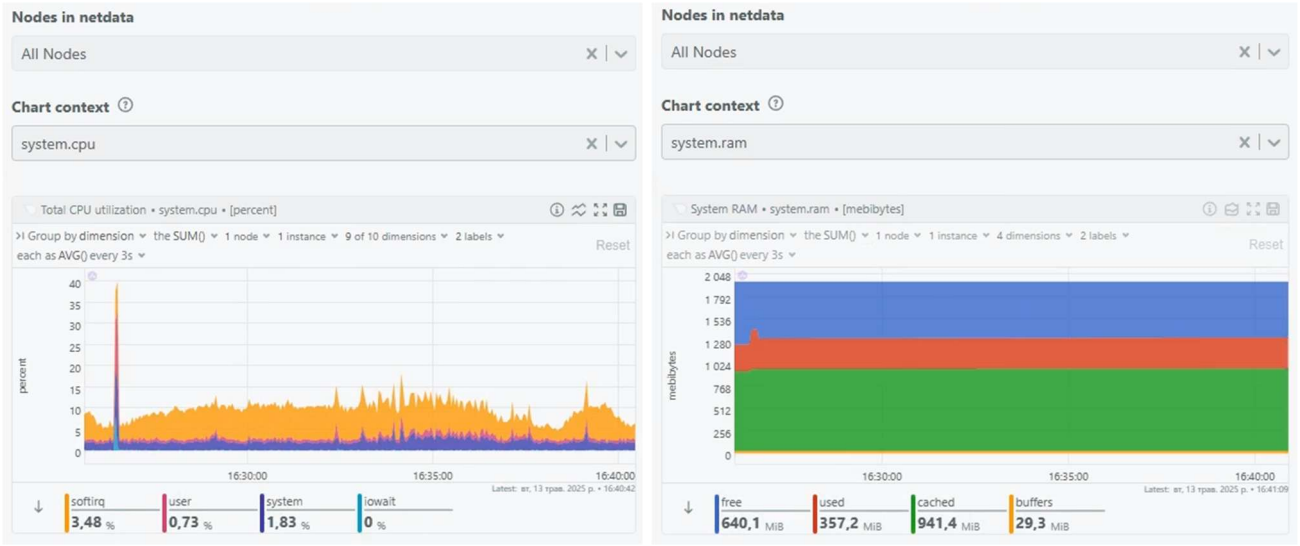


Рисунок 2.17 – Інтерпретація метрик процесора та пам’яті

У цьому контексті важливим є своєчасне виявлення витоків пам’яті або надмірного кешування, що може свідчити про некоректну роботу додатків, які не звільняють ресурси після завершення обробки [12].

Наступним блоком метрик, який становить інтерес з точки зору аналізу стабільності системи, є дискове введення-виведення — показники читання та запису ("read/write operations") дають змогу оцінити навантаження на накопичувачі. Значні затримки у виконанні цих операцій часто свідчать про перевантаження конкретного пристрою зберігання даних, що у свою чергу може викликати загальне уповільнення роботи всієї системи або конкретних сервісів. У випадку з мережею до важливих показників належать метрики інтерфейсів, такі як "transmitted/received' bytes", "packet errors" та "dropped packets" (див. рис. 2.18).

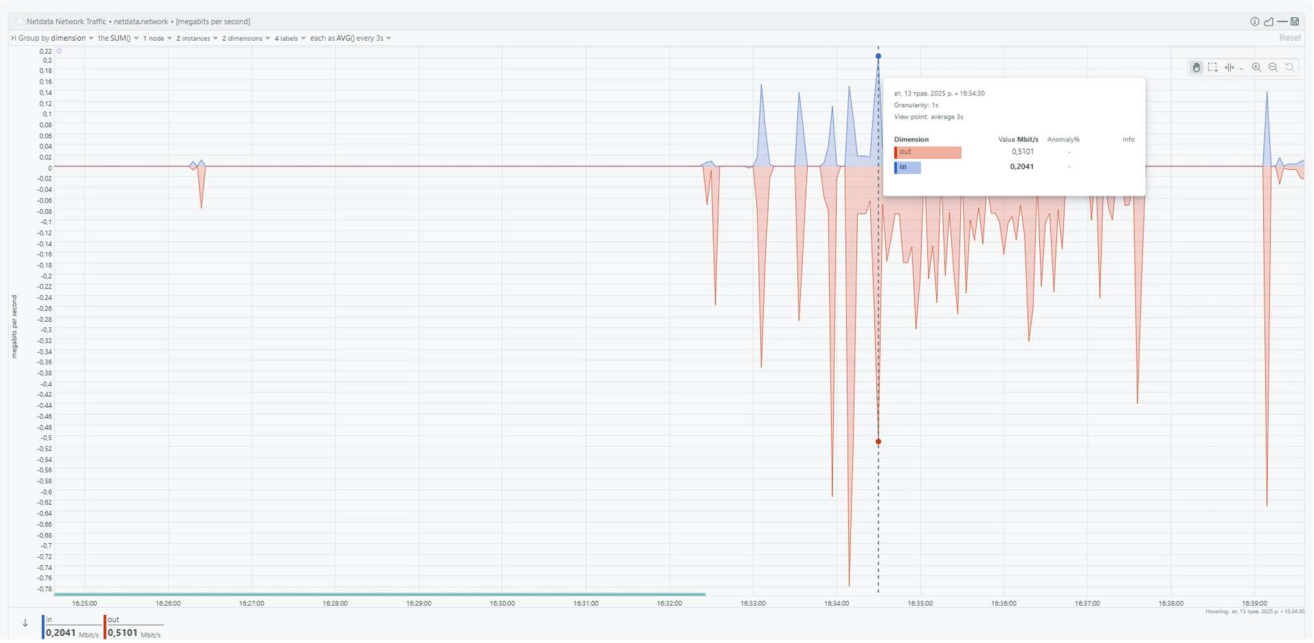


Рисунок 2.18 – Візуалізація метрик мережевого інтерфейсу

Вони дозволяють не тільки оцінювати обсяг переданого та отриманого трафіку, але й виявляти ознаки нестабільної мережевої активності, включаючи піки, втрати пакетів, помилки або повторні передачі, які потенційно можуть свідчити про внутрішні аномалії, зовнішні атаки або неправильну конфігурацію мережевого обладнання (див. рис. 2.19).

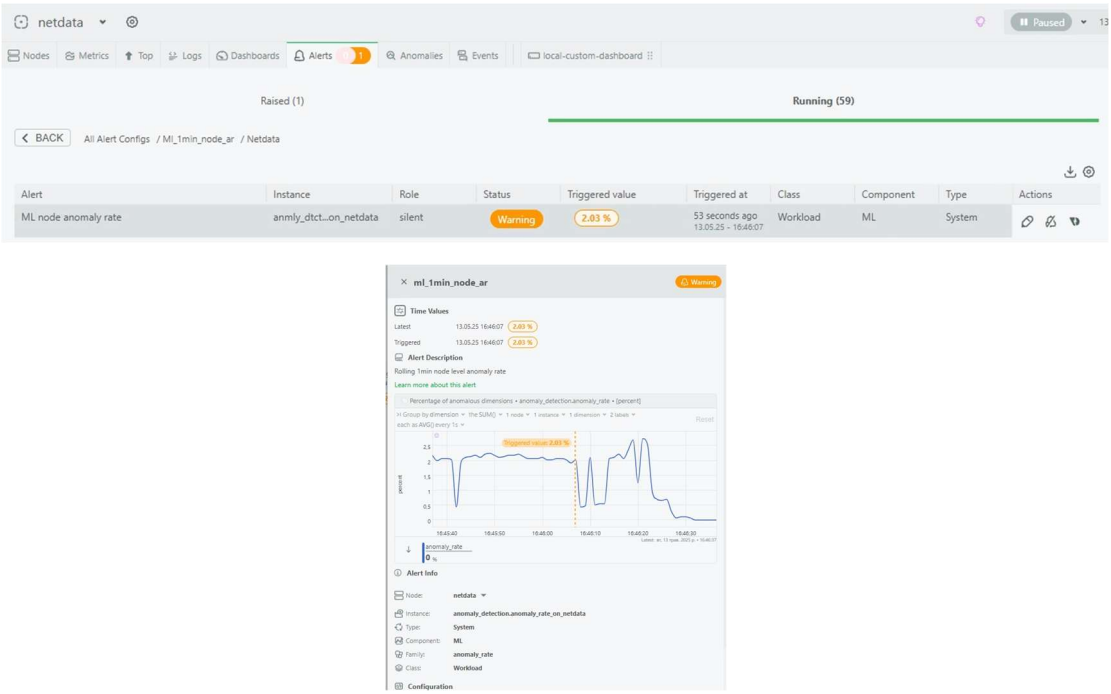


Рисунок 2.19 – Аналіз активності (аномалій) сервісів у часовому розрізі

РОЗДІЛ 3 ПЕРСПЕКТИВИ ВДОСКОНАЛЕННЯ СИСТЕМИ МОНІТОРИНГУ

3.1 Методи покращення ефективності системи

Система моніторингу у тому вигляді, в якому реалізовано її на основі Netdata, забезпечує надзвичайно швидкий запуск, мінімальні вимоги до ресурсів інфраструктури і практично не потребує додаткових налаштувань, що робить її зручною для використання в невеликих середовищах або на етапах тестування. Однак, з огляду на вимоги до масштабованості, забезпечення більш тривалого зберігання метрик, підвищення рівня безпеки і реалізацію повноцінного аналізу поведінки системи у розрізі тривалих періодів, виникає потреба в удосконаленні такої системи за допомогою низки рішень, серед яких ключовими напрямками можна вважати інтеграцію з зовнішніми базами даних, побудову централізованої архітектури, розширення функцій оповіщення та захист веб-доступу через шифрування. Усі ці напрями здатні впливати на різні аспекти функціонування системи, адаптуючи її до потреб конкретного підприємства або масштабу мережі, в якій вона використовується, що дозволяє досягати більшої надійності та ефективності у різних сценаріях застосування [27-28].

Використання зовнішніх баз даних для довгострокового зберігання метрик. Одним із основних обмежень базового варіанту Netdata є зберігання метрик лише в оперативній пам’яті або тимчасових файлах типу ".db", що призводить до автоматичного очищення даних уже за кілька годин або діб, що унеможливлює повноцінний аналіз довготривалих змін у поведінці системи, виявлення сезонних тенденцій, періодичних піків навантаження або точного планування ресурсів. Для розв’язання цієї проблеми реалізовано можливість експорту зібраних даних у зовнішні системи зберігання, зокрема у спеціалізовані "Time Series" бази даних, такі як Prometheus, який чудово підходить для зберігання великої кількості

короткотривалих, але частих спостережень, а також має готові інтеграції з Grafana для побудови дашбордів. У такому сценарії Netdata виступає в ролі endpoint-джерела, а Prometheus виконує періодичне опитування через API "/api/v1/allmetrics", отримуючи повний набір метрик і зберігаючи їх у форматі, зручному для подальшої аналітики. Окрім Prometheus, можливе використання InfluxDB — рішення, яке розроблено спеціально для обробки "Time Series" і дає змогу формувати дуже гнучкі запити з використанням складних фільтрів, математичних виразів і навіть агрегованих обчислень. Такий підхід забезпечує архівування даних протягом місяців і років, а не годин, а також дозволяє візуалізувати зміни з високою точністю та деталізацією. Ще одним прикладом є Graphite — перевірене часом рішення, яке особливо добре масштабовується у великих середовищах і здатне обслуговувати десятки тисяч потоків метрик у дата-центрах або у хмарних кластерах (див. рис. 3.1) [28].

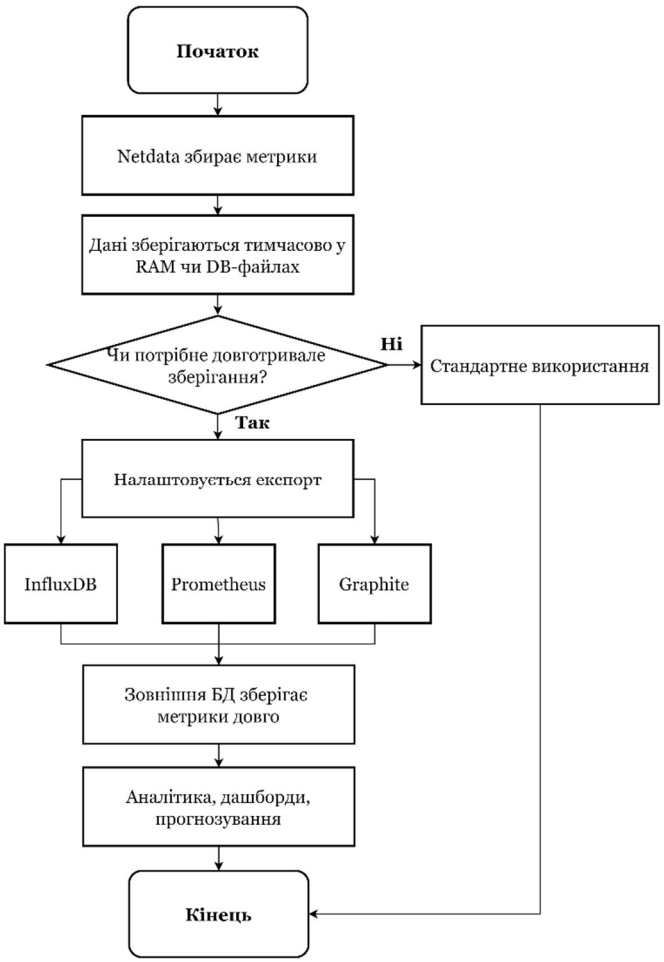


Рисунок 3.1 – Метод інтеграції з базою даних для архівації метрик

Інтеграція з подібними зовнішніми рішеннями дозволяє повністю змінити підхід до моніторингу: замість короткотермінового перегляду поточного стану система перетворюється на аналітичний інструмент для ретроспективного аналізу, моделювання навантаження, прогнозування несправностей і планування оновлень. При цьому сама реалізація такого сценарію не потребує кардинального перепроєктування системи: достатньо встановити ту ж InfluxDB або Prometheus на окремий сервер чи контейнер, а в налаштуваннях Netdata вказати відповідні параметри експорту — наприклад, для InfluxDB використовується конфігурація в секції "backends", де вказується URL, токен доступу і формат передачі. Після цього кожен агент Netdata надсилає дані не лише у локальну пам’ять, а й на зовнішнє сховище, де вони зберігаються протягом визначеного часу і стають доступними для складних запитів, аналітики або побудови дашбордів, що значно підвищує ефективність моніторингу в умовах реального середовища та надає підприємству засоби для стратегічного планування технічної інфраструктури [28].

Розгортання централізованого рішення з декількома агентами. Базовий варіант використання системи моніторингу передбачає окреме встановлення агента на кожному хості з доступом до кожного інтерфейсу локально, що створює ізольовану модель без єдиної централізації. Однак для підвищення ефективності, масштабованості й оперативності обробки інцидентів доцільно впровадити централізовану модель, у якій усі агенти передають зібрані дані до головного вузла або до зовнішнього агрегатора. Така модель дозволяє об’єднати всі вузли в одну систему спостереження, усунути розрізнені веб-інтерфейси та спростити адміністрування в цілому (див. рис. 3.2).

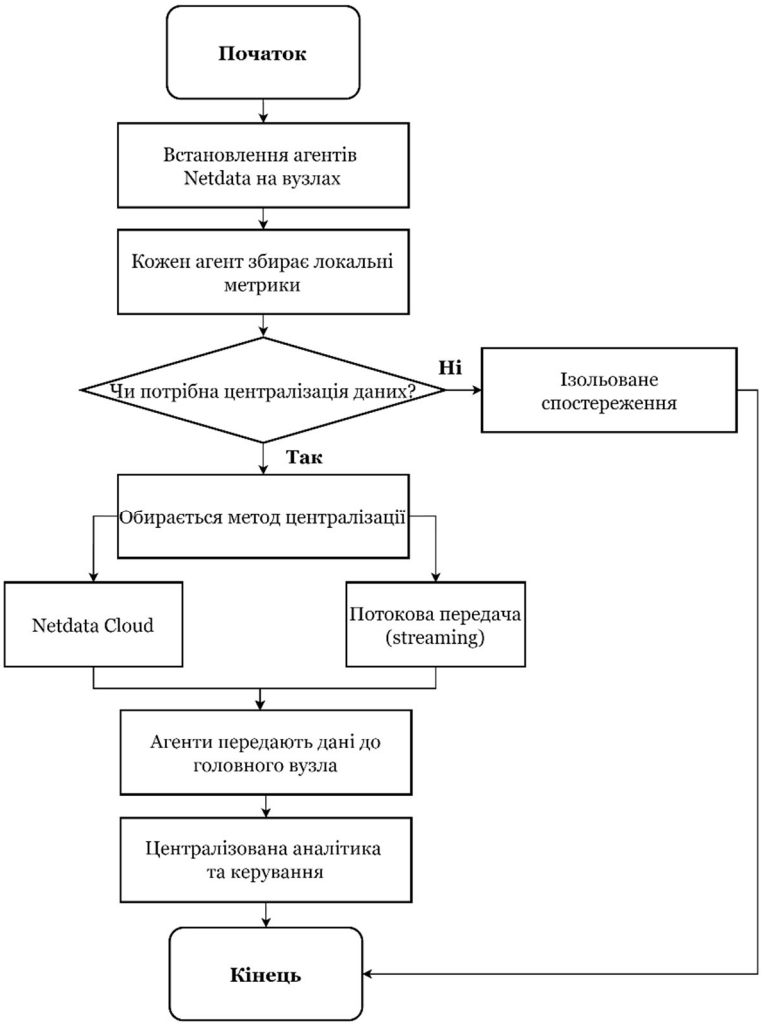


Рисунок 3.2 – Метод використання централізованої архітектури з кількома агентами Netdata

Одним із варіантів централізації є використання хмарного сервісу Netdata Cloud, який дає змогу об’єднати кілька агентів в одну керовану панель, спостерігати за всіма підключеними вузлами в реальному часі та отримувати централізовані сповіщення. Перевагою цієї моделі є простота підключення, оскільки агенти обмінюються даними через токен, без необхідності відкриття додаткових портів, що водночас покращує загальну безпеку рішення. Іншим підходом до централізації є використання потокової передачі метрик між агентами локальної мережі — в такому випадку один вузол виконує роль головного колектора, а інші надсилають йому метрики через TCP, після чого історія зберігається централізовано лише на головному сервері.

Упровадження такого сценарію дозволяє позбутися потреби у локальному зберіганні даних на кожному вузлі, що значно знижує навантаження на ресурси окремих машин. Централізація в цьому контексті також дозволяє розділити ролі вузлів, коли головний вузол може мати розширені апаратні характеристики, тоді як інші хости залишаються легкими за вимогами до ресурсів. Такий підхід забезпечує гнучке масштабування — нові вузли можуть бути швидко додані без складної процедури інтеграції чи налаштування кожного окремо, що в загальному знижує як часові, так і матеріальні витрати на обслуговування інфраструктури.

Окремою перевагою централізації виступає консолідація всієї аналітики в єдиній точці доступу, що дає змогу формувати цілісну картину роботи всієї інфраструктури без потреби в перемиканні між різними системами або інтерфейсами. Повна централізована панель спостереження дозволяє отримувати своєчасну інформацію про інциденти, знижує час реакції на критичні події та значно спрощує аналіз причин збоїв чи перевантажень, оскільки всі метрики доступні в одному місці. Такий підхід у підсумку створює більш стійку, контрольовану та зручну в управлінні систему моніторингу з підтримкою подальшого масштабування та гнучкої адаптації до зростаючих потреб.

Підключення додаткових модулів сповіщення. Оперативне реагування на відхилення параметрів системи має критичне значення для підтримки стабільної роботи IT-інфраструктури, особливо в умовах, коли від кожної хвилини залежить безперебійність бізнес-процесів, і саме в цьому контексті механізм оповіщень Netdata демонструє гнучкість та адаптивність до потреб конкретного середовища, дозволяючи налаштовувати систему так, щоб вона відповідала як масштабам, так і особливостям корпоративної структури. Підтримка різних каналів сповіщення, таких як електронна пошта через SMTP, служби на зразок Slack, Discord, Microsoft Teams, Telegram-боти, Webhook-запити або інтеграції з платформами на зразок Zabbix, Siren або Alertmanager, дозволяє побудувати як просту, так і складну логіку повідомлень, яка забезпечує доставку сповіщення до відповідальної особи або групи саме в той момент, коли параметри системи виходять за межі норми (див. рис. 3.3).

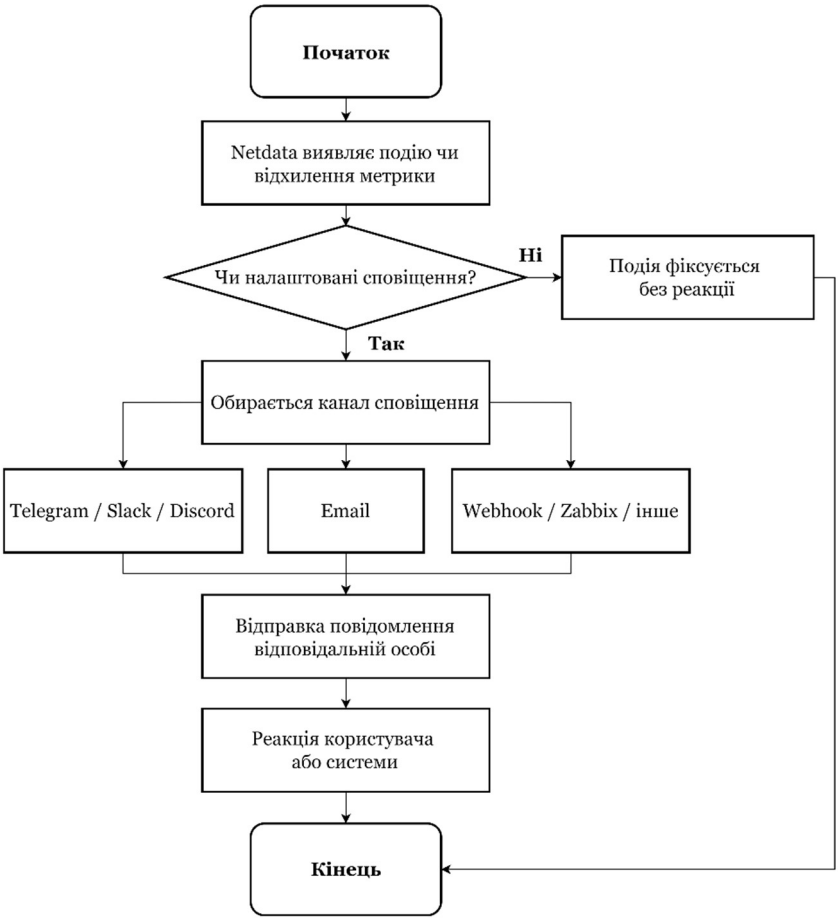


Рисунок 3.3 – Метод додавання зовнішніх модулів сповіщення

Для реалізації цього механізму необхідне попереднє налаштування конфігураційного файлу "health_alarm_notify.conf", де визначаються основні параметри: канали відправки, шаблони повідомлень, рівні пріоритету, включно з категоріями на кшталт "warning" та "critical", час затримки між повторними повідомленнями, умови активації певних каналів тощо, що дозволяє чітко структурувати поведінку системи у разі інциденту. У межах цієї конфігурації також можна розділити канали для окремих вузлів або сервісів, тим самим уникнувши інформаційного перевантаження та забезпечивши адресне реагування.

Після впровадження такої логіки виникає можливість формування чітких сценаріїв дій: у разі перевищення температури процесора понад 85°C відповідальний отримує повідомлення через Telegram-бота із вказівкою на проблему та ліміт, за яким було спрацювання; або, наприклад, якщо відбувається збій у роботі служби бази даних, система негайно надсилає оповіщення до

корпоративного чату у Slack; у випадку, коли обсяг зайнятого дискового простору перевищує 95%, автоматично створюється запит через Webhook до зовнішньої платформи, яка ініціює необхідні дії або ж фіксує інцидент у системі підтримки. Завдяки цьому механізму значно підвищується швидкість реакції на потенційні або вже наявні інциденти, що прямо впливає на зменшення часу простою серверів, запобігання втраті даних або перебоям у роботі критичних служб.

Централізація логіки сповіщення дозволяє не лише зменшити людський фактор, а й запровадити напівавтоматизовану інфраструктуру, здатну самостійно інформувати відповідальних осіб про найменші збої ще до того, як ці збої переростуть у повномасштабні проблеми, що в результаті знижує ризики для підприємства, покращує надійність обслуговування та дозволяє адміністраторам зосередитись на проактивному управлінні замість реактивного реагування.

Встановлення TLS/SSL для захисту веб-доступу. Коли мова заходить про безпеку, моніторингова система стає важливою частиною загальної інформаційної безпеки підприємства, і в контексті використання Netdata як рішення для моніторингу інфраструктури необхідність забезпечення захищеного доступу до її веб-інтерфейсу стає критичною вимогою, адже платформа відкриває детальний доступ до внутрішніх показників серверів, мереж і служб, які можуть бути використані потенційним зловмисником у разі несанкціонованого доступу. Відсутність шифрування під час доступу через HTTP створює можливість перехоплення конфіденційної інформації, включно з обліковими даними, у випадках використання спільних мережевих середовищ, наприклад у локальній мережі офісу або через відкриту Wi-Fi точку, що в сукупності може призвести до серйозних порушень захищеності інфраструктури, зокрема до несанкціонованого аналізу навантаження систем або навіть до зчитування конфігурацій, які самі по собі можуть містити службову інформацію чи критичні внутрішні параметри.

Реалізація захищеного HTTPS-доступу у випадку з Netdata, найчастіше здійснюється через конфігурацію реверс-проксі за допомогою веб-серверів на зразок Nginx або Apache, причому найпоширенішим і найбільш зручним способом є саме встановлення Nginx як фронтенда до Netdata, де на стороні проксі-сервера

реалізується підтримка TLS/SSL, базова HTTP-аутентифікація, а також інтеграція з такими сервісами, як Let’s Encrypt для автоматичного отримання сертифікатів. У результаті формується середовище, в якому доступ до моніторингу може бути відкритий через зашифроване з’єднання, при цьому доступ залишається обмеженим тільки для авторизованих користувачів, а в поєднанні з VPN-доступом така конфігурація дозволяє створити відокремлену внутрішню систему спостереження, із можливістю при цьому отримувати віддалений доступ з будь-якої точки мережі без втрати в безпеці або конфіденційності переданих даних [27].

Застосування подібної моделі не лише дозволяє уникнути критичних вразливостей, а й створює основу для відповідності корпоративним політикам з кібербезпеки, які дедалі частіше передбачають обов’язкове шифрування будь-яких доступів до адміністративних панелей, систем керування або моніторингових інструментів. У випадку розгортання подібного рішення на виробничому або корпоративному середовищі, де постійний контроль за технічним станом серверів, баз даних або мереж є умовою стабільної роботи бізнес-процесів, використання TLS/SSL для захисту веб-доступу вже не є додатковим рівнем безпеки, а перетворюється на обов’язковий базовий стандарт, без якого впровадження системи в експлуатацію не може розглядатися як завершене.

Інтеграція таких інструментів, як реверс-проксі з шифруванням, разом із іншими раніше описаними компонентами — модульною системою сповіщення, адаптивною аналітикою, виведенням метрик у зовнішні бази даних — дозволяє трансформувати Netdata з базового засобу візуалізації у повноцінну систему моніторингу, придатну для використання у середовищах із підвищеними вимогами до аналітики, масштабованості та інформаційної безпеки. При цьому зростає загальна ефективність рішень, що реалізуються на основі цієї платформи: централізація оповіщень прискорює реакцію на інциденти, а шифрування доступу створює умови, в яких конфіденційність і цілісність даних зберігаються навіть за роботи з зовнішніми підключеннями. Саме така комплексна реалізація дозволяє гарантувати не лише технічну працездатність, але й надійність та відповідність

сучасним вимогам щодо безпеки в системі моніторингу комп’ютерних мереж підприємства.

3.2 Порівняння обраного рішення з альтернативами

Під час вибору системи моніторингу комп’ютерної мережі ключовими критеріями зазвичай виступають:

— швидкість розгортання;
— глибина моніторингу;
— гнучкість конфігурації;
— наочність візуалізації;
— підтримка масштабованості.

У цьому контексті доцільно здійснити порівняння обраного рішення на основі Netdata з іншими популярними альтернативами, такими як Zabbix, Nagios або Prometheus [29].

Zabbix проти Netdata – повнота функцій проти простоти. Zabbix являє собою потужну корпоративну платформу, яка підтримує як активне, так і пасивне збирання даних, розвинуту систему тригерів, сценарії реагування, шаблони моніторингу, підтримку SNMP, IPMI, логів, зовнішніх скриптів та інтеграцію з широким спектром систем сповіщення. Наявність шаблонів для типових пристроїв суттєво спрощує розгортання у великих інфраструктурах, а збереження історичних метрик у базі даних типу MySQL або PostgreSQL дозволяє здійснювати аналіз за значний період, включаючи побудову звітів за роки. Крім того, можливість інтеграції з системами типу Service Desk або CMDB додає цінності для великих підприємств, де важливе централізоване управління інфраструктурою [29].

Попри це, Zabbix має певні обмеження, головним серед яких є складність налаштування: розгортання вимагає глибокого розуміння конфігурації сервера, баз

даних, знання SQL та підготовки середовища для зберігання метрик. Початковий запуск часто затягується, а інтерфейс виявляється менш наочним у порівнянні з рішеннями, орієнтованими на візуалізацію в реальному часі. На цьому фоні Netdata позиціонується як рішення, яке надає оперативний доступ до системних метрик із мінімальними налаштуваннями, автоматичним виявленням сервісів і високошвидкісною обробкою інформації. У практичному використанні Netdata дозволяє запускати моніторинг одразу після інсталяції, що особливо актуально для невеликих проєктів, стартапів або ізольованих середовищ внутрішнього IT-обслуговування, де головним пріоритетом є швидкий запуск, простота адміністрування та візуальна зручність.

Проведення прямого порівняння за критеріями розгортання, масштабованості, зберігання історії, візуалізації та впливу на систему дозволяє глибше зрозуміти їхні відмінності. У випадку простоти налаштування очевидною перевагою володіє Netdata, яка не вимагає складних конфігурацій і практично не потребує ручного втручання, тоді як Zabbix потребує серйозної підготовки та часу. За критерієм масштабованості Zabbix має більший потенціал завдяки використанню проксі та шаблонів, однак Netdata компенсує це можливістю використання хмарної платформи Netdata Cloud або реалізацією Stream-сервера для централізованого збору даних з кількох вузлів. Щодо збереження історії, Zabbix демонструє беззаперечну перевагу завдяки використанню повноцінних баз даних, які можуть зберігати роками історію метрик для глибокої аналітики, тоді як Netdata обмежується лише короткочасним збереженням та потребує додаткових рішень для архівування [29].

Візуалізація в Zabbix реалізована у вигляді таблиць, статичних графіків і дешбордів, що хоча й функціонально, проте менш динамічно, ніж у Netdata, де графіки оновлюються в реальному часі та дозволяють миттєво виявляти проблеми без додаткових дій. За рівнем навантаження на систему Netdata значно випереджає Zabbix, оскільки її архітектура оптимізована під мінімальне споживання ресурсів, працює в реальному часі без глибоких обчислень у фоновому режимі, не потребує

важких сервісів баз даних і не створює істотного навантаження навіть у розгортаннях на обмежених за ресурсами вузлах [29].

Nagios – хороший для подій, але не для реального часу. Серед альтернатив Netdata доцільно розглянути систему моніторингу Nagios, яка є однією з найстаріших і водночас найбільш впізнаваних у сфері контролю за станом IT-інфраструктури. Ця система побудована на концепції регулярного опитування сервісів згідно з визначеним розкладом, що робить її ефективною для дієвого моніторингу стану доступності — наприклад, перевірок на кшталт «чи працює HTTPS», «чи відповідає DNS» або «чи запущена пошта». Основною перевагою Nagios вважається його модульна архітектура та гнучке ядро, що забезпечує підтримку численних перевірок через shell-скрипти або сторонні плагіни, завдяки чому зростає гнучкість системи для специфічних потреб, пов'язаних із критичними службами [29].

Разом із тим, у сучасних умовах Nagios має істотні обмеження, які знижують його доцільність для повноцінного моніторингу навантаження або системних метрик у реальному часі. Система не передбачає вбудованих засобів графічної візуалізації метрик, що унеможливлює інтерактивний аналіз без використання додаткових інструментів, таких як PNP4Nagios або інші надбудови. Також у базовій конфігурації Nagios не зберігає часових рядів метрик, не має засобів внутрішньої історії змін стану навантаження й обмежений лише наявністю або відсутністю певних сервісів, що звужує сферу його застосування порівняно з новітніми інструментами моніторингу.

Якщо провести порівняння з Netdata, можна побачити, що підходи до моніторингу суттєво різняться: Nagios націлений на подієву модель з чітким фокусом на доступність служб, тоді як Netdata працює в режимі безперервного збору даних, орієнтованого на моніторинг навантаження й продуктивності системи в реальному часі. У контексті цільового застосування Nagios доцільніше використовувати як інструмент для оповіщення про «живі» чи «мертві» сервіси, забезпечення базового контролю доступності інфраструктури або інтеграції як складова частина більших систем типу Grafana з окремим збором метрик, тоді як

Netdata забезпечує повноцінну інтерактивну візуалізацію процесів у межах одного рішення без сторонніх надбудов, дозволяючи виявляти аномалії та піки навантаження майже миттєво. У результаті, Nagios залишається актуальним у строго визначених сценаріях використання, тоді як для моніторингу динамічних змін у комп'ютерних мережах підприємства, особливо віддалених чи гетерогенних, перевага цілком очевидно залишається за Netdata [29].

Prometheus – гнучкий, але складніший у налаштуванні. Не можна залишити поза увагою платформу Prometheus, яка є однією з найперспективніших і найсучасніших систем збору метрик, що активно використовується в DevOps-середовищах і зазвичай інтегрується з Grafana для візуалізації. Вона базується на моделі pull, коли дані збираються за розкладом безпосередньо з експортера, а не отримуються шляхом надсилання з агентів, завдяки чому система залишається гнучкою та масштабованою. Гнучкість Prometheus проявляється насамперед у використанні мови запитів PromQL, яка дозволяє створювати складні агрегації, фільтрації, обчислення та відображення даних у потрібному вигляді, що особливо важливо в середовищах із великою кількістю мікросервісів або розподілених вузлів. До сильних сторін також належить сумісність з великою кількістю експортерів, серед яких можна виділити "node_exporter", "blackbox_exporter", "mysqld_exporter" та багато інших, а також активна підтримка з боку Kubernetes, Cloud Native спільноти та широке поширення документації [29].

Водночас Prometheus не має вбудованого інтерфейсу для візуалізації графіків, що робить його залежним від сторонніх рішень на кшталт Grafana, і хоча така комбінація забезпечує виняткові можливості кастомізації, вона потребує більш складного налаштування, що може стати перешкодою на старті, особливо для новачків. Додатковою складністю є необхідність правильної конфігурації системи для захищених середовищ, оскільки модель pull може мати труднощі при роботі через фаєрволи, VPN або ізольовані сегменти мережі, де простіше працювати з push-механізмами. Крім цього, розгортання Prometheus у складних середовищах із великою кількістю компонентів вимагає значної попередньої підготовки, розуміння структури конфігураційних файлів та підтримки окремих

компонентів моніторингу, включаючи експортери, сервіси автообнаруження, додаткові бази для довготривалого зберігання метрик і так далі [29].

У порівнянні з Netdata, Prometheus суттєво виграє в гнучкості завдяки потужності мови PromQL, однак поступається в простоті першого розгортання та зручності для початківців. Якщо Netdata дозволяє практично одразу після запуску бачити інтерактивні графіки та метрики системи в реальному часі без потреби у складних налаштуваннях, то Prometheus вимагає щонайменше встановлення та конфігурації Grafana, а також написання дашбордів і запитів вручну. У плані ролі в інфраструктурі Prometheus зазвичай виступає частиною більших екосистем, де окремо реалізовано збір метрик, їхнє зберігання, візуалізацію й оповіщення, тоді як Netdata є самодостатнім агентом, який одночасно збирає, аналізує й відображає інформацію з мінімальними вимогами до оточення [29].

Підсумовуючи, Prometheus чудово підходить для моніторингу великих кластерів, хмарних або контейнеризованих середовищ, де необхідна масштабованість, точність і розширені можливості аналітики, тоді як у випадку простіших інфраструктур або локальних серверів, а також для користувачів без глибокої підготовки в DevOps, доцільніше починати з Netdata як більш інтуїтивного рішення з усім необхідним «з коробки», що й було обрано у поточному проєкті.

Netdata — найкраще для швидкого старту і візуального моніторингу. Netdata позиціонується як універсальне та інтуїтивно зрозуміле рішення для швидкого старту моніторингу, засноване на принципі «встановив і забув», що дозволяє розгорнути базовий набір засобів спостереження за станом системи без потреби в складних конфігураціях або зовнішніх інтеграціях. Завдяки автоматичному виявленню сервісів і метрик, воно забезпечує глибоке охоплення параметрів навіть без ручного втручання. Наявність інтерактивних вбудованих графіків робить можливим миттєве візуальне представлення зібраної інформації без необхідності встановлення таких рішень як Grafana, а низьке споживання ресурсів дозволяє використовувати платформу навіть на слабких або старих машинах, не погіршуючи продуктивність основної системи. Підтримка потокової передачі даних,

можливість інтеграції з хмарними сервісами та реалізація сповіщень доповнюють функціональність рішення, роблячи його зручним інструментом для моніторингу стану домашніх або офісних серверів [29].

Особливо ефективним Netdata є в сценаріях, де немає потреби в довготривалому зберіганні історії або глибокій аналітиці, наприклад під час тестування навантаження на нові сервери, віртуальні машини або під час розробки внутрішніх сервісів. Серед обмежень системи варто зазначити зберігання метрик переважно в оперативній пам’яті, що не дозволяє повертатися до даних за тривалий період, а також обмежену можливість формування складних аналітичних запитів, характерних для таких систем як Prometheus, де реалізовано власну мову запитів та підтримку агрегованих обчислень [29].

Порівнюючи Netdata з іншими популярними системами, такими як Zabbix, Nagios або Prometheus, простежується чітке позиціонування кожного рішення відповідно до сценаріїв використання. Наприклад, Zabbix орієнтований передусім на корпоративне середовище з великою кількістю вузлів і складною топологією, забезпечує високу масштабованість, підтримку шаблонів, агентів і проксі, має власну базу даних для зберігання метрик і гнучкі можливості для подієвого моніторингу. Nagios зосереджується переважно на контролі доступності сервісів, меншою мірою на збиранні метрик, потребує додаткових плагінів для реалізації графіків і має обмежену масштабованість. Prometheus, у свою чергу, призначений для DevOps-інфраструктури, хмарних середовищ і кластеризованих систем, дозволяє будувати складні запити на основі pull-моделі, однак потребує зовнішніх рішень для візуалізації і має складніший процес налаштування, що робить його менш зручним для початкових впроваджень (див. табл. 3.1) [29].

Таблиця 3.1 – Порівняння систем моніторингу

Критерій	**Netdata**	**Zabbix**	**Nagios**	**Prometheus**
Простота встановлення	**Дуже проста**	Складна	Помірна	Потребує налаштувань
Візуалізація з коробки	**Так, інтерактивні графіки**	Так, графіки та дашборди	Ні (потрібні плагіни)	Ні (потрібна Grafana)
Зберігання історії	**Обмежене, в RAM**	У базі даних	Немає	У time-series базі
Глибина моніторингу	**Висока (автоматичне виявлення)**	Дуже висока (агенти, шаблони)	Середня (сервіси)	Висока (через експортери)
Масштабованість	**Через stream або Netdata Cloud**	Висока (проксі, шаблони)	Обмежена	Висока (clustering, federation)
Подієвий моніторинг	**Обмежено**	Так	Так	Ні
Складність конфігурації	**Мінімальна**	Висока	Середня	Середня – висока
Підтримка DevOps / Cloud Native	**Обмежена**	Часткова	Ні	Так (Kubernetes, CI/CD)
Ідеальне застосування	**Локальні сервери, швидка діагностика**	Великі підприємства, дата-центри	Контроль доступності сервісів	Хмарні середовища, мікросервіси

На цьому тлі Netdata демонструє найкращий баланс між швидкістю впровадження, мінімальною конфігурацією та візуальною наочністю, що робить його особливо зручним для реалізації прототипів, моніторингу окремих вузлів або

внутрішніх проєктів, де важливо швидко отримати доступ до стану системи без значних інженерних витрат. Його простота розгортання, що зводиться до запуску скрипта без ручного налаштування, дозволяє практично миттєво розпочати спостереження, а вбудовані інструменти забезпечують повноцінне уявлення про навантаження, стан мережі, процесів і апаратного забезпечення (див. рис. 3.4-3.5).

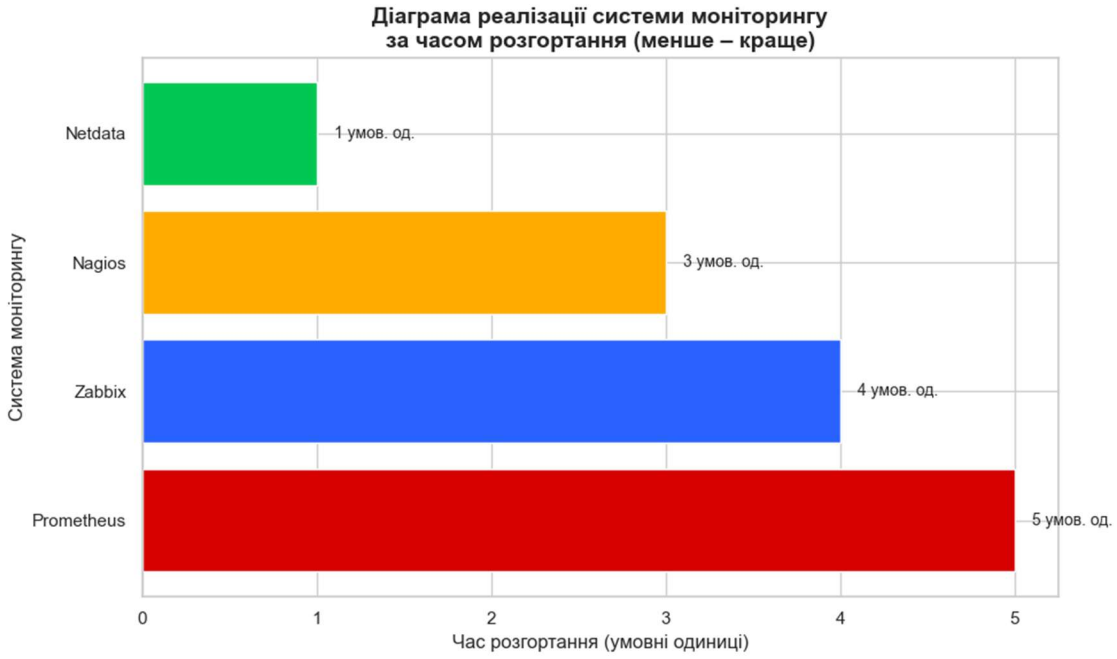


Рисунок 3.4 – Порівняння часу розгортання систем моніторингу

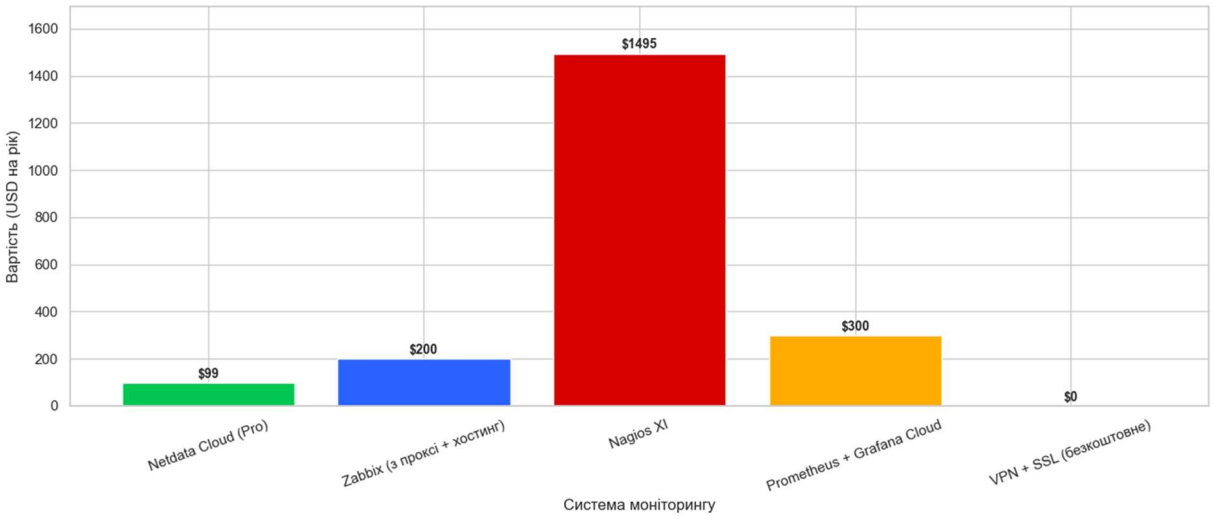


Рисунок 3.5 – Вартість реалізації віддаленого моніторингу

У сукупності ці якості роблять Netdata оптимальним вибором у ситуаціях, де час, простота і візуальна доступність переважають над необхідністю у складних аналітичних звітах чи глибокому історичному аналізі.

3.3 Можливості розширення у майбутньому

Побудова системи моніторингу на базі Netdata відкриває широкі можливості для подальшого розвитку, модернізації та масштабування, навіть якщо початкова реалізація виконана у найпростішому вигляді як локальне розгортання на одному сервері без використання контейнерів. Таке базове розгортання, хоча й демонструє обмежений набір функцій, у разі встановлення на повноцінному сервері, а не у віртуальній машині, вже здатне адаптуватися до складніших сценаріїв. Саме тому подальші кроки мають бути спрямовані на розширення функціональності, інфраструктурну адаптацію під більш вимогливе корпоративне середовище та інтеграцію з іншими системами, такими як Home Assistant, Node-RED або OpenHAB, а також на розгортання в мережах із багатьма вузлами та подальше впровадження елементів аналітики й прогнозування [30].

Контейнеризація (Docker) для кращої мобільності. У цьому контексті контейнеризація є одним із основних напрямків розширення, що дозволяє не лише пришвидшити розгортання сервісу, але й забезпечити більшу мобільність та гнучкість в експлуатації. Netdata може бути запущено як контейнерний сервіс через Docker, що забезпечує просте та уніфіковане розгортання незалежно від середовища — достатньо сумісного хоста, щоб миттєво розгорнути повноцінну систему моніторингу. Крім того, завдяки ізоляції середовища, яку забезпечує контейнер, виключається ймовірність конфліктів між різними службами на одному сервері, що є особливо критичним у випадках обмежених ресурсів або розміщення на спільних хостах. Такий підхід дозволяє також автоматизувати оновлення — використовуючи інструменти на кшталт "docker-compose", можливо регулярно

отримувати останні версії сервісу без ручного втручання, що відповідає сучасним підходам безперервної інтеграції й доставки (див. рис. 3.6) [30].

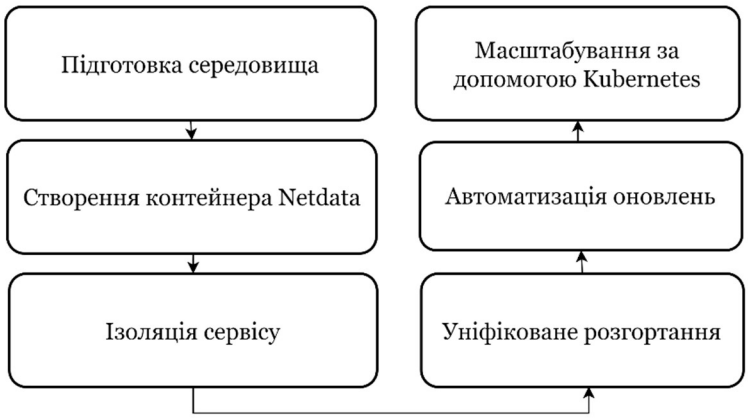


Рисунок 3.6 – Контейнеризація системи моніторингу

Завдяки контейнеризації відкривається перспектива масштабування, зокрема шляхом використання оркестраторів на кшталт Kubernetes. Такий підхід дозволяє автоматично масштабувати інстанси агента Netdata в міру збільшення кількості вузлів у системі або змін у навантаженні. Ця гнучкість особливо важлива для DevOps-команд та динамічних проєктів, де зміни відбуваються постійно, а від моніторингової системи очікується адаптивність та надійність. Завдяки всім цим властивостям контейнеризований Netdata стає не лише інструментом для швидкого старту, а й платформою, готовою до професійної експлуатації на більш складному рівні, з можливістю безболісного переходу до масштабних розгортань без зміни базової логіки роботи [30].

Інтеграція з іншими сервісами – Node-RED, Home Assistant. Завдяки відкритій архітектурі рішення, яке було обрано як основу системи моніторингу, тобто Netdata, з’являється можливість легкої інтеграції з будь-якими зовнішніми системами автоматизації, візуалізації та обробки подій, що дозволяє перетворити базовий моніторинговий інструмент у складнішу адаптивну екосистему. Відкритий доступ через IP-інтерфейс, підтримка REST API, Webhooks та стрімінгових метрик у реальному часі створює середовище, у якому Netdata не є ізольованим рішенням, а навпаки, може виступати центральним джерелом телеметрії для інших платформ.

Приклади таких інтеграцій охоплюють Node-RED та Home Assistant, які можуть використовувати дані від Netdata для реалізації сценаріїв реагування, контролю навантаження, автоматизації або побудови складніших візуальних панелей (див. рис. 3.7).



Рисунок 3.7 – Приклад інтеграції Netdata з Home Assistant
(візуальні панелі – дашборди)

У межах такого підходу, наприклад, можна реалізувати сценарій, за якого при зростанні навантаження на процесор або зменшенні доступної оперативної пам’яті Node-RED може автоматично згенерувати повідомлення у Telegram або Slack, забезпечуючи оперативне інформування відповідальних осіб. Подібний механізм у середовищі Home Assistant також може використовуватися для домашніх мереж або невеликих офісів, де необхідний централізований контроль за серверним обладнанням або мережею в цілому, з можливістю налаштування подій і сценаріїв з урахуванням часу доби, стану інших пристроїв або даних з інших сенсорів. Завдяки широким можливостям REST API, метрики можуть бути вибірково отримані в реальному часі, що дозволяє створювати власні віджети, ботів,

автономні панелі управління або розширення для наявних платформ, коли функціональності базового інтерфейсу виявляється недостатньо.

Інтеграція в цьому контексті дає змогу не просто реагувати на певні технічні події, а формувати логіку дій на їх основі з прив’язкою до контексту середовища, тим самим перетворюючи систему моніторингу на універсальний інструмент, здатний бути основою для платформ обробки подій, адаптованих до специфіки організації або індивідуальних потреб. У довготривалій перспективі така зв’язка дозволяє відійти від статичного аналізу телеметрії і перейти до побудови адаптивних систем підтримки рішень з повною автоматизацією реакцій, масштабованістю та гнучкістю в налаштуванні, що може стати основою для складніших систем класу IoT або індустріального моніторингу.

Розгортання в корпоративному середовищі з багатьма вузлами. Одним із основних напрямів подальшого розвитку будь-якої системи моніторингу є масштабування до рівня корпоративної мережі, де з’являється потреба не лише у розширенні кількості вузлів, а й у централізації контролю, організації доступу, забезпеченні безперервності збору метрик і підтримці ролей. У контексті Netdata такий перехід від найпростішої локальної інсталяції до розгалуженої структури з десятками або сотнями вузлів можливий завдяки використанню агентно-централізованої архітектури, де кожен сервер або пристрій працює як окремий агент, який передає дані на головний вузол або хмарне сховище через Netdata Cloud, що дозволяє централізовано переглядати, аналізувати та взаємодіяти з усіма точками моніторингу в реальному часі (див. рис. 3.8).

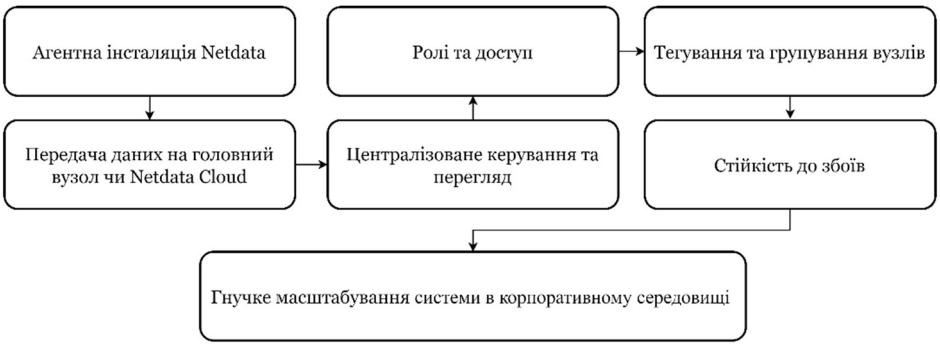


Рисунок 3.8 – Масштабування системи в корпоративному середовищі

У такій структурі забезпечується можливість масштабування системи без втрати точності, затримки або конфліктів між вузлами, оскільки всі агенти працюють автономно, а передача даних відбувається з використанням оптимізованого стрімінгу. Додатково до цього, при розгортанні у корпоративному середовищі постає питання керування доступом, оскільки в організаціях з багатьма працівниками важливо мати контроль за тим, хто може переглядати або змінювати параметри моніторингу. Вбудована система керування ролями у Netdata Cloud дозволяє визначати права окремих користувачів або цілих груп з можливістю обмеження доступу до певних вузлів, що критично у багаторівневій організаційній структурі.

Іншим важливим аспектом стає впорядкування великої кількості вузлів, яке досягається за допомогою тегування та групування, що дає змогу класифікувати агентів за ролями, типами інфраструктур (бази даних, веб-сервери, файлові сервери, маршрутизатори тощо) і спростити навігацію, аналіз та візуалізацію даних у межах дашборду. Такий підхід дозволяє не лише полегшити керування, але й впровадити сценарії розмежування зон відповідальності між різними підрозділами організації, коли кожна команда отримує доступ до власної підсистеми або вузлів.

Масштабування також не обмежується лише розширенням кількості агентів, адже у міру збільшення мережі та складності середовища постає потреба у побудові стійкої до збоїв архітектури з можливістю реплікації, резервного копіювання конфігурацій та автоматизованого деплою, що відкриває перспективу поступового переходу до повноцінної корпоративної системи моніторингу з розподіленою логікою, централізованим управлінням і адаптивністю до динамічних змін інфраструктури. У результаті навіть просте рішення, розгорнуте на одному вузлі, може еволюціонувати до гнучкої багатовузлової системи з підвищеною керованістю, прозорістю та ефективністю, орієнтованої на потреби сучасного корпоративного середовища.

Розширення функцій аналітики та прогнозування. З часом зростає потреба не лише у відображенні поточних показників роботи системи, а й у здатності передбачати її майбутні стани, виявляти аномалії до того, як вони призведуть до

збоїв, та реалізовувати автоматизований аналіз накопичених даних, що виводить систему моніторингу за межі суто реактивної моделі. У такому випадку актуальним стає розширення функцій аналітики та прогнозування, реалізоване через експорт метрик до зовнішніх сховищ, які підтримують довготривале зберігання, що дозволяє накопичувати історичні дані та опрацьовувати їх за допомогою сторонніх інструментів для статистичного аналізу та побудови моделей. Така можливість відкриває шлях до інтеграції алгоритмів машинного навчання, які, працюючи з великими обсягами телеметричних даних, здатні визначати приховані закономірності, адаптивно оцінювати стан інфраструктури та формувати прогнози майбутнього використання ресурсів, що особливо корисно для стратегічного планування масштабування, оновлення або оптимізації навантажень (див. рис. 3.9).

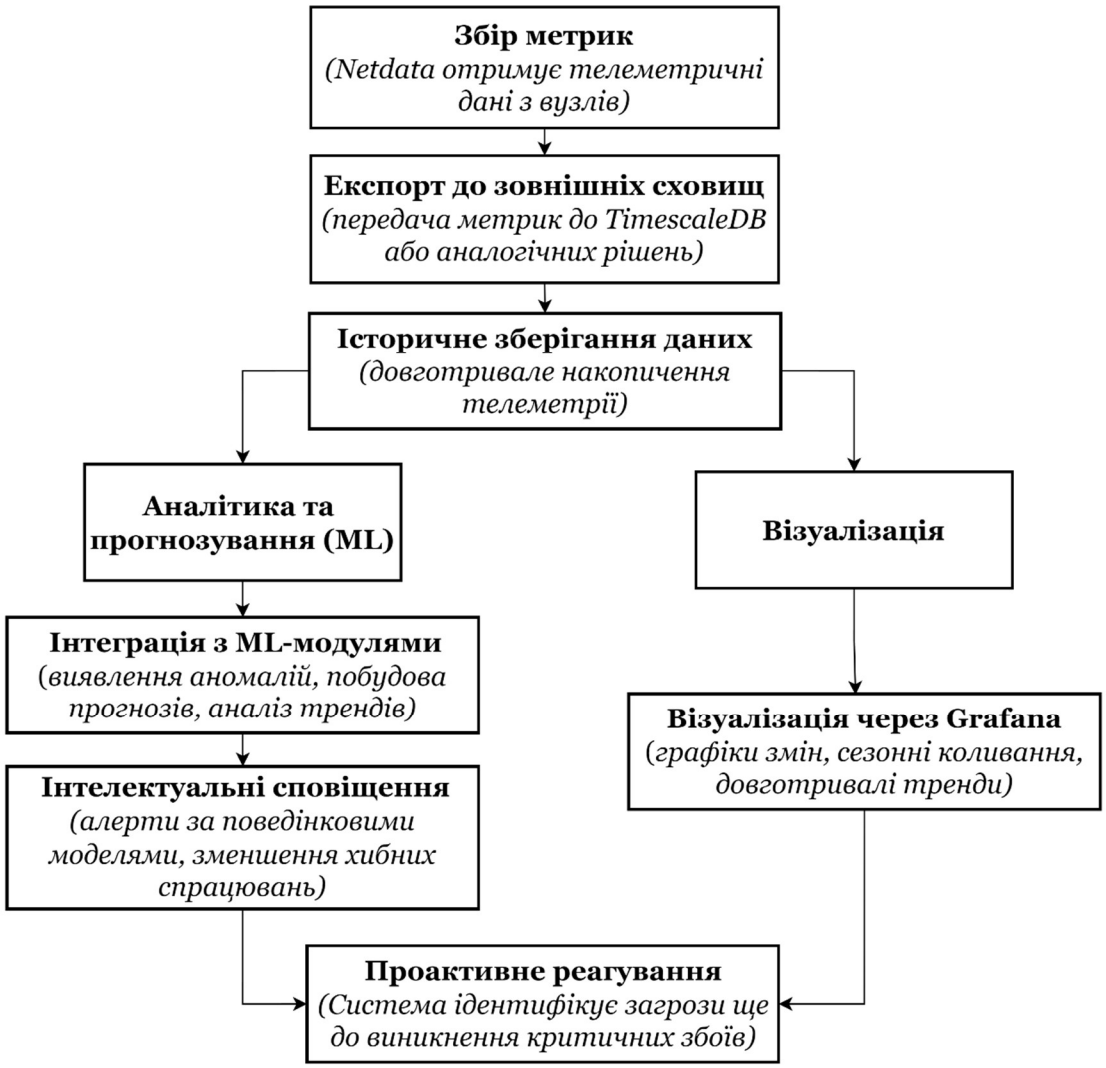


Рисунок 3.9 – Схема вдосконаленої аналітики із застосуванням ML-модуля

Паралельно із цим доцільно реалізовувати візуалізацію довготривалих трендів, яка можлива завдяки інтеграції з системами на кшталт Grafana та базами даних серій часових рядів, як-от TimescaleDB. Побудова графіків змін за дні, тижні чи місяці дозволяє помітити циклічність, тенденції спаду чи росту, сезонні впливи та інші закономірності, що виводить аналітичний інструментарій за межі класичного миттєвого моніторингу. Окрім цього, застосування інтелектуальних механізмів сповіщення, які ґрунтуються не на фіксованих порогових значеннях, а на динамічних межах, що формуються з урахуванням поведінки системи, дозволяє значно знизити кількість хибнопозитивних алертів, забезпечити релевантність повідомлень і підвищити ефективність реагування у разі дійсної загрози.

У цьому ключі, аналітичне розширення надає можливість побудувати не лише систему, яка відстежує події у реальному часі, а й платформу, здатну діяти на випередження, ідентифікуючи проблеми до моменту їх критичного впливу. Таке перетворення з традиційного інструменту моніторингу на проактивну систему аналізу дозволяє досягти нового рівня керованості, автоматизації та відповідності вимогам реального виробничого середовища.

Можливість розширення функціоналу на базі Netdata свідчить про її гнучкість, відкритість і масштабованість, що дає змогу пройти шлях від простого локального рішення до комплексної багатофункціональної платформи з глибокою інтеграцією, сучасними засобами безпеки та потужною системою аналітики. Такий підхід робить обране рішення релевантним не лише для навчального чи демонстраційного середовища, а й цілком придатним для використання у масштабованих мережевих структурах, корпоративних середовищах і критичних інфраструктурах, де якість моніторингу має безпосередній вплив на стабільність та надійність всієї системи.

ВИСНОВКИ

У результаті впровадження системи віддаленого моніторингу стану комп’ютерної мережі було побудовано тестове середовище на базі Ubuntu Server із встановленим програмним забезпеченням Netdata. Було визначено базову архітектуру рішення, що включає центральний вузол з веб-інтерфейсом та агенти на цільових системах. Реалізація проводилася без використання контейнеризації, що спростило налаштування й дозволило зосередитись на базовій функціональності системи.

На основі встановленого рішення здійснено налаштування збору основних метрик: навантаження на процесор, використання оперативної пам’яті, мережевої активності та стану дисків. Була протестована робота системи в реальному часі та перевірено стабільність веб-доступу через стандартний порт 19999. Після базового налаштування сформовано дашборд для візуального відображення параметрів, що дозволило отримати загальне уявлення про стан вузла й оцінити можливості масштабування.

Під час реалізації також перевірено можливості розширення функціоналу за допомогою модулів сповіщення, а також протестовано сценарій додавання другого агента. Робота підтвердила, що Netdata може виступати базовим рішенням для моніторингу невеликих або середніх мереж без потреби у складних конфігураціях, з можливістю подальшого розширення за рахунок інтеграції з зовнішніми сервісами зберігання або системами аналітики.

Окрему увагу було приділено методам підвищення ефективності впровадженої системи: розглянуто варіанти захисту трафіку за допомогою TLS/SSL, підключення централізованої бази для довгострокового зберігання метрик, а також впровадження агентно-орієнтованого підходу для спостереження за кількома вузлами одночасно, що створює підґрунтя для подальшого розвитку системи у напрямку повноцінного корпоративного рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Definition of network monitoring. Cloudflare. URL: https://www.cloudflare.com/learning/network-layer/what-is-network-monitoring (дата звернення: 25.02.2025)
2. Types of network monitoring. SolarWinds. URL: https://www.solarwinds.com/resources/it-glossary/network-monitoring (дата звернення: 28.02.2025)
3. Benefits of network monitoring. ManageEngine. URL: https://www.manageengine.com/network-monitoring/benefits.html (дата звернення: 28.02.2025)
4. Corporate use cases for network monitoring. Paessler PRTG. URL: https://www.paessler.com/it-explained/network-monitoring (дата звернення: 01.03.2025)
5. Key components of a network monitoring system. Nagios. URL: https://www.nagios.org/solutions/network-monitoring (дата звернення: 03.03.2025)
6. Centralized vs distributed monitoring. Zabbix. URL: https://www.zabbix.com/documentation/current/en/manual/concepts/distributed_monitoring (дата звернення: 05.03.2025)
7. Network monitoring protocols explained. Cisco. URL: https://www.cisco.com/c/en/us/solutions/enterprise-networks/network-management/index.html (дата звернення: 06.03.2025)
8. Security in network monitoring systems. Rapid7. URL: https://www.rapid7.com/fundamentals/network-security-monitoring (дата звернення: 08.03.2025)
9. Overview of monitoring tools. Comparitech. URL: https://www.comparitech.com/net-admin/network-monitoring-tools (дата звернення: 11.03.2025)
10. How to choose network monitoring software. TechTarget. URL: https://www.techtarget.com/searchnetworking/tip/How-to-choose-the-best-network-monitoring-tools (дата звернення: 14.03.2025)
11. Cloud vs on-premise monitoring solutions. LogicMonitor. URL: https://www.logicmonitor.com/blog/cloud-vs-on-premises-network-monitoring (дата звернення: 16.03.2025)
12. What is Netdata and how it works. Netdata Docs. URL: https://learn.netdata.cloud/docs/overview/what-is-netdata (дата звернення: 21.03.2025)
13. Key metrics to monitor in a network. Datadog. URL: https://www.datadoghq.com/blog/network-performance-monitoring-metrics (дата звернення: 23.03.2025)
14. Network services status monitoring. Icinga. URL: https://icinga.com/docs/icinga-2/latest/doc/10-network-monitoring (дата звернення: 30.03.2025)

15. Alerting and event handling. Prometheus. URL: https://prometheus.io/docs/alerting/latest/overview (дата звернення: 02.04.2025)
16. Downtime visualization and capacity planning. Grafana. URL: https://grafana.com/docs/grafana/latest/dashboards/ (дата звернення: 03.04.2025)
17. Minimum requirements for monitoring server. Zabbix Documentation. URL: https://www.zabbix.com/documentation/current/en/manual/installation/requirements (дата звернення: 08.04.2025)
18. Ubuntu Server for monitoring. Canonical. URL: https://ubuntu.com/server/docs (дата звернення: 11.04.2025)
19. Network ports for Netdata. Netdata Docs. URL: https://learn.netdata.cloud/docs/configure/nodes/ports (дата звернення: 15.04.2025)
20. Netdata installation methods. Netdata GitHub. URL: https://github.com/netdata/netdata (дата звернення: 17.04.2025)
21. Ubuntu in VirtualBox setup. VirtualBox Manual. URL: https://www.virtualbox.org/manual/ch01.html (дата звернення: 21.04.2025)
22. Installing Netdata on Ubuntu. Netdata Community Forum. URL: https://community.netdata.cloud/t/how-to-install-netdata-on-ubuntu/52 (дата звернення: 25.04.2025)
23. Accessing Netdata dashboard. Netdata Docs. URL: https://learn.netdata.cloud/docs/visualize/dashboard (дата звернення: 25.04.2025)
24. Data buffering and visualization pipeline. Netdata Internals. URL: https://learn.netdata.cloud/docs/architecture/internal-architecture (дата звернення: 25.04.2025)
25. Health and anomaly detection. Netdata Health Monitoring. URL: https://learn.netdata.cloud/docs/monitor/health (дата звернення: 28.04.2025)
26. Setting up a test monitoring environment. Linuxize. URL: https://linuxize.com/post/how-to-create-a-virtual-lab-on-ubuntu (дата звернення: 30.04.2025)
27. Remote access and security settings. Netdata Security Docs. URL: https://learn.netdata.cloud/docs/configure/nodes/security (дата звернення: 01.05.2025)
28. Long-term storage with external DB. Netdata with Prometheus/InfluxDB. URL: https://learn.netdata.cloud/docs/store/external-databases (дата звернення: 03.05.2025)
29. Comparing Netdata with other tools. ITPro Today. URL: https://www.itprotoday.com/devops/comparing-network-monitoring-tools-netdata-prometheus-nagios (дата звернення: 05.05.2025)
30. Docker vs Virtual Machines for deployment. Scale Computing. URL: https://www.scalecomputing.com/resources/whats-the-difference-between-virtualization-vs-containerization (дата звернення: 05.05.2025)
31. Кондратюк Р.Ю., Торошанко Я.І. IoT-система для віддаленого контролю корпоративної мережевої інфраструктури. «Сучасний стан та перспективи розвитку IoT» : VI Науково-техн. конф., м.Київ, 15 квітня 2025 р. (подано до публікації).
32. Кондратюк Р.Ю., Торошанко Я.І. Моделі збору та аналізу даних у системах віддаленого моніторингу мережевої інфраструктури. «Застосування

програмного забезпечення в ІКТ» : Всеукр. науково-техн. конф., м.Київ, 24 квітня 2025 р. С.575-577.

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Комп'ютерної інженерії



КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

НА ТЕМУ: «СИСТЕМА ВІДДАЛЕНОГО МОНІТОРИНГУ СТАНУ КОМП'ЮТЕРНОЇ МЕРЕЖІ ПІДПРИЄМСТВА»

Виконав студент групи КІД-42
Кондратюк Роман Юрійович

Керівник дипломної роботи:
Вєчерковська Анастасія Сергіївна,
канд. техн. наук, доц.

Київ – 2025

Мета, об’єкт, предмет дослідження

Мета роботи – спрощення розгортання та підвищення ефективності системи віддаленого моніторингу стану комп’ютерної мережі підприємства за рахунок використання відкритих інструментів, централізованої архітектури та модулів візуалізації

Об’єкт дослідження – процеси контролю та забезпечення стабільної роботи комп’ютерної мережі підприємства

Предмет дослідження – технології моніторингу, аналітики та оповіщення, що використовуються для побудови простих у впровадженні систем моніторингу на базі відкритого ПЗ

Актуальність роботи

У роботі розглянуто принципи та підходи до побудови систем моніторингу для локальних і корпоративних мереж, зокрема використання інструмента Netdata на сервері з Ubuntu Server. Підкреслено переваги прямого розгортання без Docker для швидкої демонстрації можливостей.

Реалізовано базову систему моніторингу з візуалізацією навантаження CPU, мережі, дисків тощо, а також розглянуто підключення агентів для масштабування та зовнішніх баз даних для довготривалого зберігання метрик. Запропоновано способи підвищення захищеності доступу (TLS/SSL), а також механізми покращення ефективності шляхом додавання модулів оповіщення. Результати демонструють ефективність використання Netdata у малих і середніх IT-інфраструктурах.

Класифікація та задачі систем моніторингу

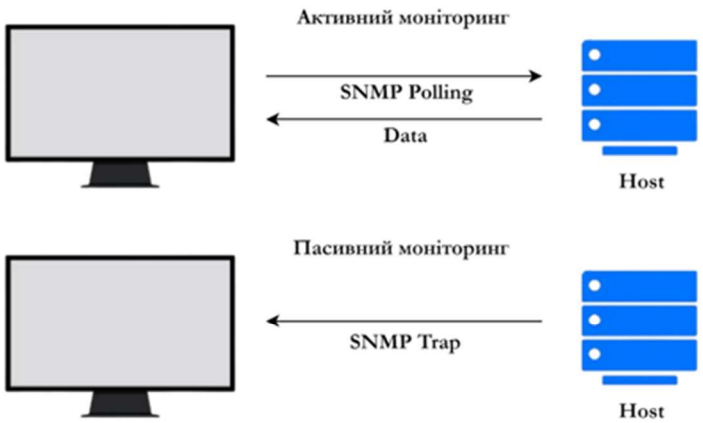


Рис. 1.1 – Класифікація типів моніторингу в комп'ютерних мережах



Рис. 1.2 – Основні задачі систем моніторингу

Архітектура системи віддаленого моніторингу

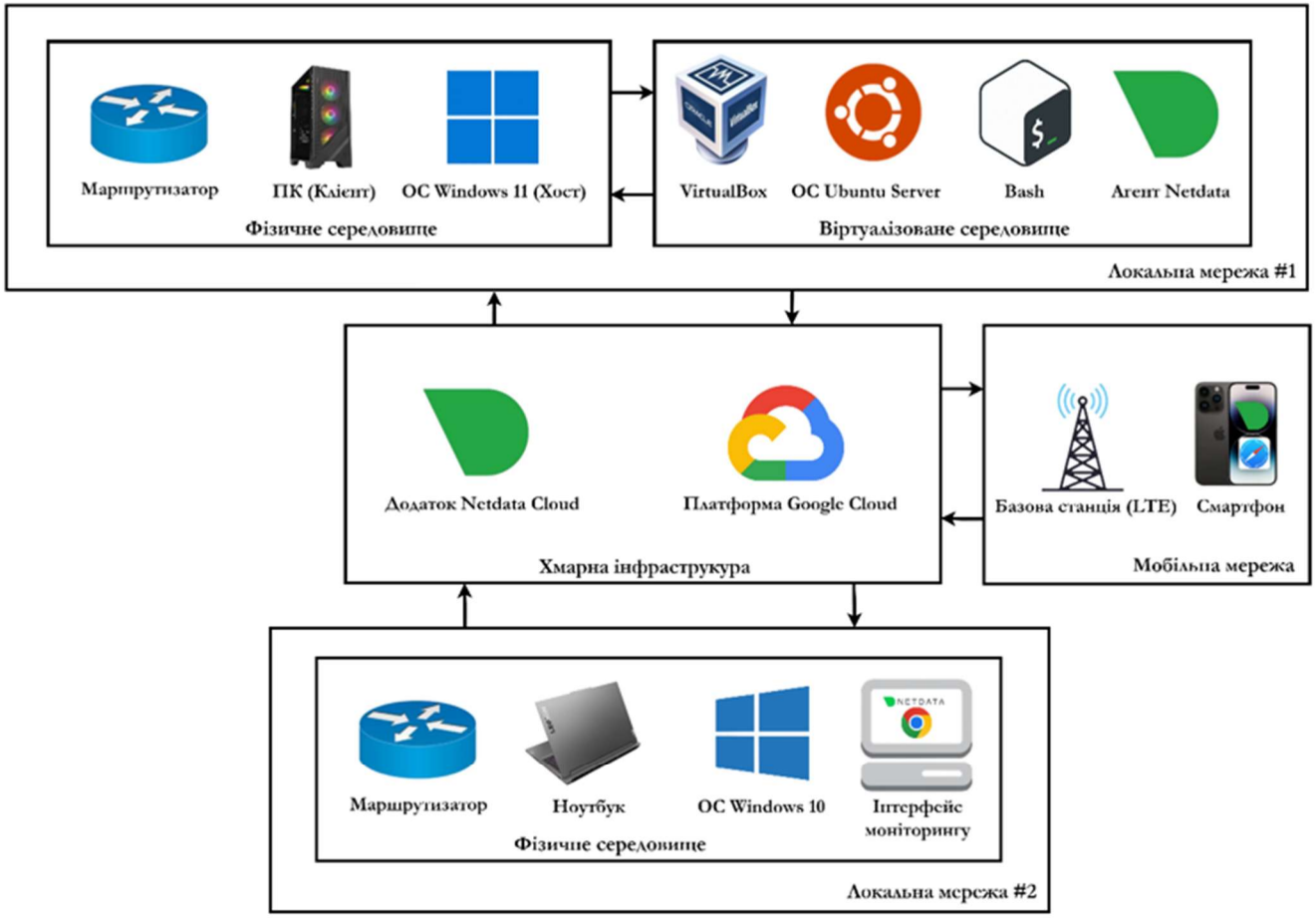


Рис. 2.1 – Архітектура системи віддаленого моніторингу комп'ютерних мереж

Практична реалізація системи моніторингу у VMware

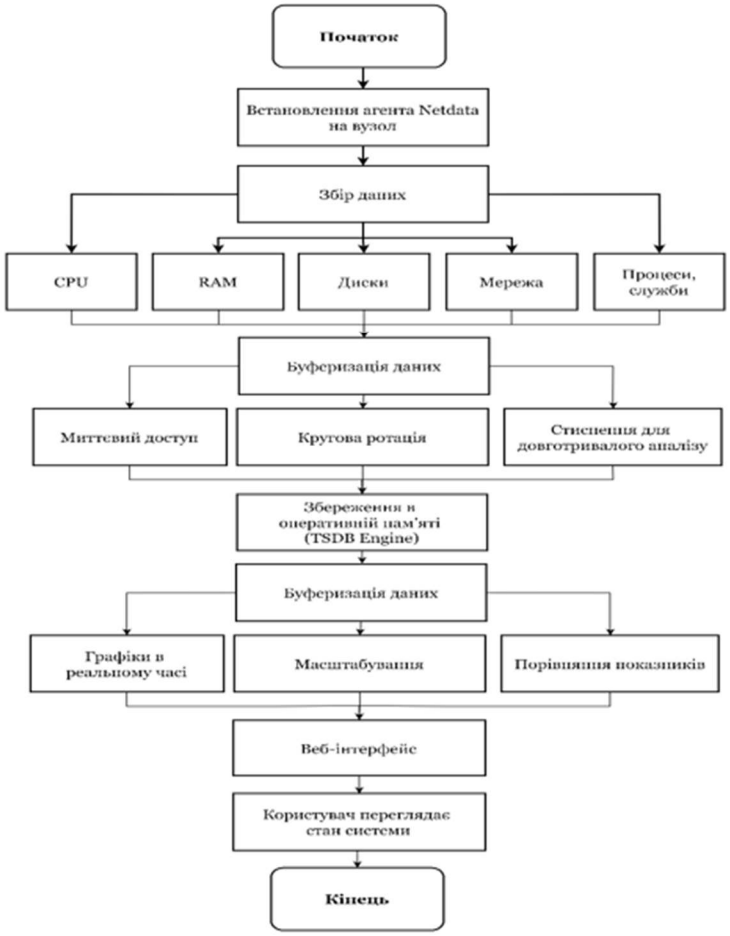


Рис. 2.10 – Загальний принцип роботи системи Netdata

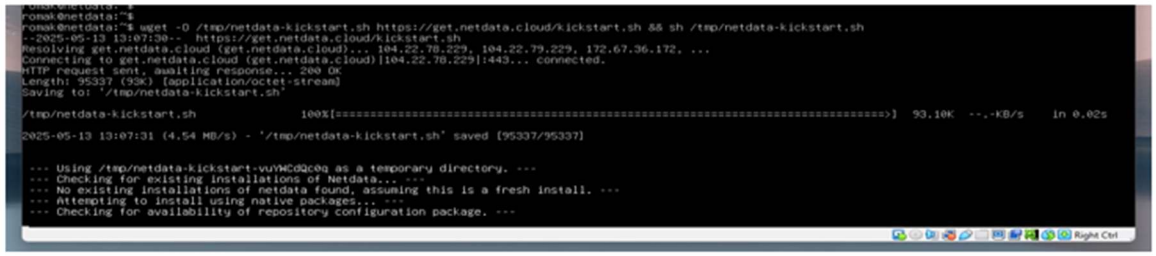


Рис. 2.6 – Інсталяція Netdata через офіційний bash-інсталятор

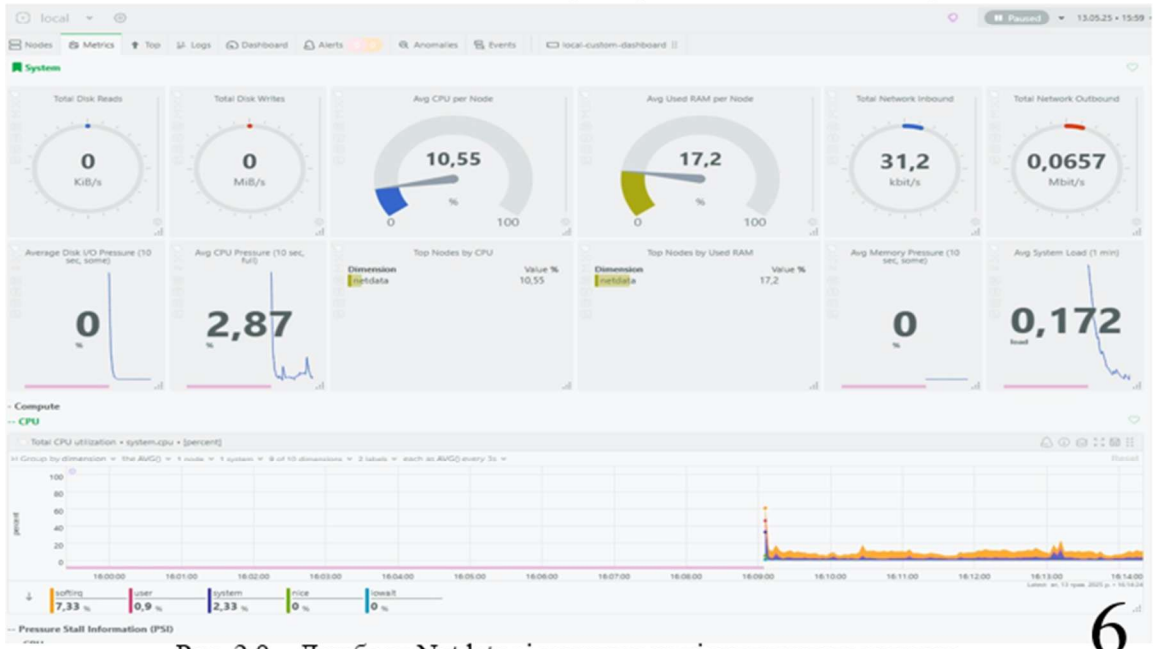


Рис. 2.9 – Дашборд Netdata зі метриками після першого запуску

Забезпечення віддаленого доступу до моніторингу

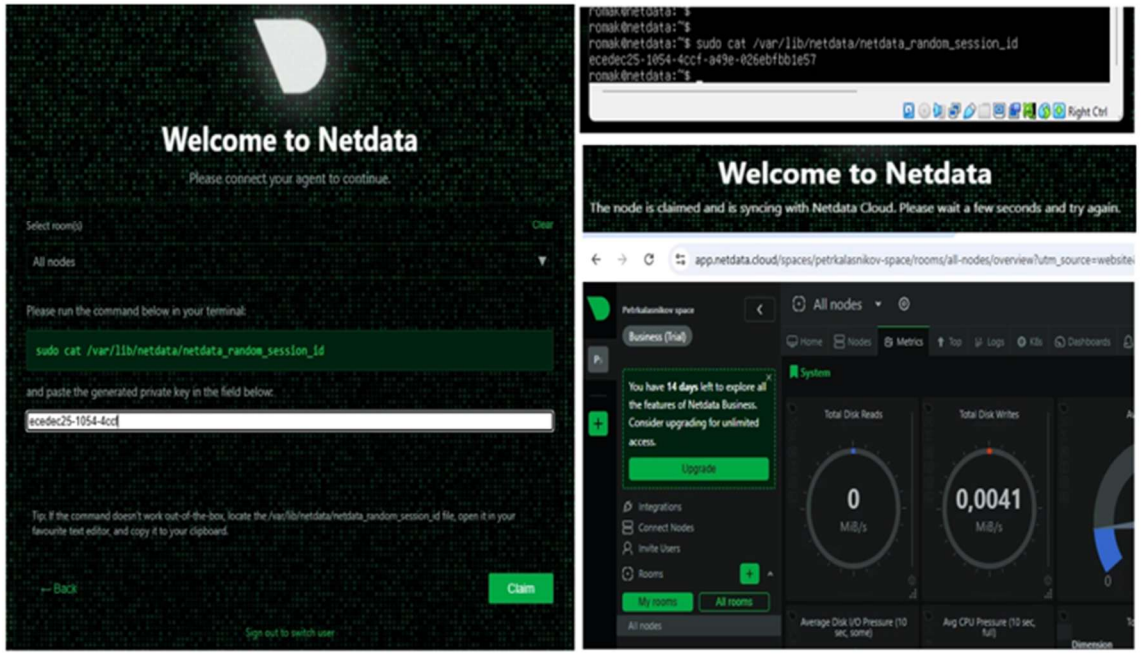


Рис. 2.14 – Приклад віддаленого доступу до моніторингу

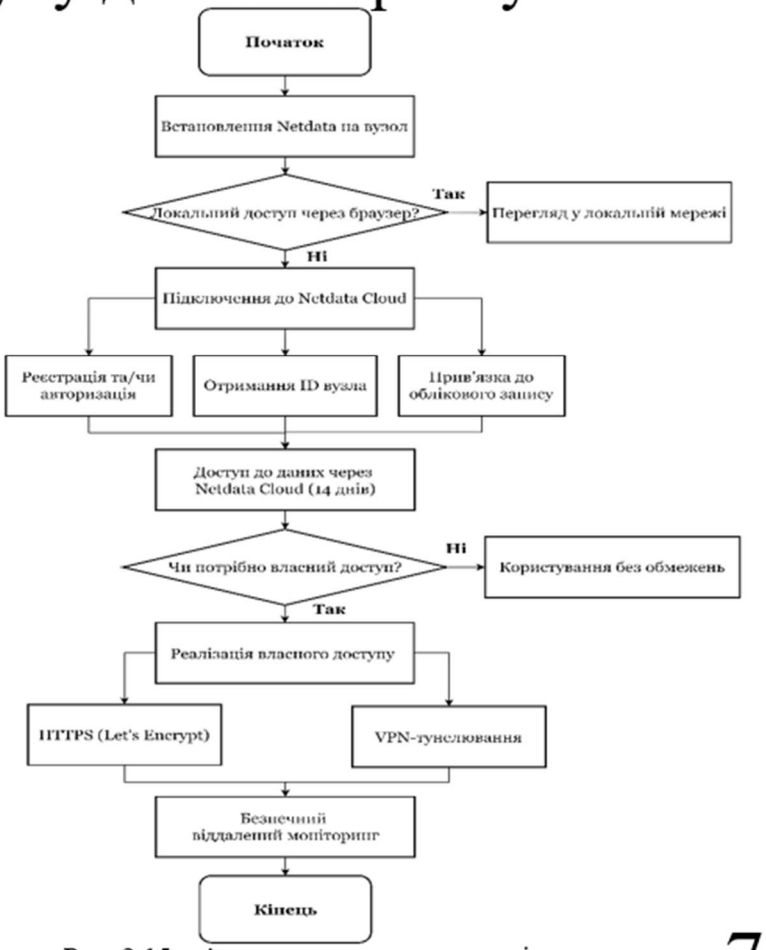


Рис. 2.15 – Алгоритм впровадження віддаленого доступу до моніторингу

Власний дашборд для перевірки ключових метрик

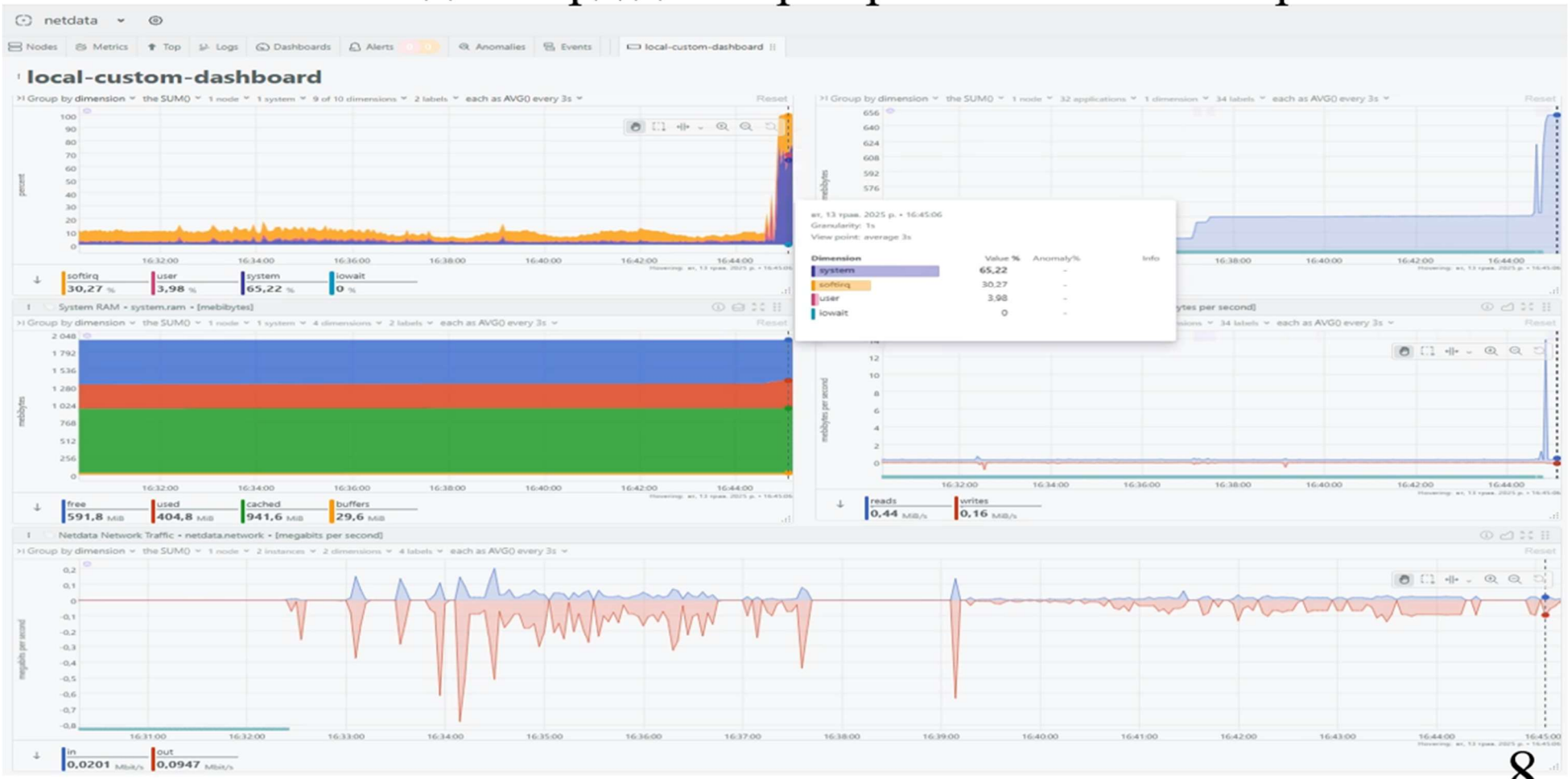


Рис. 2.16 – Знімок дашборду у період пікового навантаження

8

Перевірка системи на виявлення аномалій (відхилень)

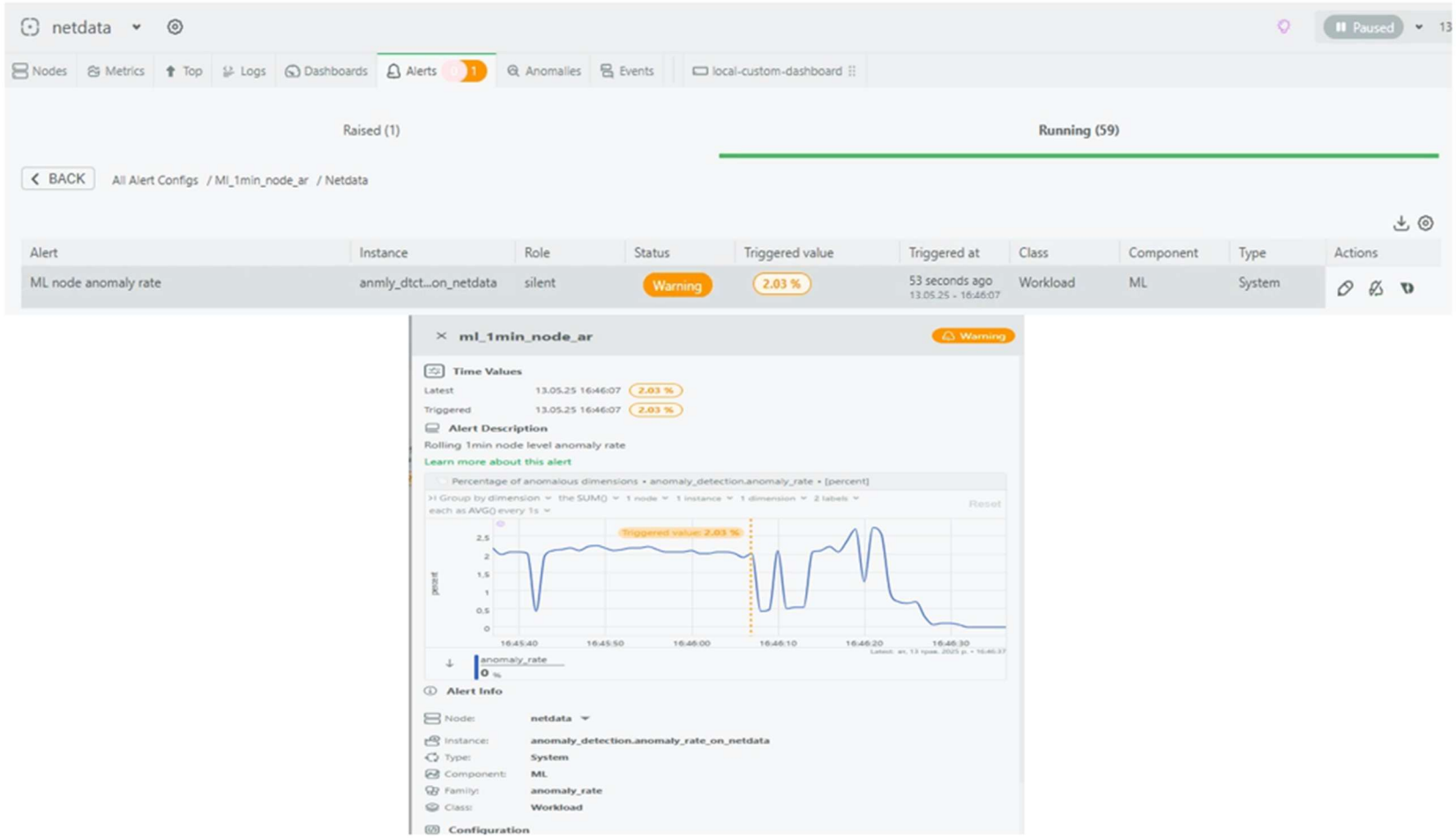


Рис. 2.19 – Аналіз активності (аномалій) сервісів у часовому розрізі

Методи масштабування та інтеграції

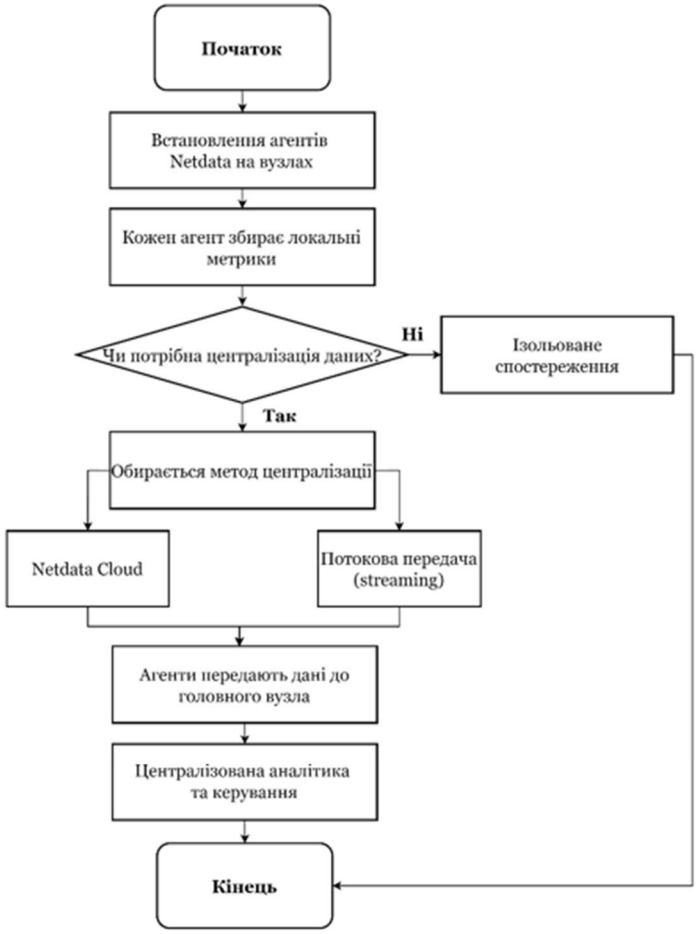


Рис. 3.2 – Метод використання централізованої архітектури з кількома агентами Netdata



Рис. 3.7 – Приклад інтеграції Netdata з Home Assistant (візуальні панелі – дашборди)

Методи масштабування та інтеграції

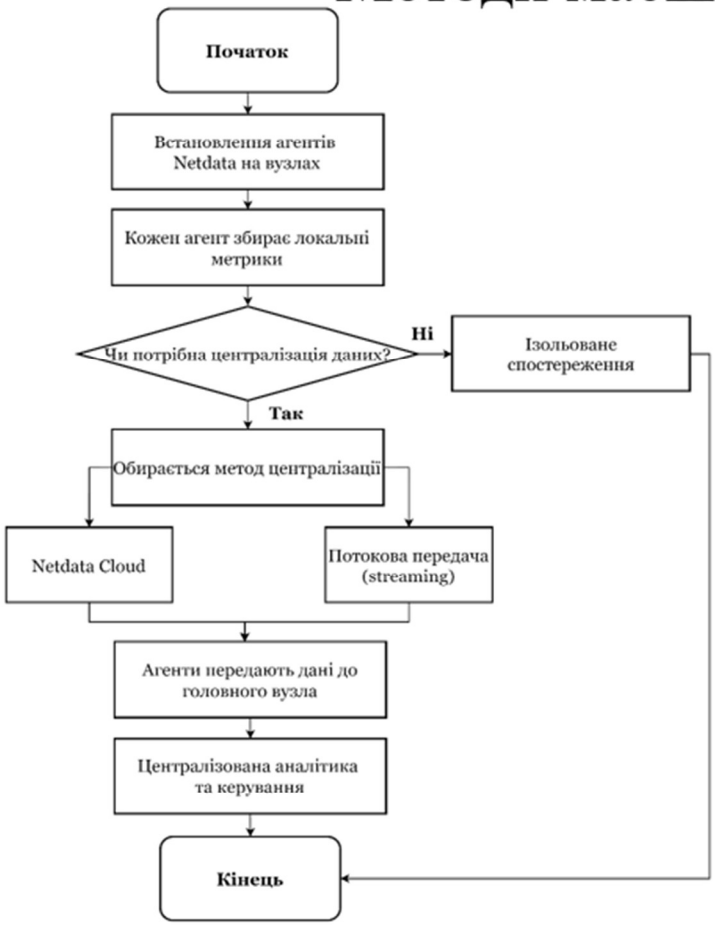


Рис. 3.1 – Метод інтеграції з базою даних для архівації метрик

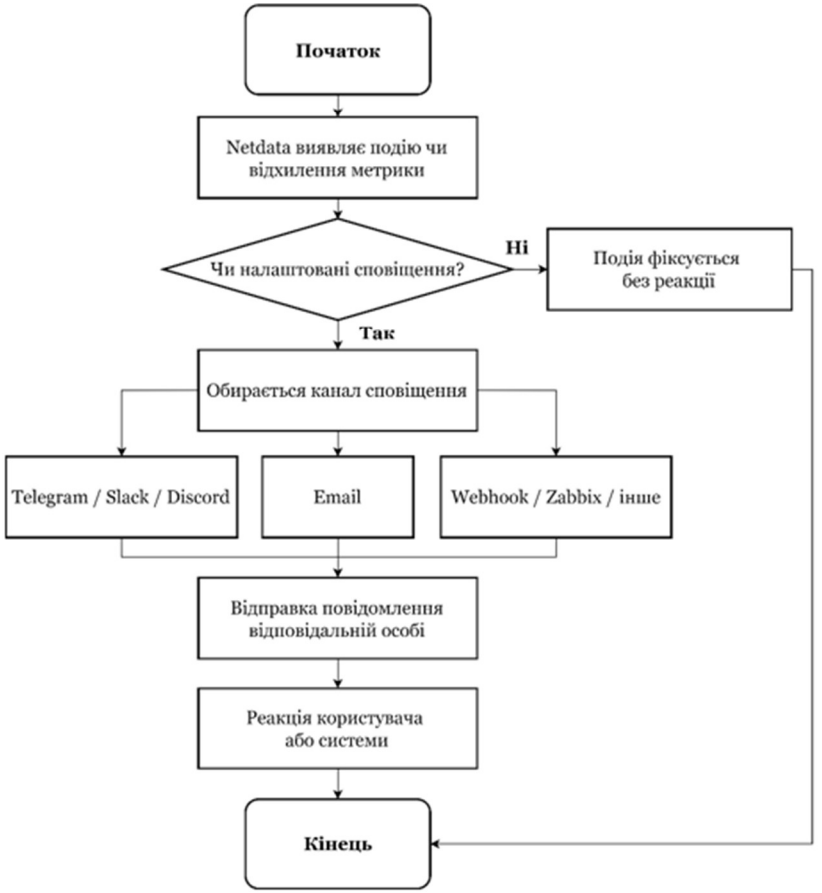


Рис. 3.3 – Метод додавання зовнішніх модулів сповіщення

Порівняльний аналіз швидкості розгортання існуючих рішень

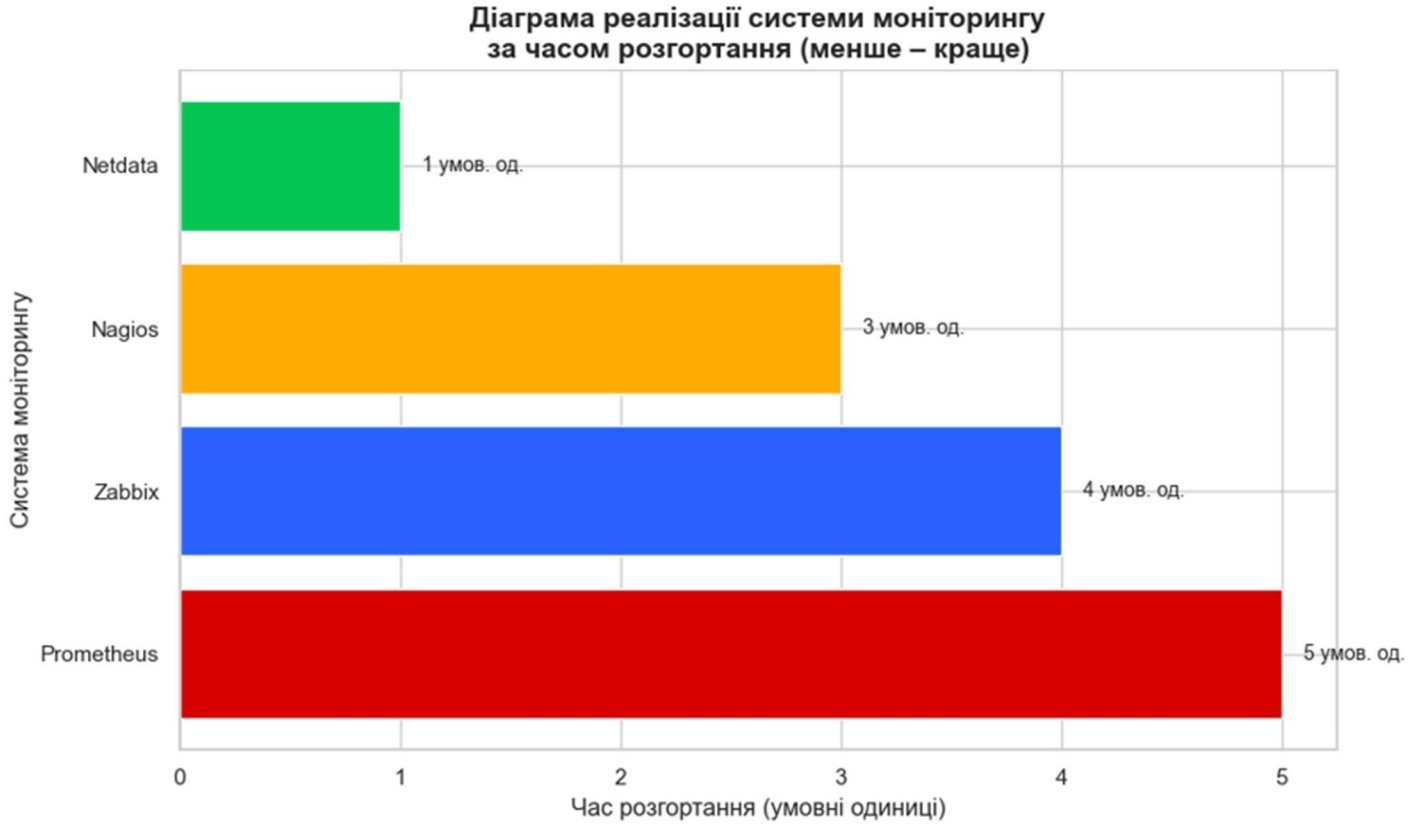


Рис. 3.4 – Порівняння часу розгортання систем моніторингу

Кошторис реалізації та безкоштовне рішення

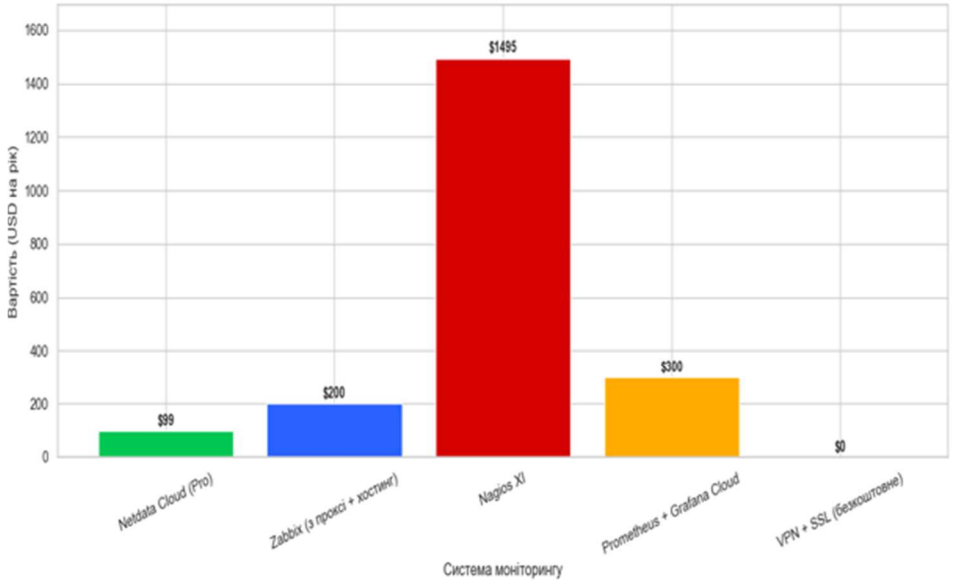


Рис. 3.5 – Вартість реалізації віддаленого моніторингу

Табл. 3.1 –Порівняння систем моніторингу

Критерій	Netdata	Zabbix	Nagios	Prometheus
Простота встановлення	**Дуже проста**	Складна	Помірна	Потребує налаштувань
Візуалізація з коробки	**Так, інтерактивні графіки**	Так, графіки та дашборди	Ні (потрібні плагіни)	Ні (потрібна Grafana)
Зберігання історії	**Обмежене, в RAM**	У базі даних	Немає	У time-series базі
Глибина моніторингу	**Висока (автоматичне виявлення)**	Дуже висока (агенти, шаблони)	Середня (сервіси)	Висока (через експортери)
Масштабованість	**Через stream або Netdata Cloud**	Висока (проксі, шаблони)	Обмежена	Висока (clustering, federation)
Подієвий моніторинг	**Обмежено**	Так	Так	Ні
Складність конфігурації	**Мінімальна**	Висока	Середня	Середня – висока
Підтримка DevOps / Cloud Native	**Обмежена**	Часткова	Ні	Так (Kubernetes, CI/CD)
Ідеальне застосування	**Локальні сервери, швидка діагностика**	Великі підприємства, дата-центри	Контроль доступності сервісів	Хмарні середовища, мікросервіси

Висновки

У межах реалізації системи віддаленого моніторингу було побудовано тестове середовище на базі Ubuntu Server із встановленим Netdata. Визначено базову архітектуру з центральним вузлом та агентами на цільових системах, здійснено збір основних метрик (ЦП, пам’ять, мережа, диски) і створено дашборд для візуалізації стану вузла в реальному часі. Роботу системи протестовано через стандартний веб-інтерфейс, що підтвердило її стабільність та функціональність.

Окрему увагу приділено розширенню можливостей: налаштування сповіщень, додавання другого агента, підвищення безпеки через TLS/SSL та розгляд централізованого зберігання метрик. Отримані результати засвідчили придатність Netdata як базового рішення для моніторингу невеликих і середніх мереж із потенціалом масштабування та інтеграції у складніші корпоративні інфраструктури.

Публікації та апробація роботи

1. Кондратюк Р.Ю., Торошанко Я.І. IoT-система для віддаленого контролю корпоративної мережевої інфраструктури. «Сучасний стан та перспективи розвитку IoT» : VI Науково-техн. конф., м.Київ, 15 квітня 2025 р. (подано до публікації).
2. Кондратюк Р.Ю., Торошанко Я.І. Моделі збору та аналізу даних у системах віддаленого моніторингу мережевої інфраструктури. «Застосування програмного забезпечення в ІКТ» : Всеукр. науково-техн. конф., м.Київ, 24 квітня 2025 р. С.575-577.