Міністерство освіти і науки України

Луцький національний технічний університет
(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій
(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки
(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

ЗАСОБИ АДМІНІСТРУВАННЯ ВІРТУАЛЬНИХ МАШИН ТА СЕРВІСІВ

VIRTUAL MACHINES AND SERVICES ADMINISTRATION TOOLS

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Лахтюк Владислав Леонідович

(підпис)

Керівник:
к.е.н., доцент
Гордєєва Дар'я Валеріївна

(підпис)

Кваліфікаційну роботу
допущено до захисту
«_____» червня 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерної інженерії та кібербезпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп'ютерна інженерія*
Освітня програма: *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ проф. Н. Чернящук
«_____« _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Лахтюку Владиславу Леонідовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Засоби адміністрування віртуальних машин та сервісів*

Керівник роботи *к.е.н., доцент Гордєєва Дар'я Валеріївна*

затверджені наказом закладу вищої освіти від «28» грудня 2022 року № 982/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 01.06.2023р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання , опубліковані зарубіжні та вітчизняні роботи в даній області*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ
Засоби адміністрування
IT інфраструктура Автоматизація задач в корпоративному середовищі
Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз проблеми з а темою роботи та постановка завдань дослідження	*Гордєєва Д.В.*		
Теоретичне дослідження та практична реалізація	*Гордєєва Д.В.*		
Практична реалізація об'єкта проектування	*Гордєєва Д.В.*		
Висновки	*Гордєєва Д.В.*		

7. Дата видачі завдання: 01.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури із досліджуваної проблеми*	До 15.11.2022 р.	
2.	*Обгрунтування теми*	До 17.12.2022 р.	
3.	*Огляд підходів до контролю стану серверів*	До 11.01.2023 р.	
4.	*Моделювання моніторингу серверів за допомогою Zabbix*	До 04.02.2023 р.	
5.	*Функціонал оповіщення адміністратора*	До 22.02.2023 р.	
6.	*Захист інформації та криптографічні протоколи*	До 28.02.2023 р.	
7.	*Оцінка продуктивності та доступності баз даних*	До 01.03.2023 р.	
8.	*Інсталяція та конфігурація веб-сервера*	До 11.03.2023 р.	
9.	*Інтеграція PHP у середовище веб-сервера Nginx*	До 16.05.2023 р.	
10.	*Організація моніторингу продуктивності*	До 20.05.2023 р.	
11.	*Інструментальна перевірка на академічний плагіат*	До 22.05.2023 р.	
12.	*Представлення кваліфікаційної роботи бакалавра до захисту*	До 01.06.2023 р.	

Здобувач вищої освіти _____ (підпис) Лахтюк В.Л. (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (підпис) Гордєєва Д.В. (прізвище, ініціали)

АНОТАЦІЯ

Лахтюк В.Л. Засоби адміністрування віртуальних машин та сервісів. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп’ютерна інженерія» спеціальності 123 Комп’ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду предметної області, тут розглядається моніторинг серверів та його важливість.

В другому розділі подані принципи та особливості роботи системи моніторингу серверів Zabbix.

Третій розділ присвячено опису реалізації моніторингу серверів за допомогою програмного продукту Zabbix.

Об'єкт – система моніторингу віртуальних серверів Zabbix.

Предмет – виступає вивчення можливостей налаштування моніторингу сервера баз даних MSSQL за допомогою встановлення з’єднання по ODBC-конектору між Zabbix-сервером та екземпляром баз даних.

Метою роботи є налаштування моніторингу серверів та баз даних MSSQL за допомогою моніторингової системи Zabbix.

Ключові слова: віртуальні машини, системи моніторингу, панель керування, Zabbix-сервер, ODBC-з’єднання.

ANNOTATION

Lakhtiuk V.L. Means of administration of virtual machines and services. Manuscript.

Bachelor's qualifying thesis of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2023.

The qualification work consists of an introduction, three sections, conclusions, and a list of used sources.

The first section is a domain overview, looking at server monitoring and its importance.

The second section presents the principles and features of the zabbix server monitoring system.

The third section is devoted to the description of the implementation of server monitoring using the zabbix software product.

The object is the monitoring system of Zabbix virtual servers.

The subject is studying the possibilities of setting up monitoring of the MSSQL database server by establishing a connection via the ODBC connector between the Zabbix server and the database instance.

The purpose of the work was to configure the monitoring of servers and MSSQL databases using the Zabbix monitoring system.

Keywords: virtual machines, monitoring systems, control panel, Zabbix server, ODBC connection.

ЗМІСТ

ВСТУП

Актуальність теми. У зв’язку з бажанням компаній або організацій розробити відмовостійкі системи, кожен компонент яких не впливатиме на інші, щоб зменшити ризик збою всієї IT-інфраструктури, розгортається багато віртуальних серверів для розгортання різного програмного забезпечення. на них.

У цьому випадку є два варіанти розвитку подій: або організація збільшує штат, який займатиметься в основному лише підтримкою існуючих систем, або впроваджує автоматизовані системи моніторингу віртуальних серверів, тим самим збільшуючи час для співробітників на розробку існуючих. і створювати нові IT-інфраструктури без збільшення чисельності робочої сили, заощаджуючи значну суму грошей, яку довелося б витратити на покриття зарплат працівників.

Потреба в децентралізації та розгалуженні систем виникла в той час, коли провідні IT-компанії почали відмовлятися від побудови інфраструктури за так званим методом «накопичення навколо ядра», коли була одна велика система, навколо якої створювалися всі інші інформаційні системи.

Тоді в разі поломки одного елемента системи відбувався збій всієї системи, і відстежити джерело аномалії було практично неможливо, т.к.

відповідний слід помилки в системі могли залишити всі її елементи, яких ця помилка торкнулася.

Цілі та завдання дослідження. Метою дослідження та завданням даної кваліфікаційної роботи було налагодження моніторингу серверів та баз даних MSSQL за допомогою системи моніторингу Zabbix.

Об'єкт дослідження. Об’єктом дослідження є система моніторингу віртуального сервера Zabbix.

Предмет дослідження. Предметом даного дослідження є вивчення можливостей налаштування моніторингу сервера бази даних MSSQL шляхом встановлення з’єднання через конектор ODBC між сервером Zabbix та екземпляром бази даних.

Методи дослідження. Для виконання поставлених завдань були задіяні такі методи:

- аналіз та вивчення різноманітних систем моніторингу серверів, їх особливостей та функцій;

- проектування системи моніторингу віртуальних машин та їх впровадження;

- проектування моніторингу служби MSSQL, встановленої на вищевказаних віртуальних машинах;

- експериментування з роботою систем моніторингу шляхом завантаження на досліджувані віртуальні сервери.

Наукова новизна. Після відносно недавнього випуску версії програмного забезпечення Zabbix 5.0 була додана можливість підключення до системи за допомогою з’єднань ODBC, що дозволило фахівцям підключатися до екземпляра бази даних MSSQL для подальшого вилучення з нього даних моніторингу.

Практичне значення отриманих результатів. В результаті наукової роботи буде представлений варіант конфігурації системи моніторингу, який зможе показувати не тільки поточний стан віртуального сервера, а й екземпляр бази даних MSSQL. Це рішення допоможе системним адміністраторам отримувати всю необхідну інформацію про поточний стан різних джерел даних з одного місця та в зручний спосіб, а також отримувати повідомлення про певні небезпечні події на серверах.

Публікації. На науково-практичній конференції на тему «Інформаційні моделі, системи та технології», яка відбулася 7-8 грудня 2022 року в ТНТУ ім. інструмент для керування та аналізу журналів «Graylog».

РОЗДІЛ 1
СИСТЕМИ ТА ПРИНЦИПИ КОНТРОЛЮ СТАНУ СЕРВЕРНОЇ ІНФРАСТРУКТУРИ

1.1 Огляд підходів до контролю стану серверів

Формалізація поняття моніторингу серверних систем є складним завданням, що зумовлено різноманіттям архітектур, типів розгортання та функціонального призначення серверів у сучасних інформаційних середовищах. Під сервером може розумітися як окремий фізичний обчислювальний вузол, так і логічно ізольований віртуальний екземпляр, розміщений у межах спільного апаратного ресурсу центрів обробки даних. При цьому одна й та сама апаратно-програмна платформа може одночасно обслуговувати значну кількість незалежних користувацьких середовищ, кожне з яких експлуатує власні серверні сервіси.

Функціональне призначення серверів охоплює широкий спектр завдань, зокрема забезпечення роботи вебзастосунків, обробку електронної кореспонденції, керування мережевими периферійними пристроями, підтримку мікросервісних архітектур, а також зберігання й обробку даних у системах керування базами даних. Наведені приклади не вичерпують повного переліку серверних ролей, що використовуються при розробці та експлуатації сучасних програмних рішень.

Моніторинг серверів у цьому контексті доцільно розглядати як безперервний процес збору, аналізу та інтерпретації інформації про стан і поведінку серверних ресурсів незалежно від їх фізичної або віртуальної природи. У процесі експлуатації такі системи здатні одночасно обробляти значні обсяги запитів на приймання та передачу даних, а відмова або недоступність окремого елемента інфраструктури може призвести до порушення функціонування великої кількості взаємопов’язаних сервісів.

Наслідки збоїв у роботі серверних систем часто мають не лише фінансовий характер, але й можуть спричиняти критичні соціальні та технічні ризики, включаючи загрозу безпеці майна, здоров'ю або життю людей. У зв'язку з цим ключовим завданням управління IT-інфраструктурою є забезпечення високого рівня надійності, доступності та відмовостійкості програмних і апаратних компонентів.

Реалізація ефективного контролю за станом різнорідних серверів потребує застосування комплексних технологічних рішень. Практика показує, що універсальні готові засоби моніторингу не завжди здатні забезпечити повну прозорість процесів, які відбуваються на фізичних або віртуальних вузлах. У результаті фахівці змушені або адаптувати існуючі інструменти під специфічні вимоги, що може ускладнювати їх подальшу експлуатацію, або впроваджувати сукупність окремих моніторингових сервісів, функціонування яких потребує постійного контролю та врахування притаманних їм обмежень.

1.2 Актуальні виклики адміністрування серверної інфраструктури

Адміністрування серверів розглядається як сукупність організаційних і технічних заходів, спрямованих на підтримання стабільного функціонування серверних ресурсів упродовж усього життєвого циклу їх експлуатації. Ключовими цілями цього процесу є досягнення високого рівня надійності, продуктивності, доступності та коректності виконання серверних операцій.

Діяльність з управління серверною інфраструктурою має регулярний характер і орієнтована насамперед на забезпечення безперервності сервісів, від якої безпосередньо залежить якість взаємодії кінцевих користувачів із програмними системами. Масштаб і складність таких завдань істотно варіюються залежно від типу організації, архітектури інфраструктури, а також кількості та призначення серверів, що перебувають в експлуатації.

У більшості практичних сценаріїв управління віртуальними машинами охоплює постійний контроль їхнього стану, розгортання та конфігурування

нових елементів інфраструктури, своєчасне оновлення програмних і апаратних компонентів, локалізацію збоїв, а також реалізацію превентивних заходів для мінімізації ймовірності відмов. Окрему увагу приділяють аналізу навантажень, які можуть різко зростати у визначені періоди, зокрема під час формування фінансової або звітної документації, що створює пікове навантаження на інформаційні сервіси.

Ефективне управління передбачає здатність прогнозувати такі коливання та своєчасно коригувати обчислювальні ресурси для запобігання дефіциту продуктивності. Водночас надмірне резервування потужностей у періоди низької активності є економічно недоцільним і може спричиняти зростання витрат, що негативно впливає на задоволеність замовників. Прийняття обґрунтованих рішень у цьому випадку базується на аналізі історичних даних, отриманих у ході попередньої експлуатації систем.

Додаткові складнощі управління виникають під час роботи з віртуалізованими середовищами, де відсутній безпосередній фізичний доступ до апаратних компонентів, які часто розміщені у віддалених центрах обробки даних. У протилежному випадку, за наявності локального доступу до серверного обладнання, з’являються інші виклики, пов’язані з обслуговуванням апаратних ресурсів. Незалежно від способу розгортання, ефективне управління вимагає комплексного контролю програмного забезпечення, апаратної складової, пропускної здатності мережі, стабільності електроживлення та ефективності систем охолодження, що є необхідною умовою надійної експлуатації серверної інфраструктури.

1.3 Віртуальні сервери як елемент сучасної IT-інфраструктури

Віртуальні сервери являють собою програмні обчислювальні середовища, що відтворюють функціональні можливості фізичних серверних платформ шляхом програмної абстракції апаратних ресурсів. Їх активне впровадження стало наслідком усвідомлення того, що значна частина обчислювального

потенціалу традиційних серверів у багатьох організаціях залишається незадіяною.

За умов неповного використання апаратних ресурсів доцільним є їх логічний поділ і повторне застосування, що дозволяє ефективніше розподіляти наявні обчислювальні потужності між кількома незалежними середовищами. При цьому експлуатація фізичних серверів передбачає значні витрати, пов’язані з технічним обслуговуванням, адмініструванням, забезпеченням інформаційної безпеки та реалізацією засобів контролю, що істотно впливає на загальну вартість володіння інфраструктурою. У такому контексті перенесення серверних сервісів у віртуалізоване середовище є економічно обґрунтованим рішенням.

Найчастіше доступ до віртуальних серверів надається через спеціалізованих сертифікованих провайдерів, які експлуатують масштабні центри обробки даних із великою кількістю фізичних вузлів, розміщених у різних географічних регіонах. Такі серверні ресурси можуть орендуватися або керуватися віддалено, що надає адміністраторам можливість оперативно змінювати параметри інфраструктури відповідно до поточних навантажень, зокрема у випадках їх різкого зростання або зменшення.

Додатковою перевагою використання віртуальних серверів є модель оплати, що ґрунтується на фактичному споживанні ресурсів, зокрема обчислювальної потужності та електроенергії, а також мінімальних витратах на технічний супровід. Завдяки цьому віртуалізовані серверні рішення зазвичай є більш фінансово ефективними порівняно з традиційними фізичними серверами.

1.4 Засоби управління серверними системами та їх роль в IT-інфраструктурі

Системи керування серверами являють собою програмні рішення, призначені для адміністрування кластерів серверних ресурсів. Вони забезпечують збір даних у реальному часі щодо ключових показників роботи серверів, таких як завантаження центрального процесора, обсяг використаного

дискового простору, рівень споживання оперативної пам’яті, пропускна здатність мережевого обладнання, кількість активних підключень, сповіщення про безпеку та виявлені вразливості, а також іншу інформацію, важливу для підтримки стабільності системи.

У контексті віртуалізованих середовищ варто зазначити, що системи керування серверами не слід ототожнювати з гіпервізорами, або «моніторами віртуальних машин». Гіпервізор відповідає за створення та управління віртуальними машинами, забезпечуючи їхню роботу відповідно до заданих конфігурацій, але не контролює продуктивність або стан віртуальних серверів у реальному часі (табл. 1.1).

Таблиця 1.1 – Порівняння систем керування серверами та гіпервізорів

Параметр	Системи керування серверами	Гіпервізори
Основна функція	Моніторинг і адміністрування серверних ресурсів	Створення та управління віртуальними машинами
Контроль продуктивності	Так, збір показників CPU, RAM, дисків, мережі	Ні, лише розподіл ресурсів між VM
Моніторинг безпеки	Так, виявлення вразливостей та сповіщення	Обмежений, без аналітики безпеки VM
Збереження історії даних	Так, для аналізу тенденцій і планування ресурсів	Ні, гіпервізор не зберігає історію стану VM
Мета використання	Підтримка стабільності та оптимальної роботи IT-інфраструктури	Забезпечення роботи кількох віртуальних машин на одному хості

Отримані дані відображаються на інформаційній панелі, що дозволяє фахівцям оперативно оцінювати стан серверів. Крім того, системи керування забезпечують збереження історії подій, що дозволяє аналізувати тенденції та приймати обґрунтовані рішення щодо оптимізації роботи серверної інфраструктури.

Таблиця демонструє відмінності між двома типами систем, які часто плутають у практиці адміністрування серверів.

Системи керування забезпечують повний моніторинг і збереження даних для аналізу, тоді як гіпервізори відповідають лише за управління ресурсами віртуальних машин, не надаючи інформації про їх продуктивність або стан.

1.5 Основні категорії систем моніторингу

Сучасні системи моніторингу серверів поділяються на кілька основних моделей розгортання, які відрізняються за способом взаємодії з інфраструктурою, рівнем контролю, вимогами до встановлення та масштабованістю.

Перша група – це локальні системи моніторингу. Ці рішення встановлюються безпосередньо в IT-середовищі організації та працюють на її власних серверах й обладнанні. Локальні моніторингові платформи зберігають дані всередині корпоративної мережі та дають максимальний контроль над конфігураціями, доступом і безпекою, але вимагають значних ресурсів для обслуговування, оновлення й адміністрування [1]. Такі рішення традиційно вибирають підприємства зі строгими вимогами до конфіденційності та відповідності стандартам (рис. 1.1).

Друга група систем – це хмарні сервіси моніторингу, відомі як SaaS-сервіси – це рішення, які надаються постачальником через інтернет і не потребують розгортання на локальних серверах. Користувачі отримують доступ до готових інструментів через веб-інтерфейс, а самі сервіси управляються й підтримуються постачальником. Такий підхід дозволяє скоротити час

впровадження, знизити витрати на підтримку й швидко масштабувати рішення відповідно до навантаження, але частково обмежує можливості глибокої персоналізації. Багато постачальників SaaS-моніторингу працюють на підписці без довгострокових контрактів, що підсилює конкуренцію на ринку й стимулює розвиток функціональності [2].

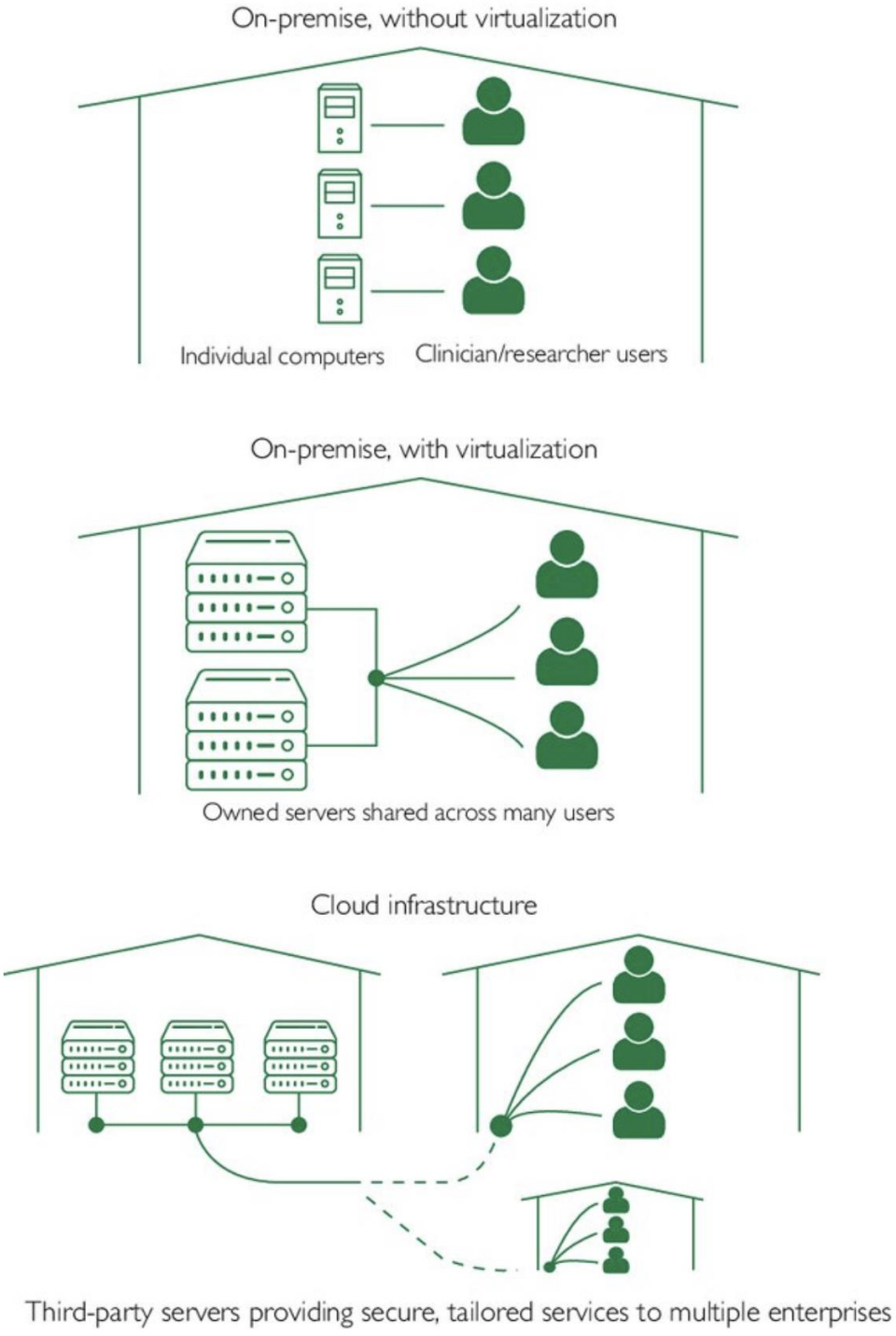

Рисунок 1.1 – Схема локальних серверів з та без віртуалізації [1]

Група мобільних систем моніторингу орієнтована на доступ із мобільних пристроїв, надаючи адміністраторам можливість отримувати сповіщення, переглядати показники та реагувати на події без прив'язки до стаціонарного робочого місця. Вони можуть бути частиною SaaS-платформи або функціонувати як модулі в існуючих рішеннях (рис. 1.2).

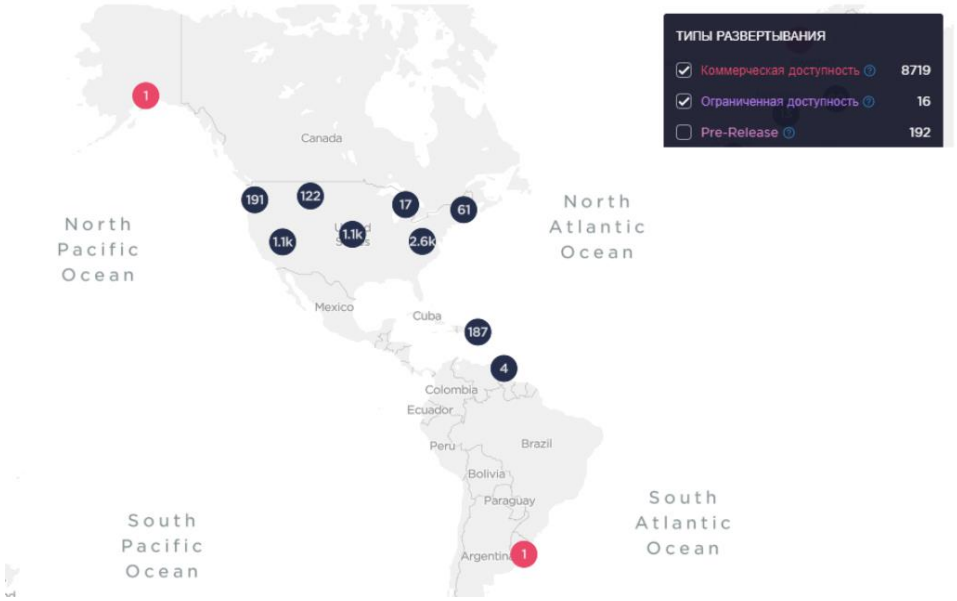

Рисунок 1.2 – Мобільний моніторинг мереж 5G [3]

Існують також гібридні моніторингові системи, що поєднують можливості локальних і хмарних систем. Такі рішення дозволяють зберігати критичну інформацію всередині корпоративної мережі, одночасно використовуючи хмарні сервіси для обробки великих обсягів даних або забезпечення зовнішньої видимості. Така архітектура часто застосовується в складних IT-ландшафтах, де частина сервісів розміщена локально, а частина – у публічних чи приватних хмарах, і вимагає єдиного огляду всіх метрик.

1.6 Огляд і ранжування інструментів моніторингу серверів

При виборі інструментів для збору та аналізу даних про працездатність серверів експерти повинні оцінювати їх здатність охоплювати всі типи серверів

організації, включно з локальними та хмарними, а також ефективність роботи з оновленими версіями систем у майбутньому. Важливою є можливість інструменту формувати сповіщення у зручному форматі, підтримувати різні канали їх доставки та забезпечувати доступ до інформації усім необхідним фахівцям. Не менш значущим є наявність механізмів для визначення причин виникнення помилок з урахуванням контексту, а не лише повідомлення про сам факт їх появи [4]. Крім того, при оцінці необхідно враховувати простоту використання інтерфейсу, зручність моніторингу та сортування подій, а також швидкість реакції на них. Останнім критерієм є якість взаємодії користувача з технічною підтримкою постачальника програмного продукту.

РОЗДІЛ 2
ПРИНЦИПИ РОБОТИ ТА КЛЮЧОВІ КОМПОНЕНТИ ZABBIX

2.1 Моделювання моніторингу серверів за допомогою Zabbix

Моніторинг серверної інфраструктури за допомогою Zabbix передбачає систематичне виконання низки послідовних дій, спрямованих на забезпечення безперервного збору, обробки та аналізу даних про стан серверів [5]. Процес починається з ідентифікації ключових серверних ресурсів та визначення метрик, які будуть відстежуватися, таких як завантаження процесора, використання оперативної пам’яті, обсяг зайнятого дискового простору, пропускна здатність мережевого обладнання та доступність критичних сервісів. Наступним кроком є налаштування агентів Zabbix на серверах, що дозволяє збирати дані в режимі реального часу та передавати їх на центральний сервер для агрегації і аналізу. Після цього визначаються тригери та порогові значення, які дозволяють автоматично виявляти відхилення від нормальної роботи, і встановлюються сповіщення для оповіщення відповідальних фахівців [6]. Завершальним етапом є періодичний аналіз історичних даних та коригування конфігурацій моніторингу для підвищення точності виявлення проблем та оптимізації ресурсів.

Для наочного відображення цього процесу доцільно представити алгоритм моніторингу у вигляді таблиці 2.1. Застосування цього алгоритму дозволяє досягти кількох важливих результатів.

По-перше, забезпечується безперервна видимість стану серверної інфраструктури, що дозволяє своєчасно реагувати на потенційні збої та зменшувати ризик критичних відмов.

По-друге, накопичення історичних даних дає змогу проводити аналітичні дослідження і прогнозувати пікові навантаження, що дозволяє оптимально розподіляти обчислювальні ресурси.

По-третє, централізоване управління конфігураціями і сповіщеннями підвищує ефективність взаємодії IT-персоналу та скорочує час реагування на інциденти.

Таблиця 2.1 – Алгоритм моніторингу серверів за допомогою Zabbix

Етап	Дія	Результат
1	Визначення ключових серверних ресурсів та метрик	Сформований перелік показників для моніторингу
2	Встановлення та налаштування агентів Zabbix на серверах	Забезпечення збору даних у реальному часі
3	Конфігурування тригерів та порогових значень	Автоматичне виявлення відхилень від нормальної роботи
4	Налаштування сповіщень для відповідальних фахівців	Миттєве оповіщення про проблеми та інциденти
5	Збір та збереження історичних даних	Формування бази для аналізу тенденцій та планування ресурсів
6	Аналіз даних та коригування конфігурацій	Підвищення ефективності моніторингу та оптимізація роботи серверів

Таким чином, системне впровадження Zabbix у моделюванні моніторингу серверів є ефективним інструментом підтримки стабільності, надійності та продуктивності IT-інфраструктури.

2.2 Конструкція та компоненти системи моніторингу Zabbix

Система Zabbix функціонує завдяки комплексу взаємопов'язаних програмних компонентів, кожен з яких виконує певну роль у процесі моніторингу. Центральним елементом є сервер Zabbix, який відповідає за збір інформації з агентів, встановлених на моніторингових хостах.

Сервер виконує роль основного сховища, де накопичуються конфігураційні параметри, статистичні дані та відомості про стан систем і служб [7].

Для збереження отриманих даних використовується система управління базами даних. Підтримуються різні платформи, зокрема MySQL, PostgreSQL, Oracle, SQLite та IBM DB2, що забезпечує гнучкість інтеграції з існуючими IT-інфраструктурами. Доступ до інформації та налаштування системи забезпечується через веб-інтерфейс Zabbix, який характеризується логічною структурою та інтуїтивно зрозумілим користувацьким середовищем.

Це дозволяє адміністраторам керувати моніторингом з будь-якого пристрою, включаючи персональні комп'ютери, планшети та смартфони.

Зібрані дані обмежуються попередньо визначеними метриками, налаштованими адміністратором для цільового моніторингу (рис. 2.1).

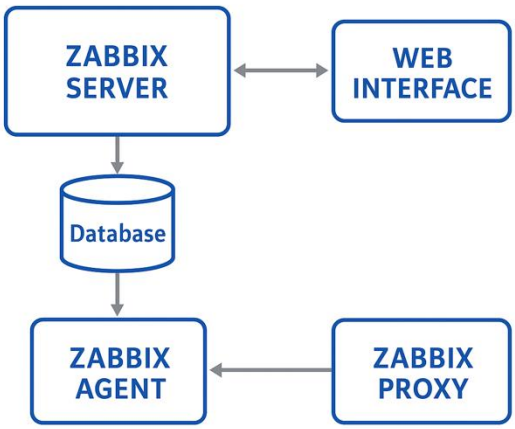

Рисунок 2.1 – Архіткектурна схема системи моніторингу

Веб-інтерфейс зазвичай розгортається на тому ж сервері, що й Zabbix-сервер, проте за потреби може функціонувати на окремій машині.

Zabbix agent є програмним модулем, встановленим на хостах, який здійснює збір даних про локальні ресурси та процеси. Агент передає інформацію на центральний сервер, повідомляючи про поточний стан систем і події.

Завдяки цим компонентам Zabbix здатен відстежувати тисячі показників роботи серверів, віртуальних машин, додатків та мережевого обладнання в реальному часі. Це дозволяє своєчасно виявляти потенційні проблеми та вживати заходів до того, як вони вплинуть на роботу сервісів та користувацький досвід

2.3 Процеси передачі та обробки даних

Агент Zabbix здійснює збір інформації, що представляє інтерес для системних адміністраторів, зокрема даних щодо поточного завантаження процесора, продуктивності окремих сервісів (HTTP, SSH, FTP тощо) та інших системних параметрів. Отримана інформація передається на центральний сервер, де вона формується у вигляді зручних для сприйняття таблиць та графіків [8].

Збереження даних здійснюється у реляційних базах даних, доступ до яких забезпечується через веб-інтерфейс з інтуїтивно зрозумілим керуванням та навігацією.

Агент Zabbix підтримує два основні підходи збору інформації: пасивний із опитуванням сервера та активний із перехопленнями. Пасивний режим передбачає, що сервер Zabbix або проксі надсилає агенту запит на отримання конкретного значення, після чого агент обробляє запит та повертає результат.

У активному режимі агент самостійно ініціює опитування, спочатку отримуючи від сервера або проксі список елементів, що підлягають перевірці, а потім періодично надсилає зібрані дані відповідно до цього списку.

Починаючи з версії Zabbix 3.0, реалізована підтримка зашифрованого каналу зв’язку між агентом і сервером, що забезпечує безпечну передачу даних та підвищує загальний рівень інформаційної безпеки системи

2.4 Функціонал оповіщення адміністратора

Система оповіщення Zabbix забезпечує оперативне інформування адміністратора про зміни стану моніторингових об’єктів та виникнення потенційних проблем. Вона ґрунтується на попередньо визначених тригерах, які реагують на певні умови, наприклад перевищення порогових значень завантаження процесора, збої сервісів або недоступність хостів [9].

Повідомлення можуть надсилатися різними каналами зв’язку, такими як електронна пошта, SMS, месенджери або веб-повідомлення, що дозволяє своєчасно реагувати на інциденти та знижувати ризики простою системи. Конфігурація системи оповіщення передбачає гнучке налаштування інтервалів, отримувачів та пріоритетів сповіщень, що забезпечує ефективне управління інцидентами та підвищує загальну надійність IT-інфраструктури.

.

2.5 Захист інформації та криптографічні протоколи

Система Zabbix забезпечує шифрування даних між компонентами за допомогою протоколу Transport Layer Security версій 1.2 та 1.3, вибір конкретної версії визначається використовуваною криптографічною бібліотекою. Підтримується як шифрування на основі сертифікатів, так і шифрування з використанням попередньо узгодженого ключа. Зашифроване з’єднання може застосовуватися для обміну даними між сервером Zabbix, проксі, компонентом Zabbix-відправник, який відповідає за надсилання даних, та Zabbix_get, що забезпечує отримання інформації. Крім того, шифрування підтримується при взаємодії між базою даних та веб-інтерфейсом системи [10-11].

Використання шифрування у Zabbix є необов’язковим і може налаштовуватися для окремих компонентів системи. Деякі проксі або агенти можуть застосовувати шифрування з попередньо узгодженим ключем, інші використовують пряме з’єднання без шифрування, а деякі працюють із сертифікатами. При цьому сервер або проксі здатні одночасно підтримувати різні конфігурації шифрування для різних джерел даних. Однією з переваг системи є те, що демон Zabbix використовує один порт для обробки як зашифрованих, так і незашифрованих повідомлень, що спрощує налаштування міжмережевих екранів.

Необхідно враховувати певні обмеження шифрування. Закриті ключі зберігаються у відкритому тексті та зчитуються компонентами Zabbix під час запуску. Вбудоване шифрування не гарантує захист від певних видів атак, а відкриті ключі зберігаються у базі даних у відкритому вигляді. Функція виявлення мережі не підтримує шифрування, тому якщо агент налаштований на відмову від незашифрованих підключень, відповідні перевірки виконані не будуть. Всі зашифровані з’єднання відкриваються через повне TLS-рукостискання, кешування сесій поки не реалізоване. Використання шифрування збільшує час встановлення з’єднання та передачі даних, що залежить від затримки мережі. Наприклад, при затримці пакета близько 100 мілісекунд встановлення TCP-з’єднання та передача даних займають приблизно 200 мілісекунд, тоді як з увімкненим TLS цей час може зростати до 1000 мілісекунд.

2.6 Оцінка продуктивності та доступності баз даних

Усі сервери, незалежно від того, чи є вони фізичними чи віртуальними, призначені для розгортання на них певного програмного забезпечення. Відповідно, стан і продуктивність встановлених систем та служб безпосередньо впливають на роботу всієї IT-інфраструктури. Тому завдання моніторингу включає не лише контроль завантаження центрального процесора, використання

оперативної пам’яті або дискового простору, а й спостереження за роботою програмного забезпечення, що функціонує на серверах.

Система Zabbix дозволяє налаштовувати моніторинг більшості відомих програмних продуктів, проте для цього потрібне належне конфігурування адміністратором. Окрім встановлення агента Zabbix на сервері, з якого здійснюється збір даних, необхідно також інтегрувати додаткове програмне забезпечення. Наприклад, для отримання інформації про стан баз даних сервер Zabbix використовує ODBC, що забезпечує відкрите підключення до баз даних. Підтримуються різні СУБД, серед яких Oracle, PostgreSQL, MySQL, MSSQL, Sybase ASE, SAP HANA та DB2 [12].

Кожна база даних має власний драйвер ODBC, що може суттєво відрізнятися за функціональністю. Це означає, що спосіб підключення до однієї СУБД може бути непридатним для іншої, а більшість драйверів ODBC не реалізують повний набір функцій стандарту. По суті, ODBC забезпечує доступ до бази даних через її API по мережі, при цьому тісного інтегрованого зв’язку між сервером Zabbix та базою даних немає. Сервер лише формує запит із визначеною періодичністю, який передається через ODBC до бази даних для виконання. Час очікування відповіді визначається параметрами ODBC і не накладає додаткових обмежень на сервер Zabbix

. Більш детальну схему функціонування цього ланцюга процесів можна переглянути на рисунку 2.2.

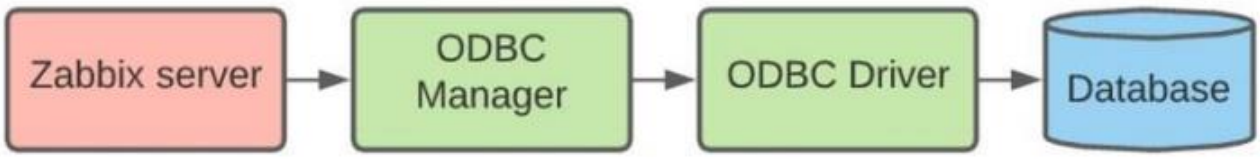

Рис. 2.2 – Ланцюг процесу передачі запиту від сервісу Zabbix на базу даних, за участю ODBC-драйвера

Адміністратору необхідно враховувати, що для налаштування з’єднання через ODBC слід перевірити сумісність версії драйвера з поточною версією

сервера Zabbix. Після інсталяції драйвера потрібно вказати дані DSN, тобто назву джерела даних. Невірно введена назва призведе до помилки підключення, і система виведе відповідне повідомлення про збої в з'єднанні.

Після налаштування драйвера та конфігураційних файлів odbc.ini і odbcinst.ini користувач може перевірити підключення до джерела даних безпосередньо з сервера Zabbix через консоль [13]. Для цього використовують команду isql, яка дозволяє здійснити спробу підключення до потрібної бази даних (як показано на рисунку 2.3). Наприклад, у випадку з MSSQL, дані про стан всієї СУБД зберігаються в базі даних msdb.

Рисунок 2.3 – Використання команди «isql MySQL» для тестування налаштування конфігурації ODBC

У разі успішного підключення можна стверджувати, що драйвер налаштовано коректно, а база даних має наданий повний необхідний доступ для виконання запитів.

.

2.7 Автоматизація реакцій на події через веб-хуки

Веб-хуки є механізмом, що дозволяє додавати або змінювати поведінку веб-додатків та автоматично реагувати на події шляхом зворотних викликів. У контексті системи моніторингу Zabbix веб-хук виконує виклик сторонньої

служби через протоколи HTTP або HTTPS при настанні певної події, наприклад виникненні помилки, повідомляючи про неї зовнішній сервіс.

Багато сучасних рішень надають API, що забезпечує інтеграцію із зовнішніми системами за допомогою веб-хуків.

У Zabbix веб-хуки реалізовані на мові JavaScript, що спрощує процес написання скриптів, оскільки не потребує знання специфічного синтаксису системи. Широке поширення JavaScript дозволяє швидко навчитися створювати власні сценарії або скористатися готовими рішеннями з відкритих джерел.

Функціонально веб-хук виконує послідовність дій для досягнення певного результату.

Наприклад, у разі інтеграції з сервісом ARI, JavaScript-сценарій може передавати, отримувати або оновлювати дані у зовнішньому сервісі.

Практичне використання може включати такі кроки: авторизацію на сервісі для отримання токена, створення запиту з маркером для генерації нового квитка та додавання коментаря до створеного квитка з інформацією про проблему.

Така схема дозволяє автоматизувати взаємодію між Zabbix та зовнішніми сервісами, підвищуючи ефективність реагування на події (схематичний приклад показаний на рисунку 2.4).

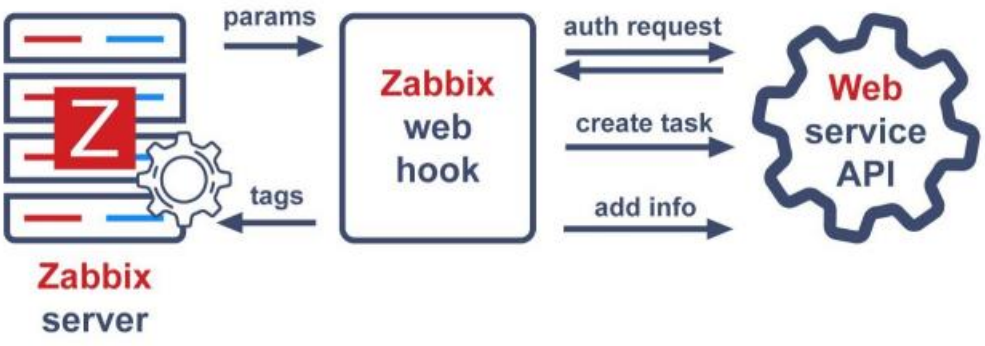

Рисунок 2.4 – Схема передачі даних між Zabbix, веб-хуком та API

Хоча конкретні деталі реалізації можуть відрізнятися залежно від сервісу, загальна концепція веб-хуків залишається незмінною. На даний час Zabbix

пропонує широкий набір готових веб-хуків для найпопулярніших зовнішніх сервісів, тому їх налаштування зазвичай зводиться до кількох простих кроків. Зокрема, необхідно згенерувати ключ API у сторонньому сервісі, вказати його в конфігурації Zabbix, встановити URL кінцевої точки та визначити необхідні параметри для роботи веб-хука.

При розробці власного веб-хука слід враховувати, що, хоча всі API функціонують за подібним принципом, методи викликів та структура запитів можуть відрізнятися. Крім того, адміністратору важливо розуміти загальний процес взаємодії, оскільки ефективна інтеграція неможлива без розуміння того, як Zabbix взаємодіє з інтегрованими сервісами.

РОЗДІЛ 3
РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ

3.1 Підготовчі заходи та перевірка інтеграційної сумісності систем

Для належної роботи системи Zabbix необхідно вибрати одну з операційних систем сімейства Linux, таких як CentOS, Debian, Oracle Linux, Raspberry Pi OS, Red Hat Enterprise Linux, Rocky Linux, SUSE Linux Enterprise Server або Ubuntu, зокрема її версію для архітектури arm64. Вибір програмного забезпечення для налаштування моніторингу був обґрунтований кількома критеріями.

Для цього було обрано сервер Zabbix версії 6.0 LTS, оскільки ця версія пропонує останні оновлення та підтримку на довгостроковій основі, що робить її більш стабільною для використання в порівнянні з версіями 6.2 та 6.4, що знаходяться на стадії попереднього випуску. Операційна система Ubuntu 22.04 (Jammy) була вибрана через свою популярність серед користувачів Linux, а також завдяки наданій довгостроковій підтримці, що робить її ідеальним вибором для розгортання Zabbix. Крім того, її широка поширеність сприяє більш легкому пошуку рішень для вирішення технічних проблем.

Як система керування базами даних було обрано MySQL, оскільки вона є однією з найбільш поширених СУБД завдяки своїй простоті у використанні та широкому документуванню. Ці фактори значно полегшують адміністрування та пошук необхідних ресурсів під час налаштування. Веб-сервер Nginx був обраний через свою високу ефективність порівняно з Apache2, завдяки більшій гнучкості та простоті налаштування, зокрема у частині перенаправлення запитів.

3.2 Інсталяція та конфігурація веб-сервера

Для встановлення необхідного програмного забезпечення було створено віртуальний сервер з назвою «graylog-lab». Перед початком інсталяції слід перевірити систему на наявність оновлень та, у разі їх виявлення, виконати

процедуру оновлення. Це забезпечує актуальність компонентів операційної системи та стабільність роботи встановлюваних програмних модулів:

- sudo apt-get update (команда для пошуку та завантаження оновлень);
- sudo apt-get upgrade -y (команда для встановлення оновлень).

Після завершення оновлення системи можна перейти до безпосередньої інсталяції веб-сервера Nginx. Для цього в консолі слід виконати команду sudo apt install nginx, результат виконання якої наведено на рисунку 3.1.

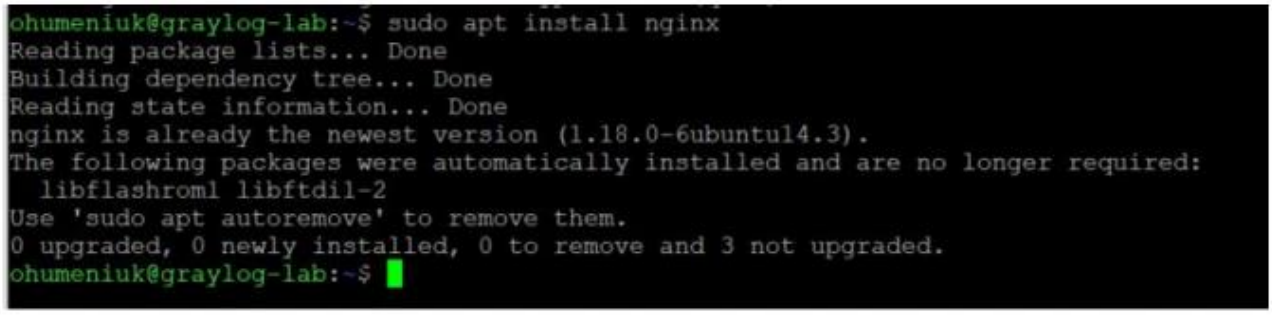

Рисунок 3.1 – Встановлення Nginx

Після завершення інсталяції слід провести тестову перевірку працездатності веб-сервера Nginx шляхом відкриття відповідного URL-адреси у веб-браузері. У випадку даного сервера, він має назву graylog-lab.

3.3 Інтеграція MySQL у систему моніторингу Zabbix

Для інсталяції системи керування базами даних MySQL на віртуальній машині з операційною системою Ubuntu необхідно виконати команду sudo apt install mysql-server (рис. 3.2). Після виконання цієї команди користувач може перевірити коректність роботи серверу за допомогою команди sudo systemctl start mysql.service, що ініціює запуск служби MySQL.

Крім того, користувач може негайно приступити до налаштування системи, якщо сервер успішно запущено.

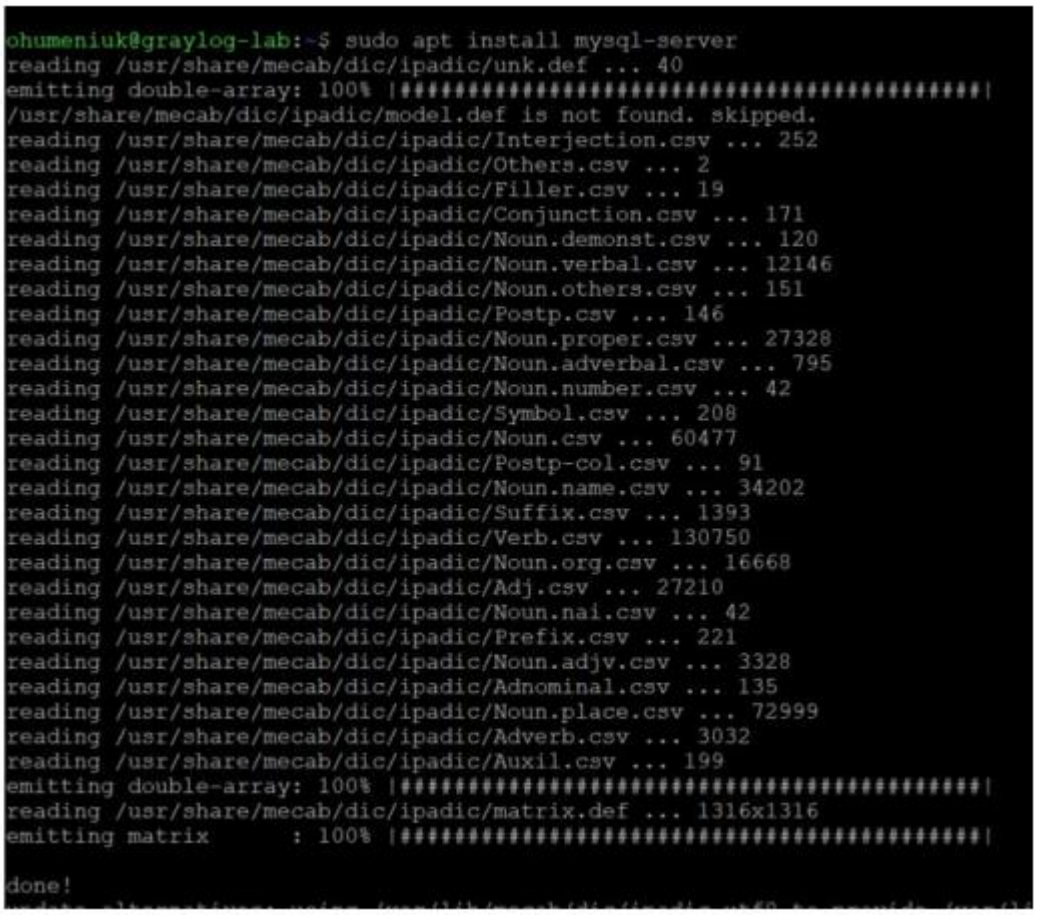

Рисунок 3.2 – Процес інсталяції сервера MySQL на платформі Ubuntu

Після виконання команди інсталяції сервер MySQL було встановлено та запущено, проте він не запропонував встановити пароль або внести інші зміни конфігурації. Такий стан справ робить нову інсталяцію менш безпечною, тому необхідно вжити додаткових заходів для підвищення рівня безпеки.

Зазвичай для цього запускають сценарій безпеки, що входить до складу MySQL.

Він змінює деякі менш безпечні параметри за замовчуванням, наприклад налаштування доступу користувачів і віддалених входів для облікового запису root.

Раніше для цього використовувалась команда mysql_secure_installation. Проте з липня поточного року при спробі запуску цього сценарію користувач може отримати помилку (рис. 3.3), оскільки за замовчуванням обліковий запис root у Ubuntu не налаштовано для автентифікації за допомогою пароля.

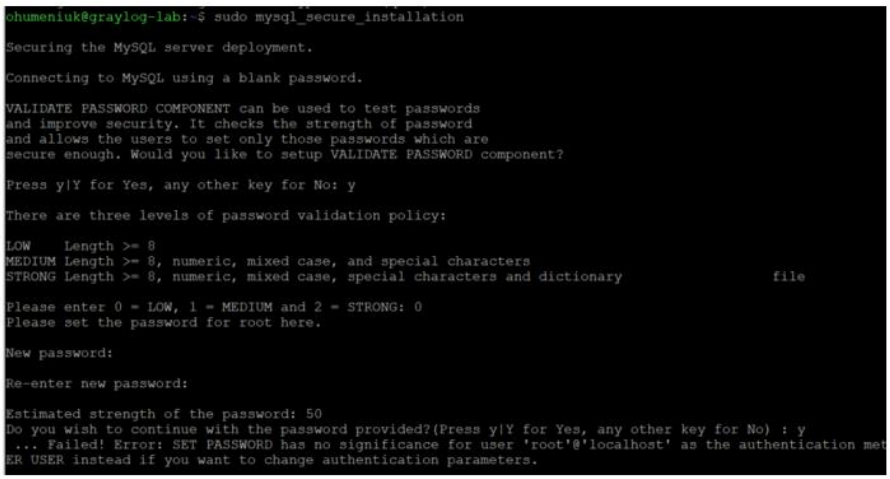

Рисунок 3.3 – Помилка, що виникає при спробі запуску сценарію «mysql_secure_installation»

Незважаючи на це, сценарій mysql_secure_installation виконує ряд важливих дій для підвищення безпеки інсталяції, тому його рекомендовано запускати перед експлуатацією MySQL для управління даними. Щоб забезпечити можливість запуску сценарію, необхідно змінити метод автентифікації користувача root на використання пароля. Для цього слід увійти до системи управління даними за допомогою команди sudo mysql і виконати команду для зміни способу автентифікації: ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';. Результат виконання цієї команди наведено на рисунку 3.4.

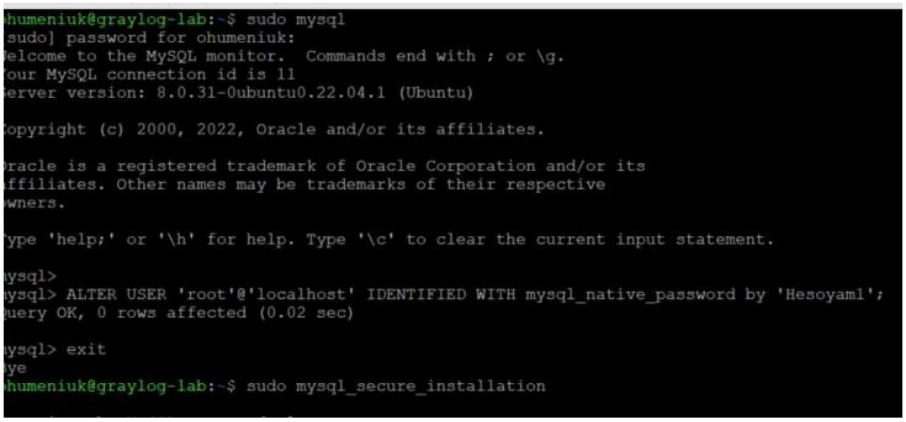

Рисунок 3.4 – Створення та застосування паролю для аутентифікації кореневого користувача

Такий підхід дозволяє зменшити ризики несанкціонованого доступу, забезпечити контроль над обліковими записами та підготувати систему до надійного збору та збереження даних моніторингу.

3.4 Інтеграція PHP у середовище веб-сервера Nginx

Для коректної роботи динамічних веб-додатків у середовищі веб-сервера Nginx необхідно забезпечити інтеграцію з PHP-мовою програмування. PHP виконує функції обробки серверної логіки, що дозволяє генерувати динамічний контент на веб-сторінках, взаємодіяти з базами даних та забезпечувати обробку запитів користувачів.

Інтеграція PHP з Nginx передбачає налаштування проміжного процесу обробки запитів, який передає HTTP-запити від веб-сервера до PHP і повертає сформовані відповіді клієнту. У сучасних конфігураціях для цього зазвичай використовують PHP-FPM, що дозволяє ефективно керувати процесами виконання PHP, забезпечуючи високу продуктивність і стабільність роботи веб-сервера. (рис. 3.5).

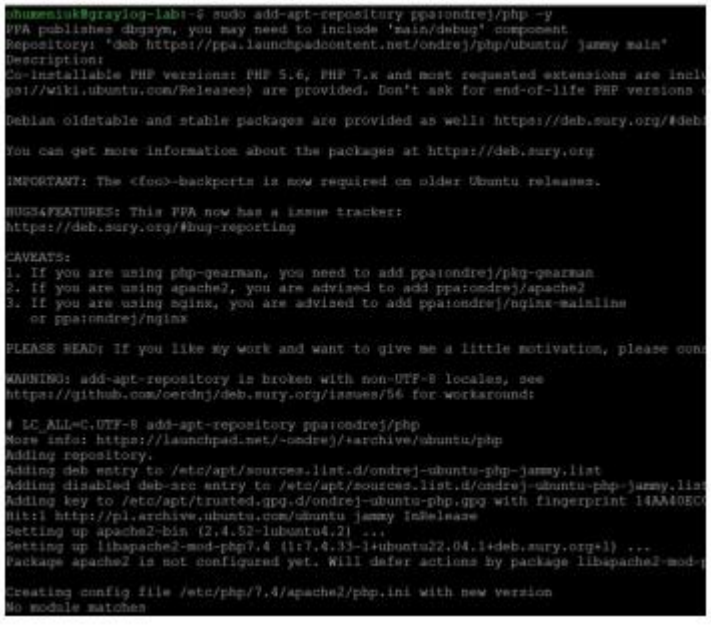

Рисунок 3.5 – Результат виконання установки PHP v7.4 з окремого репозиторію

У процесі налаштування слід встановити необхідні пакети PHP та PHP-FPM на сервері, а також відредагувати конфігураційні файли Nginx для визначення обробки запитів до PHP-скриптів.

Це забезпечує коректну маршрутизацію запитів та інтеграцію з іншими компонентами системи, такими як база даних MySQL, що дозволяє створювати функціональні веб-інтерфейси для моніторингу та управління серверами (рис. 3.6).

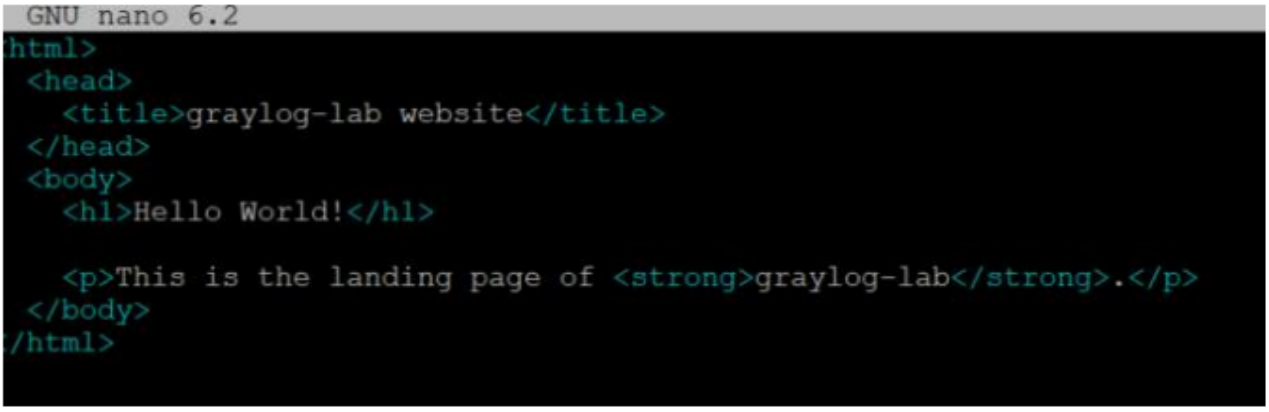

Рисунок 3.6 – Налаштування файлу конфігурації для перевірки роботи Nginx та PHP

Правильна конфігурація PHP-FPM та налаштування маршрутизації запитів у Nginx забезпечують стабільну і продуктивну роботу веб-сервера, а також коректну взаємодію з базами даних і іншими компонентами системи моніторингу.

Завдяки цьому створюється надійне середовище для реалізації веб-інтерфейсів, необхідних для управління та моніторингу серверних ресурсів.

3.5 Інсталяція сервера та конфігурація доступу до веб-інтерфейсу

Для забезпечення централізованого моніторингу серверів та програмного забезпечення необхідно розгорнути сервер Zabbix та налаштувати доступ до його веб-інтерфейсу. Веб-інтерфейс надає користувачам зручні інструменти для

візуалізації зібраних даних, управління об’єктами моніторингу та налаштування сповіщень про події. Коректна інсталяція та конфігурація сервера, а також забезпечення безпечного доступу до веб-інтерфейсу є ключовими для ефективного використання системи моніторингу та стабільної роботи всієї інфраструктури (рис. 3.7).

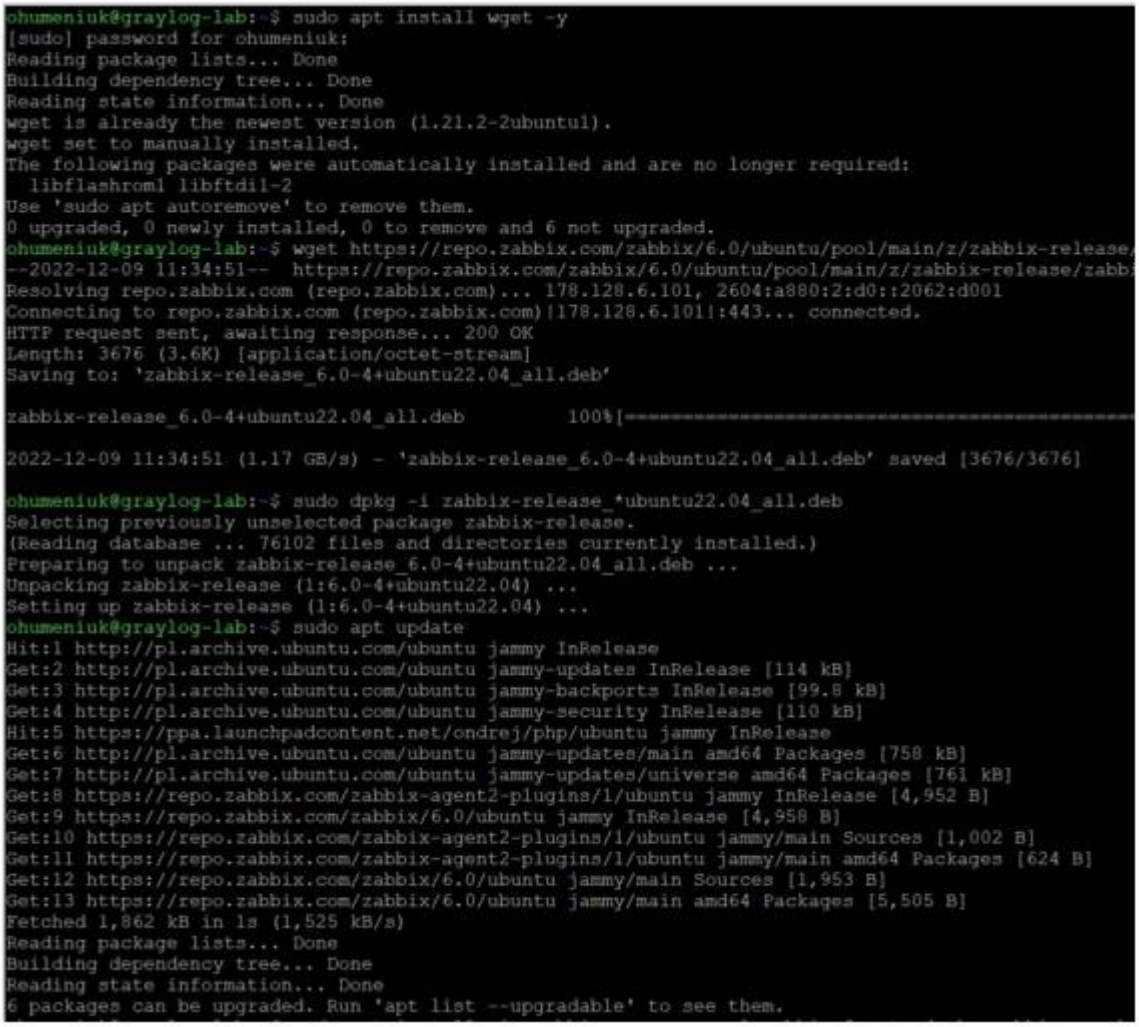

Рисунок 3.7 – Завантаження та підготовка файлів Zabbix із офіційного сховища

Процес встановлення сервера Zabbix включає декілька ключових етапів, які забезпечують коректну роботу системи моніторингу та інтеграцію з іншими компонентами інфраструктури. На першому етапі здійснюється завантаження необхідних пакунків із офіційного сховища Zabbix та їх підготовка для інсталяції. Після цього виконується інсталяція серверних компонентів, що

включає сам Zabbix-сервер, веб-інтерфейс та інструменти для роботи з базою даних (рис. 3.8).

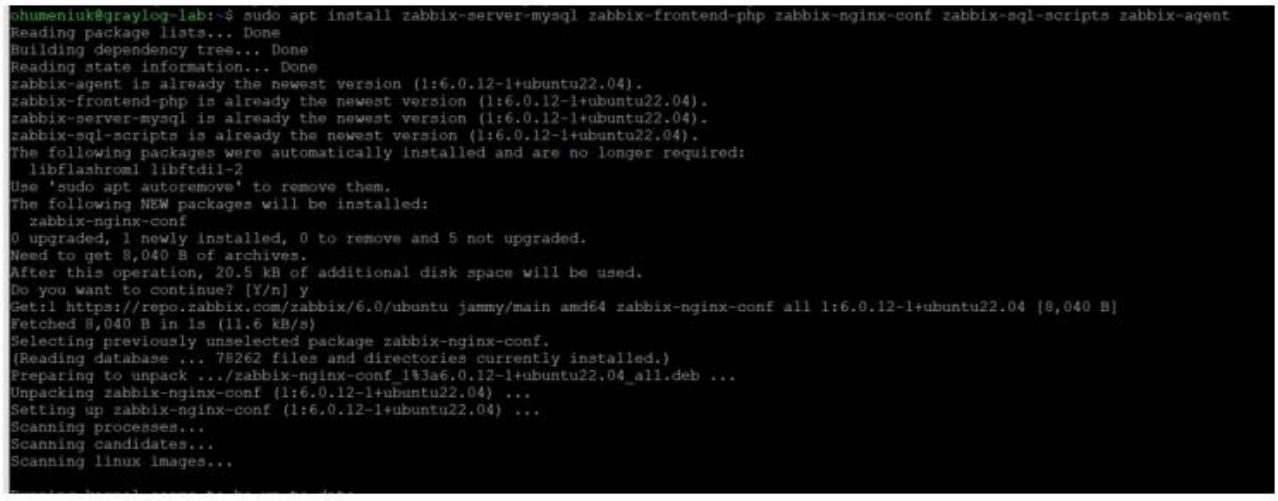

Рисунок 3.8 – Послідовність кроків при успішному розгортанні Zabbix-сервера

Підключення до системи управління базами даних. Для роботи з базою даних Zabbix зазвичай використовується MySQL/MariaDB або PostgreSQL. Спершу необхідно підключитися до СУБД від імені адміністративного користувача (рис. 3.9).

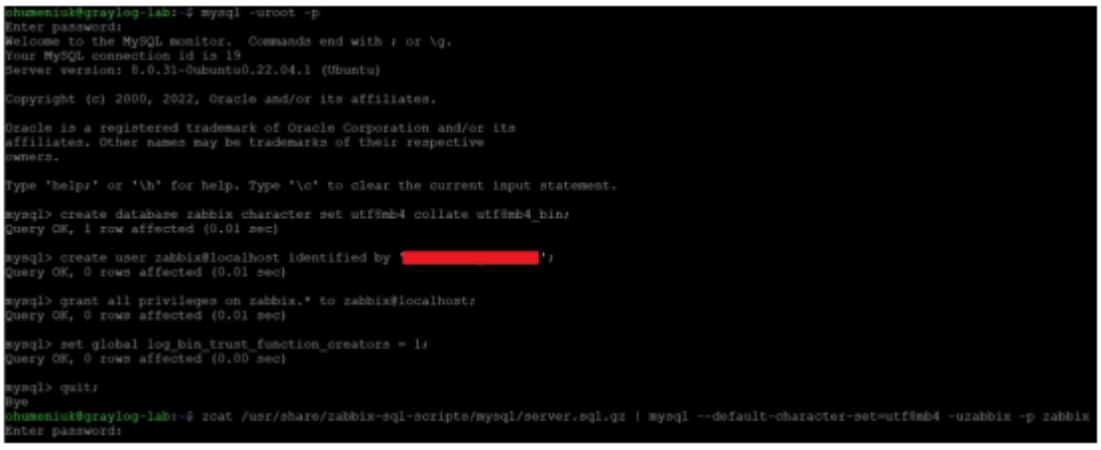

Рисунок 3.9 – Створення облікового запису Zabbix і імпорт структури бази даних

Параметр log_bin_trust_function_creators у MySQL/MariaDB визначає, чи дозволяється створення користувачами бази даних функцій та процедур у

середовищі з увімкненою бінарною реплікацією. Під час імпорту початкової структури бази Zabbix цей параметр часто тимчасово активується (ON), щоб уникнути помилок при створенні збережених функцій та тригерів.

Після успішного завершення імпорту бази даних рекомендується деактивувати цей параметр (рис. 3.10) для підвищення безпеки, оскільки увімкнення дозволяє користувачам створювати функції без додаткових перевірок, що може становити ризик у продуктивному середовищі.

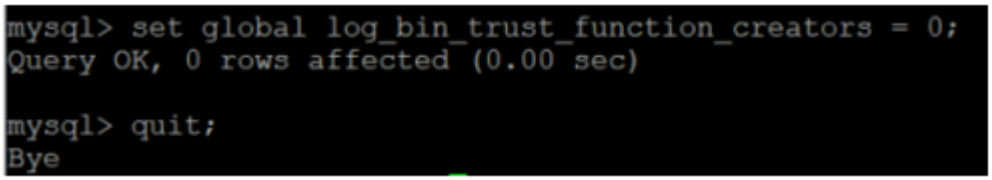

Рисунок 3.10 – Вимкнення параметра безпеки після імпорту бази даних сервера

Для коректної роботи Zabbix-сервера необхідно вказати параметри підключення до бази даних у конфігураційному файлі (рис. 3.11).

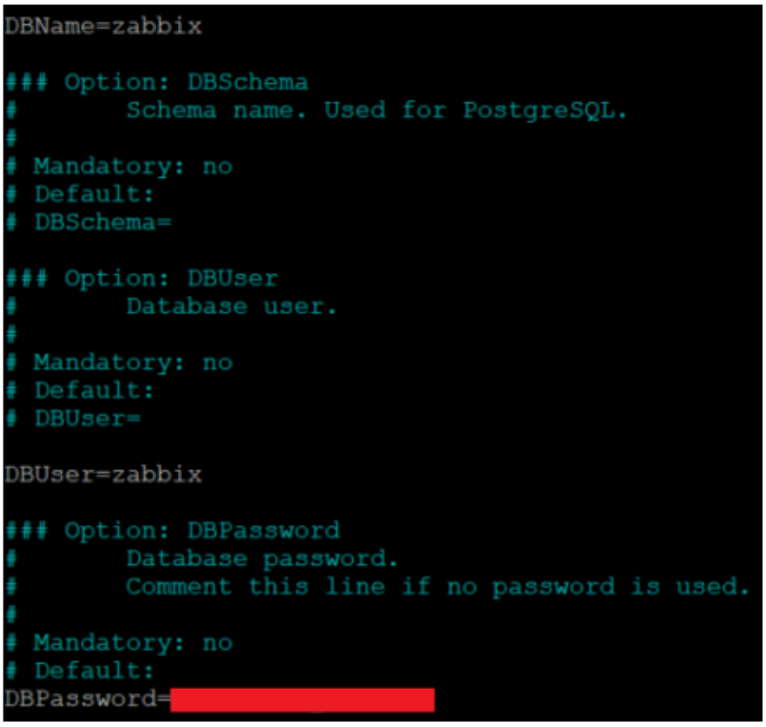

Рисунок 3.11 – Запис у конфігураційному файлі Zabbix-сервера даних для підключення до БД

Після виконання наведених вище налаштувань, щоб підключення до веб-інтерфейсу Zabbix-сервера відображалося як безпечне в браузері, можна підключити сертифікати, порти для доступу до ресурсу, а також перенаправити на правильну веб-адресу. Ці налаштування для Zabbix-сервера показано на малюнку 3.12.

Рисунок 3.12 – Огляд змін у конфігураційному файлі сервісу Nginx

Інформація про адресу веб-інтерфейсу задається у конфігураційному файлі служби Nginx, розташованому на сервері Ubuntu за стандартним шляхом. Це дозволяє при доступі до веб-інтерфейсу відображати не стандартну адресу

graylog-lab, а задану в конфігурації назву сервісу, наприклад zabbix-lab.domain.com. У конкретному випадку домен сайту визначено як cfg.com.ua, оскільки сервер належить до інфраструктурного середовища сторонньої компанії, яка надала доступ до ресурсів (рис. 3.13).

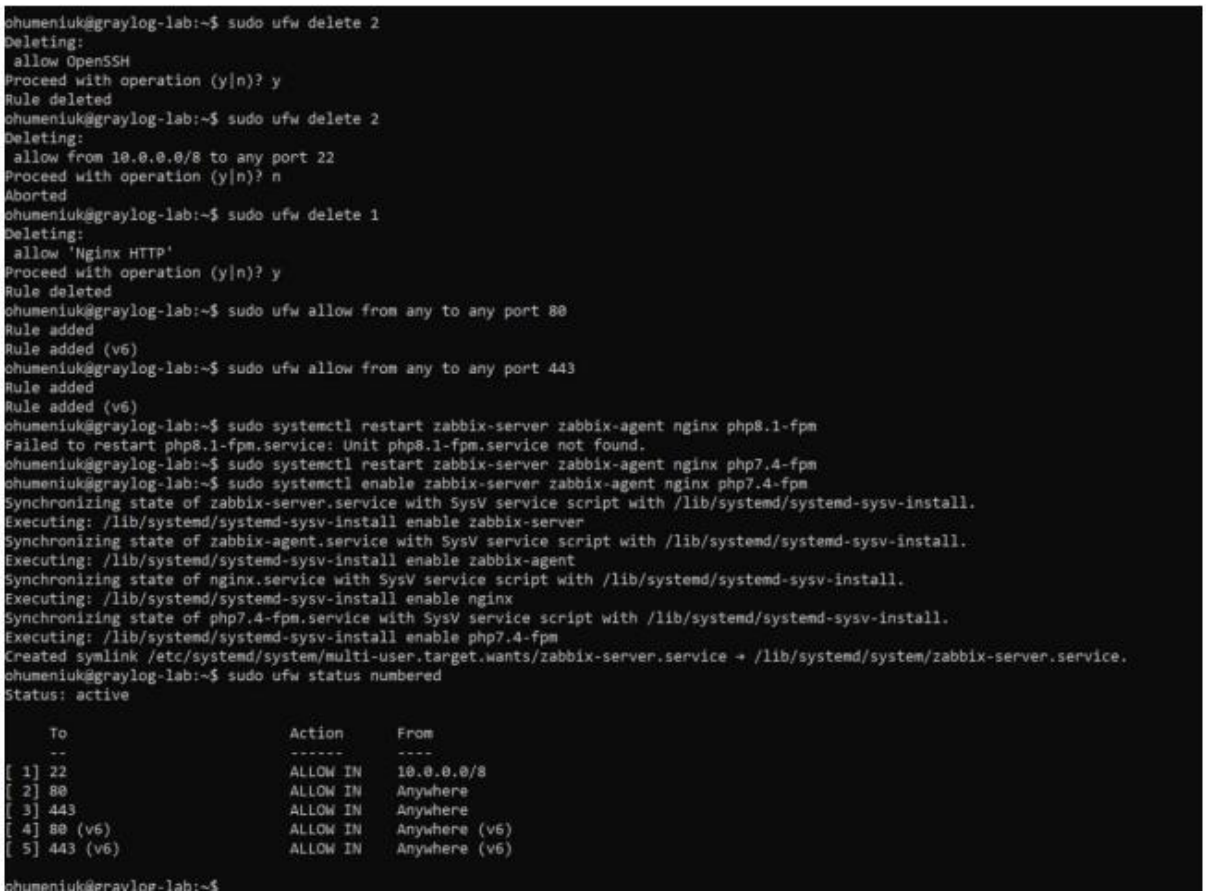

Рисунок 3.13 – Налаштування правил брандмауера для роботи веб-клієнта Zabbix-сервера

Перевірка стану роботи сервісу Zabbix (рис. 3.14) здійснюється за допомогою системних команд, які дозволяють визначити, чи запущено сервер та чи функціонує він коректно. У середовищі Ubuntu для цього використовується команда перевірки статусу служби, яка відображає стан процесу, час останнього запуску та наявність можливих помилок. При необхідності сервіс можна перезапустити або переглянути журнал роботи для діагностики неполадок.

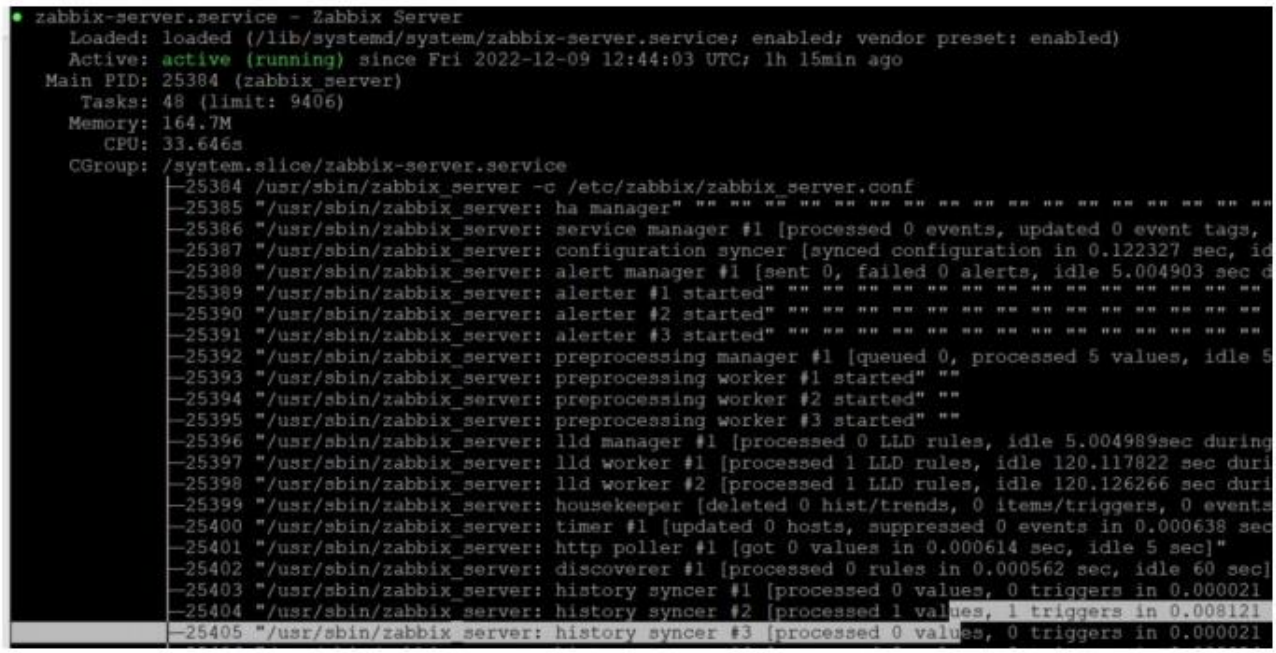

Рисунок 3. 14 – Перевірка стану роботи сервісу Zabbix

Вдалий запуск веб-інтерфейсу Zabbix-сервера (рис. 3.15) свідчить про те, що сервер коректно підключився до бази даних, служби моніторингу та веб-сервера працюють належним чином, а конфігурація системи виконана правильно.

Рисунок 3.15 – Коректне завантаження веб-інтерфейсу Zabbix-сервера

Це означає, що користувач має доступ до адміністративних функцій, відображаються необхідні панелі та метрики, і система готова до збору та аналізу даних про стан мережі та ресурсів.

Серед мікросервісів, для встановлення яких Zabbix додатково вимагає: PHP bcmath, PHP mbstring, PHP gd, PHP xmlwriter, HP LDAP

Для встановлення цих мікросервісів було виконано декілька команд, а саме:

- sudo apt-get -y встановити php7.4-bcmath;
- sudo apt-get -y встановити php7.4-mbstring;
- sudo apt -y встановити php7.4-gd;
- sudo apt-get install php7.4-xml;
- sudo apt-get install php7.4-ldap.

Після виконання зазначених команд слід оновити сторінку веб-інтерфейсу для перевірки коректності налаштувань сервера веб-агента. У разі успішної установки можна переходити до наступного етапу конфігурації. На цьому кроці веб-служба запитує введення параметрів підключення до системи управління базами даних MySQL, зокрема логіну та пароля користувача, створеного під час попереднього налаштування СУБД, що ілюструється на рисунку 3.16.

Рисунок 3.16 – Внесення даних для підключення до бази даних Zabbix-сервера при налаштуванні веб-сервісу

Налаштування колірної теми, назви та часового поясу веб-сервера Zabbix дозволяє адаптувати інтерфейс системи під потреби користувача та регіональні особливості.

Колірна тема визначає візуальне оформлення панелей та графіків, забезпечуючи зручність сприйняття даних. Встановлення назви веб-сервера дозволяє однозначно ідентифікувати конкретний інстанс Zabbix у мережі, а налаштування часового поясу гарантує коректне відображення часових міток у звітах, подіях та графіках, що є критично важливим для моніторингу та аналізу стану системи в реальному часі (рис. 3.17).

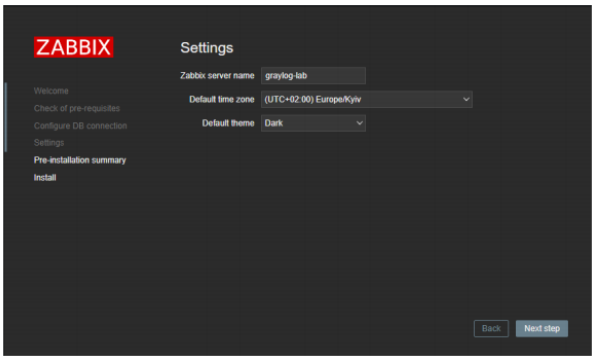

Рисунок 3. 17 – Налаштування колірної теми, назви та часового поясу веб-сервера Zabbix

Перевірка даних по налаштуванню веб-сервера Zabbix полягає у підтвердженні коректності параметрів конфігурації, введених під час початкового налаштування (рис. 3.18).

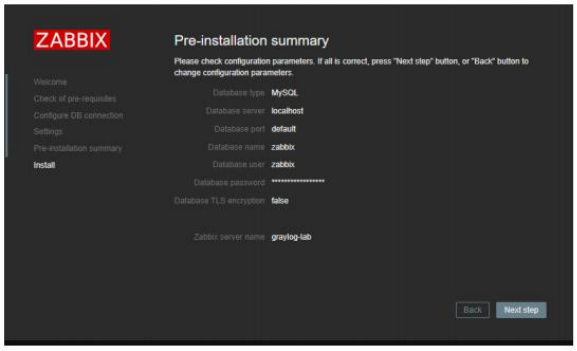

Рисунок 3.18 – Перевірка даних по налаштуванню веб-сервера Zabbix

Це включає перевірку правильності підключення до бази даних, відображення назви сервера, відповідності часового поясу та застосованої колірної теми. Успішне відображення всіх цих налаштувань у веб-інтерфейсі свідчить про готовність сервера до збору та візуалізації даних моніторингу.

Інформування про успішне встановлення та налаштування веб-сервера Zabbix полягає у наданні користувачу підтвердження того, що всі необхідні компоненти системи встановлено та налаштовано коректно. Це включає перевірку доступності веб-інтерфейсу, правильності підключення до бази даних, застосування конфігураційних параметрів, таких як назва сервера, часовий пояс та колірна тема (рис. 3.19).

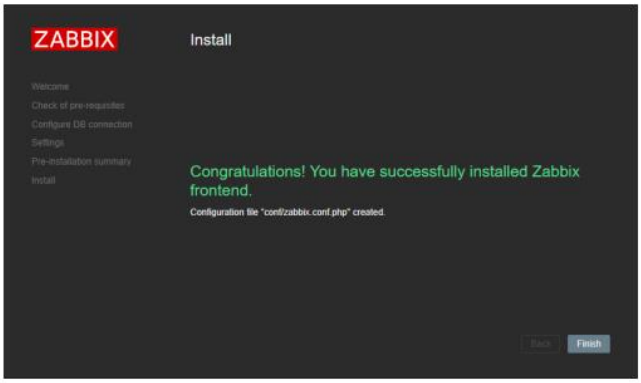

Рисунок 3. 19 – Підтвердження коректного встановлення та конфігурації веб-сервера Zabbix

Успішне повідомлення свідчить про готовність системи до роботи, збору та візуалізації даних моніторингу в реальному часі, а також підтверджує відсутність критичних помилок у процесі інсталяції та первинного налаштування.

3.6 Розгортання Zabbix-агента для моніторингу сервера

Першим кроком при налаштуванні агента Zabbix є створення хоста, який міститиме інформацію для підключення до агента. Для цього у веб-інтерфейсі на

панелі адміністрування необхідно перейти до розділу Конфігурація, далі Хости, після чого вибрати Створити хост. У відповідних полях слід вказати дані про сервер, до якого потрібно встановити доступ, що ілюструється на рисунку 3.20. Система запропонує вибір групи хостів; у випадку відсутності готової групи можна ввести нову назву, після чого група буде створена автоматично. У наведеному прикладі підключення здійснюється до віртуального сервера під назвою labsrv, який використовується для тестування різноманітного програмного забезпечення, включаючи скрипти для роботи з базами даних MSSQL.

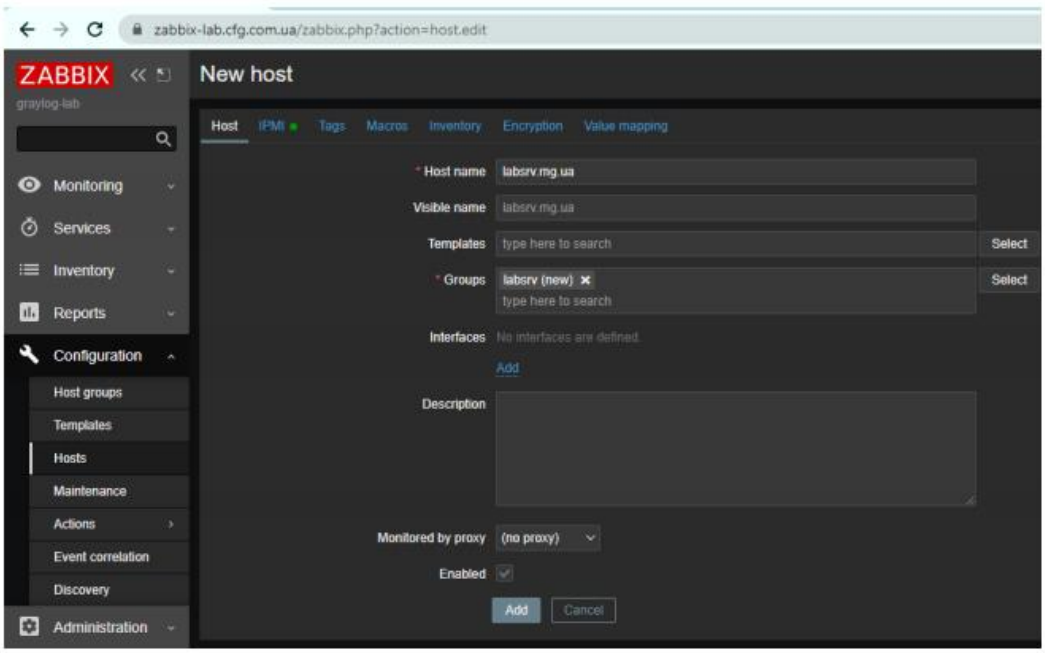

Рисунок 3.20 – Внесення інформації про назву сервера, до якого потрібно встановити підключення

Внесення облікових даних для підключення до хоста у Zabbix є необхідним кроком для забезпечення коректної взаємодії агента з сервером моніторингу. Правильне введення облікових даних гарантує успішне встановлення з’єднання, дозволяє збирати метрики та події в реальному часі, а також забезпечує безпечний та авторизований доступ до ресурсів сервера (рис. 3.21).

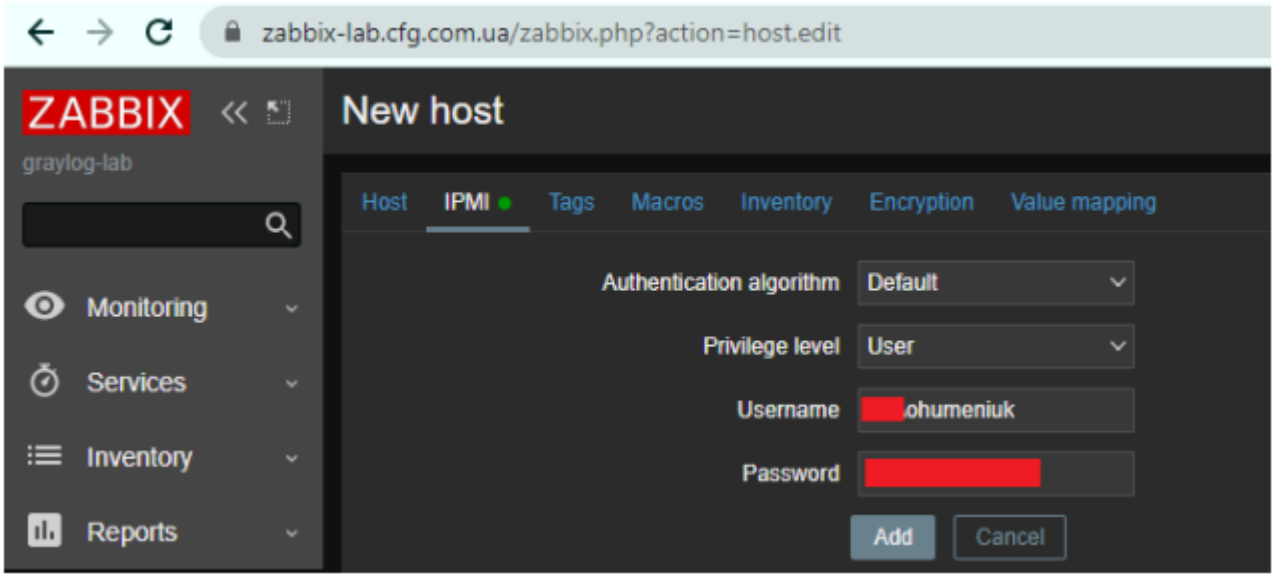

Рисунок 3.21 – Внесення облікових даних для підключення до хоста

На цьому етапі у веб-інтерфейсі вказуються параметри доступу, включаючи ім’я користувача та пароль або інші методи автентифікації, що дозволяють серверу отримувати дані про стан хоста.

Скачування агента Zabbix (рис. 3.22) для роботи на хості labsrv передбачає отримання відповідного пакета програмного забезпечення з офіційного сховища Zabbix, який відповідає операційній системі та архітектурі хоста. На цьому етапі обирається версія агента, сумісна з встановленою версією Zabbix-сервера, після чого здійснюється завантаження інсталяційного пакета на хост.

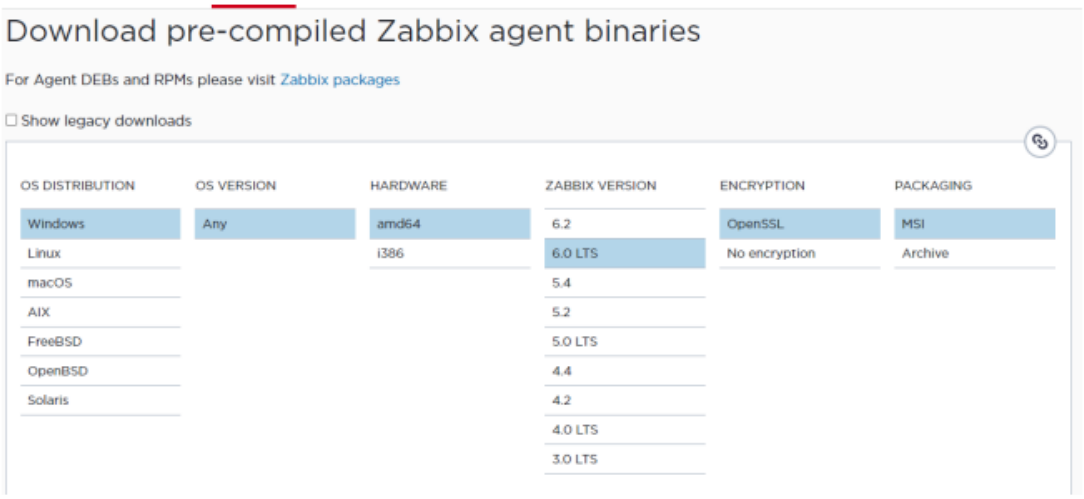

Рисунок 3.22 – Інсталяційний файл агента Zabbix для роботи на хості labsrv

Це забезпечує подальшу інсталяцію та конфігурацію агента для збору даних про стан системи та ресурсів, необхідних для моніторингу сервера labsrv.

Процес установки Zabbix-агента передбачає декілька послідовних кроків, спрямованих на розгортання клієнтського компонента системи моніторингу на цільовому хості. Спершу (рис. 3.23) завантажується пакет агента, сумісний із операційною системою та архітектурою сервера. Після цього виконується інсталяція пакета із застосуванням стандартних засобів керування програмним забезпеченням, після чого налаштовується конфігураційний файл агента, де вказуються параметри підключення до сервера Zabbix, такі як IP-адреса або доменне ім’я сервера, порт для з’єднання та інші необхідні опції.

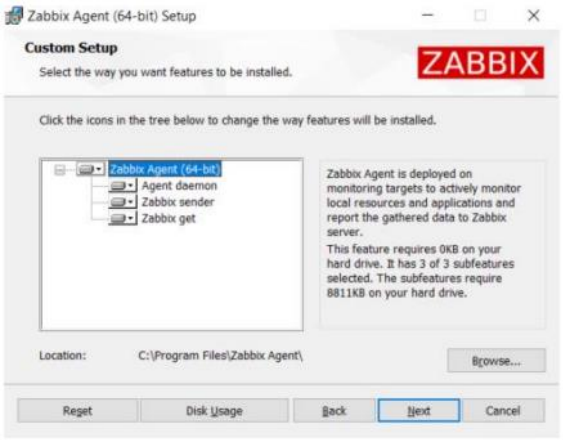

Рисунок 3.23 – Етапи інсталяції агента Zabbix на хості

Завершальним етапом є запуск служби агента та перевірка її працездатності для забезпечення стабільного збору даних про стан хоста та ресурсів у реальному часі (рис. 3.24)

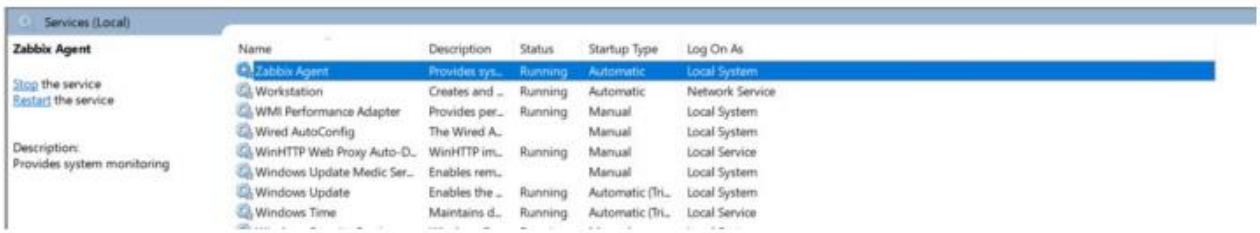

Рисунок 3.24 – Успішний запуск сервісу Zabbix-агенту

Те, що налаштовуване з’єднання встановлено, свідчить про успішну конфігурацію агента Zabbix та його готовність до взаємодії із сервером моніторингу (рис. 3.25).

Це означає, що сервер отримує авторизований доступ до хоста, а агент здатний передавати дані про стан системи та ресурси в реальному часі. Успішне встановлення з’єднання підтверджує коректність параметрів підключення, включаючи IP-адресу або доменне ім’я сервера, порт комунікації та облікові дані користувача, і гарантує надійну роботу механізмів збору та відображення метрик у веб-інтерфейсі Zabbix.

Рисунок 3.25 – Успішна конфігурація з’єднання агента з сервером

Запуск моніторингу дисків на сервері labsrv передбачає активацію збору даних про стан файлових систем, обсяг використаного та вільного дискового простору, а також інших критичних показників продуктивності сховища.

Цей процес здійснюється через налаштування відповідних елементів моніторингу в веб-інтерфейсі Zabbix або шляхом активації відповідних шаблонів для хоста.

У результаті сервер Zabbix починає регулярно опитувати агента на хості labsrv, отримуючи актуальні метрики дискової підсистеми, що дозволяє виявляти перевантаження, ризики заповнення дисків та забезпечує своєчасне реагування на потенційні проблеми (рис. 2.26).

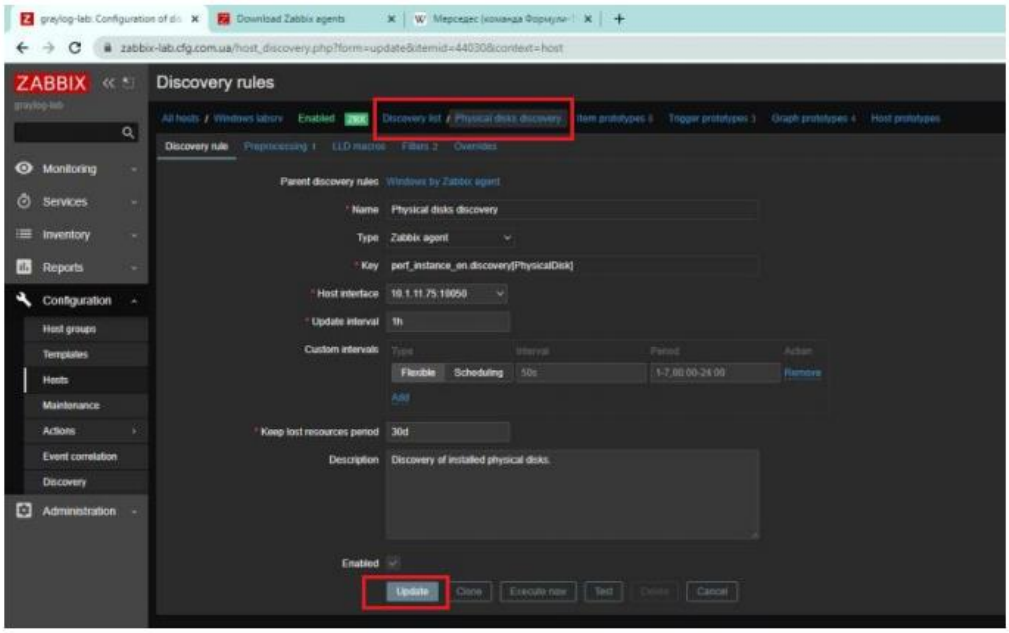

Рисунок 3.26 – Запуск моніторингу дисків на сервері labsrv

Створення нової панелі для відображення даних у Zabbix дозволяє організувати візуалізацію ключових метрик та показників моніторингу у зручному та інформативному вигляді.

Панель дає змогу групувати графіки, таблиці та інші віджети за категоріями, задавати часові інтервали відображення даних і налаштовувати формат їх візуалізації (рис. 3.27).

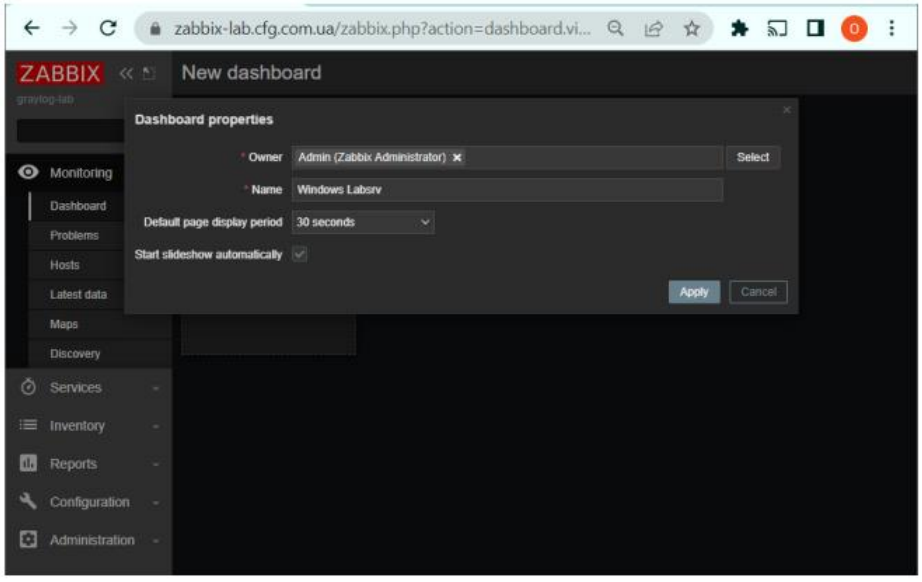

Рисунок 3.27 – Створення нової панелі для відображення даних

Створення вікна для відображення нового графіку у Zabbix дозволяє організувати на панелі моніторингу окремий простір для візуалізації конкретних метрик або показників хоста (рис. 3.28).

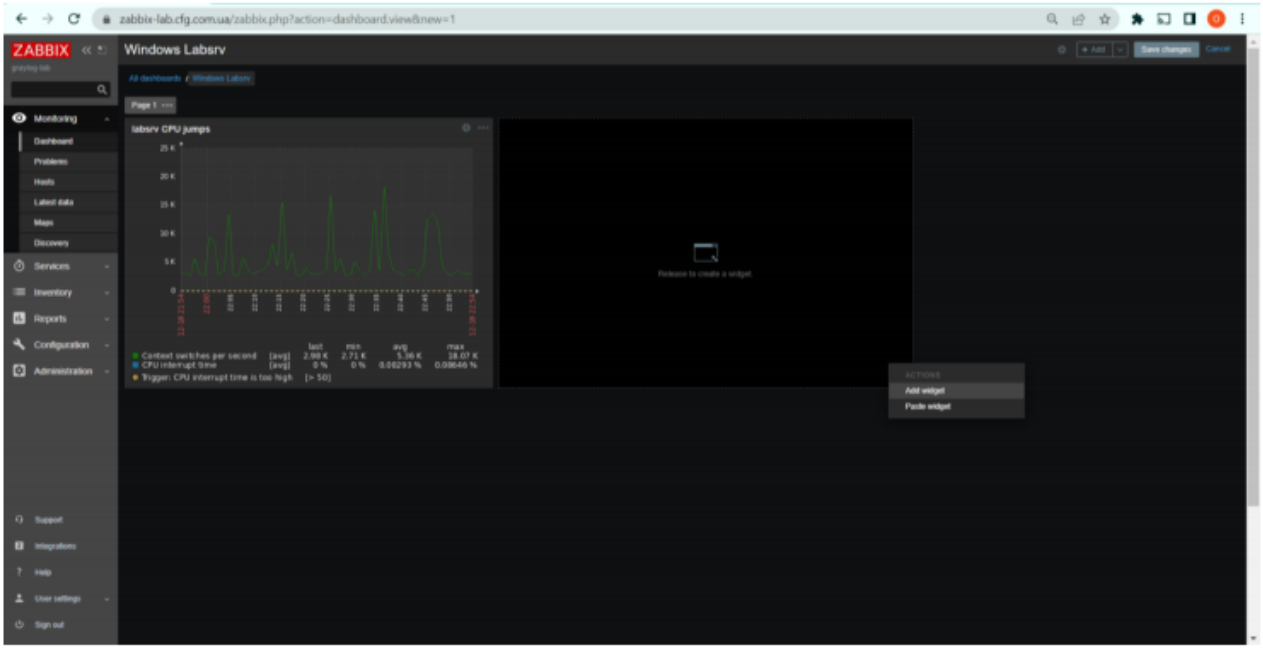

Рисунок 3.28 – Створення вікна для відображення нового графіку

У цьому вікні можна задавати тип графіку, обирати елементи для відображення, налаштовувати часові інтервали та параметри масштабу, а також визначати спосіб агрегування даних.

Такий підхід забезпечує гнучку та наочну презентацію інформації, дозволяє відстежувати зміни показників у реальному часі та оперативно реагувати на потенційні відхилення в роботі серверів та інших компонентів інфраструктури.

Створення графіку для відображення даних по використанню центрального процесора у Zabbix передбачає налаштування візуального представлення показників завантаження процесора на хості. Д

ля цього обираються відповідні елементи моніторингу, що відображають відсоток використання ЦП у реальному часі, задаються часові інтервали та параметри відображення (рис. 3.29).

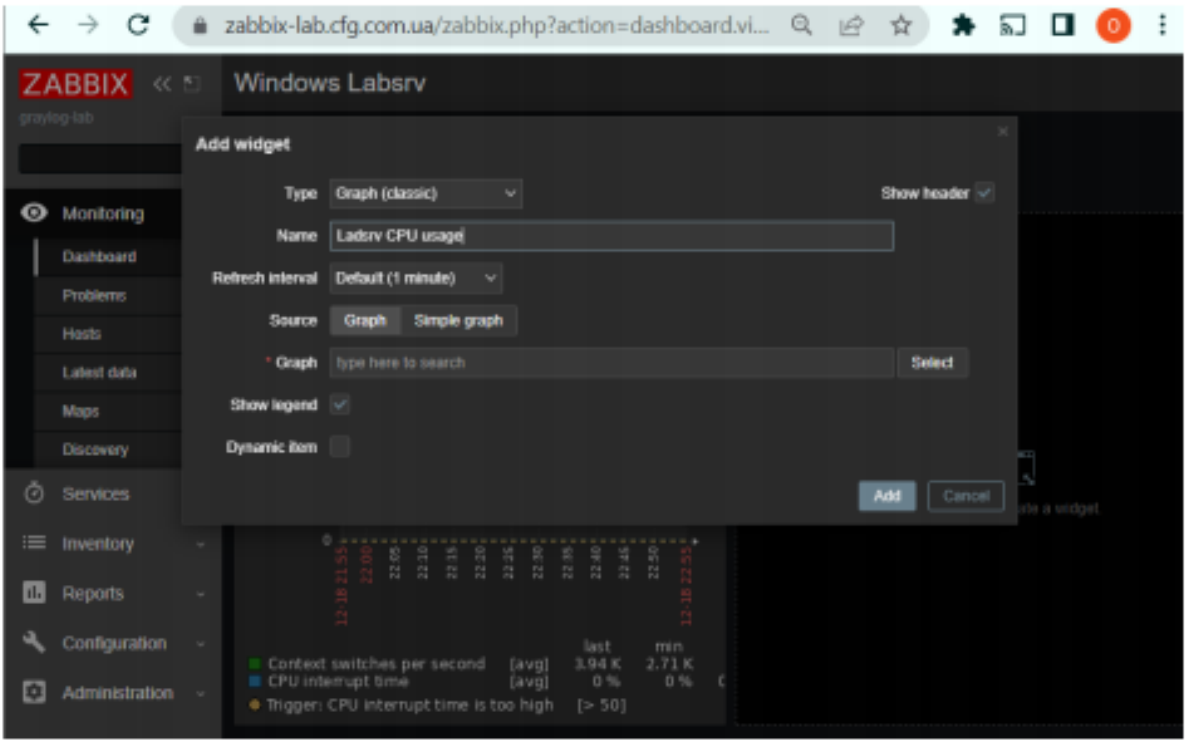

Рисунок 3.29 – Створення графіку ’ відображення даних по використанню ЦП

Графік дозволяє наочно оцінювати навантаження на процесор, виявляти пікові значення та тенденції у використанні ресурсів, що сприяє ефективному плануванню навантаження та своєчасному реагуванню на можливі проблеми в роботі серверів.

Вибір даних для відображення на графіку у Zabbix передбачає визначення конкретних елементів моніторингу, які будуть візуалізовані, та хоста, з якого ці дані мають надходити.

Це дозволяє зв’язати графік із конкретними метриками, наприклад завантаження процесора, використання пам’яті або дискового простору, та забезпечує коректну агрегацію та відображення інформації у реальному часі.

Вибір правильного хоста гарантує, що графік відображає актуальні дані саме з цільового сервера або пристрою, що підлягає моніторингу, і дозволяє проводити точний аналіз стану системи та виявляти потенційні проблеми (рис. 3.30).

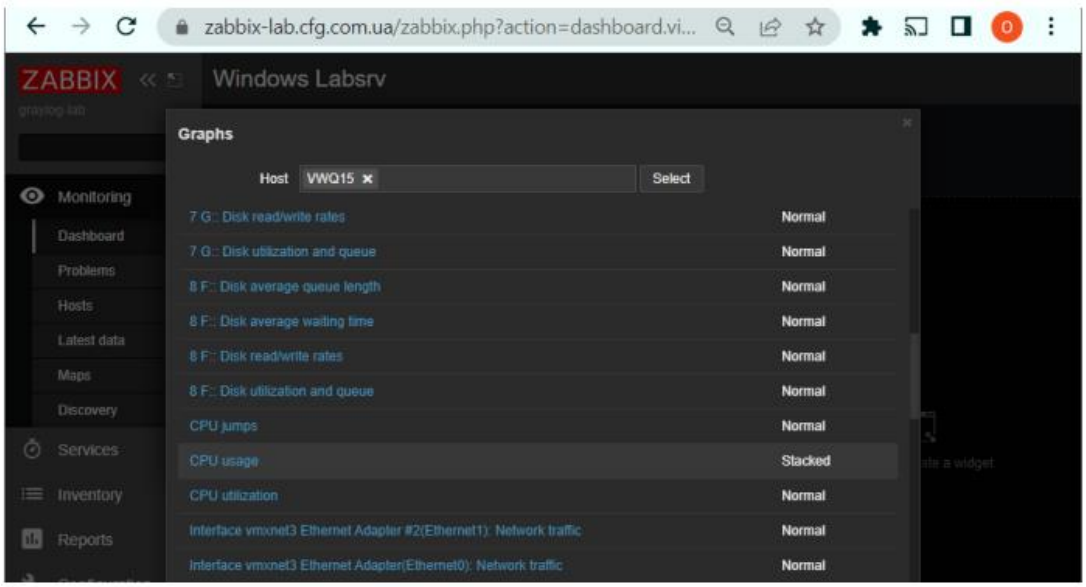

Рисунок 3.30 – Визначення хоста та параметрів для візуалізації даних

Відображення даних щодо стрибків навантаження та поточної завантаженості процесора на налаштованому хості дозволяє оцінити ефективність використання ресурсів та виявити аномальні або пікові значення завантаження (рис. 3.31).

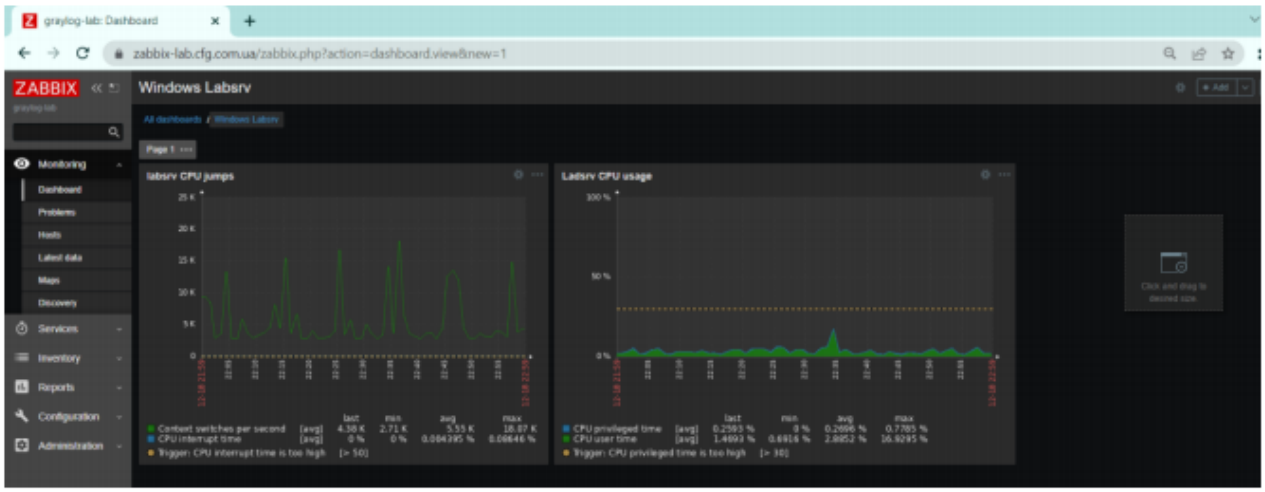

Рисунок 3.31 – Моніторинг стрибків навантаження та завантаженості процесора

За допомогою графіків у веб-інтерфейсі Zabbix користувач може візуалізувати динаміку змін завантаження ЦП у реальному часі, порівнювати різні часові інтервали та своєчасно реагувати на потенційні перевантаження

системи. Це забезпечує оперативний моніторинг продуктивності хоста та дозволяє планувати оптимальне розподілення ресурсів для забезпечення стабільної роботи серверів та служб.

У результаті виконаної роботи було здійснено комплексне встановлення та налаштування системи моніторингу Zabbix, що включало розгортання сервера, налаштування веб-інтерфейсу, створення користувача та бази даних, а також імпорт початкової структури бази.

3.7 Організація моніторингу продуктивності MSSQL на платформі Zabbix

На момент випуску Zabbix 5.0 система не мала вбудованих можливостей для налаштування моніторингу баз даних, що змушувало адміністраторів вручну конфігурувати агент для збору відповідної інформації. Цей процес був трудомістким, займав багато часу та часто призводив до помилок навіть при виконанні базового моніторингу сервера. З виходом п’ятої версії Zabbix з’явилася підтримка підключень через ODBC.

Як згадувалося в попередніх розділах, ODBC є системою для віддаленого підключення різних служб до баз даних, включаючи MSSQL. Це дозволяє встановити з’єднання між MSSQL і ODBC та збирати необхідну інформацію для моніторингу без потреби переналаштовувати агент Zabbix. Дані надсилаються через ODBC і обробляються на сервері Zabbix. При цьому адміністратор повинен мати змогу обирати потрібну інформацію, формувати запити та налаштовувати параметри збору даних, що буде розглянуто в рамках цієї роботи.

Першим кроком є правильне налаштування підключення ODBC до сервера MSSQL. ODBC необхідно встановити та сконфігурувати на сервері Zabbix за допомогою командного рядка. Для цього слід увійти в режим адміністратора в Ubuntu, виконавши команду sudo su та ввівши пароль користувача root. Після цього можна приступати до встановлення самого конектора, використовуючи команди для додавання ключів Microsoft та налаштування репозиторію пакетів.

Після інсталяції ODBC на сервері Ubuntu наступним кроком є оновлення системи та встановлених пакетів. Це забезпечує сумісність усіх компонентів та готовність сервера до подальшої конфігурації конектора.

Далі виконується налаштування ODBC-конектора, який дозволяє встановлювати зв'язок між сервером Zabbix і базою даних MSSQL. На цьому етапі здійснюється конфігурація параметрів драйвера, визначення джерел даних та підготовка до підключення.

Наступним кроком є налаштування конфігурації ODBC-конектора для підключення до бази даних з використанням статичного порту. Це забезпечує стабільне та передбачуване з'єднання з екземпляром MSSQL-сервера та дозволяє серверу Zabbix отримувати необхідні метрики для моніторингу.

Після завершення налаштувань система інформує про успішне встановлення підключення до екземпляра MSSQL-сервера за допомогою ODBC-конектора. Це підтверджує правильність введених параметрів, готовність підключення до збору даних і забезпечує подальшу роботу інструментів моніторингу на сервері Zabbix.

Надання доступів для технічного користувача є важливим етапом налаштування моніторингу баз даних MSSQL у Zabbix. Для забезпечення коректного збору даних створюється спеціальний обліковий запис користувача, який отримує мінімальні необхідні права (рис. 3.32).

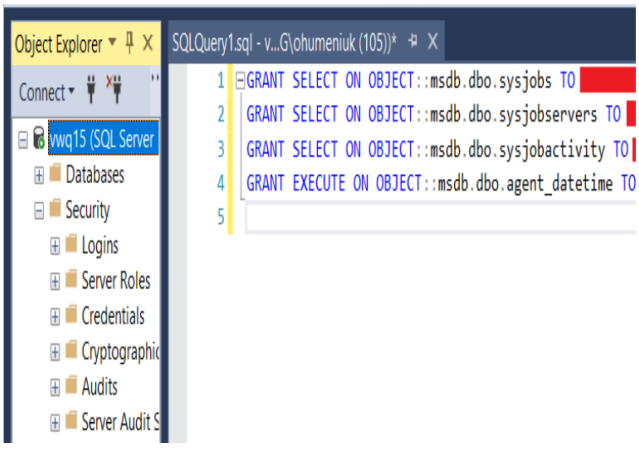

Рисунок 3.32 – Конфігурація прав доступу технічного користувача для читання таблиць та виконання збережених процедур

Імпорт шаблону у Zabbix є ключовим етапом налаштування моніторингу, який дозволяє швидко застосувати заздалегідь підготовлені конфігурації для збору даних із хостів або сервісів. Шаблон містить набір елементів моніторингу, тригерів, графіків та панелей, що полегшує стандартизацію моніторингу та забезпечує повторне використання налаштувань для кількох хостів.

Процес імпорту передбачає завантаження XML-файлу шаблону через веб-інтерфейс Zabbix, перевірку сумісності з версією сервера та застосування всіх включених конфігураційних елементів до системи. Після імпорту шаблон можна прив’язати до конкретних хостів, що забезпечує автоматичне налаштування збору метрик та візуалізації даних без необхідності вручну створювати елементи моніторингу. Це значно спрощує процес впровадження моніторингу нових серверів або сервісів та зменшує ймовірність помилок при конфігурації (рис. 3.33).

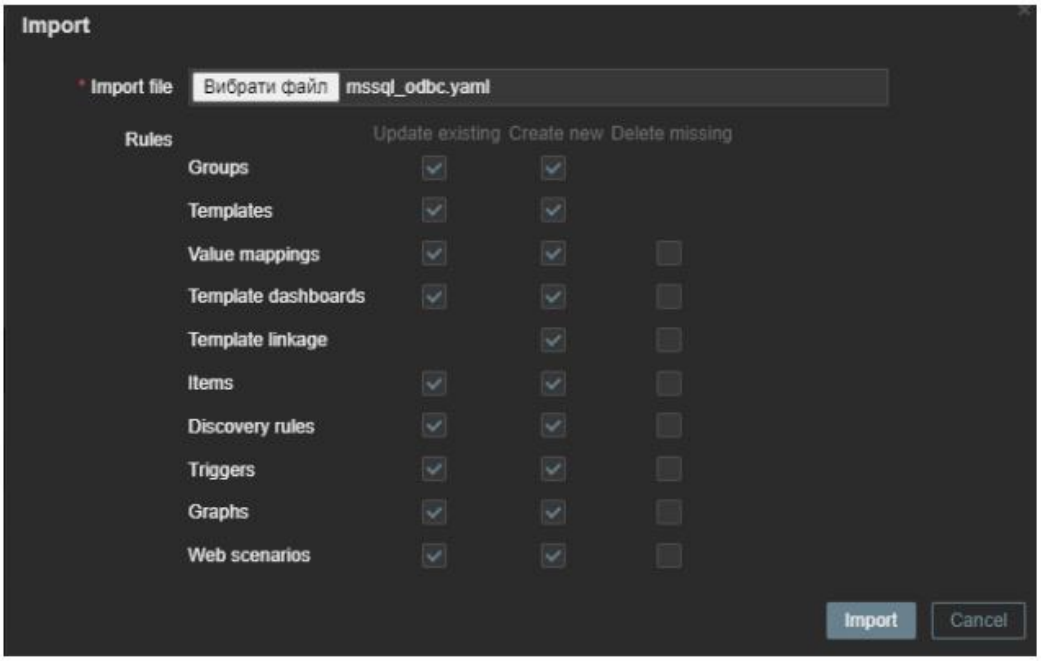

Рисунок 3.33 – Вікно імпорту шаблону

Перелік динамічних макросів у Zabbix представляє собою набір змінних, які використовуються для параметризації шаблонів та елементів моніторингу, дозволяючи адаптувати їх під конкретні хости або сервіси. Кожен макрос може

містити унікальне значення, наприклад адресу сервера, порт, ім’я бази даних або обліковий запис користувача, що забезпечує коректне підключення до моніторингових ресурсів.

Внесення конкретних значень у динамічні макроси дозволяє налаштувати шаблон таким чином, щоб одна і та ж конфігурація могла використовуватися для різних хостів без необхідності створювати окремі елементи моніторингу для кожного з них. Це забезпечує гнучкість, централізоване управління параметрами та зменшує ймовірність помилок при масштабуванні системи моніторингу (рис. 3.34)..

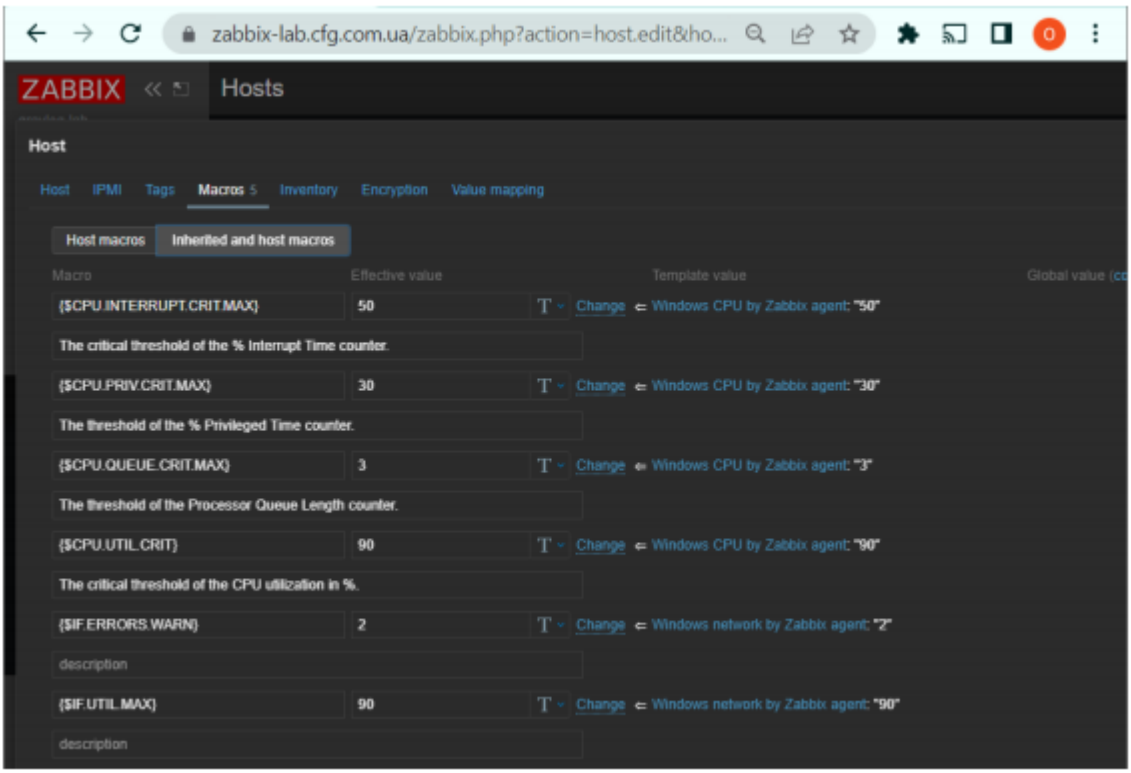

Рисунок 3.34 – Визначення значень динамічних макросів для адаптації шаблонів

Побудова графіків для моніторингу роботи MSSQL-сервера labsrv у Zabbix дозволяє візуалізувати ключові показники продуктивності бази даних, такі як завантаження процесора, використання пам’яті, кількість активних з’єднань, виконання запитів та інші критично важливі метрики (рис. 3.35).

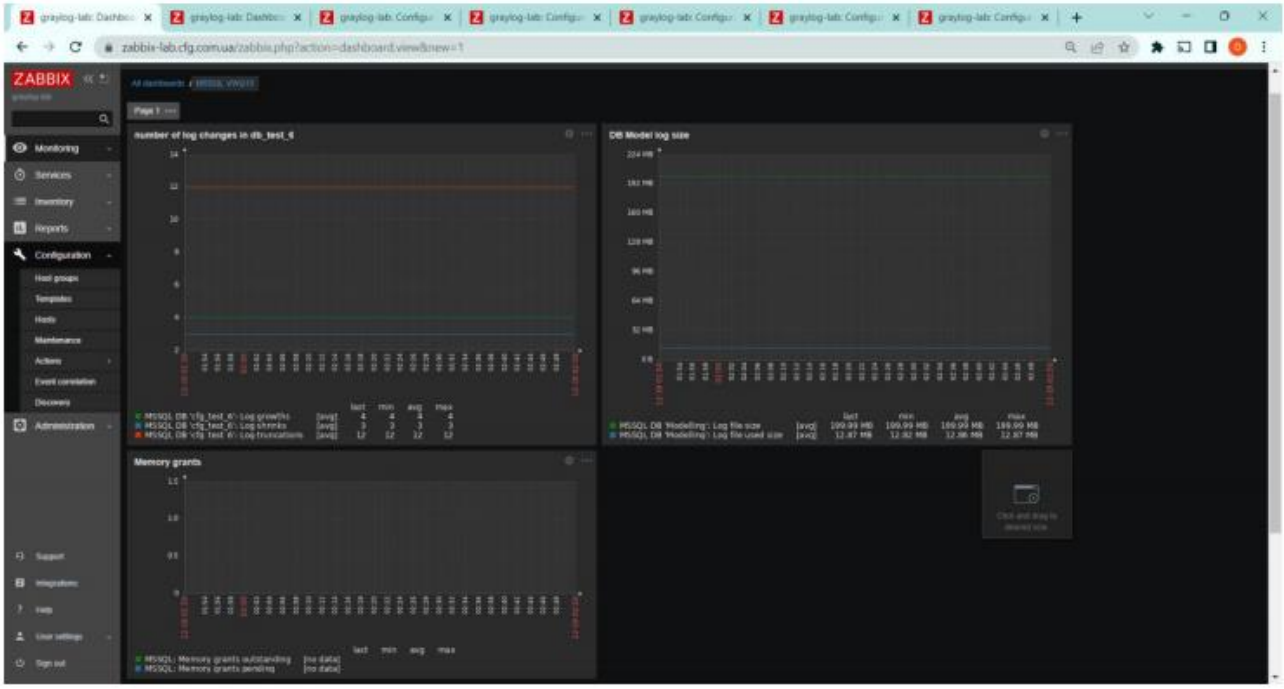

Рисунок 3.35 – Візуалізація ключових метрик роботи MSSQL-сервера labsrv

Для цього обираються відповідні елементи моніторингу, які збираються через ODBC-конектор, і створюються графіки у веб-інтерфейсі Zabbix.

Кожен графік можна налаштовувати за типом відображення, часовими інтервалами та параметрами агрегації даних, що забезпечує наочне відстеження змін у роботі сервера. Побудовані графіки дозволяють оперативно виявляти пікові навантаження, аналізувати тенденції використання ресурсів та своєчасно реагувати на потенційні проблеми у роботі MSSQL-сервера labsrv, забезпечуючи стабільність і ефективність роботи системи.

Отримані результати підтверджують працездатність та ефективність налаштованої системи моніторингу, забезпечують централізоване управління параметрами збору даних і надають зручні інструменти для аналізу стану баз даних у реальному часі. Запропоновані методи можуть бути використані для масштабування системи та інтеграції додаткових хостів і сервісів.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проаналізовано сучасні підходи до моніторингу серверної інфраструктури та вивчено функціональні можливості поширених систем моніторингу серверів. У результаті аналізу виявлено, що найбільш ефективними є комплексні рішення, які забезпечують централізований збір даних, підтримку агентного та безагентного моніторингу, масштабованість, гнучку систему оповіщень і можливість інтеграції з різними службами та базами даних. Система Zabbix була визначена як оптимальна платформа для реалізації поставлених завдань завдяки її функціональній повноті та відкритій архітектурі.

У процесі проектування та впровадження системи моніторингу віртуальних машин було розроблено архітектуру взаємодії між сервером Zabbix, агентами та веб-інтерфейсом. Реалізовано інсталяцію та налаштування Zabbix-сервера, веб-сервера та агентів на досліджуваних віртуальних серверах. У результаті впровадження забезпечено безперервний контроль ключових показників продуктивності, зокрема використання центрального процесора, оперативної пам’яті, дискових ресурсів і мережевої активності.

Окрему увагу було приділено проектуванню моніторингу служби MSSQL, розгорнутої на віртуальних серверах. У ході роботи налаштовано інтеграцію Zabbix з MSSQL за допомогою ODBC-конектора, що дозволило здійснювати збір статистичних даних без модифікації конфігурації Zabbix-агента. У результаті реалізовано моніторинг стану баз даних, продуктивності запитів та використання ресурсів СУБД, а також побудовано відповідні графіки для наочного аналізу отриманих даних.

З метою оцінки ефективності впровадженої системи моніторингу було проведено експериментальні дослідження шляхом штучного створення навантаження на віртуальні сервери та службу MSSQL. За результатами експериментів виявлено, що система Zabbix коректно фіксує зміни показників продуктивності в режимі реального часу, відображає пікові навантаження та

забезпечує своєчасне інформування адміністратора про відхилення від нормальних режимів роботи.

Таким чином, поставлені завдання кваліфікаційної роботи були повністю виконані. Отримані результати підтверджують доцільність використання системи Zabbix для моніторингу віртуальних серверів і служб баз даних у сучасних IT-інфраструктурах та можуть бути використані для подальшого розвитку і масштабування систем адміністрування серверних ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ryu A. J., Magnuson D. R., Kingsley T. C. Why Mayo Clinic Is Embracing the Cloud and What This Means for Clinicians and Researchers. *Mayo Clinic Proceedings: Innovations, Quality & Outcomes*. 2021. Vol. 5, no. 6. P. 969–973.

2. Білоусова Л. І., Житєньова Н. В. Хмарні сервіси як ефективний інструмент візуалізації. *New computer technology*. 2019. Т. 17. С. 25-30.

3. «ІНТЕРНЕТ РЕЧЕЙ» – ЯК ПЕРСПЕКТИВА РОЗВИТКУ МОБІЛЬНИХ СИСТЕМ / Г. СОКОЛ та ін. *ITSynergy*. 2021. № 1. С. 49–57.

4. Krupat E., Nagios H., Register A. Evaluative Biases of Pharmacy and Nonpharmacy Students. *American Journal of Pharmaceutical Education*. 1986. Vol. 50, no. 1. P. 48–51.

5. Pradana A., Widiasari I. R., Efendi R. Implementasi Sistem Monitoring Jaringan Menggunakan Zabbix Berbasis SNMP. *AITI*. 2022. Vol. 19, no. 2. P. 248–262.

6. Oktiawati U., Hartono A. Pemantauan Router CPE pada Jaringan Metro Ethernet Menggunakan Zabbix Berbasis Raspberry Pi. *Journal of Internet and Software Engineering*. 2021. Vol. 2, no. 1. P. 29–38.

7. Download and install Zabbix. URL: https://www.zabbix.com/download (дата звернення 19.11.2022).

8. Setting up database monitoring. URL: https://subscription.packtpub.com/book/cloudandnetworking/9781800202238/2/ch02lvl1sec18/setting-up-database-monitoring (дата звернення 20.11.2022).

9. How Virtualization Changed IT Roles. URL: https://www.ecpi.edu/blog/what-does-virtual-server-administrator-do (дата звернення 14.11.2022).

10. Virtual Server Management. URL: https://www.manageengine.com/network-monitoring/virtual-servermanagement.html (дата звернення 14.11.2022).

11. What Is Server Management for Physical and Virtual Servers. URL: https://www.parkplacetechnologies.com/blog/what-is-server-management physicalvirtual-servers/ (дата звернення 16.11.2022).

12. What is Zabbix and How it works? An Overview and Its Use Cases. URL: https://www.devopsschool.com/blog/what-is-zabbix-and-how-it-works-an-overviewand-its-use-cases/ (дата звернення 21.11.2022).

13. Робота системи моніторингу Zabbix з API. URL: https://www.zabbix.com/documentation/current/en/manual/api (дата звернення 29.11.2022).