МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА СИСТЕМ ІНФОРМАЦІЙНОГО ТА КІБЕРНЕТИЧНОГО ЗАХИСТУ

«На правах рукопису»
УДК 004.11.06

«До захисту допущено»
Завідуючий кафедрою СІКЗ
_____ к.т.н. Г.В. Шуклін
«____» _____ 2022 р.

БАКАЛАВРСЬКА АТЕСТАЦІЙНА РОБОТА

зі спеціальності 125 “Кібербезпека”

на тему: «АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ МОНІТОРИНГУ ІТ-ІНФРАСТРУКТУРИ ПІДПРИЄМСТВ»

студент групи СЗД-41 Хайбуллін Владислав Олександрович _____
(підпис)

Науковий керівник: к.т.н., доцент Пепа Юрій Володимирович _____
(підпис)

Нормоконтроль: Гребенніков Асаді Болдхоядович _____
(підпис)

КИЇВ – 2022

«ЗАТВЕРДЖУЮ»

Завідувач кафедри СІКЗ

_____ к.т.н. Г.В. Шуклін

(підпис)

«_____»_____ 2022 р.

ЗАВДАННЯ

на атестаційну роботу магістра

студенту: Хайбулліну Владиславу Олександровичу

1. Тема роботи: «Аналіз програмних засобів моніторингу іт-інфраструктури підприємств», затверджена наказом по університету від « ___ » _____ 2022 р. за № _____ .

2. Термін здачі студентом оформленої роботи « 2 » червня 2022 р.

3. Об'єкт дослідження: властивості інформації, які змінюються під впливом зовнішнього втручання.

4. Предмет дослідження: ризик та методи його оцінювання щодо безпеки даних.

5. Мета роботи: розробка методику функціонального оцінювання кібератак і протидії та оцінити економічну ефективність системи захисту.

6. Перелік питань, які мають бути розроблені:

1. Аналіз властивостей інформації, яка підлягає захисту на підприємстві.
2. Провести функціональне оцінювання ефективності програмних засобів захисту.
3. Запропонувати програмне забезпечення з урахуванням особливостей інформації, що підлягає захисту.

7. Перелік публікацій:

8. Перелік ілюстративного матеріалу: Презентація виконана на слайдах для подання за допомогою світлопроектору та комп'ютерних засобів.

9. Дата видачі завдання « 16 » лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

Дата видачі завдання «16» лютого 2022 р.

№ з/п	Назва етапів атестаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури	20.03.22 р.	виконано
2	Написання першого розділу роботи	05.04.22 р.	виконано
3	Написання другого розділу роботи	30.04.22 р.	виконано
4	Написання третього розділу роботи	12.05.22 р.	виконано
5	Оформлення атестаційної роботи	14.05.22 р.	виконано
6	Підготовка демонстраційних матеріалів	20.05.22 р.	виконано

Студент: СЗД-41 Хайбуллін В.О. _____
(підпис)

Науковий керівник: к.т.н., доц. Пепа Ю.В. _____
(підпис)

Нормоконтроль: Гребенніков А.Б. _____
(підпис)

РЕФЕРАТ

Атестаційна робота містить: 52 сторінки, 43 рисунки, 2 таблиці та 20 джерел.

Важливість роботи полягає в дослідженні питань економічних ризиків втрати конфіденційної інформації на підприємстві від різних чинників, а також всебічному аналізі функціональних методів критеріальної оцінки вторгнень і протидії кібератакам. На основі цих досліджень проведено якісний та кількісний аналіз економічної ефективності вкладання коштів у побудову системи захисту інформації на виділеному об’єкті.

Метою роботи є розробка методику функціонального оцінювання кібератак і протидії та оцінити економічну ефективність системи захисту.

Завдання роботи:

1. Проаналізувати існуючі властивості інформації, яка підлягає захисту на підприємстві.

2. Провести функціональне оцінювання ризиків втрати інформації та ефективної протидії.

3. Оцінити ефективність програмних засобів і ризиків втрат інформації.

Об’єктом дослідження є властивості інформації, які змінюються під впливом зовнішнього втручання.

Предметом дослідження є ризик та методи його оцінювання щодо безпеки даних.

Методи дослідження – аналіз, експертне оцінювання, систематизація, ймовірнісний підхід.

Галузь використання – кібербезпека та інформаційна безпека.

Ключові слова: ВТРАТИ, АТАКА, ПРОТИДІЯ, КІБЕРБЕЗПЕКА, ЗАХИСТ ІНФОРМАЦІЇ, КРИТЕРІЙ, РИЗИК, ЕФЕКТИВНІСТЬ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API	–	Advanced Mobile Phone Service
CLI	–	Command Line Interface
FTP	–	File Transfer Protocol
GSM	–	Global System for Mobile Communications
HTTP	–	HyperText Transfer Protocol
IETF	–	Internet Engineering Task Force
IP	–	Internet Protocol
IPMI	–	Intelligent Platform Management Interface
JMX	–	Java Management Extensions
LAN	–	Local Area Network
MIB	–	Management Information Base
NTP	–	Network Time Protocol
ODBC	–	Open DataBase Connectivity
OID	–	Object identifier
OSI	–	The Open Systems Interconnection model
POP	–	Post Office Protocol
RRD	–	Round-robin Database
SCCM	–	System Center Configuration Manager
SCE	–	System Center Essentials
SCOM	–	System Center Operations Manager
SCSM	–	System Center Service Manager
SCVMM	–	System Center Virtual Machine Manager
SLA	–	Service Level Agreement
SMS	–	Short Message Service
SMTP	–	Simple Mail Transfer Protocol
SNMP	–	Simple Network Management Protocol
SSH	–	Secure Shell

TCP	–	Transmission Control Protocol
UDP	–	User Datagram Protocol
USB	–	Universal Serial Bus
WAN	–	Wide Area Network
ІБУ	–	Інформаційні бази управління
ІТ	–	Інформаційні технології
ІТС	–	Інтелектуальна транспортна система
МБ	–	Мегабайт
ОС	–	Операційна система
СУБД	–	Система управління базами даних
ЦП	–	Центральний процесор

ВСТУП

Актуальність теми. Експлуатація, обслуговування та подальший розвиток IT-інфраструктури є дуже важливими завданнями для забезпечення безперебійного робочого процесу. Ефективність різного роду організацій дуже часто безпосередньо залежить від використовуваних інструментів та IT-інфраструктури, як з точки зору пропонованих послуг, так і з точки зору рішень, що використовуються працівниками організації. Навіть при невеликих IT-рішеннях потрібен певний обсяг роботи, щоб забезпечити доступність систем та послуг. Тому існує необхідність постійного контролю стану показників з використанням засобів моніторингу в IT-інфраструктурі.

Відповідно до поставленої задачі в кваліфікаційній роботі ставилися і вирішувалися наступні взаємозалежні завдання:

1. Вивчення, аналіз та порівняльна характеристика
сучасних програмних засобів моніторингу IT-інфраструктури.
2. Аналіз існуючих факторів та вимоги до апаратних засобів.
3. Планування розгортання системи моніторингу IT-інфраструктури.
4. Розробка алгоритму розгортання системи моніторингу IT- інфраструктури.

1 МОНІТОРИНГ ІТ-ІНФРАСТРУКТУРИ

1.1. Призначення та завдання систем моніторингу ІТ- інфраструктури

Система моніторингу – група пристроїв та програмне забезпечення, що забезпечує систематичний збір і обробку інформації, яка може бути використана для поліпшення процесу прийняття рішення, а також, побічно, для інформування громадськості або прямо як інструмент зворотного зв'язку з метою здійснення проектів, оцінки програм або вироблення політики.

Система моніторингу ІТ-інфраструктури призначена для моніторингу стану, аналізу працездатності та управління апаратними і програмними засобами функціональних і технологічних підсистем ІТС, мережами, підмережами, технологіями, обладнанням провайдерів інформаційно-комунікаційних послуг.

Основними завданнями систем моніторингу є надання актуальної інформації для аналізу стану ІТ-інфраструктури та швидкого виявлення виниклих проблем і їх оперативного усунення. Системи моніторингу дозволяють кардинально зменшити витрати на обслуговування ІТ, автоматично виконувати рутинні дії ІТ-фахівців, ІТ-фахівцям вчасно помітити зниження продуктивності і визначити “слабкі місця” в ІТ-інфраструктурі. Постійний моніторинг допомагає підтримувати всі ІТ-сервіси в робочому стані, а також уникнути простоїв в роботі чим зберігає необхідний рівень їх якості.

Інструменти мережевого моніторингу надають широкий спектр сканування та аналізу для різних типів пристроїв та послуг. Цього вдалося досягти, використовуючи різні типи протоколів, що працюють на різних рівнях OSI.

1.2 Функції та задачі системного адміністратора організації

Особа, відповідальна за моніторинг, повинна визначити, чи всі компоненти перебувають у межах необхідних параметрів і чи не перевищені

порогові значення. Таким чином, моніторинг включає реєстрацію та системний запис процесів або операцій в IT-інфраструктурі. Спеціальне програмне забезпечення для моніторингу та системи спостереження служить як засіб контролю. Якщо IT-процеси не будуть працювати безперебійно, системний адміністратор може втрутитися в процес, якщо це необхідно, і виправити всі помилки. Для того, щоб мати змогу правильно аналізувати результати, слід регулярно проводити моніторинг.

Раніше роль моніторингу здійснювали адміністратори, а інформація про стан систем у кращому випадку збиралася ними ж у будь-яких неспеціалізованих програмах, у гіршому ж взагалі ніяк не накопичувалась і не оброблювалась. Всі відомості про систему були прив'язані до практичного досвіду роботи з інфраструктурою у конкретного фахівця і повністю губилися після того як він залишив робоче місце.

На початковому етапі системному адміністратору необхідно організувати моніторинг IT-інфраструктури на рівні обладнання, сервісів і додатків (рис. 1.1).

IT-моніторинг гарантує виявлення проблем на ранніх стадіях і таким чином, заощаджує величезні витрати. Загалом, це досягається за допомогою історичного моніторингу та моніторингу у реальному часі. Різниця між двома методами полягає в періоді спостереження, який використовується у відповідному моніторингу.

"Історичний" моніторинг – процес у якому адміністратору потрібна активна робота, тобто перспективні та планувальні дії. Довгострокова статистика автоматично генерується, щоб забезпечити можливість планування потужності на додаток до фактичного завдання моніторингу. Це також може підтримувати планування бюджету організації, серед іншого. Крім того, систематичні проблеми можна виявити на ранніх стадіях.

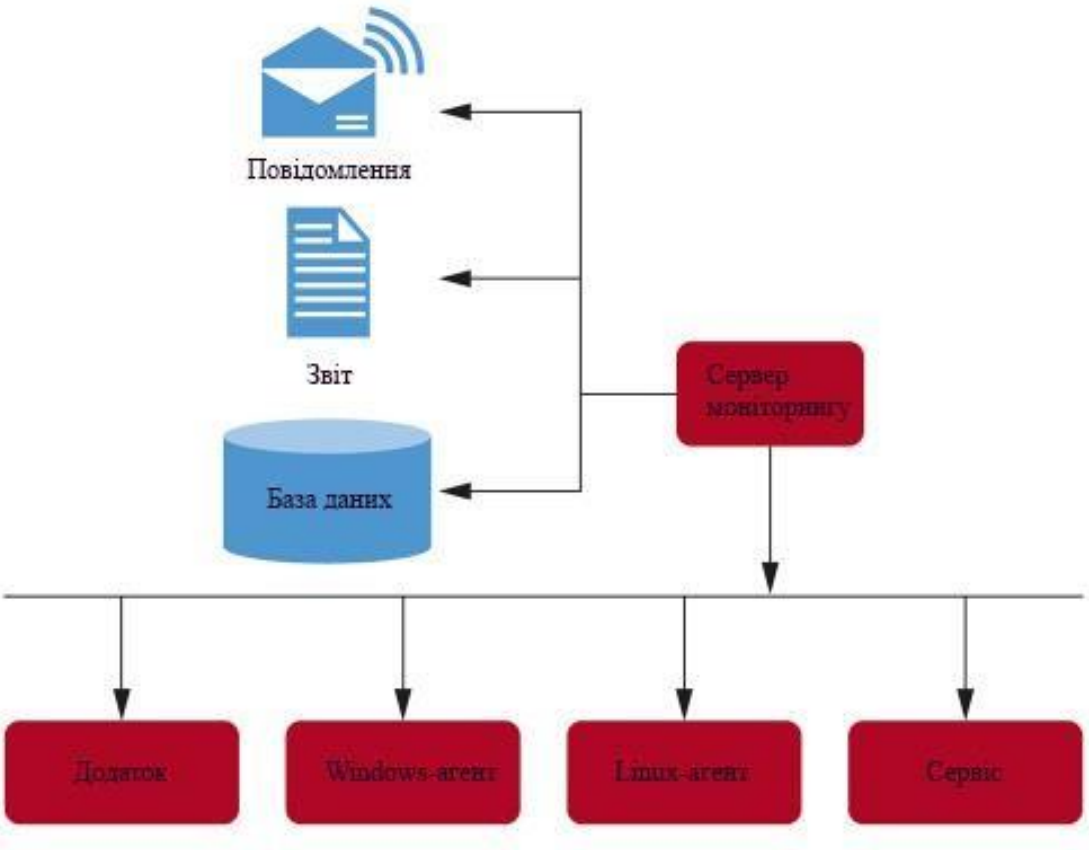

Рисунок 1.1 – Архітектура системи моніторингу IT-інфраструктури

Моніторинг у реальному часі – процес у якому адміністратор повинен реагувати на вже виниклі проблеми. Постійний моніторинг служб щодо функціональності гарантує виявлення проблем та інформування можливих потерпілих користувачів. Всі помилки реєструються та виправляються до того, як користувач навіть помітить їх або обмежиться їм.

1.3 Методи отримання даних системами моніторингу IT- інфраструктури

Поступово з’являється безліч напів- і повністю автоматизованих систем моніторингу, завданням яких є аналіз стану системи та збір інформації в колекції, які згодом можна переглянути при необхідності.

Існують спеціальні програми для моніторингу, які в задані проміжки часу циклічно емулюють дії користувача – запускають спеціальний “скрипт-сценарій”, а потім надсилають повідомлення про успіх виконання дій або про

виниклі в процесі помилки. Такий вид моніторингу називається моніторингом від імені кінцевого користувача.

Великий обсяг інформації, який збирається у процесі моніторингу потрібно десь зберігати, тому зазвичай використовують конфігураційну базу даних об’єкт моніторингу якої представлений як набір конфігураційних одиниць. Всі мережеві пристрої, всі сервери – це одиниці, які зберігаються в централізованій базі даних. За допомогою такого представлення є можливість інтегрувати систему моніторингу для візуального представлення у вигляді діаграм або графіків.

Майже всі системи моніторингу володіють стандартним набором виконання сервісних дій. Наприклад, є можливість очистити кошик якщо він заповнився або активувати архівування для визначених файлів, коли майже весь дисковий простір зайнятий.

Моніторинг значно видозмінюється з плином часу. Якщо раніше була необхідність відстежувати стан тільки фізичних серверів, то тепер на кожному з них може бути ще кілька віртуальних.

При розробці, впровадженні систем моніторингу спочатку необхідно визначити об’єкти за якими планується нагляду, а також критичні події та показники, які будуть визначати кількість повідомлень при поломці, частоту сканування та інші наслідки і параметри. Для великих інфраструктур, на кшталт дата-центру, перед фінальним впровадженням зазвичай розгортають тестовий майданчик, на якому можна оцінити доцільність зроблених рішень і ухвалених порогових значень для параметрів.

Впровадження тестових майданчиків є особливо важливим рішенням при використанні сервісного підходу до діяльності IT-підрозділів, коли всі процеси переглядаються з точки зору, що надається підрозділом IT-сервісів. Для кожного сервісу корпоративної системи по можливості задається певний рівень якості його надання. Який описується в системі моніторингу як набір взаємопов’язаних компонентів IT-інфраструктури. В результаті формується угода про рівень якості сервісів (SLA). Згідно якої система здійснює зберігання

і збір інформації про якість надання ІТ-сервісів. На базі накопиченої інформації формуються звіти за певний період часу. Аналіз звітної інформації допомагає здійснювати:

- модернізацію ІТ-інфраструктури;
- реорганізацію діяльності ІТ-підрозділу;
- перегляд рівня надання ІТ-сервісу.

1.4 Порівняльна характеристика сучасних програмних засобів моніторингу ІТ-інфраструктури

Сучасні програмні засоби моніторингу орієнтовані на споживачів різного рівня. Зазвичай величезну кількість різноманітних функцій використовують лише у великій ІТ-інфраструктурі, а для маленьких буває достатньо відправлення оповіщення та спільного аналізу. Серед основних функцій моніторингу можна виділити наступні: спостереження, зберігання інформації, побудова звітів, візуалізація, пошук слабких місць та автоматизація сценаріїв.

Спостереження – основна функція, яка включає в себе періодичний збір показників з вузлів устаткування, сервісів.

Зберігання інформації – ця функція є доповнення до функції спостереження. Здійснюється збір інформації за основними показниками кожного об’єкта моніторингу, для зберігання зазвичай використовуються бази даних.

Побудова звітів – здійснюється як на основі поточних даних стеження, так і по довго тривало збережених даних. Наприклад, навантаження на сервер може попередити довготривалий моніторинг, якщо споживані ресурси весь час збільшуються, значить необхідно збільшити кількість засобів або перенести частину завдань на інший сервер, вибір якого теж можна здійснити на основі довготривалого звіту.

Візуалізація – звіти в візуальному представленні: у вигляді графіків та діаграм. Допомагають легко сприймати інформації, а також можливий вибір для візуалізації декількох найважливіших індикаторів, тоді як в звітах будуть

представлені всі показник.

Пошук слабких місць – на основі аналітичних даних моніторингу можливо дізнатися, яке місце інфраструктури найбільш сильно знижує загальні показники продуктивності.

Автоматизація сценаріїв – функція, яка звільняє адміністраторів від рутинних завдань.

Завдяки наявності коштів для реалізації вище вказаних функцій адміністратору більше не потрібно перевіряти вручну стан кожної складової системи, проблеми вирішуються і поломки усуваються більш оперативно, діагностика здійснюється багатовимірно і точно, а також можна планувати розширення інфраструктури.

Використання систем моніторингу та управління дозволяє:

- оптимізувати використання інформаційних ресурсів;
- підвищити якість IT-сервісів і швидкість усунення збоїв в роботі обладнання та програмного забезпечення, мінімізувати час простою сервісів;
- забезпечити надійність, безпеку і узгоджене функціонування всіх компонентів IT-інфраструктури;
- полегшити модернізацію IT-інфраструктури;
- в кілька разів підвищити ефективність роботи IT-підрозділу.

Для порівняльної характеристики розглянемо наступні найбільш широко використовувані сучасні програмні засоби моніторингу: Microsoft SCOM, Zabbix, Cacti, Nagios.

1.4.1 Microsoft SCOM

System Center Operations Manager – система наскрізного моніторингу від Microsoft, у тому числі активного спостереження за станом мереж (спостереження за будь-якими мережевими пристроями, що підтримують SNMP, аж до рівня портів, а також виявлення віртуальних локальних мереж і комутаторів в таких мережах). В останніх версіях з’явилася можливість стеження не тільки за системами, під управлінням операційних систем

сімейства Windows, але і за гетерогенними середовищами, що включають UNIX і Linux. SCOM призначений головним чином для організацій з числом машин більше 500 і числом серверів більше 30. Для менших організацій існує продукт SCE, що включає в себе частину функціоналу продуктів SCOM і SCCM, але призначений для малих і середніх підприємств.

Microsoft також надає можливість інтеграції продукту з SCSM, завдяки чому з'являється можливість автоматичного створення інцидентів на основі повідомлень SCOM.

Що стосується пильного спостереження за віртуальними середовищами, є засоби для інтеграції з пакетом SCVMM, який буде передавати SCOM інформацію про віртуальні машини, служби, приватні хмари і вузли.

Основні переваги:

- виняткова продуктивність і працездатність додатків для програмних середовищ Microsoft;
- забезпечує наскрізне управління службами для сервісів вашого центру обробки даних;
- сприяє поліпшенню ефективності і управління середовищами центрів обробки даних;
- уніфікований контроль в рамках приватних і загальнодоступних хмарних сервісів;
- підтримка Windows PowerShell 2.0.

Отже, SCOM – це моніторинг високої доступності з спрощеною інфраструктурою, що використовується в організаціях з великим парком машин під управлінням різних сімейств операційних систем, що включає безліч різноманітних засобів стеження, в тому числі за мережевим обладнанням, а також розширеними засобами представлення зібраної інформації.

Однак у даної системи є ряд недоліків з точки зору вирішення конкретної технічної задачі:

- система моніторингу охоплює безліч загальних показників системи, але непридатна для стеження за специфічними параметрами (рис. 1.2);

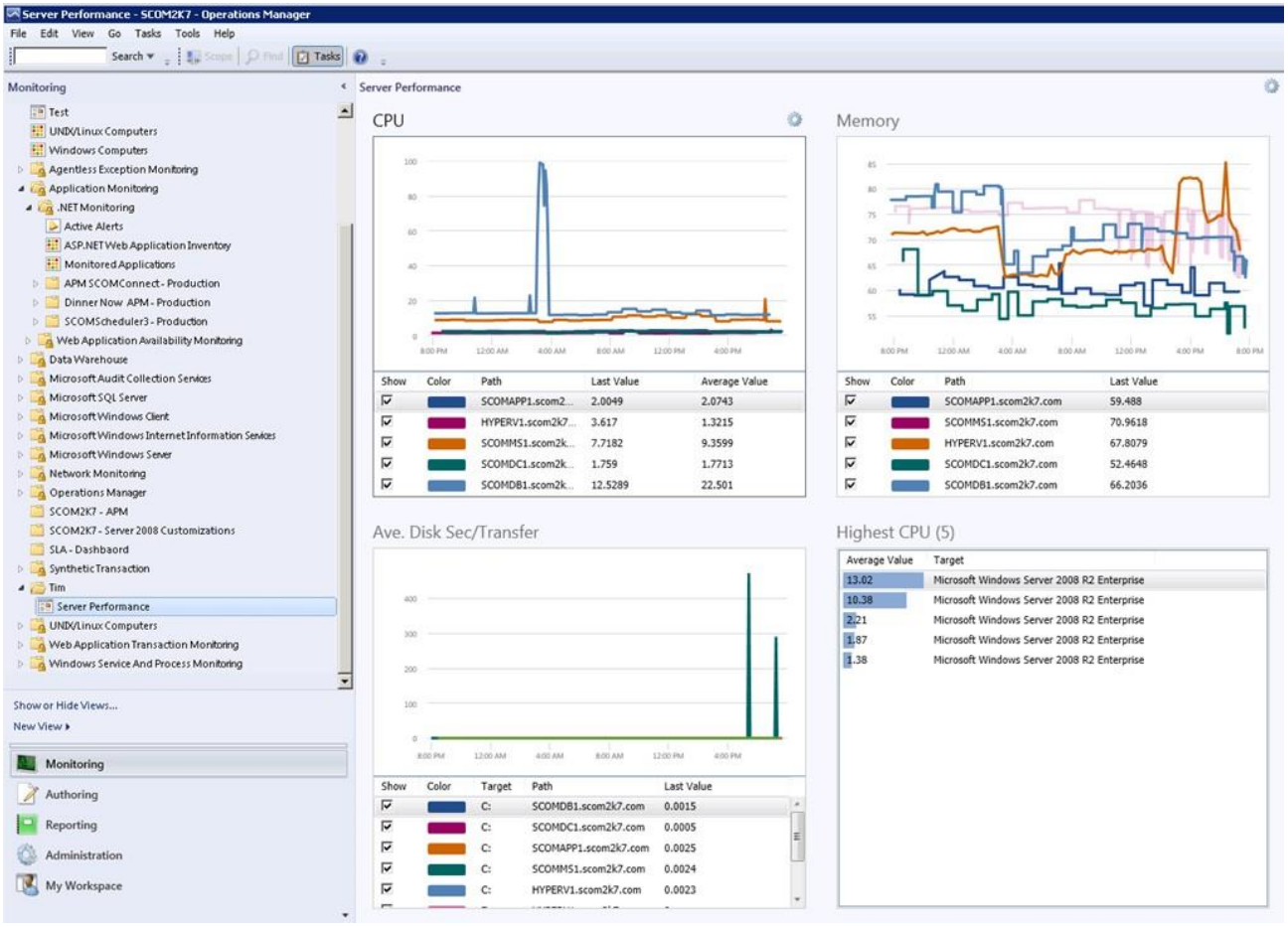

Рисунок 1.2 – Інтерфейс Microsoft SCOM

- робота з операційними системами поза сімейства Windows нестабільна;
- неймовірна громіздкість і складність налаштування продукту “під себе”: система швидше підходить для моніторингу загального стану і збору основних відомостей про великі структури (наприклад, великої кількості клієнтських і серверних машин в домені).

Останній недолік обумовлює відмову від цієї системи адже є необхідність розгортати глобальний сервіс, встановлювати агента на всі відслідковуємі машини і налаштовувати безліч параметрів - скасовувати показники, що збираються за замовчуванням. Система відноситься до продуктів для контролю загального стану великої IT-структури без спостереження за конкретними специфічними показниками. Також істотний недолік системи полягає у високій вартості даного програмного продукту.

1.4.2 Zabbix

Zabbix – вільно поширювана система для комплексного моніторингу мережевого обладнання, серверів та сервісів. Її використовують тисячі компаній, наприклад, DELL, Salesforce. За допомогою Zabbix можна здійснювати розподілений моніторинг до 1000 вузлів, де конфігурація молодших вузлів контролюється старшими в ієрархії. Програма підтримує безліч платформ (Linux, Mac OS, Windows) і доступна через веб-інтерфейс. З його допомогою можна отримати доступ до даних моніторингу з будь-якого ПК. Також продукт включає централізований моніторинг лог-файлів, можливість створювати мапи мереж (рис. 1.3) (вручну за шаблоном), виконання запитів в різних базах даних, генерацію звітів і тенденцій, виконання сценаріїв на основі моніторингу, підтримку інтелектуального інтерфейсу управління платформами (IPMI).

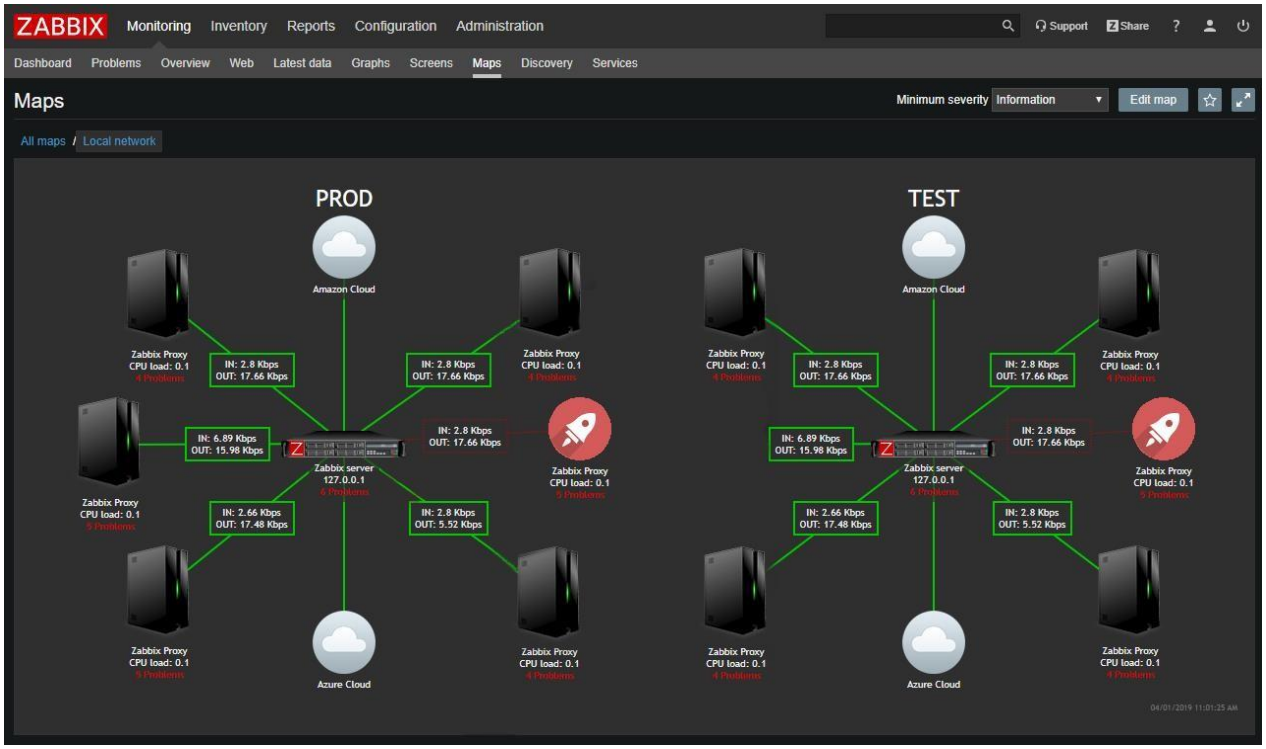

Рисунок 1.3 – Мапа мереж в Zabbix

Zabbix надає гнучкі можливості для налаштуванню умов-тригерів, які спрацьовують при аварії чи неполадках, система починає блимати червоними

квадратиками, оповіщаючи адміністратора про можливі поломки. Також, при включенні тригера, веб-інтерфейс має звукову сигналізацію.

Zabbix досить самостійний і сам зможе відправити повідомлення на пошту, в Jabber або SMS-повідомлень за допомогою GSM-модему, або навіть спробувати самостійно підняти “впавший” сервіс, виконавши заздалегідь певні дії, які запускаються при спрацьовуванні певних тригерів. Для відображення логічної структури мережі можна вручну створювати мапи мережі, що відображають саме розташування вузлів мережі і зв’язки між ними, причому поточний стан вузлів буде відображатися на мапі.

Автоматичне виявлення:

- автоматичне виявлення за діапазоном IP-адрес, доступним сервісам і SNMP перевірка;
- автоматичний моніторинг виявлених пристроїв;
- автоматичне видалення відсутніх хостів;
- розподіл по групах і шаблонами в залежності від того, що повертається у якості результату.

У Zabbix є ще багато різних функцій, які дозволяють ще більше спростити спостереження за мережею, такі як моніторинг стану веб-сайту за допомогою автоматичного виконання сценарію на кшталт імітації призначених для користувача дій на сайті. У підсумку це одна з найпотужніших і обширніших систем моніторингу.

Однак варто відзначити громіздкість сервісу, відсутність повної документованості можливостей проекту, а також необхідність установки агента забезпечення на всі машини.

В якості ще одного мінуса варто відзначити складність делегування прав пристрій з сервісом найчастіше керується операційною системою сімейства *nix, що робить трудомістким взаємодію з доменними користувачами і правами з Active Directory (Windows системи).

1.4.3 Nagios

Nagios – система моніторингу з відкритим вихідним кодом, призначена для спостереження, контролю стану та оповіщення адміністратора, якщо якась служба припинить свою роботу. Спочатку розроблена для операційних систем на базі Linux, зараз однаково добре працює також і під Sun Solaris, FreeBSD, AIX і HPUX. За допомогою цієї програми доступні комплексне спостереження за всією IT-інфраструктурою, виявлення проблем відразу після їх виникнення, можливість ділитися отриманими при спостереженні даними із зацікавленими особами, моніторинг безпеки системи, як наслідок, скорочення часу простою і комерційних втрат.

Але головними причинами відмови від використання системи є:

- “загальний” характер моніторингу показників (рис. 1.4);

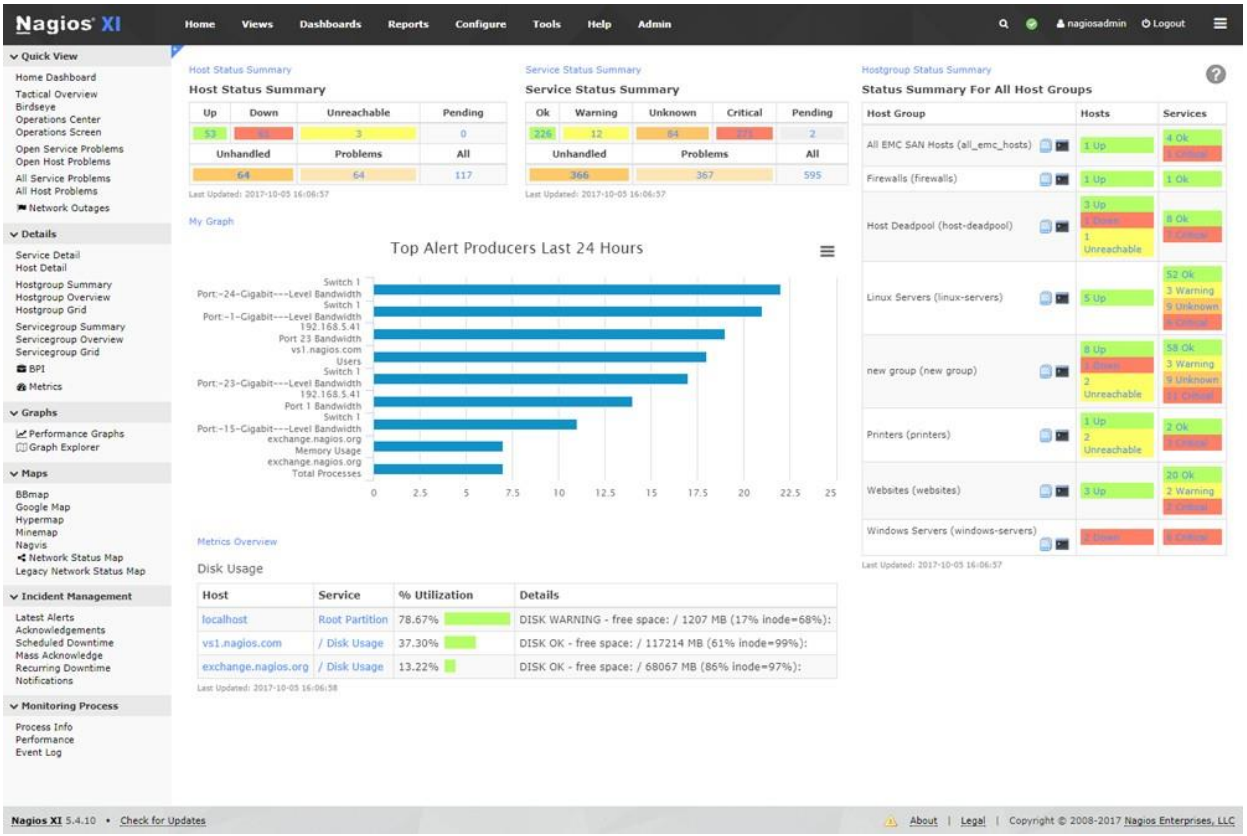

Рисунок 1.4 – Моніторинг показників в Nagios

- проблема взаємодії з серверами під управлінням Windows;
- “мережна” спрямованість моніторингу.

1.4.4 Cacti

Cacti – безкоштовний додаток моніторингу, що дозволяє збирати статичні дані за певні часові інтервали і відображати їх в графічному вигляді за допомогою RRD tool утиліти, призначеної для роботи з круговими базами даних (RRD), які використовуються для зберігання інформації про зміні збору включають статистику по завантаженню процесора, кількість виділеної оперативної пам'яті, кількістю запущених процесів, використання вхідного/вихідного трафіку.

Cacti розроблений в інфраструктурі Apache-PHP-MySQL, дозволяє налаштовувати збір і відображення даних моніторингу на основі веб-інтерфейсу, з юзер-френдлі організацією. Є можливість дописування власних агентів збору даних.

Інтерфейс відображення статистики, дозібраної з пристроїв, представлений у вигляді дерева, структура якого задається самим користувачем. Як правило, графіки групують за певними критеріями, причому один і той же графік може бути присутнім в різних гілках дерева. Є варіант перегляду заздалегідь складеного набору графіків, і є режим попереднього перегляду. Кожен з графіків можна розглянути окремо, при цьому він буде представлений за останні день, тиждень, місяць і рік. Можливо самому обрати часовий відрізок, за який буде створено графік.

Переваги Cacti:

- висока швидкість розгортання при мінімальному додатковому програмуванні;
- простота і зручність інтерфейсу перегляду графіків і їх налаштування (рис. 1.5).

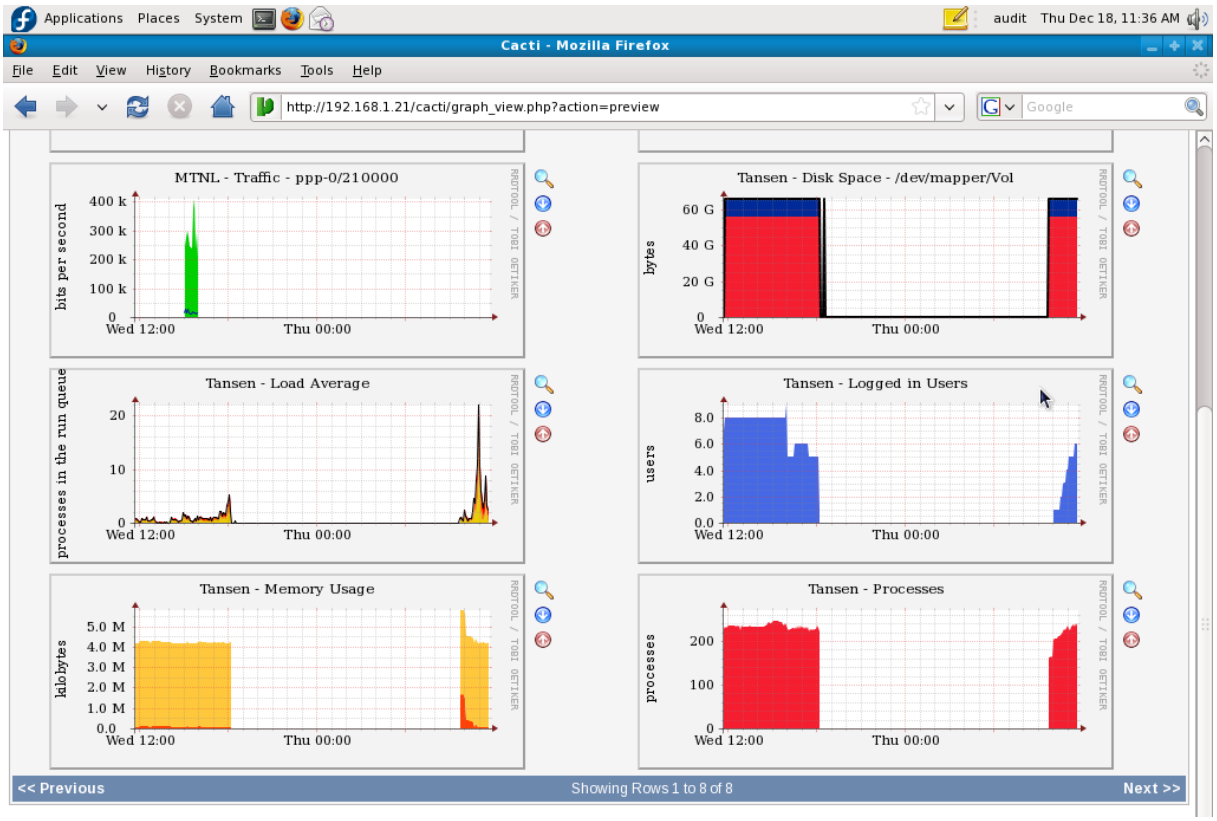

Рисунок 1.5 – Графіки в Cacti

Недоліки Cacti:

- досить швидке збільшення кількості однотипних налаштувань в разі великого числа середовищ і серверів;
- обмежена продуктивність “нерідних” JMX рішень для Cacti;
- відсутність можливості інвентаризації.

Висновки. Належне функціонування та безпека IT-системи – це, перш за все, питання продуманої концепції. Часто не потрібні великі інвестиції, а потрібна цілісна концепція, яка також може бути реалізована економічно.

Тому, Zabbix – це система, яка активно влазить в управління процесами IT-інфраструктури. Як результат, помилки попередньо розпізнаються на ранніх стадіях і навіть вирішуються до їх виникнення. Доступність системи збільшується завдяки ініціативному підходу. Помилки зведені до мінімуму, що дозволяє уникнути зайвих витрат, а можливість створювати мапи мереж та підтримка інтелектуального інтерфейсу управління платформами надає можливість стежити за всіма наявними пристроями без зайвих зусиль.

2 СИСТЕМА МОНІТОРИНГУ ZABBIX

2.1 Склад та можливості системи моніторингу Zabbix

Zabbix – вільно розповсюджувана система для комплексного моніторингу мережевого обладнання, серверів та сервісів. Zabbix – високо інтегроване рішення моніторингу мережі, яке пропонує безліч функцій в одному пакеті.

Збір даних складається з чотирьох частин:

1. Сервер моніторингу (ядро) – виконує періодичне опитування і отримання даних, обробляє їх, аналізує, також здійснює запуск сценаріїв для розсилки повідомлень. Може віддалено перевіряти мережеві сервіси, є сховищем, в якому зберігаються всі конфігураційні, статистичні та оперативні дані;

2. Проксі – збирає дані про продуктивність і доступність від імені Zabbix сервера. Всі зібрані дані заносяться в буфер на локальному рівні і передаються на Zabbix сервер, до якого належить проксі сервер. Zabbix проксі є ідеальним рішенням для централізованого віддаленого моніторингу місць та мереж, які не мають локальних адміністраторів. Він може бути також використаний для розподілу навантаження одного Zabbix сервера. У цьому випадку, проксі тільки збирає дані, тим самим на сервер лягає менше навантаження;

3. Агент – спеціальний “демон”, який запускається на пристроях за якими відбувається спостереження та надає дані серверу, здійснюючи контроль локальних ресурсів і додатків (таких як жорсткі диски, пам’ять, статистика процесора і т. д.) на системах мережі, тобто ці системи повинні працювати з увімкнутим Zabbix агентом (проте моніторинг можна проводити не тільки за допомогою нього, але і з використанням протоколу SNMP версій 1, 2, 3, запуском зовнішніх “скриптів”, що видають дані, і кілька видів зумовлених вбудованих перевірок, таких як ping, запит по HTTP, SSH, FTP і іншим протоколам, а також вимір часу відповіді цих сервісів). Zabbix агенти є надзвичайно ефективними через використання вбудованих системних викликів для збору інформації про статистику. Zabbix-агенти підтримуються не тільки на

UNIX операційних системах, а й на AIX і Windows;

4. Веб-інтерфейс – засіб візуального представлення Zabbix, реалізований на PHP, для запуску вимагає наявності веб-сервера.

За допомогою Zabbix можна здійснювати розподілений моніторинг до 1000 вузлів, де конфігурація молодших вузлів контролюється старшими в ієрархії (рис. 2.1). Також продукт включає централізований моніторинг лог-файлів, можливість створювати мапи мереж (вручну за шаблоном), виконання запитів в різних базах даних, генерацію звітів і тенденцій, виконання сценаріїв на основі моніторингу, підтримку інтелектуального інтерфейсу управління платформами (IPMI).

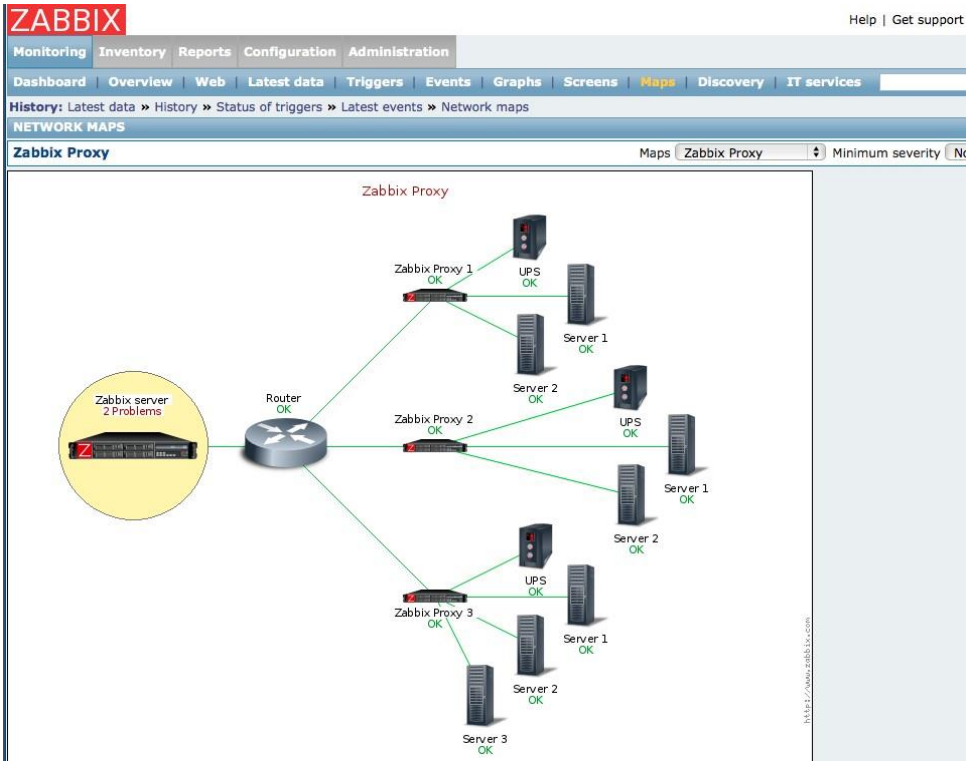

Рисунок 2.1 – Ієрархія вузлів Zabbix

Для збору інформації Zabbix використовує системні виклики, завдяки чому, вплив на продуктивність мінімальний.

У систему моніторингу вже вбудований ряд стандартних метрик:

- навантаження на процесор, в тому числі окремими процесами;
- обсяг вільної оперативної пам’яті;
- активність жорсткого диска;

- обсяг вільної фізичної пам’яті;

- мережева активність;

- пінг.

А також інші перевірки загального призначення і для найпоширеніших сервісів, таких як веб-сервер, СУБД, SSH, Telnet, VMware, NTP, POP, SMTP, FTP та інших. Щоб задати реакцію при відхиленні будь-яких метрик від норми, використовуються спеціальні умови – тригери (рис. 2.2). Наприклад, якщо пінг відсутній п’ять хвилин, виводиться повідомлення адміністратору і виконується команда перезапуску різноманітних служб. Для виходу з нештатної ситуації застосовуються окремі умови, тому незначне поліпшення метрики не є достатнім для усунення помилки. Наприклад, якщо вільного місця на жорсткому диску залишилося менше 15%, спрацює аварійний тригер і щоб він вимкнувся, значення має перевищувати 40%. Якщо готового функціонала недостатньо, то можна використовувати свій – налаштувати реакцію на певний висновок команд або використовуючи API створений додаток.

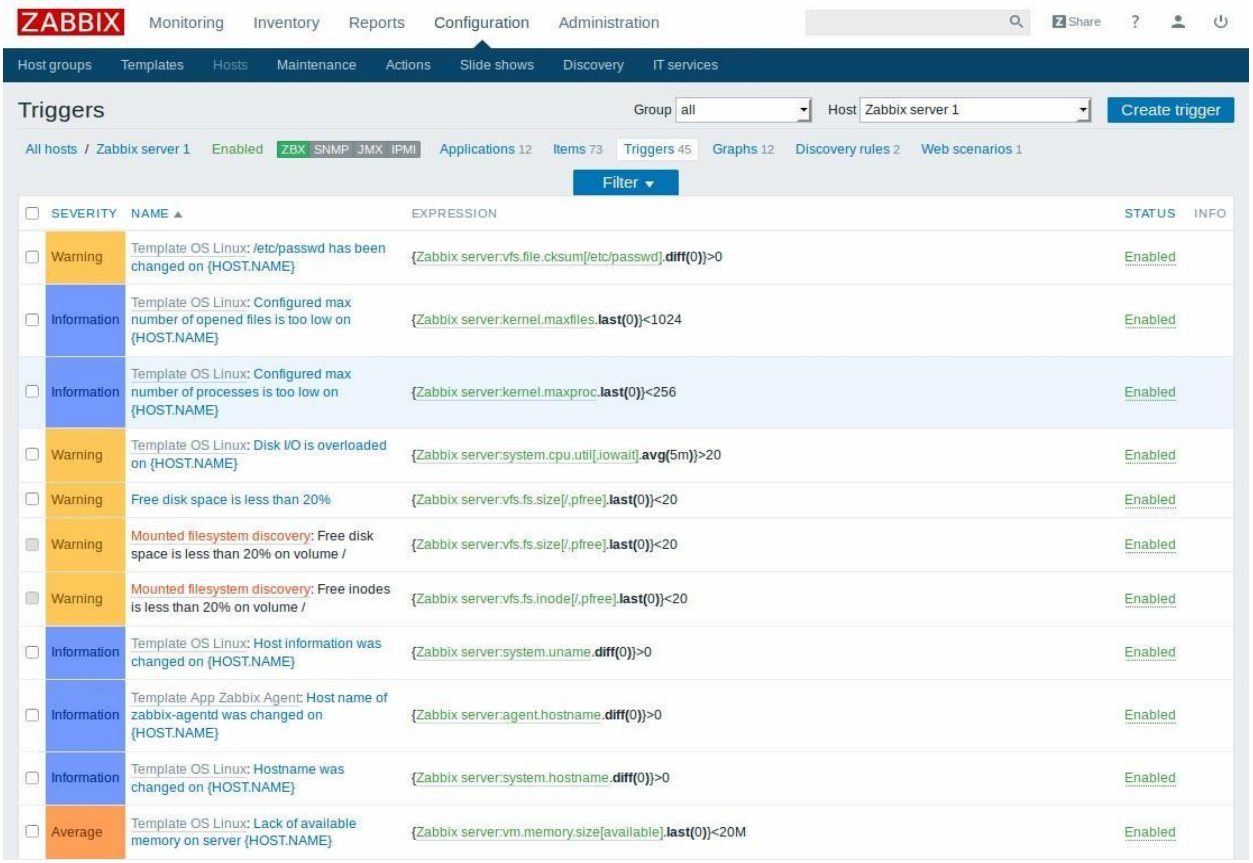

Рисунок 2.2 – Тригери в Zabbix

Тригери представляють собою логічні вирази, мета яких обробляти накопичені дані. Їх можна складати як вручну, так і за допомогою конструктора. Є функція тестування тригерів на довільних значеннях. Для складання тригерів використовуються оператори Zabbix, підставляють необхідні дані, в тому числі з конкретної перевірки або за заданий інтервал часу.

Встановлення агента не є обов'язковим, тому що на вибір адміністратора є багато шляхів здійснення збору інформації з сервера. Наприклад, за допомогою Simple check, SNMP agent, Zabbix trapper та інших.

Система також має низькорівневе виявлення, яке призначається для автоматичного створення елементів і тригерів, щоб відстежувати стан різних систем підконтрольного сервера. Таким чином Zabbix виявляє:

- файлові системи;
- мережеві інтерфейси;
- процесори і їх ядра;
- поширені OID, використовувані SNMP;
- наявність ODBC;
- служби Windows.

Додатково є можливість задати свої типи виявлення, використовуючи формат JSON.

Дуже часто в Zabbix використовують проксі адже інфраструктура досить велика, щоб на одиночний сервер не було надто великого навантаження. Проксі виступає в ролі проміжної ланки, що збирає дані з агентів, як це робить основний сервер. Потім дані з буфера відправляються на центральний сервер. Але це не єдина причина, по якій може знадобиться використання проксі. Він так само потрібен, якщо деякі агенти знаходяться в значно віддалених місцях, що позначається на величині ping і їх доступності.

2.2 Протокол мережевого управління SNMP

Простий протокол мережевого управління (SNMP) – це мережевий протокол, який використовується для управління та моніторингу підключених

до мережі пристроїв у мережах Internet Protocol на основі TCP/UDP. Протокол SNMP вбудований у кілька локальних пристроїв, таких як маршрутизатори, комутатори, сервери, брандмауери та бездротові точки доступу, які доступні за допомогою IP-адреси (рис 2.3). SNMP забезпечує механізм для передачі інформації управління мережевих пристроїв в межах одного або кількох постачальників середовищ LAN або WAN. Це протокол прикладного рівня в рамках моделі OSI.

Зазвичай протокол SNMP реалізується за допомогою User Datagram Protocol. UDP – це бездротовий протокол, який працює як протокол TCP, але передбачає, що перевірка помилок та послуги відновлення не потрібні. Натомість UDP безперервно надсилає датаграми одержувачу, незалежно від того, отримують вони їх чи ні.

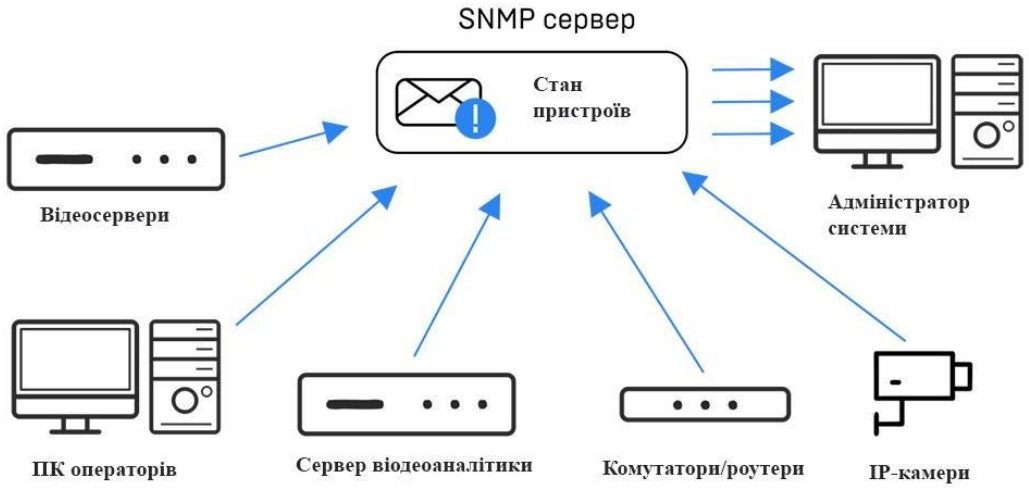

Рисунок 2.3 – Приклад роботи SNMP протоколу

Інформаційні бази управління SNMP – це структури даних, які визначають, що можна збирати з локального пристрою, а що можна змінювати та конфігурувати. Існує багато ІБУ, визначених спеціальними установами з питань стандартів, таких як IETF та ISO, а також запатентовані ІБУ, які визначені конкретними постачальниками IT-обладнання, такими як Cisco, та постачальниками програмного забезпечення, такими як Microsoft та Oracle.

Основними компонентами роботи SNMP є:

- пристрої та ресурси на яких працює агент SNMP;

- агент – збирає дані про різні показники, такі як використання центрального процесора, використання смуги пропускання або дискового простору;

- менеджер – запускає програму управління SNMP у багатьох різних середовищах операційної системи.

- ІБУ – ця структура даних являє собою текстовий файл (із розширенням файлу .mib), який описує всі об’єкти даних, що використовуються певним пристроєм, які можна запитувати або контролювати за допомогою SNMP, включаючи контроль доступу.

В процесі роботи SMNP використовує одного або декількох адміністративних менеджерів SNMP, які контролюють групи мережевих комп’ютерів та пов’язаних з ними пристроїв. Постійно запущена програма, яка називається агентом, передає інформацію менеджерам за допомогою SNMP. Агенти створюють змінні з даних та упорядковує їх у ієрархії, описані ІБУ.

Пильний моніторинг пристроїв SNMP - це основна частина забезпечення безперебійної роботи та безпропускної здатності мережі.

Протокол SNMP настільки популярний, що більшість мережевих пристроїв постачаються в комплекті з агентами SNMP. Однак, щоб скористатися протоколом, адміністратори мережі потрібно спочатку змінити налаштування конфігурації за замовчуванням для своїх мережевих пристроїв, щоб агенти SNMP могли “спілкуватися” із системою управління мережею. Існує три версії SNMP:

- SNMPv1 – перша версія SNMP забезпечувала мінімальні функції управління мережею. SNMPv1 набагато менш безпечний, ніж SNMPv3, оскільки немає контролю над тим, кому в мережі дозволено виконувати операції SNMP і отримувати доступ до об’єктів у модулі MIB. Операціями протоколу, які виконувались через SNMPv1, були Get, GetNext, Set і Trap;

- SNMPv2 – ця версія не покращила безпеку. Нові протоколи включали GetBulk та Inform. Хоча ця версія була потужнішою, ніж SNMPv1, вона також

була більш складною;

- SNMPv3 – ця версія запровадила посилений захист управління IT-системами та мережами. Автентифікація, контроль доступу та зашифровані пакети даних були одними з ключових компонентів, що використовуються для значного покращення параметрів безпеки в SNMPv3.

SNMP є частиною оригінального IP, визначеного робочою групою IETF. Остання версія протоколу, SNMPv3, включає механізми безпеки для автентифікації, шифрування та контролю доступу.

2.3 Моніторинг доступності та показників серверного обладнання з використанням інтерфейсу управління IPMI

IPMI – це стандартизований інтерфейс для віддаленого управління “вимкненням світла” або “поза діапазонних” комп’ютерних систем. Інтерфейс дозволяє контролювати стан апаратного забезпечення безпосередньо за допомогою так званих “позасмугових” карток управління, незалежно від операційної системи або від того, чи машина взагалі ввімкнена.

Zabbix пропонує кілька інтерфейсів для збору даних. Встановлення так званого агента на пристрій, який слід контролювати, пропонує найбільший спектр функцій. Доставляти дані можна пасивно (за допомогою опитування) або активно (за допомогою push-передачі).

Всі елементи системи знаходяться узагальненими відповідно до їх призначення, застосування або основи операційної системи відповідного пристрою у вигляді груп, так званих шаблонів. Ця абстракція дозволяє включати нові пристрої до моніторингу дуже простим та ефективним способом. Після визначення основних функцій, таких як ім’я або IP-адреса пристрою, потрібно обрати відповідні шаблони з пулу власної інсталяції Zabbix, наприклад шаблон для відповідної операційної системи та інші шаблони для перевірки наданих послуг. Отже, шаблони – це альфа та омега Zabbix. Після того, як з’являється широкий набір різних сценаріїв тестування, Zabbix стає потужним збирачем даних і дозволяє легко розширюватися за допомогою нових хостів.

Шаблони надаються офіційним сайтом, а також активною спільнотою користувачів. Той, хто справді серйозно ставиться до моніторингу свого системного ландшафту, рано чи пізно почне створювати власні шаблони. Відтоді доведеться підтримувати конфігураційні файли. Тому краще створити свій індивідуальний шаблон (рис. 2.3), який буде доступний для легкого розгортання через веб-інтерфейс.

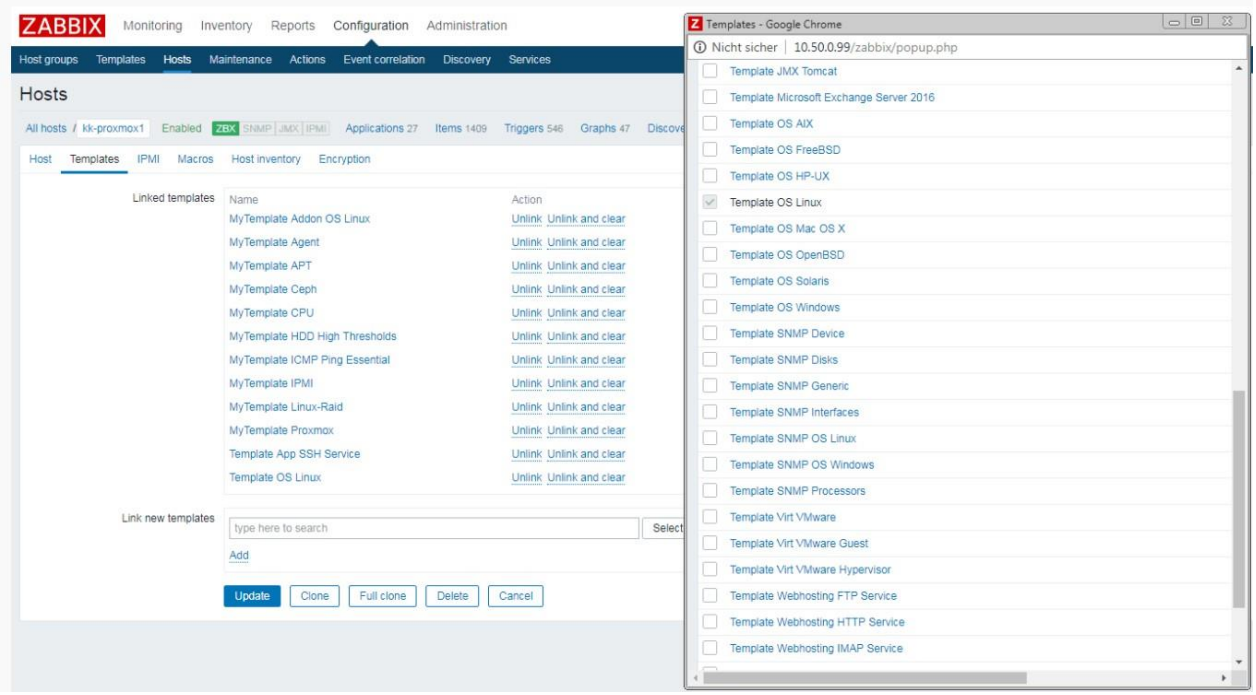

Рисунок 2.3 – Конструктор шаблонів Zabbix

Як доповнення до елементів, які спочатку використовуються лише для збору даних, Zabbix пропонує логічне продовження опції встановлення будильників, наприклад, коли елемент перевищує порогове значення.

Створення динамічних елементів та тригерів за допомогою шаблонів Zabbix можна охарактеризувати як особливість серед інших систем моніторингу. Наприклад, шаблон для запису температури жорсткого диска. Більшість цільових пристроїв матимуть більше 1 жорсткого диска. За допомогою статичних елементів та тригерів можна забезпечити тести на певну кількість жорстких дисків, наприклад 4 диска. Для пристроїв із менше ніж 4 жорсткими дисками це призведе до порожніх або невиконаних елементів та тригерів. Однак на інших пристроях, що мають більше 4 жорстких дисків,

компоненти не будуть враховуватися. Наразі є можливість створювати окремі шаблони для різних конфігурацій жорсткого диска. Але це не потрібно в Zabbix, оскільки шаблони підтримують динамічні елементи та тригери за замовчуванням залежно від кількості фактично знайдених окремих компонентів. Визначення цих компонентів у Zabbix називається Discovery (рис. 2.4). Потім елементи та тригери можуть бути динамічно визначені для всіх записів у масиві JSON. Достатньо одного продуманого шаблону в Zabbix для повного охоплення цілої заявки, тобто відбувається відносно проста у реалізації процедура розпізнавання, яка повертає фактично існуючі окремі компоненти до шаблону Zabbix у вигляді масиву JSON. Потім елементи та тригери можуть бути динамічно визначені для всіх записів у масиві JSON.

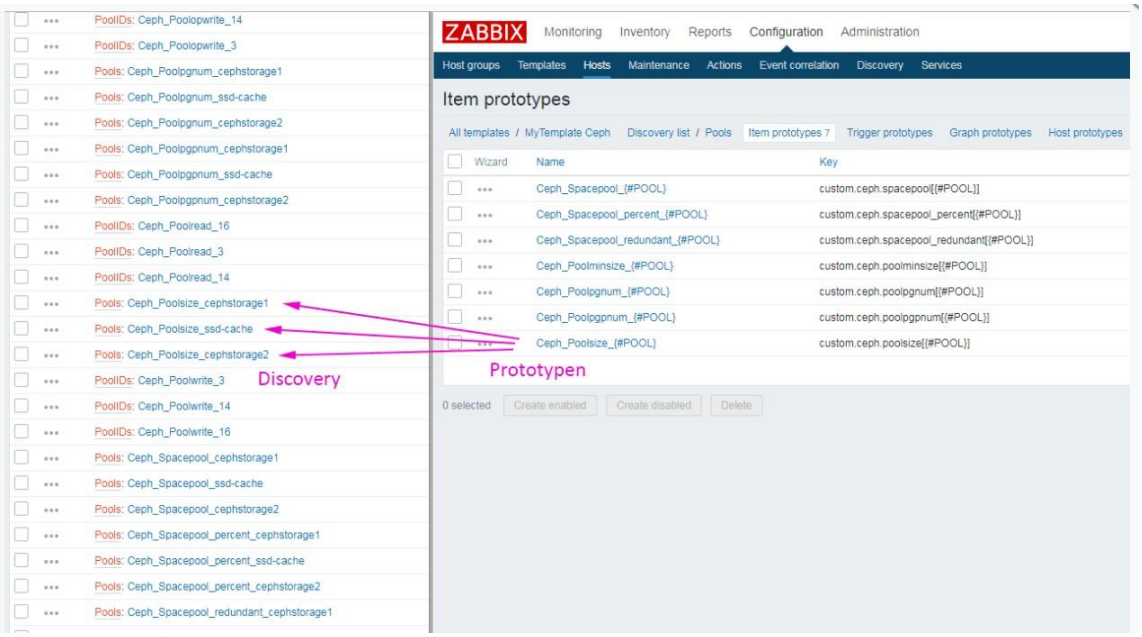

Рисунок 2.4 – Окремі компоненти Discovery

Сценарій інтерфейсу Zabbix також слід згадати, адже він має можливість вільного розширення. Особливо ефективним рішенням для роботи API Zabbix є “скрипти” Python Zabbix CLI, за допомогою якого, наприклад, хости можуть створюватися повністю автоматично в Zabbix, але із заздалегідь визначеними властивостями конфігурації з файлів XML.

Grafana – це потужне програмне забезпечення з відкритим кодом для візуалізації даних, яке можна використовувати для створення чудових,

інтуїтивно зрозумілих інформаційних панелей та звітів. Якщо доповнити функцію моніторингу Zabbix параметрами візуалізації Grafana (рис. 2.5), то можна отримати загальну систему, яка не має аналогів як на стороні моніторингу, так і в області візуального представлення.

Рисунок 2.5 – Візуалізація даних в Zabbix за допомогою Grafana

Zabbix пропонує можливість створювати індивідуальні інформаційні панелі в IPMI (рис. 2.6). Нижче представлений простий екран для обраного хоста (вузол Proxmox з тестового кластера). 6 скомпільованих графіків можна адаптувати або розширити за допомогою вбудованого редактора у веб-інтерфейсі Zabbix, якщо потрібно.

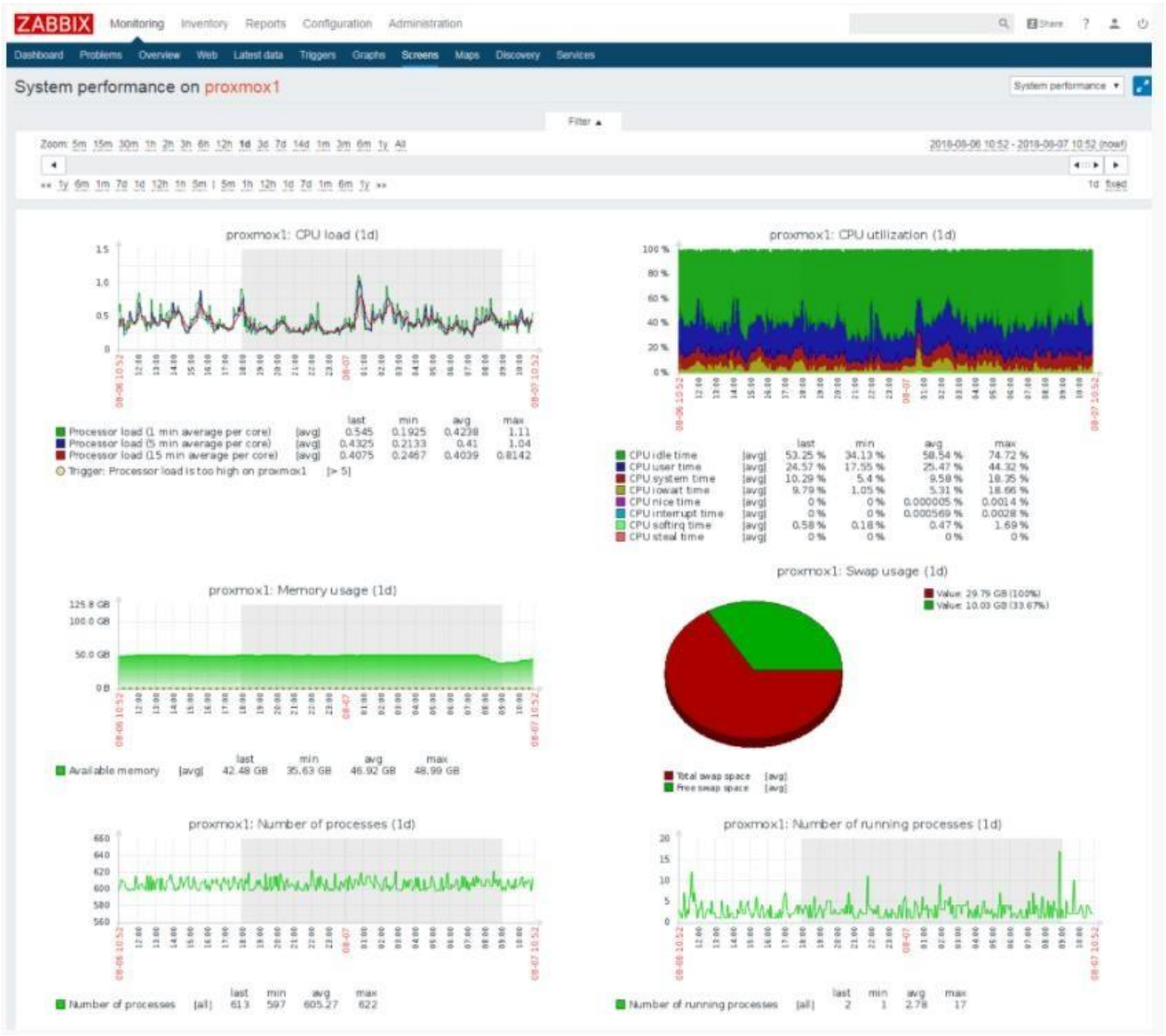

Рисунок 2.6 – Приклад індивідуальної інформаційної панелі

Мапи дають змогу графічно представити зображення з великою кількістю варіантів дизайну. Вони можуть бути використані, наприклад, для створення різних топології (рис. 2.7). Маленькі зелені символи ОК під пристроями відображають поточний стан системи моніторингу. У разі несправності підсвічується відповідний пристрій. Також можна оглянути елементи мапи на інформаційній панелі.

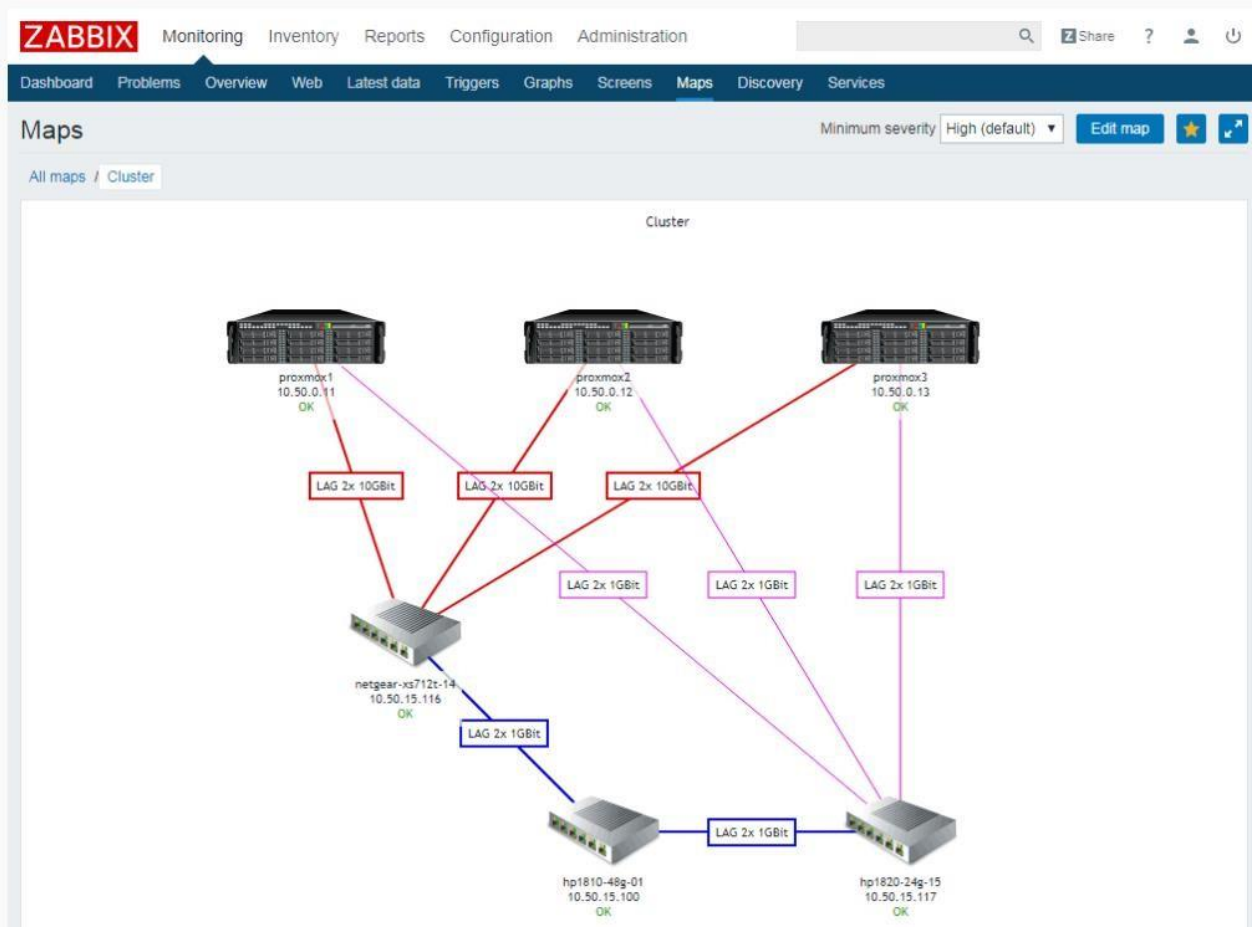

Рисунок 2.7 – Топологічна мапа

Висновки. На теперішній час Zabbix є багатофункціональною системою моніторингу, в якій є все необхідне для повноцінного спостереження за IT-інфраструктурою, включаючи моніторинг мережі, серверів та додатків. Всі функції Zabbix об’єднуються, щоб створити дуже простий і зручний моніторинг. Шаблони моніторингу скорочують обсяг ручного налаштування, яке необхідно виконати для перегляду певної мережі.

А помилки виявляються і усуваються надзвичайно ефективно, адже користувач отримує повідомлення і система сама вживає заходів для усунення цих помилок.

3 ПЛАНУВАННЯ РОЗГОРТАННЯ СИСТЕМИ МОНІТОРИНГУ ZABBIX

3.1 Вимоги до апаратних засобів системи моніторингу Zabbix

Для стабільної роботи Zabbix потрібно мати достатній об’єм оперативної пам’яті та фізичної пам’ять на жорсткому диску. Мінімум потрібно мати 128 МБ оперативної пам’яті та вільні 256 МБ на жорсткому диску. Взагалі об’єм дискової пам’яті залежить від кількості вузлів мережі та параметрів за якими стежить системний адміністратор. Довготривале збереження даних історії в базі даних потребує хоча б декілька гігабайт пам’яті. Кожен із процесів “демона” Zabbix потребує декілька підключень до бази даних. Від налаштувань бази даних буде залежати об’єм пам’яті, який потрібен для підключення бази даних. Швидкість роботи системи залежить від об’єму оперативної пам’яті на пристрої.

Потрібно мати потужний процесор так як Zabbix і особливо база даних може вимагати значних процесорних ресурсів в залежності від кількості спостережуваних параметрів і обраної бази даних.

Zabbix потрібен послідовний порт передачі даних і GSM-модем для використання вбудованих SMS повідомлень.

В табл. 3.1 приведені декілька варіантів апаратних конфігурацій обладнання.

У зв’язку з вимогами безпеки і критично важливим характером роботи системи моніторингу, єдиною операційною системою, яка може забезпечити необхідну продуктивність, відмовостійкість і гнучкість є операційна система UNIX. Zabbix працює на всіх провідних версіях ОС. Наприклад: Linux, Windows, Mac OS X, Solaris та багато інших.

Таблиця 3.1

Апаратні конфігурації обладнання

Платформа	CPU/Пам’ять	База даних	Моніторинг вузлів мережі
CentOS	Віртуальна машина	MySQL InnoDB	20
CentOS	2 ядра CPU / 2ГБ	MySQL InnoDB	500
RedHat Enterprise Linux	4 ядра CPU / 8ГБ	RAID10 MySQL InnoDB або PostgreSQL	>1000
RedHat Enterprise Linux	8 ядра CPU / 16ГБ	Швидкий RAID10 MySQL InnoDB або PostgreSQL	>10000

В табл. 3.2 приведені варіанти доступних для використання та найбільш популярних баз даних у Zabbix.

Таблиця 3.2

Популярні бази даних

Програма	Версія	Коментарі
MySQL	5.0.3 – 8.0.x	Потрібна, якщо MySQL використовується як основна база даних Zabbix. Потрібна InnoDB engine. MariaDB також працює з Zabbix.
Oracle	10g або більш нова	Потрібна, якщо Oracle використовується як основна база даних Zabbix
PostgreSQL	8.1 або більш нова	Потрібна, якщо PostgreSQL використовується як основна база даних Zabbix
IBM DB2	9.7 або більш нова	Потрібна, якщо IBM DB2 використовується як основна база даних Zabbix.
SQLite	3.3.5 або більш нова	Підтримується тільки на стороні Zabbix проксі. Потрібна, якщо SQLite використовується базою даних Zabbix проксі.

3.2 Алгоритм розгортання системи моніторингу Zabbix

Встановлення Zabbix 5 на Ubuntu:

Крок 1 – підключення репозиторія Zabbix в систему за допомогою відповідної (рис. 3.1) команди в терміналі Ubuntu.

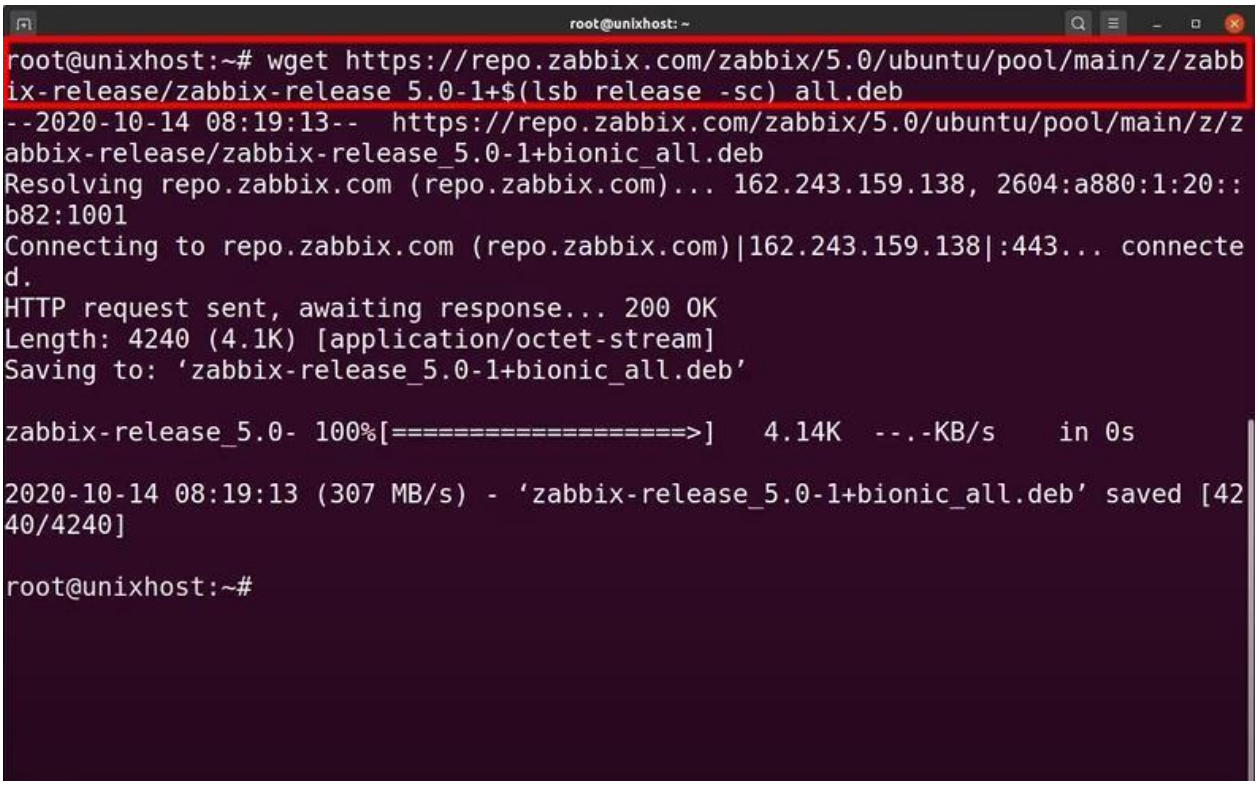

Рисунок 3.1 – Команда для підключення репозиторія Zabbix

Крок 2 – встановлення підключеного репозиторія Zabbix (рис. 3.2).

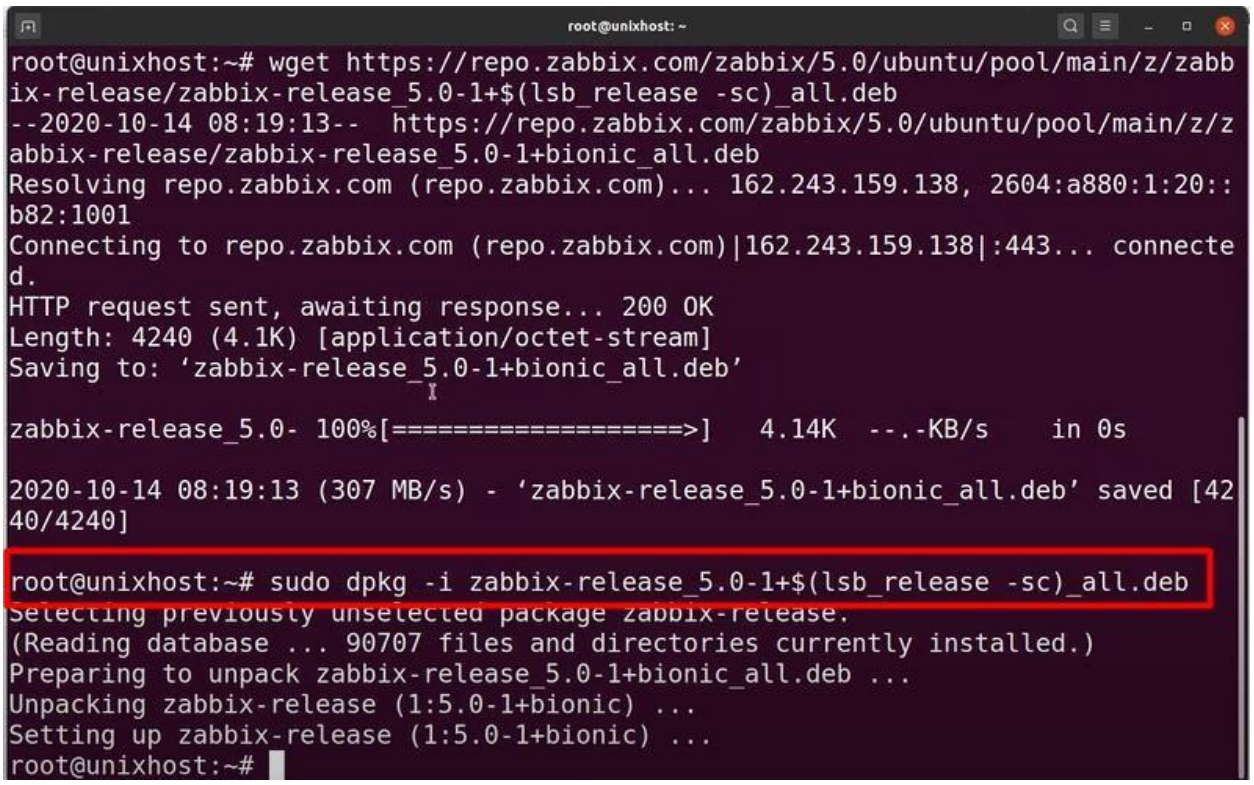

Рисунок 3.2 – Команда для встановлення репозиторія Zabbix

Крок 3 – оновлення системних пакетів (рис. 3.3).

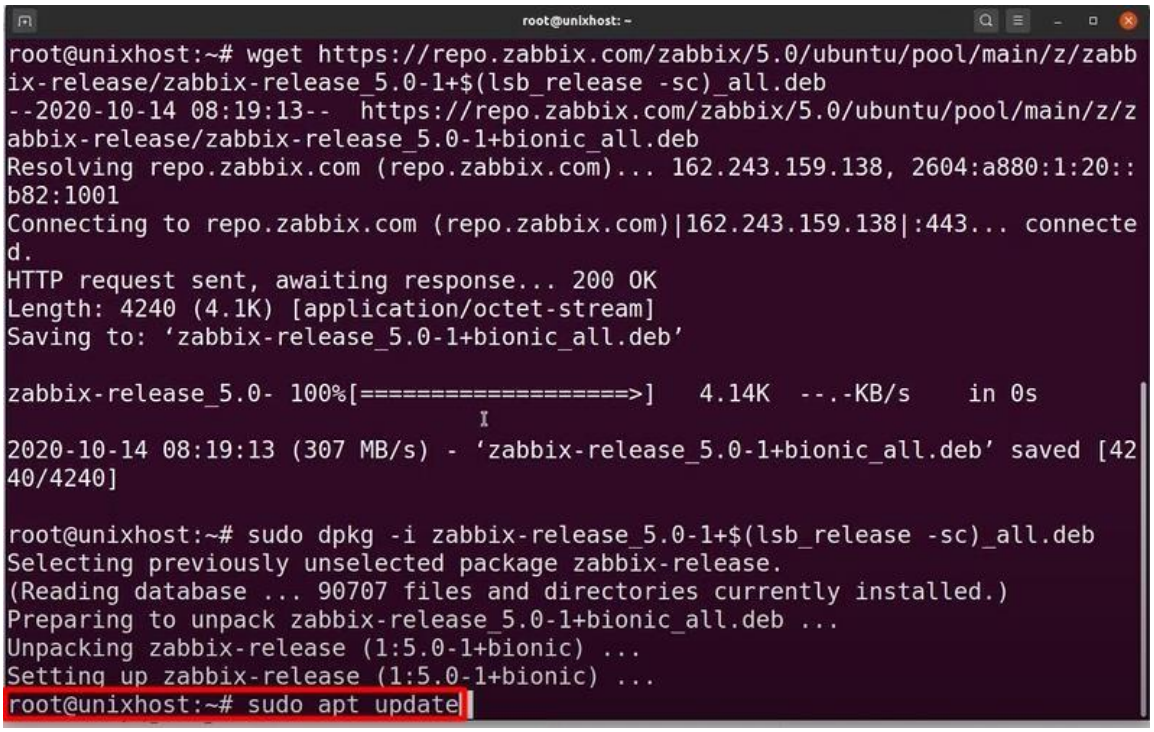

Рисунок 3.3 – Команда для оновлення пакетів системи

Крок 4 – встановлення Zabbix-Server, Zabbix-FrontEnd, налаштування над Apache та Zabbix Agent (рис. 3.4).

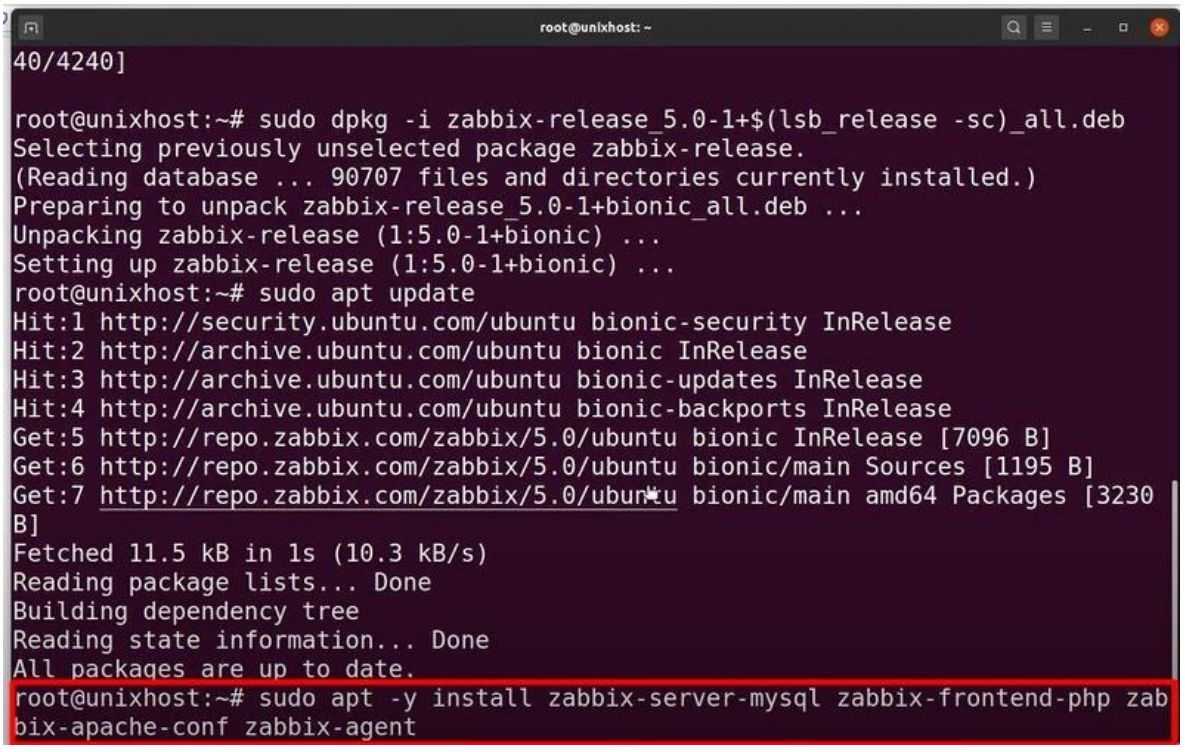

Рисунок 3.4 – Команда для встановлення Zabbix-Server, Zabbix-FrontEnd, налаштування над Apache та Zabbix-Agent

Крок 4 – встановлення серверу бази даних Maria DB та клієнта (рис. 3.5).

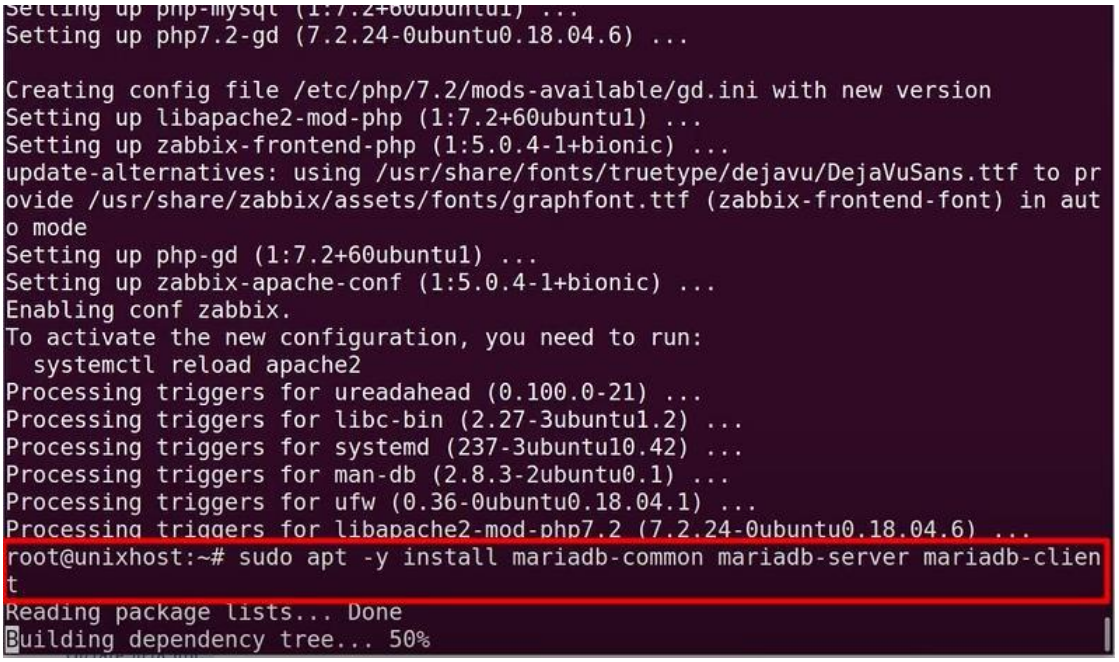

Рисунок 3.5 – Команда для встановлення серверу бази даних Maria DB та клієнта

Крок 5 – встановлення паролю для використання бази даних за допомогою команди (рис. 3.6).

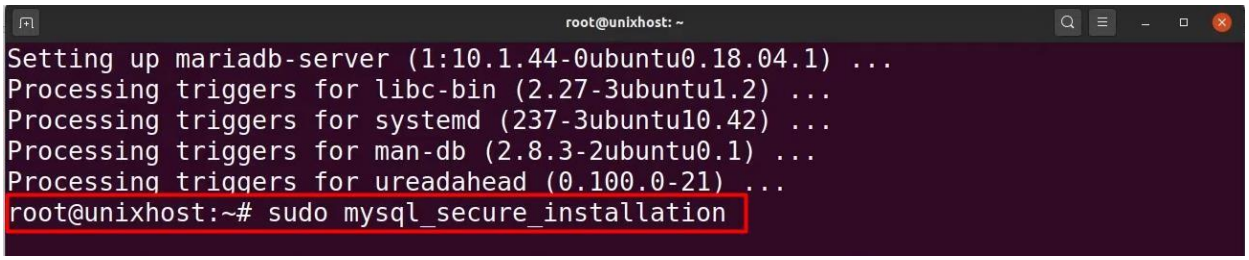

Рисунок 3.6 – Команда для встановлення пароля

Крок 6 – потрібно підтвердити свої наміри натиснувши “Y” + ENTER (рис 3.7).

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] Y
```

Рисунок 3.7 – Підтвердження наміру встановити пароль

Крок 7 – для безпечної роботи вводиться пароль два рази (рис. 3.8).

```
New password:
Re-enter new password:
```

Рисунок 3.8 – Підтвердження пароля

Крок 8 – видалення гостьового користувача, заборона доступ із зовні до бази даних та видалення тестової бази даних після натиснення “Y” + ENTER три рази (рис. 3.9).

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
```

Рисунок 3.9 – Видалення гостьового користувача, заборона доступ із зовні до бази даних та видалення тестової бази даних

Крок 9 – створення бази даних для збереження даних із Zabbix (рис. 3.10).

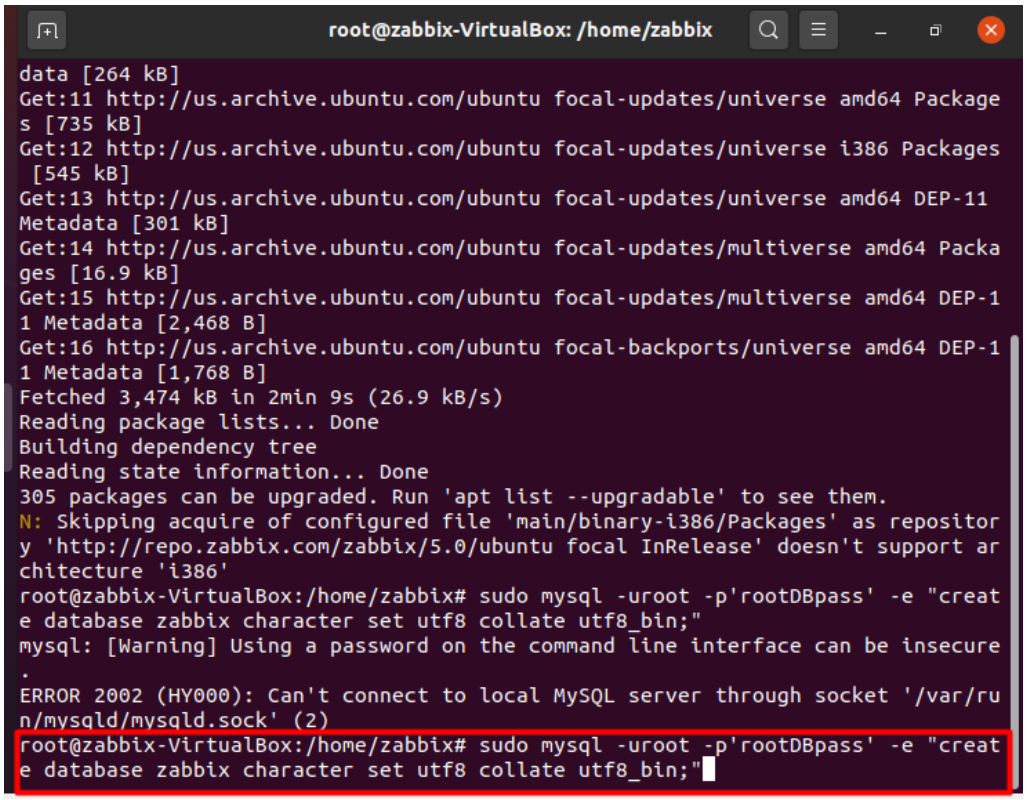

Рисунок 3.10 – Команда для створення бази даних

Крок 10 – створення користувача Zabbix (рис. 3.11).

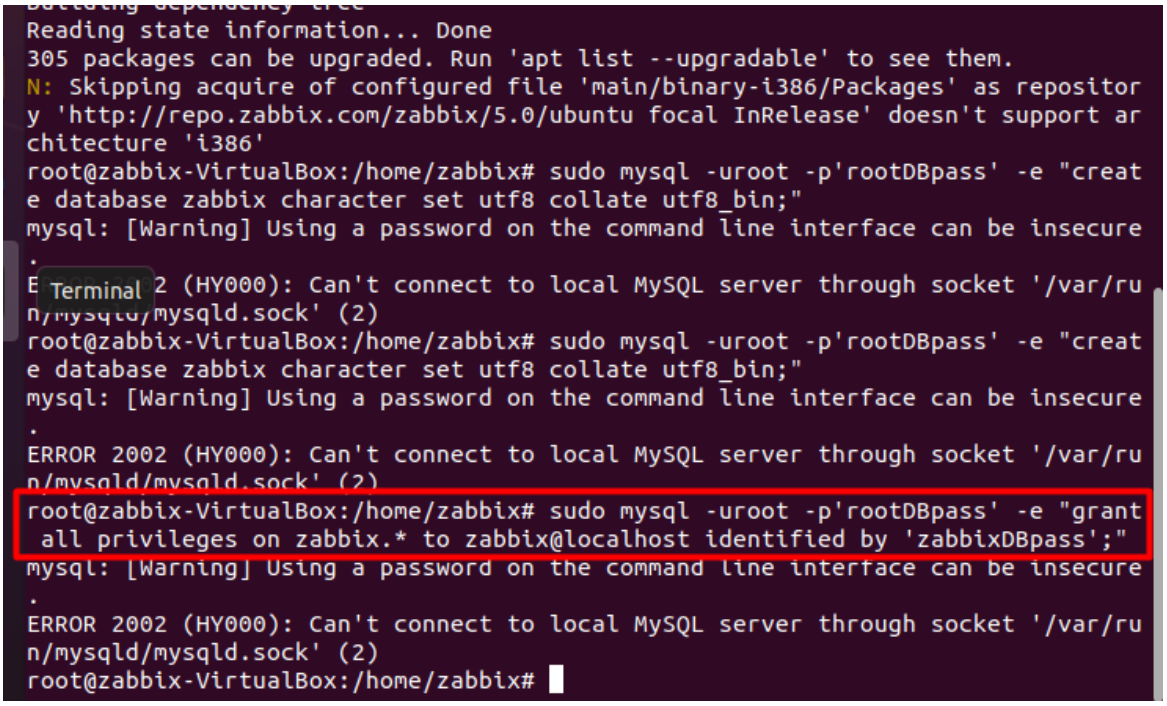

Рисунок 3.11 – Створення користувача Zabbix

Крок 11 – імпортування бази даних (рис. 3.12-3.13).

```
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/ru
n/mysqld/mysqld.sock' (2)
root@zabbix-VirtualBox:/home/zabbix# sudo mysql -uroot -p'rootDBpass' -e "grant
 all privileges on zabbix.* to zabbix@localhost identified by 'zabbixDBpass';"
mysql: [Warning] Using a password on the command line interface can be insecure
.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/ru
n/mysqld/mysqld.sock' (2)
root@zabbix-VirtualBox:/home/zabbix# sudo mysql -uroot -p'rootDBpass' zabbix -e
 "set global innodb_strict_mode='OFF';"
mysql: [Warning] Using a password on the command line interface can be insecure
.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/ru
n/mysqld/mysqld.sock' (2)
root@zabbix-VirtualBox:/home/zabbix# sudo mysql -uroot -p'rootDBpass' zabbix -e
 "set global innodb_strict_mode='OFF';"
```

Рисунок 3.12 – Команда для імпорту бази даних

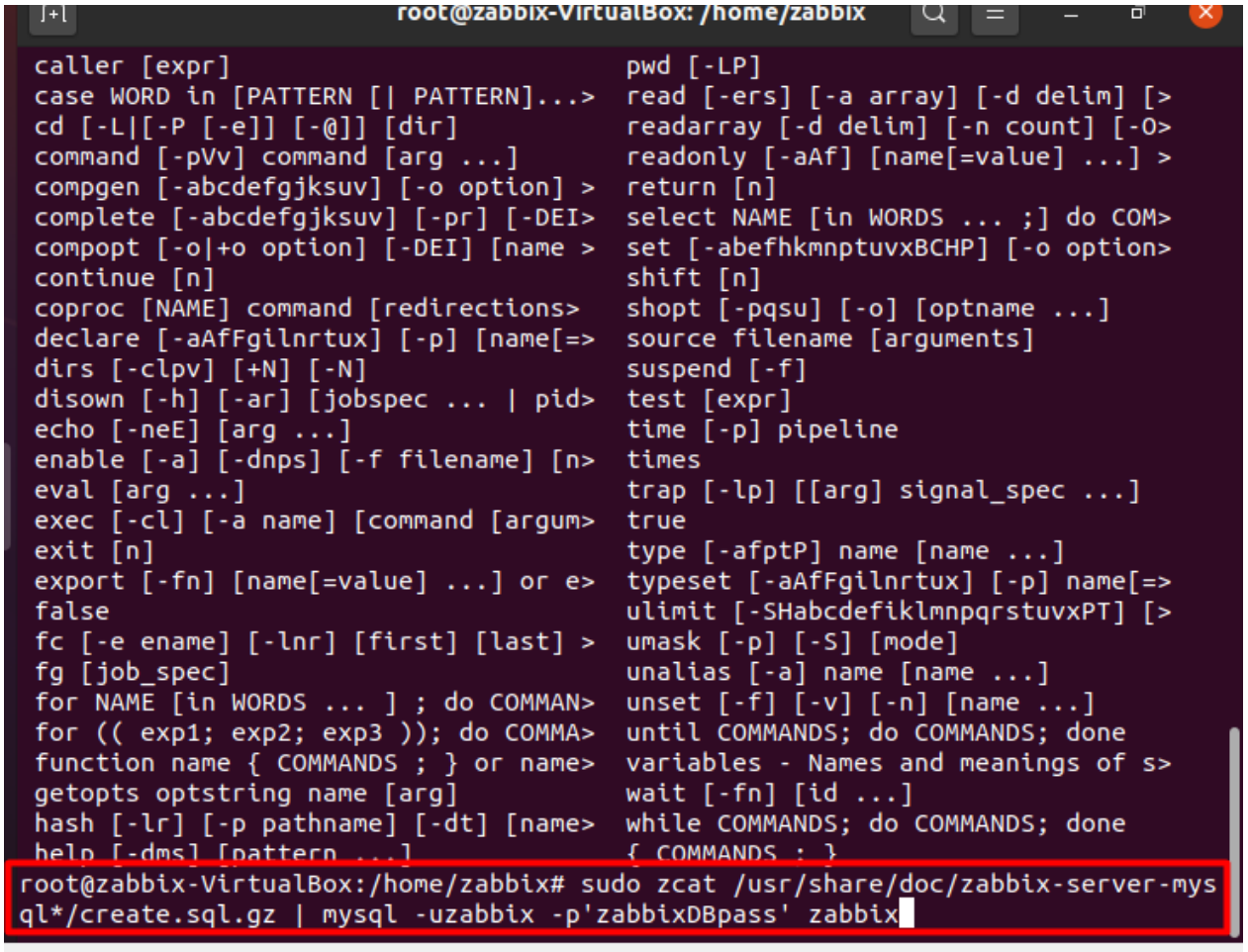

Рисунок 3.13 – Продовження команди для імпорту бази даних

Крок 12 – активація innodb_strict_mode (рис. 3.14).

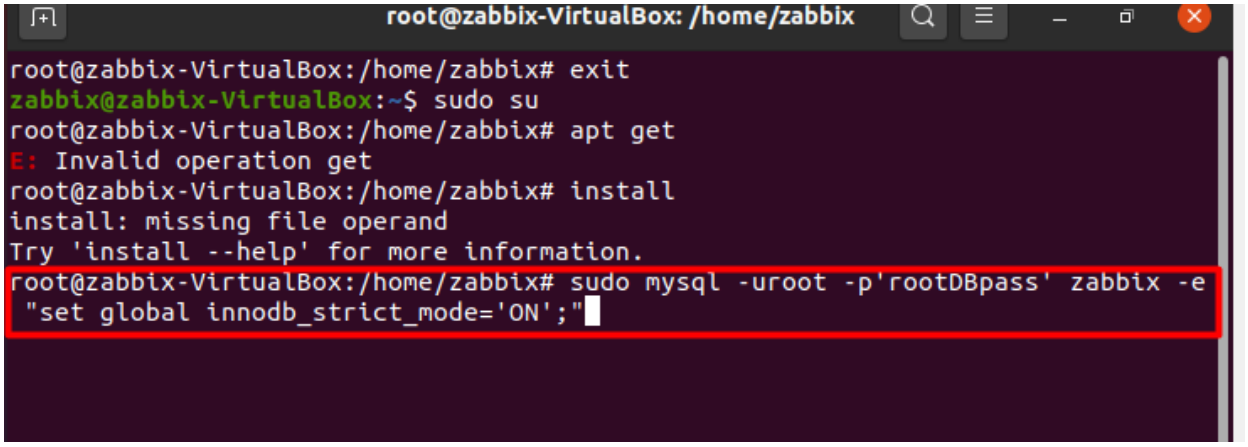

Рисунок 3.14 – Команда для активації innodb_strict_mode

Крок 13 – редагування файлу конфігурацій Zabbix шляхом “розкоментування” рядка, а також ввести в нього пароль та зберегти (рис. 3.15-3.17).

Рисунок 3.15 – Команда для входу у файл конфігурацій Zabbix

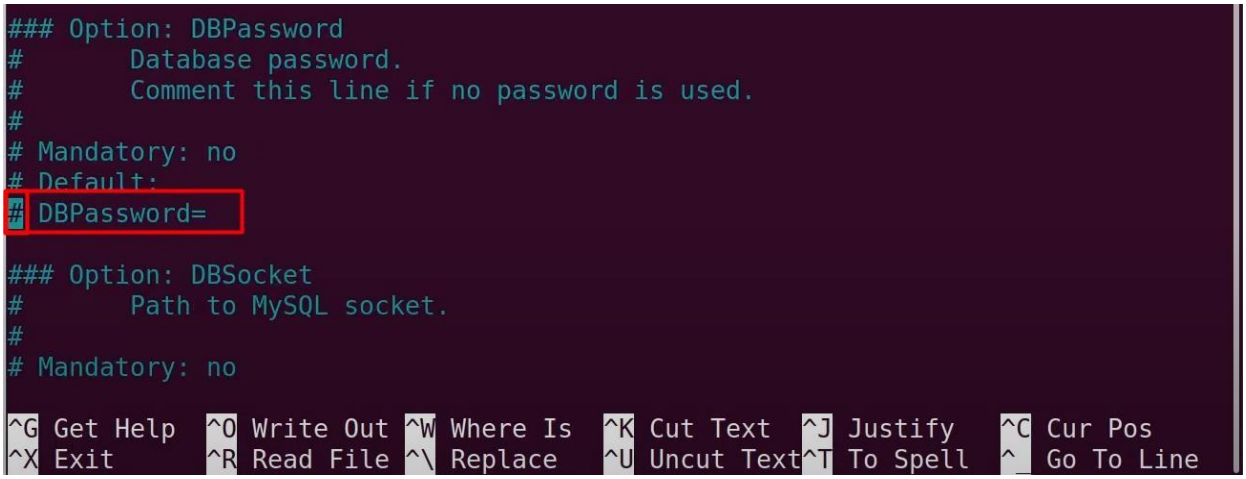

Рисунок 3.16 – Рядок, який потрібно “розкоментувати”

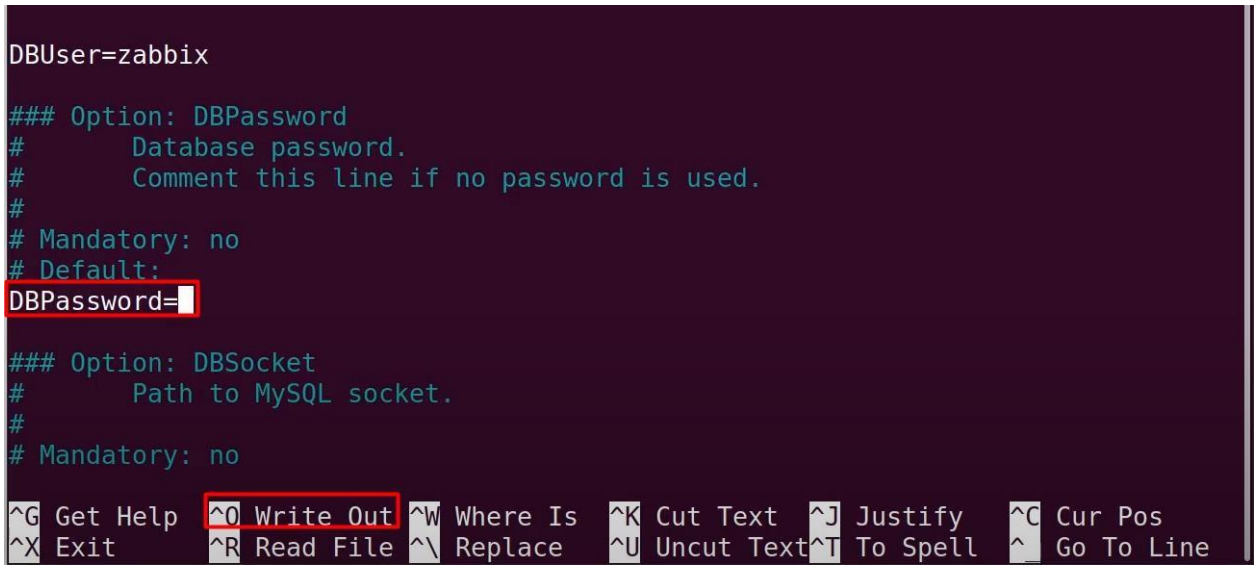

Рисунок 3.17 – Рядок для введення пароля

Крок 14 – запуск Zabbix-server (рис. 3.18).

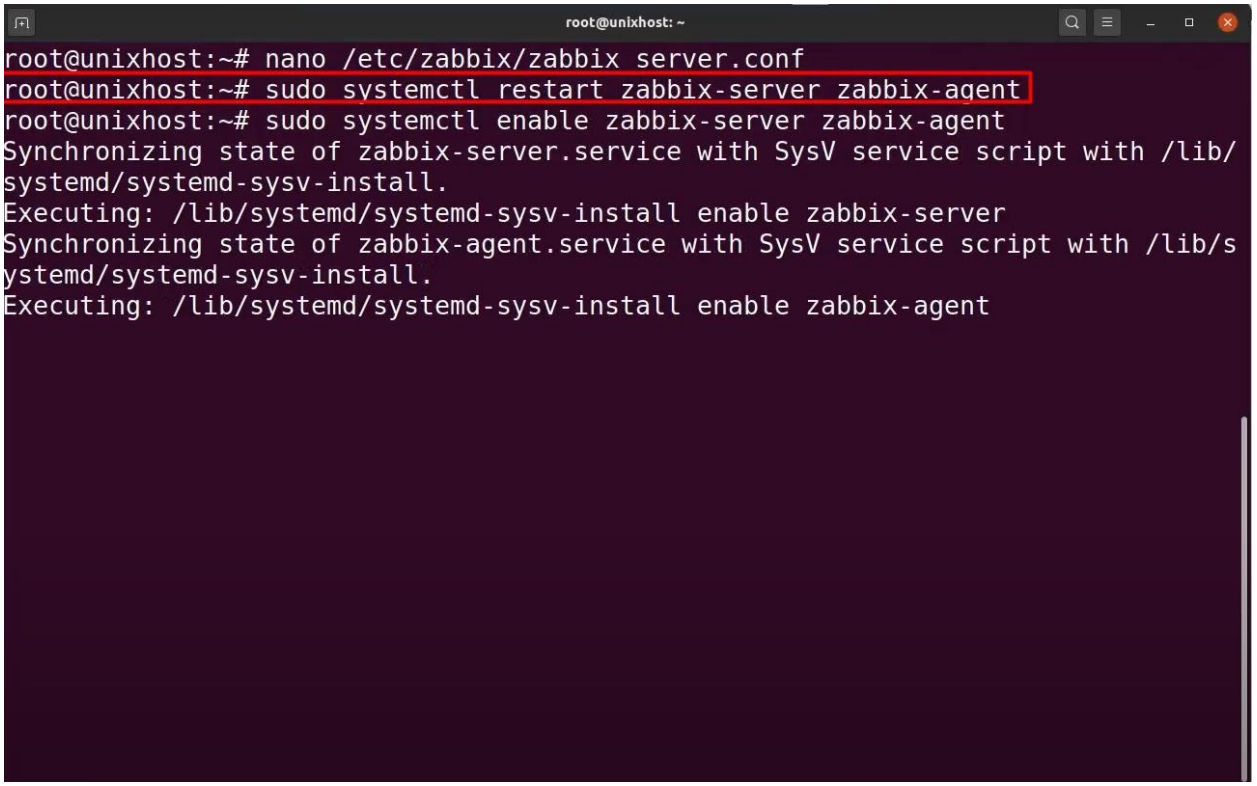

Рисунок 3.18 – Команда для запуску Zabbix-server

Крок 15 – запуск Zabbix-agent (рис. 3.19).

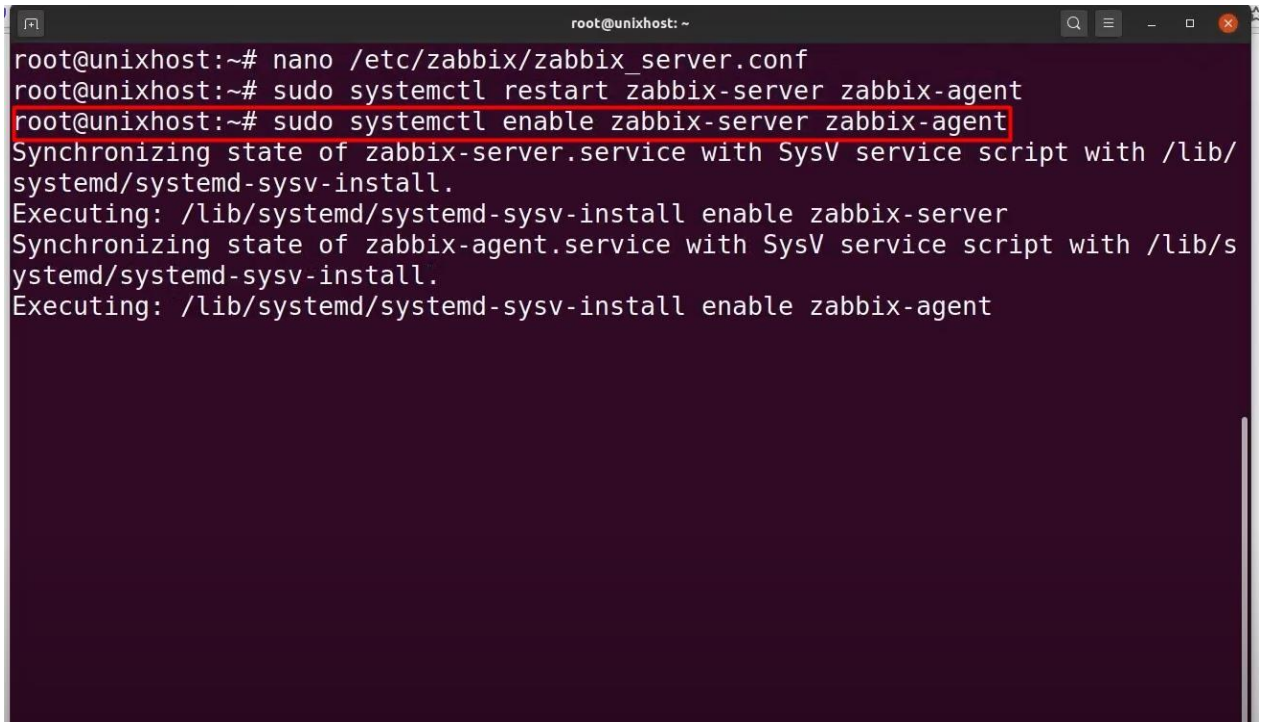

Рисунок 3.19 – Команда для запуску Zabbix-agent

Крок 16 – перезавантаження веб-сервера Apache (рис. 3.20).

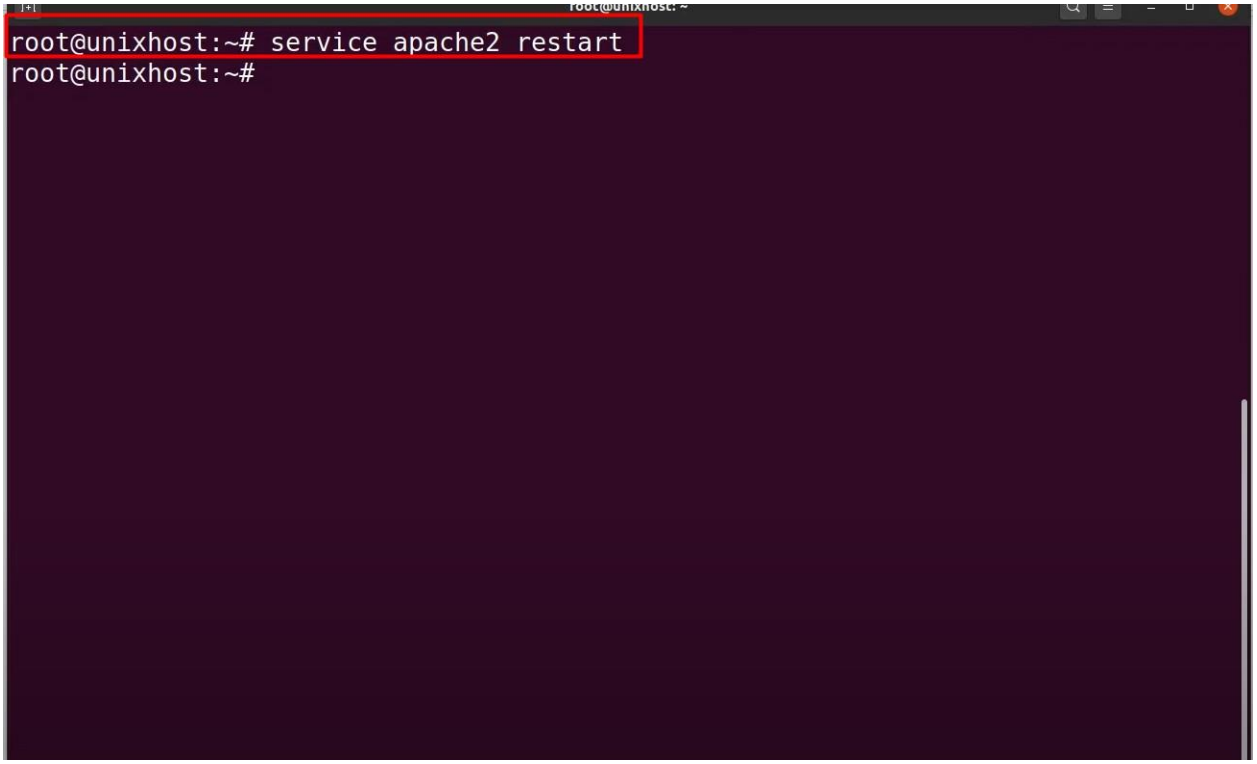

Рис. 3.20 – Команда для перезавантаження веб-сервера

Крок 17 – перехід на веб-інтерфейс Zabbix. Для цього потрібно ввести в браузері посилання (“IP-адреса”/zabbix/setup.php) та натиснути "Next step" (рис. 3.21).

Рисунок 3.21 – Веб інтерфейс Zabbix

Крок 18 – Zabbix перевіряє чи всі параметри справно працюють, якщо це так, то напроти кожного параметра буде відмітка "ОК" і треба натиснути "Next step" (рис. 3.22).

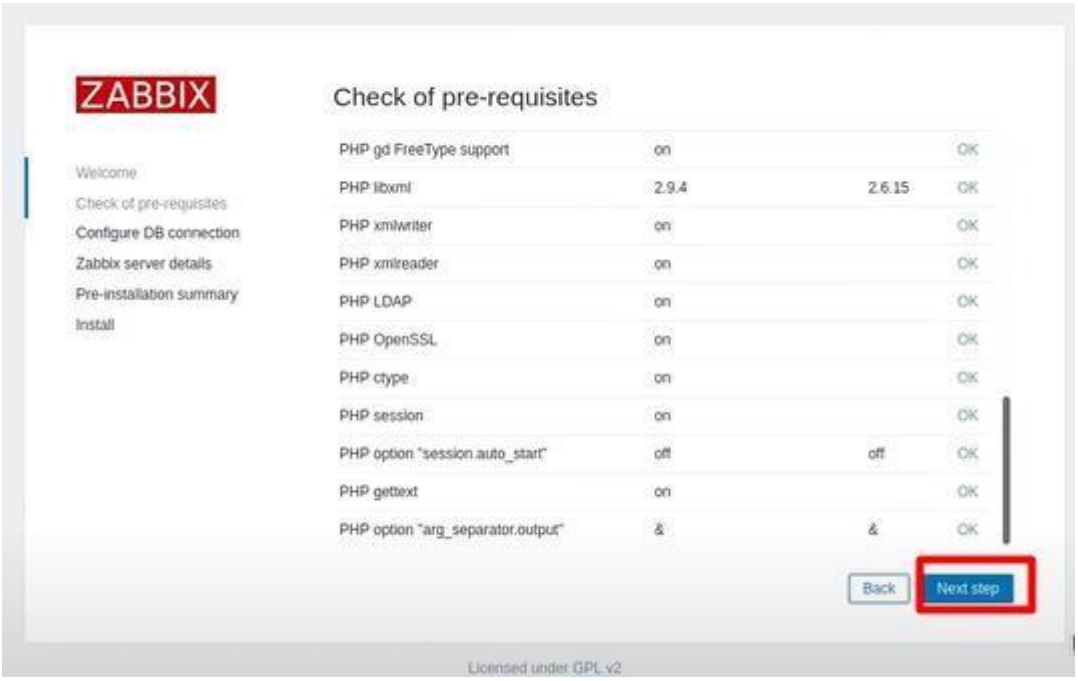

Рисунок 3.22 – Zabbix перевіряє справність параметрів

Крок 19 – задавши пароль користувача для бази даних та натиснути "Next step" (рис. 3.23).

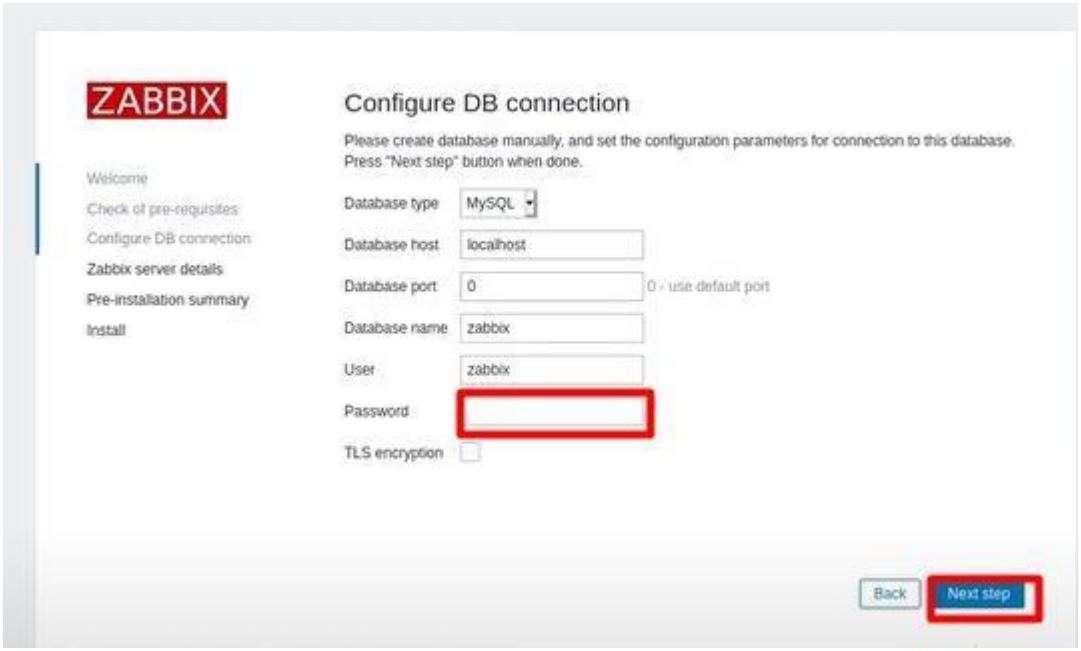

Рисунок 3.23 – Поле для введення пароля

Крок 20 – у виділене поле потрібно ввести IP-адресу сервера та натиснути "Next step" (рис. 3.24).

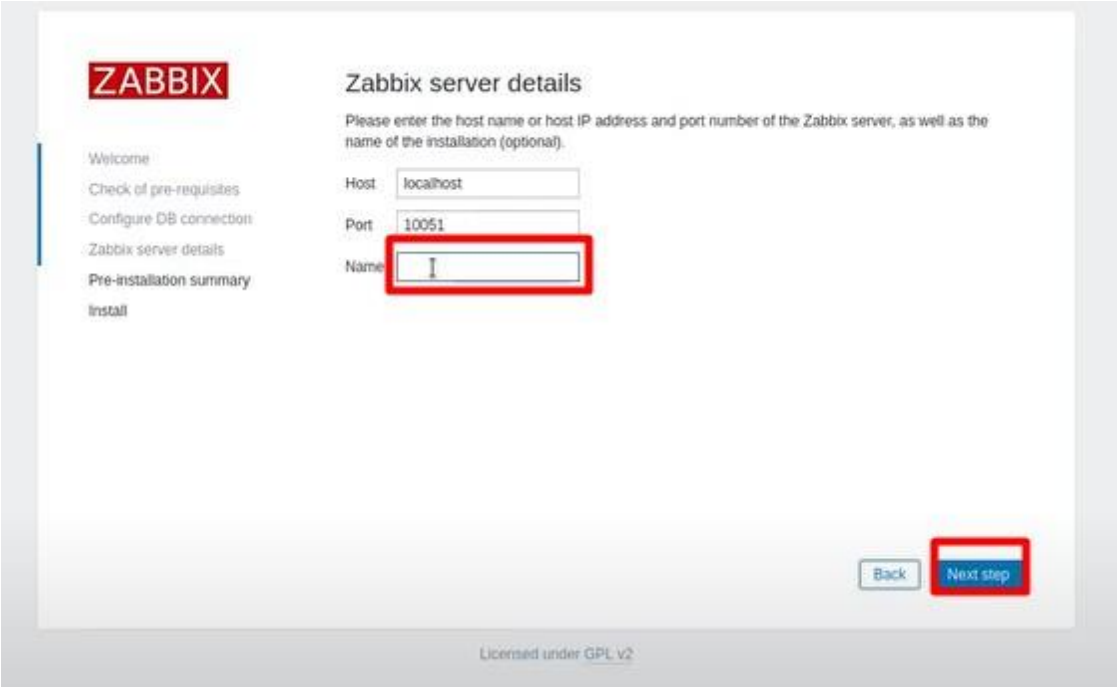

Рисунок 3.24 – Поле для IP-адреси

Крок 21 – перевірка введених даних і натиснути "Next step" (рис. 3.25).

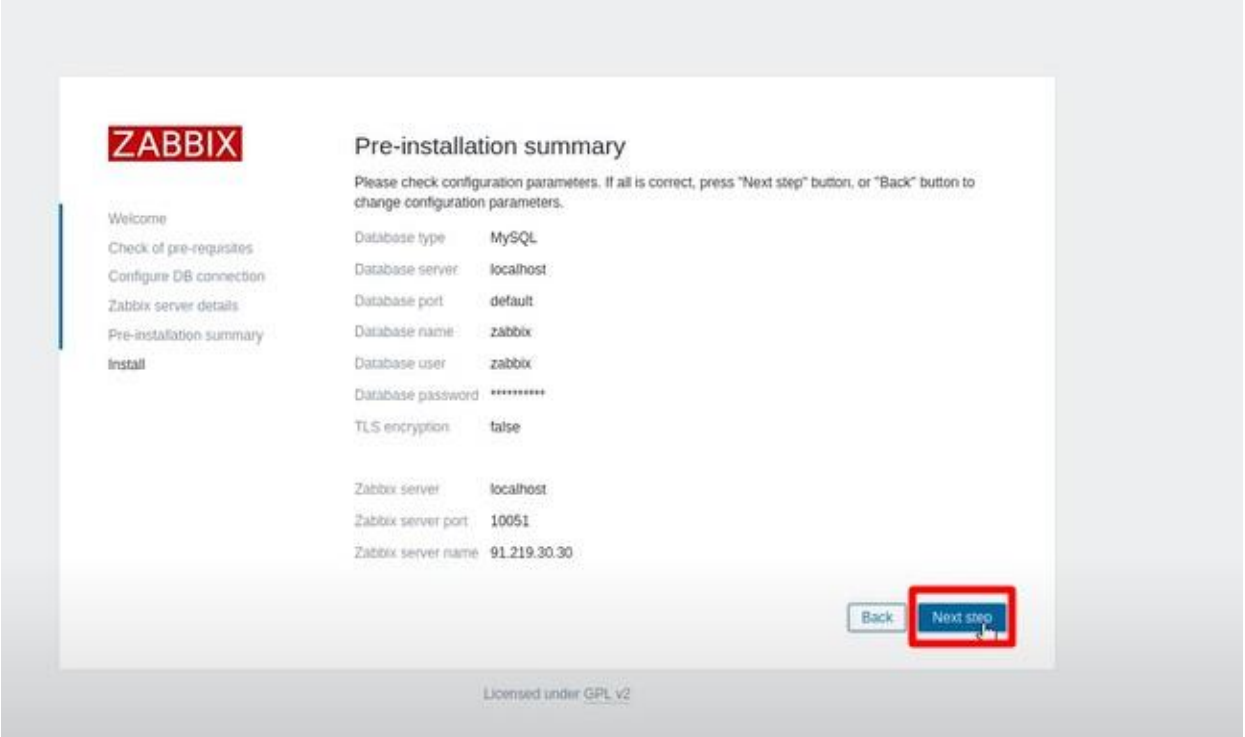

Рисунок 3.25 – Всі введені дані

Крок 22 – завершальний етап інсталяції, треба натиснути "Finish" (рис. 3.26).

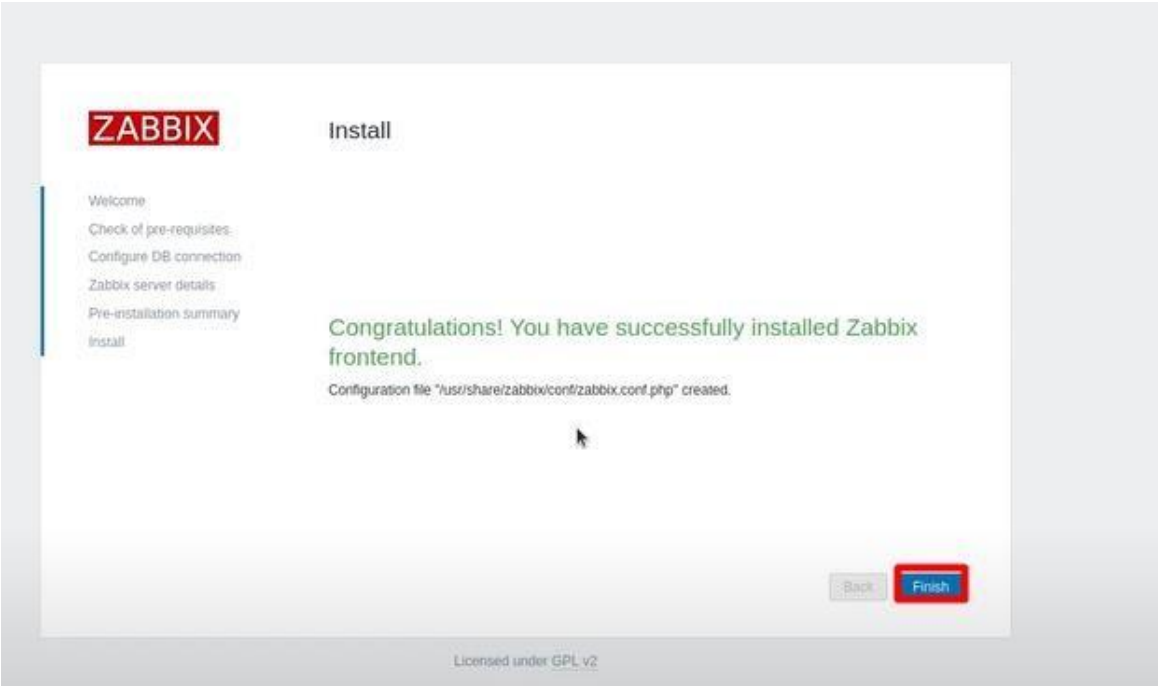

Рисунок 3.26 – Місце знаходження кнопки "Finish"

Крок 23 – вводиться ім’я користувача (Admin, за замовчуванням) та пароль, натискаю "Sign in" (рис. 3.27).

Рисунок 3.27 – Вікно входу у обліковий запис користувача

Встановлення та налаштування Zabbix успішно завершене. Після входу в обліковий запис користувача в головному меню можна побачити кількість підключених пристроїв "Number of hosts" (рис. 3.28). Щоб оглянути детальну інформацію про сервер потрібно натиснути "Hosts"(рис. 3.28).

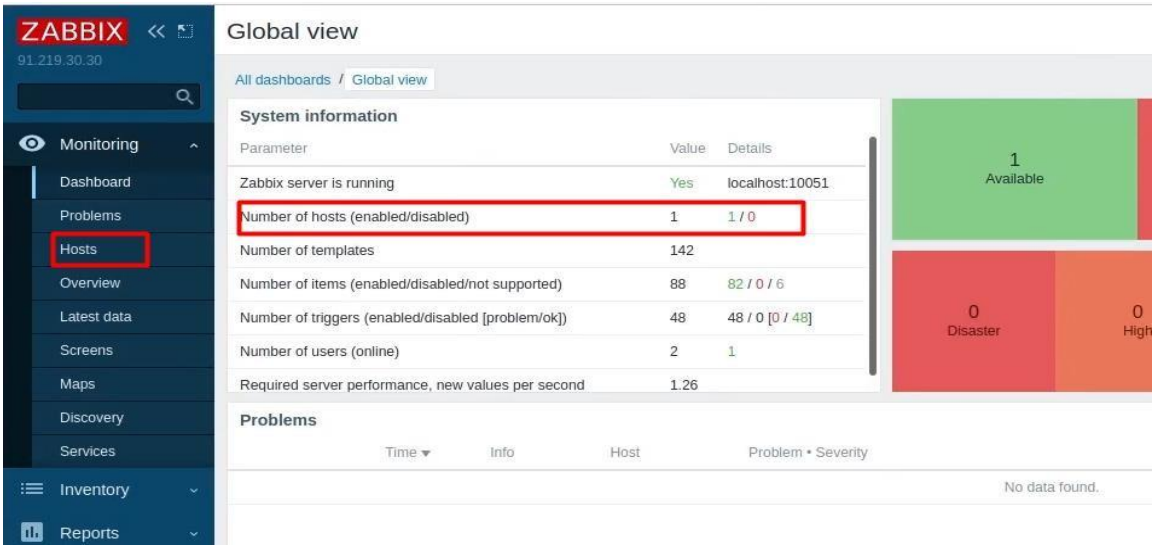

Рисунок 3.28 – Головне меню Zabbix

На рис. 3.29 у розгорнутій вкладці "Hosts" виводяться всі пристрої.

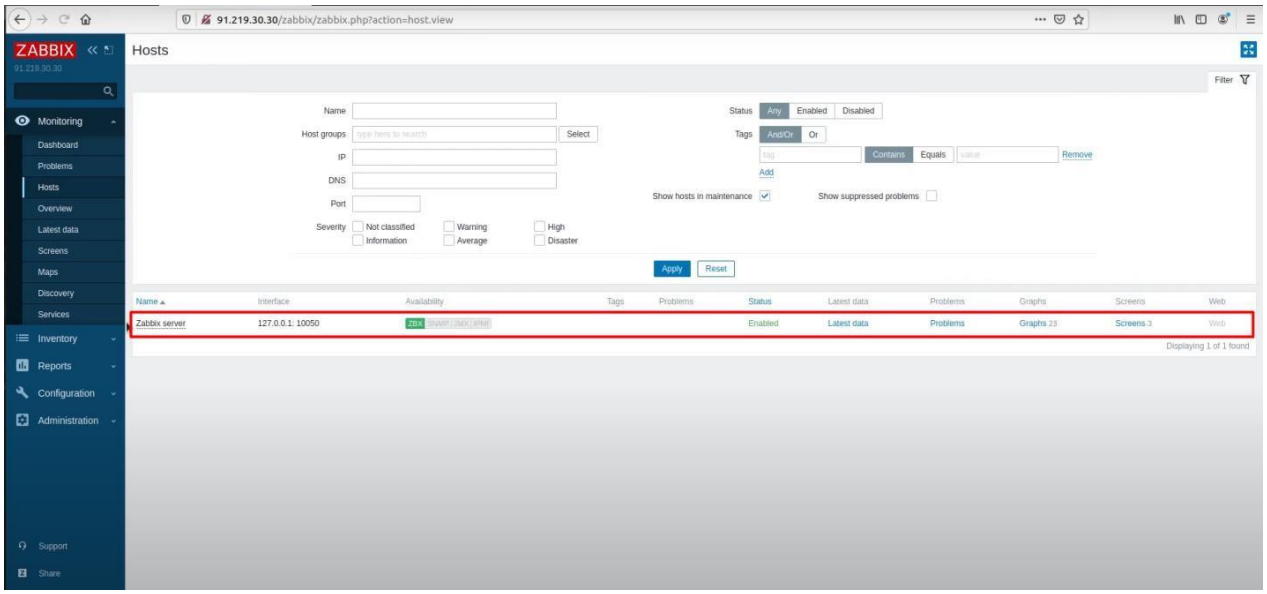

Рисунок 3.29 – Налаштований сервер Zabbix

На рис. 3.30 графік роботи встановленого сервера.

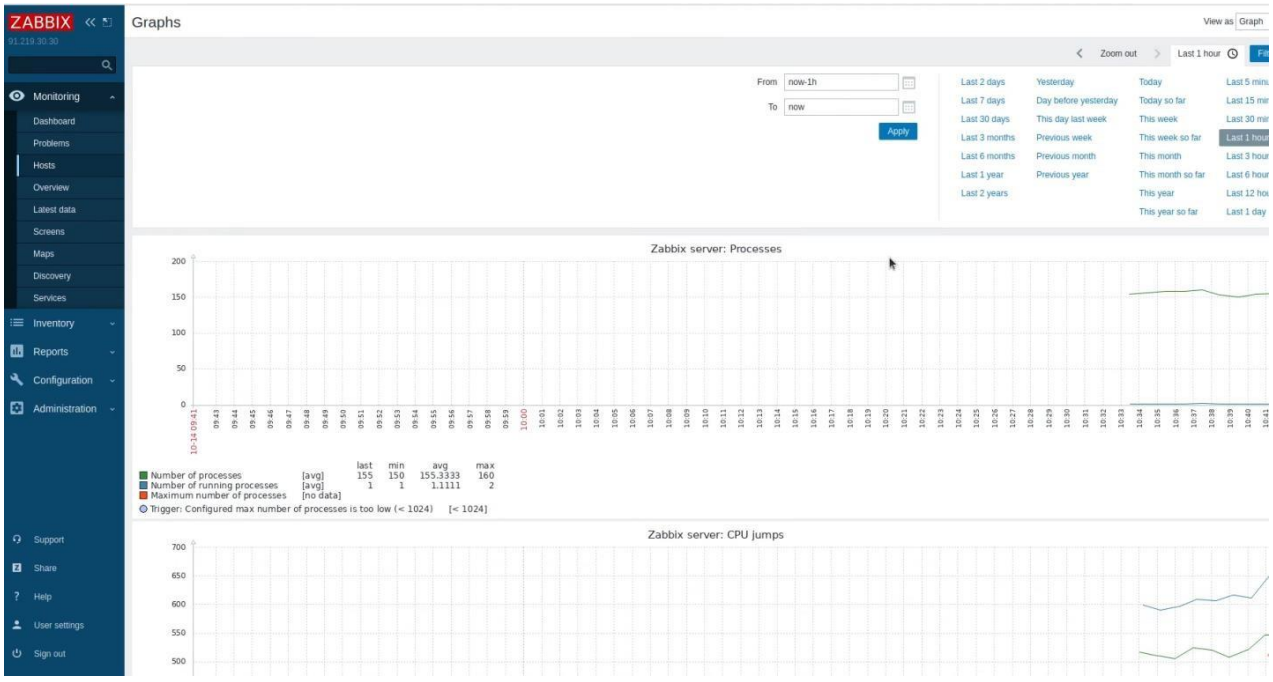

Рисунок 3.30 – Графік роботи сервера Zabbix

Висновки. Великий об’єм оперативної пам’яті дозволяє Zabbix швидко обробляти дані, які потім зберігаються у базах даних.

Просте і надійне рішення моніторингу – Zabbix, має простий процес встановлення та налаштування. Після швидкого встановлення можна одразу приступати до моніторингу та аналізу підконтрольного обладнання.

ВИСНОВКИ

Щоб забезпечити безпомилкову працездатність IT-інфраструктури необхідно заздалегідь виявляти слабкі місця в конфігурації систем і мереж, а також швидко дізнаватися про наявні поломки і їх причини. В організаціях це все досягається за рахунок систем моніторингу.

В роботі проведено аналіз низки систем моніторингу різних розробників та здійснено їх порівняльну характеристику.

Системи моніторингу бувають вартісні та вільно поширювані, а також розрізняються за своїм функціоналом. При плануванні, дизайні та впровадженні систем моніторингу спочатку потрібно визначити об’єкти моніторингу, а також критичні події та показники, які будуть визначати кількість повідомлень при помилках, частоту сканування і алгоритм роботи при виявленні цих помилок та інші параметри. Оцінювання показників в першу чергу потрібно здійснювати не з точки зору системного адміністратора, а з точки зору кінцевого користувача.

Використання протоколу SNMP дозволяє отримувати точні дані від пристроїв і керувати ними в автоматичному режимі, тестувати і активно використовувати в повсякденних задачах моніторінга IT-ресурсів мережі. Використання Zabbix забезпечує своєчасне виявлення помилок, що полегшує, автоматизує та підвищить ефективність роботи мережевих адміністраторів, а також допоможе уникнути зайвих витрат на придбання нового обладнання.

Необхідно зазначити, що Zabbix забезпечить спостереження за IT-інфраструктурою, в автоматичному режимі здійснить оцінку стану IT-інфраструктури та значно полегшить прогнозування наслідків неполадок.

ПЕРЕЛІК ДЖЕРЕЛ

1. Matti K. Network monitoring with Zabbix / Matti Koivisto, 2015. – 71 с.

2. Jackie F. Key Trends to Watch in Gartner Emerging Technologies Hype Cycle / Jackie Fenn, 2012 – 255 с.

3. Turnbull J. The Art of Monitoring / James Turnbull, 2016. – 645 с.

4. Ligus S. Effective Monitoring and Alerting: For Web Operations / Slawek Ligus, 2012. – 166 с.

5. Julian M. Practical Monitoring / Mike Julian, 2017. – 229 с.

6. Brazil B. Prometheus: Up & Running: Infrastructure and Application Performance Monitoring / Brian Brazil, 2018. – 563 с.

7. Liefting N. Zabbix 5 IT Infrastructure Monitoring Cookbook: Explore the new features of Zabbix 5 for designing, building, and maintaining your Zabbix setup / Nathan Liefting, 2021. – 464 с.

8. Davis J. Modern System Administration: Building and Maintaining Reliable Systems / J. Davis, C. Devers, T. Sable, 2021. – 300 с.

9. Eckerson W. Performance Dashboards: Measuring, Monitoring, and Managing Your Business / Wayne Eckerson, 2010. – 379 с.

10. Система моніторингу IT-інфраструктури [Електронний ресурс] – Режим доступу до ресурсу: https://cutt.ly/9kcnQc5.

11. Засоби моніторингу IT-інфраструктури [Електронний ресурс] – Режим доступу до ресурсу: https://www.osp.ru/os/2015/04/13047967.

12. Можливості Zabbix [Електронний ресурс] – Режим доступу до ресурсу: https://www.zabbix.com/documentation/2.2/ru/manual/introduction/features.

13 Використання Zabbix для моніторинга [Електронний ресурс] – Режим доступу до ресурсу: https://www.it-lite.ru/blog/iaas/zabbix-dlya-monitoringa.

14. Принцип роботи SNMP [Електронний ресурс] – Режим доступу до ресурсу: https://www.site24x7.com/network/what-is-snmp.html.

15. IPMI перевірки [Електронний ресурс] – Режим доступу до ресурсу: https://www.zabbix.com/documentation/4.0/manual/config/items/itemtypes/ipmi.

16. SNMP: розуміння простого протоколу управління мережею [Електронний ресурс] – Режим доступу до ресурсу: https://www.kaseya.com/blog/2020/09/14/snmp-simple-network-management-protocol.

17. Простий протокол управління мережею (SNMP) [Електронний ресурс] Режим доступу до ресурсу: https://searchnetworking.techtarget.com/definition/SNMP.

18. Управління IT – інфраструктурою [Електронний ресурс] – Режим доступу до ресурсу: https://acts.kpi.ua/uk/upravlinnya-it-infrastrukturoyu.

19. Аналіз існуючих систем моніторинга [Електронний ресурс] – Режим доступу до ресурсу: https://documentbase.net/744175.

20. Wojciech K. Learning Nagios / W. Kocjan, P. Beltowski. – Packt Publishing, 2016. – 414 с.