Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: «Розробка програмного забезпечення для моніторингу показників якості корпоративних мереж»

Виконав: студент ___IV___ курсу
групи ____4ПІ-20б____
спеціальності
___121 – Інженерія програмного забезпечення___
(шифр і назва напряму підготовки, спеціальності)
_____Зварич В. М._____
(прізвище та ініціали)
Керівник: к.т.н., доц. каф. ПЗ Коваленко О. О.
(прізвище та ініціали)
Рецензент: к.т.н., доц. каф. ОТ Крупельницький Л.В.
(прізвище та ініціали)

Допущено до захисту
Зав. кафедри _____
«_10_» __червня_____ 2024 р.

ВНТУ – 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
«_12_» __травня__ 2024 р.

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Зваричу Василю Миколайовичу

1. Тема роботи – «Розробка програмного забезпечення для моніторингу показників якості корпоративних мереж».

Керівник роботи: Коваленко Олена Олексіївна, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від 11 березня 2024 року №80.

2. Термін подання студентом роботи: 3 червня 2024 року.

3. Вихідні дані до роботи: середовище розробки – Zabbix. Мови програмування, сумісні з середовищем. Розподілені системи моніторингу

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз стану інструментарію моніторингу комп'ютерних мереж, постановка задач дослідження; розробка методу, моделі та алгоритмів роботи системи; розробка програмних модулів програми; тестування програми; висновки; список використаних джерел; додатки; ілюстративна частина.

5. Перелік ілюстративного матеріалу: титульний слайд – автор, науковий керівник бакалаврської дипломної роботи, тема; мета, об'єкт та предмет дослідження; задачі дослідження; новизна; практична цінність отриманих

результатів; результати розробки; апробація та публікації результатів роботи;
фінальний слайд.

6. Консультанти розділів бакалаврської дипломної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1 – 4	Коваленко О. О., к.т.н., доцент кафедри ПЗ	13.03.24	03.06.24

7. Дата видачі завдання 13 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітки
1	Аналіз розвитку програмного забезпечення моніторингу комп'ютерних мереж	14.03.2024 – 31.03.2024	
2	Розробка методу, моделі та алгоритмів роботи ПЗ	01.04.2024– 15.04.2024	
3	Розробка програмних модулів	16.04.2024 – 18.05.2024	
4	Тестування програми	19.05.2024 – 24.05.2024	
5	Оформлення матеріалів до захисту БДР	25.05.2024 – 30.05.2024	

Студент _____ (підпис) **Зварич В.М.** (прізвище та ініціали)

Керівник бакалаврської дипломної роботи _____ (підпис) **Коваленко О.О.** (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 50 сторінок формату А4, на яких є 32 рисунки, 1 таблицю, список використаних джерел містить 20 найменувань.

У бакалаврській дипломній роботі було розроблено програмні модулі для системи моніторингу показників якості комп'ютерних мереж.

Розроблені програмні модулі дозволяють працювати в середовищі моніторингу, отримувати візуальні показники статики та динаміки роботи комп'ютерних мереж.

Використання застосунку дозволяє підвищити продуктивність роботи мережевих інженерів.

Застосунок реалізовано з використанням мови програмування Python та PHP.

У результаті виконання бакалаврської дипломної роботи було створено програмне середовище для моніторингу показників якості комп'ютерних мереж.

Отримані в бакалаврській дипломній роботі результати можна використати для роботи з мережевим обладнанням різних організацій.

Ключові слова: програмне забезпечення, комп'ютерні мережі, показники якості, моніторинг комп'ютерних мереж.

ABSTRACT

The bachelor thesis consists of 50 pages of A4 format, on which there are 22 figures, 1 table, the list of used sources contains 20 items.

In the bachelor thesis, software modules were developed for the system of monitoring quality indicators of computer networks.

The developed software modules allow working in a monitoring environment, obtaining visual indicators of the statics and dynamics of computer networks.

Using the application allows you to increase the productivity of network engineers.

The application is implemented using the Python programming language.

As a result of the bachelor thesis, a software environment was created for monitoring the quality indicators of computer networks.

The results obtained in the bachelor thesis can be used to work with network equipment of various organizations.

Keywords: software, computer networks, quality indicators, computer network monitoring.

ЗМІСТ

ВСТУП

Обґрунтування вибору теми дослідження. Серед різноманітного програмного забезпечення можна виділити керуюче програмне забезпечення. Програмне забезпечення управління, також відоме як менеджерське програмне забезпечення, - це набір програм, які використовуються для контролю різних аспектів комп'ютерної системи або мережі. Воно включає в себе різні інструменти для моніторингу, аналізу, керування, налаштування та автоматизації різних процесів, що відбуваються у системі [1].

Це програмне забезпечення використовується для управління різними аспектами комп'ютерних мереж, включаючи моніторинг трафіку, контроль доступу, налаштування мережевих пристроїв та інше. Також його можна використовувати для управління серверами, зокрема управління потоком даних, контроль доступу, моніторинг дискового простору та інші функції.

Програмне забезпечення управління може бути створене для виконання різноманітних завдань, таких як керування хмарними послугами, віртуалізація, управління базами даних, аналіз даних та інші функції. Важливою функцією такого програмного забезпечення є забезпечення безпеки та захисту даних у системі. Таке програмне забезпечення використовується для моніторингу та контролю функціонування комп’ютерних мереж.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою роботи є розробка програмного забезпечення для моніторингу показників якості функціонування комп’ютерних мереж.

Відповідно до визначеної мети, сформовані основні завдання дослідження.

Завдання дослідження:

– аналіз стану програмного забезпечення для моніторингу комп’ютерних мереж;

– розробка методів, моделей моніторингу та контролю комп’ютерних мереж;

– розробка алгоритмів моніторингу та контролю комп’ютерних мереж ;

– створення програмного забезпечення для моніторингу функціонування комп’ютерної мережі;

– тестування модулів програмного забезпечення моніторингу та контролю мережі.

Об'єктом дослідження виступає процес розробки програмного забезпечення для моніторингу та контролю комп’ютерної мережі.

Предметом дослідження є методи моніторингу та контролю комп’ютерних мереж та їх автоматизація для створення програмного забезпечення.

Методи дослідження. У дослідженні було використано методи аналізу, синтезу, моделювання, алгоритмізації; програмування, тестування.

Новизна отриманих результатів:

Подальшого розвитку набув метод моніторингу та візуалізації показників якості комп’ютерних мереж, який, на відміну від існуючих дозволяє сформувати робочий простір моніторингу за визначеними показниками та динамічними графіками змін в мережі, що дозволяє покращити продуктивність роботи мережевого інженера.

Практична цінність отриманих результатів. Практичне значення полягає у використанні розробленого програмного забезпечення для автоматизованої системи моніторингу та контролю комп’ютерної мережі. Розроблений модуль може бути використаний для корпоративних комп’ютерних мереж, зокрема для комп’ютерної мережі університету.

Практичне впровадження буде виконано на прикладі комп’ютерної мережі ВНТУ.

У першому розділі детально розглядається сучасний стан систем моніторингу та контролю комп’ютерних мереж.

Другий розділ присвячений розробці методів та моделей моніторингу та контролю комп’ютерних мереж.

Третій розділ містить основні алгоритми та особливості сучасних технологічних рішень.

Четвертий розділ містить опис процесу тестування системи, яке дозволило виявити та усунути потенційні помилки і недоліки.

Особистий внесок здобувача. Усі наукові результати, викладені у бакалаврській дипломній роботі, отримані автором особисто. У друкованій роботі [2], опублікованій у співавторстві, автору належать такі результати: розробка моделі програмного забезпечення моніторингу комп’ютерної мережі.

Апробація матеріалів бакалаврської дипломної роботи. Матеріали бакалаврської дипломної роботи доповідалися на конференції: Інформаційне суспільство, Тернопіль.

Публікації. Результати дослідження були опубліковані в матеріалах конференції Інформаційне суспільство, Тернопіль [2].

Структура та обсяг роботи. Бакалаврська дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, що містить 20 найменувань, 4 додатки. Робота містить 22 ілюстрації та 1 таблицю.

1 АНАЛІЗ РОЗВИТКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОНІТОРИНГУ ПОКАЗНИКІВ ЯКОСТІ КОРПОРАТИВНИХ МЕРЕЖ

1.1 Особливості систем моніторингу комп'ютерних мереж

Системи моніторингу комп'ютерних мереж відносяться до керуючого програмного забезпечення. Програмне забезпечення для керування – це набір програмних засобів, які використовуються для керування різними аспектами комп'ютерної системи чи мережі. Такий комплекс може включати різні інструменти для моніторингу, аналізу, управління, налаштування та автоматизації різних процесів, що відбуваються в системі [1].

Програмне забезпечення для керування можна використовувати для керування різними аспектами комп'ютерної мережі, включаючи моніторинг трафіку, контроль доступу, налаштування мережевих пристроїв тощо. Його також можна використовувати для керування серверами, включаючи контроль потоку даних, контроль доступу, моніторинг дискового простору та інші функції. Програмне забезпечення для керування може бути розроблено для виконання різноманітних завдань, включаючи керування хмарними службами, віртуалізацію, керування базами даних, аналіз даних та багато інших. Важливою функцією керуючого програмного забезпечення є забезпечення безпеки та захисту даних у системі.

Моніторинг комп'ютерної мережі – це процес постійного моніторингу комп'ютерної мережі на наявність повільних або несправних компонентів і перевірка стану індикаторів, особливо індикаторів якості обслуговування. Цей процес включає сповіщення адміністратора мережі (через електронну пошту, текстове повідомлення чи інше сповіщення) про несправність або іншу проблему. Моніторинг мережі є частиною керування мережею [2].

Системи моніторингу мережі відстежують проблеми мережі, спричинені перевантаженням або забоєм серверів, мережевих з'єднань або інших пристроїв.

Загальні вимірювання включають час відповіді, доступність і безвідмовну роботу. Показники узгодженості та надійності також стають входять в список параметрів вимірювання ефективності та продуктивності роботи мережі. Розповсюдження інструментів оптимізації WAN негативно вплинуло на більшість інструментів моніторингу мережі, особливо коли справа доходить до вимірювання точної наскрізної затримки, оскільки вони обмежують можливість тестування затримки в обидва кінці.

Моніторинг мережі означає використання програмного забезпечення для моніторингу мережі для моніторингу стану та надійності комп'ютерної мережі . Системи моніторингу продуктивності мережі (NPM) зазвичай генерують топологічні карти та корисну інформацію на основі зібраних і проаналізованих даних про продуктивність.

У результаті цього відображення мережі IT-команди отримують повну видимість мережевих компонентів, моніторингу продуктивності додатків і відповідної IT-інфраструктури . Ця видимість дозволяє їм відстежувати загальний стан мережі, виявляти червоні прапорці та оптимізувати потік даних.

Система моніторингу мережі стежить за несправними мережевими пристроями та перевантаженими ресурсами. Система робить це незалежно від того, чи є мережеві ресурси локальними, у центрі обробки даних, у постачальника хмарних послуг чи є частиною гібридної екосистеми. Наприклад, він може виявити перевантажені процесори на серверах, високий рівень помилок на комутаторах і маршрутизаторах або раптові стрибки мережевого трафіку. Ключовою особливістю програмного забезпечення NPM є сповіщення мережевих адміністраторів про виявлення проблеми з продуктивністю.

Системи моніторингу мережі також збирають дані для аналізу потоку трафіку та вимірювання продуктивності та доступності. Одним із методів моніторингу проблем продуктивності та вузьких місць є налаштування порогових значень, щоб ви отримували миттєві сповіщення про порушення

порогових значень. Деякі пороги є простими статичними порогами. Однак сучасні системи NPM використовують машинне навчання (ML) для визначення нормальної продуктивності всіх показників мережі на основі часу доби та дня тижня. Системи NPM із такими керованими базовими показниками ML створюють сповіщення, які зазвичай є більш ефективними.

Організації часто здійснюють моніторинг мережі в центрі мережевих операцій (NOC), який відстежує мережеві пристрої та з'єднання між усіма залежностями, не знаючи кінцевих користувачів про те, що NOC працює за лаштунками [3].

Збої в мережі можуть призвести до збоїв у бізнесі, що може означати втрату клієнтів, продуктивності праці та грошей. Насправді 91% середніх і великих підприємств повідомляють, що одна година простою IT коштує їм щонайменше 300 000 доларів США на годину. Майже половина цієї групи сказала, що це коштує мільйони.

Інвестування в програмне забезпечення для моніторингу мережі, будь то комерційне чи з відкритим вихідним кодом, означає застосування проактивного підходу до підтримки працездатності вашої мережевої інфраструктури та максимального збільшення часу безвідмовної роботи. Цей підхід запобігає очікуванню, поки кінцевий користувач повідомить про проблеми з мережею.

Моніторинг інфраструктури дає вам змогу точно знати, де виникає мережева проблема, дозволяючи усунути несправності до того, як ситуація призведе до збою. Раннє виявлення та вирішення першопричини проблеми може скоротити час реагування, підвищити рівень задоволеності клієнтів, заощадити гроші та захистити репутацію компанії.

Типи засобів моніторингу мережі

Існує три основні типи інструментів моніторингу мережі.

1. Інструменти на основі SNMP використовують простий протокол керування мережею (SNMP) для взаємодії з мережевим обладнанням. Ці інструменти відстежують у реальному часі стан і використання ресурсів,

наприклад статистику ЦП, споживання пам’яті, передані й отримані байти та інші показники. SNMP є одним із найпоширеніших протоколів моніторингу разом із Microsoft Windows Management Instrumentation (WMI) для серверів Windows і Secure Shell (SSH) для серверів Unix і Linux.

2. Інструменти на основі потоку відстежують потік трафіку, щоб надати статистику щодо протоколів і користувачів. Деякі також перевіряють послідовності пакетів, щоб виявити проблеми з продуктивністю між двома IP-адресами. Ці інструменти потоку збирають дані про рух транспорту та надсилають їх до центрального колектора для обробки та зберігання.

3. Рішення для активного моніторингу мережі вводять пакети в мережу та вимірюють наскрізну доступність, час проходження в обидві сторони, пропускну здатність, втрату пакетів, використання каналу тощо. Проводячи та вимірюючи транзакції в реальному часі з точки зору користувача, ці рішення дозволяють швидше та надійніше виявляти збої та погіршення продуктивності [5].

Існують також як агентний, так і безагентний методи моніторингу мережі. Моніторинг за допомогою агента передбачає встановлення агента, невеликої програми або частини програмного забезпечення, на контрольований пристрій. Безагентний моніторинг (з використанням протоколів SNMP і SSH) не потребує інсталяції; натомість програмне забезпечення для моніторингу мережі входить безпосередньо в контрольований пристрій.

Під час використання інструменту моніторингу мережі першим кроком є визначення мережевих пристроїв для моніторингу та встановлення показників продуктивності. Потім виберають інтервал моніторингу, який відповідає вашій ситуації. Моніторинг маршрутизаторів, комутаторів і серверів зазвичай виконується частіше, оскільки такі мережеві пристрої важливіші для бізнесу.

Після встановлення інструменти моніторингу мережі сканують проблеми з мережею. Методи можуть бути такими ж простими, як ping, щоб

переконатися, що хост доступний. Вони також можуть бути ширшими, наприклад моніторинг доступу до брандмауера, використання пропускної здатності, споживання ресурсів, час безвідмовної роботи та неочікувані зміни в мережевому трафіку. Ці інструменти гарантують, що комутатори, маршрутизатори, сервери, брандмауери та інші кінцеві точки мають прийнятний рівень пропускної здатності. Інструменти також виконують балансування навантаження та відстежують високий рівень помилок.

Ці інструменти пропонують візуалізацію всієї мережевої інфраструктури за допомогою настроюваних інформаційних панелей. Інформаційні панелі надають графіки продуктивності в режимі реального часу та інші звіти, які показують, як виглядають компоненти та чи є незвичні параметри, які потребують подальшого дослідження.

Рішення для моніторингу мережі надсилають сповіщення електронною поштою або SMS адміністраторам мережі, коли вони виявляють проблеми, які потребують уваги. Вони також обмінюються сповіщеннями про сповіщення з різними операційними IT-інструментами, такими як системи AIOps.

Основна перевага інструментів моніторингу мережі полягає в простому та легкому для розуміння перегляду підключених пристроїв усієї мережі та способів переміщення даних між ними. Сучасні системи моніторингу продуктивності мережі надають базову інформацію, яка дозволяє автоматично порівнювати дані та виявляти будь-яке зниження продуктивності мережі. Перегляд усієї цієї інформації на інформаційній панелі означає, що ваші мережеві адміністратори можуть швидко виявити проблеми в будь-якій точці мережі, визначити першопричину та визначити, хто повинен її виправити.

Рішення NPM вимагає менше часу для вирішення проблем продуктивності мережі. Раніше виявлення проблеми означає, що ви зможете усунути несправність і виправити її набагато швидше, заощадивши час і гроші. Моніторинг мережі допомагає підприємствам оптимізувати ефективність, виявляючи та усуваючи проблеми до того, як вони вплинуть на роботу та

клієнтів. Це, у свою чергу, зменшує час простою, гарантує, що співробітники завжди мають доступ до необхідних ресурсів, і підвищує доступність API і веб-сторінок, щоб клієнти мали доступ, коли їм це потрібно. Моніторинг продуктивності мережі також надає історичні дані та дозволяє усунути минулі проблеми з мережею, щоб можна було уникнути подібних проблем у майбутньому [6].

Рішення для моніторингу мережі надають вам надійні та гнучкі інструменти та можливості керування, включаючи попередньо налаштовані шаблони для конкретних постачальників, таких як Cisco, Juniper, Arista та Aruba, серед інших. Ці шаблони допомагають переконатися, що все працює добре. Вони допоможуть вам дотримуватися галузевих стандартів і державних постанов. Вони можуть вказати на аномалії, які можуть бути першими ознаками кібератаки. Інструменти моніторингу також можуть допомогти відстежувати та порівнювати показники продуктивності мережі, що допомагає встановлювати цілі та покращувати продуктивність мережі.

Сучасні системи моніторингу мережі йдуть далі, підтримуючи програмно визначені мережі нового покоління, розгорнуті для запуску сучасних програм. Такі нові, сучасні мережі вимагають зовсім інших способів збору та аналізу даних про продуктивність за допомогою API, а також підтримують існуючі методи, такі як SNMP. Підтримка як нових, так і існуючих мереж допомагає мережевим операторам і командам інженерів правильно планувати перехід до мереж наступного покоління.

Моніторинг продуктивності мережі також дозволяє відстежувати мережі, які з часом змінюються, ростуть і стають складнішими. З появою пристроїв IoT, які дозволяють мережам об’єднувати тисячі пристроїв IoT або більше, IT-мережі стають величезними та складними.

Моніторинг мережі – це не те саме, що моніторинг безпеки мережі . Організації використовують моніторинг безпеки мережі, щоб захистити мережу від несанкціонованого доступу, неправильного використання та крадіжки. Навпаки, рішення для моніторингу мережі в основному

використовується для оптимізації доступності мережі та загальної продуктивності.

Хоча безпека не є основною метою моніторингу мережі, найкращі інструменти моніторингу продуктивності мережі можуть допомогти дізнатися, чи пристрої безпеки функціонують ефективно. Деякі інструменти моніторингу мережі включають програмне забезпечення для керування конфігурацією, яке допомагає підвищити безпеку та надійність мережі, пропонуючи повністю автоматизовані конфігурації для мережевих пристроїв та інтерфейсів.

Знаючи, як працює ваша мережа, можна привернути увагу до перших ознак компрометації чи атаки. Коли програмне забезпечення для моніторингу мережі вказує на аномалії продуктивності, IT-команда може легше виявляти мережеві загрози та усунути порушення даних та інші атаки.

Сучасні мережеві інфраструктури, створені для цифрової трансформації, вимагають рішень, які можуть бути такими ж динамічними, гнучкими та масштабованими, як і нові середовища. IT-лідери дотримуються вищих стандартів, поєднуючи постійну продуктивність і автоматизацію з орієнтованими на бізнес угодами SLA.

Отже, розробка програмного забезпечення для моніторингу комп’ютерних мереж є актуальною та може бути використана мережевими інженерами в практичній діяльності.

1.2 Порівняльний аналіз аналогів

Програми моніторингу мережі – незамінний помічник кожного системного адміністратора. Вони дозволяють швидко реагувати на незвичні дії в локальній мережі, отримувати видимість усіх мережевих процесів і автоматизувати частини повсякденної діяльності адміністратора: перш за все ті, що пов’язані із забезпеченням безпеки мережі.

Розглянемо основні аналоги застосунку для моніторингу та контролю роботи комп’ютерної мережі.

Olympus Networks – конструктор скриптів, який дозволяє позбутися примітивних перевірок, які виконують певні ситуації, які не дозволяють враховувати роботу пристрою (див. рисунок 1.1). З його допомогою можна організувати сценарії моніторингу будь-якої складності для точного виявлення проблем і несправностей і автоматизації процесу усунення. В основі скрипта лежить датчик, з якого можна побудувати логічний ланцюжок, який в залежності від успішності перевірки буде генерувати різні сповіщення та дії, призначені для вирішення вашого завдання. Кожен елемент у ланцюжку можна редагувати в будь-який час і буде негайно застосовано до всіх пристроїв, до яких прикріплено сценарій. Уся мережева активність буде відстежуватися за допомогою журналів активності та спеціальних звітів [7].

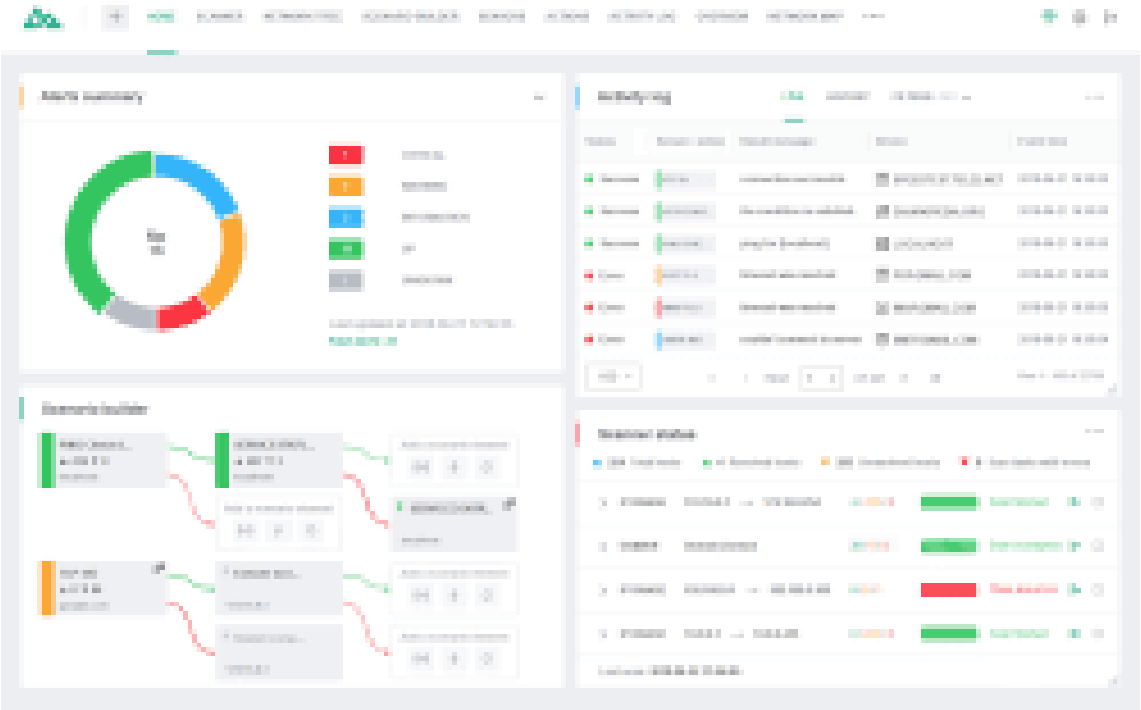


Рисунок 1.1 – Приклад роботи застосунку Olympus Networks

До переваг можна віднести можливості формування сценаріїв моніторингу, багатофункціональність, простоту налаштування, групові

сенсори. До недоліків можна віднести наявність тільки веб-інтерфейсу, робота тільки з Windows, відсутність багатокористувацького доступу.

Observium – застосунок, що використовує протокол SNMP, не тільки дозволяє в режимі реального часу досліджувати стан мережі будь-якого розміру, але й аналізує рівень її продуктивності. Рішення інтегрується з пристроями Cisco, Windows, Linux, HP, Juniper, Dell, FreeBSD, Brocade, Netscaler, NetApp та інших постачальників. Завдяки ідеально розробленому графічному інтерфейсу програма пропонує системним адміністраторам безліч параметрів налаштування – від автоматичного визначення областей до збору даних протоколу SNMP, необхідних для інформації про мережу [8].

Доступ до даних про технічні характеристики всіх пристроїв, підключених до мережі. Observium може представити всі звіти у вигляді діаграм і графіків, наочно показуючи «слабкі» сторони мережі. Застосунок має демоверсію.

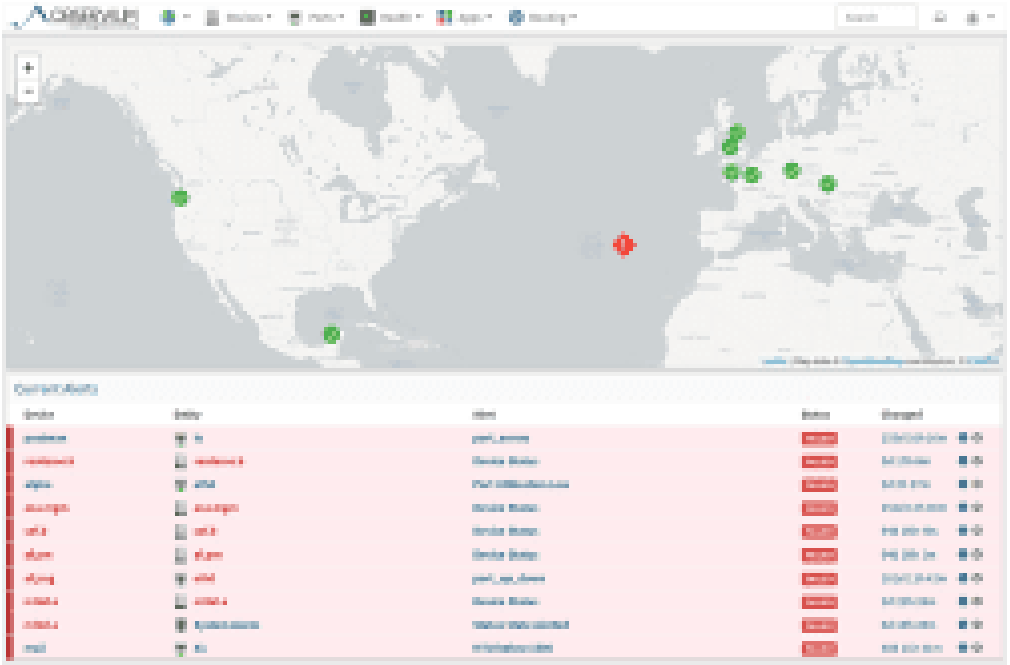


Рисунок 1.2 – Приклад роботи застосунку Observium

Перевагами є робота з багатьма операційними системами, комплекс граничних сигналів, функції автоматичного виявлення потреб в налаштуванні. Недоліками є складність установки, відсутність підтримки мобільних пристроїв, складні налаштування для великих мереж.

Nagios – передове веб-рішення для моніторингу. Завдяки значній інтернет-спільноті та вичерпній документації налаштування застосунку є складним, але доступним.

За допомогою Nagios системні адміністратори мають можливість дистанційно регулювати навантаження користувачів або пристроїв вищого рівня мережі (комутатори, маршрутизатори, сервери), стежити за рівнем завантаження резервів пам'яті в базах даних, а також контролювати фізичні показники компонентів мережевого обладнання (таких як материнські плати) температура, вигорання є однією з найпоширеніших поломок у цій галузі) тощо.

Коли виявляється мережева аномалія, Nagios автоматично надсилає попереджувальне повідомлення на адресу, попередньо встановлену адміністратором – або адресу електронної пошти, або номер телефону мобільного оператора [9].

Рисунок 1.3 – Приклад візуалізації роботи мережі за допомогою застосунку Nagios

До переваг можна віднести високу гнучкість, корисні шаблони, інтеграцію з іншими застосунками. До недоліків – трудомістке налаштування, особливо для великих мереж.

Ще одним рішенням для моніторингу мережі є застосунок Zabbix [10].

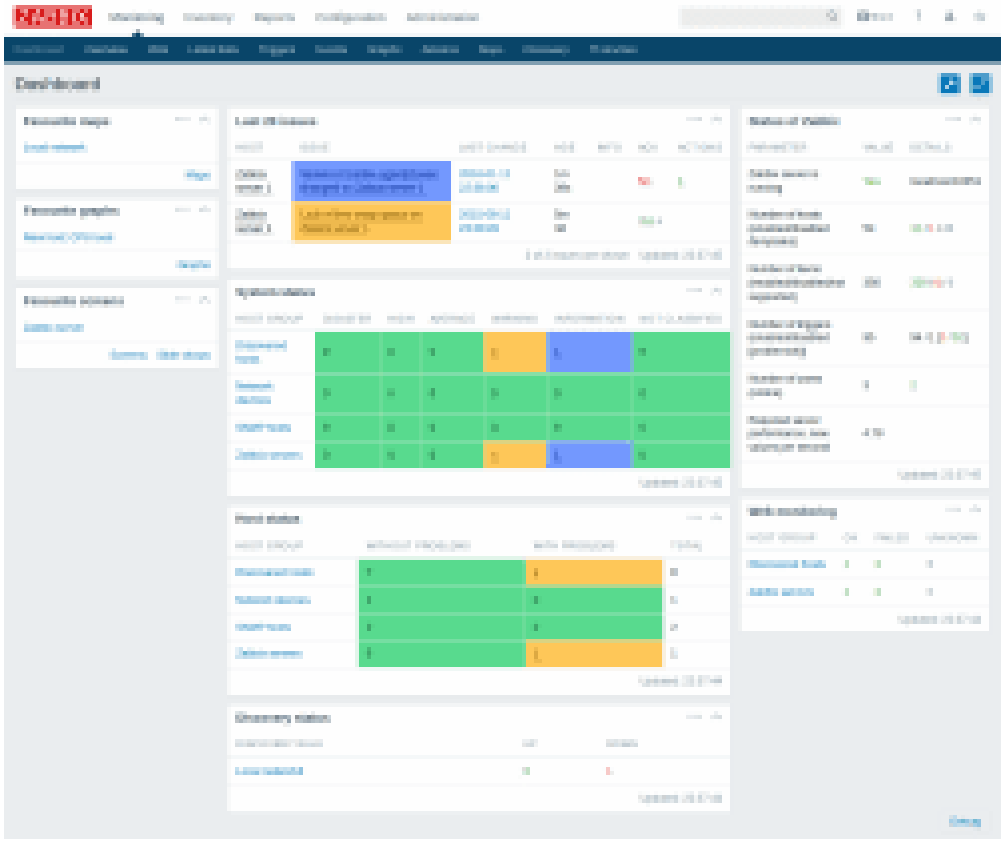


Рисунок 1.4 – Приклад моніторингу роботи мережі за допомогою застосунку Zabbix

Універсальне рішення для моніторингу мережі з відкритим вихідним кодом, яке можна налаштувати для окремих моделей мережі. Переважно він працює на системах з багатосерверною архітектурою (особливо Zabbix, інтегрований із серверами Linux/FreeBSD/Windows). Програма дозволяє одночасно керувати сотнями вузлів мережі, що робить його надзвичайно ефективним інструментом для організації роботи системних адміністраторів на великих підприємствах. Щоб розгорнути Zabbix у вашій локальній мережі, потрібно запустити програмний агент (демон) або використовувати протокол SNMP (або інший протокол для безпечного віддаленого доступу); для керування необхідно освоїти веб-інтерфейс PHP. Крім того, програмне

забезпечення надає повний набір інструментів для моніторингу стану частин мережевого обладнання. PHP, Python – мови, які можуть бути використані для удосконалення та адаптації застосунку для конкретної мережі.

Застосунок є безкоштовним, має багато плагінів, але має незручний інтерфейс, не має дашбордів.

Аналіз аналогів показав, що кожен з них має свої переваги та недоліки і для кожної окремої комп'ютерної мережі необхідно адаптувати використання застосунку. Найбільш прийнятним є останній застосунок, модернізація якого з подальшим розвитком системи оповіщення на сайти, в канали соціальних мереж дозволять отримати повне програмне рішення для моніторингу та контролю.

Таблиця 1.1 – Порівняльні характеристика систем моніторингу та контролю комп'ютерної мережі

Показники	Olympus Networks	Observium	Nagios	Zabbix	Власна розробка Zabbix +
Підтримка багатьох операційних систем	0	1	1	1	1
Функціонал для моніторингу та контролю	1	1	1	1	1
Функціонал для оповіщення	0	1	1	0	1
Багатокористувацький інтерфейс	0	0	1	1	1
Можливість адаптації за допомогою мов програмування	0	0	0	1	1
Сумарний коефіцієнт	1	3	4	4	5

Відповідно до наведеної таблиці порівняльного аналізу, найкращим є застосунок на основі програмного забезпечення Zabbix з подальшою

адаптацією та формуванням модулю оповіщення для системного адміністратора мережі.

Отже, розробка програмного забезпечення моніторингу та контролю роботи комп'ютерної мережі доцільно здійснювати на основі програмного забезпечення Zabbix.

1.3 Аналіз методів розв'язання задачі

Усі системи управління та моніторингу розроблені для зменшення впливу людського фактора на роботу IT-інфраструктури. Зрозуміло, що системні адміністратори повинні стежити за величезною кількістю інформації. Тому фахівець повинен бути достатньо кваліфікованим, щоб вирішити проблему. Слід розуміти, що системи моніторингу не можуть замінити кваліфікацію користувачів і адміністраторів. Це лише дає їм інформацію, необхідну для прийняття правильного рішення. Системи моніторингу повинні давати чіткі відповіді на питання щодо ефективності роботи підсистем IT-інфраструктури. При цьому СМ забезпечує подальший розвиток IT-інфраструктури у збалансованому та стратегічно правильному напрямку в міру зниження вартості наявних на об'єкті інформаційних технологій. Впровадження нових послуг і технологій вимагає певного рівня розвитку IT-інфраструктури. У більшості випадків впровадження нового програмного та апаратного забезпечення неможливе через «слабку» IT-інфраструктуру, а коли система впроваджена, вона не є повністю функціональною. Необхідно визначити ключові питання, пов'язані з управлінням інформацією, комунікаціями та моніторингом IT. Моніторинг серверного обладнання досягається шляхом визначення важливих показників продуктивності, своєчасного звітування системою та своєчасного усунення виявлених відхилень. Це особливо важливо через складність IT-процесів і пов'язані з ними ризики. Його мета - забезпечити інформаційну підтримку, яка допомагає організації досягати поставлених цілей і виконувати завдання серверного

обладнання. Основними об'єктами моніторингу є: сервери, мережеве обладнання, додатки, бази даних, станції користувача та виділені системи.

Основним завданням системи моніторингу є надання актуальної інформації для аналізу стану IT-інфраструктури та швидкого виявлення несправностей і своєчасного їх усунення. Системи моніторингу продуктивності дозволяють IT-фахівцям оперативно помічати зниження продуктивності та виявляти «вузькі місця» в IT-інфраструктурі [1].

Постійний моніторинг допомагає уникнути простоїв у роботі, підтримувати всі IT-сервіси в робочому стані та підтримувати необхідний рівень якості, планувати їх модернізацію. Раніше роль моніторингу виконували адміністратори, і інформація про стан системи в кращому випадку збиралася ними в будь-якій непрофесійній програмі (через їх відсутність), а в гіршому взагалі не накопичувалася і не агрегувалася. Вся інформація про систему пов'язана з реальним досвідом роботи конкретного фахівця з інфраструктурою і повністю втрачається під час його обслуговування. Планувальник завдань запускає спеціальний попередньо визначений сценарій, який повідомляє про успішне виконання операції або помилки, що виникли під час процесу. Для зберігання отриманої інформації зазвичай використовуються бази даних конфігурації під різними СУБД: Інформація про об'єкти моніторингу представлена у вигляді набору вуликів.

Кожен сервер, кожне мережеве пристрій – це одиниця, і все це зберігається в системі моніторингу у візуальному вигляді: діаграми, графіки тощо. З часом структура спостереження істотно змінилася. Наприклад, з появою і широким розповсюдженням віртуалізації з'явилася тонкість: якщо раніше потрібно було стежити лише за станом фізичних серверів, то тепер на кожному сервері може бути кілька віртуальних серверів. Систему моніторингу також можна налаштувати для виконання будь-якої стандартної сервісної операції. Вибираючи, розробляючи та впроваджуючи систему моніторингу, необхідно визначити, що будемо контролювати, а також ключові події та

індикатори, які визначатимуть кількість повідомлень, частоту сканування та статус у разі виникнення збою. Інші параметри та наслідки.

Для створення програмного застосунку управління мережею доцільно використовувати компонентний та агентний методи. Вони дозволять використовувати модульність, програмування та повторне використання компонентів. Абстрактні моделі, що включають в себе логічні та математичні алгоритми, дозволять сформувати загальну концепцію та сценарії моніторингу і контролю роботи мережі. Моделі поведінки дозволять генерувати деталі сценаріїв та інтерфейсів системи моніторингу. Саме тому для створення програмного застосунку доцільно використовувати об’єктно-орієнтований та компонентно-орієнтований методи з можливістю адаптації програмного застосунку до особливостей корпоративної локальної мережі.

1.4 Постановка задачі дослідження

Після аналізу сильних та слабких сторін існуючих систем моніторингу та контролю показників функціонування комп’ютерних мереж була сформована постановка задачі:

Ці завдання спрямовані на створення зручної системи моніторингу та контролю, яка може бути використана для корпоративної комп’ютерної мережі.

Завдання:

розробка методів, моделей моніторингу та контролю комп’ютерних мереж;

розробка алгоритмів моніторингу та контролю комп’ютерних мереж ;

створення програмного забезпечення для моніторингу функціонування комп’ютерної мережі;

тестування модулів програмного забезпечення моніторингу та контролю мережі.

1.5 Висновки до розділу 1

У першому розділі було проведено аналіз особливостей систем моніторингу та контролю мережі, проведено аналіз аналогів, визначені методи створення програмного забезпечення системи моніторингу та контролю мережі, виконана постановка задачі.

2 РОЗРОБКА МЕТОДІВ ТА МОДЕЛЕЙ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОНІТОРИНГУ КОМП'ЮТЕРНИХ МЕРЕЖ

2.1 Методи моніторингу та контролю комп'ютерної мережі

Методи моніторингу та контролю комп'ютерної мережі відповідають стандарту – угоді рівнів обслуговування SLA. Система збирає та зберігає інформацію про якість IT-послуг згідно SLA. На основі накопиченої інформації протягом певного часу формується звіт.

Системи моніторингу можуть бути спрямовані на різні рівні споживачів. Для великих систем часто використовується велика кількість різноманітних функцій, а для невеликих систем часто менше виконується спільний аналіз вузлів і відправка попереджень. Серед основних функцій моніторингу можна виділити наступні:

- Відстеження. Основні функції включають регулярний збір показників, таких як вузли пристроїв і служби.

- Зберігання інформації. Надбудова для відстеження. Інформація збирається на основі основних показників кожного об'єкта моніторингу та зазвичай зберігається в базі даних.

- Побудова звітів. Він базується як на поточних даних відстеження, так і на довгостроковому зберіганні.

Наприклад, тривалий моніторинг навантаження на сервер може попереджати про збільшення споживання ресурсів, що означає необхідність збільшення доступних ресурсів або перенесення частини завдань на інший сервер, вибір якого також можна зробити на основі довгострокова звітність[12].

- Візуалізація. Наочно представлені звіти: у вигляді діаграм, спливаючих підказок, діаграм. Допомагає легко сприймати інформацію, а також можлива можливість візуалізації кількох найважливіших показників, при цьому всі показники будуть представлені у звіті.

- Пошук вузьких місць. Аналізуючи дані моніторингу, можна дізнатися, де в інфраструктурі загальні показники продуктивності впали найбільше.

- Автоматизація сценаріїв. Ця функція звільняє адміністраторів від щоденних завдань. Завдяки програмному забезпеченню, яке забезпечує всі ці можливості, адміністраторам більше не потрібно вручну перевіряти стан кожного компонента системи, проблеми вирішуються та усуваються швидше, діагностика є багатовимірною та точною, і можна планувати розширення інфраструктури.

Використання систем моніторингу та управління дозволяє:

- Оптимізувати використання інформаційних ресурсів;

- Підвищити якість IT-послуг і швидкість усунення несправностей обладнання та програмного забезпечення, а також мінімізувати час простою сервісу;

- Забезпечити надійність і безпеку всіх компонентів Сприяти модернізації IT-інфраструктури.

Відповідно до рекомендацій ISO інструменти управління мережею можна розділити на такі функції:

Керування мережевою конфігурацією – включає конфігурацію мережевих компонентів, включаючи їх розташування, мережеві адреси та ідентифікатори, керування параметрами мережевої операційної системи, підтримку мережевих сценаріїв: ці функції також використовуються для іменованих об'єктів.

Обробка помилок – це виявлення та усунення наслідків збоїв мережі.

Аналіз продуктивності – допомагає оцінити час відгуку системи та трафік, а також спланувати розвиток мережі на основі накопиченої статистичної інформації.

Управління безпекою –включаючи контроль доступу та збереження цілісності даних. Функціональність включає процеси автентифікації, перевірку дозволів, підтримку ключа шифрування та керування дозволами. До

цієї ж групи входять важливі механізми для керування паролями, зовнішнім доступом і з’єднаннями з іншими мережами.

Облік мережі – включає реєстрацію та управління ресурсами та обладнанням, що використовуються. Ця функція працює з такими поняттями, як час використання та плата за ресурси.

Як видно з наведеного переліку, система управління виконує не тільки функції моніторингу та аналізу роботи мережі, які необхідні для отримання вихідних даних для налаштування мережі, але також включає функції, які активно впливають на мережу – налаштування і управління безпекою, яке є необхідним для розробки планів конфігурації мережі та оптимізації. Хоча деякі системи керування мають експертні підсистеми, які допомагають адміністраторам або інтеграторам визначити необхідні заходи конфігурації мережі, етап створення планів конфігурації мережі часто залишається за межами функціональності системи керування.

Засоби адміністрування системи зазвичай виконують такі функції:

Облік використаного обладнання та програмного забезпечення. Система автоматично збирає інформацію про комп’ютери та створює записи про апаратні та програмні ресурси в базі даних. Тоді адміністратори можуть швидко дізнатися, що вони мають і де вони знаходяться. Наприклад, дізнайтеся, для яких комп’ютерів потрібно оновити драйвери принтера, на яких комп’ютерах достатньо пам’яті та місця на диску тощо.

Розповсюдження та встановлення програмного забезпечення. Після завершення розслідування адміністратори можуть створювати пакети розповсюдження програмного забезпечення, що є дуже ефективним способом зниження вартості такого процесу. Система також централізовано встановлює та керує програмами, що працюють із файлових серверів, і дозволяє кінцевим користувачам запускати такі програми з будь-якої робочої станції в мережі.

Дистанційний аналіз продуктивності та виникаючих проблем. Адміністратори можуть віддалено керувати ресурсами будь-якого ПК в мережі. База даних системи управління зберігає детальну інформацію про

конфігурацію всіх комп’ютерів у мережі, щоб проблеми, що виникають, можна було аналізувати дистанційно.

2.2 Моделі моніторингу та контролю комп’ютерної мережі

Корпоративні мережі можуть бути різнорідними, саме тому інструменти управління не можуть відображати особливості однієї системи чи мережі. Найпоширенішим протоколом керування мережею є SNMP (Simple Network Management Protocol), який підтримується сотнями виробників. Основними перевагами протоколу SNMP є простота, доступність і незалежність від виробника. Значною мірою саме популярність SNMP затримала впровадження CMIP, варіанту протоколу керування OSI. Протокол SNMP призначений для керування маршрутизаторами в Інтернеті та є частиною стеку TCP/IP [13].

У системах управління, побудованих на основі протоколу SNMP, стандартизовані такі елементи:

Договір про взаємодію агент-менеджер;

Мова опису для моделей MIB і повідомлень SNMP;

Мова абстрактної синтаксичної нотації ASN.1 (Стандарт ISO 8824:1987, Рекомендація ITU-T X.208);

Моделі MIB (MIB-I, MIB-II, RMON, RMON 2), імена об’єктів яких зареєстровані в дереві стандартів ISO.

Хоча існує стійка тенденція до використання стандартів ITU-T (включаючи протокол CMIP) у сфері управління телекомунікаційними мережами, є багато прикладів успішного використання управління SNMP. Агенти SNMP вбудовані в аналогові модеми, модеми ADSL, комутатори ATM тощо.

Модель системи керування показана на Рис. 2.1.

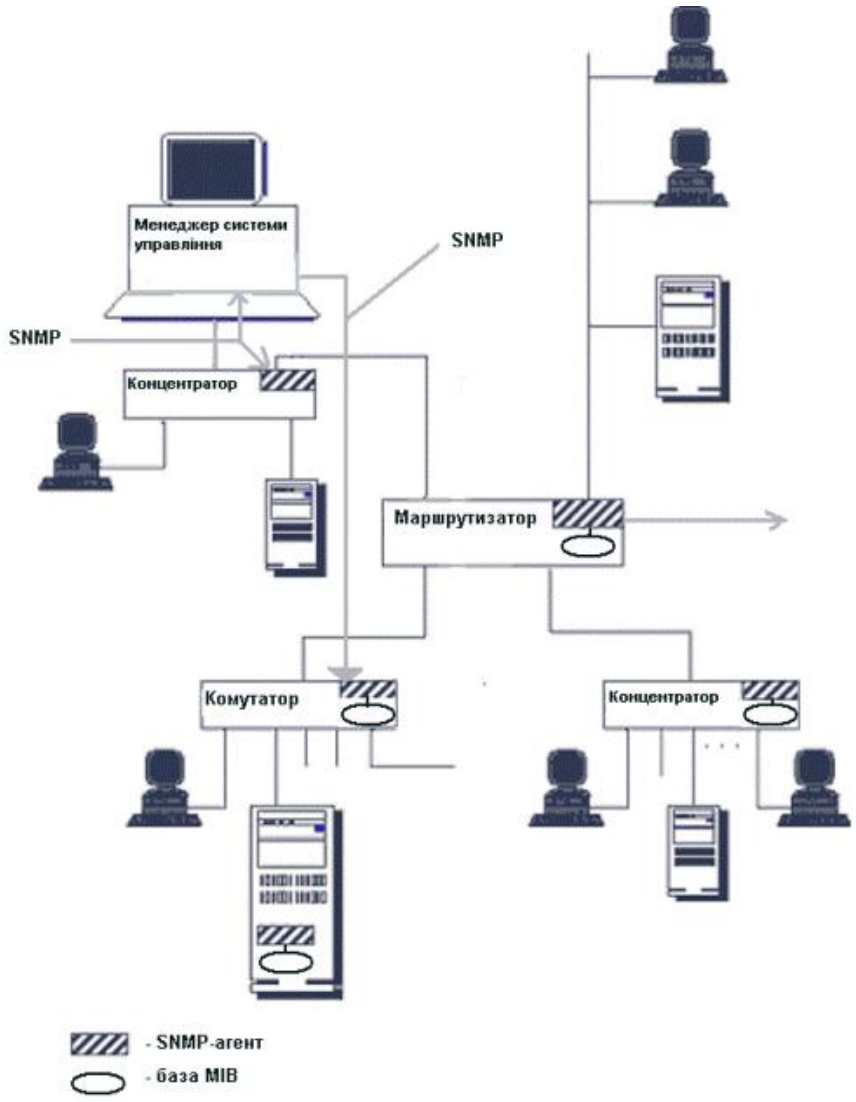


Рисунок 2.1 – Модель моніторингу

SNMP – це протокол, який використовується для отримання інформації від мережевих пристроїв про їхній стан, продуктивність і характеристики, яка зберігається в спеціальній базі даних мережевого пристрою під назвою MIB (Management Information Base). Існують стандарти, які визначають структуру MIB, включаючи набір типів її змінних (об’єктів у термінології ISO), їхні імена та операції, дозволені над цими змінними (наприклад, читання). MIB може зберігати таку інформацію, як мережева та/або MAC-адреса пристрою, значення лічильника та помилки для оброблених пакетів, інформацію про нумерацію, пріоритет і статус порту [14].

Деревоподібна структура MIB містить обов’язкові (стандартні) піддерева, а також може містити приватні (приватні) піддерева, що дозволяє

виробникам інтелектуальних пристроїв реалізувати будь-яку конкретну функціональність на основі їхніх конкретних змінних.

Агент у протоколі SNMP є елементом, який надає менеджерам, розташованим на станціях керування мережею, доступ до значень змінних MIB, таким чином дозволяючи їм реалізовувати функції керування пристроєм і моніторингу.

Загальна модель моніторингу дозволяє сформувати деталізовані моделі для застосунку відповідно до поставлених задач моніторингу та контролю.

Важливим також є аналіз протоколів мережі. На рис. 2.2 представлено алгоритм моніторингу протоколів.

Рисунок 2.2 – Алгоритм моніторингу протоколів

Аналізатори мережевих протоколів можна використовувати для:

локалізація складних питань;

Аналізу на неавторизоване програмне забезпечення;

Аналіз базових шаблонів та показників використання мережі;

Ідентифікація протоколів;

Ідентифікація вторгнень;

Прослуховування трафіку.

Аналізатор протоколів – це автономний спеціальний пристрій, який під’єднується до мережі та працює з програмним забезпеченням. Алгоритм роботи аналізатора мережі може бути використаний як загальний алгоритм моніторингу та контролю роботи мережі.

Вимоги до власного застосунку для створення робочого простору моніторингу визначаються розподіленням системи з можливістю працездатності при відмові окремого мережевого обладнання та роботи мережевого обладнання в різних корпусах управління з різних точок мережі.

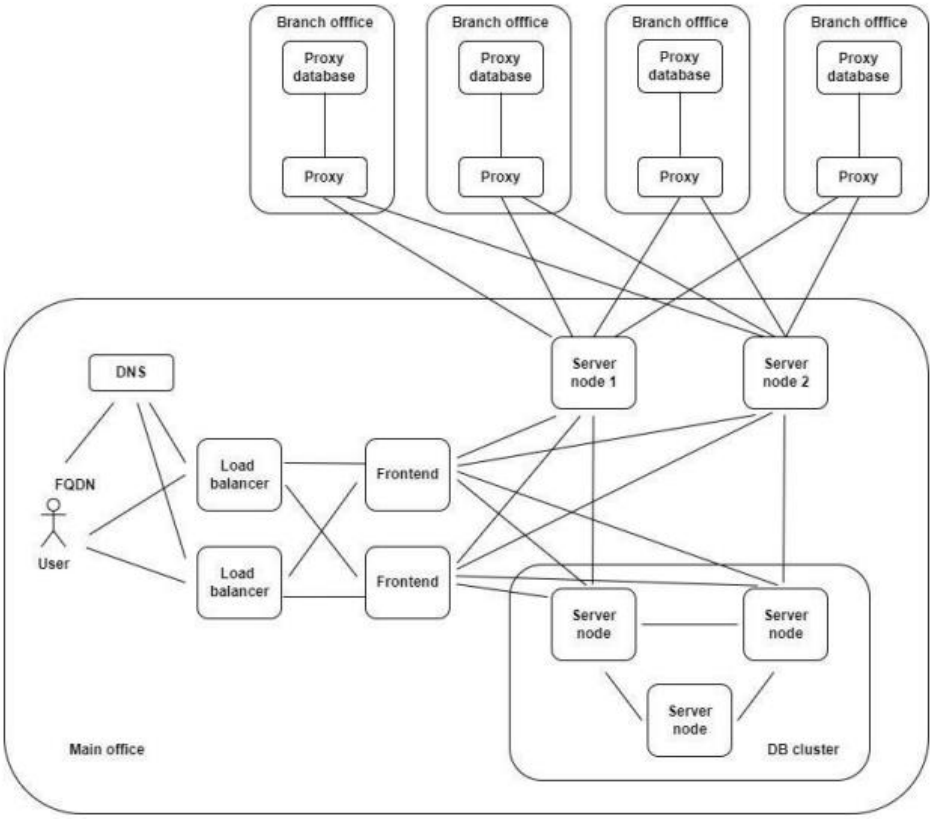


Рисунок 2.3 – Структура системи створення робочого простору моніторингу мережі

Для досягнення цих цілей запропоноване архітектурне рішення враховує різні аспекти.

Щоб створити стійку до збоїв установку, головний офіс оснащений двома серверами Zabbix. Ця конфігурація гарантує, що в разі збою одного сервера другий сервер безперешкодно візьме на себе роботу, зберігаючи мережеву інфраструктуру в повній працездатності.

Цілісність бази даних життєво важлива для бізнес-операцій. Збій у системі бази даних може призвести до значної втрати показників із серйозними наслідками.

Доступ користувачів до мережі полегшується через зовнішні інтерфейси, де будь-які збої можуть вплинути на підключення та взаємодію з користувачем. Впровадження налаштування подвійного інтерфейсу підвищує відмовостійкість, гарантуючи, що якщо один інтерфейс виходить з ладу, другий може негайно взяти на себе роботу, не порушуючи доступ користувача.

Система балансування використовує циклічний механізм для рівномірного розподілу запитів між двома балансувальниками навантаження, які, у свою чергу, розумно розподіляють запити між двома інтерфейсами. Такий механізм запобігає перевантаженню інтерфейсу та, у разі збою інтерфейсу, перенаправляє трафік до функціонального інтерфейсу, зберігаючи безперервність обслуговування. Проксі-сервери дозволяють мережевим адміністраторам оперативно оцінювати стан мережі кожного регіонального офісу, виявляти потенційні проблеми та підвищувати безпеку та ефективність мережі шляхом керування трафіком і запобігання зловмисним запитам. Ця архітектура ретельно розроблена, щоб гарантувати, що корпоративне рішення моніторингу є відмовостійким і розподіленим, задовольняючи основні вимоги щодо постійного доступу до інформації та здатності працювати ефективно в різних географічних місцях.

На рис. 2.4 представлена структурна схема моніторингу.

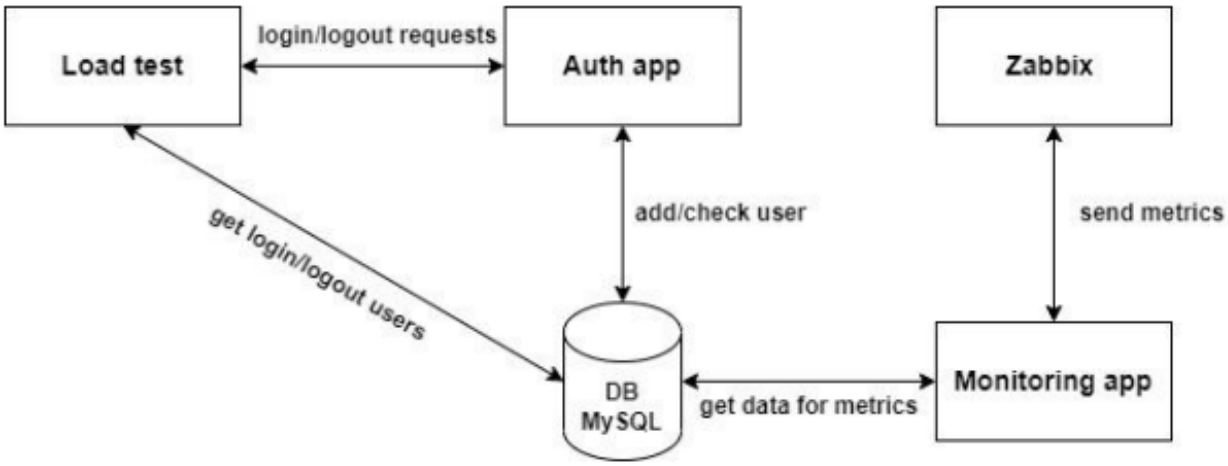


Рисунок 2.4 – Структура системи моніторингу та тестування з авторизацією

Для створення власного рішення було прийнято використовувати головний принцип розподілення. Система моніторингу повинна бути працездатною в ситуаціях відмови різних компонент, працювати з інформацією можна з різних робочих місць.

Для цього сценарію рішення моніторингу Zabbix було реалізовано на місці, щоб фіксувати статус хостів у реальному часі, використовуючи агенти Zabbix для передачі цих даних назад на сервер.

Ця програма призначена для відстеження показників активності користувачів відповідно до певних атрибутів і подальшої передачі цієї інформації на сервер. Додаток складається з трьох ключових елементів: автентифікація користувача, симуляція трафіку та моніторинг. Важливо, що функція моніторингу запускається одночасно з процесом автентифікації користувача, забезпечуючи негайний збір і передачу даних про дії користувача.

2.3 Висновки до розділу 2

В розділі досліджено загальну модель та алгоритм моніторингу протоколів для аналізу роботи мережі. Запропоновані моделі можуть бути використані для створення програмного забезпечення моніторингу та контролю мережі.

Запропонований метод дозволяє сформувати роббочий простір моніторингу з доступом в різні корпуси, об’єднані мережею, працювати з різ них робочих місць.

3. ВИБІР ТЕХНОЛОГІЙ ТА РОЗРОБКА МОДЕЛЕЙ СИСТЕМИ МОНІТОРИНГУ ТА КОНТРОЛЮ КОМП'ЮТЕРНОЇ МЕРЕЖІ

3.1 Технології Zabbix та можливості їх модернізації для корпоративної мережі

Застосунок на основі Zabbix складається з кількох основних програмних компонентів (Рис. 3.1).

Рисунок 3.1 – Головні компоненти застосунку на основі Zabbix

1. Сервер. Сервер Zabbix є основним компонентом, якому агент повідомляє інформацію та статистику щодо доступності та цілісності. Сервер є основним репозиторієм, де зберігаються всі конфігураційні дані, статистика і навіть робочі дані.

2. Бази даних. Вся конфігураційна інформація та дані, зібрані Zabbix, зберігаються в базі даних.

3. Агент. Надає веб-інтерфейс для легкого доступу до Zabbix з будь-якого місця та на будь-якій платформі. Інтерфейс є частиною сервера Zabbix і зазвичай (але не обов'язково) працює на тій же фізичній машині, що й сервер. Під час використання SQLite веб-інтерфейс Zabbix має працювати на тому ж фізичному комп'ютері, що й сервер [15].

Агенти Zabbix можуть збирати дані про продуктивність і доступність від імені серверів Zabbix. Проксі є необов'язковою частиною Zabbix, однак він

може бути корисним для розподілу навантаження на один сервер Zabbix. Агенти Zabbix розгортаються на контрольованих системах для проактивного моніторингу локальних ресурсів і програм і надсилання зібраних даних на сервери або агенти Zabbix.

4.Потік даних Важливо враховувати весь потік даних у Zabbix. Щоб створити елемент даних, який збиратиме дані, потрібно спочатку створити мережевий вузол. Відповідно до базової структури, формується застосунок для моніторингу як адаптивний модуль системи моніторингу та контролю.

Рисунок 3.2 – Архітектура застосунку на основі Zabbix

Далі потрібно створити тригер, який створює дію, тому, якщо необхідно отримувати сповіщення про високе використання ЦП на сервері X, тому спочатку потрібно створити запис мережевого вузла для елементів даних сервера, які відстежують ЦП. Тригер, який спрацьовує, якщо навантаження на процесор занадто велике, генерує дію, яка надсилає електронний лист. Хоча може здатися, що кроків багато, використання шаблону може значно спростити процес. Однак ця збірка системи дозволяє створити дуже гнучку інсталяцію, при цьому сервер Zabbix є основним процесом програмного забезпечення Zabbix [16]. Функціональність базового сервера Zabbix розділена

на три незалежні компоненти: сервер Zabbix, веб-інтерфейс і сховище бази даних.

Будь-які зміни у веб-інтерфейсі Zabbix будуть відображені в розділі останніх даних із затримкою до двох хвилин.

3.2 Мови програмування для реалізації застосунку моніторингу

Zabbix, програмне забезпечення корпоративного рівня, призначене для моніторингу в реальному часі мільйонів показників, зібраних із різних серверів, мережевих пристроїв і програм, є досить універсальним, коли йдеться про мови програмування, які використовуються для його розробки та роботи. Роглянемо основні мови програмування, пов’язані з Zabbix, і як вони сприяють його функціональності.

C є основною мовою, яка використовується для розробки сервера та агента Zabbix. Його використання має вирішальне значення через ефективність і продуктивність C, які необхідні для обробки великих обсягів даних у режимі реального часу. Сервер і агенти Zabbix відповідають за збір даних, їх обробку та виконання заздалегідь визначених дій, таких як надсилання сповіщень або виконання команд на контрольованих хостах. Вибір C гарантує, що ці компоненти можуть витримувати значні навантаження з мінімальним споживанням ресурсів.

PHP відіграє значну роль у Zabbix, особливо у веб-інтерфейсі. Веб-інтерфейс Zabbix, через який користувачі взаємодіють із системою, написаний на PHP. Цей вибір значною мірою пояснюється широким використанням PHP і його сумісністю з різними веб-серверами та базами даних. PHP полегшує генерацію динамічного вмісту, надаючи можливість оновлення в реальному часі та взаємодії на інформаційній панелі Zabbix, що є критично важливим для моніторингу середовищ [18-20].

Java використовується в Zabbix для інтеграції програм на основі Java та моніторингу JMX. Zabbix включає підтримку моніторингу лічильників JMX

(Java Management Extensions) без необхідності встановлення агента в цільовій системі. Це особливо корисно для моніторингу серверів програм Java та іншого програмного забезпечення, яке підтримує JMX. Незалежність Java від платформи та широке використання в корпоративних середовищах роблять її логічним вибором для цього аспекту функціональності Zabbix.

Python і Perl не використовуються в основній розробці самого Zabbix, але підтримуються для розширення можливостей Zabbix. Zabbix підтримує зовнішні перевірки та спеціальні сценарії, написані на Python, Perl або практично на будь-якій іншій мові сценаріїв, яка може виконувати системні команди або мережеві запити. Ця гнучкість дозволяє користувачам адаптувати механізми моніторингу та сповіщень Zabbix до своїх конкретних потреб, що робить його інструментом, який добре адаптується для широкого діапазону середовищ.

Сценарії оболонки, хоча і не є мовою програмування як такої, також часто використовуються з Zabbix для різних зовнішніх перевірок і операцій. Завдяки прямому доступу до системних команд і утиліт, сценарії оболонки є потужним інструментом для спеціальних завдань моніторингу та автоматизованих дій на основі тригерів у Zabbix.

Підсумовуючи, Zabbix використовує комбінацію мов програмування, кожну з яких вибрано за її переваги в певних сферах функціональності системи. Від високопродуктивних можливостей обробки даних C до динамічного веб-інтерфейсу на базі PHP і розширюваності, яку пропонують такі мови сценаріїв, як Python і Perl, Zabbix є свідченням потужності використання правильного інструменту для правильної роботи. Для створення робочого простору було використано мови Python та PHP.

3.3 Розробка модуля моніторингу для корпоративної мережі

Модуль моніторингу створюється відповідно до технологій кластеру високої доступності та використання веб-інтерфейсу, API та JSON.

Програма складається з агентів та управляючого менеджера. Є активні та резервні вузли.

Тільки один вузол може бути активним (робочим) одночасно. На резервному вузлі працює лише один процес – менеджер високої доступності. Резервний вузол не виконує збір даних, обробку чи іншу звичайну діяльність сервера; він не має мінімальних з’єднань з базою даних.

Активний і резервний вузли оновлюють час останнього доступу кожні 5 секунд. Кожен резервний вузол відстежує час останнього доступу активного вузла. Якщо час останнього доступу основного вузла перевищує «затримку відновлення після відмови», резервний вузол стає основним і призначає статус «Недоступний» попередньому основному вузлу.

Активний вузол відстежує власне з'єднання з базою даних - якщо воно втрачено довше ніж "затримка перемикання - 5" секунд, він повинен припинити всю подальшу роботу і перейти в режим очікування. Основний вузол також відстежує стан резервного вузла - якщо останній доступ до резервного вузла здійснювався довше ніж «затримка відмови» в секундах, резервному вузлу буде призначено статус «недоступний».

Для формування віддалених запитів необхідно сформувати файл api_jsonrpc.php і надіслати його до каталогу фронтенду.

Приклад запиту:

curl --request POST \ --url 'https://example.com/zabbix/api_jsonrpc.php' \ --header 'Content-Type: application/json-rpc' \ --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'

Запит повинен мати заголовок Content-Type, встановлений в одне з цих значень: application/json-rpc, application/json або application/jsonrequest.

Об'єкт запиту містить такі властивості:

jsonrpc - версія протоколу JSON-RPC, що використовується API (Zabbix API реалізує JSON-RPC версії 2.0);

method - метод API, що викликається;

params - параметри, які будуть передані методу API;

id - довільний ідентифікатор запиту.

Якщо запит правильний, відповідь, повернута API, має виглядати так:

{ "jsonrpc": "2.0", "result": "6.4.0", "id": 1}

Об'єкт відповіді, в свою чергу, містить такі властивості:

jsonrpc - версія протоколу JSON-RPC;

result - дані, які повертає метод;

id - ідентифікатор відповідного запиту.

Аналогічно формуються файли автотенфікації, авторизації, отримання хосту. Фрагменти коду представлені в додатку.

Важливим процесом є створення та оновлення тригерів. Для цього встановлюється статус тригерів, а також може бути написаний окремий код.

Успішна відповідь міститиме ідентифікатори оновлених тригерів:

{ "jsonrpc": "2.0", "result": { "triggerids": ["13938", "13939"] }, "id": 6}

Це найкращий метод оновлення. При написанні коду можуть бути сформовані помилки в різних випадках. Наприклад, при використанні неправильних вхідних даних, таймауті сеансу або спробі отримати доступ до неіснуючих об'єктів. Застосунок може коректно обробляти такі помилки та формувати відповідне сповіщення.

Щоб створити симульоване середовище трафіку в системі, був розроблений сценарій Python для автоматизації процесів входу та виходу з системи для різної кількості користувачів протягом визначеного періоду часу. Основна функція під назвою `mock_traffic`, описана в Додатку, який містить лістинг, служить шлюзом для ініціювання симуляції трафіку. Ця функція відповідає за почерговий виклик двох інших функцій, `login_users` і `logout_users`, для створення активності в програмі. Функція `login_users` призначена для імітації дій користувачів, які входять у систему. Це досягається шляхом отримання списку неактивних користувачів із бази даних за допомогою допоміжної функції `get_active_uses`. Для кожного з цих користувачів створюється екземпляр `UserLogin`, використовуючи адресу

електронної пошти та пароль користувача, який потім використовується для надсилання запиту POST до кінцевої точки `/user/login` системи. Успішність кожної спроби входу перевіряється шляхом перевірки коду статусу відповіді. І навпаки, функція `logout_users` імітує процес виходу користувачів, націлюючи певну кількість користувачів на дезактивацію за допомогою кінцевої точки `/user/logout`. Щоб додати варіативність моделюванню, код включає використання випадкових чисел у визначеному діапазоні для зміни кількості користувачів, які беруть участь у вході та виході з системи, ефективно імітуючи різну інтенсивність трафіку в системі.

Створення програми автентифікації користувача RESTful включає кілька кроків, кожен з яких зосереджується на іншому аспекті функціональності системи. Використовуючи мову програмування Python і фреймворк FastAPI, розробники можуть ефективно створювати та керувати кінцевими точками програми для реєстрації користувачів, входу та виходу з системи. Цей підхід використовує високу продуктивність і простоту використання FastAPI для створення веб-API.

Процес реєстрації є критично важливим для початку взаємодії користувача з програмою. Коли користувач вирішує зареєструватися, він надає важливу інформацію, таку як своє повне ім’я, адреса електронної пошти, пароль, роль і місцезнаходження. Ці дані надсилаються до програми за допомогою запиту POST до кінцевої точки `/user/signup`. Потім серверна частина виконує SQL-запит, щоб вставити новий запис користувача в базу даних. Якщо ця операція виконана успішно, функція повертає документ JSON, що представляє об’єкт моделі користувача. Цей процес не лише зберігає інформацію про користувача, але й створює основу для автентифікації користувача.

Вхід є простою, але важливою операцією. Коли користувач намагається увійти через кінцеву точку `/user/login`, програма перевіряє базу даних на наявність користувача з наданими обліковими даними. Якщо користувач існує, його статус змінюється з неактивного на активний. Ця зміна статусу є

важливою, оскільки її можна використовувати для керування сеансами користувачів і обмеження доступу до певних частин програми лише для автентифікованих користувачів. Функція входу гарантує, що лише користувачі з дійсними обліковими даними можуть отримати доступ до системи, підвищуючи безпеку програми.

Процес виходу є, по суті, зворотним процесу входу. Надсилаючи запит POST до кінцевої точки `/user/logout`, користувач може завершити свій сеанс. Потім програма повертає статус користувача до неактивного. Ця функція має вирішальне значення для підтримки цілісності сеансів користувача, оскільки вона явно позначає кінець активного сеансу користувача та допомагає запобігти неавторизованому доступу до програми.

Контейнер MySQL відіграє ключову роль у зберіганні та управлінні даними користувача. У ньому міститься таблиця користувачів, яка містить поля для повного імені, електронної пошти, пароля, ролі, розташування та прапорця `is_active`. Ця структура підтримує функціональність програми, надаючи засоби безпечного та ефективного зберігання інформації користувача. Класи типу Enum для розташування та ролі використовуються, щоб гарантувати, що ці поля містять дійсні значення, що ще більше підвищує цілісність даних.

Розробка програми автентифікації користувача RESTful із Python і FastAPI передбачає створення системи, яка може ефективно обробляти процеси реєстрації, входу та виходу користувача. Використання контейнера MySQL для зберігання бази даних гарантує безпечне керування інформацією користувача. Разом ці компоненти утворюють надійну програму, яка може автентифікувати користувачів і керувати їхніми сеансами, забезпечуючи безпечний і ефективний спосіб доступу до функцій системи.

3.4 Висновки до розділу 3

У третьому розділі було виконано аналіз архітектури програмного застосунку Zabbix та спеціального адаптованого модуля для моніторингу та контролю локальної мережі. Визначена архітектура та формування програми управління вузлами корпоративної мережі дозволяє отримати візуальну інформацію для моніторингу та контролю роботи мережі.

4 ТЕСТУВАННЯ ПРОГРАМИ МОНІТОРИНГУ ТА КОНТРОЛЮ ФУНКЦІОНУВАННЯ МЕРЕЖІ

4.1 Тестування системи моніторингу та контролю Zabbix +

Тестування програмного застосунку здійснюються відповідно до базового середовища Zabbix, запуск якого представлено на рис. 4.1.

Рисунок 4.1 – Запуск Zabbix +

Після формування даних для графіків візуалізації (рис. 4.2), користувач може отримати результат зчитування інформації за допомогою SNNP протоколу.

Такий протокол є довідником для мережевого інженера і може бути використаний для аналізу в табличній та візуальній формі. З досвідом такі довідникові дані дозволяють мережевому інженеру швидко прийняти рішення щодо налаштування мережі.

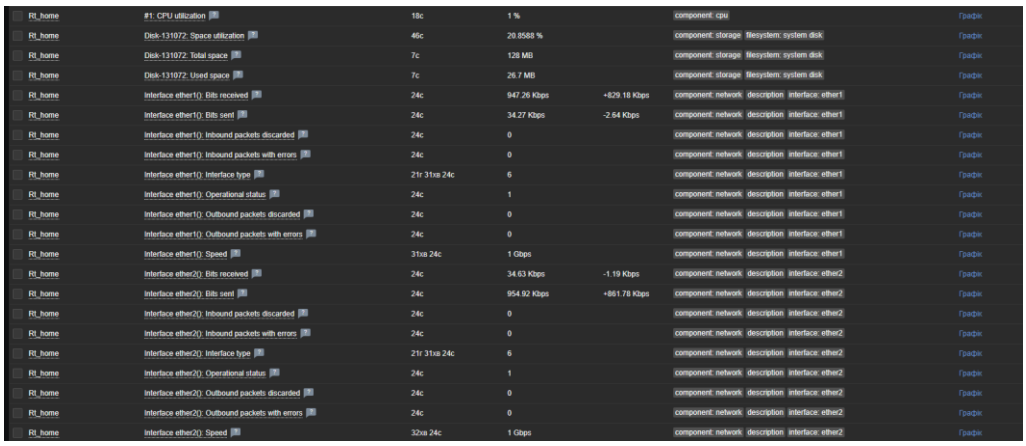


Рисунок 4.2 – Результат зчитування інформації за допомогою SNMP протоколу

Результат зчитування інформації за допомогою SNMP протоколу здійснюється з використанням MIB файлів та виконується обробка інформації для простішого сприйняття інформації користувачем (рис. 4.3).

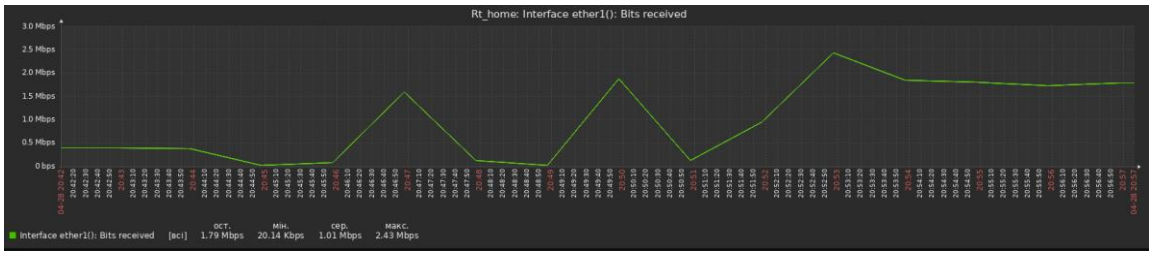


Рисунок 4.3 – Результат зчитування інформації за 15 хв.

Результатом представленої інформації є графік за період 15 хв. Побудова графіка на основі збору інформації на конкретному портові (аплінк) за період 15хв відповідає стандарту.

Інформація меж стиснення даних також формується графічним способом (рис. 4.4).

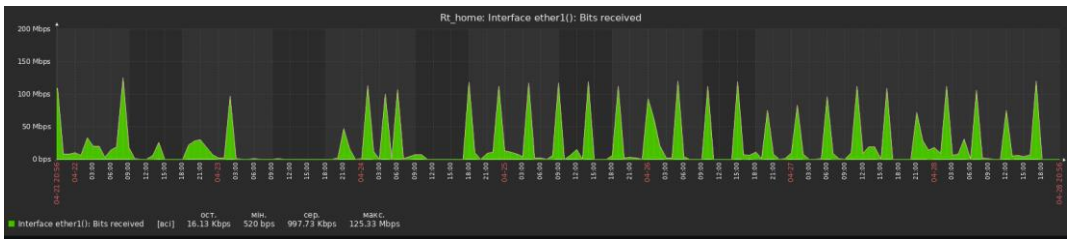


Рисунок 4.4 – Результати процедури стиснення даних

Побудова графіка на основі збору інформації на конкретному портові (аплінк) для оптимізації бази даних на проміжку 7 днів після процедури стиснення відповідає стандарту.

Користувач також може використати мультиграфік з визначених даних.

Так, на рис. 4.5 представлено мультиграфік який об’єднує 4 види вхідних даних (In, out; лічильник помилок на порту, статус інтерфейс активний чи неактивний).

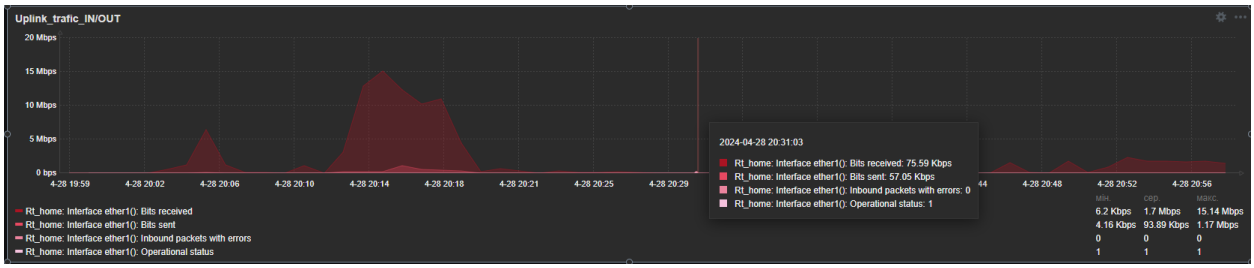
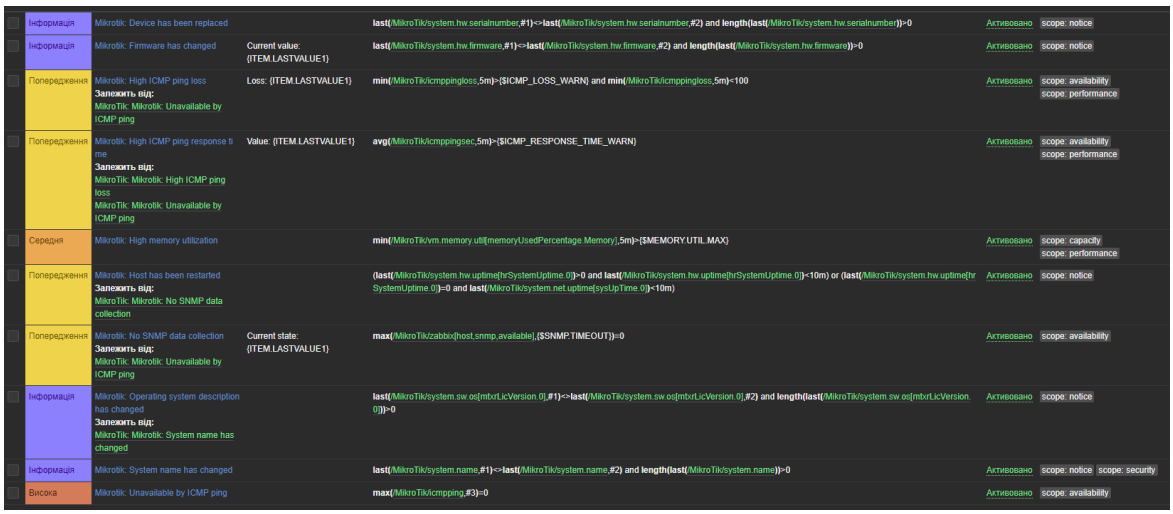


Рисунок 4.5 – Мульти-графік порта (аплінк)

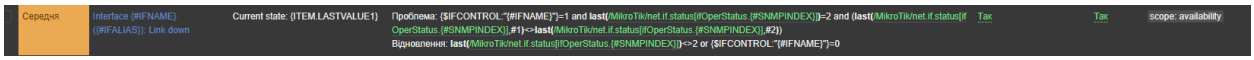
Тригери – це функція для обробки зміни в потоці конкретної інформації (рис. 4.6). . Користувач визначає або формує необхідні йому тригери.



Інформація	Mikrotik: Device has been replaced		last(/MikroTik/system.hw.serialnumber,#1)<>last(/MikroTik/system.hw.serialnumber,#2) and length(last(/MikroTik/system.hw.serialnumber))>0	Активовано	scope: notice
Інформація	Mikrotik: Firmware has changed	Current value: {ITEM.LASTVALUE1}	last(/MikroTik/system.hw.firmware,#1)<>last(/MikroTik/system.hw.firmware,#2) and length(last(/MikroTik/system.hw.firmware))>0	Активовано	scope: notice
Попередження	Mikrotik: High ICMP ping loss Залежить від: MikroTik: Mikrotik: Unavailable by ICMP ping	Loss: {ITEM.LASTVALUE1}	min(/MikroTik/icmppingloss,5m)>{$ICMP_LOSS_WARN} and min(/MikroTik/icmppingloss,5m)<100	Активовано	scope: availability scope: performance
Попередження	Mikrotik: High ICMP ping response time Залежить від: MikroTik: Mikrotik: High ICMP ping loss MikroTik: Mikrotik: Unavailable by ICMP ping	Value: {ITEM.LASTVALUE1}	avg(/MikroTik/icmppingsec,5m)>{$ICMP_RESPONSE_TIME_WARN}	Активовано	scope: availability scope: performance
Середня	Mikrotik: High memory utilization		min(/MikroTik/vm.memory.util[memoryUsedPercentage.Memory],5m)>{$MEMORY.UTIL.MAX}	Активовано	scope: capacity scope: performance
Попередження	Mikrotik: Host has been restarted Залежить від: MikroTik: Mikrotik: No SNMP data collection		(last(/MikroTik/system.hw.uptime[hrSystemUptime.0])>0 and last(/MikroTik/system.hw.uptime[hrSystemUptime.0])<10m) or (last(/MikroTik/system.hw.uptime[hrSystemUptime.0])=0 and last(/MikroTik/system.net.uptime[sysUpTime.0])<10m)	Активовано	scope: notice
Попередження	Mikrotik: No SNMP data collection Залежить від: MikroTik: Mikrotik: Unavailable by ICMP ping	Current state: {ITEM.LASTVALUE1}	max(/MikroTik/zabbix[host,snmp,available],{$SNMP.TIMEOUT})=0	Активовано	scope: availability
Інформація	Mikrotik: Operating system description has changed Залежить від: MikroTik: Mikrotik: System name has changed		last(/MikroTik/system.sw.os[mtxrLicVersion.0],#1)<>last(/MikroTik/system.sw.os[mtxrLicVersion.0],#2) and length(last(/MikroTik/system.sw.os[mtxrLicVersion.0]))>0	Активовано	scope: notice
Інформація	Mikrotik: System name has changed		last(/MikroTik/system.name,#1)<>last(/MikroTik/system.name,#2) and length(last(/MikroTik/system.name))>0	Активовано	scope: notice scope: security
Висока	Mikrotik: Unavailable by ICMP ping		max(/MikroTik/icmpping,#3)=0	Активовано	scope: availability

Рисунок 4.6 – База тригерів для певних видів задач

На рисунку 4.7 представлено тригер, який реагує на зміну статусу інтерфейсу



Середня	Interface {#IFNAME} ({#IFALIAS}): Link down	Current state: {ITEM.LASTVALUE1}	Проблема: {$IFCONTROL:"{#IFNAME}"}=1 and last(/MikroTik/net.if.status[ifOperStatus,{#SNMPINDEX}])=2 and (last(/MikroTik/net.if.status[ifOperStatus,{#SNMPINDEX}],#1)<>last(/MikroTik/net.if.status[ifOperStatus,{#SNMPINDEX}],#2)) Відновлення: last(/MikroTik/net.if.status[ifOperStatus,{#SNMPINDEX}])<>2 or {$IFCONTROL:"{#IFNAME}"}=0	Так	Так	scope: availability

Рисунок 4.7 – Тригер зміни статусу

Для роботи тригера використовується два потоки інформації це програмний статус порта та фізичний статус порта. Логіка роботи якщо програмний статус порта активний (1), а фізичний статус порта опущений (2) то в такому разі зпрацьовує тригер що порт опущений, у разі відновлення роботи порта потрібно щоб фізичний статус інтерфейс не дорівнював опущений (2). На рис. 4.8 представлено вікно налаштування тригеру.

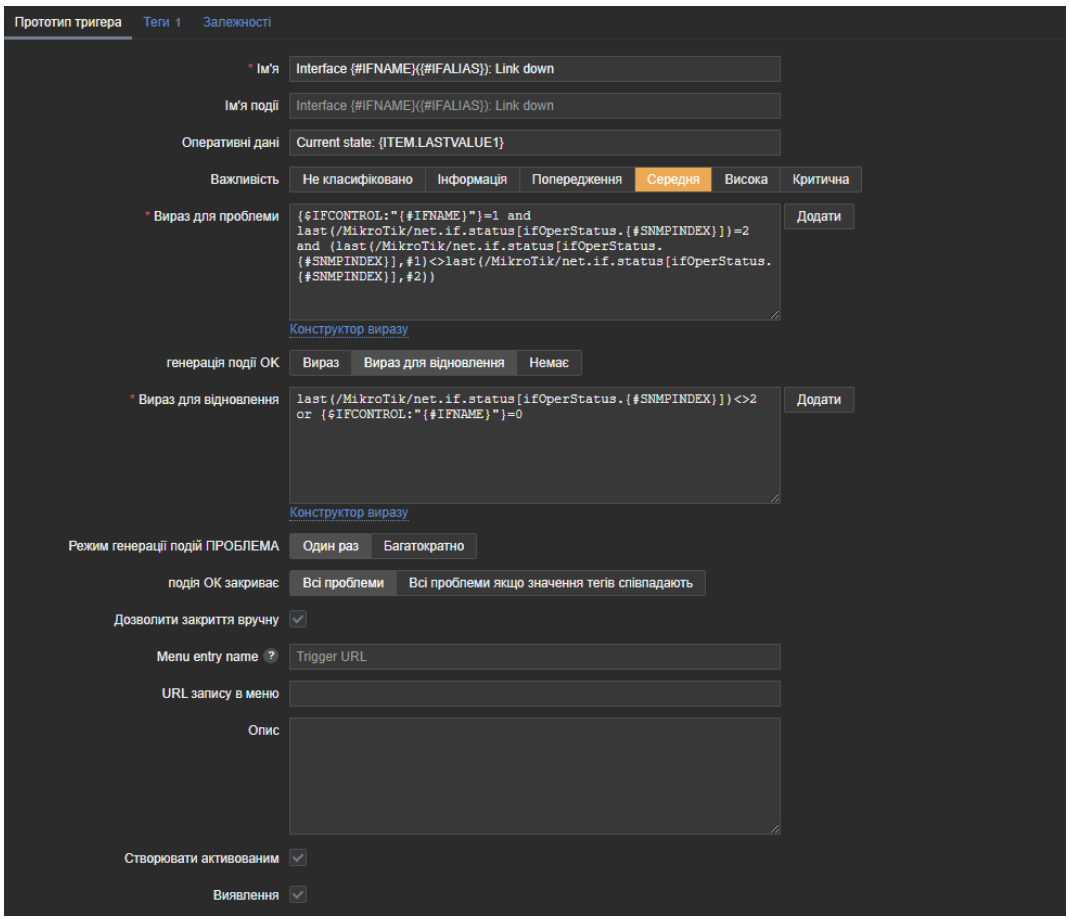


Рисунок 4.8 – Вікно налаштування тригерів

Веб-панель використовується для сповіщення про роботу тригерів. Вона представлена на рис. 4.9.

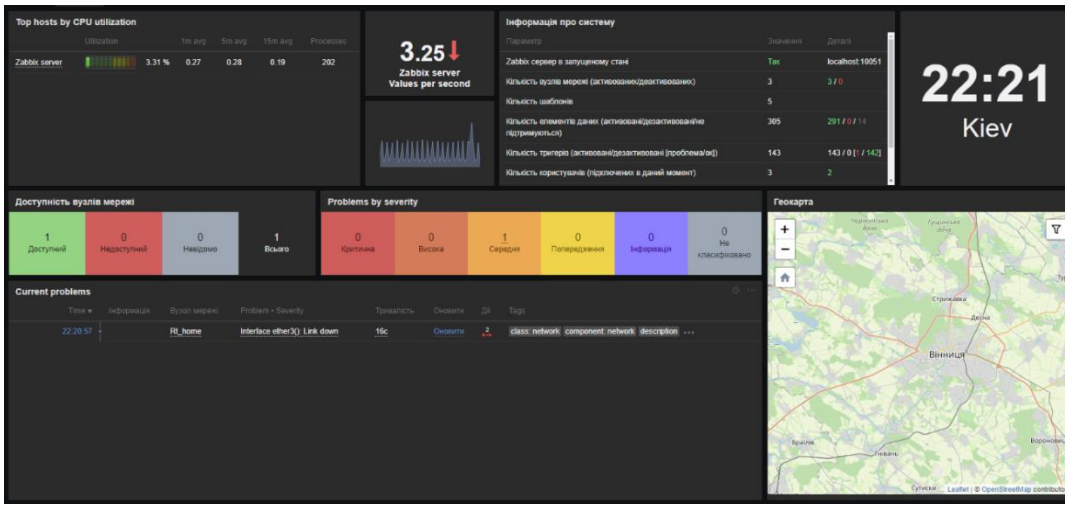


Рисунок 4.9 – Відображення спрацьованого тригери на веб панелі

Також можна відслідкувати історію попередніх спрацьовувань тригерів (рис. 4.10).

Рисунок 4.10 – Історія попередніх спрацювань тригерів

Отримані результати передаються в телеграм –бот або на окремий веб-сайт.

В залежності від кваліфікації користувача, інформація може бути оброблена та представлена в табличному або/графічному вигляді (рис. 4.11).

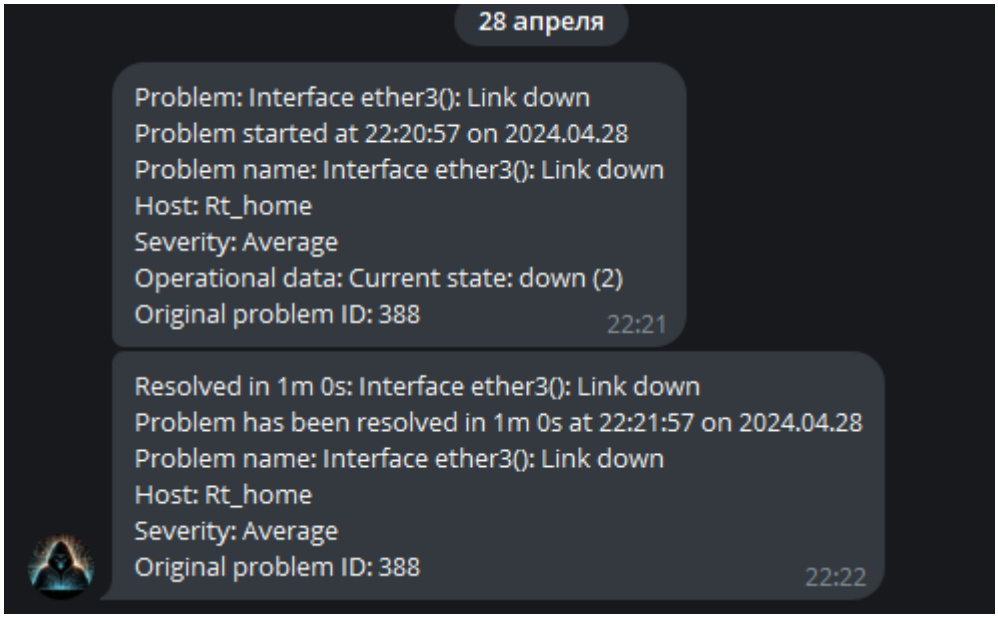


Рисунок 4.11 – Результат спрацювання тригера в телеграм бот

Оброблені результати представляються для користувача в табличному вигляді (рис. 4.12).

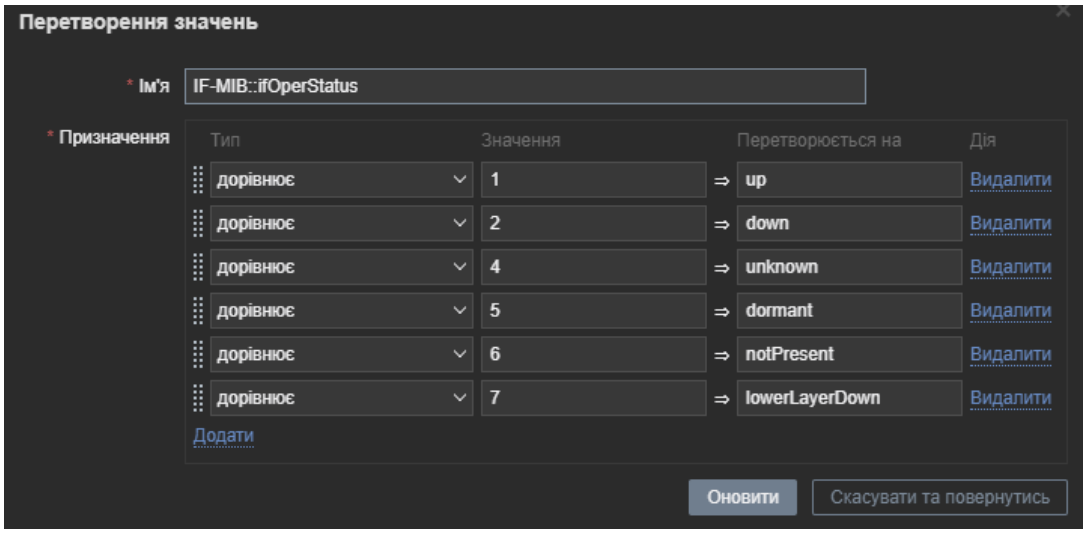


Рисунок 4.12 – Перетворення отриманих значень для простішого сприйняття користувачем

Отримані результати дозволяють мережевим інженерам, а також системним адміністраторам отримувати динамічну інформацію про стан мережі.

4.2 Розгортання та використання програмного забезпечення

Технічні вимоги для запуску програмного продукту:

Сервер із встановленим Zabbix 1.0 (RedHat Linux 8.0, ядро 2.4.18-14, MySQL / MyISAM 3.23.54a-4, Pentium IV 1.5 ГГц, 256 Мб, IDE) може збирати більше 200 параметрів в секунду з спостережуваних серверів затримок у мережі).

Якщо кожен сервер спостерігає десять параметрів, а користувач хоче оновлювати ці параметри один раз на 30 секунд, то Zabbix здатний обробляти 600 серверів (чи 6000 перевірок). Для застосунку Zabbix+ необхідно врахувати час обробки та час передачі даних в телеграм-бот або на будь який веб-сайт.

Після відкриття застосунку Zabbix+ користувач авторизує себе та свою мережу, вибирає параметри моніторингу та тригери. Після цього запускає цикл моніторингу та може контролювати параметри за вікнами.

Контейнерізація Docker відіграє вирішальну роль у модернізації та спрощенні управління інфраструктурою систем моніторингу, таких як Zabbix. Вибираючи Docker, ви використовуєте потужність контейнеризації, яка по суті інкапсулює середовище Zabbix у контейнерах. Ці контейнери забезпечують послідовне та ізольоване налаштування для кожного компонента системи, гарантуючи, що поведінка залишається узгодженою на різних платформах і розгортаннях. Це особливо корисно для систем моніторингу Zabbix, які потребують надійного моніторингу різних показників у різних середовищах. Використання Docker Compose додатково спрощує цей процес, дозволяючи розгортати кілька служб, таких як MariaDB Galera Cluster, Zabbix Server, Zabbix Agent і Zabbix Web, за допомогою однієї команди. Docker Compose полегшує конфігурацію цих служб для бездоганної взаємодії в приватній мостовій мережі, яка називається «zabbix-net», підвищуючи безпеку та ефективність зв’язку між службами. Агент Zabbix є ключовим компонентом, завданням якого є збір даних щодо безлічі показників, включаючи використання процесора, споживання пам’яті, активність диска та статистику мережі. Ці дані необхідні для повноцінного моніторингу Zabbix Server.

Сам сервер діє як ядро системи, де він обробляє, агрегує та зберігає зібрані дані в базі даних MariaDB. MariaDB відіграє тут вирішальну роль, забезпечуючи цілісність, безпеку та швидкий доступ до історичних даних і даних моніторингу в реальному часі. Крім того, веб-інтерфейс Zabbix використовує ці дані, надаючи адміністраторам і користувачам потужний інструмент для візуалізації показників, керування системою моніторингу та визначення спеціальних параметрів моніторингу. Цей аспект Zabbix є вирішальним для прийняття обґрунтованих рішень і тонкого налаштування моніторингу для задоволення конкретних потреб.

Конфігурація серверу Zabbix і контейнерів агента, як детально описано у файлі docker-compose.yml, ілюструє гнучкість і налаштування, можливі за допомогою Docker. Наприклад, конфігурація контейнера Zabbix Server включає параметри підключення до бази даних, конфігурації мережі та змінні середовища, які адаптують роботу сервера до конкретних вимог розгортання. Подібним чином налаштування контейнера Zabbix Agent гарантує належне підключення до Zabbix-сервера для передачі даних і налаштовано для привілейованого доступу до ресурсів хоста, необхідного для детального моніторингу системи. Розгортання цих служб за допомогою команди `docker compose up` створює повністю працездатну взаємопов’язану мережу моніторингу. Це оптимізоване розгортання не тільки спрощує процес початкового налаштування, але й полегшує подальше масштабування та керування системою моніторингу. Контейнерний підхід Docker у поєднанні з Docker Compose для оркестровки представляє потужну методологію для розгортання та керування середовищами моніторингу Zabbix, гарантуючи, що вони є надійними та адаптованими до мінливих вимог.

4.3 Висновки до розділу 4

У четвертому розділі було проведено тестування функцій моніторингу та контролю мережі та виконано аналіз отриманих візуальних даних, передачі сповіщення в телеграм бот та обробки інформації для більш легкого сприйняття користувачем.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено програмне забезпечення моніторингу та контролю функціонування локальної мережі. Аналіз відомих підходів до моніторингу комп'ютерних мереж дозволив сформувати загальні вимоги до застосунку, моделі та архітектуру застосунку. Аналіз аналогів дозволив обґрунтовано вибрати платформу для формування адаптованого застосунку з можливостями формування інформації для телеграм-боту або інших сповіщень щодо проблем або стабільної роботи мережі.

В бакалаврській дипломній роботі виконано:

розробку методів, моделей моніторингу та контролю комп'ютерних мереж;

розробку алгоритмів моніторингу та контролю комп'ютерних мереж ;

створення програмного забезпечення для моніторингу функціонування комп'ютерної мережі;

тестування модулів програмного забезпечення моніторингу та контролю мережі.

Теоретичні знання концепції моніторингу, роботи аналогів стали основою для моделей та алгоритмів створення адаптованого програмного застосунку моніторингу та контролю показників роботи комп'ютерної мережі.

Практичне значення виконаної роботи полягає в використанні розробленого програмного забезпечення для автоматизованої системи моніторингу та контролю комп'ютерної мережі. Керуюче програмне забезпечення активно використовується для контролю показників роботи мережі.

Розроблений модуль може бути використаний для корпоративних комп'ютерних мереж, зокрема для комп'ютерної мережі університету. Практичне впровадження буде виконано на прикладі комп'ютерної мережі ВНТУ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Network-monitoring URL: https://www.ibm.com/topics/network-monitoring

2. Зварич В.М., Коваленко О.О. Програмне забезпечення моніторингу комп'ютерних мереж. Збірник матеріалів інтернет-конференції «Інформаційне суспільство», Тернопіль. 2024. №89.

3. Pang-Wei Tsai, Chun-Wei Tsai, Chia-Wei Hsu, Chu-Sing Yang. Network Monitoring in Software-Defined Networking: A Review // IEEE Systems Journal. — 2018. — Т. 12, вип. 4 (1 грудня). — С. 3958–3969.

4. Моніторинг обладнання: URL: https://hs.whitehat.one/ukua/blog/monitorynh-pratsezdatnosti-vashoho-obladnannya

5. Топ-10 кращих программ для моніторингу мережі [Електронний ресурс] https://www.softinventive.ru/best-network-monitoring-tools/ 4. Nagios [Електронний ресурс]: Nagios Documentation – http://www.nagios.org/

6. Mohd F., Nur F., Muhammad N. Integrated Network Monitoring using Zabbix with Push Notification via Telegram. Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1, 2022. – pp. 158-166.

7. Касовська І.В., Шаповаленко О.Д., Луценко І.М. Програмні комплекси мережевого моніторингу для підвищення ефективності захисту мереж. *Сучасний захист інформації*, №1(45), 2021. – С. 47–52.

8. Білобровець І. В. Розробка структури системи захисту хмарної бази даних від вторгнень. Зб. тез доповідей на наук.-практ. конференції «Цифрова трансформація кібербезпеки» 20 квітня 2022 року. К.: ДУТ. – С. 6–8.

9. Топ 10 програм для моніторингу мережі у 2024 році - Softinventive Lab. URL: https://www.softinventive.com.ua/best-network-monitoring-tools

10. Лоуренс Вільямс 12 найкращих програмних засобів моніторингу системи (2024). URL: https://www.guru99.com/uk/best-system-monitoring-software.html

11. Технології управління комп'ютерними мережами. 2016 URL: https://ot.vntu.edu.ua/wp-content/uploads/2021/02/sil_monitor.pdf

12. Комп'ютерні мережі. 2. Інтернет-протоколи. 2021 URL: https://www.scs.kpi.ua/wp-content/uploads/2021/07/pdf/komp

13. Адміністрування комп'ютерних мереж - Wireshark (безкоштовна), Операційна система Windows (Ліцензована), Операційна система Linux (безкоштовна) URL: https://document.kdu.edu.ua/metod/2023_7941.pdf

14. 14 найкращих інструментів моніторингу мережі (2024) URL: https://www.guru99.com/uk/network-monitoring-tools.html

15. Програмне забезпечення комп'ютерних мереж (Knowledge Ware), BPwin (Logic Works), CDEZ Tods (Oracle), Clear Case (Alria Software) URL: https://pidru4niki.com/12800528/bankivska_sprava/programne_zabezpechennya_kompyuternih_informatsiynih_sistem

16. Топ 10 програм для моніторингу мережі у 2024 році - Network Olympus, Observium URL: https://www.scs.kpi.ua/wpcontent/uploads/2021/07/pdf/komp%E2%80%98yuterni-merezhi-2.pdf

17. Overview of Zabbix URL: https://www.zabbix.com/documentation/1.8/en/manual/about/overview_of_zabbix

18. Using Zabbix for Risk Management URL: https://blog.zabbix.com/usingzabbix-for-risk-management/6570/

19. Гринкевич Г.О., Домрачева К.О., Якимчук С.П. Налаштування системи моніторингу ZABBIX . Наукові записки УНДІЗ. № 2(58), С. 12-20. DOI: 10.31673/2518

20. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 121 "Інженерія програмного забезпечення" / Уклад.: О. Романюк, Г. Черноволик. Вінниця : ВНТУ, 2022. 52 с.

ДОДАТКИ

54

Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
"__12__"____березня_____2024 р.

Технічне завдання

на бакалаврську дипломну роботу

«Розробка програмного забезпечення для моніторингу показників якості корпоративних мереж»

за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:
_____ к.т.н., доц. Коваленко О. О.
« 12 » __березня__ 2024 р.

Виконав:
_____студент гр. 4ПІ-20б Зварич В.М.
« 12 » березня 2024 р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка програмного забезпечення для моніторингу показників якості корпоративних мереж».

Галузь застосування – корпоративні мережі

2. Підстава для розробки.

Завдання на роботу, яке затверджене на засіданні кафедри програмного забезпечення – протокол №17 від 20 березня 2024 р.

3. Мета та призначення розробки.

Метою роботи є розробка програмного забезпечення для моніторингу показників якості функціонування комп'ютерних мереж.

Призначення полягає у використанні розробленого програмного забезпечення для автоматизованої системи моніторингу та контролю комп'ютерної мережі. Розроблений модуль може бути використаний для корпоративних комп'ютерних мереж, зокрема для комп'ютерної мережі університету.

4. Вихідні дані для проведення НДР

1. Network-monitoring URL: https://www.ibm.com/topics/network-monitoring

2. Overview of Zabbix URL: https://www.zabbix.com/documentation/1.8/en/manual/about/overview_of_zabbix

3. Using Zabbix for Risk Management URL: https://blog.zabbix.com/usingzabbix-for-risk-management/6570/

4. Гринкевич Г.О., Домрачева К.О., Якимчук С.П. Налаштування системи моніторингу ZABBIX . Наукові записки УНДІЗ. № 2(58), С. 12-20. DOI: 10.31673/2518

5. Технічні вимоги

Сервер із встановленим Zabbix 1.0 (RedHat Linux 8.0, ядро 2.4.18-14, MySQL / MyISAM 3.23.54a-4, Pentium IV 1.5 ГГц, 256 Мб, IDE) може збирати більше 200 параметрів в секунду з спостережуваних серверів затримок у мережі).

6. Конструктивні вимоги.

Система має відповідати функціональним та технічним вимогам. Текстова та графічна документація повинна відповідати стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до бакалаврської дипломної роботи;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи
1	Аналіз стану розвитку систем моніторингу комп'ютерних мереж	14.03.2024 – 31.03.2024
2	Розробка методу, моделей моніторингу	01.04.2024– 15.04.2024
3	Розробка модулів застосунку	16.04.2024 – 18.05.2024
4	Тестування програми	19.05.2024 – 24.05.2024
5	Оформлення матеріалів до захисту БДР	25.05.2024 – 30.05.2024

10. Порядок контролю та прийняття

Всі етапи бакалаврської роботи контролюються науковим керівником згідно плану по виконанню роботи. Прийняття бакалаврської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком захисту. Дозволяється корегування бакалаврської дипломної роботи.

Додаток Б – Протокол перевірки бакалаврської дипломної роботи на наявність текстових запозичень

Назва роботи: «Розробка програмного забезпечення для моніторингу показників якості корпоративних мереж»

Тип роботи: БДР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Коваленко О.О.

Оригінальність	92%
Схожість	8%

Аналіз звіту подібності

■ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Черноволик Г. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichek

Автор роботи _____ Зварич В.М.

Керівник роботи _____ Коваленко О.О.

Додаток В – Лістинг програми


```
{

            "name": "Templates/Network devices"
          }
        ],
        "templates": [
          {
            "uuid": "318fd61c22fa4f1a92a71376814d6c32",
            "template": "MikroTik",
            "name": "MikroTik",
            "description": "The template for monitoring Ethernet router MikroTik RB2011iLS-IN.\n\nDesktop metal case, 5xEthernet, 5xGigabit Ethernet, SFP cage, PoE out on port\n10, 600MHz CPU, 64MB RAM, RouterOS L4\n\nMIBs used:\nHOST-RESOURCES-MIB\nSNMPv2-MIB\nMIKROTIK-MIB\nIF-MIB\n\nGenerated by official Zabbix template tool \"Templator\"",
            "vendor": {
              "name": "Zabbix",
              "version": "6.4-0"
            },
            "groups": [
              {
                "name": "Templates/Network devices"
              }
            ],
            "items": [
              {
                "uuid": "a7fc5ad5ffb14a6185e288617294aa98",
                "name": "Mikrotik: ICMP ping",
                "type": "SIMPLE",
                "key": "icmpping",
                "history": "7d",
                "valuemap": {
                  "name": "Service state"
                },
                "tags": [
                  {
                    "tag": "component",
                    "value": "health"
                  },
                  {
                    "tag": "component",
                    "value": "network"
                  }
                ],
                "triggers": [
                  {
                    "uuid": "7257cb86a65547b5901cac08cacee3aa",
                    "expression": "max(/MikroTik/icmpping,#3)=0",
                    "name": "Mikrotik: Unavailable by ICMP ping",
                    "priority": "HIGH",
                    "description": "Last three attempts returned timeout.  Please check device connectivity.",
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "availability"
                      }
                    ]
                  }
                ]
              },
              {
                "uuid": "783ce77b38434492a9944dc5b3531a35",
                "name": "Mikrotik: ICMP loss",
                "type": "SIMPLE",
                "key": "icmppingloss",
                "history": "7d",
                "value_type": "FLOAT",
                "units": "%",
                "tags": [
                  {
                    "tag": "component",
                    "value": "health"
```

```
      },
      {
        "tag": "component",
        "value": "network"
      }
    ],
    "triggers": [
      {
        "uuid": "ae82608801a642d9b3ef8ddd06cee7f6",
        "expression": "min(/MikroTik/icmppingloss,5m)>{$ICMP_LOSS_WARN} and min(/MikroTik/icmppingloss,5m)<100",
        "name": "Mikrotik: High ICMP ping loss",
        "opdata": "Loss: {ITEM.LASTVALUE1}",
        "priority": "WARNING",
        "dependencies": [
          {
            "name": "Mikrotik: Unavailable by ICMP ping",
            "expression": "max(/MikroTik/icmpping,#3)=0"
          }
        ],
        "tags": [
          {
            "tag": "scope",
            "value": "availability"
          },
          {
            "tag": "scope",
            "value": "performance"
          }
        ]
      }
    ]
  },
  {
    "uuid": "bc7299499e2847eeb8d4e9ebcbc2519a",
    "name": "Mikrotik: ICMP response time",
    "type": "SIMPLE",
    "key": "icmppingsec",
    "history": "7d",
    "value_type": "FLOAT",
    "units": "s",
    "tags": [
      {
        "tag": "component",
        "value": "health"
      },
      {
        "tag": "component",
        "value": "network"
      }
    ],
    "triggers": [
      {
        "uuid": "d2f60c5c4a124a549eb02f62874dba8b",
        "expression": "avg(/MikroTik/icmppingsec,5m)>{$ICMP_RESPONSE_TIME_WARN}",
        "name": "Mikrotik: High ICMP ping response time",
        "opdata": "Value: {ITEM.LASTVALUE1}",
        "priority": "WARNING",
        "dependencies": [
          {
            "name": "Mikrotik: High ICMP ping loss",
            "expression": "min(/MikroTik/icmppingloss,5m)>{$ICMP_LOSS_WARN} and min(/MikroTik/icmppingloss,5m)<100"
          },
          {
            "name": "Mikrotik: Unavailable by ICMP ping",
            "expression": "max(/MikroTik/icmpping,#3)=0"
          }
        ],
        "tags": [
          {
            "tag": "scope",
            "value": "availability"
          },
          {
            "tag": "scope",
```

```
                        "value": "performance"
                      }
                    ]
                  }
                ]
              },
              {
                "uuid": "aec4de5980224cb8b4f7e573866ea86d",
                "name": "Mikrotik: SNMP walk network interfaces",
                "type": "SNMP_AGENT",
                "snmp_oid":
"walk[1.3.6.1.2.1.2.2.1.8,1.3.6.1.2.1.2.2.1.7,1.3.6.1.2.1.31.1.1.1.18,1.3.6.1.2.1.31.1.1.1.1,1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3,1.3.6.1.2.1.31.1.1
.1.6,1.3.6.1.2.1.31.1.1.1.10,1.3.6.1.2.1.2.2.1.14,1.3.6.1.2.1.2.2.1.20,1.3.6.1.2.1.2.2.1.19,1.3.6.1.2.1.2.2.1.13,1.3.6.1.2.1.31.1.1.1.15]",
                "key": "net.if.walk",
                "history": "0",
                "trends": "0",
                "value_type": "TEXT",
                "description": "Discovering interfaces from IF-MIB.",
                "tags": [
                  {
                    "tag": "component",
                    "value": "raw"
                  }
                ]
              },
              {
                "uuid": "3d2748be91c24dc69202b9fed8ecd8b8",
                "name": "Mikrotik: SNMP walk wireless interfaces",
                "type": "SNMP_AGENT",
                "snmp_oid":
"walk[1.3.6.1.4.1.14988.1.1.14.1.1.2,1.3.6.1.2.1.31.1.1.1.18,1.3.6.1.2.1.2.2.1.3,1.3.6.1.2.1.2.2.1.7,1.3.6.1.4.1.14988.1.1.16.1.1.2,1.3.6.1.4.1.149
88.1.1.16.1.1.4,1.3.6.1.4.1.14988.1.1.16.1.1.3,1.3.6.1.4.1.14988.1.1.16.1.1.7,1.3.6.1.4.1.14988.1.1.1.3.1.4,1.3.6.1.4.1.14988.1.1.1.3.1.8,1.3.6.1.
4.1.14988.1.1.1.3.1.9,1.3.6.1.4.1.14988.1.1.1.3.1.6,1.3.6.1.4.1.14988.1.1.1.3.1.11,1.3.6.1.4.1.14988.1.1.1.7.1.5,1.3.6.1.4.1.14988.1.1.1.7.1.4,1.3
.6.1.4.1.14988.1.1.1.7.1.2,1.3.6.1.4.1.14988.1.1.1.7.1.3]",
                "key": "net.if.wireless.walk",
                "history": "0",
                "trends": "0",
                "value_type": "TEXT",
                "tags": [
                  {
                    "tag": "component",
                    "value": "raw"
                  }
                ]
              },
              {
                "uuid": "7bde673035aa487fb10ddc79535b4b9a",
                "name": "Mikrotik: SNMP walk CPU temperature sensors",
                "type": "SNMP_AGENT",
                "snmp_oid": "walk[1.3.6.1.4.1.14988.1.1.3.11]",
                "key": "sensor.cpu.temp.walk",
                "history": "0",
                "trends": "0",
                "value_type": "TEXT",
                "description": "MIB: MIKROTIK-MIB\nDiscovering CPU temperature sensors.",
                "tags": [
                  {
                    "tag": "component",
                    "value": "raw"
                  }
                ]
              },
              {
                "uuid": "89de96cb2feb47e587590ef882d90bff",
                "name": "Mikrotik: SNMP walk temperature sensors",
                "type": "SNMP_AGENT",
                "snmp_oid": "walk[1.3.6.1.4.1.14988.1.1.3.10]",
                "key": "sensor.temp.walk",
                "history": "0",
                "trends": "0",
                "value_type": "TEXT",
                "description": "MIB: MIKROTIK-MIB\nDiscovering temperature sensors.",
                "tags": [
                  {
```

```
                "tag": "component",
                "value": "raw"
              }
            ]
          },
          {
            "uuid": "397dd775fe6d4046aa91dfca36d3be96",
            "name": "Mikrotik: SNMP traps (fallback)",
            "type": "SNMP_TRAP",
            "key": "snmptrap.fallback",
            "delay": "0",
            "history": "7d",
            "trends": "0",
            "value_type": "LOG",
            "description": "The item is used to collect all SNMP traps unmatched by other snmptrap items",
            "logtimefmt": "hh:mm:sszyyyy/MM/dd",
            "tags": [
              {
                "tag": "component",
                "value": "network"
              }
            ]
          },
          {
            "uuid": "30bc4fcfe70a45ceb6b3a8b945779e31",
            "name": "Mikrotik: System contact details",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.1.4.0",
            "key": "system.contact[sysContact.0]",
            "delay": "15m",
            "history": "7d",
            "trends": "0",
            "value_type": "CHAR",
            "description": "MIB: SNMPv2-MIB\nThe textual identification of the contact person for this managed node, together with information on how to contact this person.  If no contact information is known, the value is the zero-length string.",
            "inventory_link": "CONTACT",
            "preprocessing": [
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "12h"
                ]
              }
            ],
            "tags": [
              {
                "tag": "component",
                "value": "system"
              }
            ]
          },
          {
            "uuid": "dd634cc1e804449b90693bf1925dc3d4",
            "name": "Mikrotik: SNMP walk system CPUs",
            "type": "SNMP_AGENT",
            "snmp_oid": "walk[1.3.6.1.2.1.25.3.3.1.1,1.3.6.1.2.1.25.3.3.1.2]",
            "key": "system.cpu.walk",
            "history": "0",
            "trends": "0",
            "value_type": "TEXT",
            "description": "MIB: HOST-RESOURCES-MIB\nDiscovering system CPUs.",
            "tags": [
              {
                "tag": "component",
                "value": "raw"
              }
            ]
          },
          {
            "uuid": "042947145fd142f3a1f6ae86826480eb",
            "name": "Mikrotik: System description",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.1.1.0",
            "key": "system.descr[sysDescr.0]",
```

```
        "delay": "15m",
        "history": "7d",
        "trends": "0",
        "value_type": "CHAR",
        "description": "MIB: SNMPv2-MIB\nA textual description of the entity. This value should\ninclude the full name and version identification of the system's hardware type, software operating-system, and\nnetworking software.",
        "preprocessing": [
          {
            "type": "DISCARD_UNCHANGED_HEARTBEAT",
            "parameters": [
              "12h"
            ]
          }
        ],
        "tags": [
          {
            "tag": "component",
            "value": "system"
          }
        ]
      },
      {
        "uuid": "ed51e55b8fa64c5599399fc93d81d329",
        "name": "Mikrotik: Firmware version",
        "type": "SNMP_AGENT",
        "snmp_oid": "1.3.6.1.4.1.14988.1.1.7.4.0",
        "key": "system.hw.firmware",
        "delay": "1h",
        "history": "7d",
        "trends": "0",
        "value_type": "CHAR",
        "description": "MIB: MIKROTIK-MIB\nCurrent firmware version.",
        "preprocessing": [
          {
            "type": "DISCARD_UNCHANGED_HEARTBEAT",
            "parameters": [
              "1d"
            ]
          }
        ],
        "tags": [
          {
            "tag": "component",
            "value": "system"
          }
        ],
        "triggers": [
          {
            "uuid": "2349e4eac44548349cc9f7806f8726db",
            "expression": "last(/MikroTik/system.hw.firmware,#1)<>last(/MikroTik/system.hw.firmware,#2) and length(last(/MikroTik/system.hw.firmware))>0",
            "name": "Mikrotik: Firmware has changed",
            "opdata": "Current value: {ITEM.LASTVALUE1}",
            "priority": "INFO",
            "description": "Firmware version has changed. Acknowledge to close the problem manually.",
            "manual_close": "YES",
            "tags": [
              {
                "tag": "scope",
                "value": "notice"
              }
            ]
          }
        ]
      },
      {
        "uuid": "692ccc1f29fe4f15ae7ca0bee54bf6b6",
        "name": "Mikrotik: Hardware model name",
        "type": "SNMP_AGENT",
        "snmp_oid": "1.3.6.1.2.1.1.1.0",
        "key": "system.hw.model",
        "delay": "1h",
        "history": "7d",
        "trends": "0",
```

```
            "value_type": "CHAR",
            "inventory_link": "MODEL",
            "preprocessing": [
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "1d"
                ]
              }
            ],
            "tags": [
              {
                "tag": "component",
                "value": "system"
              }
            ]
          },
          {
            "uuid": "0645ecb1a28040d9b89af280ca3093ea",
            "name": "Mikrotik: Hardware serial number",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.4.1.14988.1.1.7.3.0",
            "key": "system.hw.serialnumber",
            "delay": "1h",
            "history": "7d",
            "trends": "0",
            "value_type": "CHAR",
            "description": "MIB: MIKROTIK-MIB\nRouterBOARD serial number.",
            "inventory_link": "SERIALNO_A",
            "preprocessing": [
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "1d"
                ]
              }
            ],
            "tags": [
              {
                "tag": "component",
                "value": "system"
              }
            ],
            "triggers": [
              {
                "uuid": "f43a3aa77c8e4e9f81add08ac624be88",
                "expression": "last(/MikroTik/system.hw.serialnumber,#1)<>last(/MikroTik/system.hw.serialnumber,#2) and
length(last(/MikroTik/system.hw.serialnumber))>0",
                "name": "Mikrotik: Device has been replaced",
                "event_name": "Mikrotik: Device has been replaced (new serial number received)",
                "priority": "INFO",
                "description": "Device serial number has changed. Acknowledge to close the problem manually.",
                "manual_close": "YES",
                "tags": [
                  {
                    "tag": "scope",
                    "value": "notice"
                  }
                ]
              }
            ]
          },
          {
            "uuid": "806003915aa24c368a822a424f123d45",
            "name": "Mikrotik: Uptime (hardware)",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.25.1.1.0",
            "key": "system.hw.uptime[hrSystemUptime.0]",
            "delay": "30s",
            "history": "7d",
            "trends": "0",
            "units": "uptime",
            "description": "MIB: HOST-RESOURCES-MIB\nThe amount of time since this host was last initialized. Note that this is different
from sysUpTime in the SNMPv2-MIB [RFC1907] because sysUpTime is the uptime of the network management portion of the system.",
```

```
            "preprocessing": [
              {
                "type": "CHECK_NOT_SUPPORTED",
                "parameters": [
                  ""
                ],
                "error_handler": "CUSTOM_VALUE",
                "error_handler_params": "0"
              },
              {
                "type": "MULTIPLIER",
                "parameters": [
                  "0.01"
                ]
              }
            ],
            "tags": [
              {
                "tag": "component",
                "value": "system"
              }
            ]
          },
          {
            "uuid": "59b84e78871849fd8e2da3608c4fd45f",
            "name": "Mikrotik: System location",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.1.6.0",
            "key": "system.location[sysLocation.0]",
            "delay": "15m",
            "history": "7d",
            "trends": "0",
            "value_type": "CHAR",
            "description": "MIB: SNMPv2-MIB\nThe physical location of this node (e.g., `telephone closet, 3rd floor').  If the location is unknown, the value is the zero-length string.",
            "inventory_link": "LOCATION",
            "preprocessing": [
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "12h"
                ]
              }
            ],
            "tags": [
              {
                "tag": "component",
                "value": "system"
              }
            ]
          },
          {
            "uuid": "e68119b9b48e4a49a4ff3ce41b8bf613",
            "name": "Mikrotik: System name",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.1.5.0",
            "key": "system.name",
            "delay": "15m",
            "history": "7d",
            "trends": "0",
            "value_type": "CHAR",
            "description": "MIB: SNMPv2-MIB\nAn administratively-assigned name for this managed node.By convention, this is the node's fully-qualified domain name.  If the name is unknown, the value is the zero-length string.",
            "inventory_link": "NAME",
            "preprocessing": [
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "12h"
                ]
              }
            ],
            "tags": [
              {
```

```
                "tag": "component",
                "value": "system"
              }
            ],
            "triggers": [
              {
                "uuid": "f64be38a3557482087d75ed423ced500",
                "expression": "last(/MikroTik/system.name,#1)<>last(/MikroTik/system.name,#2) and
length(last(/MikroTik/system.name))>0",
                "name": "Mikrotik: System name has changed",
                "event_name": "Mikrotik: System name has changed (new name: {ITEM.VALUE})",
                "priority": "INFO",
                "description": "The name of the system has changed. Acknowledge to close the problem manually.",
                "manual_close": "YES",
                "tags": [
                  {
                    "tag": "scope",
                    "value": "notice"
                  },
                  {
                    "tag": "scope",
                    "value": "security"
                  }
                ]
              }
            ]
          },
          {
            "uuid": "56f939dcf88843fda2fe4b062cca2d6a",
            "name": "Mikrotik: Uptime (network)",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.1.3.0",
            "key": "system.net.uptime[sysUpTime.0]",
            "delay": "30s",
            "history": "7d",
            "trends": "0",
            "units": "uptime",
            "description": "MIB: SNMPv2-MIB\nThe time (in hundredths of a second) since the network management portion of the system
was last re-initialized.",
            "preprocessing": [
              {
                "type": "MULTIPLIER",
                "parameters": [
                  "0.01"
                ]
              }
            ],
            "tags": [
              {
                "tag": "component",
                "value": "system"
              }
            ]
          },
          {
            "uuid": "08476c93ef374c8baba1e6f80c5544cf",
            "name": "Mikrotik: System object ID",
            "type": "SNMP_AGENT",
            "snmp_oid": "1.3.6.1.2.1.1.2.0",
            "key": "system.objectid[sysObjectID.0]",
            "delay": "15m",
            "history": "7d",
            "trends": "0",
            "value_type": "CHAR",
            "description": "MIB: SNMPv2-MIB\nThe vendor's authoritative identification of the network management subsystem contained in
the entity.  This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for
determining`what kind of box' is being managed.  For example, if vendor`Flintstones, Inc.' was assigned the subtree1.3.6.1.4.1.4242, it could
assign the identifier 1.3.6.1.4.1.4242.1.1 to its `Fred Router'.",
            "preprocessing": [
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "12h"
                ]
```

```
            }
          ],
          "tags": [
            {
              "tag": "component",
              "value": "system"
            }
          ]
        },
        {
          "uuid": "551f25631a13497b924c1b285b93ee5f",
          "name": "Mikrotik: Operating system",
          "type": "SNMP_AGENT",
          "snmp_oid": "1.3.6.1.4.1.14988.1.1.4.4.0",
          "key": "system.sw.os[mtxrLicVersion.0]",
          "delay": "1h",
          "history": "7d",
          "trends": "0",
          "value_type": "CHAR",
          "description": "MIB: MIKROTIK-MIB\nSoftware version.",
          "inventory_link": "OS",
          "preprocessing": [
            {
              "type": "DISCARD_UNCHANGED_HEARTBEAT",
              "parameters": [
                "1d"
              ]
            }
          ],
          "tags": [
            {
              "tag": "component",
              "value": "os"
            }
          ],
          "triggers": [
            {
              "uuid": "76d4c429deaf491982b7b1049634550a",
              "expression": "last(/MikroTik/system.sw.os[mtxrLicVersion.0],#1)<>last(/MikroTik/system.sw.os[mtxrLicVersion.0],#2) and
length(last(/MikroTik/system.sw.os[mtxrLicVersion.0]))>0",
              "name": "Mikrotik: Operating system description has changed",
              "priority": "INFO",
              "description": "The description of the operating system has changed. Possible reasons are that the system has been updated
or replaced. Acknowledge to close the problem manually.",
              "manual_close": "YES",
              "dependencies": [
                {
                  "name": "Mikrotik: System name has changed",
                  "expression": "last(/MikroTik/system.name,#1)<>last(/MikroTik/system.name,#2) and
length(last(/MikroTik/system.name))>0"
                }
              ],
              "tags": [
                {
                  "tag": "scope",
                  "value": "notice"
                }
              ]
            }
          ]
        },
        {
          "uuid": "c654378ae0234392b1526df325125b0a",
          "name": "Mikrotik: SNMP walk mounted filesystems",
          "type": "SNMP_AGENT",
          "snmp_oid": "walk[1.3.6.1.2.1.25.2.3.1.3,1.3.6.1.2.1.25.2.3.1.4,1.3.6.1.2.1.25.2.3.1.2,1.3.6.1.2.1.25.2.3.1.6,1.3.6.1.2.1.25. 2.3.1.5]",
          "key": "vfs.fs.walk",
          "history": "0",
          "trends": "0",
          "value_type": "TEXT",
          "description": "MIB: HOST-RESOURCES-MIB\nDiscovering mounted filesystems.",
          "tags": [
            {
              "tag": "component",
```

```
            "value": "raw"
          }
        ]
      },
      {
        "uuid": "c077abb2ecc84c88b2173bff8e9ab9aa",
        "name": "Mikrotik: Total memory",
        "type": "SNMP_AGENT",
        "snmp_oid": "1.3.6.1.2.1.25.2.3.1.5.65536",
        "key": "vm.memory.total[hrStorageSize.Memory]",
        "history": "7d",
        "units": "B",
        "description": "MIB: HOST-RESOURCES-MIB\nThe size of the storage represented by this entry, in\nunits of
hrStorageAllocationUnits. This object is\nwritable to allow remote configuration of the size of\nthe storage area in those cases where such
an\noperation makes sense and is possible on the\nunderlying system. For example, the amount of main\nmemory allocated to a buffer pool
might be modified or\nthe amount of disk space allocated to virtual memory\nmight be modified.",
        "preprocessing": [
          {
            "type": "MULTIPLIER",
            "parameters": [
              "1024"
            ]
          }
        ],
        "tags": [
          {
            "tag": "component",
            "value": "memory"
          }
        ]
      },
      {
        "uuid": "4029907022564beeb405c065ae4febff",
        "name": "Mikrotik: Used memory",
        "type": "SNMP_AGENT",
        "snmp_oid": "1.3.6.1.2.1.25.2.3.1.6.65536",
        "key": "vm.memory.used[hrStorageUsed.Memory]",
        "history": "7d",
        "units": "B",
        "description": "MIB: HOST-RESOURCES-MIB\nThe amount of the storage represented by this entry that is allocated, in units of
hrStorageAllocationUnits.",
        "preprocessing": [
          {
            "type": "MULTIPLIER",
            "parameters": [
              "1024"
            ]
          }
        ],
        "tags": [
          {
            "tag": "component",
            "value": "memory"
          }
        ]
      },
      {
        "uuid": "5521e149584846118dcb280496b3aca4",
        "name": "Mikrotik: Memory utilization",
        "type": "CALCULATED",
        "key": "vm.memory.util[memoryUsedPercentage.Memory]",
        "history": "7d",
        "value_type": "FLOAT",
        "units": "%",
        "params": "last(//vm.memory.used[hrStorageUsed.Memory])/last(//vm.memory.total[hrStorageSize.Memory])*100",
        "description": "Memory utilization in %.",
        "tags": [
          {
            "tag": "component",
            "value": "memory"
          }
        ],
        "triggers": [
          {
```

```
            "uuid": "cfca5a71a1534e199e9c3b867b8c6c3b",
            "expression": "min(/MikroTik/vm.memory.util[memoryUsedPercentage.Memory],5m)>{$MEMORY.UTIL.MAX}",
            "name": "Mikrotik: High memory utilization",
            "event_name": "Mikrotik: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)",
            "priority": "AVERAGE",
            "description": "The system is running out of free memory.",
            "tags": [
              {
                "tag": "scope",
                "value": "capacity"
              },
              {
                "tag": "scope",
                "value": "performance"
              }
            ]
          }
        ]
      },
      {
        "uuid": "33e8ea0575174bc69002753a5e5d90bd",
        "name": "Mikrotik: SNMP agent availability",
        "type": "INTERNAL",
        "key": "zabbix[host,snmp,available]",
        "history": "7d",
        "description": "Availability of SNMP checks on the host. The value of this item corresponds to availability icons in the host list.\nPossible value:\n0 - not available\n1 - available\n2 - unknown",
        "valuemap": {
          "name": "zabbix.host.available"
        },
        "tags": [
          {
            "tag": "component",
            "value": "health"
          },
          {
            "tag": "component",
            "value": "network"
          }
        ],
        "triggers": [
          {
            "uuid": "81af7d744ebf422ba065308fcb3c225a",
            "expression": "max(/MikroTik/zabbix[host,snmp,available],{$SNMP.TIMEOUT})=0",
            "name": "Mikrotik: No SNMP data collection",
            "opdata": "Current state: {ITEM.LASTVALUE1}",
            "priority": "WARNING",
            "description": "SNMP is not available for polling. Please check device connectivity and SNMP settings.",
            "dependencies": [
              {
                "name": "Mikrotik: Unavailable by ICMP ping",
                "expression": "max(/MikroTik/icmpping,#3)=0"
              }
            ],
            "tags": [
              {
                "tag": "scope",
                "value": "availability"
              }
            ]
          }
        ]
      }
    ],
    "discovery_rules": [
      {
        "uuid": "4de6b86871874118b83ed7cfd31f3113",
        "name": "CPU discovery",
        "type": "DEPENDENT",
        "key": "hrProcessorLoad.discovery",
        "delay": "0",
        "description": "HOST-RESOURCES-MIB::hrProcessorTable discovery.",
        "item_prototypes": [
          {
```

```
                "uuid": "4dcba527c8de49eb90a9cc58e881202e",
                "name": "#{#SNMPINDEX}: CPU utilization",
                "type": "DEPENDENT",
                "key": "system.cpu.util[hrProcessorLoad.{#SNMPINDEX}]",
                "delay": "0",
                "history": "7d",
                "value_type": "FLOAT",
                "units": "%",
                "description": "MIB: HOST-RESOURCES-MIB\nThe average, over the last minute, of the percentage of time that this processor was not idle. Implementations may approximate this one minute smoothing period if necessary.",
                "preprocessing": [
                  {
                    "type": "SNMP_WALK_VALUE",
                    "parameters": [
                      "1.3.6.1.2.1.25.3.3.1.2.{#SNMPINDEX}",
                      "0"
                    ]
                  }
                ],
                "master_item": {
                  "key": "system.cpu.walk"
                },
                "tags": [
                  {
                    "tag": "component",
                    "value": "cpu"
                  }
                ],
                "trigger_prototypes": [
                  {
                    "uuid": "1edf137678504b9d8dced37d9a1d3f5a",
                    "expression": "min(/MikroTik/system.cpu.util[hrProcessorLoad.{#SNMPINDEX}],5m)>{$CPU.UTIL.CRIT}",
                    "name": "#{#SNMPINDEX}: High CPU utilization",
                    "event_name": "#{#SNMPINDEX}: High CPU utilization (over {$CPU.UTIL.CRIT}% for 5m)",
                    "opdata": "Current utilization: {ITEM.LASTVALUE1}",
                    "priority": "WARNING",
                    "description": "CPU utilization is too high. The system might be slow to respond.",
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "performance"
                      }
                    ]
                  }
                ]
              }
            ],
            "graph_prototypes": [
              {
                "uuid": "28b696fbb7ce4cf4941932ceb04700cb",
                "name": "#{#SNMPINDEX}: CPU utilization",
                "ymin_type_1": "FIXED",
                "ymax_type_1": "FIXED",
                "graph_items": [
                  {
                    "drawtype": "GRADIENT_LINE",
                    "color": "199C0D",
                    "item": {
                      "host": "MikroTik",
                      "key": "system.cpu.util[hrProcessorLoad.{#SNMPINDEX}]"
                    }
                  }
                ]
              }
            ],
            "master_item": {
              "key": "system.cpu.walk"
            },
            "preprocessing": [
              {
                "type": "SNMP_WALK_TO_JSON",
                "parameters": [
                  "{#SNMPVALUE}",
                  "1.3.6.1.2.1.25.3.3.1.1",
```

```
                "0"
              ]
            },
            {
              "type": "DISCARD_UNCHANGED_HEARTBEAT",
              "parameters": [
                "1h"
              ]
            }
          ]
        },
        {
          "uuid": "e738626cc9af4c9ba47f18ede6d6ffa3",
          "name": "Temperature CPU discovery",
          "type": "DEPENDENT",
          "key": "mtxrHlProcessorTemperature.discovery",
          "delay": "0",
          "description": "MIKROTIK-MIB::mtxrHlProcessorTemperature\nSince temperature of CPU is not available on all Mikrotik
hardware, this is done to avoid unsupported items.",
          "item_prototypes": [
            {
              "uuid": "0f06d2c64a2f4b6690736e3f74ea8e59",
              "name": "CPU: Temperature",
              "type": "DEPENDENT",
              "key": "sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "value_type": "FLOAT",
              "units": "\u00b0C",
              "description": "MIB: MIKROTIK-MIB\nmtxrHlProcessorTemperature Processor temperature in Celsius (degrees C).\nMight be
missing in entry models (RB750, RB450G..).",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
                  "parameters": [
                    "1.3.6.1.4.1.14988.1.1.3.11.{#SNMPINDEX}",
                    "0"
                  ]
                },
                {
                  "type": "MULTIPLIER",
                  "parameters": [
                    "0.1"
                  ]
                },
                {
                  "type": "DISCARD_UNCHANGED_HEARTBEAT",
                  "parameters": [
                    "3m"
                  ]
                }
              ],
              "master_item": {
                "key": "sensor.cpu.temp.walk"
              },
              "tags": [
                {
                  "tag": "component",
                  "value": "temperature"
                }
              ],
              "trigger_prototypes": [
                {
                  "uuid": "e9ab4a4bc78e47a19409d2bfe4d1d5fa",
                  "expression":
"avg(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)>{$TEMP_CRIT:\"CPU\"}",
                  "recovery_mode": "RECOVERY_EXPRESSION",
                  "recovery_expression":
"max(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)<{$TEMP_CRIT:\"CPU\"}-3",
                  "name": "CPU: Temperature is above critical threshold",
                  "event_name": "CPU: Temperature is above critical threshold: >{$TEMP_CRIT:\"CPU\"}",
                  "opdata": "Current value: {ITEM.LASTVALUE1}",
                  "priority": "HIGH",
                  "description": "This trigger uses temperature sensor values as well as temperature sensor status if available.",
```

```
                        "tags": [
                          {
                            "tag": "scope",
                            "value": "availability"
                          },
                          {
                            "tag": "scope",
                            "value": "performance"
                          }
                        ]
                      },
                      {
                        "uuid": "1e81f6308a2748adbed17064d648b566",
                        "expression":
"avg(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)>{$TEMP_WARN:\"CPU\"}",
                        "recovery_mode": "RECOVERY_EXPRESSION",
                        "recovery_expression":
"max(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)<{$TEMP_WARN:\"CPU\"}-3",
                        "name": "CPU: Temperature is above warning threshold",
                        "event_name": "CPU: Temperature is above warning threshold: >{$TEMP_WARN:\"CPU\"}",
                        "opdata": "Current value: {ITEM.LASTVALUE1}",
                        "priority": "WARNING",
                        "description": "This trigger uses temperature sensor values as well as temperature sensor status if available.",
                        "dependencies": [
                          {
                            "name": "CPU: Temperature is above critical threshold",
                            "expression":
"avg(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)>{$TEMP_CRIT:\"CPU\"}",
                            "recovery_expression":
"max(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)<{$TEMP_CRIT:\"CPU\"}-3"
                          }
                        ],
                        "tags": [
                          {
                            "tag": "scope",
                            "value": "availability"
                          },
                          {
                            "tag": "scope",
                            "value": "performance"
                          }
                        ]
                      },
                      {
                        "uuid": "3a679a3d653d42cabed24e238fa01c38",
                        "expression":
"avg(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)<{$TEMP_CRIT_LOW:\"CPU\"}",
                        "recovery_mode": "RECOVERY_EXPRESSION",
                        "recovery_expression":
"min(/MikroTik/sensor.temp.value[mtxrHlProcessorTemperature.{#SNMPINDEX}],5m)>{$TEMP_CRIT_LOW:\"CPU\"}+3",
                        "name": "CPU: Temperature is too low",
                        "event_name": "CPU: Temperature is too low: <{$TEMP_CRIT_LOW:\"CPU\"}",
                        "opdata": "Current value: {ITEM.LASTVALUE1}",
                        "priority": "AVERAGE",
                        "tags": [
                          {
                            "tag": "scope",
                            "value": "availability"
                          },
                          {
                            "tag": "scope",
                            "value": "performance"
                          }
                        ]
                      }
                    ]
                  }
                ],
                "master_item": {
                  "key": "sensor.cpu.temp.walk"
                },
                "preprocessing": [
                  {
                    "type": "SNMP_WALK_TO_JSON",
```

```
                "parameters": [
                  "{#SNMPVALUE}",
                  "1.3.6.1.4.1.14988.1.1.3.11",
                  "0"
                ],
                "error_handler": "DISCARD_VALUE"
              },
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "1h"
                ]
              }
            ]
          },
          {
            "uuid": "6a9998434d8b4a93a7fba87eaf642600",
            "name": "Temperature sensor discovery",
            "type": "DEPENDENT",
            "key": "mtxrHlTemperature.discovery",
            "delay": "0",
            "description": "MIKROTIK-MIB::mtxrHlTemperature\nSince temperature sensor is not available on all Mikrotik hardware, this is done to avoid unsupported items.",
            "item_prototypes": [
              {
                "uuid": "53d76ef4e9224a63bd30e3d1fc89542f",
                "name": "Device: Temperature",
                "type": "DEPENDENT",
                "key": "sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}]",
                "delay": "0",
                "history": "7d",
                "value_type": "FLOAT",
                "units": "\u00b0C",
                "description": "MIB: MIKROTIK-MIB\nmtxrHlTemperature Device temperature in Celsius (degrees C).\nMight be missing in entry models (RB750, RB450G..).\n\nReference: http://wiki.mikrotik.com/wiki/Manual:SNMP",
                "preprocessing": [
                  {
                    "type": "SNMP_WALK_VALUE",
                    "parameters": [
                      "1.3.6.1.4.1.14988.1.1.3.10.{#SNMPINDEX}",
                      "0"
                    ]
                  },
                  {
                    "type": "MULTIPLIER",
                    "parameters": [
                      "0.1"
                    ]
                  },
                  {
                    "type": "DISCARD_UNCHANGED_HEARTBEAT",
                    "parameters": [
                      "3m"
                    ]
                  }
                ],
                "master_item": {
                  "key": "sensor.temp.walk"
                },
                "tags": [
                  {
                    "tag": "component",
                    "value": "temperature"
                  }
                ],
                "trigger_prototypes": [
                  {
                    "uuid": "88fc1612d1b14108a1d90313fa1cb0e7",
                    "expression": "avg(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)>{$TEMP_CRIT:\"Device\"}",
                    "recovery_mode": "RECOVERY_EXPRESSION",
                    "recovery_expression": "max(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)<{$TEMP_CRIT:\"Device\"}-3",
                    "name": "Device: Temperature is above critical threshold",
                    "event_name": "Device: Temperature is above critical threshold: >{$TEMP_CRIT:\"Device\"}",
```

```
                    "opdata": "Current value: {ITEM.LASTVALUE1}",
                    "priority": "HIGH",
                    "description": "This trigger uses temperature sensor values as well as temperature sensor status if available.",
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "availability"
                      },
                      {
                        "tag": "scope",
                        "value": "performance"
                      }
                    ]
                  },
                  {
                    "uuid": "048ffb5e510746e6b56daec287db6739",
                    "expression": "avg(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)>{$TEMP_WARN:\"Device\"}",
                    "recovery_mode": "RECOVERY_EXPRESSION",
                    "recovery_expression":
"max(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)<{$TEMP_WARN:\"Device\"}-3",
                    "name": "Device: Temperature is above warning threshold",
                    "event_name": "Device: Temperature is above warning threshold: >{$TEMP_WARN:\"Device\"}",
                    "opdata": "Current value: {ITEM.LASTVALUE1}",
                    "priority": "WARNING",
                    "description": "This trigger uses temperature sensor values as well as temperature sensor status if available.",
                    "dependencies": [
                      {
                        "name": "Device: Temperature is above critical threshold",
                        "expression":
"avg(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)>{$TEMP_CRIT:\"Device\"}",
                        "recovery_expression":
"max(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)<{$TEMP_CRIT:\"Device\"}-3"
                      }
                    ],
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "availability"
                      },
                      {
                        "tag": "scope",
                        "value": "performance"
                      }
                    ]
                  },
                  {
                    "uuid": "fcc70d3662ac49499a0f5bb68a13c007",
                    "expression":
"avg(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)<{$TEMP_CRIT_LOW:\"Device\"}",
                    "recovery_mode": "RECOVERY_EXPRESSION",
                    "recovery_expression":
"min(/MikroTik/sensor.temp.value[mtxrHlTemperature.{#SNMPINDEX}],5m)>{$TEMP_CRIT_LOW:\"Device\"}+3",
                    "name": "Device: Temperature is too low",
                    "event_name": "Device: Temperature is too low: <{$TEMP_CRIT_LOW:\"Device\"}",
                    "opdata": "Current value: {ITEM.LASTVALUE1}",
                    "priority": "AVERAGE",
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "availability"
                      },
                      {
                        "tag": "scope",
                        "value": "performance"
                      }
                    ]
                  }
                ]
              }
            ],
            "master_item": {
              "key": "sensor.temp.walk"
            },
            "preprocessing": [
```

```
      {
        "type": "SNMP_WALK_TO_JSON",
        "parameters": [
          "{#SNMPVALUE}",
          "1.3.6.1.4.1.14988.1.1.3.10",
          "0"
        ],
        "error_handler": "DISCARD_VALUE"
      },
      {
        "type": "DISCARD_UNCHANGED_HEARTBEAT",
        "parameters": [
          "1h"
        ]
      }
    ]
  },
  {
    "uuid": "53adc733f3c34ddd943e5ccb41d1b536",
    "name": "LTE modem discovery",
    "type": "DEPENDENT",
    "key": "mtxrLTEModem.discovery",
    "delay": "0",
    "filter": {
      "evaltype": "AND",
      "conditions": [
        {
          "macro": "{#IFNAME}",
          "value": "{$IFNAME.LTEMODEM.MATCHES}",
          "formulaid": "A"
        },
        {
          "macro": "{#IFTYPE}",
          "value": "^1$",
          "formulaid": "B"
        }
      ]
    },
    "description": "MIKROTIK-MIB::mtxrLTEModemInterfaceIndex.",
    "item_prototypes": [
      {
        "uuid": "23c7a946500a420a87ead3d7c923f176",
        "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSRP",
        "type": "DEPENDENT",
        "key": "lte.modem.rsrp[mtxrLTEModemSignalRSRP.{#SNMPINDEX}]",
        "delay": "0",
        "history": "7d",
        "value_type": "FLOAT",
        "units": "dbm",
        "description": "MIB: MIKROTIK-MIB\nmtxrLTEModemSignalRSRP Reference Signal Received Power.",
        "preprocessing": [
          {
            "type": "SNMP_WALK_VALUE",
            "parameters": [
              "1.3.6.1.4.1.14988.1.1.16.1.1.4.{#SNMPINDEX}",
              "0"
            ]
          }
        ],
        "master_item": {
          "key": "net.if.wireless.walk"
        },
        "tags": [
          {
            "tag": "component",
            "value": "interface"
          },
          {
            "tag": "component",
            "value": "network"
          },
          {
            "tag": "interface",
            "value": "{#IFNAME}"
```

```
                }
              ],
              "trigger_prototypes": [
                {
                  "uuid": "d9acbd875f514ce8a19638125c096aed",
                  "expression": "max(/MikroTik/lte.modem.rsrp[mtxrLTEModemSignalRSRP.{#SNMPINDEX}],5m) <
{$LTEMODEM.RSRP.MIN.WARN}",
                  "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSRP is low",
                  "event_name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSRP is low (below {$LTEMODEM.RSRP.MIN.WARN}dbm
for 5m)",
                  "priority": "WARNING",
                  "tags": [
                    {
                      "tag": "scope",
                      "value": "availability"
                    },
                    {
                      "tag": "scope",
                      "value": "performance"
                    }
                  ]
                }
              ]
            },
            {
              "uuid": "93254f62b3e74acaa3c5839b9f844363",
              "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSRQ",
              "type": "DEPENDENT",
              "key": "lte.modem.rsrq[mtxrLTEModemSignalRSRQ.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "value_type": "FLOAT",
              "units": "db",
              "description": "MIB: MIKROTIK-MIB\nmtxrLTEModemSignalRSRQ Reference Signal Received Quality.",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
                  "parameters": [
                    "1.3.6.1.4.1.14988.1.1.16.1.1.3.{#SNMPINDEX}",
                    "0"
                  ]
                }
              ],
              "master_item": {
                "key": "net.if.wireless.walk"
              },
              "tags": [
                {
                  "tag": "component",
                  "value": "interface"
                },
                {
                  "tag": "component",
                  "value": "network"
                },
                {
                  "tag": "interface",
                  "value": "{#IFNAME}"
                }
              ],
              "trigger_prototypes": [
                {
                  "uuid": "569c192c940a49f7b180162e4b83be24",
                  "expression": "max(/MikroTik/lte.modem.rsrq[mtxrLTEModemSignalRSRQ.{#SNMPINDEX}],5m) <
{$LTEMODEM.RSRQ.MIN.WARN}",
                  "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSRQ is low",
                  "event_name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSRQ is low (below {$LTEMODEM.RSRQ.MIN.WARN}db for
5m)",
                  "priority": "WARNING",
                  "tags": [
                    {
                      "tag": "scope",
                      "value": "availability"
                    },
```

```
                {
                  "tag": "scope",
                  "value": "performance"
                }
              ]
            }
          ]
        },
        {
          "uuid": "67eb22e623174b06a24460985521d3d4",
          "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSSI",
          "type": "DEPENDENT",
          "key": "lte.modem.rssi[mtxrLTEModemSignalRSSI.{#SNMPINDEX}]",
          "delay": "0",
          "history": "7d",
          "value_type": "FLOAT",
          "units": "dbm",
          "description": "MIB: MIKROTIK-MIB\nmtxrLTEModemSignalRSSI Received Signal Strength Indicator.",
          "preprocessing": [
            {
              "type": "SNMP_WALK_VALUE",
              "parameters": [
                "1.3.6.1.4.1.14988.1.1.16.1.1.2.{#SNMPINDEX}",
                "0"
              ]
            }
          ],
          "master_item": {
            "key": "net.if.wireless.walk"
          },
          "tags": [
            {
              "tag": "component",
              "value": "interface"
            },
            {
              "tag": "component",
              "value": "network"
            },
            {
              "tag": "interface",
              "value": "{#IFNAME}"
            }
          ],
          "trigger_prototypes": [
            {
              "uuid": "7b31d412d15140d7be66d4b9963150de",
              "expression": "max(/MikroTik/lte.modem.rssi[mtxrLTEModemSignalRSSI.{#SNMPINDEX}],5m) <
{$LTEMODEM.RSSI.MIN.WARN}",
              "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSSI is low",
              "event_name": "Interface {#IFNAME}({#IFALIAS}): LTE modem RSSI is low (below {$LTEMODEM.RSSI.MIN.WARN}dbm for
5m)",
              "priority": "WARNING",
              "tags": [
                {
                  "tag": "scope",
                  "value": "availability"
                },
                {
                  "tag": "scope",
                  "value": "performance"
                }
              ]
            }
          ]
        },
        {
          "uuid": "dda3ef44c9204c46a8affb68aeeec803",
          "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem SINR",
          "type": "DEPENDENT",
          "key": "lte.modem.sinr[mtxrLTEModemSignalSINR.{#SNMPINDEX}]",
          "delay": "0",
          "history": "7d",
          "value_type": "FLOAT",
```

```
                "units": "db",
                "description": "MIB: MIKROTIK-MIB\nmtxrLTEModemSignalSINR Signal to Interference & Noise Ratio.",
                "preprocessing": [
                  {
                    "type": "SNMP_WALK_VALUE",
                    "parameters": [
                      "1.3.6.1.4.1.14988.1.1.16.1.1.7.{#SNMPINDEX}",
                      "0"
                    ]
                  }
                ],
                "master_item": {
                  "key": "net.if.wireless.walk"
                },
                "tags": [
                  {
                    "tag": "component",
                    "value": "interface"
                  },
                  {
                    "tag": "component",
                    "value": "network"
                  },
                  {
                    "tag": "interface",
                    "value": "{#IFNAME}"
                  }
                ],
                "trigger_prototypes": [
                  {
                    "uuid": "7512216ac6e24d5f88a187621d0d7285",
                    "expression": "max(/MikroTik/lte.modem.sinr[mtxrLTEModemSignalSINR.{#SNMPINDEX}],5m) <
{$LTEMODEM.SINR.MIN.WARN}",
                    "name": "Interface {#IFNAME}({#IFALIAS}): LTE modem SINR is low",
                    "event_name": "Interface {#IFNAME}({#IFALIAS}): LTE modem SINR is low (below {$LTEMODEM.SINR.MIN.WARN}db for
5m)",
                    "priority": "WARNING",
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "availability"
                      },
                      {
                        "tag": "scope",
                        "value": "performance"
                      }
                    ]
                  }
                ]
              }
            ],
            "graph_prototypes": [
              {
                "uuid": "360063285da04ed29523562086c769c4",
                "name": "Interface {#IFNAME}({#IFALIAS}): Modem Signal. {#SNMPINDEX}",
                "graph_items": [
                  {
                    "color": "199C0D",
                    "item": {
                      "host": "MikroTik",
                      "key": "lte.modem.sinr[mtxrLTEModemSignalSINR.{#SNMPINDEX}]"
                    }
                  },
                  {
                    "sortorder": "1",
                    "color": "F63100",
                    "item": {
                      "host": "MikroTik",
                      "key": "lte.modem.rsrq[mtxrLTEModemSignalRSRQ.{#SNMPINDEX}]"
                    }
                  },
                  {
                    "sortorder": "2",
                    "color": "00611C",
```

```
            "item": {
              "host": "MikroTik",
              "key": "lte.modem.rsrp[mtxrLTEModemSignalRSRP.{#SNMPINDEX}]"
            }
          },
          {
            "sortorder": "3",
            "color": "F7941D",
            "item": {
              "host": "MikroTik",
              "key": "lte.modem.rssi[mtxrLTEModemSignalRSSI.{#SNMPINDEX}]"
            }
          }
        ]
      }
    ],
    "master_item": {
      "key": "net.if.wireless.walk"
    },
    "preprocessing": [
      {
        "type": "SNMP_WALK_TO_JSON",
        "parameters": [
          "{#IFNAME}",
          "1.3.6.1.4.1.14988.1.1.14.1.1.2",
          "0",
          "{#IFALIAS}",
          "1.3.6.1.2.1.31.1.1.1.18",
          "0",
          "{#IFTYPE}",
          "1.3.6.1.2.1.2.2.1.3",
          "0"
        ]
      },
      {
        "type": "DISCARD_UNCHANGED_HEARTBEAT",
        "parameters": [
          "1h"
        ]
      }
    ]
  },
  {
    "uuid": "65ce926cc8bb4e55859204eb287c640b",
    "name": "AP channel discovery",
    "type": "DEPENDENT",
    "key": "mtxrWlAp.discovery",
    "delay": "0",
    "filter": {
      "evaltype": "AND",
      "conditions": [
        {
          "macro": "{#IFADMINSTATUS}",
          "value": "^1$",
          "formulaid": "A"
        },
        {
          "macro": "{#IFTYPE}",
          "value": "^71$",
          "formulaid": "B"
        }
      ]
    },
    "description": "MIKROTIK-MIB::mtxrWlAp.",
    "item_prototypes": [
      {
        "uuid": "bb7c28ff27104b0a8aa8f7052f71dd32",
        "name": "Interface {#IFNAME}({#IFALIAS}): AP authenticated clients",
        "type": "DEPENDENT",
        "key": "ssid.authclient[mtxrWlApAuthClientCount.{#SNMPINDEX}]",
        "delay": "0",
        "history": "7d",
        "description": "MIB: MIKROTIK-MIB\nmtxrWlApAuthClientCount Number of authentication clients.",
        "preprocessing": [
```

```
      {
        "type": "SNMP_WALK_VALUE",
        "parameters": [
          "1.3.6.1.4.1.14988.1.1.1.3.1.11.{#SNMPINDEX}",
          "0"
        ]
      }
    ],
    "master_item": {
      "key": "net.if.wireless.walk"
    },
    "tags": [
      {
        "tag": "component",
        "value": "interface"
      },
      {
        "tag": "component",
        "value": "network"
      },
      {
        "tag": "interface",
        "value": "{#IFNAME}"
      }
    ]
  },
  {
    "uuid": "437edde63aa74619a3f4531039c1e73d",
    "name": "Interface {#IFNAME}({#IFALIAS}): AP band",
    "type": "DEPENDENT",
    "key": "ssid.band[mtxrWlApBand.{#SNMPINDEX}]",
    "delay": "0",
    "history": "7d",
    "trends": "0",
    "value_type": "CHAR",
    "description": "MIB: MIKROTIK-MIB\nmtxrWlApBand",
    "preprocessing": [
      {
        "type": "SNMP_WALK_VALUE",
        "parameters": [
          "1.3.6.1.4.1.14988.1.1.1.3.1.8.{#SNMPINDEX}",
          "0"
        ]
      },
      {
        "type": "DISCARD_UNCHANGED_HEARTBEAT",
        "parameters": [
          "1h"
        ]
      }
    ],
    "master_item": {
      "key": "net.if.wireless.walk"
    },
    "tags": [
      {
        "tag": "component",
        "value": "interface"
      },
      {
        "tag": "component",
        "value": "network"
      },
      {
        "tag": "interface",
        "value": "{#IFNAME}"
      }
    ]
  },
  {
    "uuid": "091618c93d8a405a94d4147d18d985d7",
    "name": "Interface {#IFNAME}({#IFALIAS}): SSID",
    "type": "DEPENDENT",
    "key": "ssid.name[mtxrWlApSsid.{#SNMPINDEX}]",
```

```
    "delay": "0",
    "history": "7d",
    "trends": "0",
    "value_type": "CHAR",
    "description": "MIB: MIKROTIK-MIB\nmtxrWlApSsid Service Set Identifier.",
    "preprocessing": [
      {
        "type": "SNMP_WALK_VALUE",
        "parameters": [
          "1.3.6.1.4.1.14988.1.1.1.3.1.4.{#SNMPINDEX}",
          "0"
        ]
      },
      {
        "type": "DISCARD_UNCHANGED_HEARTBEAT",
        "parameters": [
          "1h"
        ]
      }
    ],
    "master_item": {
      "key": "net.if.wireless.walk"
    },
    "tags": [
      {
        "tag": "component",
        "value": "interface"
      },
      {
        "tag": "component",
        "value": "network"
      },
      {
        "tag": "interface",
        "value": "{#IFNAME}"
      }
    ]
  },
  {
    "uuid": "af82a882030b45ca90805a13bf4390f5",
    "name": "Interface {#IFNAME}({#IFALIAS}): AP noise floor",
    "type": "DEPENDENT",
    "key": "ssid.noise[mtxrWlApNoiseFloor.{#SNMPINDEX}]",
    "delay": "0",
    "history": "7d",
    "value_type": "FLOAT",
    "description": "MIB: MIKROTIK-MIB\nmtxrWlApNoiseFloor",
    "preprocessing": [
      {
        "type": "SNMP_WALK_VALUE",
        "parameters": [
          "1.3.6.1.4.1.14988.1.1.1.3.1.9.{#SNMPINDEX}",
          "0"
        ]
      },
      {
        "type": "DISCARD_UNCHANGED_HEARTBEAT",
        "parameters": [
          "15m"
        ]
      }
    ],
    "master_item": {
      "key": "net.if.wireless.walk"
    },
    "tags": [
      {
        "tag": "component",
        "value": "interface"
      },
      {
        "tag": "component",
        "value": "network"
      },
```

```
                {
                  "tag": "interface",
                  "value": "{#IFNAME}"
                }
              ]
            },
            {
              "uuid": "616f6787b7e8427aa727711c115afbed",
              "name": "Interface {#IFNAME}({#IFALIAS}): AP registered clients",
              "type": "DEPENDENT",
              "key": "ssid.regclient[mtxrWlApClientCount.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "description": "MIB: MIKROTIK-MIB\nmtxrWlApClientCount Client established connection to AP, but didn't finish all
authentication procedures for full connection.",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
                  "parameters": [
                    "1.3.6.1.4.1.14988.1.1.1.3.1.6.{#SNMPINDEX}",
                    "0"
                  ]
                }
              ],
              "master_item": {
                "key": "net.if.wireless.walk"
              },
              "tags": [
                {
                  "tag": "component",
                  "value": "interface"
                },
                {
                  "tag": "component",
                  "value": "network"
                },
                {
                  "tag": "interface",
                  "value": "{#IFNAME}"
                }
              ]
            }
          ],
          "master_item": {
            "key": "net.if.wireless.walk"
          },
          "preprocessing": [
            {
              "type": "SNMP_WALK_TO_JSON",
              "parameters": [
                "{#IFNAME}",
                "1.3.6.1.4.1.14988.1.1.14.1.1.2",
                "0",
                "{#IFALIAS}",
                "1.3.6.1.2.1.31.1.1.1.18",
                "0",
                "{#IFTYPE}",
                "1.3.6.1.2.1.2.2.1.3",
                "0",
                "{#IFADMINSTATUS}",
                "1.3.6.1.2.1.2.2.1.7",
                "0"
              ]
            },
            {
              "type": "DISCARD_UNCHANGED_HEARTBEAT",
              "parameters": [
                "1h"
              ]
            }
          ]
        },
        {
          "uuid": "9ab6be8cd3f44b7ba6df32589b844e77",
```

```
"name": "CAPsMAN AP channel discovery",
"type": "DEPENDENT",
"key": "mtxrWlCMChannel.discovery",
"delay": "0",
"filter": {
  "evaltype": "AND",
  "conditions": [
    {
      "macro": "{#IFNAME}",
      "value": "{$IFNAME.WIFI.MATCHES}",
      "formulaid": "A"
    },
    {
      "macro": "{#IFTYPE}",
      "value": "^1$",
      "formulaid": "B"
    }
  ]
},
"description": "MIKROTIK-MIB::mtxrWlCMChannel.",
"item_prototypes": [
  {
    "uuid": "29efb756bbc44a2bbd772633279cd754",
    "name": "Interface {#IFNAME}({#IFALIAS}): AP authenticated clients",
    "type": "DEPENDENT",
    "key": "ssid.authclient[mtxrWlCMAuthClientCount.{#SNMPINDEX}]",
    "delay": "0",
    "history": "7d",
    "description": "MIB: MIKROTIK-MIB\nmtxrWlCMAuthClientCount Number of authentication clients.",
    "preprocessing": [
      {
        "type": "SNMP_WALK_VALUE",
        "parameters": [
          "1.3.6.1.4.1.14988.1.1.1.7.1.3.{#SNMPINDEX}",
          "0"
        ]
      }
    ],
    "master_item": {
      "key": "net.if.wireless.walk"
    },
    "tags": [
      {
        "tag": "component",
        "value": "interface"
      },
      {
        "tag": "component",
        "value": "network"
      },
      {
        "tag": "interface",
        "value": "{#IFNAME}"
      }
    ]
  },
  {
    "uuid": "0921eb633d6d44f2add1031cbc5ae2fa",
    "name": "Interface {#IFNAME}({#IFALIAS}): AP channel",
    "type": "DEPENDENT",
    "key": "ssid.channel[mtxrWlCMChannel.{#SNMPINDEX}]",
    "delay": "0",
    "history": "7d",
    "trends": "0",
    "value_type": "CHAR",
    "description": "MIB: MIKROTIK-MIB\nmtxrWlCMChannel",
    "preprocessing": [
      {
        "type": "SNMP_WALK_VALUE",
        "parameters": [
          "1.3.6.1.4.1.14988.1.1.1.7.1.5.{#SNMPINDEX}",
          "0"
        ]
      },
```

```
                {
                  "type": "DISCARD_UNCHANGED_HEARTBEAT",
                  "parameters": [
                    "1h"
                  ]
                }
              ],
              "master_item": {
                "key": "net.if.wireless.walk"
              },
              "tags": [
                {
                  "tag": "component",
                  "value": "interface"
                },
                {
                  "tag": "component",
                  "value": "network"
                },
                {
                  "tag": "interface",
                  "value": "{#IFNAME}"
                }
              ]
            },
            {
              "uuid": "a7a9c42aa3054035b9190a38f3456e44",
              "name": "Interface {#IFNAME}({#IFALIAS}): AP registered clients",
              "type": "DEPENDENT",
              "key": "ssid.regclient[mtxrWlCMRegClientCount.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "description": "MIB: MIKROTIK-MIB\nmtxrWlCMRegClientCount Client established connection to AP, but didn't finish all authentication procedures for full connection.",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
                  "parameters": [
                    "1.3.6.1.4.1.14988.1.1.1.7.1.2.{#SNMPINDEX}",
                    "0"
                  ]
                }
              ],
              "master_item": {
                "key": "net.if.wireless.walk"
              },
              "tags": [
                {
                  "tag": "component",
                  "value": "interface"
                },
                {
                  "tag": "component",
                  "value": "network"
                },
                {
                  "tag": "interface",
                  "value": "{#IFNAME}"
                }
              ]
            },
            {
              "uuid": "23b17acfff4246208f744864e5af2108",
              "name": "Interface {#IFNAME}({#IFALIAS}): AP state",
              "type": "DEPENDENT",
              "key": "ssid.state[mtxrWlCMState.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "trends": "0",
              "value_type": "CHAR",
              "description": "MIB: MIKROTIK-MIB\nmtxrWlCMState Wireless interface state.",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
```

```
                    "parameters": [
                      "1.3.6.1.4.1.14988.1.1.1.7.1.4.{#SNMPINDEX}",
                      "0"
                    ]
                  },
                  {
                    "type": "DISCARD_UNCHANGED_HEARTBEAT",
                    "parameters": [
                      "1h"
                    ]
                  }
                ],
                "master_item": {
                  "key": "net.if.wireless.walk"
                },
                "tags": [
                  {
                    "tag": "component",
                    "value": "interface"
                  },
                  {
                    "tag": "component",
                    "value": "network"
                  },
                  {
                    "tag": "interface",
                    "value": "{#IFNAME}"
                  }
                ],
                "trigger_prototypes": [
                  {
                    "uuid": "bcb6d9302db14ee3a7dd120304b7a5a5",
                    "expression": "last(/MikroTik/ssid.state[mtxrWlCMState.{#SNMPINDEX}])<>\"running-ap\"",
                    "name": "Interface {#IFNAME}({#IFALIAS}): AP interface {#IFNAME}({#IFALIAS}) is not running",
                    "priority": "WARNING",
                    "description": "Access point interface can be not running by different reasons - disabled interface, power off, network
link down.",
                    "tags": [
                      {
                        "tag": "scope",
                        "value": "availability"
                      },
                      {
                        "tag": "scope",
                        "value": "performance"
                      }
                    ]
                  }
                ]
              }
            ],
            "master_item": {
              "key": "net.if.wireless.walk"
            },
            "preprocessing": [
              {
                "type": "SNMP_WALK_TO_JSON",
                "parameters": [
                  "{#IFNAME}",
                  "1.3.6.1.4.1.14988.1.1.14.1.1.2",
                  "0",
                  "{#IFALIAS}",
                  "1.3.6.1.2.1.31.1.1.1.18",
                  "0",
                  "{#IFTYPE}",
                  "1.3.6.1.2.1.2.2.1.3",
                  "0"
                ]
              },
              {
                "type": "DISCARD_UNCHANGED_HEARTBEAT",
                "parameters": [
                  "1h"
                ]
```

```
      }
    ]
  },
  {
    "uuid": "fee6ff9551144a1c9888afccc73cddc2",
    "name": "Network interfaces discovery",
    "type": "DEPENDENT",
    "key": "net.if.discovery",
    "delay": "0",
    "filter": {
      "evaltype": "AND",
      "conditions": [
        {
          "macro": "{#IFADMINSTATUS}",
          "value": "{$NET.IF.IFADMINSTATUS.MATCHES}",
          "formulaid": "A"
        },
        {
          "macro": "{#IFADMINSTATUS}",
          "value": "{$NET.IF.IFADMINSTATUS.NOT_MATCHES}",
          "operator": "NOT_MATCHES_REGEX",
          "formulaid": "B"
        },
        {
          "macro": "{#IFALIAS}",
          "value": "{$NET.IF.IFALIAS.MATCHES}",
          "formulaid": "C"
        },
        {
          "macro": "{#IFALIAS}",
          "value": "{$NET.IF.IFALIAS.NOT_MATCHES}",
          "operator": "NOT_MATCHES_REGEX",
          "formulaid": "D"
        },
        {
          "macro": "{#IFDESCR}",
          "value": "{$NET.IF.IFDESCR.MATCHES}",
          "formulaid": "E"
        },
        {
          "macro": "{#IFDESCR}",
          "value": "{$NET.IF.IFDESCR.NOT_MATCHES}",
          "operator": "NOT_MATCHES_REGEX",
          "formulaid": "F"
        },
        {
          "macro": "{#IFNAME}",
          "value": "{$NET.IF.IFNAME.MATCHES}",
          "formulaid": "G"
        },
        {
          "macro": "{#IFNAME}",
          "value": "{$NET.IF.IFNAME.NOT_MATCHES}",
          "operator": "NOT_MATCHES_REGEX",
          "formulaid": "H"
        },
        {
          "macro": "{#IFOPERSTATUS}",
          "value": "{$NET.IF.IFOPERSTATUS.MATCHES}",
          "formulaid": "I"
        },
        {
          "macro": "{#IFOPERSTATUS}",
          "value": "{$NET.IF.IFOPERSTATUS.NOT_MATCHES}",
          "operator": "NOT_MATCHES_REGEX",
          "formulaid": "J"
        },
        {
          "macro": "{#IFTYPE}",
          "value": "{$NET.IF.IFTYPE.MATCHES}",
          "formulaid": "K"
        },
        {
          "macro": "{#IFTYPE}",
```

```
                "value": "{$NET.IF.IFTYPE.NOT_MATCHES}",
                "operator": "NOT_MATCHES_REGEX",
                "formulaid": "L"
              }
            ]
          },
          "description": "Discovering interfaces from IF-MIB.",
          "item_prototypes": [
            {
              "uuid": "296295e0f95c4302a6b96216f8a3011e",
              "name": "Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded",
              "type": "DEPENDENT",
              "key": "net.if.in.discards[ifInDiscards.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "description": "MIB: IF-MIB\nThe number of inbound packets which were chosen to be discarded\neven though no errors had been detected to prevent their being deliverable to a higher-layer protocol.\nOne possible reason for discarding such a packet could be to free up buffer space.\nDiscontinuities in the value of this counter can occur at re-initialization of the management system,\nand at other times as indicated by the value of ifCounterDiscontinuityTime.",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
                  "parameters": [
                    "1.3.6.1.2.1.2.2.1.13.{#SNMPINDEX}",
                    "0"
                  ]
                },
                {
                  "type": "CHANGE_PER_SECOND",
                  "parameters": [
                    ""
                  ]
                },
                {
                  "type": "DISCARD_UNCHANGED_HEARTBEAT",
                  "parameters": [
                    "3m"
                  ]
                }
              ],
              "master_item": {
                "key": "net.if.walk"
              },
              "tags": [
                {
                  "tag": "component",
                  "value": "network"
                },
                {
                  "tag": "description",
                  "value": "{#IFALIAS}"
                },
                {
                  "tag": "interface",
                  "value": "{#IFNAME}"
                }
              ]
            },
            {
              "uuid": "59ec7c3ac8304d2885623ab64bad205e",
              "name": "Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors",
              "type": "DEPENDENT",
              "key": "net.if.in.errors[ifInErrors.{#SNMPINDEX}]",
              "delay": "0",
              "history": "7d",
              "description": "MIB: IF-MIB\nFor packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.  For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.",
              "preprocessing": [
                {
                  "type": "SNMP_WALK_VALUE",
                  "parameters": [
                    "1.3.6.1.2.1.2.2.1.14.{#SNMPINDEX}",
```

```
            "0"
          ]
        },
        {
          "type": "CHANGE_PER_SECOND",
          "parameters": [
            ""
          ]
        },
        {
          "type": "DISCARD_UNCHANGED_HEARTBEAT",
          "parameters": [
            "3m"
          ]
        }
      ],
      "master_item": {
        "key": "net.if.walk"
      },
      "tags": [
        {
          "tag": "component",
          "value": "network"
        },
        {
          "tag": "description",
          "value": "{#IFALIAS}"
        },
        {
          "tag": "interface",
          "value": "{#IFNAME}"
        }
      ]
    },
    {
      "uuid": "6590c395369244baa3a145197fb929b7",
      "name": "Interface {#IFNAME}({#IFALIAS}): Bits received",
      "type": "DEPENDENT",
      "key": "net.if.in[ifHCInOctets.{#SNMPINDEX}]",
      "delay": "0",
      "history": "7d",
      "units": "bps",
      "description": "MIB: IF-MIB\nThe total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.",
      "preprocessing": [
        {
          "type": "SNMP_WALK_VALUE",
          "parameters": [
            "1.3.6.1.2.1.31.1.1.1.6.{#SNMPINDEX}",
            "0"
          ]
        },
        {
          "type": "CHANGE_PER_SECOND",
          "parameters": [
            ""
          ]
        },
        {
          "type": "MULTIPLIER",
          "parameters": [
            "8"
          ]
```


```
ENABLE_TIMESCALEDB: true ZBX_STARTREPORTWRITERS: 2
#       Map container ports to the host machine ports:
-       "10051:10051"
#       Link the container to the mariadb container links:
-       mariadb
#       Depend on the mariadb container to start up first depends_on:
-       mariadb
```

```
    #  Zabbix agent container for monitoring host metrics zabbix-agent:
    image: zabbix/zabbix-agent:ubuntu-6.4-latest user: root networks:
    - zabbix-net
    #  Link the container to the zabbix-server container links:
    - zabbix-server
    #  Set privileged access mode for allowing resource access privileged: true
    environment:
    ZBX_HOSTNAME: Zabbix server ports:
    -  "10050:10050"
    #  Zabbix web interface container for viewing monitoring data zabbix-web:
    image: zabbix/zabbix-web-apache-mysql:ubuntu-6.4-latest ports:
    -  "80:8080"
    -  "443:8443"
    networks:
    - zabbix-net container_name: zabbix-web environment:
    DB_SERVER_HOST:  'mariadb'
    MYSQL_USER:  'root'
    MYSQL_PASSWORD:  '12345'
    ZBX_SERVER_HOST:  'zabbix-server'
    PHP_TZ: Europe/Kiev
    #  Set up links between the container and the mariadb and zabbix-server containers links:
    - mariadb
    - zabbix-server depends_on:
    - mariadb
    - zabbix-server
    mysql:
    image: mysql:5.7 environment:
    MYSQL_DATABASE:  'user'
    MYSQL_USER:  'mysql'
    MYSQL_PASSWORD:  'root'
    MYSQL_ROOT_PASSWORD:  'root'


    import enum
    #  define Role enum with two possible values class Role(enum.Enum):
    USER = "user"
    ADMIN = "admin"
    #  define Location enum with three possible values class Location(enum.Enum):
    EUROPE = "europe"
    ASIA = "asia"
    NORTH_AMERICA = "north america"
    #  define Metrics enum with different metrics that can be used to evaluate the system class Metrics(enum.Enum):
    TOTAb_NUMBER_OF_ACTIVE_CLIENTS = "total_number_of_active_clients"
    TOTAb_NUMBER_OF_ACTIVE_USERS = "total_number_of_active_users"
    TOTAIi_NUMBER_OF_ACTIVE_ADMINS = "total_number_of_active_admins"
    NUMBER_OF_CLIENTS_FROM_EUROPE = "number_of_clients_from_europe"
    NUMBER_OF_CLIENTS_FROM_ASIA = "number_of_clients_from_asia"
    NUMBER_OF_CLIENTS_FROM_NORTH_AMERICA = "number_of_clients_from_north._america"


    import requests import datetime
    from app.schema import UserLogin import time import random
    from app.utils import run_query
    #  utility function to get active users def get_active_users(is_active):
    query = "SELECT fullname, email, password, role, location FROM users WHERE is_active=%s" return run_query(query,
    (is_active,), False)
    #  function to simulate traffic def mock_traffic():
    start_time = datetime.datetime.now()
    end_time = start_time + datetime.timedelta(minutes=5) count = 1
while datetime.datetime.now() < end_time: print("...") login_users(count) logout_users(count) count +=
    random.randint(0, 4) time.sleep(5)
    #  function to simulate login actions for a specified number of users def login_users(count):
```

```
user_list = get_active_users(is_active=0) if user_list:
        for i in range(count + random.randint(0, 10)):
_, email, password, _, _ = user_list[i]
user = UserLogin(email=email, password=password)
response = requests.post("http://localhost:8001/user/login", user.json()) if response.status_code ==
200: print("login") else:
print("login error")

import datetime import time
from pyzabbix import ZabbixSender, ZabbixMetric
from app.utils import run_query
from app.enums import Role, Location, Metrics
#  the total number of clients who are active def get_login_clients():
query = "SELECT COUNT(*) FROM users WHERE is_active = 1" return run._query(query, [], False)
#  the number of clients who are active and have a specific role def get_login_clients_by_role(role):
query = "SELECT COUNT(*) FROM users WHERE role=%s AND is_active = 1" return run_query(query, (role,), False)
#  the number of clients who are active and located in a specific region def get_login_clients_by_location(location):
query = "SELECT COUNT(*) FROM users WHERE location=%s AND is_active = 1" return run._query(query, (location,),
False)
#  Define a class for Zabbix monitoring class ZabbixMonitoring:
#      Constructor function to initialize the class with the necessary hostname values
def  ________init ___(self):
self.zabbix_hostname = 'Zabbix server' self.agent_hostname = 'localhost'
#      Function to send a metric to Zabbix server
def send_metric(self, metric_name: Metrics, metric_value): zbx = ZabbixSender(self.agent_hostname)
print(f"{metric_name.value} {zbx.send(metric)}")
    #  Function to collect the total number of active clients and send it as a metric to Zabbix server
    def collect_total_clients(self):
#           Set the start and end time for monitoring start_time = datetime.datetime.now()
end_time = start_time + datetime.timedelta(minutes=6) while datetime.datetime.now() < end_time:
print("Monitoring clients...") num_list = get_login_clients() num_active_clients = num_list[0][0]
self.send_metric(Metrics.TOTAL_NUMBER_OF_ACTIVE_CLIENTS, num_active_clients)
#                 Wait for 20 seconds before monitoring again time.sleep(20)
    #  Function to collect the total number of active users and send it as a metric to Zabbix server
    def collect_total_users(self):
start_time = datetime.datetime.now()
end_time = start_time + datetime.timedelta(minutes=6) while datetime.datetime.now() < end_time:
print("Monitoring users...")
num_list = get_login_clients_by_role(Role.USER.value) num_active_users = num_list[0][0]
self.send_metric(Metrics.TOTAL_NUMBER_OF_ACTIVE_USERS, num_active_users) time.sleep(15)
    #  Function to collect the total number of active admins and send it as a metric to Zabbix server
    def collect_total_admins(self):
start_time = datetime.datetime.now()
end_time = start_time + datetime.timedelta(minutes=6) while datetime.datetime.now() < end_time:
print("Monitoring admins...")
num_list = get_login_clients_by_role(Role.ADMIN.value) num_active_admins = num_list[0][0]
self.send_metric(Metrics.TOTAL_NUMBER_OF_ACTIVE_ADMINS, num_active_admins) time.sleep(15)
    #  Function to collect the total number of active clients from Europe and send it as a metric to Zabbix server
                def collect_total_clients_from_europe(self): start_time = datetime.datetime.now()
end_time = start_time + datetime.timedelta(minutes=6) while datetime.datetime.now() < end_time:
print("Monitoring europe...")
num_list = get_login_clients_by_location(Location.EUROPE.value) num_clients_from_europe =
num_list[0][0]
self.send_metric(Metrics.NUMBER_OF_CLIENTS_FROM_EUROPE, num_clients_from_europe)
time.sleep(15)
    #  Function to collect the total number of active clients from Asia and send it as a metric to Zabbix server
                def collect_total_clients_from_asia(self): start_time = datetime.datetime.now()
end_time = start_time + datetime.timedelta(minutes=6) while datetime.datetime.now() < end_time:
print("Monitoring asia...")
```

Додаток Г – Ілюстративний матеріал

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВИЗНАЧЕННЯ ТА РЕАЛІЗАЦІЇ ПРІОРИТЕТНИХ ЗАВДАНЬ

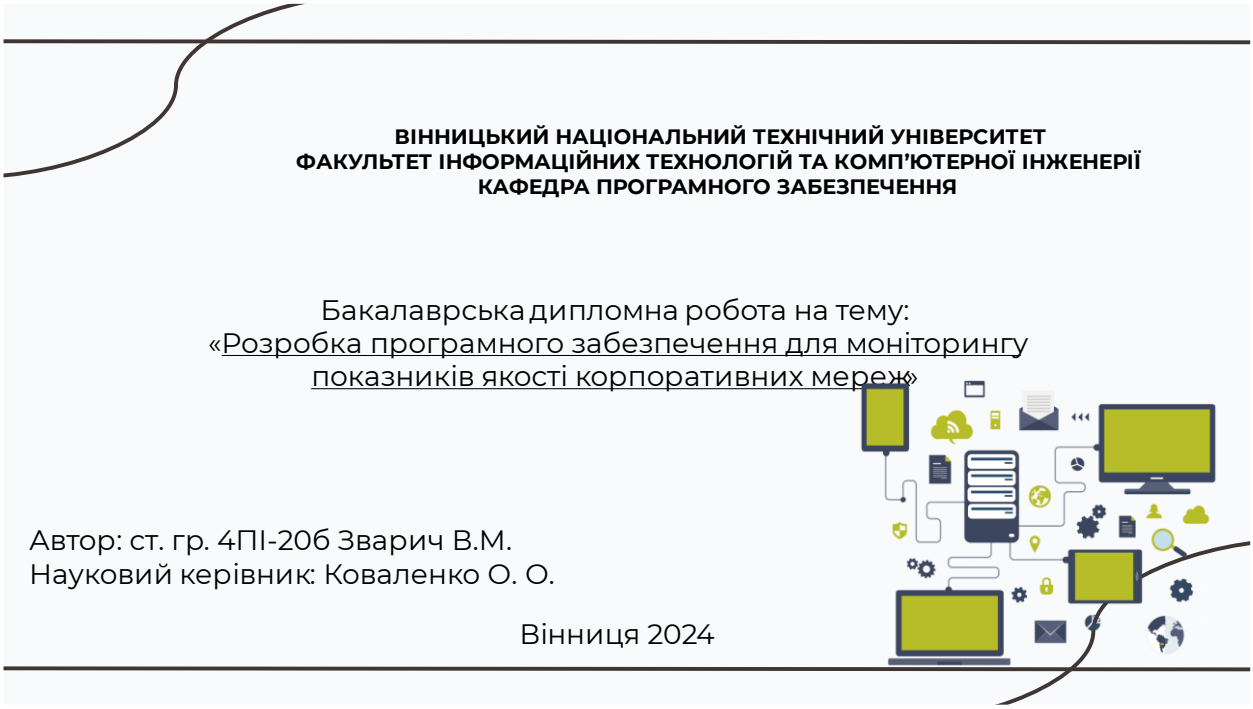


Рисунок Г.1 – Титульний слайд

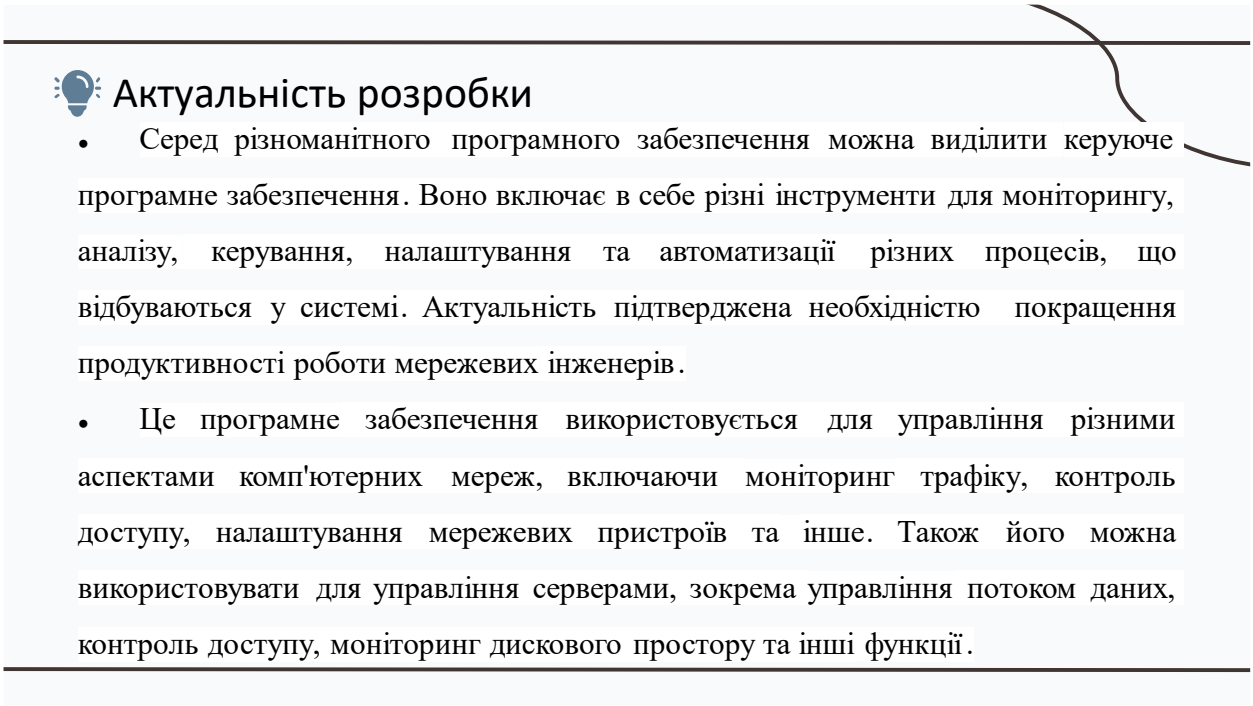


Рисунок Г.2 – Слайд «Актуальність розробки»

Мета, предметта об’єкт дослідження

Метою роботи є розробка програмного забезпечення для моніторингу показників якості функціонування комп’ютерних мереж.

Об'єктом дослідження виступає процес розробки програмного забезпечення для моніторингу та контролю комп’ютерної мережі.

Предметом дослідження є методи моніторингу та контролю комп’ютерних мереж та їх автоматизація для створення програмного забезпечення .

Рисунок Г.3 – Слайд «Мета, предмет та об’єкт дослідження»

Постановка задачі

- аналіз стану програмного забезпечення для моніторингу комп’ютерних мереж;
- розробка методів, моделей моніторингу та контролю комп’ютерних мереж;
- розробка алгоритмів моніторингу та контролю комп’ютерних мереж ;
- створення програмного забезпечення для моніторингу функціонування комп’ютерної мережі;
- тестування модулів програмного забезпечення моніторингу та контролю мережі.

Рисунок Г.4 – Слайд «Постановка задачі»

Рисунок Г.5 – Слайд «Новизна»

Рисунок Г.6 – Слайд «Практична цінність отриманих результатів»

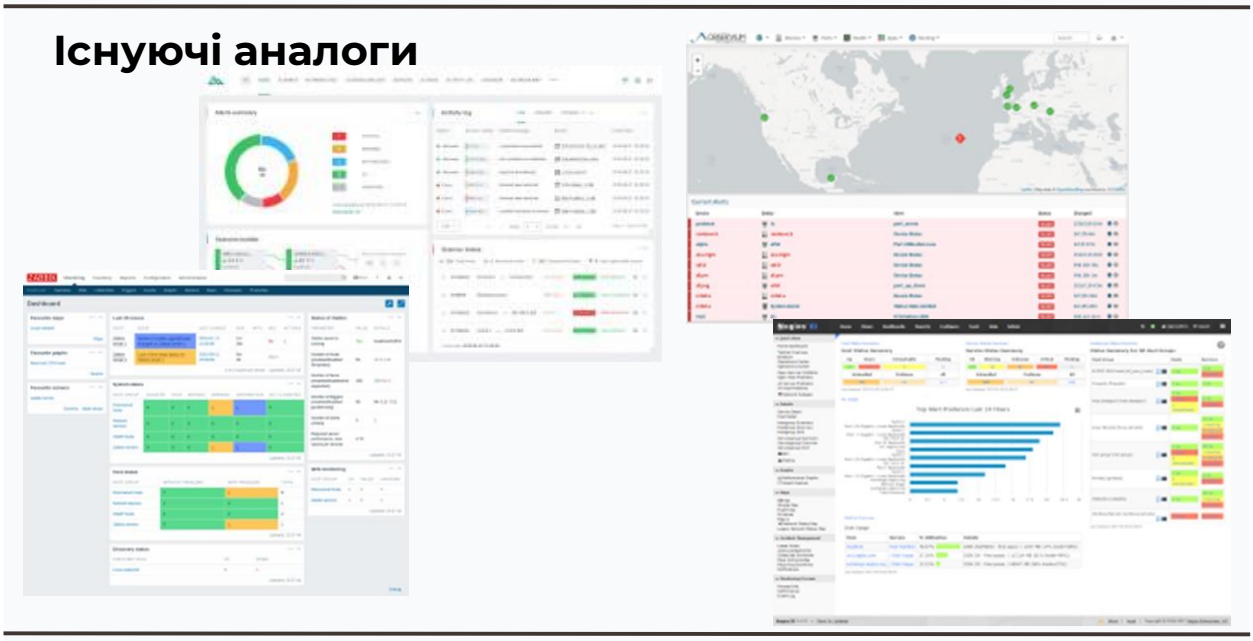
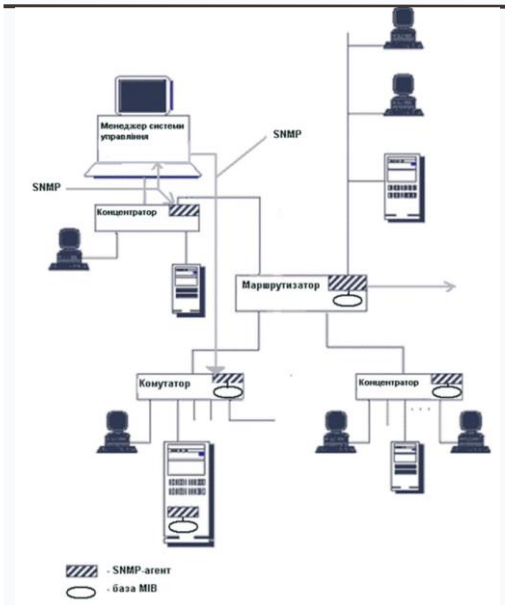


Рисунок Г.7 – Слайд «Існуючі аналоги» 1

Порівняльні характеристика систем моніторингу та контролю комп ’ютерної мережі

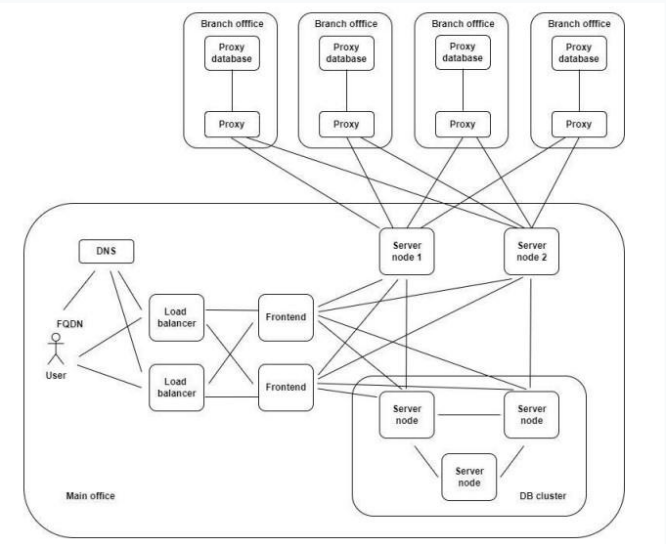
Показники	Olympus Networks	Observium	Nagios	Zabbix	Власна розробка Zabbix +
Підтримка багатьох операційних систем	0	1	1	1	1
Функціонал для моніторингу та контролю	1	1	1	1	1
Функціонал для оповіщення	0	1	1	0	1
Багатокористувацький інтерфейс	0	0	1	1	1
Можливість адаптації за допомогою мов програмування	0	0	0	1	1
Сумарний коефіцієнт	1	3	4	4	5

Рисунок Г.8 – Слайд Порівняння аналогів



Загальна модель моніторингу

Рисунок Г.9 – Загальна модель моніторингу



Алгоритм моніторингу протоколів та загальна модель системи моніторингу мережі в різних корпусах

Рисунок Г.10 – Слайд «Алгоритм моніторингу протоколів та модель моніторингу розподіленої мережі»

Рисунок Г.11 – Слайд «Модель моніторінгу ZABBIX+»

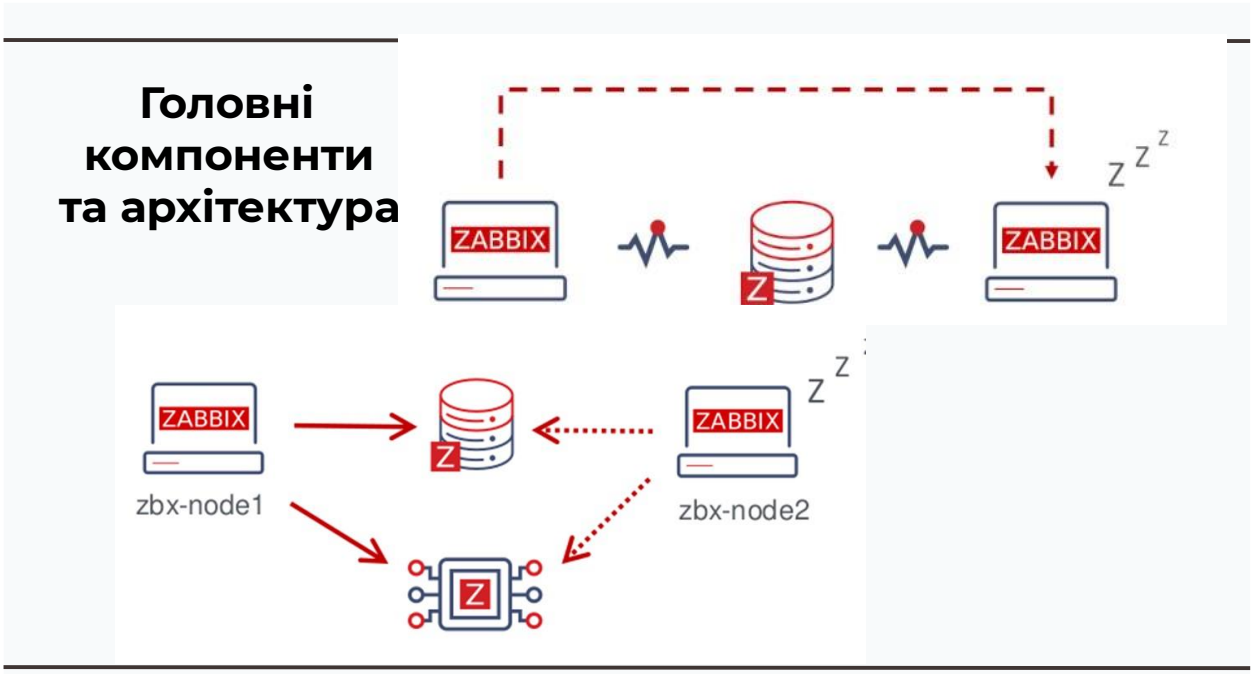


Рисунок Г.12 – Слайд «Моделі архітектури»

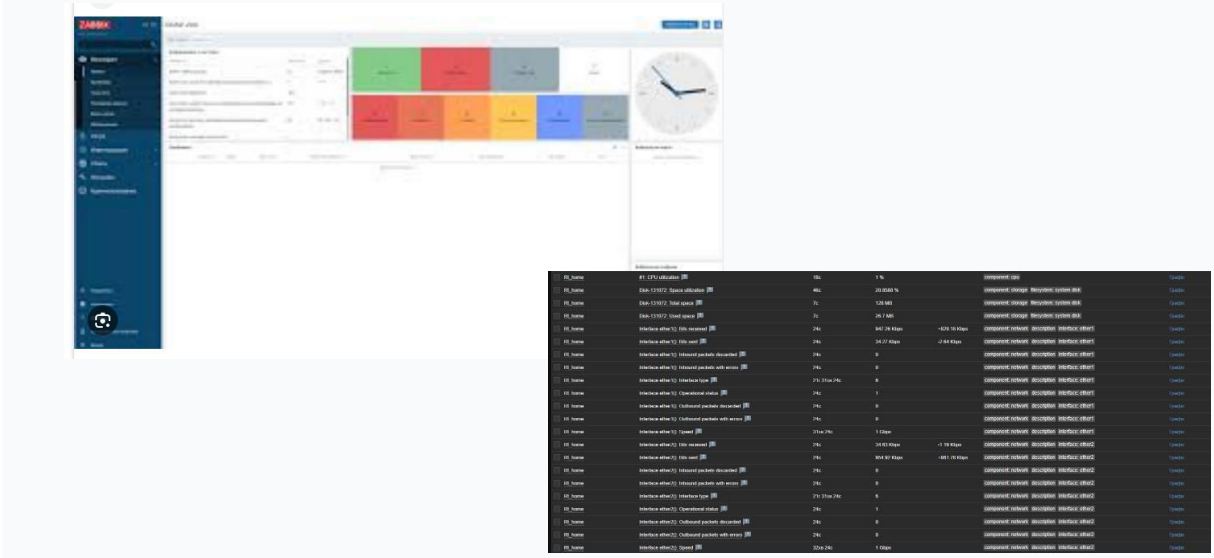


Рисунок Г.13 – Слайд « Тестування програмного забезпечення моніторингу»1

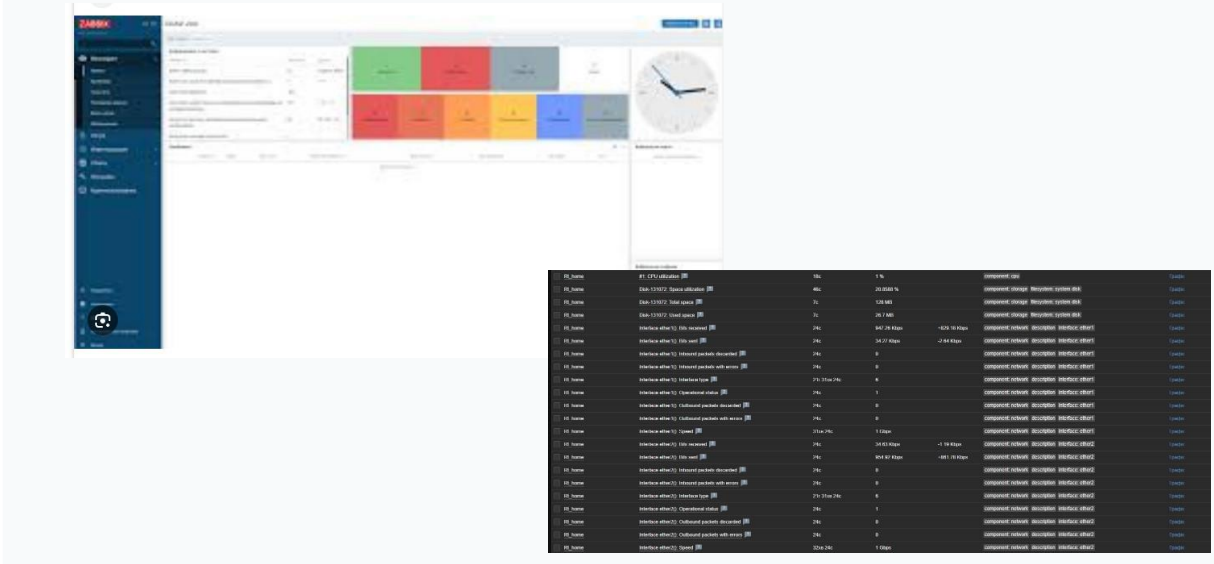


Рисунок Г.14 – Слайд « Тестування програмного забезпечення моніторингу»2

Апробація та публікація результатів роботи

Матеріали бакалаврської дипломної роботи доповідалися на:

Інтернетконференції Інформаційне супільство, Тернопіль, 2024.

За тематикою дослідження опубліковано 1 наукову працю у збірниках матеріалів конференції.

Рисунок Г.15 – Слайд «Апробація»

Висновки

розробку методів, моделей моніторингу та контролю комп’ютерних мереж;

розробку алгоритмів моніторингу та контролю комп’ютерних мереж ;

створення програмного забезпечення для моніторингу функціонування комп’ютерної мережі;

тестування модулів програмного забезпечення моніторингу та контролю мережі.

Теоретичні знання концепції моніторингу, роботи аналогів стали основою для моделей та алгоритмів створення адаптованого програмного застосунку моніторингу та контролю показників роботи комп’ютерної мережі.

Рисунок Г.16 – Слайд «Висновки»