

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки
Факультет комп'ютерних наук та технологій
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему **РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ**
ПП «СЕВАН УКРАЇНА»

Виконав: студент 4 курсу, групи **КНТ-527**
спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Комп'ютерна інженерія»

Волкова Катерина Сергіївна

(прізвище та ініціали)

Керівник **Тягунова М.Ю.**

(прізвище та ініціали)

Рецензент **Куляба-Харитоновна Т.І.**

(прізвище та ініціали)

м. Запоріжжя
2021 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти (освітній ступінь) бакалаврський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерна інженерія
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮЗавідувач кафедри Кудерметов Р.К.

“ ” 2021 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Волкова Катерина Сергіївна

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка системи моніторингу ПП «СЕВАН Україна»керівник проекту (роботи) Тягунова Марія Юріївна, к. т. н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “17” березня 2021 року № 81

2. Строк подання студентом проекту (роботи) 06 травня 2021 року3. Вихідні дані до проекту (роботи) локальна мережа підприємства, мережеві пристрої, мережеві протоколи (SMTP, POP3, HTTP і інші поширені мережеві протоколи) існуюча система моніторингу мережі

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

1. Аналіз існуючих систем моніторингу локальних мереж;

2. Проектування модернізованої системи моніторингу;

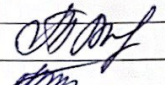
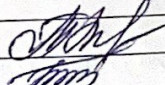
3. Реалізація системи;

4. Результат роботи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Креслення 1 – Блок-схема алгоритма роботи системи;Плакат 1 – Діаграма взаємодії компонентів системи;Плакат 2 – Процеси в ядрі системи;Плакат 3 – Статистика інцидентів.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	Тягунова М. Ю., к. т. н., доцент		
нормконтроль	Щербак Н.В., ст. викл.		

7. Дата видачі завдання 01.03.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз підходів до моніторингу локальних мереж	01.03.2021 р.	
2	Аналіз систем моніторингу локальних мереж		
3	Розробка вимог до модернізованої системи моніторингу локальної мережі приватного підприємства "СЕВАН УКРАЇНА"		
4	Розробка структури системи	25.03.2021 р.	
5	Розробка алгоритму роботи системи	30.03.2021 р.	
6	Реалізація системи уточненого моніторингу	01.04.2021 р.	
7	Аналіз отриманих результатів роботи системи	10.04.2021 р.	
8	Оформлення отриманих результатів у ПЗ	01.05.2021 р.	
	Оформлення графічного матеріалу	13.05.2021 р.	
	Оформлення допоміжного матеріалу	20.05.2021 р.	
9	Захист роботи	24.05.2021 р.	

Студент

(підпис)

К.С. Волкова

(ініціали та прізвище)

Керівник проекту (роботи)

(підпис)

М.Ю. Тягунова

(ініціали та прізвище)

РЕФЕРАТ

ПЗ: 58 с., 23 рис., 25 джерел.

NAGIOS, PHP, ВЕБ-ІНТЕРФЕЙС, ЛОКАЛЬНІ КОМП'ЮТЕРНІ МЕРЕЖІ, ОБ'ЄКТИ МОНІТОРИНГУ, ПЛАГІНИ СНЕСКМК, СЕРВІСИ, СИСТЕМА МОНІТОРИНГУ, ХОСТИ

Об'єкт дослідження – локальні комп'ютерні мережі.

Предмет дослідження – розробка і застосування системи моніторингу локальної мережі.

Мета роботи – підвищення надійності комп'ютерної мережі ПП "СЕВАН УКРАЇНА" за рахунок впровадження системи моніторингу мережі на підприємстві.

Проект складається з чотирьох розділів.

У першому розділі розглянуто предметну область, визначено поняття моніторингу мережі, його важливість в роботі підприємства, типова структура системи моніторингу. Визначено вимоги до системи та проведено аналіз існуючих систем моніторингу мережі.

У другому розділі розглянуто інструменти для проектування власної системи моніторингу. Розглянуті основні компоненти ядра Nagios, його доповнення, можливості для реалізації нової системи, що відповідає вимогам підприємства.

У третьому розділі описано основні компоненти нової системи.

У четвертому розділі проаналізовано результати впровадження нової системи моніторингу.

ЗМІСТ

Вступ	6
1 Аналіз систем моніторингу мереж	9
1.1 Особливості систем моніторингу	9
1.1.1 Вимоги до системи моніторингу	10
1.1.2 Типова структура системи моніторингу	12
1.2 Порівняльна характеристика систем	13
1.2.1 Система моніторингу Cacti	13
1.2.2 Система моніторингу Zabbix	16
1.2.3 Система моніторингу Nagios	17
1.3 Вибір системи	20
2 Проектування системи моніторингу	23
2.1 Структура системи моніторингу	24
2.2 Принципи роботи системи	25
2.3 Функціонування системи	28
2.3.1 Перевірка хостів і сервісів	28
2.3.2 Відстеження поточного стану хостів та сервісів	30
3 Реалізація системи моніторингу	36
3.1 Розробка веб-інтерфейсу системи	36
3.2 Налаштування компонентів системи	40
4 Результати роботи системи	45
Висновки	53
Перелік джерел посилання	55

ВСТУП

Моніторинг мережі — це процес моніторингу доступності мережі, часу безвідмовної роботи, роботи і продуктивності складних мереж. Моніторинг включає відстежування і аналіз мережевих компонентів, таких як маршрутизатори, комутатори і брандмауери, а також з'єднань між ними. На додаток в моніторинг мережі входить спостереження за різними рівнями даних, кінцевими точками мережі і мережевими інтерфейсами.

Маршрутизатори, комутатори і вузли сполучають безліч робочих станцій з найважливішими прикладними програмами на безлічі серверів і в Інтернеті. Крім того, встановлені численні утиліти і додатки безпеки і зв'язку, включаючи брандмауери, віртуальні приватні мережі (VPN) і антивіруси.

Перевірка працездатності і продуктивності інтерфейсів на предмет їх збоїв допомагає діагностувати, оптимізувати і управляти різними мережевими ресурсами як на місці, так і віддалено. Завдяки даним у формі таблиць, діаграм, графіків, інформаційних панелей і звітів, мережевий моніторинг допомагає системним адміністраторам скоротити середній час на ремонт (MTTR) і вирішити проблеми з продуктивністю мережі в реальному часі. Коли такі проблеми виявляються, система повідомляє системних адміністраторів безпосередньо або через систему підтримки, щоб вони могли знайти рішення проблеми якнайшвидше.

Розуміння структури і складності мережі, обізнаність про роботу усіх окремих її елементів у будь-який час, є ключовим чинником успіху в підтримці працездатності і цілісності мережі компанії і її клієнтів. Для моніторингу в мережі можуть бути тисячі точок даних, і у край важливо мати доступ до значимої, точної і оновленої інформації у будь-який час. Системні адміністратори мають бути проінформовані про те, що відбувається в кожному сегменті мережі у будь-який момент часу.

Мережа, як правило, має внутрішніх і зовнішніх користувачів, включаючи співробітників, клієнтів, партнерів і інших зацікавлених суб'єктів. Некоректна робота мережі по-різному впливає на підприємство залежно від типу користувача. Наприклад, якщо співробітники не можуть отримати доступ до інформації, яка їм потрібна для виконання своєї роботи, це означає зниження продуктивності праці, фінансові втрати, і надалі може завдати збитку репутації.

Кожен компонент мережі є потенційною точкою збою. Саме тому украй важливо впровадити стратегію зведення до мінімуму вірогідності простою. Таким чином, якщо один сервер або маршрутизатор не працює, інший, чекаючи необхідності, може автоматично підключитися до мережі, щоб пом'якшити наслідки настання непрацездатності основного устаткування. Не усі проблеми можуть бути вирішені прогнозуючим чином до того, як реальні загрози стануть очевидними.

Проте, якщо є можливість активно здійснювати контроль мережі в режимі реального часу, можна виявити проблеми до того, як вони стануть глобальними. Наприклад, перегружений сервер можна замінити до того, як він втратить працездатність, але тільки якщо відомо, що це неминуче.

Більшість мережевих проблем трапляються через помилки в конфігурації. Проблеми конфігурації, що здаються незначними, часто призводять до значного простою мережі. Щоб запобігти цьому, моніторинг мережі здатний виявити дублікати IP і проблеми з маршрутизацією пристрою.

Моніторинг мереж може також відігравати важливу роль в перспективному плануванні мереж. Шляхом інформування адміністраторів про використовувані ресурси і інші проблеми, які можуть привести до перевантаження мережі система моніторингу може допомогти адміністраторам ІТ-інфраструктури швидко адаптуватися до значного зростання бізнесу або збільшення числа співробітників.

За допомогою мережевого моніторингу системний адміністратор повинен знати про стан мережі, і уміти своєчасно робити відповідні заходи, необхідні для усунення проблем.

Первинне значення має отримання інформації про стан поточних мережевих пристроїв (наприклад, маршрутизаторів і комутаторів) і ключових мережевих серверів. Мережевий адміністратор повинен також знати, що основні послуги (наприклад, послуги електронної пошти, веб-сайту і передачі файлів) доступні на постійній основі. Таким чином, вибрана тема дипломної роботи бакалавра є актуальною.

Мета роботи — підвищення надійності комп'ютерної мережі ПП "СЕВАН УКРАЇНА" за рахунок впровадження системи моніторингу мережі на підприємстві.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз існуючих систем моніторингу;
- визначити вимоги до системи моніторингу комп'ютерної мережі ПП "СЕВАН УКРАЇНА";
- спроектувати поліпшену систему моніторингу, що максимально підходить до вимог підприємства;
- реалізувати спроектовану систему;
- проаналізувати отримані результати роботи.

1 АНАЛІЗ СИСТЕМ МОНІТОРИНГУ МЕРЕЖ

1.1 Особливості систем моніторингу

Нижче приведені ключові типи інформації про статус мережі, які необхідно знати щомиті :

- доступність усіх критично важливих сервісів мережі - втрата доступності сервера електронної пошти, HTTP або FTP навіть на одну годину може привести до закриття компанії;
- доступність усіх мережевих пристроїв - більшість мереж є комбінацією різнорідних пристроїв, тому треба контролювати Windows, Linux, UNIX і інші типи серверів, робочих станцій і принтерів;
- середній показник використання пам'яті, процесора і пам'яті;
- об'єм дискового простору, використовуваного на ключових серверах, оскільки додаткам потрібно дискову місткість, необхідно знати про будь-які зміни місткості диска, які можуть вказувати на проблему з конкретним застосуванням або системою;
- функція брандмауера, антивірусний захист, оновлення сервера і захист від шкідливого програмного забезпечення.

У разі виникнення проблем слід негайно повідомити адміністратора мережі про це за допомогою звукових попереджень, спливаючих вікнах або e-mail листах, автоматично генерованих системою моніторингу мережі.

Виділимо зовнішні по відношенню до системи моніторингу об'єкти і суб'єкти, з якими вона повинна взаємодіяти :

- об'єкти моніторингу;
- одна або більше за системи зберігання даних, наприклад, система управління базами даних;

– користувачі — це особи, відповідальні за справну роботу об'єктів моніторингу, обслуговуючий персонал самої системи моніторингу і користувачі, яким надається інформація про поточний стан об'єктів моніторингу.

Функції (окремі підпрограми) і програмні модулі (фрагмент опису процесу, що оформляється як самостійний програмний продукт) підсистем можна розділити на дві категорії:

– ті, що реалізують внутрішні процеси самої системи моніторингу - інформаційні потоки між підсистемами, механізми передачі даних, запуску процедур збору даних, збереження і архівація даних, функції обробки даних, повідомлення про збої);

– ті, що здійснюють взаємодію із зовнішніми об'єктами - їх реалізація залежить від виду цих об'єктів і протоколів взаємодії або інтерфейсів, що надаються ними [1, с. 33].

Функції і модулі підсистем першої категорії мають бути реалізовані максимально просто і універсально. Їх сукупність можна назвати ядром системи моніторингу.

1.1.1 Вимоги до системи моніторингу

Ядро повинне надавати системному адміністраторові гнучкий інструмент для оптимального вирішення завдань, що стоять перед ним, пов'язаних з конкретним набором зовнішніх об'єктів.

Система повинна дозволяти легко змінювати конкретну реалізацію функцій і модулів другої категорії (і додавати нові) відповідно до завдання, що стоїть.

Адміністратор повинен мати можливість реалізувати і використати в системі власні модулі повідомлення, виведення, збору і аналізу даних, працюючи у рамках підтримуваної схеми, що регламентується ядром.

Ядро системи складають компоненти, які виконують її основні завдання, тобто збір і накопичення даних про об'єкти моніторингу, діагностика позаштатних ситуацій і сповіщення про них, ухвалення рішення про правильне рішення цих проблем.

Компоненти ядра здійснюють виклик підключених необхідних модулів, що реалізують рішення завдань та відповідають сфері застосування системи (моніторинг конкретних типів об'єктів і служб, взаємодія з системами зберігання даних, відображення стану системи в різних форматах і тому подібне). Варто відмітити, що функції підсистеми виведення цілком винесені з ядра, оскільки вони не вимагають взаємодії з самими процесами моніторингу [1].

Веб-інтерфейс грає важливу роль у виборі системи моніторингу, оскільки саме від нього залежить взаємодія системного адміністратора з нею, налаштування моніторингу, управління мережею і рішення проблем. Тому при виборі системи необхідно враховувати простоту інтерфейсу у використанні, згідно мети дипломної роботи.

Варто згадати, що ця розробка призначена тільки для відображення існуючих інцидентів на великому моніторі і мінімальної взаємодії з ними командою моніторингу ПП "СЕВАН УКРАЇНА". Для більшого функціонала системи Nagios (налаштування безпосередньо самого моніторингу, відображення графіків, налаштування повідомлень тощо) використовується графічний інтерфейс Centreon (рис. 1.1).

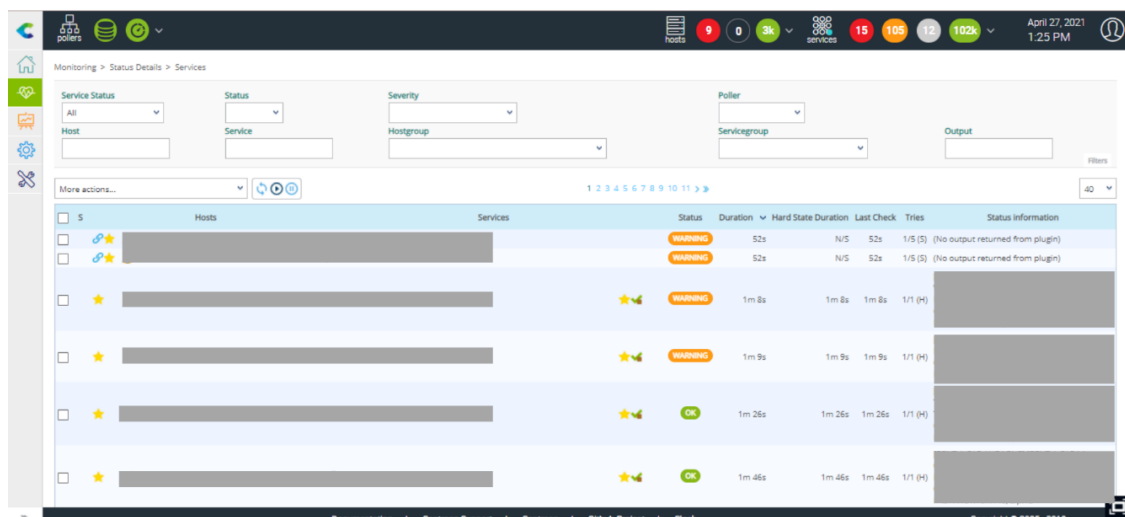


Рисунок 1.1 — Дешборд Centreon

1.1.2 Типова структура системи моніторингу

Вчасно виявлений і неусунений збій або відмова ІТ- інфраструктури в найкоротші терміни може привести до тимчасової зупинки діяльності усієї її компанії [1]. Система моніторингу мережі — це система, яка виконує постійний контроль над вузлами і каналами мережевої інфраструктури підприємства. Система устаткування моніторингу мережі займається пошуком повільних і несправних систем [2].

На рис. 1.2 приведена архітектурна схема типової системи моніторингу.

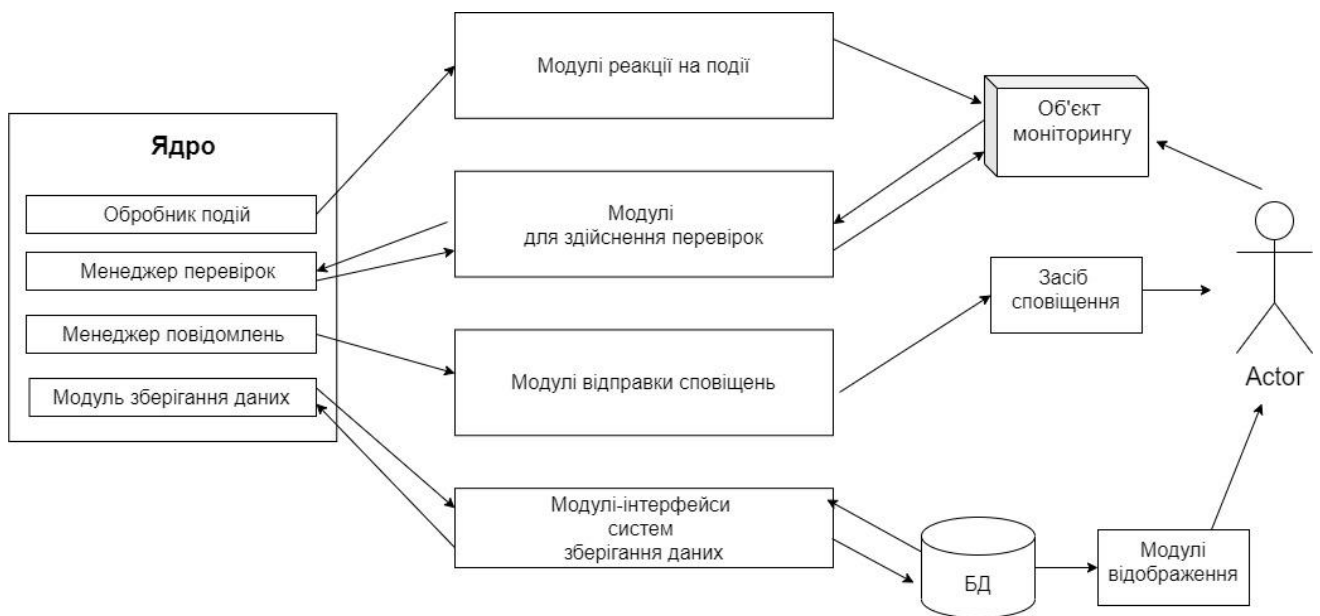


Рисунок 1.2 — Архітектура системи моніторингу

Функціонал сучасних систем моніторингу дозволяє відстежувати стан таких сервісів, як:

- доступність сервера за допомогою відправки запитів протоколу ICMP і фіксації відповідей, що поступають;
- доступність web-сервера за допомогою відправки запитів HTTP;
- доступність поштових сервісів за допомогою відправки запитів протоколу SMTP [3].

Найпоширеніші системи моніторингу локальних вычислительных мереж:

- Nagios;
- Zabbix;
- Cacti.

1.2 Порівняльна характеристика систем

1.2.1 Система моніторингу Cacti

Cacti — інструмент моніторингу мережі з відкритим початковим кодом, який можна встановити в ОС Linux або Windows. Ця система найчастіше використовується для моніторингу мережевого трафіку, опитуючи мережеві інтерфейси маршрутизатора або комутатора, використовую протокол SNMP.

Cacti був створений з метою графічного відображення життєво важливих параметрів пристроїв, підключених до мережі, за допомогою надзвичайно простого зрозумілого інтерфейсу. Система моніторингу на основі Cacti складається з сервера, на якому встановлено програмне забезпечення Cacti, який управляється (як для перегляду графіків, так і для налаштування) через веб-інтерфейс. Сервер Cacti збирає дані з хостів через фіксовані інтервали (за умовчанням кожні 5 хвилин) через опитувач і зберігає їх у базі даних RRD [4].

База даних RRDtool, яку використовує Cacti, є програмою з відкритим початковим кодом і дозволяє зберігати дані, отримані за певний проміжок часу і створювати діаграми, що відображають мережевий трафік, використання ресурсів тощо [5].

RRDtool працюють за принципом циклічного перебору (Round Robin Archive) (рис. 1.3) [6], тобто коли нові елементи додаються поверх старих даних. На практиці

база даних кругова, тому, як тільки вона досягає кінця, показчик повертається до першого елементу і починає перезапис даних.

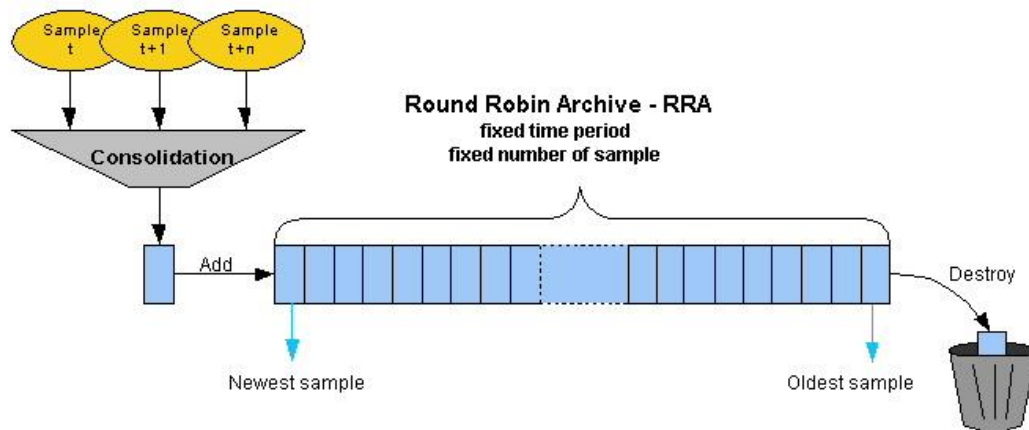


Рисунок 1.3 — Зберігання даних в архівах Round Robin

Система опитує пристрої із заданими інтервалами часу і на основі отриманих даних від RRDTool малює діаграми результуючих даних, таких як навантаження CPU, завантаженість мережі, дані про температуру (за останні 24 години, або за попередній тиждень, або за інші певні інтервали) і інші, зазвичай використовуючи дані з SNMP

Це дуже ефективний спосіб зберігання даних впродовж певного часу при використанні відомого об'єму дискового простору.

Додаток має зовнішній веб-інтерфейс, написаний на PHP. Веб-інтерфейс для додавання нових хостів або параметрів для моніторингу не дуже зрозумілий, існує ризик плутанини між шаблонами відображення, сценаріями для отримання даних і типом графіку.

Sacti має широкий спектр налаштувань, починаючи від додавання комп'ютерів, які треба відстежувати, і закінчуючи визначенням нових джерел даних, доступних через SNMP або спеціальні сценарії [5]. У разі конфігурації складних систем відсутність інвентаризації може стати істотним недоліком, оскільки додавання нових хостів вимагає безліч операцій, що повторюються і великий об'єм роботи [4].

Однаке конфігурація цієї системи є складним і стомливим процесом. Якщо допустити помилку в процесі налаштування, її потім складно виправити. До того ж конфігурація Састі завжди має бути актуальною. Налаштування інтерфейсів через веб-інтерфейс це стомливий процес, тому доведеться скористатися передбаченими скриптами командного рядка.

Система сповіщень відсутня зовсім в цій системі моніторингу, що є істотним недоліком. Але за допомогою плагіну можна її усунути. Наприклад, thold може відправляти сповіщення по електронній пошті або SMS. Це програмне забезпечення доцільно використати у поєднанні з повнішим програмним забезпеченням, яка не здатна побудувати діаграми. У більшості випадків він використовується для моніторингу мереж разом з Nagios.

Приклад діаграми Састі наведений на рис. 1.4 [7].

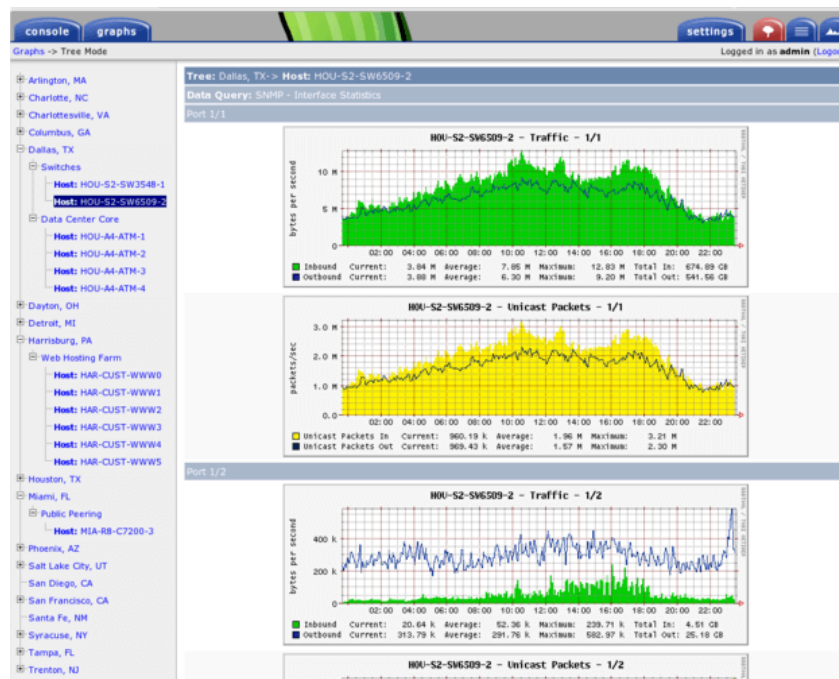


Рисунок 1.4 — Діаграми Састі

1.2.2 Система моніторингу Zabbix

Zabbix — безкоштовне програмне забезпечення (корпоративного рівня, призначене для здійснення моніторингу мережі, сервера, хмарної інфраструктури, додатків і сервісів. Таким чином, це дуже повна і функціональна програма моніторингу.

Zabbix підтримує здійснення моніторингу через SNMP і надає найкращу звітність [8].

Можна відстежувати різну інформацію, таку як використання смуги пропускання, стан мережі, а також зміни, внесені в конфігурацію мережі. Дані про продуктивність в Zabbix пов'язані і взаємозв'язані через SNMP і інтерфейс інтелектуального управління платформою(IPMI). Програмне забезпечення також включає аналіз тенденцій для виявлення мережевих проблем ще до того, як вони виникнуть, і автоматично виправляє їх або повідомляє системного адміністратора за допомогою SMS.

Веб-інтерфейс Zabbix є відмінною рисою цієї системи. Багатофункціональний, але одночасно і простий у використанні веб-інтерфейс поставляється в пакеті за умовчанням. Веб-інтерфейс надає доступ, що не лякає, але в той же час багатофункціональний і можливість налаштувань інсталяцій великих розмірів. Веб-інтерфейс Zabbix передбачає налаштування моніторингу і перегляд зібраних даних. Мережевий адміністратор може переглядати і управляти усім в одному інтерфейсі [9].

Встановити систему моніторингу Zabbix не складе труднощів навіть для фахівця, який знає Unix-подібні операційні системи. Для більшості операційних систем установка Zabbix доступна з офіційного репозиторія. Необхідні компоненти LAMP автоматично встановлюються залежностями. Основна частина установки виконується в консолі сервера. Загальні параметри певного екземпляра системи моніторингу Zabbix налаштовуються у веб-інтерфейсі [10].

Глобальна панель Zabbix проілюстрована на рис. 1.5.

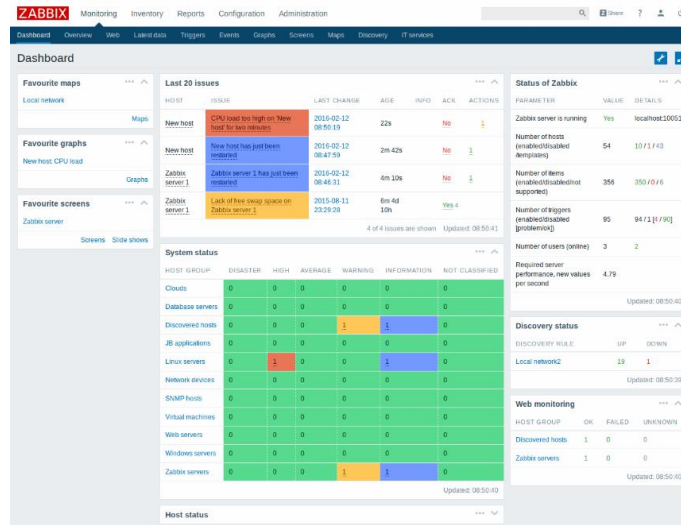


Рисунок 1.5 — Дешборд Zabbix

1.2.3 Система моніторингу Nagios

Компанія Nagios, заснована в 1999 році, є одним з лідерів галузі в наданні рішень для моніторингу систем і мереж від малих до корпоративних інфраструктур. Nagios Core — інструмент з відкритим кодом для моніторингу систем і мереж, у тому числі робочих станцій працівників підприємства, IT- сервісів і активних мережевих компонентів [11].

Спочатку Nagios була розроблена для роботи під GNU Linux, але вона також добре працює і під іншими Unix- подібними до ОС, поширюється за ліцензією GNU General Public License Version 2.

Nagios здатний контролювати практично усі типи компонентів, такі як мережеві протоколи, операційні системи, системні показники, додатки, служби, веб-сервери, веб-сайти, проміжне ПЗ тощо. Програма стежить за вказаними вузлами і службами, і оповіщає адміністратора у тому випадку, якщо служба припиняє (чи поновлює) свою роботу.

Nagios Core має додатковий веб-інтерфейс, який відображає стан мережі, повідомлення, файли журналів і багато що інше. Core може повідомити системного

користувача про проблеми з сервером або хостом. Крім того, Core може відстежувати мережеві служби, такі як SMTP, HTTP і Ping.

Так само існує комерційна Nagios XI — це розширений інтерфейс Nagios Core, задуманий як версія засобу моніторингу корпоративного рівня. Nagios XI має додаткові можливості, сучасніший і простіший в навігації веб-інтерфейс, що пропонує інтерактивну інформаційну панель з оглядом хостів, сервісів і мережевих пристроїв.

Можливість побудови графіків тенденцій і наявність наочних інструментів планування потужності допомагають в підготовці модернізацій інфраструктури.

Можливості системи моніторингу Nagios:

- моніторинг мережевих служб (SMTP, POP3, HTTP, NNTP, ICMP, SNMP);
- використовує топологію для визначення залежностей;
- у разі виникнення проблем, вони може бути усунені автоматично у міру їх виявлення в процесі моніторингу;
- якщо служби працюють нормально, то немає необхідності перевіряти, що хост активний;
- відносно масштабований, керований і безпечний моніторинг стану серверів (завантаження процесора, використання диска, системні балки);
- підтримка видаленого моніторингу через шифровані тунелі SSH або SSL;
- організація спільної роботи декількох систем моніторингу з метою підвищення надійності і створення розподіленої системи моніторингу;
- виведення загальної статистики по усіх хостах, над якими проводиться моніторинг, за допомогою утиліти nagiosstats;
- відправка повідомлень у разі виникнення проблем із службою або хостом (будь-яким певним користувачем через модуль системи);
- відстежує доступність вузлів і час їх безвідмовної роботи;
- стежить за часом відгуку кожного вузла;

- архітектура продукту дозволяє легко писати нові плагіни будь-якою мовою по вашому вибору;
- підтримується Microsoft і Linux;
- розрахований на багато користувачів доступ [12].

Ця система працює на механізмі моніторингу Core 4, який забезпечує високий рівень продуктивності за рахунок використання меншої кількості ресурсів сервера [8].

Nagios Core є безкоштовним, Nagios XI необхідно придбавати у Nagios Enterprises. Окрім тих же функцій, що і Core, XI додає заздалегідь налагоджені віртуальні машини, призначений інтерфейс веб-конфігурації (UI), графік продуктивності, мобільне застосування, інформаційні панелі, звіти за розкладом, технічну підтримку по електронній пошті і багато що інше.

Архітектура Nagios досить проста, тому створення нових плагінів на будь-якій мові програмування вибору (Shell, C++, Perl, Python, PHP, C# і інші) не є складним завданням. Існують тисячі плагінів.

Установка системи моніторингу Nagios Core здається складнішою, тому що система вимагає компіляції з початкових кодів, які слід завантажити з офіційного сайту. Після установки Nagios Core необхідно встановити плагіни Nagios (пакет плагінів Nagios, який містить більшість часто використовуваних плагінів для повноцінної роботи системи моніторингу). Процес розгортання системи моніторингу Nagios добре документований тільки англійською мовою.

Проте, перевага Nagios полягає в його розширюваності, тобто не складе особливих труднощів додати окремі компоненти мережі (користувачів, комп'ютерів, додатків, служб), збільшити довжину сегментів кабелів і замінити наявну апаратуру на потужнішу. Проте, як згадувалося вище, конфігурація цієї системи не проста — система моніторингу цілком складається з текстових конфігураційних файлів, зв'язки між якими проходять наскрізь, і не завжди просто знайти помилку.

Дані також зберігаються в RRD, по яких будуються графіки, проте довгий інтервал між перевітками і вимірами параметрів робить значення даних більше

середнім, в цьому випадку страждає точність даних. Існує платна версія Nagios XI, де є графічне середовище для налаштувань, але його вартість складає від 57.835 грн (Standard Edition) до 188.290 грн (Unlimited Node), залежно від кількості вузлів управління [13].

1.3 Вибір системи

Безперечна перевага Zabbix полягає в тому, що система абсолютно безкоштовна і без яких-небудь обмежень функціонала. Розробник заробляє на сертифікації партнерів і технічній підтримці. Конфігурації системи розміщені у базі, управління відбувається через інтерфейс. Мінімальний інтервал між вимірами дорівнює 1 секунді, тим самим точність даних вища, ніж у Nagios.

Обидві системи мають перевагу в можливості інтеграції з облаштуваннями Netring для моніторингу мікроклімату і управління електроживленням, що дозволяє системам моніторингу не лише отримувати відомості від різного устаткування і інформаційних систем, але і зведення про стан мікроклімату серверних кімнат.

Sacti — це простий інструмент для ведення журналів і побудови графіків, який можна ефективно використати в промислових або технічних умовах, де для звітів, графіків і презентацій потрібно елементарні дані. Той факт, що це відкритий вихідний код означає, що систему можна використати без яких-небудь витрат для компанії, що є великим бонусом. Якщо необхідно відстежувати декілька серверів або додатків, вести детальні журнали подій, а також дані `char` для звітів про працездатність і аналізу даних, то Sacti безперечно є хорошим вибором.

Проте, у разі моніторингу глибшого рівня, який може видавати попередження в реальному часі, знадобиться бібліотека спеціалізованих плагінів для унікальних вимог до моніторингу, то у такому разі краще зробити вибір на користь Nagios. В порівнянні

з Cacti, Nagios — сучасніший і з великою кількістю функцій і можливостей. Підприємства масштабного рівня можуть віддати перевагу цій системі, якщо мережа потребує постійного моніторингу.

Наочний порівняльний аналіз популярних систем моніторингу приведений в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика систем

	Cacti	Zabbix	Nagios
Графіки	Так	Так	Так
Метод зберігання даних	RRDtool, MySQL, PostgreSQL	MySQL, PostgreSQL, SQLite або Oracle	MySQL, MariaDB
SNMP	Так	Так	За допомогою плагіну
Плагіни	Так	Так	Так
Доступ через Web-сторінку	Повний доступ	Повний доступ	Перегляд, управління, звіти
Інвентаризація	Ні	Так	За допомогою плагіну
Ліцензія	GNU GPL 2	GNU GPL 2	GNU GPL
Вартість	Безкоштовно	Безкоштовно	Стандартна версія - 2000\$, а корпоративна - 3495\$
Географічна карта вузлів	За допомогою плагіну	Так	Динамічні і такі, що налаштовуються
Платформа	Windows, UNIX	UNIX	UNIX
Остання дата виходу	30.11.2020	25.01.2021	28.04.2020
Остання версія релізу	1.2.16	5.2.4	4.4.6

Порівняльний аналіз існуючого програмного забезпечення, що відповідає переліченим вище вимогам до ядра, дозволив зупинити вибір на системі Nagios, яка здійснює виклик модулів перевірки, сповіщення і реакції відповідно до конфігураційних файлів, які описують об'єкти моніторингу і правила взаємодії з відповідальними за їх функціонування особами.

Застосування конкретно цієї системи моніторингу дозволить забезпечити надійність, безпеку і погоджене функціонування усіх компонентів мережі компанії, а також підвищити якість сервісів, що надаються, усунути збої в роботі устаткування і програмного забезпечення, оптимізувати використання інформаційних ресурсів.

На сьогодні системні адміністратори компанії тримають під контролем близько трьох тисяч пристроїв, включаючи сервера, комутатори, принтери тощо. У разі збою в роботі устаткування або програмного забезпечення адміністратори будуть оповіщені за допомогою електронної пошти або за допомогою повідомлення. Таке своєчасне сповіщення дозволяє максимально скоротити простої мережі та обладнання і час відсутності сервісів, що надаються.

Не менш значимим є веб-інтерфейс користувача, тобто системного адміністратора. Вважається, що за допомогою веб-інтерфейсу адміністратор тільки бачитиме виникаючі інциденти моніторингу та відмічає їх позначками. Тобто потрібен максимально простий інтерфейс для взаємодії адміністратора з системою моніторингу.

Варто помітити, що ця розробка призначена тільки для відображення існуючих інцидентів на великому моніторі і мінімальної взаємодії з ними командою моніторингу ПП "СЕВАН УКРАЇНА". Для більшого функціонала системи Nagios (налаштування безпосередньо самого моніторингу, відображення графіків, налаштування повідомлень тощо) використовується графічний інтерфейс Centreon.

Приклад дашборда Centreon наведений на рис. 1.6.

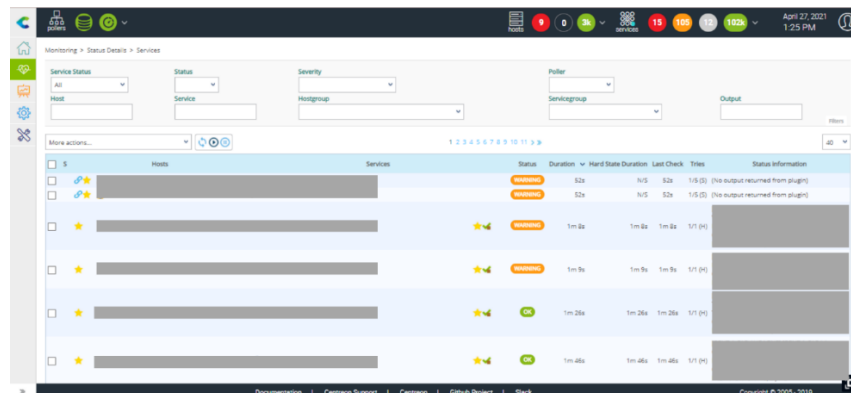


Рисунок 1.6 — Дашборд Centreon

Велика частина зовнішнього і внутрішнього інтерфейсу Nagios XI написана на РНР, включаючи підсистему, обробники подій і повідомлення, і Python, що використовується для створення звітів планування потужності і інших звітів.

РНР розроблений для простої інтеграції у веб-сайт. Одно з найбільш поширених застосувань РНР — отримання вмісту з бази даних і виведення його на HTML-сторінку. РНР може підключатися до багатьох типів серверів баз даних.

2 ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ

Згідно проведеного аналізу було прийнято рішення розробити поліпшену систему моніторингу на базі системи Nagios XI. Для цього на підприємстві буде налаштовано безпосередньо систему Nagios XI та розроблено зручний веб-інтерфейс роботи з нею. Та спочатку розглянемо архітектуру системи моніторингу та принципи її роботи.

2.1 Структура системи моніторинга

Поліпшена система моніторингу складається з таких компонентів, як:

- ядро Nagios;
- плагіни;
- платформа Centreon - контролює усю інфраструктуру підприємства за допомогою панелей моніторингу в реальному часі, аналітики і попереджень;
- новий веб-інтерфейсу системного адміністратора.

Модулі Nagios реалізуються на PHP, bash і Perl. Його модулі є виконуваними файлами і повертають текстові значення. Модулі можуть викликатися незалежно від самої системи, що спрощує їх написання і відладку.

Nagios має архітектуру сервер-агент. Сервер Nagios встановлений на хості, а плагіни на віддалених хостах або серверах.

Планувальник — це компонент серверної частини Nagios.

Процеси, які відбуваються в ядрі системи (рис. 2.1):

- 1) планувальник відправляє сигнал для виконання плагінів на віддаленому хості;
- 2) плагін отримує статус від віддаленого хоста, збирає дані (про використання CPU, використання пам'яті тощо);
- 3) плагін відправляє дані в планувальник процесів;
- 4) планувальник процесів оновлює графічний інтерфейс і повідомлення відправляються адміністраторам [14].



Рисунок 2.1 — Процеси в ядрі системи

2.2 Принципи роботи системи

Nagios Core не включає ніяких внутрішніх механізмів для перевірки стану хостів та служб в мережі. Замість цього він покладається на сторонні програми, що називаються плагінами, для виконання усієї «брудної» роботи.

Плагіни — це скомпільовані виконувані файли або сценарії (сценарії Perl, сценарії командного рядка, Python, PHP, Ruby тощо), які можна запустити з командного рядка для перевірки стану вузла або сервісу.

Щоб виконати доповнення на Perl, Nagios компілює цю програму за допомогою вбудованого інтерпретатора Perl, Embedded Perl Interpreter. Без інтерпретатора Nagios компілює зовнішні програми (плагіни), написані на Perl і інших мовах програмування, виконуючи їх як зовнішню команду [15].

Плагіни діють як віртуальний, абстрактний рівень між логікою моніторингу, присутньою в демоні Nagios Core, і тими налаштуваннями і сервісами, які бачить кінцевий користувач, тобто системний адміністратор (рис. 2.2).

Перевага цієї архітектури плагінів полягає в тому, що є можливість відстежувати практично все, що завгодно. Вже створено безліч плагінів для

моніторингу основних ресурсів, таких як завантаження процесора, використання диска, швидкість роботи команди "ping" і тощо.

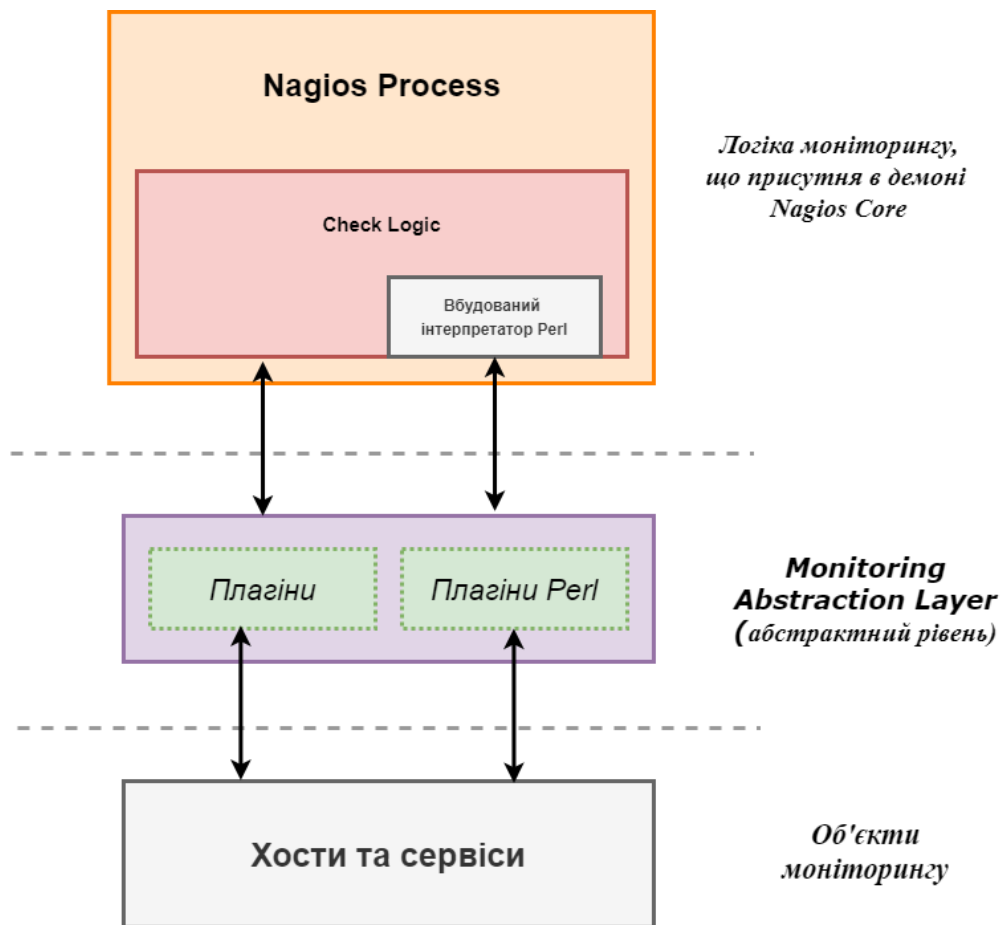


Рисунок 2.2 — Логіка роботи плагінів

Checkmk — один з кращих плагінів системи моніторингу Nagios, який дозволяє створювати конфігурацію з використанням Python і полегшити роботу ядра Nagios, щоб поліпшити його масштабування і дозволити контролювати більше сервісів та хостів з одного сервера Nagios.

Checkmk є дуже ефективним та простим в налаштуванні, особливо у великих середовищах моніторингу.

Великий плюс цього плагіну в тому, що більшість звичайних сервісів перевіряються тільки при одному підключенні до віддаленого хосту, а результати

передаються в Nagios як пасивні. Це покращує продуктивність, зменшує мережевий трафік і, звичайно ж, знижує навантаження на віддалений хост, який якраз таки відстежується [16].

Checkmk також надає наступні можливості:

- значне скорочення використання центрального процесора на хості Nagios;
- автоматична інвентаризація предметів для перевірки на хостах;
- автоматичне створення файлів конфігурації Nagios;
- обробка кластерів без службової IP-адреси;
- зручний моніторинг журналу подій;
- підтримує SNMP і автоматичне виявлення служб для багатьох комутаторів Cisco, маршрутизаторів і файлових серверів NetApp;
- все, що підтримує Nagios.

Для Checkmk треба встановити агент `check_mk`, тобто простий сценарій оболонки `check_mk_agent`, і прив'язати цей сценарій до TCP порту 6556 за допомогою демона `xinetd`.

Архітектуру Checkmk показано на рис. 2.3 [17].

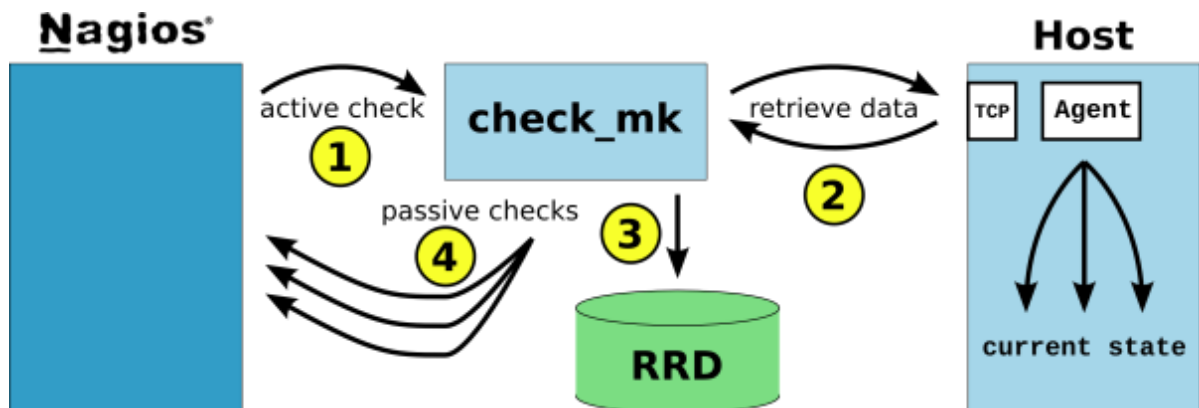


Рисунок 2.3 — Архітектура Checkmk

2.3 Функціонування системи

2.3.1 Перевірка хостів і сервісів

У Nagios використовуються перевірки, щоб побачити, чи працюють хости і сервіси належним чином або ні. Для того, щоб перевірити який-небудь параметр, сервер повинен виконати певну команду. Наприклад, запуск сценарію, який виконає підключення до сервера і проаналізує відповідь.

У Nagios є 2 методи перевірки статусу служб "активні перевірки" (active Checks) і "пасивні перевірки" (passive Checks).

Активні перевірки ініціюються процесом Nagios, а потім виконуються на регулярній основі. Логіка перевірки усередині процесу Nagios запускає активну перевірку [18].

Для моніторингу хостів і сервісів, працюючих на видалених машинах, Nagios запускає плагіни і повідомляє, яку інформацію необхідно збирати. Плагін, зібравши необхідні дані, відправляє їх назад демонові Nagios. Залежно від статусу, отриманого на хостах і сервісах, робляться відповідні дії [18]. На приведеному нижче рис. 2.4 показана активна перевірка.

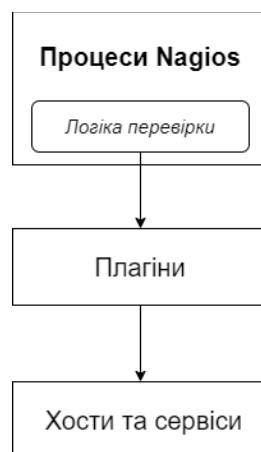


Рисунок. 2.4 — Активна перевірка

Активні перевірки виконуються через рівні проміжки часу, як визначено в директивах `check_interval` і `retry_interval`.

Пасивні перевірки виконуються зовнішніми процесами, а результати повертаються в Nagios для обробки. Зовнішнє застосування перевіряє стан хостів і сервісів і записує результат в зовнішній командний файл [19].

Коли демон Nagios читає зовнішній командний файл, він читає і відправляє усі пасивні перевірки в черги, щоб обробити їх пізніше. Періодично, коли ці перевірки обробляються, відправляються повідомлення залежно від інформації в результаті перевірки [17].

На приведеному нижче рис. 2.5 показана пасивна перевірка.



Рисунок 2.5 — Пасивна перевірка

Таким чином, різниця між активною і пасивною перевіркою полягає в тому, що активні перевірки виконуються Nagios, а пасивні перевірки зовнішніми

застосунками. Має сенс робити ці перевірки, коли неможливо контролювати хости і сервіси на постійній основі [18].

2.3.2 Відстеження поточного стану хостів та сервісів

Nagios зберігає стан хостів і сервісів, які він відстежує, щоб визначити, чи правильно вони працюють. Існувала б безліч випадків, коли збої відбуватимуться випадково і вони будуть тимчасовими. Отже, ця система моніторингу використовує «стани» для перевірки поточного стану хоста або служби.

Поточний стан відстежуваних сервісів і хостов визначається двома компонентами:

- статус служби або хоста (OK, Warning, Critical, Unknown, UP, Down, Unreachable);
- тип стану, в якому знаходиться служба або хост.

Коли Nagios перевіряє стан хостів, він може визначити, коли хост перемикається між станами "Up", "Down" і "Unreachable", і вжити відповідні заходи. Ці зміни статусу приводять до різних типів стану (HARD або SOFT), які можуть ініціювати запуск обробників подій і відправку повідомлень контактній особі. Коли хости міняють стан занадто часто, вони вважаються такими, що "блимають" (англ. "flapping") [20].

Перевірки статусу сервісів виконуються плагінами, які можуть повертати стан "OK", "Warning", "Unknown" або "Critical". Ці стани плагіну безпосередньо переводяться в стани служби. Наприклад, плагін, який повертає стан "Warning", буде викликати у сервісу стан "Warning". Стан сервісів змінюється так само, як і стан вузла [20].

У Nagios Core є два типи станів – SOFT ("м'яка" помилка) і HARD ("жорстка" помилка). Ці типи станів є важливою частиною логіки моніторингу і використовуються для визначення того, коли виконуються обробники подій і коли відправляються повідомлення [18].

Щоб запобігти неправдивим спрацьовуванням тривожних сигналів через тимчасові проблеми, Nagios дозволяє визначити, скільки разів сервіс або хост мають бути (повторно) перевірені, перш, ніж проблема вважатиметься "реальною". Це контролюється опцією `max_check_attempts` у визначеннях хоста і служби. Директива `max_check_attempts` використовується для визначення того, коли Nagios повинен розглядати проблему як справжню загрозу для мережі. Установка цього значення в 1 змусить Nagios генерувати попередження без повторної перевірки хост [21].

Nagios перевірятиме хост або службу кожні X хвилин, і якщо є зміна стану, система моніторингу повторно перевірятиме Y разів з інтервалом Z хвилин:

- `check_interval` X ;
- `retry_interval` Z ;
- `max_check_attempts` Y .

Поки він це визначає, проблема визначається SOFT-станом.

SOFT-стани виникають в наступних ситуаціях:

- коли перевірка сервісу або хоста призводить до стану "не ОК" або "НЕ ПРАЦЮЄ" і перевірка служби ще не була (повторно) перевірена N - кількість разів, вказана директивою `max_check_attempts` у визначенні служби або хоста. Це називається м'якою помилкою;

- коли сервіс або хост відновлюється після м'якої помилки. Це вважається м'яким відновленням. Коли хости або сервіси переходять в стан SOFT, цей стан реєструється і виконуються обробники подій для обробки стану SOFT [18].

HARD-стану виникають в наступних ситуаціях:

- коли перевірка хоста або сервісу призводить до "не ОК" або "НЕ ПРАЦЮЄ", і вона була (повторно) проведена N - кількість разів (опція `max_check_attempts`) у визначенні хоста або служби. Цей стан серйозної помилки;
- коли хост або сервіс переходить з одного стану "жорсткої" помилки в інший стан помилки (наприклад, Warning в Critical);

- коли перевірка сервісу призводить до стану, відмінного від ОК, і відповідний хост знаходиться в стані DOWN або UNREACHABLE;
- коли хост або сервіс виходить із стану "жорсткої" помилки. Це вважається важким відновленням.

Коли вузли або сервіси переходять в HARD-стан, цей стан реєструється. Далі виконуються обробники подій для стану HARD і сповіщення вказаних осіб про проблему або відновлення хоста або сервісу [18].

2.3.3 Алгоритм роботи системи

Nagios збирає статистику роботи сервера за допомогою плагінів NRPE (Nagios Remote Plugin Executor), Checkmk або за допомогою мережевого протоколу SNMP. Nagios Core запускати плагін через заданий інтервал часу, та перевіряє стан віддаленої служби або хоста (рис 2.6).

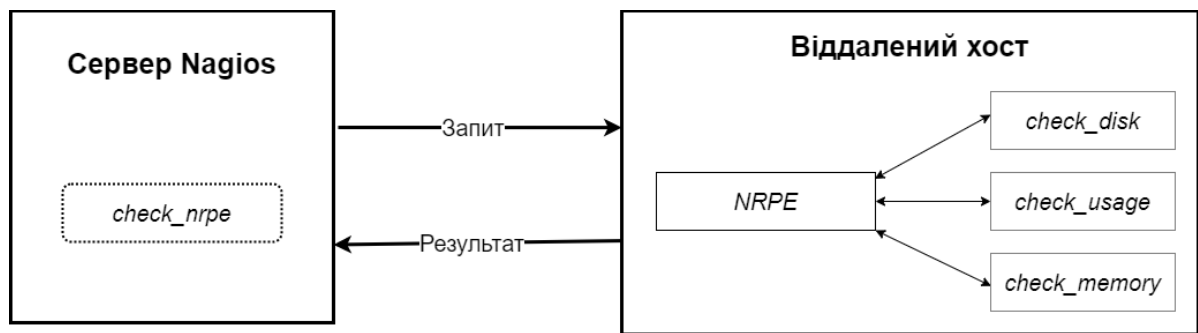


Рисунок 2.6 – Взаємодія сервера Nagios та віддаленого хоста

Для виконання перевірки стану хоста плагін робить задані дії, а потім просто повертає результати в Nagios Core. Nagios обробляє результати, отримані від плагіну, і робить усі необхідні дії (запуск обробника подій, відправка повідомлення тощо).

Обробник подій виконується при виникненні тих або інших подій, ініційованих перевірками сервісів або серверів. Цей перш, ніж хто-небудь з контактних осіб буде оповіщений про проблему.

Якщо значення метрики перевищує деякий зумовлений поріг, система відправляє попередження системному адміністратору, наприклад, на електронну пошту, та призначає статус службі або хосту (OK, Warning, Critical, Unknown).

Nagios записує дані моніторингу у базу даних за допомогою модуля NDMOD. Модуль NDMOD експортує дані про конфігурації, а також інформацію про різні події, які відбуваються в процесі моніторингу, із демона Nagios.

Кожен існуючий вузол (їх декілька в мережі підприємства) використовує модуль NDMOD для передачі даних, зібрані системою Nagios, на центральний, головний вузол.

Центральний вузол запускає демон NDO2DB, щоб обробити дані з кожного вузла і записати дані моніторингу у базу даних. Утиліта NDO2DB призначена для отримання даних, вилучених з компонентів NDMOD, і збереження їх у базі даних MySQL.

Centreon створює новий файл конфігурації у базі даних MySQL і генерує конфігурацію локального екземпляра Nagios. Потім Centreon викликає Nagios Plugins для передачі файлу конфігурації локального екземпляра на кожен підвузол і заміни початкового файлу конфігурації.

Centreon прочитує дані з бази даних і надає інформацію про моніторинг в режимі реального часу у своєму веб-інтерфейсі. Новий веб-інтерфейс експортує і передає дані з бази даних Centreon, MySQL.

Діаграма взаємодії компонентів системи наведена на рис. 2.7.

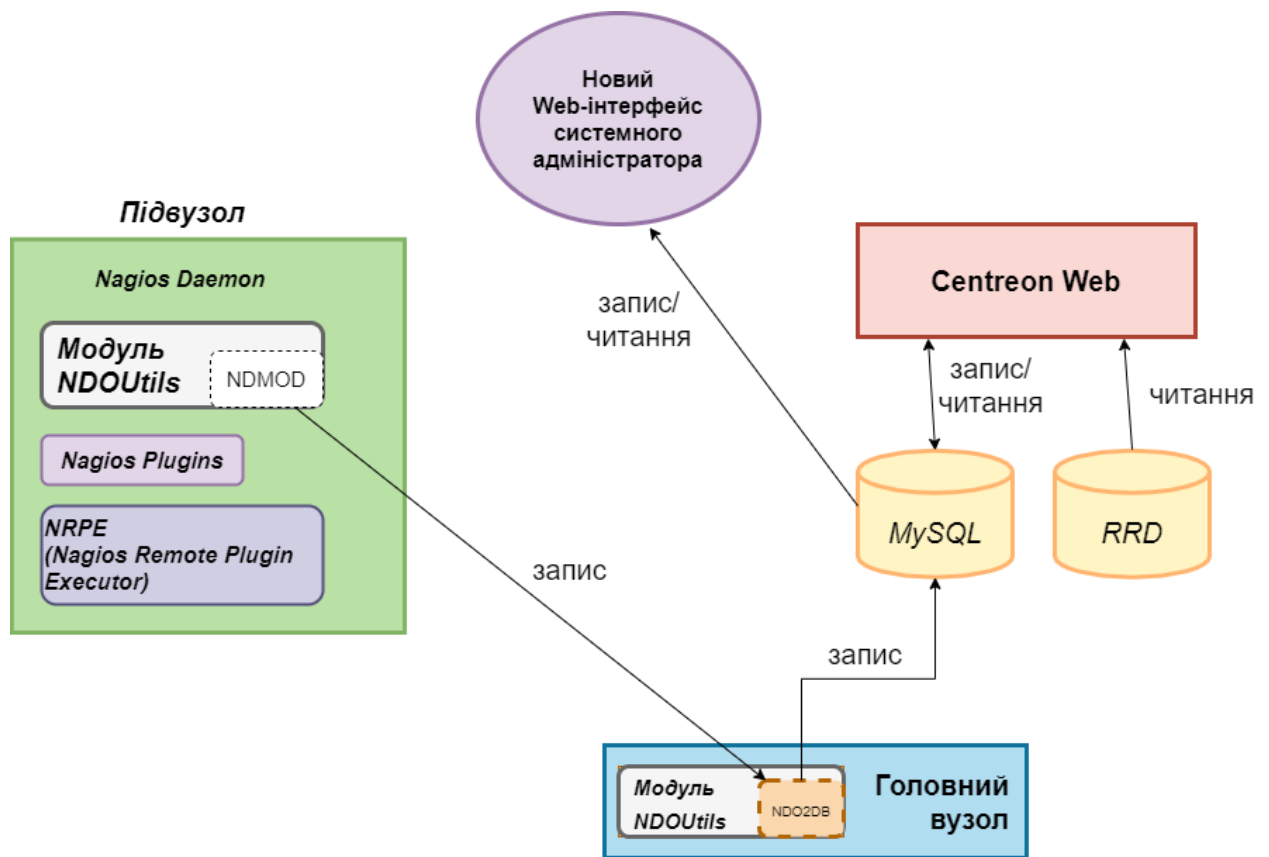


Рисунок 2.7 – Діаграма взаємодії компонентів системи

Блок – схема алгоритму наведена на рис. 2.8 для 1 вузла. Для інших вузлів алгоритм роботи аналогічний в паралельному режимі часу.

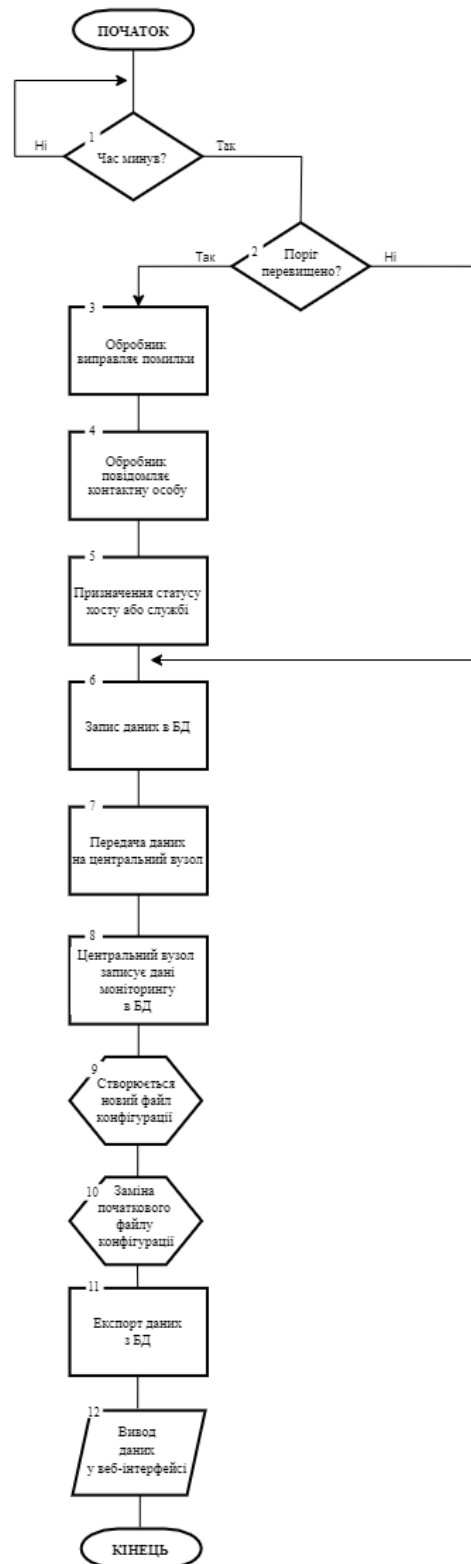


Рисунок 2.8 – Блок – схема алгоритму роботи системи

3 РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ

3.1 Розробка веб-інтерфейсу системи

Веб-інтерфейс системного адміністратора дозволяє користувачеві взаємодіяти з контентом або програмним забезпеченням, запущеним на видаленому сервері через будь-який веб-браузер. Контент завантажується з веб-сервера, і користувач може взаємодіяти з цим контентом у веб-браузері, який грає роль клієнта.

За головну сторінку проекту відповідає index.php.

Для верстання адаптивного дизайну дашборда була використана система сіток фреймворку Bootstrap.

У index.php створені чотири групи кнопок, які відповідають за фільтрацію інформації, що відображається, на дашборді (рис. 3.1) :

- сортування (sort order) — по тривалості чека, назві сервісу, сервісам;
- type — сповіщення, відфільтровані для кожного підрозділу компанії;
- status (статус сповіщення) — Acknowledged, Unacknowledged, All;
- state (стан сповіщення) — Critical, OK, All.



Рисунок 3.1 — Групи кнопок

Приклад визначення однієї з групи наведено в лістингу 3.1. Аналогічним чином створюються групи «type», «status», «state».

Листинг 3.1 – Створення групи кнопок «sort order»

```
<div class="btn-group" role="group" aria-label="sort order">
  <button type="button" id="timeSort" class="btn btn-outline-info"
onclick="changeSort('timeDESC');">
```

```

    Duration
    <span> <i id="timeSortArrow" class="fas fa-sort-up"></i>
    </span>
    </button>
    <button type="button" id="hostnameSort" class="btn btn-outline-
info" onclick="changeSort('hostnameASC'); " >
    Hostname
    <span> <i id="hostnameSortArrow" class="fas fa-sort-up"> </i>
</span> </button>
    <button type="button" id="serviceSort" class="btn btn-outline-info"
onclick="changeSort('serviceASC');" >
    Service
    <span> <i id="hostnameSortArrow" class="fas fa-sort-up"></i>
</span>
    </button></div>

```

У атрибутах onclick призначений обробник JavaScript, який по натисненню на кнопку Duration, Hostname і Service сортує об'єкти на сторінці за збільшенням (ASC) або по убутанню (DESC).

Сортування в групі кнопок "type" відбувається у функції changeType (type), прописаною в JavaScript-файлі. Оголошується дві порожніх строкових змінних - urlParams і tmpParams (Лістинг 3.2).

Лістинг 3.2 – Оголошення змінних

```

var urlParams = '';
var tmpParams = [];

```

Далі оголошується строкова змінна items – один з трьох способів фільтрації інформації на дашборде. Ця змінна містить об'єкт window.location.search.substr(1), який повертає вміст конкретного GET-параметра без знаку питання (Лістинг 3.3). Split("&") – для розділення на масиви ключ / значення.

Лістинг 3.3 – Оголошення змінної items

```

var items = window.location.search.substr(1).split("&");

```

GET-параметр – динамический параметр URL, который позволяет получить и даже изменить часть документа. В данном случае этим параметром является «type».

GET-параметр – динамічний параметр URL, який дозволяє отримати і навіть змінити частину документа. В даному випадку цим параметром є "type".

На виході отримуємо нову адресу URL, що містить в собі :

- window.location.protocol – використовуваний веб-протокол (https:);
- "/" – подвійний слеш;
- window.location.host – доменне ім'я веб-хостингу;
- window.location.pathname – шлях та ім'я файлу поточної сторінки;
- urlParams – отриманий URL, включаючи знак питання.

В результаті на екрані виводяться інциденти, які знаходяться під наглядом підрозділу ToIP (рис. 3.2). Так само відбувається фільтрація на кнопках інших груп.

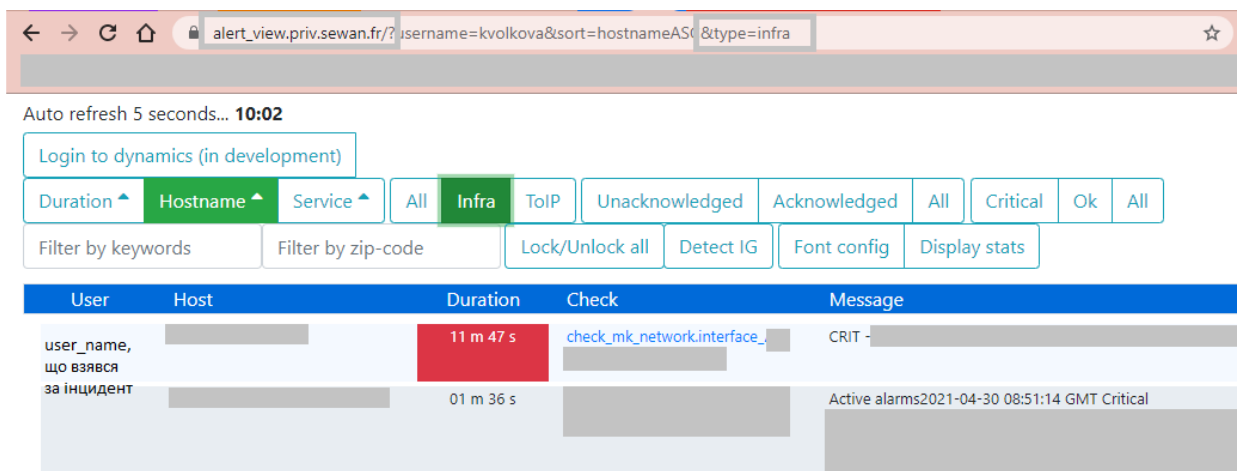


Рисунок 3.2 — Результат роботи функції фільтрації елементів

Нижче кнопок фільтрації розташований контейнер AlertsContainer, розділений на дві частини – блок (alerts), розділений на п'ять колонок і рядок (header), на якому написані назви колонок (Лістинг 3.4). Колонки мають назви, що відповідають характеру інформації у блоці (рис. 3.3):

- User – користувач, який працює з інцидентом;
- Host – назва хоста, на якому стався інцидент;

- Duration – тривалість стану сервісу;
- Message – повідомлення про помилку, яке безпосередньо відображає, що саме сталося.

User	Host	Duration	Check	Message
username, що взявся за інцидент		27 m 00 s	check_mk_network.interface_AT M2/0.166-aal5 layer	CRIT

Рисунок 3.3 — Контейнер AlertsContainer

Лістинг 3.4 — Оголошення контейнера AlertsContainer

```
<div id="AlertsContainer" class="container-fluid" style="height:
100%; ">
<div class="row border mb-1" id="header">
<div class="col-1" align="center" valign="center"> User </div>
<div class="col-2" valign="center"> Host </div>
<div class="col-1" align="center" valign="center"> Duration </div>
<div class="col-2" valign="center"> Check </div>
<div class="col-6" valign="center"> Message </div>
```

До головної сторінки проекту підключені декілька скриптів, написаних на JavaScript (Лістинг 3.5).

Лістинг 3.5 – Підключення скриптів JS

```
<script src="js/uncompressed/custom.js"></script>
<script src="js/uncompressed/animation.js"></script>
<script src="js/uncompressed/ajax.js"></script>
<script src="js/uncompressed/dynamics.js"></script>
```

Щоб не перезавантажувати сторінку кожного разу при натисненні кнопки, була введена технологія AJAX. Скрипт ajax.js містить обробники, які викликаються по натисненню на кнопки. Функції в ajax.js викликають запити XMLHttpRequest (наприклад, редирект на ...alert view/index.php?lastComments=true&....).

Об'єкти XMLHttpRequest використовуються для взаємодії з серверами. За допомогою цих запитів можна отримати дані за URL без повної перезагрузки сторінки. Це дозволяє веб-сторінці оновити частину вмісту, не перешкоджаючи роботі користувача. XMLHttpRequest часто використовується в програмуванні з використанням технології Ajax.

Також ці запити перехоплюють файл ajax.php, який утримує виклики вже конкретних бекенд-обработчиков на PHP (робота з базою, оновлення даних, виклик темплейтов за допомогою Smarty і так далі).

До проекту також підключений jquery.timericker, який забезпечує перезапуск усіх PHP-файлів кожні 5 секунд для оновлення даних на дэшборді.

3.2 Налаштування компонентів системи

Архітектура системи моніторингу буде мати наступні компоненти:

- Apache 2.4.6 — веб-сервер;
- Nagios XI — система моніторингу;
- Checkmk Raw Edition — розширення системи моніторингу Nagios ;
- Centreon Web 19.10.1 — графічний інтерфейс, ґрунтований на Nagios;
- PHP — скриптова мова загального призначення;
- технологія Smarty 3 — це компілюючий обробник шаблонів для PHP;
- веб-інтерфейс — новий вдосконалений інтерфейс.

Для установки системи були потрібні веб-сервер Apache, PHP і деякі бібліотеки, такі як gcc, glibc, glibc - common і GD, а також бібліотеки розробки. Для цього використовувався установник пакетів "yum" (Лістинг 3.6).

Лістинг 3.6 – Установка бібліотек

```
[root@user13]# yum install -y httpd httpd-tools php gcc glibc glibc-
common gd gd-devel make net-snmp
```

Наступний крок - створення нового користувача nagios за допомогою команди useradd і обліковому запису групи nagcmd, встановили пароль. Потім додали користувача nagios і користувача apache в групу nagcmd за допомогою команди usermod (Лістинг 3.7).

Лістинг 3.7 – Створення нового користувача

```
[root@user13]# useradd nagios
[root@user13]# groupadd nagcmd
[root@user13]# usermod -G nagcmd nagios
[root@user13]# usermod -a -G nagios,nagcmd www-data
```

Завантажили останні оновлення Nagios, розпакували їх і увійшли до каталогу (Лістинг 3.8).

Лістинг 3.8 – Завантажили оновлення

```
[root@user13]# wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.3.4.tar.gz
[root@user13]# tar -xzf nagios*.tar.gz
[root@user13]# cd nagios-*
```

Перед компіляцією програмного забезпечення визначили користувача і групу для користування Nagios. Виконали компіляцію програмного забезпечення командою sudo make all (Лістинг 3.9).

Лістинг 3.9 – Визначили користувача та групу

```
[root@user13]# sudo ./configure --with-nagios-group=nagios --with-command-group=nagcmd
[root@user13]# sudo make all
[root@user13]# sudo make install
[root@user13]# sudo make install-commandmode
[root@user13]# sudo make install-init
[root@user13]# sudo make install-config
```

Був використаний Apache в якості веб-сервера для інтерфейсу Nagios. Apache - кросплатформене програмне забезпечення. Apache має високу надійність і гнучкість. Основна робота Apache це обробка і відповідь на HTTP- запити і генерування динамічний зміст сторінок [22].

Nagios надає зразок файлу конфігурації, який був використаний в проекті.

Скопіювали файл конфігурації в каталог `usr/bin/install -c -m 644 sample-config/httpd.conf/etc/apache2/sites-available/nagios.conf`.

Встановили плагіни Nagios Plugins, щоб отримати набір інструментів для моніторингу різних служб.

Налаштували, скомпілювали і встановили плагіни (Лістинг 3.10).

Лістинг 3.10 – Установка плагінів

```
[root@user13]# sudo apt-get install build-essential libssl-dev gcc
[root@user13]# wget https://nagios-plugins.org/download/nagios-
plugins-2.3.3.tar.gz
[root@user13]# tar -xzf nagios-plugins*.tar.gz
[root@user13]# cd nagios-plugins*
[root@user13]# sudo ./configure --with-nagios-user=nagios --with-
nagios-group=nagios --with-openssl
[root@user13]# sudo make install
```

Встановлені Nagios і необхідні плагіни.

Конфігурація Nagios відбувається таким чином. За допомогою текстового редактора vim відкрили файл `nagios.cfg` для налаштування каталогу з хостами і прибрати символ `#`, розкаментували рядок `cfg_dir=/usr/local/nagios/etc/servers`. Створили папку для зберігання конфігурацій контрольованих серверів (Лістинг 3.11).

Лістинг 3.11 – Створення папки для конфігурацій

```
[root@user13]# sudo mkdir /usr/local/nagios/etc/servers
```

Далі зайшли у файл `contacts.cfg`, щоб змінити рядок `email`. На вказану електронну адресу приходитимуть повідомлення системи (Лістинг 3.12).

Лістинг 3.12 – Змінили електронну адресу

```
email user13@sevan.ua ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS
*****
```

Наступний крок - конфігурація веб-сервера Apache. Підключили модулі CGI та `mod_rewrite` (Лістинг 3.13).

Лістинг 3.13 – Підключення модулів

```
[root@user13]# sudo a2enmod rewrite
[root@user13]# sudo a2enmod cgi
```

Створили адміністративного користувача, щоб користуватися веб-інтерфейсом системи, задали пароль. Включили Apache (Лістинг 3.14).

Лістинг 3.14 – Створення користувача та активізація веб-серверу

```
[root@user13]# sudo htpasswd -c
/usr/local/nagios/etc/htpasswd.users admin_n
[root@user13]# sudo ln -s /etc/apache2/sites-available/nagios.conf
/etc/apache2/sites-enabled/
```

Перезавантажили Nagios Plugins, Apache и Nagios.

За умовчанням Nagios не надає файл конфігурації `systemd`, тому його треба створити самостійно. Файл містить інформацію, де `systemd` може знайти виконуваний файл Nagios і файли конфігурації.

Створили файл `nagios.service` і записали в нього вміст лістингу 3.15.

Лістинг 3.15 – Вміст файлу nagios.service

```
[Unit]
Description=Nagios
BindTo=network.target
[Install]
WantedBy=multi-user.target
[Service]
User=nagios
Group=nagios
Type=simple
ExecStart=/usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
```

Зберегли зміни у файлі та запустили сервіс `systemctl`.

Сервер Nagios готовий.

У цьому проекті використовується NRPE – це агент Nagios, що дозволяє здійснювати видалений моніторинг системи за допомогою сценаріїв, розміщених на віддалених хостах. Агент підлягає моніторингу такі ресурси, як використання диска, завантаження системи або кількість користувачів, що увійшли до системи в даний момент [23].

На видалений сервер завантажується файл `npre.tar.gz` і розпаковується. Далі виконується компіляція.

Створюються користувачі і групи користувачів, конфігураційні файли. Далі треба відредагувати файл `npre.cfg`, вказавши IP-адрес сервера Nagios.

На Nagios Server створюється файл конфігурації в каталозі `/usr/local/nagios/etc/servers/` для кожного видаленого хоста, який треба відстежувати. Наприклад, для хоста "opora9" створений файл `opora9.cfg` (Лістинг 3.16).

Лістинг 3.16 – Вміст файлу opora9.cfg

```
define host {
    use                linux-server
    host_name          opora9
    alias              Remote Host
    address            165.172.168.169
    register           1}

define service {
```

```

host_name                opora9
service_description      PING
check_command            check_ping!100.0,20%!500.0,60%
max_check_attempts       2
check_interval           3
retry_interval           2
check_period             24x7
check_freshness          1
contact_groups           admins
notification_interval    2
notification_period      24x7
notifications_enabled    1
register                 1}

```

Цей файл відстежуватиме, чи відповідає віддалений хост на команду "ping", використання диска хостом і чи запущена служба SSH.

Centreon — це програмний пакет з відкритим початковим кодом, який дозволяє контролювати усі інфраструктури і додатки, що становлять інформаційну систему. Новий веб-інтерфейс "спілкується" з платформою Centreon за допомогою створеного класу Centreon_Communicator.

Checkmk — це програмне забезпечення, розроблене на Python і C ++ для моніторингу IT-інфраструктури. Він використовується для моніторингу серверів, додатків, мереж, хмарних інфраструктур (загальнодоступних, приватних, гібридних), контейнерів, сховищ, баз даних і датчиків середовища [24] [25].

4 РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ

Метою роботи було підвищення надійності комп'ютерної мережі за рахунок впровадження системи моніторингу на підприємстві.

У зв'язку з тим, що моніторинг побудований на основі Nagios, з яким розробники і системні адміністратори вже мали величезний досвід роботи на підприємстві, витрати часу на навчання їх взаємодії із засобами системи були

мінімальні. Внаслідок інтеграції із вже існуючою і налагодженою системою, делегування прав на редагування зборок моніторингу, перегляд відомостей про стан системи і інші операції мало мінімальну трудомісткість.

У кінцевого користувача, тобто у системного адміністратора, є можливість створити власні плагіни Checkmk і без проблем впровадити їх в моніторинг. Модульність дозволяє легко і швидко змінювати конфігурацію відстежування. Знаючи Python, можна написати будь-яку функцію, яка витягне необхідну інформацію з бази даних, або написати власний скрипт сповіщення про інцидент. Плагін легко модернізується.

Факт того, що тепер програмні модулі розробляються здебільшого вручну, а не використовуються загальнодоступні плагіни Checkmk, надає можливість збору даних з об'єктів моніторингу, абсолютно будь-якого характеру, ідеально підлаштовуючи систему моніторингу під потреби компанії та її клієнтів. Це дозволяє вмить визначати характер інциденту, а отже значно понизити час простою мережі і вмить визначити фахівця відповідального за рішення виниклої задачі.

Система моніторингу мережі Nagios надає звіти про роботу мережевих компонентів щомісячно. У звіті надана статистика інцидентів, які були усунені безпосередньо самою системою моніторингу. Для наглядності результуюча таблиця була поділена на три частини (рис.4.1- 4.3).

Date	Notification Events		Nb of Alert	
	Warn	Crit	Warn	Crit
2021 10th May	695	736	312	603
2021 9th May	536	219	77	255
2021 8th May	387	238	78	139
2021 7th May	637	464	261	344
2021 6th May	1020	1436	627	1838
2021 5th May	652	746	410	6566
2021 4th May	757	674	347	383
2021 3rd May	620	727	284	480
2021 2nd May	322	276	78	278
2021 1st May	304	161	79	114
2021 30th April	424	675	258	2487
2021 29th April	711	1598	448	2798
2021 28th April	470	394	256	354
2021 27th April	736	1092	409	975
2021 26th April	548	389	309	360
2021 25th April	446	176	86	225
2021 24th April	377	169	127	131
2021 23rd April	465	349	260	316
2021 22nd April	542	641	300	665
2021 21st April	541	708	316	596
2021 20th April	474	434	328	342
2021 19th April	512	512	276	476
2021 18th April	416	143	87	207
2021 17th April	344	122	69	115
2021 16th April	492	385	228	327
2021 15th April	505	391	277	339
2021 14th April	526	416	264	381
2021 13th April	443	416	241	363
2021 12th April	576	561	327	429
2021 11th April	354	169	82	222
2021 10th April	362	230	90	206

Рисунок 4.1 — Перша частина таблиці

Solved (Warn + Crit)		Warn Solved	Crit Solved
820	89.6%	298	522
312	94.0%	76	236
204	94.0%	76	128
525	86.8%	249	276
2390	97.0%	606	1784
6874	98.5%	389	6485
657	90.0%	332	325
675	88.4%	270	405
281	78.9%	71	210
175	90.7%	75	100
2674	97.4%	240	2434
3078	94.8%	364	2714
541	88.7%	242	299
1248	90.2%	354	894
577	86.2%	292	285
294	94.5%	82	212
223	86.4%	122	101
483	83.9%	229	254
858	88.9%	283	575
798	87.5%	302	496
611	91.2%	315	296
611	81.2%	256	355
270	91.8%	83	187
163	88.6%	67	96
499	89.9%	217	282
543	88.1%	265	278
555	86.0%	242	313
546	90.4%	230	316
672	88.9%	310	362
279	91.8%	78	201
245	82.8%	76	169

Рисунок 4.2 — Друга частина таблиці

Crit solved < 15mn		Crit solved 15-60mn		Crit solved > 60mn	
199	38.1%	260	49.8%	63	12.1%
185	78.4%	38	16.1%	13	5.5%
93	72.7%	21	16.4%	14	10.9%
154	55.8%	77	27.9%	45	16.3%
1527	85.6%	182	10.2%	75	4.2%
5844	90.1%	553	8.5%	88	1.4%
154	47.4%	106	32.6%	65	20.0%
198	48.9%	114	28.1%	93	23.0%
172	81.9%	21	10.0%	17	8.1%
63	63.0%	27	27.0%	10	10.0%
2217	91.1%	121	5.0%	96	3.9%
2511	92.5%	103	3.8%	100	3.7%
145	48.5%	92	30.8%	62	20.7%
612	68.5%	113	12.6%	169	18.9%
159	55.8%	75	26.3%	51	17.9%
174	82.1%	30	14.2%	8	3.8%
71	70.3%	22	21.8%	8	7.9%
147	57.9%	46	18.1%	61	24.0%
412	71.7%	97	16.9%	66	11.5%
328	66.1%	85	17.1%	83	16.7%
141	47.6%	79	26.7%	76	25.7%
193	54.4%	84	23.7%	78	22.0%
162	86.6%	21	11.2%	4	2.1%
68	70.8%	23	24.0%	5	5.2%
163	57.8%	66	23.4%	53	18.8%
158	56.8%	78	28.1%	42	15.1%
191	61.0%	80	25.6%	42	13.4%
168	53.2%	85	26.9%	63	19.9%
214	59.1%	73	20.2%	75	20.7%
173	86.1%	21	10.4%	7	3.5%
103	60.9%	33	19.5%	33	19.5%

Рисунок 4.3 — Третя частина таблиці

На рис. 4.4 зображений графік, що відображає сумарну кількість ліквідованих інцидентів за 1 день упродовж одного місяця (з 10.04.2021 по 10.05.2021), на які необхідно звернути увагу (статус "Warning") або зовсім несправні хости і сервіси ("Critical").

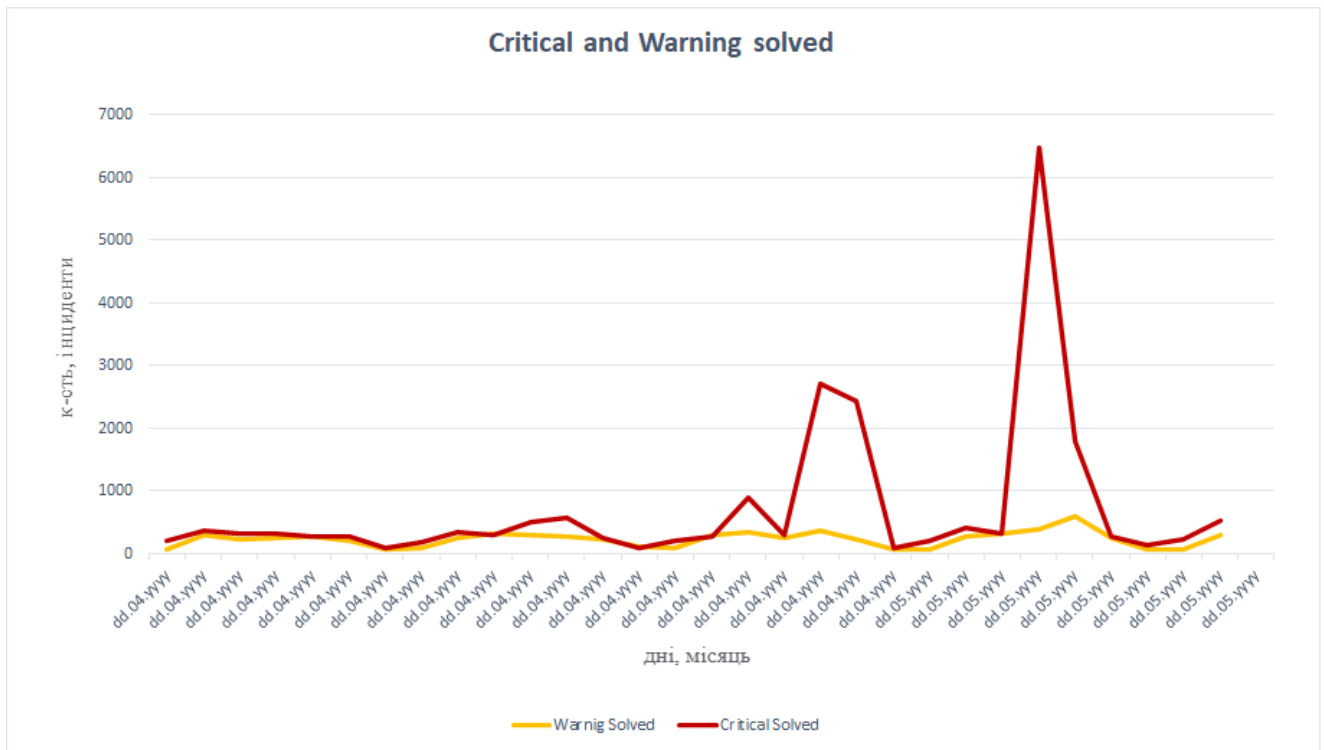


Рисунок 4.4 — Сумарна кількість інцидентів за 1 день

Інциденти із статусом "Critical" розділяється на 3 підгрупи:

- усунені менше, ніж за 15 хв;
- усунені протягом 15-60 хв;
- усунені більше, ніж за 60 хв.

Тенденція їх виникнення і усунення приведена на рис. 4.5.

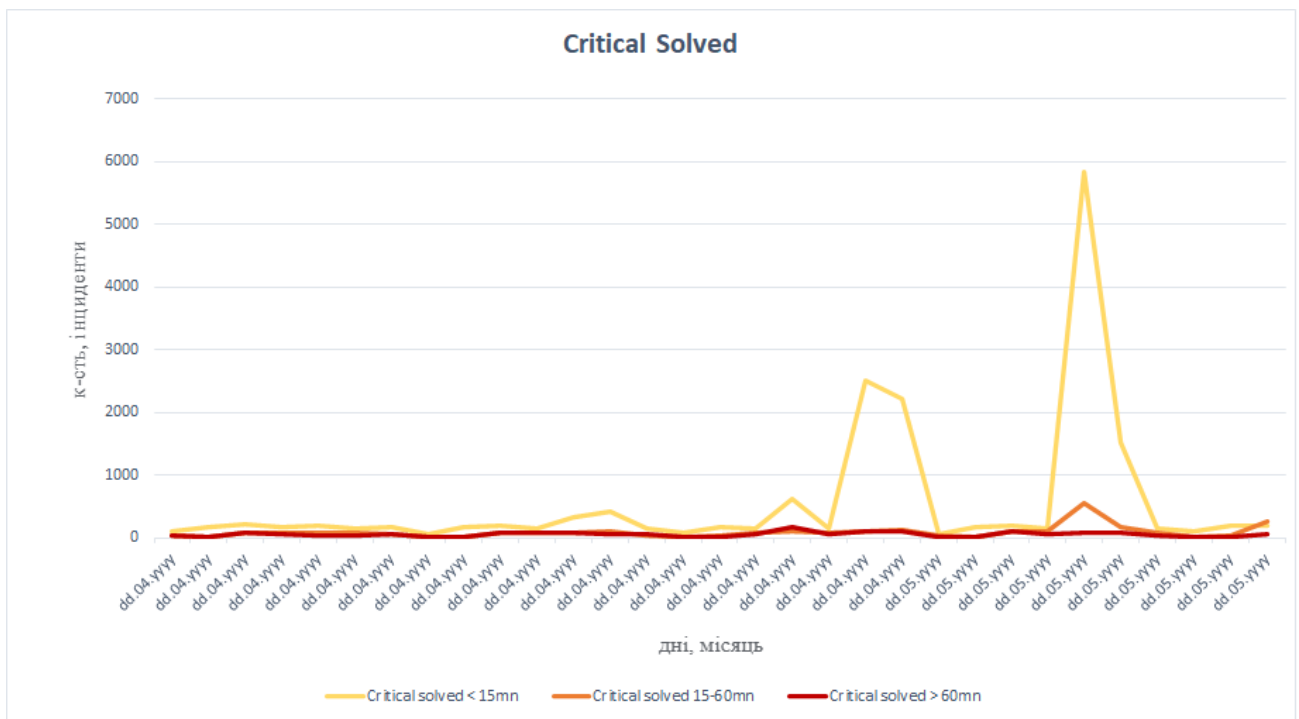


Рисунок 4.5 — Тенденція виникнення і ліквідації інцидентів

На підставі отриманої статистики можна зробити висновок про те, що більшість несправностей із статусом "Critical" ліквідовуються менше, ніж за 15 хвилин. Отже, система моніторингу здатна самостійно, без участі системного адміністратора, усунути несправність, тим самим значно понизити вірогідність простою мережі, підвищити надійність і відмовостійкість мережі підприємства.

Що стосовно веб-інтерфейсу, то варто згадати, що одним із завдань проекту була його модифікація, простіший інтерфейс. Веб-інтерфейс безпосередньо самої системи Nagios і його платформу Centreon складно назвати простими і, глянувши на екран, системний адміністратор може "загубитися" у безлічі рядків, кнопок тощо.

Тому було прийнято рішення створити новий інтерфейс, неабиякий і без зайвих кнопок. Ця розробка призначена тільки для відображення існуючих інцидентів на великому моніторі (телевізорі) (рис. 4.6) і мінімальної взаємодії з ними.

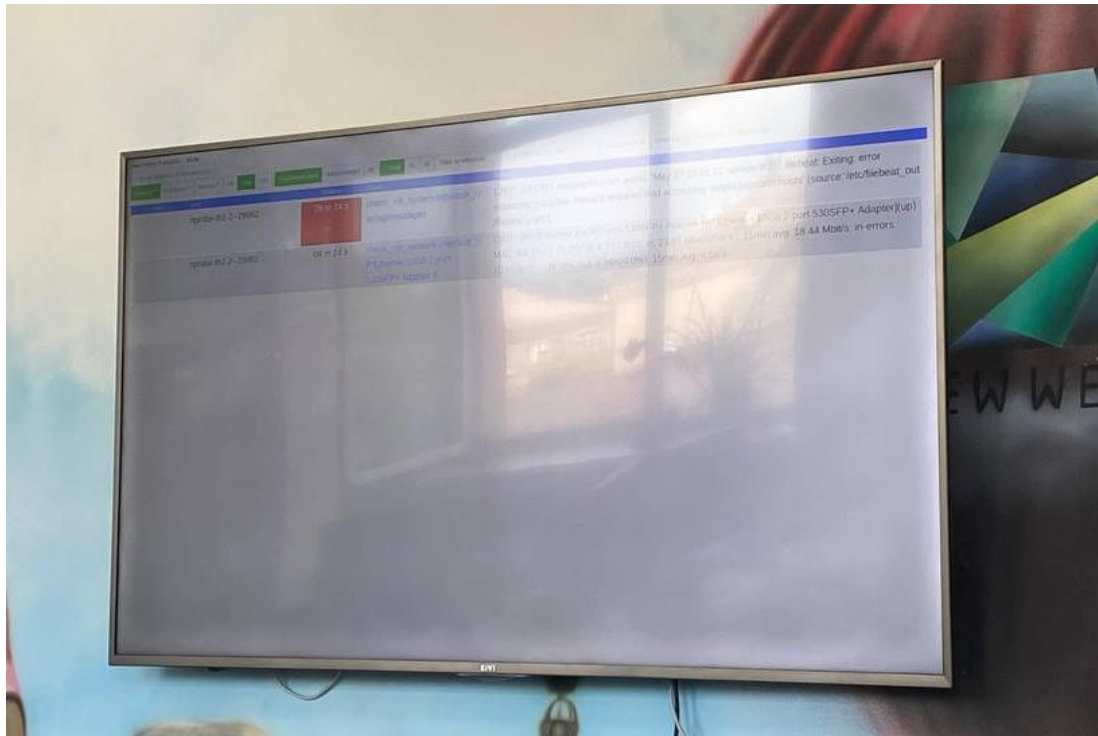


Рисунок 4.6 — Демонстрація інцидентів на телевізорі

У будь-який момент при необхідності детально налаштування моніторингу або сповіщення, можна звернутися безпосередньо до платформи Centreon. Інтерфейс Centreon простіший, ніж рідний інтерфейс Nagios, але він явно програє новому.

ВИСНОВКИ

Сьогодні бізнес будь-якої сфери діяльності залежить від працездатності його ІТ-інфраструктури. Щоб забезпечити цю працездатність, необхідно заздалегідь виявити вразливі місця в конфігурації мереж і систем, бути поінформованими про інциденти і оперативно робити заходи щодо їх вирішення. Для цих потреб в компаніях, де подібне стеження нездійснено за рахунок тільки фахівців, прийнято використати системи моніторингу.

Системи моніторингу бувають платні і вільно поширювані, а також розрізняються по своєму наповненню "з коробки", масштабованості, необхідним ресурсам і рівнем знань, необхідним для оптимального налаштування.

Хід виконання роботи повністю відповідав поставленим у вступі завданням. Спершу були висунені загальні вимоги до моніторингу спостережуваних об'єктів. Потім детально розглянуті існуючі системи моніторингу хостів і сервісів, проаналізовані їх позитивні і негативні сторони, відповідність їх сформульованим раніше вимогам.

В ході використання цієї системи відділами моніторингу і розробки програмного забезпечення були відмічені ряди позитивних аспектів впливу на роботу:

- підвищення відмовостійкості сервісів;
- простота інтеграції з різними зовнішніми програмами і сервісами;
- автоматизація рутинних завдань системного адміністратора;
- мінімальний час навчання роботи з системою;
- максимально швидко і точно інформує про можливий або виниклий інцидент.

З моменту впровадження системи пройшов місяць і, виходячи із статистичних даних, була відмічена доцільність впровадження системи моніторингу Nagios. Впровадження системи моніторингу мережі забезпечує додаткову надійність, повний

контроль і централізацію управління. Завдяки вбудованій системі сповіщення системним адміністраторам простіше швидко виявити проблему і визначити, чи представляє вона загрозу безпеки.

В цілому пропонований продукт для моніторингу хостів та служб, а також автоматизації завдань і інвентаризації допоможе значно скоротити час простою мережі, підвищити якість послуг, що надаються клієнтам компанією, а також значно поліпшити продуктивність, уникнувши, при цьому, невиправданих вкладень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. В.В. Кореньков, В.В. Мицын, П.В. Дмитриенко. Архитектура системы мониторинга центрального информационно - вычислительного комплекса ОИЯИ. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ. 2012. № 3. С. 31–42 [Электронный ресурс]. – Режим доступа: http://www.isa.ru/jitcs/images/documents/2012-03/31_42.pdf (дата обращения: 01.04.2021).
2. Мониторинг состояния/исправности оборудования сети – CDDI [Электронный ресурс]. – Режим доступа: <https://cddi.ru/monitoring-oborudovaniya/> (дата обращения: 01.04.2021).
3. Система мониторинга сети. База знаний – Мерион Нетворкс [Электронный ресурс]. – Режим доступа: <https://wiki.merionet.ru/servernye-resheniya/sistema-monitoringa-seti/> (дата обращения: 01.04.2021).
4. Monitoraggio di rete con software open-source – LIA [Электронный ресурс]. – Режим доступа: <http://lia.disi.unibo.it/Courses/AmmSistemi1617/monitoraggio.pdf> (data di accesso: 01.04.2021).
5. Servizi di monitoraggio per un’infrastruttura di gestione dati in ambito sanitario – Dessert Research Group, Federico II University of Naples [Электронный ресурс]. – Режим доступа: <http://www.dessert.unina.it/tesi/TESI%20Massimo%20D'Amico%20528-1538.pdf> (data di accesso: 01.04.2021).
6. Introduction to RRD - Round Robin Database – LorientPro [Электронный ресурс]. – Режим доступа: https://www.lorientpro.com/Products/Online_Documentation_V5/LorientProDoc_EN/V22-RRD_Collector_RRD_Manager/V22-A1_Introduction_RRD_EN.htm (date of access: 01.04.2021).
7. 10 Best Open Source Monitoring Software for IT Infrastructure – Geekflare [Электронный ресурс]. – Режим доступа: <https://geekflare.com/best-open-source->

monitoring-software/ (date of access: 01.04.2021).

8. 5 лучших бесплатных систем мониторинга ИТ-инфраструктуры – Networkguru.ru [Электронный ресурс]. – Режим доступа: <https://networkguru.ru/5-besplatnykh-sistem-monitoringa-it-infrastruktury/> (дата обращения: 01.04.2021).

9. Веб-интерфейс Zabbix. Zabbix: The Enterprise-Class Open Source Network Monitoring Solution [Электронный ресурс]. – Режим доступа: https://www.zabbix.com/ru/zabbix_web_frontend (дата обращения: 01.04.2021).

10. Zabbix vs Nagios. NetPing | Server Room Environmental Monitoring Solutions and Power Distribution Units [Электронный ресурс]. – Режим доступа: <http://www.netpingdevice.com/blog/sravnenie-sistem-monitoringa-zabbix-vs-nagios> (date of access: 01.04.2021).

11. Средства мониторинга ИТ-инфраструктуры – Издательство «Открытые системы» [Электронный ресурс]. – Режим доступа: <https://www.osp.ru/os/2015/04/13047967> (дата обращения: 01.04.2021)

12. Что такое Nagios система мониторинга? – Unixblog.org.ua [Электронный ресурс]. – Режим доступа: <https://unixblog.org.ua/monitoring/what-is-nagios-monitoring/> (дата обращения: 01.04.2021).

13. Купить Nagios XI лицензию в интернет-магазине Softkey.UA – SOFTKEY UA [Электронный ресурс]. – Режим доступа: https://www.softkey.ua/catalog/task_management/nagios-xi/ (дата звернения: 01.04.2021).

14. Nagios - Architecture - Tutorialspoint.com [Электронный ресурс]. – Режим доступа: https://www.tutorialspoint.com/nagios/nagios_architecture.htm (date of access: 01.04.2021).

15. Nagios Plugins · Nagios Core Documentation. Nagios - Network, Server and Log Monitoring Software [Электронный ресурс]. – Режим доступа: <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/plugins.html> (date of access: 01.04.2021).

16. Check_MK best Nagios plugin – IndianGNU.org [Электронный ресурс]. – Режим доступа: https://www.indiangnu.org/2012/check_mk-best-nagios-plugin/ (date of access: 01.04.2021).

17. check_mk - a performant and easy to use generic monitoring plugin – Nagios Exchange [Электронный ресурс]. – Режим доступа: https://exchange.nagios.org/directory/Plugins/*-Remote-Check-Tunneling/check_mk--2D-a-performant-and-easy-to-use-generic-monitoring-plugin/details (date of access: 01.04.2021).

18. Nagios - Краткое руководство – CoderLessons.com [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/devops/vyuchi-nagios/nagios-kratkoe-rukovodstvo> (дата обращения: 01.04.2021).

19. State Types · Nagios Core Documentation. Nagios – Network, Server and Log Monitoring Software. [Электронный ресурс]. – Режим доступа: <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/statetypes.html> (date of access: 01.04.2021).

20. Paolini A. Preparing plugin for nagios. Home – Agenda (Indico) [Электронный ресурс]. – Режим доступа: https://agenda.infn.it/event/4133/contributions/50572/attachments/35698/42120/prepare_Plugin_for_nagios.pdf (date of access: 01.04.2021).

21. Object definitions. Nagios - Network, Server and Log Monitoring Software [Электронный ресурс]. – Режим доступа: <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/objectdefinitions.html> (date of access: 01.04.2021).

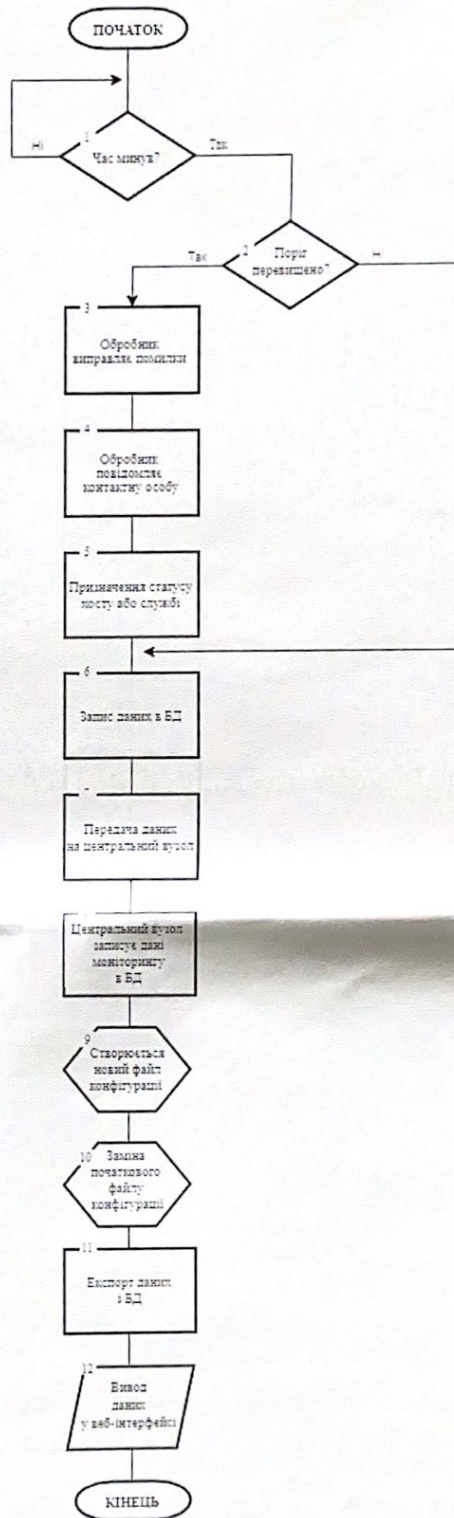
22. Что такое веб сервер Apache, описание и настройки – hostgid.net [Электронный ресурс]. – Режим доступа: <https://hostgid.net/baza-znaniy/hosting/chto-takoe-apache.html> (дата обращения: 01.04.2021).

23. Contributors to Wikimedia projects. Nagios – Wikipedia [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/Nagios#NRPE> (date of access:

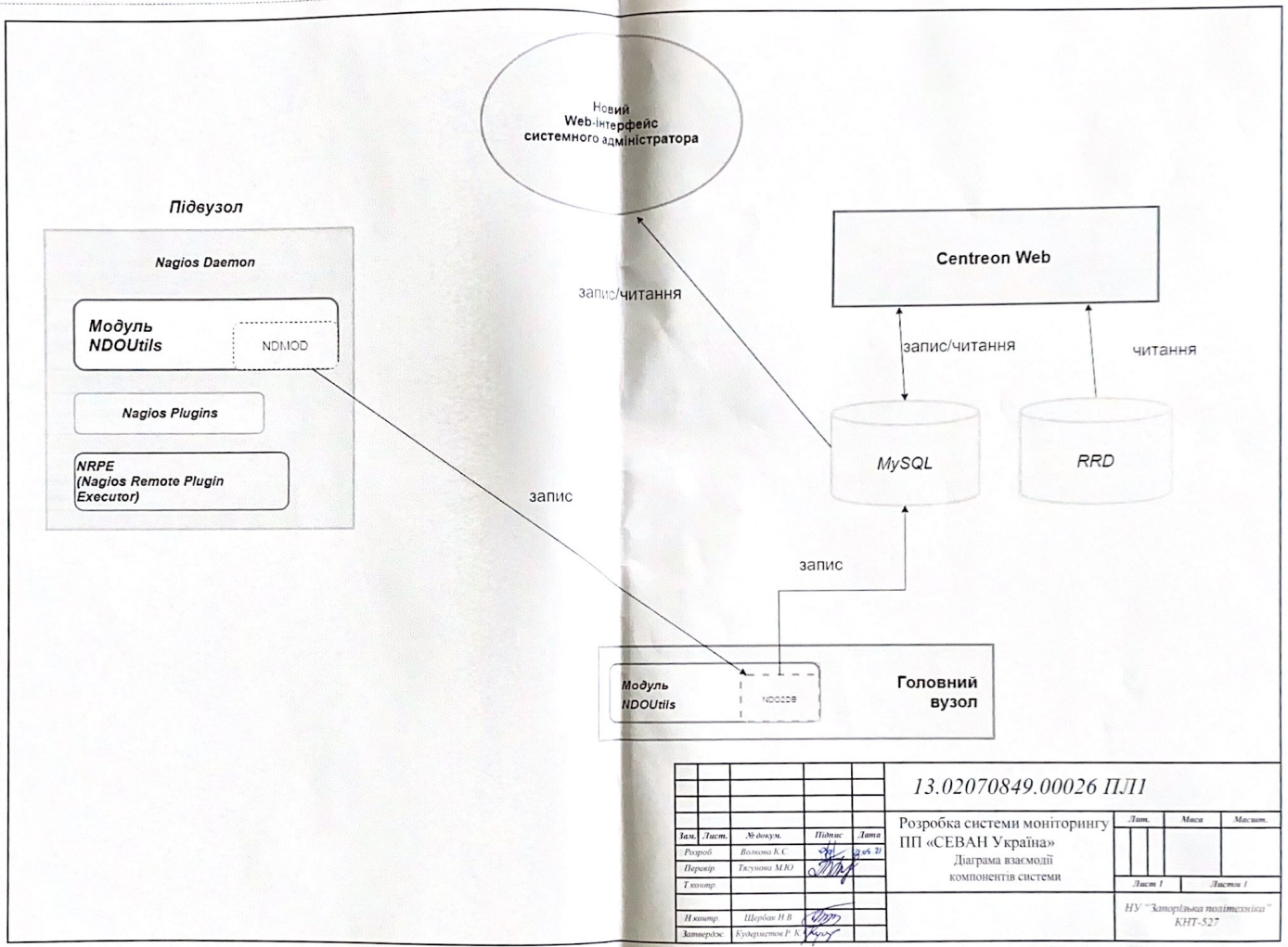
01.04.2021).

24. Checkmk guide on monitoring basics – Checkmk [Электронный ресурс]. – Режим доступа: <https://checkmk.com/monitoring> (date of access: 01.04.2021).

25. Contributors to Wikimedia projects. Check MK – Wikipedia [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Check_MK#cite_note-Usecase-3 (date of access: 01.04.2021).

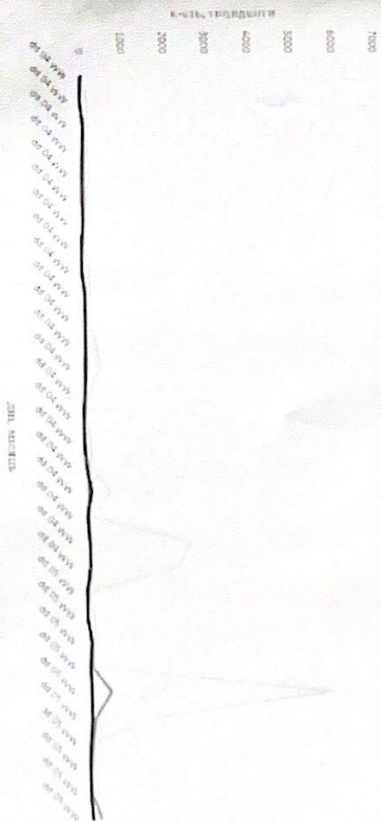


					13.02070849.00026 A1			
					Розробка системи моніторингу			
					ПП «СЕВАН Україна»			
					Блок-схема алгоритму роботи системи			
Зам.	Лист	№ докум.	Підпис	Дата				
Розроб		Волошин К.С.		06.04.11				
Перевір		Гасунова М.Ю.						
Т.контр								
П.контр		Щербак Н.В.						
Атестован		Кудрявцева Р.К.						
					Лист 1		Листів 1	
					НУ "Запорізька політехніка"		КНТ-527	

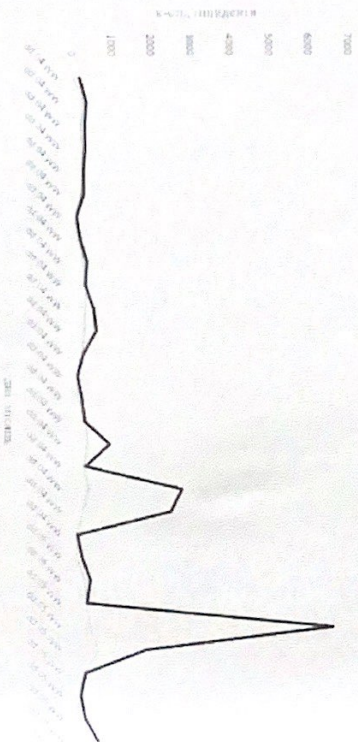


						13.02070849.00026 ПЛІ		
						Розробка системи моніторингу		
						ПП «СЕВАН Україна»		
						Діаграма взаємодії		
						компонентів системи		
Ім'я	Піст.	№ докум.	Підпис	Дата		Лист	Маса	Мас. лист.
Розроб		Володимир К. С.	<i>[Signature]</i>	26.02.21				
Перевір		Ткаченко М. Ю.	<i>[Signature]</i>					
Головний						Лист 1		Листов 1
Монітор		Шаріков Д. В.	<i>[Signature]</i>			НУ "Запорізька політехніка"		
Завершено		Кудриченко Р. В.	<i>[Signature]</i>					
						КНТ-527		

Critical Solved



Critical and Warning solved

[illegible]