МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Факультет математики і інформатики

Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

бакалавр

на тему:

«Розробка скриптів для системи управління мережевою інфраструктурою організації»

Виконав: студент 4 курсу, групи МФ-42

спеціальність 122 Комп'ютерні науки

освітньо-професійна програма

«Інформатика»

Статкевич А. О.

Керівник Меняйлов Є.С

Рецензент_____

(прізвище та ініціали)

Харків – 2023 року

ЗМІСТ

1. ВСТУП

1.1Формулювання мети роботи, задач та обґрунтування актуальності теми

Управління мережевою інфраструктурою відіграє вирішальну роль в організаціях, оскільки безпосередньо впливає на ефективність роботи, масштабованість і безпеку. Однак ручне керування мережевою інфраструктурою може бути трудомістким, тривалим і схильним до людських помилок. Щоб подолати ці проблеми, автоматизація та створення сценаріїв стали основними інструментами в управлінні мережею.

Метою даної кваліфікаційної роботи є розробка сценаріїв для системи управління мережевою інфраструктурою організації з метою оптимізації та автоматизації різноманітних мережевих завдань. Ці завдання включають керування конфігурацією, моніторинг пристроїв, оптимізацію продуктивності та усунення несправностей. Використовуючи мови сценаріїв і фреймворки, ми можемо підвищити ефективність, надійність і масштабованість процесів керування мережею.

Актуальність цієї теми очевидна у зростанні складності та масштабності сучасних мережевих інфраструктур. Організації стикаються зі зростаючими проблемами в управлінні різноманітними мережевими пристроями, забезпеченні узгоджених конфігурацій і оперативному реагуванні на події в мережі. Розробляючи сценарії, адаптовані до конкретних вимог організації, ми можемо значно скоротити ручні зусилля та людські помилки, дозволяючи мережевим адміністраторам зосередитися на завданнях вищого рівня.

Крім того, автоматизація за допомогою сценаріїв покращує гнучкість і оперативність керування мережею, забезпечуючи швидше розгортання, легшу масштабованість і проактивний моніторинг. Це не тільки підвищує ефективність роботи, але й покращує безпеку мережі шляхом забезпечення узгоджених

конфігурацій і впровадження стандартизованих політик безпеки у загальній інфраструктурі.

Ця робота має на меті зробити внесок у наявний масив знань шляхом розробки практичних сценаріїв та надання уявлень про їх впровадження та ефективність. Вирішуючи проблеми управління мережевою інфраструктурою за допомогою сценаріїв, ми можемо продемонструвати потенційні переваги та заохочувати впровадження автоматизації в організаціях.

1.2 Стислий огляд відомих результатів

Великі дослідження та практичні проекти підкреслили ефективність мов сценаріїв, таких як Python, Bash і PowerShell, для автоматизації мережевих завдань і керування компонентами інфраструктури. Ці мови пропонують потужні бібліотеки, фреймворки та інструменти, які полегшують автоматизацію мережі.

Наприклад, Python набув популярності в мережевій автоматизації завдяки своїй простоті, читабельності та широкій підтримці спільноти. Численні бібліотеки з відкритим вихідним кодом, такі як Paramiko та Netmiko, дозволяють мережевим інженерам автоматизувати керування конфігурацією, ініціалізацію пристроїв і завдання моніторингу.

Іншим відомим інструментом є Ansible, який забезпечує декларативний підхід до автоматизації мережі. Це дозволяє адміністраторам визначати бажані стани мережі за допомогою посібників на основі YAML, спрощуючи автоматизацію складних завдань на кількох пристроях і платформах.

Дослідження та практичні впровадження продемонстрували успішні результати в різних областях автоматизації мереж. До них належать оркестровка мережі, програмно-визначена мережа (SDN) та інтеграція практик DevOps у робочі процеси

керування мережею. Помітні досягнення включають ефективне надання мережевих послуг, динамічну реконфігурацію мережі та покращену безпеку мережі завдяки автоматизованому застосуванню політики.

Важливо налаштувати та адаптувати ці відомі результати до конкретних вимог і проблем, з якими стикаються окремі організації. Ця кваліфікаційна робота має на меті спиратися на ці висновки шляхом розробки спеціальних сценаріїв для управління мережевою інфраструктурою організації, зосереджуючись на покращенні можливостей автоматизації та оптимізації процесів керування мережею.

1.3 Відомості про одержані результати та їх новизна

Отримані результати кваліфікаційної роботи демонструють успішну розробку сценаріїв для управління мережевою інфраструктурою організації. Впроваджені алгоритми, включаючи алгоритм інвентаризації пристроїв, алгоритм виявлення мережі, алгоритм моніторингу пристроїв і алгоритм керування конфігурацією пристроїв, довели свою ефективність в автоматизації різних завдань керування мережею.

Новизна отриманих результатів полягає в комплексному підході до управління мережевою інфраструктурою, використовуючи рішення на основі сценаріїв для покращення процесів виявлення пристроїв, управління запасами, моніторингу та керування конфігурацією. Завдяки використанню автоматизації та спеціальних сценаріїв розроблене рішення забезпечує ефективне керування мережевими пристроями, покращує видимість мережі, проактивний моніторинг і спрощене керування конфігурацією.

Реалізовані сценарії та алгоритми пропонують кілька нових функцій і внесків у сферу управління мережею. Вони включають автоматичне виявлення та збір атрибутів пристрою, моніторинг справності та продуктивності пристрою в реальному часі.

Сценарії, які були розроблені у цій роботі, надають мережевим адміністраторам практичні інструменти для покращення можливостей усунення несправностей, зменшення ручних зусиль і полегшення своєчасного виявлення та вирішення мережевих проблем.

Результати кваліфікаційної роботи дають цінну інформацію про практичне впровадження сценаріїв керування мережею та підкреслюють їх потенційні переваги для організацій. Розроблені сценарії пропонують масштабоване та гнучке рішення, яке можна налаштувати та розширити відповідно до конкретних вимог керування мережею. Отримані результати поповнюють сукупність знань у галузі управління мережами та створюють основу для подальших досліджень і розробок у цій галузі.

1.4 Теоретичне та практичне значення результатів, можливі області використання результати

Теоретичне значення: неможливо переоцінити переваги сценаріїв у автоматизації та оптимізації мережевих операцій. Розробка сценаріїв для системи управління мережевою інфраструктурою організації дає значні теоретичні наслідки, прокладаючи шлях до нових практик і методів для вирішення певних проблем мережі. Це дослідження дає цінну інформацію, яка може допомогти в подальших дослідженнях у сфері автоматизації мереж і стимулювати подальший прогрес у механізмах керування мережею.

Практичні наслідки розробки цього сценарію також значні. Автоматизуючи конфігурацію та роботу мережі організації, ці сценарії потенційно забезпечують швидший час відповіді, покращують продуктивність системи та покращують масштабованість — усе це сприяє загальній надійності та ефективності мережевої інфраструктури організації. Крім того, автоматизація звільняє мережевих

адміністраторів від повсякденних, чорних завдань, дозволяючи їм зосередитися на більш важливій діяльності.

Розроблені сценарії мають потенційне застосування в різних сферах управління мережею в різних галузях промисловості та організаціях. Можливі сфери використання включають:

- Корпоративні мережі: великі підприємства зі складною мережевою інфраструктурою можуть використовувати ці сценарії для автоматизації оновлень конфігурації, моніторингу пристроїв і оптимізації продуктивності. Це забезпечує більш ефективне керування мережею та допомагає забезпечити стабільність і безпеку мережі.
- Постачальники послуг: Постачальники мережевих послуг можуть скористатися перевагами цих сценаріїв, автоматизуючи надання послуг, моніторинг мережі та усунення несправностей. Це може призвести до швидшого надання послуг, покращення взаємодії з клієнтами та зниження операційних витрат.
- Хмарна інфраструктура: Сценарії, розроблені для керування мережевою інфраструктурою, можна застосовувати до хмарних середовищ, забезпечуючи автоматизоване керування конфігурацією, оптимізацію робочого навантаження та забезпечення безпеки. Це допомагає організаціям ефективно керувати своїми хмарними мережами та забезпечувати стабільну продуктивність і безпеку.
- Мережі IoT: Інтернет речей (IoT) відноситься до мережі взаємопов’язаних фізичних пристроїв, транспортних засобів, приладів та інших об’єктів із вбудованими датчиками, програмним забезпеченням і можливостями підключення. Ці пристрої збирають і обмінюються даними, що дозволяє їм взаємодіяти та спілкуватися один з одним. Оскільки мережі IoT продовжують поширюватися, наші розроблені сценарії відіграють вирішальну роль в управлінні та захисті цих мереж. Завдяки автоматизації в основі наших сценаріїв полегшується підключення пристроїв, керування конфігурацією та моніторинг у

середовищах IoT. Спрощуючи ці процеси, наші сценарії підвищують ефективність і результативність керування мережами IoT, забезпечуючи безперебійну роботу та оптимальну продуктивність. Практична значущість отриманих результатів і потенційні сфери їх використання підкреслюють реальну застосовність і цінність розроблених скриптів. Вони пропонують вирішення проблем, з якими стикаються організації в управлінні своєю мережевою інфраструктурою, що веде до підвищення ефективності, підвищеної безпеки та оптимізації мережевих операцій.

2. ОСНОВНА ЧАСТИНА

2.1.Постановка задачі

Метою кваліфікаційної роботи є розробка спеціалізованого автоматизованого рішення, яке може покращити систему управління мережевою інфраструктурою організації. Цей метод управління має бути керованим, вільним від помилок і масштабованим; всі характеристики відсутні в ручних процесах. Для цього робота буде зосереджена на розробці сценаріїв, які спрощують ключові операції керування мережею, такі як налаштування, моніторинг, оптимізація та усунення несправностей. Ці сценарії будуть розроблені спеціально для задоволення конкретних потреб організації, що потенційно забезпечить їм узгоджену конфігурацію, моніторинг у режимі реального часу та більш ефективне усунення несправностей.

Дана робота спрямована на вирішення наступних завдань:

- Розробка налаштованих сценаріїв: сценарії повинні бути адаптовані до потреб організації в управлінні мережевою інфраструктурою, вирішуючи конкретні проблеми та вимоги.
- Автоматизація керування конфігурацією: сценарії, які автоматизують розгортання та керування конфігураціями мережевих пристроїв, забезпечуючи узгодженість, зменшуючи помилки та прискорюючи розгортання.
- Автоматизація моніторингу пристроїв: сценарії, які забезпечують проактивний моніторинг мережевих пристроїв, забезпечуючи раннє виявлення проблем з продуктивністю та загрози безпеці, а також сприяючи швидкому реагуванню та зменшенню ризиків.
- Керування мережею: сценарії, які автоматизують ключові завдання керування мережею, зменшуючи ручні зусилля, підвищуючи ефективність роботи та

дозволяючи мережевим адміністраторам зосередитися на стратегічних ініціативах.

- Підвищення безпеки мережі: сценарії, які забезпечують виконання політик безпеки, виявляють потенційні вразливості та реагують на них, а також підвищують загальну безпеку мережевої інфраструктури.
- Масштабованість і адаптивність: сценарії, які можуть адаптуватися до зростаючої мережевої інфраструктури організації, забезпечуючи масштабованість і гнучкість для адаптації до мінливих потреб бізнесу.

В рамках даної роботи можна визначити наступні завдання:

- Дослідження та аналіз систем управління мережевою інфраструктурою
- Збір вимог
- Розробка архітектури сценаріїв
- Розробка сценаріїв

2.2. Технічні вимоги

Спираючись на постановку задачі для системи управління мережевою інфраструктурою організації керуватимуться такі технічні вимоги:

- Мова програмування: Python

Python користується широкою перевагою для мережевої автоматизації та керування інфраструктурою через його простоту, читабельність і великі бібліотеки (наприклад, Paramiko, Netmiko, NAPALM), спеціально розроблені для мережевих операцій.

- База даних: MySQL

MySQL є популярною системою керування реляційною базою даних (RDBMS) з відкритим кодом.

- Керування конфігурацією
- Моніторинг мережі та сповіщення
- Автоматичне усунення несправностей
- Автоматизація безпеки
- Інтеграція з існуючими системами
- Масштабованість і продуктивність
- Надійна обробка помилок і ведення журналів

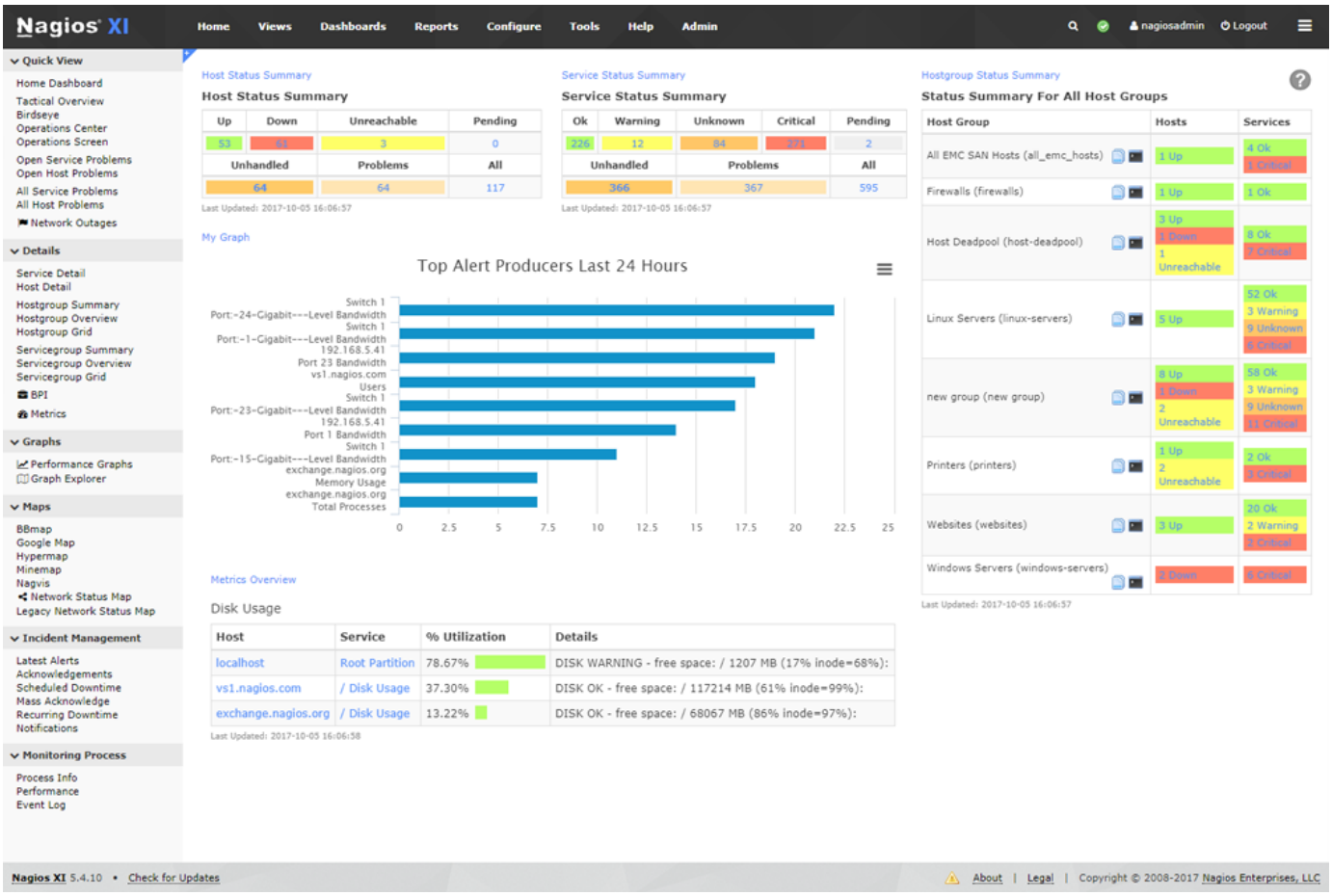
Вибравши Python мовою програмування та MySQL, як базу даних, нам вдасться використати сильні сторони Python для автоматизації мережі та можливості реляційної бази даних MySQL для зберігання пов’язаних з мережею даних. Ці варіанти допоможуть забезпечити сумісність, масштабованість і зручність обслуговування системи керування мережевою інфраструктурою.

2.3.Розвинутий огляд сучасного стану справ в області

В області мережеві інфраструктури охоплюють різноманітні типи, включаючи локальні мережі (LAN), глобальні мережі (WAN) і хмарні мережі. Ці мережі відрізняються за масштабом, складністю та різноманітністю компонентів. Організації розгортають маршрутизатори, комутатори, брандмауери, сервери та пристрої зберігання даних для побудови своїх мереж і керування ними. Наприклад, Cisco, Juniper Networks і Huawei є відомими постачальниками мережевого обладнання, що використовується в регіоні.

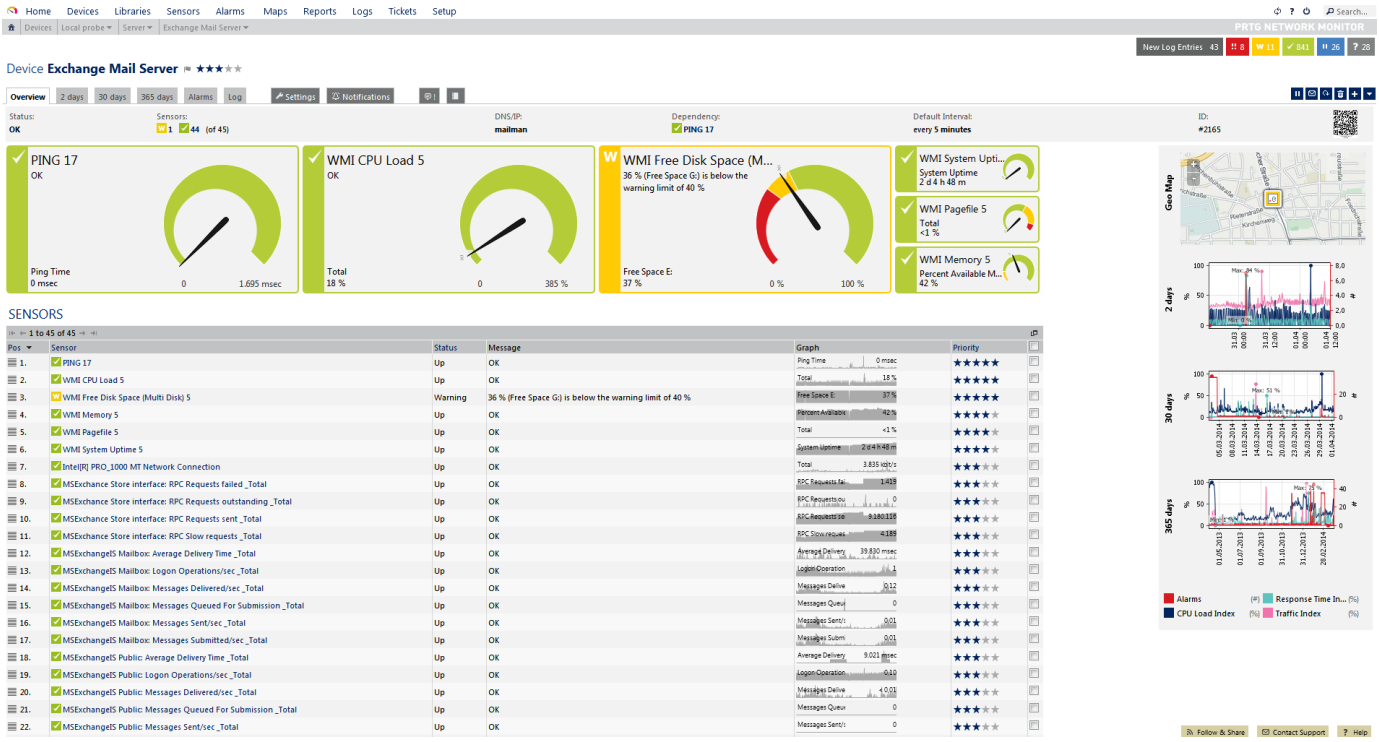
Організації в регіоні використовують різні методології, інструменти та техніки для керування своєю мережевою інфраструктурою. Мається на увазі поєднання ручних процесів і автоматизованих рішень. Інструменти керування конфігурацією, такі як Ansible і Puppet, широко використовуються для автоматизації розгортання та налаштування мережевих пристроїв. Інструменти моніторингу мережі, такі як Nagios

(*мал. 1*), PRTG *(мал. 2)* і SolarWinds *(мал. 3)*, дозволяють відстежувати продуктивність і доступність мережі в реальному часі. Заходи безпеки включають брандмауери, системи виявлення та запобігання вторгненням (IDS/IPS) і віртуальні приватні мережі (VPN).



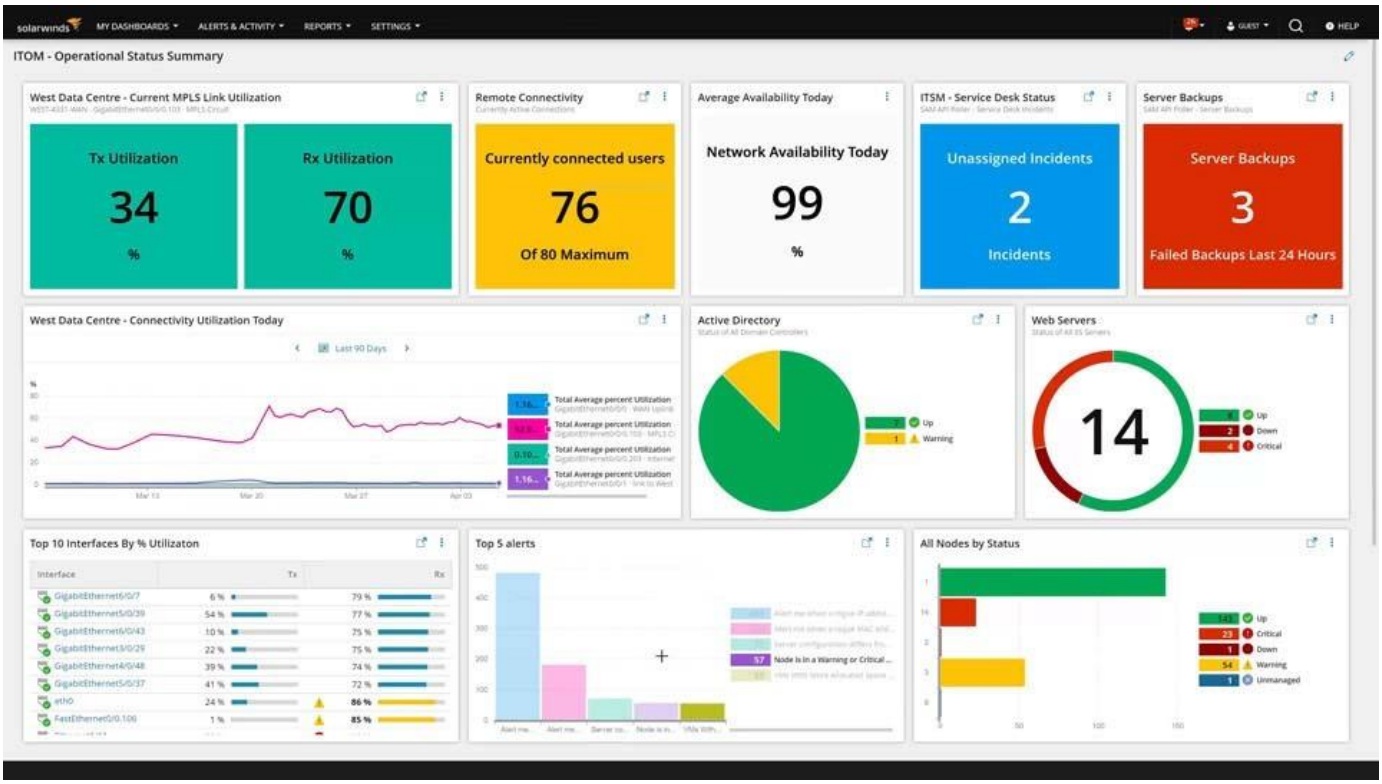
мал. 1 — приклад моніторингу інструмента "Nagios"

(https://www.nagios.com/products/nagios-xi/)



мал. 2 — приклад моніторингу інструмента "Nagios"

(https://www.paessler.com/prtg)



мал. 3 — приклад моніторингу інструмента "SolarWinds"

(https://www.solarwinds.com/hybrid-cloud-observability)

Існують також декілька проблем та обмежень в управлінні мережевою інфраструктурою, які заважають організаціям досягати своїх цілей. Проблеми з масштабованістю виникають із зростанням розмірів і складності мереж. Необхідно впроваджувати рішення, які можуть обробляти збільшений трафік і більшу кількість пристроїв. Відсутність автоматизації призводить до ручного втручання, яке може зайняти багато часу та спричинити помилки. Неузгодженість конфігурації мережевих пристроїв може спричинити проблеми з роботою та вразливості безпеки. Вузькі місця в роботі мережі можуть виникати через застаріле обладнання або неадекватний дизайн мережі. Обмеження ресурсів, наприклад обмежений бюджет і брак кваліфікованого персоналу, можуть перешкоджати ефективному управлінню мережею.

Нові тенденції та технології постійно з'являються в галузі управління мережевою інфраструктурою, формуючи спосіб організації проектування та експлуатації своїх

мереж. Ці вдосконалення пропонують можливості для підвищення ефективності, гнучкості та безпеки мережі. Деякі з помітних тенденцій і технологій в управлінні мережевою інфраструктурою включають:

- Хмарні рішення: моделі “Інфраструктура як послуга” (IaaS) і “Програмне забезпечення як послуга” (SaaS) надають організаціям гнучкі та масштабовані параметри для їх мережевої інфраструктури. Використання хмарних рішень забезпечує швидке розгортання, розподіл ресурсів за вимогою та централізоване керування, що дозволяє організаціям ефективно адаптуватися до мінливих потреб бізнесу.
- Програмно-визначена мережа (SDN): SDN — це мережева архітектура, яка відокремлює площину керування від площини даних, що дозволяє централізоване керування та програмування мережевих ресурсів. Відокремлюючи функції керування мережею та пересилання, організації можуть досягти більшої гнучкості, спрощеної конфігурації мережі та покращеної масштабованості. SDN забезпечує динамічне надання мережі, оптимізацію трафіку та ефективний розподіл ресурсів.
- Віртуалізація мережевих функцій (NFV): NFV дозволяє віртуалізувати мережеві служби, які традиційно виконуються спеціальними апаратними пристроями. Це дозволяє організаціям абстрагувати та консолідувати мережеві функції, такі як брандмауери, маршрутизатори та балансувальники навантаження, у програмні віртуальні екземпляри. Цей підхід віртуалізації зменшує залежність від фізичної інфраструктури, спрощує керування мережею та покращує використання ресурсів і масштабованість.
- Розширені заходи безпеки: оскільки кіберзагрози продовжують розвиватися, розширені заходи безпеки є вирішальними для керування мережевою інфраструктурою. Розвідка загроз, яка використовує дані та аналітику в реальному часі для виявлення та пом’якшення загроз, допомагає організаціям

випереджати потенційні атаки. Аналітика на основі машинного навчання розширює можливості виявлення загроз, аналізуючи моделі мережевого трафіку та виявляючи аномалії. Мережа з нульовою довірою, де контроль доступу базується на ідентифікації користувача та контекстній інформації, забезпечує підвищену безпеку, припускаючи, що жодному користувачеві чи пристрою не можна довіряти.

Ці нові тенденції та технології пропонують організаціям можливість оптимізувати методи управління мережевою інфраструктурою. Використовуючи хмарні рішення, програмно визначені мережі, віртуалізацію мережевих функцій і впроваджуючи розширені заходи безпеки, організації можуть підвищити продуктивність, гнучкість і стійкість мережі перед обличчям змінних вимог і загроз безпеці.

Існують найкращі практики, які допомагають забезпечити ефективну розробку сценаріїв для системи керування мережевою інфраструктурою організації. Дотримуючись цих правил, може підвищитись ефективність, надійність і безпека сценаріїв. Нижче наведено кілька ключових передових практик, які варто розглянути:

- Модульність: розбиття сценаріїв на модульні компоненти сприяє багаторазовому використанню та полегшенню обслуговування.
- Обробка помилок: реалізація надійних механізмів обробки помилок забезпечує ефективну обробку винятків.
- Перевірка введених даних: перевірка та дезінфекція введених користувачем даних запобігає вразливості системи безпеки та пошкодженню даних.
- Ведення журналів і звітування: механізми журналювання та створення звітів допомагають у вирішенні проблем і аудиті.
- Заходи безпеки: застосування відповідних заходів безпеки захищає конфіденційну інформацію.

- Контроль версій: використання систем контролю версій дозволяє легко відстежувати та керувати версіями сценаріїв.
- Оптимізація продуктивності: оптимізація сценаріїв підвищує ефективність за рахунок мінімізації мережевих запитів і оптимізації обробки даних.
- Постійне вдосконалення: регулярний перегляд та оновлення сценаріїв включає відгуки та нові технології.

Використовуючи ці найкращі практики, можна розробити сценарії, які спрощують керування мережевою інфраструктурою, підвищують надійність і загальну ефективність роботи.

2.4. Опис моделі вимог до інформаційної системи

У цьому розділі ми заглибимося в складні деталі моделі системних вимог. Ця модель служить безцінним планом для розробки сценаріїв, вказуючи різні функціональні можливості та обмеження. Такий підхід сприяє тому, що кінцевий продукт відповідає цілям цієї роботи.

2.4.1 Функціональні вимоги:

i) Модульність сценаріїв: щоб підтримувати розробку повторно використовуваних компонентів для різних завдань керування мережею, система повинна містити модульне створення сценаріїв.

ii) Автоматизація завдань конфігурації мережі: створення сценаріїв має дозволити автоматизувати загальні завдання конфігурації мережі, такі як надання пристрою, збереження резервних копій конфігурації та оновлення мікропрограми.

iii) Обробка помилок і звітування: система повинна мати оперативні механізми обробки помилок і здатна створювати вичерпні звіти про результати виконання сценарію.

iv) Сумісність з різними мережевими пристроями та постачальниками: щоб забезпечити безперебійну інтеграцію та виконання сценаріїв у всій мережевій інфраструктурі, система повинна мати можливість інтегруватися з численними мережевими пристроями та постачальниками.

2.4.2 Нефункціональні вимоги:

i) Продуктивність: без значного уповільнення система повинна мати можливість виконувати сценарії з досконалістю та керувати великомасштабною мережевою інфраструктурою.

ii) Надійність: щоб переконатися, що сценарії виконуються належним чином і що основні завдання керування мережею виконуються правильно й узгоджено, система має бути надійною.

iii) Зручність використання: щоб спростити мережевим адміністраторам і операторам виконання та створення сценаріїв, система повинна мати зручний інтерфейс або інтерфейс командного рядка.

iv) Супроводжуваність і розширюваність: призначена для забезпечення простого обслуговування, оновлень і включення нових функцій і можливостей, система повинна бути розширюваною і розроблена в модульному форматі.

2.4.3 Вимоги до сумісності:

i) Підтримка мережевих протоколів: щоб зробити зв'язок і взаємодію з різними мережевими пристроями досяжними, система повинна бути оснащена підтримкою популярних мережевих протоколів, таких як SNMP, SSH і NETCONF.

ii) Підтримка постачальника: щоб гарантувати, що сценарії розробляються та запускаються на різноманітному мережевому обладнанні, система має підходити для основних постачальників мережевих пристроїв.

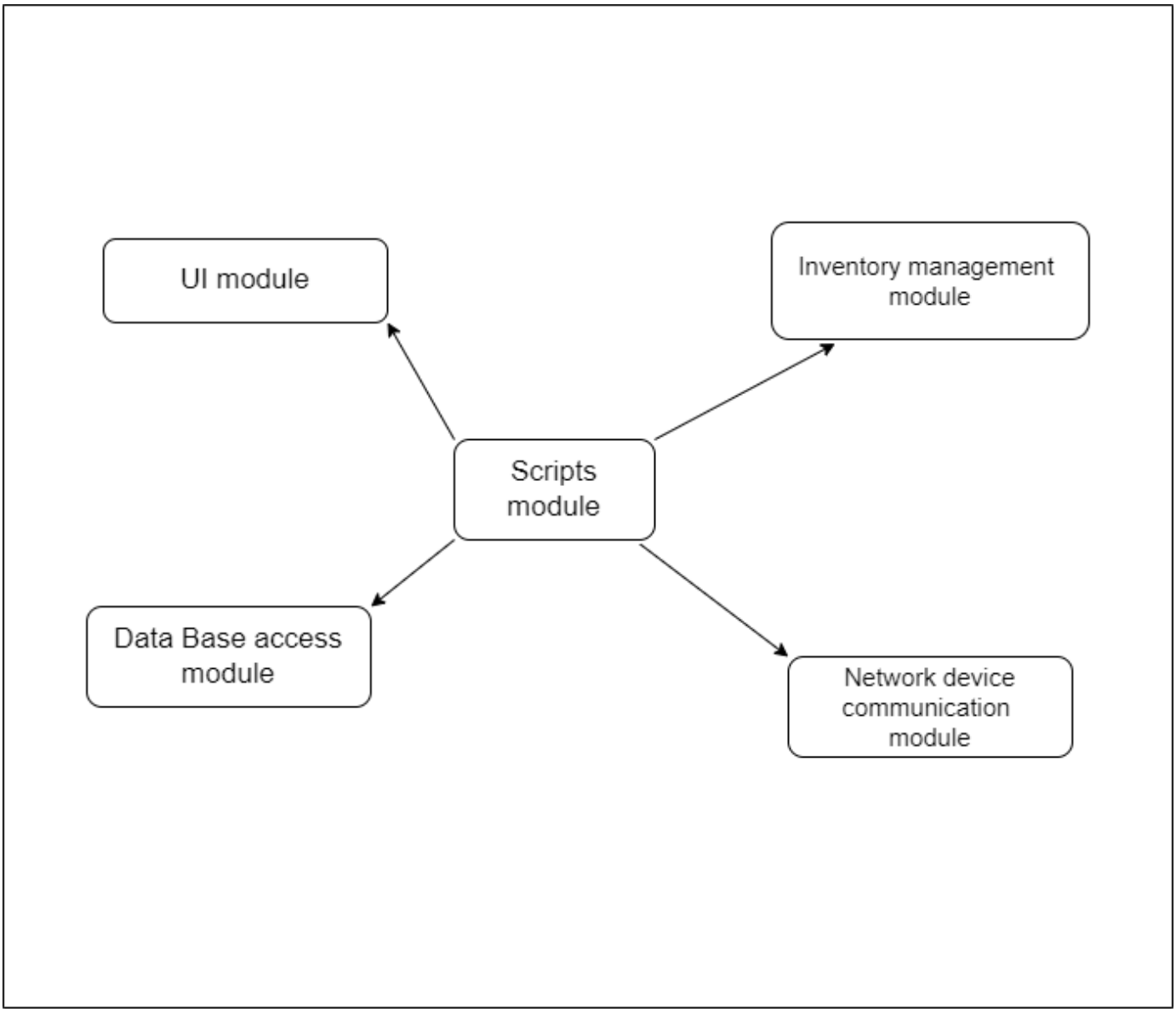
iii) Сумісність з операційною системою: пропонуючи універсальність для розробки та виконання сценаріїв, система повинна бути сумісна з різними операційними системами, включаючи Windows, Linux і macOS.

2.4.4 Вимоги до даних:

i) Формати даних: для зберігання та обробки конфігурацій мережевих пристроїв, даних моніторингу та результатів виконання сценаріїв система повинна забезпечувати підтримку стандартних форматів даних.
ii) Зберігання та отримання даних: для зберігання та відновлення пов’язаних зі сценаріями даних, таких як бібліотеки сценаріїв, шаблони конфігурації та журнали виконання, система повинна забезпечувати надійні механізми зберігання.
iii) Безпека та конфіденційність даних: щоб захистити будь-які конфіденційні дані, що зберігаються або передаються, система повинна впровадити відповідні заходи безпеки.

2.5. Опис архітектури інформаційної системи

Система буде побудована з використанням модульної архітектури *(мал. 4)*, де кожен скрипт представляє окремий модуль, відповідальний за конкретне завдання управління мережею. Кожен модуль, зібраний з окремих сценаріїв, відповідає за окреме завдання, головними цілями якого є гнучкість, можливість повторного використання та належне обслуговування сценаріїв. Щоб досягти цього, кожен модуль можна незалежно розробляти та тестувати, сприяючи модульному середовищу коду та зменшуючи збої, спричинені змінами або оновленнями сценарію.



мал. 4 — приклад модульної архітектури

Крім того, координація виконання цих окремих сценаріїв потребує впровадження механізму виконання сценаріїв – основного компонента цієї архітектури. Діючи як центральний драйвер, механізм виконання отримує вхідні параметри, починає активацію сценарію та керує потоком даних між сценаріями та мережевими пристроями. Таким чином, він забезпечує уніфікований інтерфейс для виконання та контролю керування мережевою інфраструктурою.

Щоб забезпечити взаємодію та інтеграцію з різноманітними мережевими постачальниками, бібліотеки або фреймворки, які сприяють комунікаційним протоколам, таким як SNMP, SSH або NETCONF, вбудовані в архітектуру. Ці бібліотеки служать для забезпечення безперебійного інтерфейсу з мережевими пристроями, стимулюючи виконання різноманітних завдань керування та отримання даних з інфраструктури.

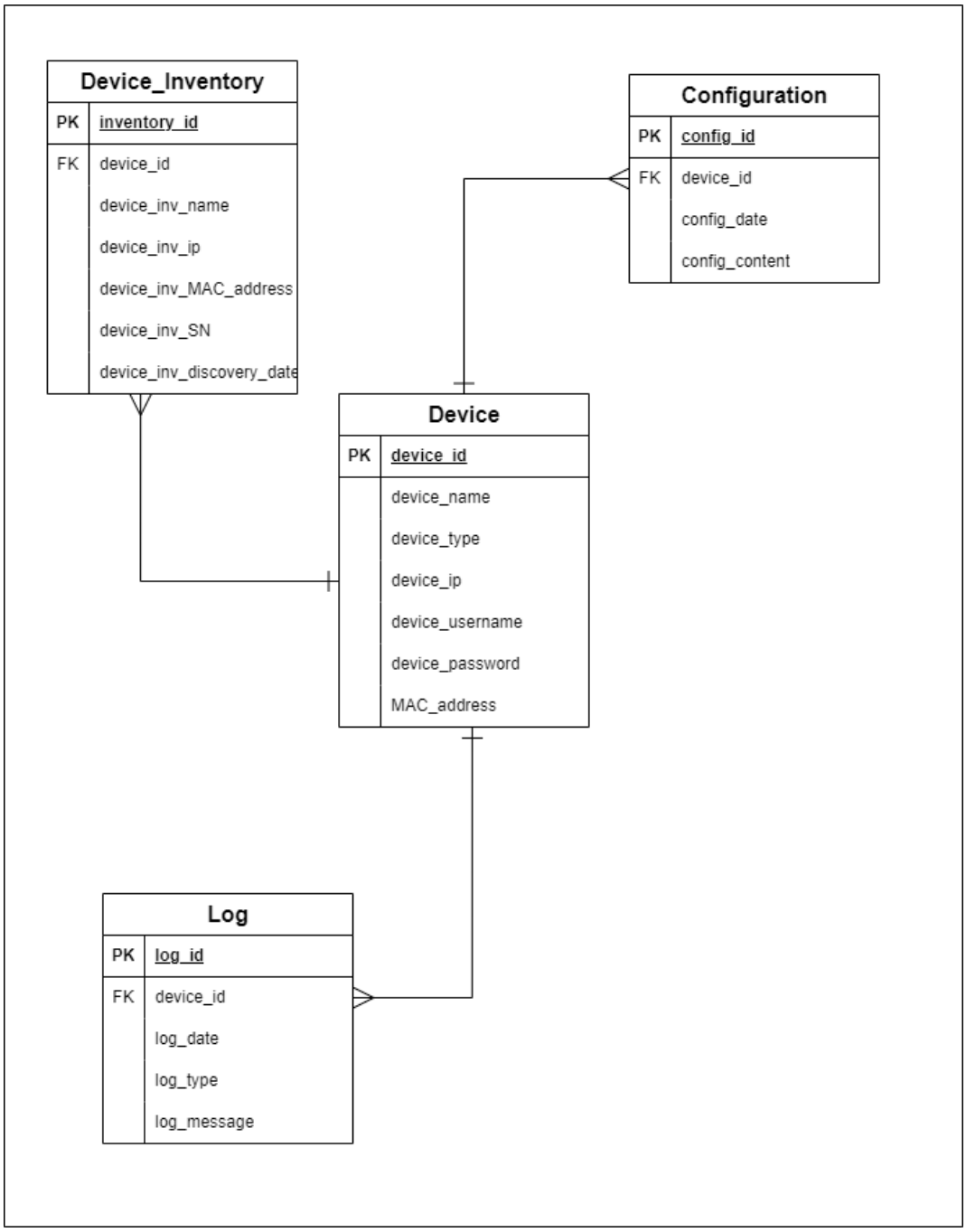
Архітектура полегшує інтеграцію з існуючими компонентами мережевої інфраструктури організації, такими як маршрутизатори, комутатори, брандмауери та сервери, за рахунок використання існуючих API, інтерфейсів CLI або інструментів сторонніх розробників, наданих мережевими пристроями та постачальниками. Це дозволяє сценаріям взаємодіяти з мережевими елементами та з легкістю виконувати бажані завдання керування.

Масштабованість і продуктивність є ключовими міркуваннями при проектуванні архітектури. Система повинна бути здатна адаптуватися до розширення та майбутнього зростання, включаючи підвищені вимоги до управління цифровою мережею. Він повинен мати можливість працювати з великою кількістю мережевих пристроїв, виконувати завдання з вищою швидкістю та ефективно керувати конфігураціями та мережевими даними. Модульна архітектура забезпечує надійну основу для розробки та керування сценаріями, забезпечуючи масштабованість та ефективне керування мережею.

2.6.Проектування бази даних

Дизайн бази даних для системи включає чотири основні таблиці: Device, Configuration, Log та Device_Inventory. Упорядковуючи дані в ці таблиці, система може ефективно зберігати, отримувати та керувати інформацією, пов’язаною з

мережевими пристроями, конфігураціями та записами журналу, сприяючи ефективному управлінню мережевою інфраструктурою.



мал. 5 — ER діаграма системи "Розробка скриптів для системи управління мережевою інфраструктурою організації"

Нижче наведено детальне пояснення кожного поля для кожної таблиці в схемі бази даних. Це забезпечить чітке розуміння мети та значення кожного поля по відношенню до відповідної таблиці.

У таблиці “Device” *(Таблиця 1)* зберігається інформація про мережеві пристрої, включаючи їх унікальні ідентифікатори, ім’я, тип, IP-адресу та облікові дані аутентифікації. Ця таблиця дозволяє керувати та отримувати інформацію про пристрій.

Таблиця 1 — Описи полів таблиці “Device”

Назва поля	**Тип даних**	**Опис**
device_id	int	унікальний ідентифікатор пристрою. (Primary key)
device_name	Varchar (50)	ім’я або мітка, призначена пристрою.
device_type	Varchar (50)	тип або категорія пристрою.
device_ip	Varchar (15)	IP-адреса пристрою.
device_username	Varchar (50)	ім’я користувача, яке використовується для аутентифікації за допомогою пристрою.
device_password	Varchar (50)	пароль, який використовується для автентифікації на пристрої.
MAC_address	Varchar (17)	MAC-адреса пристрою.

Таблиця “Configuration” *(Таблиця 2)* використовується для зберігання конфігурацій мережевих пристроїв. Кожен запис конфігурації пов’язується з пристроєм через посилання на зовнішній ключ. Він містить поля для ідентифікатора конфігурації, ідентифікатора пристрою, дати й часу зміни та фактичного вмісту конфігурації.

Таблиця 2 — Описи полів таблиці “Configuration”

Назва поля	Тип даних	Опис
config_id	Int	унікальний ідентифікатор для конфігурації. (Primary key)
device_id	Int	ідентифікатор пристрою, пов’язаний із конфігурацією. (Foreign key)
config_date	Datetime	дата й час, коли конфігурація була записана або змінена.
config_content	Text	фактичний вміст конфігурації пристрою.

Таблиця “Log” *(Таблиця 3)* веде журнал подій і дій, пов’язаних із мережевими пристроями. Він містить поля для ідентифікатора журналу, ідентифікатора пристрою, дати й часу запису журналу, типу журналу (наприклад, помилка, попередження або інформаційний) і повідомлення журналу, що містить детальну інформацію про подію.

Таблиця 3 — Описи полів таблиці “Log”

Назва поля	Тип даних	Опис
log_id	Int	унікальний ідентифікатор для запису журналу. (Primary key)
device_id	Int	ідентифікатор пристрою, пов’язаний із записом журналу. (Foreign key)
log_date	Datetime	Дата й час створення запису журналу.
log_type	Varchar (20)	тип або категорія запису журналу (наприклад, помилка, попередження, інформаційний).

log_message	Text	докладне повідомлення або опис запису журналу.

У таблиці Device_Inventory *(Таблиця 4)* зберігається інформація про інвентаризацію пристроїв у системі керування мережевою інфраструктурою. Кожен запис інвентаризації пов'язується з певним пристроєм через зовнішній ключ device_id.

Таблиця 4 — Описи полів таблиці "Device_Inventory"

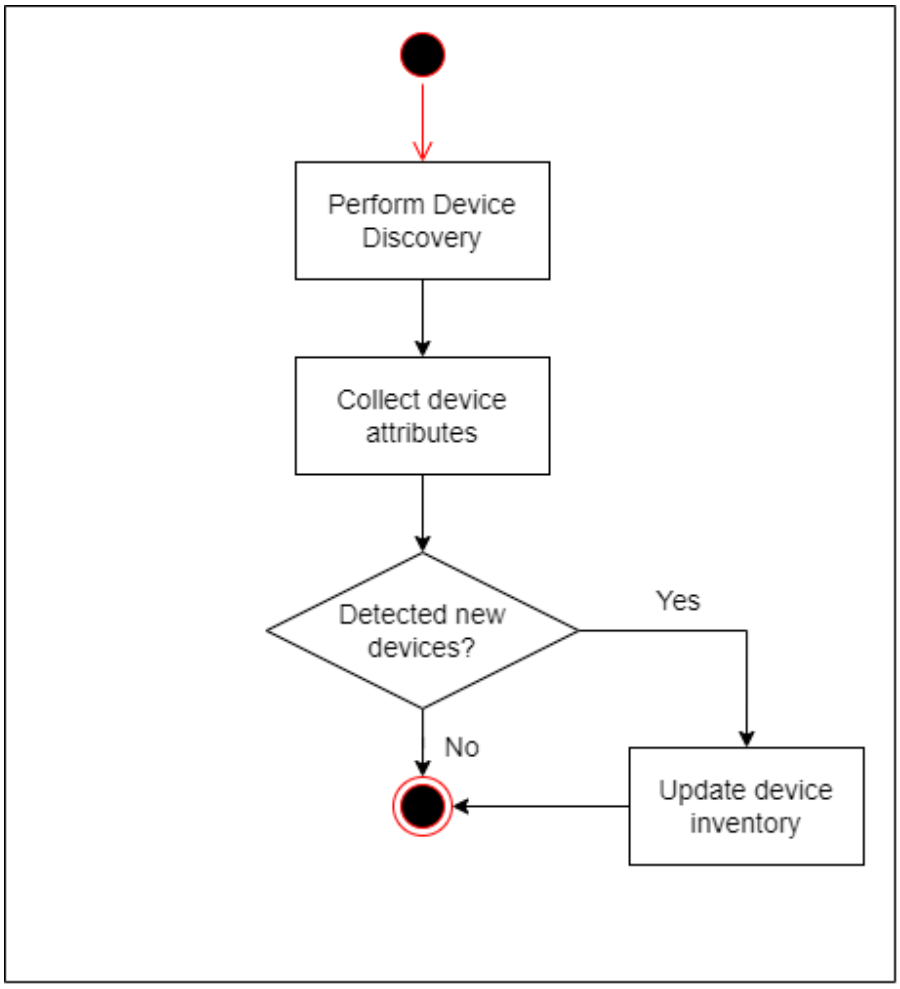
Назва поля	**Тип даних**	**Опис**
inventory_id	Int	унікальний ідентифікатор для запису інвентаризації. (Primary key)
device_id	Int	посилання на пристрій, якому належить цей запис інвентаризації. (Foreign key)
device_inv_name	Varchar (100)	назва або мітка, призначена пристрою для цілей інвентаризації.
device_inv_ip	Varchar (15)	IP-адреса, призначена пристрою.
device_inv_MAC_address	Varchar (17)	MAC-адреса пристрою.
device_inv_discover_date	Datetime	дата, коли пристрій було виявлено або додано до інвентарю.
device_inv_SN	Varchar (100)	серійний номер пристрою.

2.7.Опис та обґрунтування алгоритмів

З метою ефективного управління мережевою інфраструктурою організації розробка скриптів передбачатиме реалізацію різноманітних алгоритмів. Ці алгоритми

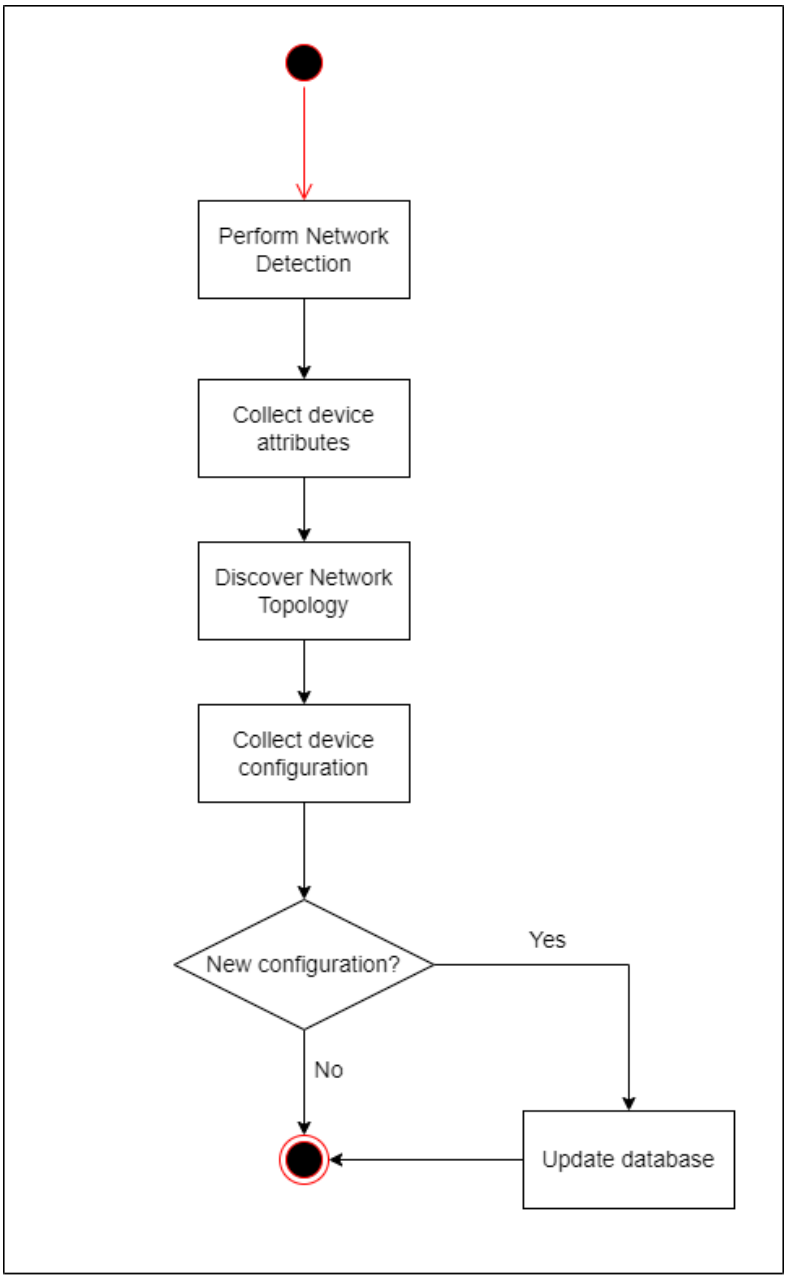
автоматизують ключові завдання керування мережею та покращать загальну ефективність.

- Алгоритм інвентаризації пристроїв (*мал. 6*) виконує автоматичне виявлення всіх мережевих пристроїв, підключених до інфраструктури, і збирає відповідні атрибути, такі як тип пристрою, IP-адреса, MAC-адреса та відомості про постачальника. Надаючи комплексне уявлення про мережеві пристрої та допомагає в управлінні та відстеженні мережевої інфраструктури. Це забезпечує актуальну інвентаризацію та дозволяє точну ідентифікацію та категоризацію пристроїв.



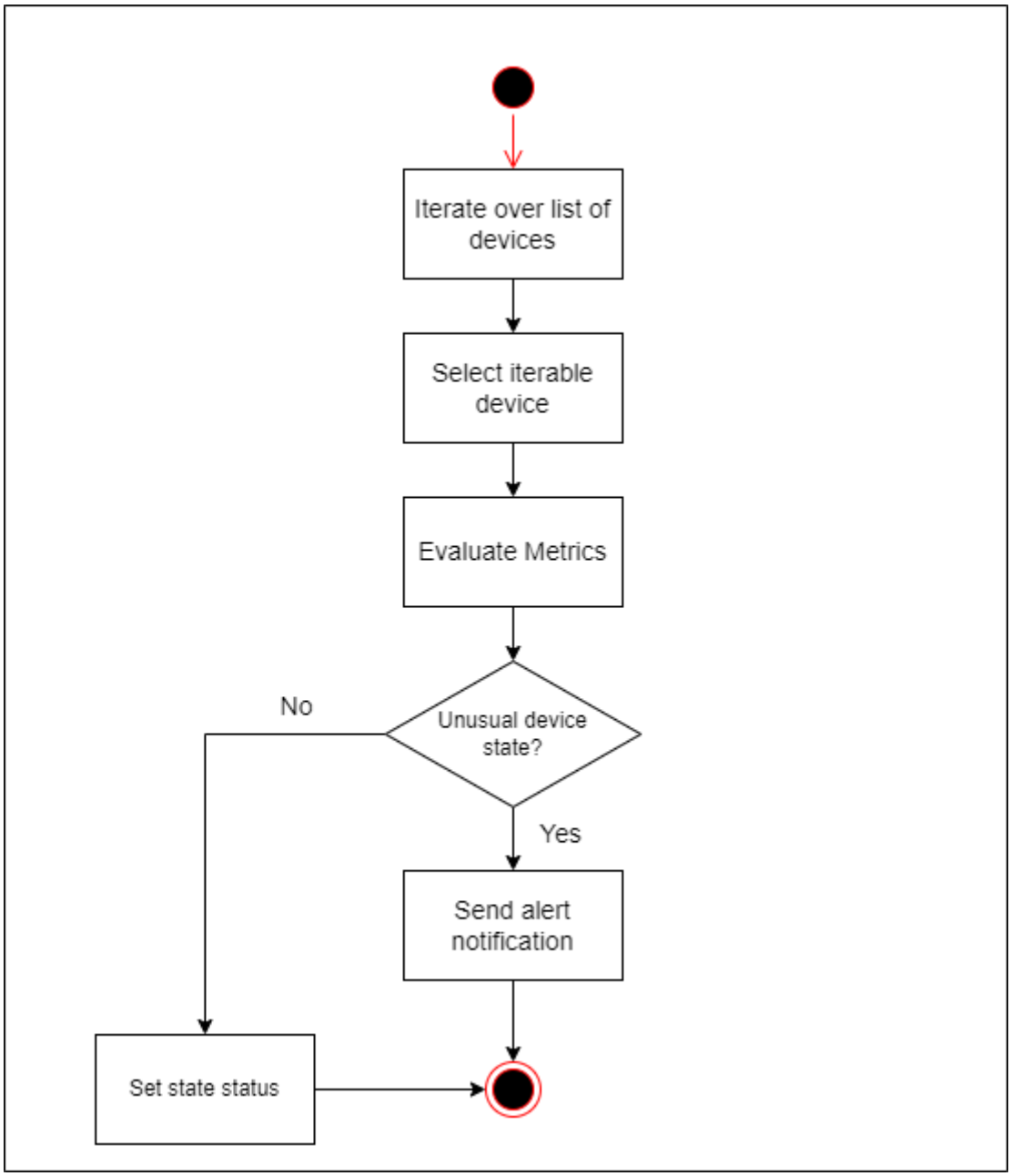
мал. 6 — діаграма послідовності алгоритму виявлення мережі

- Алгоритм виявлення мережі (*мал. 7*) виконує автоматичне виявлення мережевих пристроїв, їхніх атрибутів і зв'язків у мережевій інфраструктурі. Створюючи візуальне представлення топології мережі та збираючи конфігурації пристроїв, цей алгоритм дозволяє покращити видимість мережі, допомагаючи у вирішенні несправностей та ефективному моніторингу мережі та управлінні змінами.



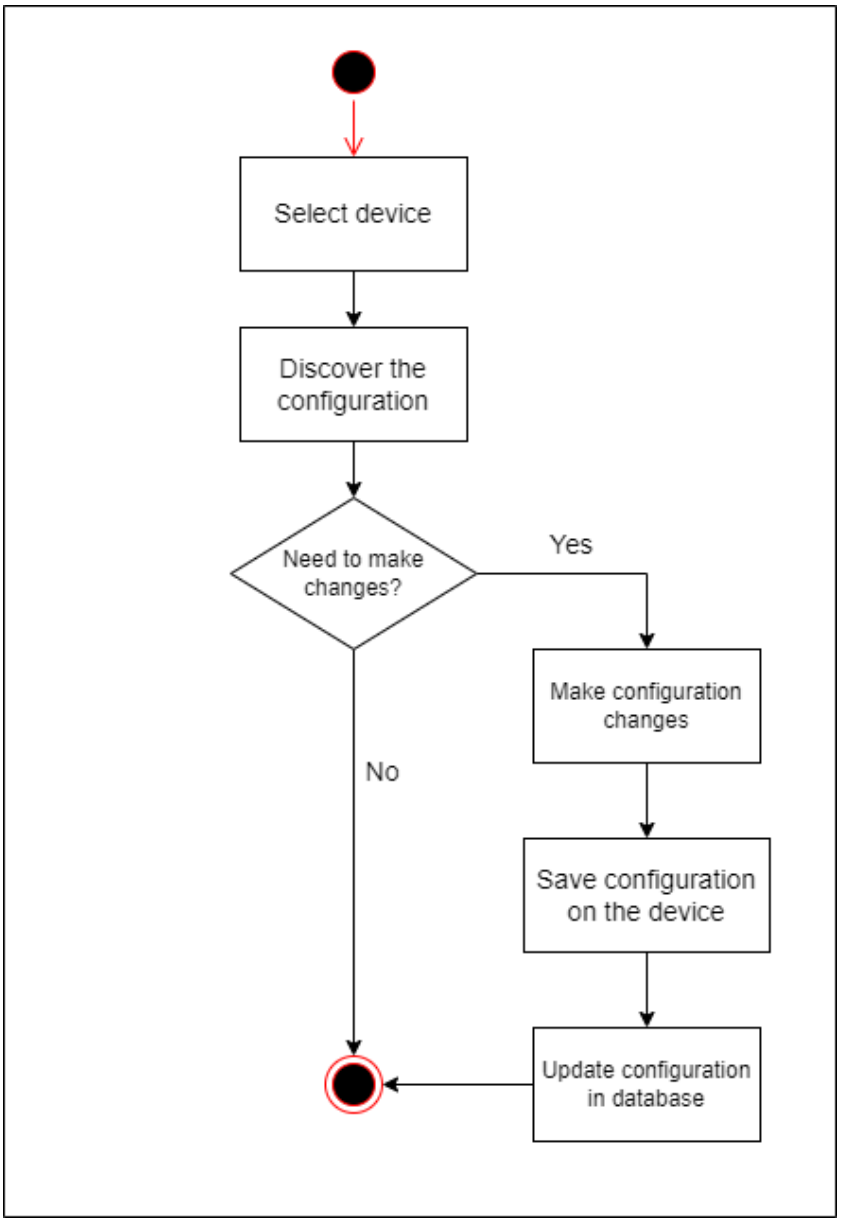
мал. 7 — діаграма послідовності алгоритму виявлення мережі

- Алгоритм моніторингу пристроїв (*мал. 8*) постійно відстежує доступність, показники продуктивності та стан працездатності мережевих пристроїв, відстежуючи стан пристроїв і генеруючи сповіщення або попередження за потреби. Спрощуючи проактивний моніторинг пристрою, цей алгоритм дозволяє своєчасно виявляти та вирішувати проблеми з мережею.



мал. 8 — діаграма послідовності алгоритму моніторингу пристроїв

- Алгоритм керування конфігурацією пристрою (*мал. 9*) збирає, зберігає та керує деталями конфігурації мережевих пристроїв, водночас забезпечуючи ефективне розгортання конфігурацій. Цей алгоритм забезпечує послідовність, точність і відповідність мережевим політикам, спрощуючи завдання керування конфігурацією та мінімізуючи людські помилки.



мал. 9 — діаграма послідовності алгоритму керування конфігурацією пристрою

Загалом, впровадження цих алгоритмів має вирішальне значення для ефективного управління мережевою інфраструктурою організації. Використовуючи ці алгоритми, сценарії можуть надавати ефективні можливості керування мережею, сприяючи створенню більш ефективної та надійної мережевої інфраструктури.

2.8. Використані технології та обґрунтування вибраного інструментарію

Розробка сценаріїв для керування мережевою інфраструктурою спирається на вибір технологій та інструментів для забезпечення ефективного та ефективного впровадження. Нижче наведено основні технології та інструменти, які використовуються в цьому проекті, а також їх обґрунтування:

Мова програмування: Python

Python — це універсальна та широко поширена мова, відома своєю простотою, читабельністю та широкою підтримкою бібліотек. Він пропонує надійні мережеві можливості, що робить його придатним для автоматизації мережевих завдань і взаємодії з мережевими пристроями.

Система управління базами даних: MySQL

MySQL — популярна та надійна система керування реляційними базами даних із відкритим кодом. Вона надає масштабоване та ефективне рішення для зберігання даних для керування даними, пов’язаними з мережевими пристроями, конфігураціями та інвентарем.

Бібліотеки та фреймворки:

- Paramiko: ми обрали Paramiko, оскільки це широко використовувана бібліотека Python для з’єднання SSH і безпечного зв’язку з мережевими пристроями. Він забезпечує міцний і надійний метод встановлення з’єднань

SSH, що є важливим для безпечного доступу до мережевих пристроїв під час виконання завдань автоматизації.

- Netmiko — це ще одна бібліотека Python, яка спрощує керування та налаштування мережевих пристроїв від різних постачальників. Він пропонує уніфікований інтерфейс і усуває складність взаємодії з різними типами та моделями пристроїв. Ця підтримка між постачальниками зробила його ідеальним вибором для нашого проекту, оскільки нам потрібно було працювати з пристроями від багатьох постачальників.
- NAPALM — це потужна бібліотека Python, яка забезпечує нейтральний для постачальника рівень абстракції для автоматизації мережі та керування конфігурацією. Це дозволяє нам взаємодіяти з мережевими пристроями за допомогою узгодженого набору методів і функцій, незалежно від реалізацій, що лежать в основі конкретного постачальника. Ця гнучкість і підхід незалежно від постачальника були ключовими факторами при виборі NAPALM для нашого проекту.
- Requests: ми включили бібліотеку requests у наш набір інструментів, щоб полегшити взаємодію з API мережевих пристроїв. Запити спрощують процес надсилання HTTP-запитів і обробки відповідей, дозволяючи нам отримувати дані з мережевих пристроїв, змінювати конфігурації та виконувати інші дії, керовані API. Ця універсальність і простота використання зробили його придатним вибором для інтеграції з функціями API мережевих пристроїв. Вибір Python як мови програмування забезпечує гнучкість і потужність, необхідні для завдань автоматизації мережі. Він підтримується багатою екосистемою бібліотек і фреймворків, спеціально розроблених для керування мережею. Обрана система керування базою даних MySQL забезпечує ефективне зберігання та пошук пов’язаних з мережею даних.

Бібліотеки та фреймворки, такі як Paramiko, Netmiko, NAPALM і Requests, розширюють можливості сценарію, забезпечуючи надійне з’єднання SSH, підтримку різних постачальників, керування конфігурацією та взаємодію з API мережевих пристроїв.

Використовуючи ці технології та інструменти, сценарії для керування мережевою інфраструктурою можуть автоматизувати завдання, оптимізувати налаштування та підвищити загальну ефективність мережі.

2.9.Результати впровадження

Фаза впровадження системи управління мережевою інфраструктурою організації вимагала ретельного виконання кроків для розробки та розгортання необхідних сценаріїв. Python було обрано як основну мову сценаріїв для цієї мети завдяки його універсальності, підтримці великої бібліотеки та простому використанню. Щоб забезпечити автоматизацію та керування мережею, для досягнення бажаних результатів використовувалися різні бібліотеки та фреймворки. Сценарії, включені в систему, мали можливість виконувати безліч завдань, від отримання та зберігання конфігураційної інформації до формування списку пристроїв.

Після виконання Paramiko використовувався для встановлення з’єднань SSH із пристроями та аутентифікації користувача перед виконанням команд або збором необхідних даних. Netmiko використовувався для резервного копіювання та відновлення конфігурацій, тоді як NAPALM дозволяв маніпулювати та отримувати конфігурації пристроїв і збережені дані. Виявлення мережевого пристрою було досягнуто за допомогою сканованих діапазонів IP-адрес і SNMP. Зібрані дані згодом зберігалися в деталізованій інвентаризації пристроїв, що дозволяло адміністраторам швидко отримувати доступ до IP-адрес пристроїв, пов’язаних MAC-адрес і дат

виявлення. Завдання варіювалися від налаштування резервного копіювання до перевірки інвентаризації пристрою.

Завдяки поєднанню потужних сценаріїв Python і використання спеціалізованих бібліотек і фреймворків система змогла з виконувати процедури мережевої автоматизації та керування. Захищені з’єднання SSH забезпечували надійний зв’язок із мережевими пристроями, а API, не залежні від постачальника, забезпечували ефективне керування пристроєм і доступ до даних. Таким чином, сценарії змогли підвищити ефективність і полегшити ефективне управління мережевою інфраструктурою організації.

3. ВИСНОВКИ

Ця кваліфікаційна робота присвячена розробці сценаріїв для системи управління мережевою інфраструктурою організації. У ході роботи ми досліджували існуючу літературу та дослідження з управління мережами, мов сценаріїв і відповідних технологій. Вирішуючи ключові проблеми в управлінні мережевою інфраструктурою, ми успішно реалізували різні алгоритми в управлінні мережею.

Впроваджені скрипти продемонстрували свою ефективність в автоматизації критичних завдань і покращенні видимості мережі, керування пристроями, моніторингу та налаштування. Алгоритм інвентаризації пристроїв уможливив автоматичне виявлення та збір інформації про мережеві пристрої, забезпечуючи актуальну інвентаризацію та точну ідентифікацію пристроїв. Алгоритм виявлення мережі покращив видимість мережі, створивши видиме представлення топології мережі та збираючи конфігурації пристроїв. Крім того, алгоритм моніторингу пристроїв сприяв проактивному моніторингу доступності пристрою, показників продуктивності та стану працездатності, дозволяючи своєчасно виявляти та вирішувати проблеми з мережею. Алгоритм управління конфігурацією пристрою оптимізував завдання керування конфігурацією, забезпечуючи послідовність, точність і відповідність мережевим політикам. Для реалізації цих реалізацій ми використовували такі мови програмування, як Python, і різні бібліотеки та інструменти, зокрема, SNMP, SQLAlchemy і MySQL. Ці технології забезпечили міцну основу для надійної функціональності та бездоганної інтеграції з існуючою інфраструктурою. Розроблені сценарії знаходять актуальність і застосовність у широкому діапазоні сценаріїв управління мережею. У традиційних корпоративних мережах чи складних розгортаннях IoT сценарії пропонують цінні можливості автоматизації, спрощують керування пристроями, підвищують безпеку та забезпечують ефективний моніторинг і усунення несправностей.

Підсумовуючи, ця дипломна робота успішно вирішила завдання розробки сценаріїв для системи управління мережевою інфраструктурою організації. Реалізовані алгоритми разом із пов’язаними з ними функціями продемонстрували потенціал автоматизації для покращення видимості мережі, керування пристроями, моніторингу та налаштування. Результати цієї роботи сприяють поповненню сукупності знань з управління мережею та служать цінним ресурсом для мережевих адміністраторів та IT-спеціалістів, які шукають практичні ідеї та рішення для ефективного управління мережевою інфраструктурою.

4. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anderson, T., & Roderick, J. (2018). Network Programmability and Automation: Skills for the Next-Generation Network Engineer. Cisco Press.
2. Braden, R. (2017). Python Network Programming: Conquer all your networking challenges with the powerful Python language. Packt Publishing.
3. Hartley, D. (2019). Practical Network Automation: Leverage the power of Python and Ansible to optimize your network. Packt Publishing.
4. Kim, Y., Edsall, T., & Steenbergen, J. (2016). Junos Automation: A Guide to Automating Junos with DevOps Tools. O'Reilly Media.
5. Morris, G., & Newcomb, B. (2017). Network Management, MIBs and MPLS: Principles, Design and Implementation. Wiley.
6. Odom, W. (2020). Network Programmability and Automation: Skills for the Next-Generation Network Engineer. Cisco Press.
7. Pearce, P. (2019). Automating Junos Administration: Doing More with Less. O'Reilly Media.
8. Rose, G. (2020). Network Automation Cookbook: Proven and actionable recipes to automate and manage network devices using Ansible. Packt Publishing.
9. Sadowski, J., & Smart, B. (2016). Mastering Python Networking: Your one-stop solution to using Python for network automation, DevOps, and Test-Driven Development. Packt Publishing.
10. Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer Networks. Pearson Education.

АНОТАЦІЯ

Основною ціллю цієї дипломної роботи було дослідження розробки скриптів для системи управління мережевою інфраструктурою організації. У ході роботи було розглянуто та проаналізовано існуючі роботи на цю тему, вивчено основні принципи управління мережевими пристроями та розробки автоматизованих скриптів для мережевого управління. Було використано широкий спектр технологій та бібліотек для реалізації системи, зокрема мову програмування Python, бібліотеки NAPALM, SQLAlchemy, інструменти для взаємодії з мережевими пристроями та роботи з базою даних MySQL. У ході дипломної роботи було досліджено різні алгоритми, такі як алгоритм інвентаризації пристроїв, алгоритм виявлення мережі та алгоритм керування конфігурацією пристроїв. Кожен з цих алгоритмів був реалізований за допомогою відповідних скриптів та логіки програми, що забезпечує їх правильну роботу та взаємодію з мережевою інфраструктурою організації. В результаті дипломної роботи були розроблені скрипти для управління мережевою інфраструктурою, яка дозволяє автоматизувати рутинні задачі з управління та моніторингу мережевих пристроїв. Реалізовані скрипти та алгоритми дозволяють здійснювати інвентаризацію пристроїв, виявлення мережі, керування конфігурацією та моніторинг мережевих пристроїв, що покращує ефективність та безпеку управління.