

Лабораторна робота №8

Додаткові налаштування системи моніторингу Nagios: користувачі, часові проміжки, контакти.

Мета: виконати додаткові налаштувати системи моніторингу для розподілення прав між користувачами, визначення часових інтервалів моніторингу, додаткового інформування про деякі критичні події та введення поняття сервісних груп.

Інструменти: гіпервізор VirtualBox, модель комп'ютерної мережі.

Завдання до лабораторної роботи

1. Налаштуйте новий часовий період системи моніторингу таким чином щоб моніторинг відбувався не більш ніж 5 днів на тиждень та 7 годин на день, з періодичністю не частіше 1 раз на 3 години. Змініть налаштування моніторингу на цей період для лінукс-сервера Serv-G-N-3. Зніміть коменти з сервісів моніторингу зовнішніх доменів, що були налаштовані у одній з попередніх робіт.
2. Створіть додаткового, гостьового користувача у системі моніторингу таким чином, щоб він не мав доступу до системних команд, команд управління сервісами та команд управління пристроями. Ім'я користувача – довільне.
3. Налаштуйте отримання поштових повідомлень про критичну подію «Вимкнення серверу контролера домену». У звіт включіть скрін отриманого повідомлення. Одразу після отримання повідомлення, увімкніть вимкнений сервер.
4. Налаштуйте базовий моніторинг мережевого шлюза NAT Network та додайте його як «батьківський» пристрій до конфігурацій хостів. У звіт включіть оновлений вигляд Map (Legacy) сайту Nagios.
5. Створіть кілька довільних груп сервісів. Наприклад «навантаження CPU хостів» та «Доступність доменів». У звіт включіть відповідний скрін.

Звіт має містити:

- лістинг використаних команд;
- скріншоти отриманих результатів моніторингу у Nagios 4;
- короткий опис редагування файлів конфігурації Nagios 4.

Теоретичні відомості

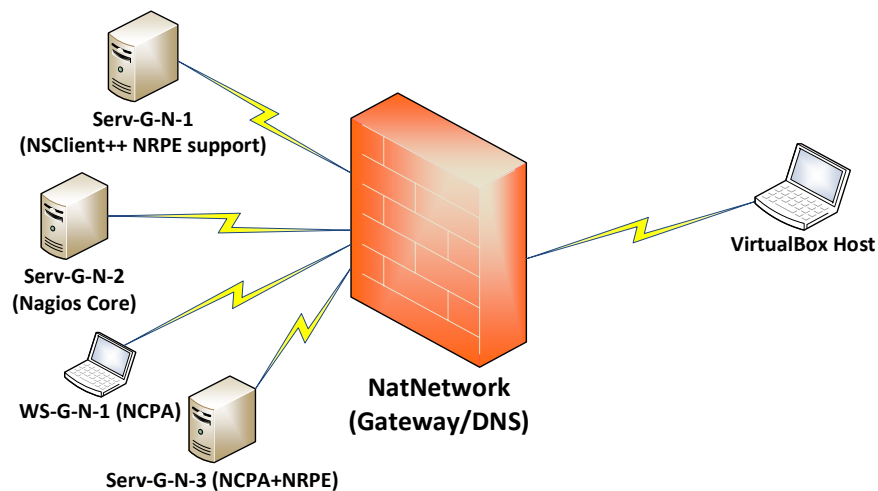


Рис. 8.1. Топологія мережі

На рис.8.1. наведена модель комп'ютерної мережі, побудована під час виконання попередніх лабораторних робіт. До серверу Serv-G-N-2 налаштовано SSH доступ через NAT Network для VirtualBox Host.

На сервері Serv-G-N-2 розгорнуто систему моніторингу на базі Nagios 4.X. Налаштовано підключення з хосту NAT Network по протоколу HTTP до систему моніторингу під користувачем nagios.

Time Periods in Nagios Core.

Налаштування часових інтервалів моніторингу дозволяє контролювати, коли можуть працювати різні аспекти логіки моніторингу та оповіщення. Наприклад, можна обмежити виконання регулярних запланованих перевірок хоста та служби, надсилати сповіщення, використовувати ескалацію сповіщень, коли діють залежності.

Визначення хостів і служб мають необов'язкову директиву `check_period`, яка дозволяє вказати період часу, який слід використовувати для обмеження, коли можна виконувати регулярні активні перевірки хоста чи служби. Якщо директиву `check_period` не використовується для визначення періоду часу, Nagios Core зможе запланувати активні перевірки хоста чи служби в будь-який час. По суті, це сценарій моніторингу 24x7.

Зазначення періоду часу в директиві `check_period` дозволяє обмежити час, протягом якого Nagios Core виконує регулярні заплановані активні перевірки хоста або служби. Коли Nagios Core намагається перепланувати перевірку хоста або служби, він переконається, що наступна перевірка потрапляє в дійсний діапазон часу в межах визначеного періоду часу. Якщо цього не відбувається, Nagios Core налаштує час наступної перевірки так, щоб він збігся з наступним «дійсним» часом у вказаний період часу. Це означає, що хост або служба можуть не перевірятися знову через годину, день чи тиждень тощо.

Приклади директив для різних періодів часу можна знайти [тут](#). Ми будемо тестово-навчальну систему моніторингу, тому налаштуємо два часових проміжки моніторингу. Ці налаштування виконуються у файлі `/usr/local/nagios/etc/objects/timeperiods.cfg`. Відкриваємо файл і одразу бачимо кілька вже визначених часових періодів та шаблонів до них. Наприклад стандарт моніторингу 24x7 виглядає наступним чином:

```
define timeperiod {  
  
    name                24x7  
    timeperiod_name     24x7  
    alias               24 Hours A Day, 7 Days A Week  
  
    sunday              00:00-24:00  
    monday              00:00-24:00  
    tuesday             00:00-24:00  
    wednesday           00:00-24:00  
    thursday            00:00-24:00  
    friday              00:00-24:00  
    saturday            00:00-24:00  
  
}
```

Створимо свій перший часовий період, що має працювати у режимі 24x7 з вимкненням моніторингу кожну неділю з 22:00 до 23:00 по Київському часу. Скажімо, що у цей час має виконуватися сервісне перезавантаження серверів. З огляду на [приклад](#) конфігурація такого часового періоду буде мати вигляд:

```
# 24/7 with downtime every Sunday from 22:00 to 23:00 EET  
define timeperiod {  
    timeperiod_name     24x7-no-sunday-night  
    alias               24/7 (except Sunday night)  
    sunday              00:00-23:00  
    monday              00:00-24:00  
    tuesday             00:00-24:00  
    wednesday           00:00-24:00  
    thursday            00:00-24:00  
    friday              00:00-24:00  
    saturday            00:00-24:00  
  
}
```

Другий часовий період, протягом якого виконується моніторинг робочих станцій та їх периферії. Період має тривати лише робочий час персоналу - з понеділка по четвер – з 9:00 до 18:00, у п'ятницю – з 9:00 до 17:00 по Київському часу. Конфігурація такого часового періоду має наступний вигляд:

```
# Work hours (Mon-Fri 9:00-18:00, Fri 9:00-17:00)
define timeperiod {
    timeperiod_name    work-hours
    alias              Work hours
    monday             09:00-18:00
    tuesday            09:00-18:00
    wednesday          09:00-18:00
    thursday           09:00-18:00
    friday             09:00-17:00
    saturday           00:00-00:00
    sunday             00:00-00:00
}

```

Додаємо часові періоди до конфігураційного файлу `/usr/local/nagios/etc/objects/timeperiods.cfg` та виконуємо перевірку вірності внесених у конфігурацію змін з перезапуском сервісу Nagios:

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
sudo service nagios restart
```

Під час виконання попередньої лабораторної ми виконали для лінукс сервера Serv-G-N-3 налаштування перевірки доступності трьох доменних імен `ztu.edu.ua` та тимчасово вимкнули їх моніторинг щоб запобігти можливому ефекту DDoS-атак через часті звернення. Редагуємо секції згаданих сервісів у файлі `/usr/local/nagios/etc/objects/linux/serv-22-1-3.cfg`. Знімаємо коментарі з рядків, параметр `check_interval` збільшуємо з 5 хв до 30, у якості періоду перевірок `check_period` замість стандартного значення 24x7 задаємо назву визначеного робочого періоду `work-hours`, так само змінюємо `notification_period`, що визначає періоди, коли Nagios буде відправляти повідомлення про недоступність сервісу. `notification_period` може відрізнятися від `check_period`.

```
define service {
    host_name          serv-22-1-3
    use                generic-service
    service_description    Check domain ztu.edu.ua
    check_command      check_nrpe!check_dns -a ztu.edu.ua
    max_check_attempts    5
    check_interval      30
    retry_interval      1
    check_period        work-hours
    notification_interval    60
    notification_period  work-hours
}

```

Для всіх інших сервісів даного серверу змінюємо `check_period` та `notification_period` на назву заданого нами часового періоду `24x7-no-sunday-night`.

Змінюємо часові налаштування моніторингу для робочої станції. Для цього редагуємо її конфігураційний файл `/usr/local/nagios/etc/objects/workstation/ws-22-1-1.cfg`, змінюючи значення `check_period` та `notification_period` з 24x7 на `work-hours`, а параметр `check_interval` збільшуємо з 5 хв до 15. Приклад змін:

```
define host {
    host_name          ws-22-1-1
    address            ws-22-1-1.falkovsky.net
    hostgroups         win-workstations
    check_command      check_ncpa!-t 'P@ssw0rd2023' -P 5693 -M system/agent_version
    max_check_attempts    5
    check_interval      15
    retry_interval      1
    check_period        work-hours
    notification_interval    60
    notification_period  work-hours
    notifications_enabled    1
}
define service {
    host_name          ws-22-1-1
    use                generic-service
    service_description    CPU Usage
    check_command      check_ncpa!-t 'P@ssw0rd2023' -P 5693 -M cpu/percent -w 20 -c 40 -q
'aggregate=avg'
}

```

```

max_check_attempts      5
check_interval          15
retry_interval          1
check_period            work-hours
notification_interval    60
notification_period     work-hours
}

```

Виконуємо перевірку вірності внесених у конфігурацію змін з перезапуском сервісу Nagios:

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
sudo service nagios restart
```

Користувачі Nagios Core.

Розширені можливості по створенню та адмініструванню користувачів та їх прав доступу у Nagios Core надає безкоштовний пакет NagiosQL3 (Nagios Web UI). Додаткові можливості пакету:

- Надає зручний інтерфейс для створення, редагування та видалення конфігураційних об'єктів Nagios, таких як хостинги, сервіси, команди, контакти і т.д.
- Забезпечує інтеграцію з конфігураційними файлами Nagios та виконанням команд через Nagios.
- Надає зручний інтерфейс для перегляду статистики моніторингу та створення звітів.
- Використовує базу даних для зберігання конфігурації, журналів та іншої інформації.
- Дозволяє адміністраторам керувати системою моніторингу безпосередньо через веб-інтерфейс, що робить процес керування більш зручним та доступним.

У додатку 1 описане встановлення та конфігурування актуальної версії Nagios Web UI, але це трудомісткий процес і він не входить до плану виконання цієї лабораторної роботи.

У конфігураційному файлі /usr/local/nagios/etc/cgi.cfg є кілька глобальних змінних для налаштування прав користувачів:

Таблиця 8.1

authorized_for_system_information	SYSTEM/PROCESS INFORMATION ACCESS	Користувачі, які мають доступ до перегляду інформації про процес Nagios.
authorized_for_configuration_information	CONFIGURATION INFORMATION ACCESS	Користувачі, які можуть переглядати ВСЮ конфігураційну інформацію (хости, команди тощо). За замовчуванням користувачі можуть переглядати лише конфігураційну інформацію для хостів і служб, для яких вони є контактами.
authorized_for_system_commands	SYSTEM/PROCESS COMMAND ACCESS	Користувачі, які можуть видавати команди завершення роботи та перезапуску Nagios за допомогою команди CGI (cmd.cgi) і можуть змінювати режим програми на активний або очікування.
authorized_for_all_services authorized_for_all_hosts	GLOBAL HOST/SERVICE VIEW ACCESS	Користувачі, які можуть переглядати інформацію для всіх хостів і служб, які контролюються. За замовчуванням користувачі можуть переглядати лише інформацію про хости чи служби, для яких вони є контактами.
authorized_for_all_service_commands authorized_for_all_host_commands	GLOBAL HOST/SERVICE COMMAND ACCESS	Користувачі, які можуть видавати команди, пов'язані з хостом або службою, за допомогою команди CGI (cmd.cgi) для всіх хостів і служб, які контролюються. За замовчуванням користувачі можуть видавати команди лише для хостів або служб, для яких вони є контактами.
authorized_for_read_only	READ-ONLY USERS	Користувачі, які мають права лише на читання в CGI. Для них блоковані будь-які служби, коментарі чи команди хосту, які зазвичай відображаються на сторінках extinfo CGI.

Стандартна задача. Необхідно надати обмежений доступ виконавцю, який буде лише переглядати статуси хостів та сервісів і отримувати на пошту повідомлення про певні хости. Створюємо користувача системи моніторингу surname:

```
sudo htpasswd /usr/local/nagios/etc/htpasswd.users surname
```

Додаємо користувача surname до файлу контактів Nagios /usr/local/nagios/etc/objects/contacts.cfg

```

define contact {
    contact_name      surname
    alias             Nagios User
    email             surname@gmail.com
    service_notification_period 24x7
}

```

```
host_notification_period 24x7
service_notification_options w,u,c,r
host_notification_options d,r
service_notification_commands notify-service-by-email
host_notification_commands notify-host-by-email
}
```

У конфігураційному файлі `/usr/local/nagios/etc/cgi.cfg` налаштуємо наступні глобальні змінні

```
authorized_for_read_only=surname
authorized_for_all_services=nagios,surname
authorized_for_all_hosts= nagios,surname
```

Виконуємо перевірку вірності внесених у конфігурацію змін. Перезапускаємо сервіси Nagios та Apache:

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo service nagios restart
sudo systemctl restart apache2
```

Налаштування відправки поштових повідомлень.

У файлах конфігурації Nagios є налаштування відсилання повідомлень у випадку проблем з хостами або сервісами. Виконаємо додаткові налаштування для повноцінної роботи цього функціоналу.

Вище описано як у файлі `/usr/local/nagios/etc/objects/contacts.cfg` визначаються контакти за замовчуванням та інші. Відповідно для кожного хосту або навіть сервісу можливо визначити свій контакт.

Визначення контакту для локального хосту у файлі `/usr/local/nagios/etc/objects/linux/localhost.cfg` виконується додаванням рядку `contacts` у відповідну секцію

```
define host {
    hostgroups          linux-servers
    use                 linux-server
    host_name           serv-22-1-2
    alias               Serv-22-1-2
    address             127.0.0.1
    contacts            surname
}
```

Визначення контакту `surname` виконано у файлі `/usr/local/nagios/etc/objects/contacts.cfg` де в полі `email` прописується адреса одержувача, наприклад, surname@ukr.net або surname@ztu.edu.ua

В описі контакту є поля, в яких описано, коли посилати повідомлення про стан хоста і сервісу:

```
define contact {
    contact_name        igor
    alias               Nagios User
    email               surname@ztu.edu.ua
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,r
    service_notification_commands notify-service-by-email
    host_notification_commands notify-host-by-email
}
```

Пояснення параметрів:

contact_name: Це ім'я контакту, яке ідентифікує його в системі.

alias: Це псевдонім контакту, який може бути використаний для відображення в інтерфейсі Nagios.

email: Адреса електронної пошти, на яку будуть надсилатися повідомлення.

service_notification_period: Період, коли служби будуть сповіщати про стан.

host_notification_period: Період, коли будуть сповіщати про стан хоста.

service_notification_commands: Команда для сповіщень про стан служби.

host_notification_commands: Команда для сповіщень про стан хоста.

<p>host_notification_options значення за замовчуванням «d, r»</p> <p>Можливі варіанти значень:</p> <ul style="list-style-type: none"> • d — повідомляти про статус DOWN • u — повідомляти про статус UNREACHABLE • r — повідомляти про підняття хоста (перший UP) • f — повідомляти про початок і закінчення «блимаючого» стану • s — повідомляти про початок і закінчення запланованого вимкнення • n — не відсилати жодних повідомлень 	<p>service_notification_options значення за замовчуванням «w,u,c,r»</p> <p>Можливі варіанти значень:</p> <ul style="list-style-type: none"> • w — повідомляти про статус WARNING service states, • u — повідомляти про статус UNKNOWN service states, • c — повідомляти про статус CRITICAL service states, • r — повідомляти про підняття сервісу (перша поява статусу OK) • f — повідомляти про початок і закінчення «блимаючого» стану • n — не відсилати жодних повідомлень
--	---

sendmail, що використовується для відсилання повідомлень у налаштуваннях за замовчуванням, дещо обмежений у можливостях. Це не завжди зручно. Змінюємо конфігурацію і скористаємося зовнішнім сервером.

Налаштування поштового сервера

Налаштуємо локальний поштовий сервер для відправки повідомлень. Використовуємо у якості поштового пакету Postfix. Команди встановлення Postfix:

sudo apt update

sudo apt install postfix

Під час встановлення вас буде запитано про тип конфігурації. Оберіть "Internet Site" і натисніть Enter. У наступному вікні введіть повністю кваліфікований доменне ім'я вашого сервера serv-G-N-2.surname.net. Після встановлення налаштуємо Postfix, відредагувавши його конфігураційний файл:

sudo nano /etc/postfix/main.cf

У цьому файлі налаштуємо різні параметри, такі як дозвіл відправки електронної пошти від імені вашого домену, обмеження розміру повідомлення та інші. Наприклад, для дозволу відправлення електронної пошти від імені домену додайте наступний рядок:

```
myorigin = nagios.local
```

Редагуємо рядок, що містить IP-адресу хоста, додаючи його повністю кваліфіковане ім'я у файлі /etc/hosts, який використовується для локального розрішення доменних імен на IP-адреси без використання DNS (Domain Name System).

```
127.0.0.1 localhost serv-G-N-2.surname.net serv-G-N-2
```

Подібні налаштування виконуємо у файлі /etc/hostname, що містить ім'я хосту (вузла) в системі

```
serv-G-N-2.surname.net
```

Виконуємо ці ж зміни для поточного сеансу командою:

sudo hostnamectl set-hostname serv-G-N-2.surname.net

Після внесення змін у конфігураційні файли, перезапускаємо службу Postfix, щоб вони набрали чинності:

sudo systemctl restart postfix

Встановлюємо пакет відправки поштових повідомлень mailutils

sudo apt install mailutils

Перевірка працездатності Postfix виконується надсиланням тестового повідомлення з командного рядка:

echo "Тестове повідомлення" | mail -s "Тестова тема" адреса_отримувача

Після відправлення перевірте свою електронну скриньку, що вказана у командному рядку в якості адреса_отримувача, щоб переконатися, що повідомлення було успішно надіслано.

Якщо повідомлення отримано, змінюємо команди відсилання повідомлень. Для цього в конфігураційному файлі команд системи /usr/local/nagios/etc/objects/commands.cfg знайдемо відповідні команди:

```
define command {
    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATES$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s "*** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATES$ ***" $CONTACTEMAIL$
}
```

```

define command {
    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$\n" | /bin/mail -s "***
$NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS/$SERVICEDESC$ is $SERVICESTATE$ ***" $CONTACTEMAILS
}

```

Та змінимо їх наступним чином:

```

define command {
    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" |
/usr/bin/mail -s "NAGIOS WARNING" $CONTACTEMAILS
}

define command {
    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService:
$SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time:
$LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$\n" | /usr/bin/mail -s "NAGIOS WARNING" $CONTACTEMAILS
}

```

Виконуємо перевірку вірності внесених у конфігурацію змін. Перезапускаємо сервіси Nagios та Apache:

```

sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```

```

sudo service nagios restart

```

Якщо буде вимкнено будь який хост, або його сервіс, за яким ведеться спостереження, система повинна відправити поштове повідомлення на налаштовану поштову скриньку. Щоб переглянути логи поштового сервісу Postfix необхідно виконати команду:

```

sudo less /var/log/mail.log

```

Моніторинг мережевого шлюза.

Додаємо у конфігураційний файл груп хостів /usr/local/nagios/etc/objects/hostgroups.cfg групу, що буде відповідати за мережеві пристрої

```

# Define a hostgroup for Switches And Routers
define hostgroup{
    hostgroup_name    switches
    alias              Network Switches
}

```

Та коментуємо визначення цієї групи у файлі конфігурації мережевих пристроїв /usr/local/nagios/etc/objects/switch.cfg

```

# Create a new hostgroup for switches
#define hostgroup {
#   hostgroup_name    switches                ; The name of the hostgroup
#   alias              Network Switches       ; Long name of the group
#}

```

Для розміщення файлів конфігурації мережевих пристроїв використовуємо каталог /usr/local/nagios/etc/objects/network, що ми створили під час виконання однієї з попередніх робіт.

Перевіряємо у конфігураційному файлі /usr/local/nagios/etc/nagios.cfg

- присутність каталогу /usr/local/nagios/etc/objects/network
- вимикаємо загальний шаблон switch.cfg
- перевіряємо дозвіл користувачам "підтверджувати отримання попередження" про проблеми з хостами та сервісами.

```

#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
cfg_dir=/usr/local/nagios/etc/objects/network
check_external_commands=1

```

Створюємо конфігураційний файл для мережевого шлюза NAT Network /usr/local/nagios/etc/objects/network/snm-gw.cfg з наступним вмістом:

```

define host{

```

```

host_name      nat-gateway
alias          NAT Network Gateway-Switch
address        192.168.22.129
hostgroups     switches
contact_groups admins
check_command  check-host-alive
max_check_attempts 5
check_interval 15
retry_interval 1
check_period   24x7
notification_interval 60
notification_period 24x7
notifications_enabled 1
    }
# Create a service to PING to switch
define service{
    use                generic-service
    host_name          nat-gateway
    service_description PING
    check_command      check_ping!200.0,20%!600.0,60%
    }

```

Налаштуємо для всіх хостів VM підпорядкування у підключенні до хосту мережевого шлюза NAT Network nat-gateway. Для цього додаємо рядок "parents", що вказує на шлюз групи, до якої належить дана машина до всіх конфігураційних файлів хостів:

```

parents        nat-gateway

```

Наприклад, секція визначення хосту у файлі /usr/local/nagios/etc/objects/linux/localhost.cfg буде мати вигляд:

```

define host {
    hostgroups     linux-servers
    use            linux-server
    host_name      serv-22-1-2
    alias          Serv-22-1-2
    address        127.0.0.1
    contacts       igor
    parents        nat-gateway
}

```

Виконуємо перевірку вірності внесених у конфігурацію змін та перезапускаємо сервіс Nagios
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo service nagios restart

Результатом цього налаштування будуть зміни у відображенні зв'язків хостів у пункті меню Map (Legacy) сайту Nagios:

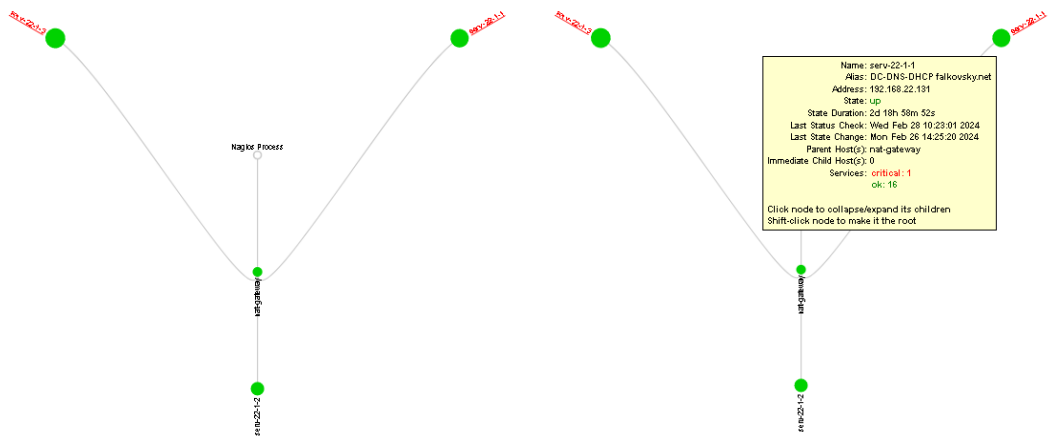


Рис. 8.2. Карта топології мережі після налаштування параметру parents хостів. Map (Legacy)

Створення конфігураційного файлу для груп сервісів

Створюємо новий файл `/usr/local/nagios/etc/objects/servicegroup.cfg`. У цьому файлі виконується визначення різних груп сервісів та прив'язка до них конкретних сервісів. Це налаштування виконується для відображення пункту меню Service Groups . Файл повинен мати вміст типу:

```
define servicegroup{
    servicegroup_name    cpuload
    alias                 CPU Load
    members               serv-22-1-1,CPU Load, serv-22-1-2,Current Load, serv-22-1-3,CPU Usage
}
```

У цьому прикладі створена група сервісів: "CPU Load" для виведення навантаження на CPU. Група має аліас та список членів.

Підключаємо конфігураційний файл до Nagios - додаємо новий рядок у конфігураційний файл Nagios `/usr/local/nagios/etc/nagios.cfg`, щоб вказати Nagios, що він має завантажити цей файл:

```
cfg_file=/usr/local/nagios/etc/objects/servicegroup.cfg
```

Виконуємо перевірку вірності внесених у конфігурацію змін та перезапускаємо сервіс Nagios:

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
sudo service nagios restart
```

The screenshot shows the Nagios web interface. On the left is a navigation menu with 'Service Groups' highlighted. The main content area displays the 'Current Network Status' (Last Updated: Wed Feb 28 18:08:13 UTC 2024), 'Host Status Totals' (4 Up, 0 Down, 0 Unreachable, 0 Pending), and 'Service Status Totals' (45 Ok, 0 Warning, 0 Unknown, 1 Critical, 0 Pending). Below this is the 'Service Overview For All Service Groups' section, which shows a table for the 'CPU Load (cpuload)' group:

Host	Status	Services	Actions
serv-22-1-1	UP	1 OK	[Icons]
serv-22-1-2	UP	1 OK	[Icons]
serv-22-1-3	UP	1 OK	[Icons]

Рис. 8.3. Налаштована група сервісів CPU Load

Встановлення та налаштування пакету NagiosQL3 (Nagios Web UI)

Методика встановлення пакету NagiosQL наведена для факультативного ознайомлення та не вимагається при виконанні курсу лабораторних робіт.

Встановлюємо необхідні пакунки:

```
sudo apt-get updates
sudo apt-get install libssh2-1 libssh2-1-dev
sudo apt-get install mysql-server
sudo apt-get install php php-gd libgd-dev libapache2-mod-php libperl-dev libssl-dev php-dev
sudo apt-get install php-php-gettext
sudo apt-get install php-mysqli
```

Завантажуємо останню версію NagiosQL з веб-сайту SourceForge, розархівуємо завантажений архів, переміщаємо отриману теку nagiosql-3.5.0 до каталогу /usr/local/nagios/share/webadmin.

```
cd /tmp
wget https://sourceforge.net/projects/nagiosql/files/latest/download
tar xzf nagiosql-3.5.0-git2023-06-18.tar.gz
sudo mv /tmp/nagiosql-3.5.0 /usr/local/nagios/share/webadmin
```

Змінюємо власника та групу усіх файлів та папок у цьому каталозі на www-data:www-data та встановлюємо права доступу на читання, запис та виконання для власника та групи, а також тільки для виконання для інших користувачів в цьому каталозі.

```
cd /usr/local/nagios/share/webadmin
sudo chown -R www-data:www-data .
chmod -R 775 .
```

Виконуємо пошук файлу php.ini в системі в директоріях, які можуть містити конфігурації PHP для Apache. Результат цього пошуку допомагає знайти шлях до конфігураційного файлу php.ini для Apache. Після цього відкриваємо для редагування знайдений файл php.ini

```
sudo find /etc -name "php.ini" | grep "apache"
sudo vi /etc/php/8.1/apache2/php.ini
```

Вносимо зміну в файлі конфігурації PHP (php.ini) у розділі [Date] - розкоментуємо рядок date.timezone та вказуємо значення Europe/Kiev для параметра date.timezone. Це налаштування визначає часовий пояс, який буде використовуватися PHP для роботи з функціями, пов'язаними з датою та часом. У нашому випадку, вказано, що часовий пояс для PHP буде встановлено в "Europe/Kiev".

```
[Date]
; Defines the default timezone used by the date functions
; https://php.net/date.timezone
date.timezone = 'Europe/Kiev'
```

Знімаємо коментар з підтримки розширення MySQLi

```
extension=mysqli
```

Перезавантажуємо apache

```
sudo systemctl restart apache2
```

Встановлюємо логін пароль підключення до MySQL . Входимо до MySQL без вимоги до пароля:

```
mysql -u root
```

Оновлюємо пароль підключення до MySQL для користувача root. Синтаксис передбачає, що використовується MySQL версії 8.0 або новіше. У наведеному прикладі у базі даних mysql для користувача root встановлюється пароль 12345.

```
USE mysql;
ALTER USER 'root'@'localhost' IDENTIFIED BY '12345';
quit;
```

Перезавантажуємо MySQL

sudo systemctl restart mysql

Створюємо каталог для розгортання пакету NagiosQL3 та перевіряємо цю дію:

sudo mkdir /etc/nagiosql

ls /etc/nagiosql

Надаємо на каталог відповідні дозволи, а саме дозволи на запис у цей каталог для користувача, який встановлює NagiosQL.

sudo chmod -R 755 /etc/nagiosql

sudo chown -R www-data:www-data /etc/nagiosql

ls -l /etc/nagiosql

Запускаємо NagiosQL3 Web Installer для налаштування пакета.

NagiosQL3 надає веб-інстальатор для встановлення. Відкриваємо URL-адресу

http://IP-NAGIOS/nagios/webadmin/install/index.php де IP-NAGIOS – адреса нашого серверу.

щоб запустити веб-інстальатор для nagiosql та виконуємо кроки, як показано на рис.8.4, 8.5. Веб-інстальатор допоможе внести будь-які необхідні зміни.

Для спрощення доступу до веб-інстальатора NagiosQL3, можливо відкрити «прокинуту» через NAT Network робочу адресу Nagios-серверу та змінити її лінк, додавши в кінець адреси webadmin/install/index.php. Слідуйте вказівкам, що наведені на рис.8.4-8.5.

The screenshot displays the NagiosQL installation wizard interface. The main window is titled 'Welcome to the NagiosQL installation wizard' and provides instructions for installing and configuring NagiosQL. It lists basic requirements for PHP (7.2.0 or above), session, gettext, filter, FTP, and SSH modules, and a database server. A secondary window on the right, titled 'NagiosQL Installation: Checking requirements', shows a progress bar and a list of checked items: Client (JavaScript enabled), PHP version (7.2.14), PHP extensions (PEAR, Security, gettext, Fileinfo), database interfaces (MySQL), php.ini settings (upload_max_filesize, session.auto_start, date.timezone), and system permissions (config/settings.php, writable settings directory, readable config files, readable templates, readable locale files, readable phpinfo.php, readable composer.json).

Рис. 8.4. WEB-інстальатор NagiosQL. Друге вікно – перевірка сумісності пакетів.

На наступному екрані необхідно ввести деталі бази даних, які будуть використовуватися для nagiosql. Зверніть увагу на створення адміністративного користувача для доступу до інтерфейсу NagiosQL. Під час встановлення потрібно вказати ім'я користувача (Initial NagiosQL User) та пароль (Initial NagiosQL Password), які ви будете використовувати для входу в систему NagiosQL. Зазвичай це користувач admin

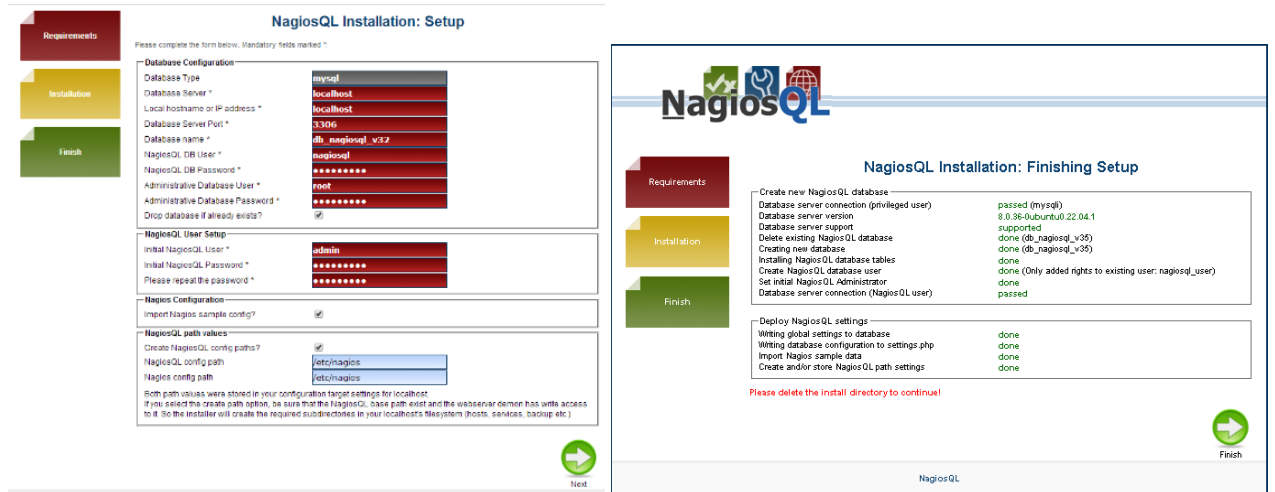


Рис. 8.5. WEB-інсталятор NagiosQL. Деталі БД для nagiosql.

Після завершення веб-інсталятора він автоматично перенаправиться до панелі адміністратора nagiosql3. Щоб відкрити його пізніше за допомогою URL <http://IP-NAGIOS /nagios/webadmin> де IP-NAGIOS – адреса нашого серверу.

Входимо до розділу адміністрування NagiosQL і перейдіть до Administration -> Administration -> Config targets та натискаємо кнопку Модифу для локальної інсталяції.

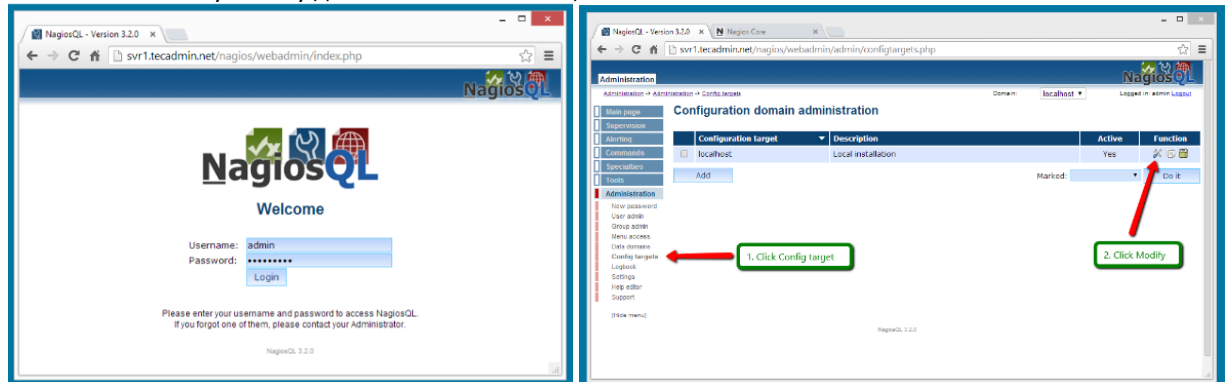


Рис. 8.6. Налаштування NagiosQL.

Коректні шляхи для параметрів Nagios configuration files and directories:

```

Nagios base directory: /usr/local/nagios/etc/
Picture base directory: (залишаємо порожнім, якщо немає окремого каталогу для зображень)
Nagios command file: /usr/local/nagios/var/rw/nagios.cmd
Nagios binary file: /usr/local/nagios/bin/nagios
Nagios process file: /usr/local/nagios/var/nagios.lock
Nagios config file: /usr/local/nagios/etc/nagios.cfg
Nagios cgi file: /usr/local/nagios/etc/cgi.cfg
Nagios resource file: /usr/local/nagios/etc/resource.cfg

```

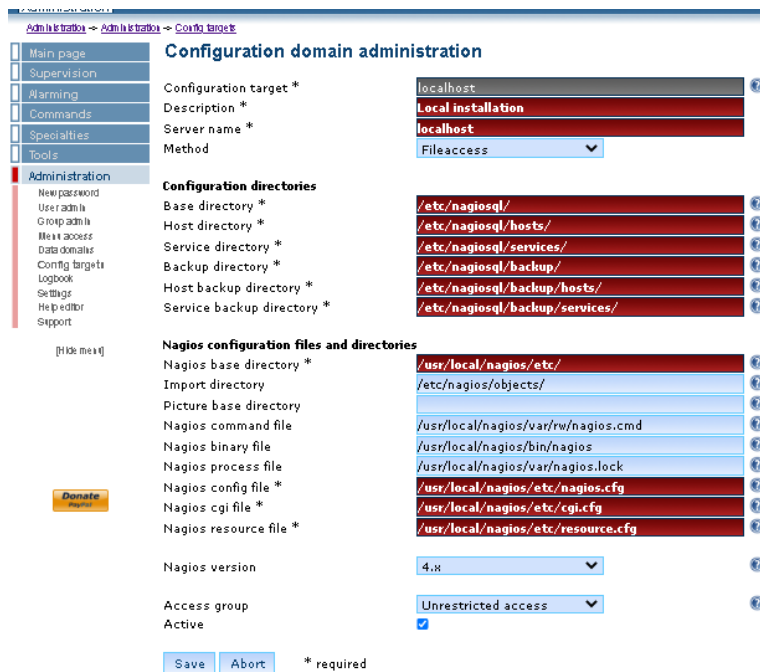


Рис. 8.7. Шляхи при налаштуванні NagiosQL.

Корисні посилання

- Nagios Core. Time Periods.
<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/timeperiods.html>
- Nagios Core. Time Period Definition
<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/objectdefinitions.html#timeperiod>
- Nagios Core. CGI Configuration File Options
<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/configcgi.html>
- How to Install and Use SendEmail on Linux
<https://tecadmin.net/how-to-install-sendemail-in-linux/>
- NagiosQL - Nagios configuration tool Files
<https://sourceforge.net/projects/nagiosql/files/nagiosql/>