



Лабораторна робота №4

Моніторинг віддалених Linux-хостів за допомогою Icinga 2 Agent.

Мета: Набути практичних навичок моніторингу віддалених Linux-хостів у системі Icinga 2. Ознайомитися з процесом додавання нових хостів до Master-сервера, забезпечити їх інтеграцію через Icinga 2 Agent, налаштувати перевірку базових сервісів (SSH, HTTP) та застосування шаблонів для уніфікованого моніторингу. Забезпечити відображення даних моніторингу у веб-інтерфейсі Icinga Web 2 та перевірити коректність обміну даними між Master та агентами. Додатково студентам пропонується факультативне ознайомлення з модулем Icinga Director для спрощеного створення хостів, шаблонів та сервісів через GUI, а також автоматичного деплою конфігурації на Core через API.

Інструменти: гіпервізор VirtualBox, модель комп'ютерної мережі.

Теоретичні відомості

У попередніх лабораторних роботах було створено віртуалізоване стендове середовище у VirtualBox, що складається з трьох хостів:

Serv-G-N-1 (Windows Server 2022) – на якому розгорнуто контролер домену з ролями AD DS, DNS і DHCP та встановлено Icinga 2 Agent для локального моніторингу серверних ресурсів;

Serv-G-N-3 (Ubuntu Server 24.04) – на якому встановлено та налаштовано систему моніторингу Icinga 2 разом із веб-інтерфейсом Icinga Web 2 і компонентами Icinga DB;

WS-G-N-1 (Windows 10) – робоча станція, включена до внутрішнього домену.

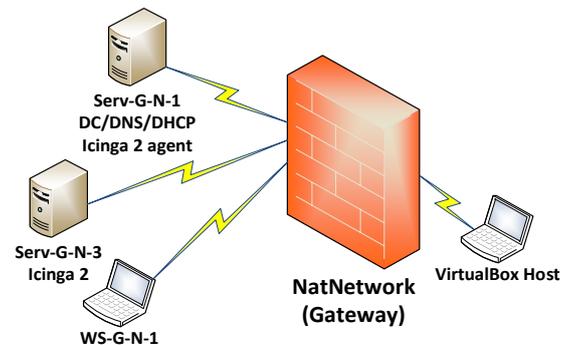


Рис. 4.1. Топологія мережі

Мережеве середовище забезпечує взаємодію між вузлами, а система Icinga 2 інтегрована з доменною інфраструктурою для подальшого моніторингу її елементів.

Імпорт з аплайнсу та налаштування Serv-G-N-5

Аналогічно процедурі, описаній у одній з попередніх робіт виконуємо імпорт віртуальної машини з аплайнсу Ubuntu сервера, що доступний по лінку

https://drive.google.com/file/d/1zhqV0hGwcXdaHpiuUf7g0a8z8oXqN6kD/view?usp=drive_link

Після імпорту віртуальної машини Serv-G-N-3 (сервер на базі Ubuntu Server) необхідно перейменувати її по шаблону Serv-G-N-5 та регенерувати MAC-адресу мережевого адаптеру (рис. 4.2).

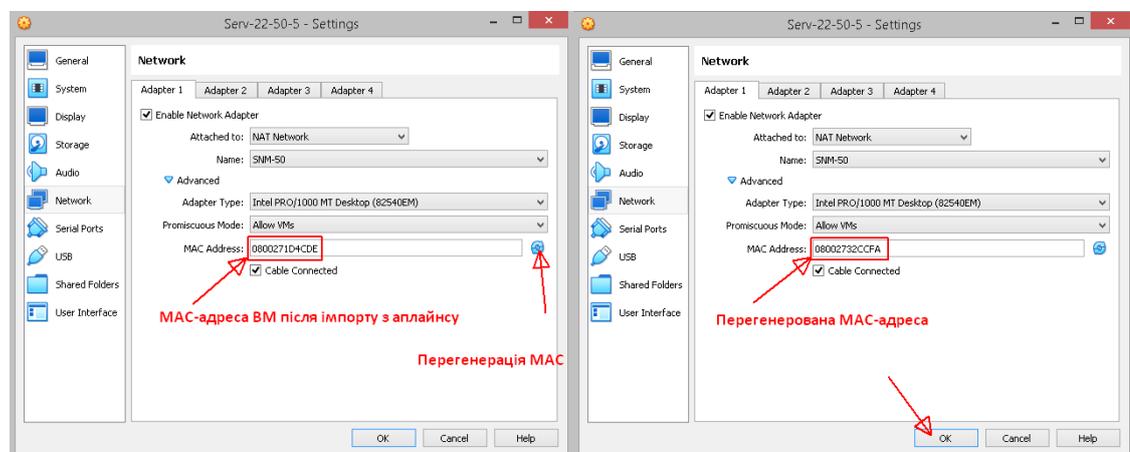


Рис. 4.2. Зміна MAC-адреси Ubuntu серверу Serv-22-50-5

Вмикаємо VM та повторюємо дії, що виконувались для підключення Serv-G-N-3 у навчальний стенд. Авторизація – обліковий запис з правами адміністратора: ім'я користувача – student, пароль – 111111

Якщо налаштування виконані вірно, сервер автоматично отримує IP-адресу від служби DHCP у NAT-мережі, а його ім'я відповідатиме шаблону Serv-G-N-3. Для уточнення мережевих параметрів, імені хосту та його перейменування використовуємо такі команди (рис.4.3):



```
ip a
hostname
sudo hostnamectl set-hostname New-Name-Server
sudo reboot
```

Після перезавантаження нове ім'я набуде чинності.

```
Serv-22-50-5 [Running] - Oracle VM VirtualBox
student@serv-g-n-3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:cc:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.16/27 metric 100 brd 192.168.50.31 scope global dynamic enp0s3
        valid_lft 690785sec preferred_lft 690785sec
    inet6 fe80::a00:27ff:fe32:ccfa/64 scope link
        valid_lft forever preferred_lft forever
student@serv-g-n-3:~$ hostname
serv-g-n-3
student@serv-g-n-3:~$ sudo hostnamectl set-hostname serv-22-50-5
student@serv-g-n-3:~$ sudo reboot
```

Рис. 4.3. Перегляд поточних IP-адрес, імені Ubuntu серверу Serv-G-N-3 та його перейменування на serv-22-50-5

Налаштуємо переадресацію для організації SSH доступу до Ubuntu серверу з фізичної машини – хосту VirtualBox. Пункт виконується у меню **[Oracle VirtualBox Manager] – [File] – [Tools] – [Network Manager] – [Nat Network] – [Properties] – [Port Forwarding] – [IPv4]**.

Назва правила NAT – “Serv-G-N-5 SSH”,
де G – група,
N – варіант, що Ви виконуєте, протокол – “TCP”.

У якості Guest IP задаємо IP-адресу сервера, яку ми отримали за допомогою команди **ip a**, Port – 22 – порт «за замовчуванням» для SSH доступу.

У якості Host IP задаємо IP-адресу нашого фізичного ПК, що використовується для налаштованого раніше правила “Serv-G-N-3 SSH”.

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
SERV-22-50-3 HTTP	TCP	192.168.56.1	80	192.168.50.5	80
SERV-22-50-3 SSH	TCP	192.168.56.1	2200	192.168.50.5	22
SERV-22-50-5 SSH	TCP	192.168.56.1	2201	192.168.50.16	22

Рис. 4.4. Налаштування правила переадресації портів для SSH до серверу Serv-22-50-5.

У результаті, конфігурація Port Forwarding для забезпечення SSH-доступу з фізичного хосту до VM Serv-G-N-5 (Ubuntu Server) має вигляд, зображений на рис. 4.4. У якості порту переадресації обрано 2201.

Всі подальші дії з командним рядком Ubuntu рекомендовано виконувати за допомогою SSH-клієнта.

Serv-22-50-5 конфігурований на динамічну адресацію і адресу він отримує від доменного DHCP. Сервери, зазвичай, працюють на статиці. Змінюємо налаштування динамічної адреси на статичну. Шукаємо назву мережевого інтерфейсу, який необхідно відредагувати ip a

Шукаємо конфігураційні файли Netplan (з розширенням YAML), що зберігаються в каталозі /etc/netplan. Ймовірно, у цьому каталозі є один або декілька файлів YAML. Назва файлу може відрізнитися залежно від налаштувань та версії операційної системи.

dir /etc/netplan

Наприклад, у каталозі знайдено файл 50-cloud-init.yaml. Робимо його копію для редагування:

```
sudo cp /etc/netplan/50-cloud-init.yaml /etc/netplan/50-natnet.yaml
```

Відкриваємо для редагування

```
sudo vi /etc/netplan/50-natnet.yaml
```



Нижче наведено вміст файлу Netplan до та після редагування.

/etc/netplan/50-cloud-init.yaml --- DHCP

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
```

/etc/netplan/50-natnet.yaml Static 192.168.50.5/27

```
network:
  version: 2
  ethernets:
    enp0s3:
      addresses: [192.168.50.7/27]
      gateway4: 192.168.50.1
      nameservers:
        addresses: [192.168.50.3,192.168.50.1]
```

Зберігаємо відредагований Netplan та застосовуємо внесені зміни:

sudo netplan apply

Перевірка ip a (рис.4.5) показує, що на інтерфейсі enp0s3 активні дві одночасні IPv4-адреси:

192.168.50.5/27 — ручна статична адреса, задана через netplan;

192.168.50.16/27 — динамічна (DHCP) адреса позначена як secondary dynamic.

Приховуємо ☺ динамічну конфігурацію та застосовуємо зміни:

sudo mv /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.bkp
sudo netplan apply

SSH-підключення в результаті «відвалюється», бо зникла адреса Guest IP Port Forwarding. Задаємо у налаштуванні нову адресу.

The screenshot shows a terminal window with the output of the 'ip a' command. The 'enp0s3' interface is shown with two IP addresses: 192.168.50.16 (secondary dynamic) and 192.168.50.7 (static). Below the terminal is a 'Port Forwarding' configuration window with a table of rules:

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
SERV-22-50-3 HTTP	TCP	192.168.56.1	80	192.168.50.5	80
SERV-22-50-3 SSH	TCP	192.168.56.1	2200	192.168.50.5	22
SERV-22-50-5 SSH	TCP	192.168.56.1	2201	192.168.50.7	22

Рис. 4.5. Застосування статичної адресації (192.168.50.7/27) після редагування конфігурації та зміна задіяної адреси у налаштуваннях Port Forwarding.

The screenshot shows a terminal window with the following commands and output:

```
student@serv-22-50-5:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:cc:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.7/27 brd 192.168.50.31 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80:a00:27ff:fe32:ccfa/64 scope link
        valid_lft forever preferred_lft forever

student@serv-22-50-5:~$ ping ztu.edu.ua
PING ztu.edu.ua (46.4.92.154) 56(84) bytes of data:
64 bytes from static.154.92.4.46.clients.your-server.de (46.4.92.154): icmp_seq=1 ttl=56 time=33.8 ms
64 bytes from static.154.92.4.46.clients.your-server.de (46.4.92.154): icmp_seq=2 ttl=56 time=33.6 ms
64 bytes from static.154.92.4.46.clients.your-server.de (46.4.92.154): icmp_seq=3 ttl=56 time=33.2 ms
^C
--- ztu.edu.ua ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 33.205/33.550/33.805/0.253 ms

student@serv-22-50-5:~$ ping falkovsky.net
PING falkovsky.net (192.168.50.3) 56(84) bytes of data:
64 bytes from 192.168.50.3: icmp_seq=1 ttl=128 time=0.553 ms
64 bytes from 192.168.50.3: icmp_seq=2 ttl=128 time=0.474 ms
64 bytes from 192.168.50.3: icmp_seq=3 ttl=128 time=0.581 ms
^C
--- falkovsky.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.474/0.536/0.581/0.045 ms

student@serv-22-50-5:~$
```

Рис. 4.6. Перевірка мережевих налаштувань, зв'язку з приватним доменом falkovsky.net та публічним доменом ztu.edu.ua



Підготовка Ubuntu-хосту та встановлення Icinga 2 Agent

Перед встановленням агенту актуалізуємо система та встановлюємо базові пакети, необхідні для роботи Icinga 2. Оновлюємо список доступних пакетів, встановлюємо останні версії наявних пакетів та встановлюємо допоміжні утиліти, які знадобляться для завантаження репозиторіїв і пакетів Icinga 2:

```
sudo apt update
sudo apt upgrade -y
sudo apt install curl gnupg wget apt-transport-https -y
```

Наступний крок – додавання офіційного репозиторію Icinga та встановлення агента. Додаємо GPG-ключ для перевірки пакетів:

```
curl https://packages.icinga.com/icinga.key | sudo gpg --dearmor -o /usr/share/keyrings/icinga-archive-keyring.gpg
```

Підключаємо репозиторій Icinga (для Ubuntu 24.04, кодова назва noble):

```
echo "deb [signed-by=/usr/share/keyrings/icinga-archive-keyring.gpg]
https://packages.icinga.com/ubuntu icinga-noble main" | sudo tee /etc/apt/sources.list.d/icinga.list
```

Оновлюємо список пакетів та встановлюємо Icinga 2 Agent:

```
sudo apt update
sudo apt install icinga2 -y
```

Перевіряємо статус служби:

```
systemctl status icinga2
```

```
student@serv-22-50-5:~$ systemctl status icinga2
● icinga2.service - Icinga host/service/network monitoring system
   Loaded: loaded (/usr/lib/systemd/system/icinga2.service; enabled; preset: enabled)
   Drop-In: /etc/systemd/system/icinga2.service.d
           └─ limits.conf
   Active: active (running) since Fri 2025-08-29 14:11:03 UTC; 36s ago
   Process: 24024 ExecStartPre=/usr/lib/icinga2/prepare-dirs /etc/default/icinga2 (code=exited, status=0/SUCCESS)
   Main PID: 24032 (icinga2)
   Status: "Startup finished."
   Tasks: 6
   Memory: 12.0M (peak: 12.2M)
   CPU: 322ms
   CGroup: /system.slice/icinga2.service
           └─ 24032 /usr/lib/x86_64-linux-gnu/icinga2/sbin/icinga2 --no-stack-klimit daemon --close-stdio -e /var/log/icinga2/error.log
             24052 /usr/lib/x86_64-linux-gnu/icinga2/sbin/icinga2 --no-stack-klimit daemon --close-stdio -e /var/log/icinga2/error.log
             24055 /usr/lib/x86_64-linux-gnu/icinga2/sbin/icinga2 --no-stack-klimit daemon --close-stdio -e /var/log/icinga2/error.log

Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 247 CheckCommands.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 1 User.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 1 UserGroup.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 3 ServiceGroups.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 3 TimePeriods.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 12 Services.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ConfigItem: Instantiated 1 ScheduledDowntime.
Aug 29 14:11:03 serv-22-50-5 icinga2[24052]: [2025-08-29 14:11:03 +0000] information/ScriptGlobal: Dumping variables to file '/var/cache/icinga2/icinga2.vars'
Aug 29 14:11:03 serv-22-50-5 icinga2[24032]: [2025-08-29 14:11:03 +0000] information/cli: Closing console log.
Aug 29 14:11:03 serv-22-50-5 systemd[1]: Started icinga2.service - Icinga host/service/network monitoring system.
student@serv-22-50-5:~$
```

Рис. 4.7. Перевірка стану Icinga 2 Agent

Виконуємо налаштування Icinga 2 Agent для інтеграції з Master (тобто сервером Icinga 2, розгорнутим на Serv-22-50-3. Тут потрібно вказати Master-сервер, з яким агент буде обмінюватися даними, налаштувати сертифікати для безпечного з'єднання та переконатися, що агент отримує конфігурацію від Master.

Зупиняємо службу агента перед налаштуванням (щоб уникнути конфліктів):

```
sudo systemctl stop icinga2
```

Запускаємо майстер налаштування агента:

```
sudo icinga2 node wizard
```

У процесі потрібно буде відповісти на питання:

```
Please specify if this is a satellite setup ('n' installs a master setup) [Y/N]: → Y (це агент)
Please specify the common name (CN) [serv-22-50-5]: → serv-g-n-5
Master/Satellite Common Name (CN): → serv-g-n-3 (CN Master-сервера)
Do you want to establish a connection to the parent node from this node? [Y/N]: → Y
Master/Satellite endpoint host (IP address or FQDN): → 192.168.50.5 (IP Master, перевірте свою адресу)
Master/Satellite endpoint port [5665]: → 5665
Add more master/satellite endpoints? [Y/N]: → N
Is this information correct? [y/N]: → Y
Please specify the request ticket generated on your Icinga 2 master (optional).
(Hint: # icinga2 pki ticket --cn 'serv-22-50-5'):
```

На цей запит необхідно переключитись у консоль Master (serv-22-50-3) та виконати команду:

```
sudo icinga2 pki ticket --cn 'serv-22-50-5'
```

(зверніть увагу, тут треба CN саме агента – serv-22-50-5).

Master видасть довгий токен (рядок типу 928a1a8fb1c631c487fadb7f844121f5fe31ee54), який необхідно вставте у вікно агенту, де він зараз чекає значення. Таким чином майстер підтвердить, що цей

```
student@serv-22-50-3:~$ sudo icinga2 pki ticket --cn 'serv-22-50-5'
[sudo] password for student:
928a1a8fb1c631c487fadb7f844121f5fe31ee54
student@serv-22-50-3:~$
```

Рис. 4.8. Генерація ticket для serv-22-50-5



агент має право підключитися.

Повертаємося до вікна icinga2 node wizard та продовжуємо:

```
Please specify the request ticket generated on your Icinga 2 master (optional).
(Hint: # icinga2 pki ticket --cn 'serv-22-50-5'): → 928a1a8fb1c631c487fadb7f844121f5fe31ee54
Please specify the API bind host/port (optional):
Bind Host []: → Enter
Bind Port []: → Enter
Local zone name [serv-22-50-5]: → serv-22-50-5
Parent zone name [master]: → serv-22-50-3
Do you want to specify additional global zones? [y/N]: → N
Do you want to disable the inclusion of the conf.d directory [Y/n]: → N
```

```
student@serv-22-50-5:~$ sudo icinga2 node wizard
Welcome to the Icinga 2 Setup Wizard!

We will guide you through all required configuration details.

Please specify if this is an agent/satellite setup ('n' installs a master setup) [Y/n]: Y

Starting the Agent/Satellite setup routine...

Please specify the common name (CN) [serv-22-50-5]: serv-22-50-5

Please specify the parent endpoint(s) (master or satellite) where this node should connect to:
Master/Satellite Common Name (CN from your master/satellite node): serv-22-50-3

Do you want to establish a connection to the parent node from this node? [Y/n]: Y
Please specify the master/satellite connection information:
Master/Satellite endpoint host (IP address or FQDN): 192.168.50.5
Master/Satellite endpoint port [5665]: 5665

Add more master/satellite endpoints? [y/N]: N
Parent certificate information:

Version:          3
Subject:          CN = serv-22-50-3
Issuer:           CN = Icinga CA
Valid From:       Aug 17 15:10:45 2025 GMT
Valid Until:      Sep 18 15:10:45 2026 GMT
Serial:          54:91:b7:a9:b8:d6:eb:c2:8e:81:e1:55:70:2e:53:b3:cc:4c:d3:09

Signature Algorithm: sha256WithRSAEncryption
Subject Alt Names:  serv-22-50-3
Fingerprint:      85 30 AC 50 C3 C8 95 63 42 55 D4 62 03 99 E6 16 06 1A E5 ED AE AC 05 5E DA C1 44 14 0A 92 33 5F

Is this information correct? [y/N]: Y

Please specify the request ticket generated on your Icinga 2 master (optional).
(Hint: # icinga2 pki ticket --cn 'serv-22-50-5'): 928a1a8fb1c631c487fadb7f844121f5fe31ee54
Please specify the API bind host/port (optional):
Bind Host []:
Bind Port []:
```

```
Bind Host []:
Bind Port []:

Accept config from parent node? [y/N]: y
Accept commands from parent node? [y/N]: y

Reconfiguring Icinga...

Local zone name [serv-22-50-5]: serv-22-50-5
Parent zone name [master]: serv-22-50-3

Default global zones: global-templates director-global
Do you want to specify additional global zones? [y/N]: n

Do you want to disable the inclusion of the conf.d directory [Y/n]: n

Done.

Now restart your Icinga 2 daemon to finish the installation!
student@serv-22-50-5:~$
```

Рис. 4.9. Виконання icinga2 node wizard на Serv-22-50-5

Після завершення Wizard слід перезапустити службу Icinga 2 на Ubuntu-хості, щоб застосувати всі налаштування агента, перевірити її статус та синтаксис усіх конфігураційних файлів агента:

```
sudo systemctl restart icinga2
sudo systemctl status icinga2
sudo icinga2 daemon -C
```




```
}  
object Zone "serv-22-50-5" {  
    parent = "master"  
    endpoints = [ "serv-22-50-5" ]  
}
```

Перевіряємо і застосовуємо оновлену конфігурацію та перевіряємо статус служби:

```
sudo icinga2 daemon -C  
sudo systemctl reload icinga2  
sudo systemctl status icinga2
```

Якщо все вірно, на master-сервері serv-22-50-3 створюємо директорію для зони хоста `/etc/icinga2/zones.d/serv-22-50-5`, а у цій директорії створюємо файл з описом хосту `/etc/icinga2/zones.d/serv-22-50-5/host.conf`, де додаємо таку конфігурацію:

```
object Host "serv-22-50-5" {  
    import "generic-host"  
    address = "192.168.50.7"  
    check_command = "hostalive"  
    vars.os = "Linux"  
}
```

Зберігаємо файл конфігурації хосту, перевіряємо синтаксис:

```
sudo icinga2 daemon -C
```

перечитуємо конфігурацію icinga2:

```
sudo systemctl reload icinga2
```

Якщо все вірно — через кілька хвилин у веб-інтерфейсі (Icinga Web 2) має з'явитися новий хост serv-22-50-5 зі станом UP/DOWN. Після додавання цього хосту система автоматично створила для нього два сервіси:

- ping4 — базова перевірка доступності вузла через ICMP (ping).
- ssh — перевірка доступності сервісу SSH на стандартному порту 22.

```
student@serv-22-50-3:~$ sudo mkdir /etc/icinga2/zones.d/serv-22-50-5  
student@serv-22-50-3:~$ sudo vi /etc/icinga2/zones.d/serv-22-50-5/host.conf  
student@serv-22-50-3:~$ sudo icinga2 daemon -C  
[2025-08-30 14:58:38 +0000] information/cli: Icinga application loader (version: r2.15.0-1)  
[2025-08-30 14:58:38 +0000] information/cli: Loading configuration file(s).  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Committing config item(s).  
[2025-08-30 14:58:38 +0000] information/ApiListener: My API identity: serv-22-50-3  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 13 Notifications.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 2 NotificationCommands.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 IcingaApplication.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 2 HostGroups.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 3 Hosts.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 Downtime.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 IcingaDB.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 FileLogger.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 4 Zones.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 CheckerComponent.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 2 Endpoints.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 2 ApiUsers.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 ApiListener.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 NotificationComponent.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 247 CheckCommands.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 User.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 UserGroup.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 3 ServiceGroups.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 3 TimePeriods.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 24 Services.  
[2025-08-30 14:58:38 +0000] information/ConfigItem: Instantiated 1 ScheduledDowntime.  
[2025-08-30 14:58:38 +0000] information/ScriptGlobal: Dumping variables to file '/var/cache/icinga2/icinga2.vars'  
[2025-08-30 14:58:38 +0000] information/cli: Finished validating the configuration file(s).  
student@serv-22-50-3:~$ sudo systemctl reload icinga2  
student@serv-22-50-3:~$
```

Рис. 4.11. Створення конфігурації хосту serv-22-50-5.

Налаштування базових сервісів моніторингу Linux-хосту

Після додавання хосту serv-22-50-5 можна створити кілька перевірок для відстеження ключових параметрів системи. На master-сервері serv-22-50-3 у директорії зони хосту створюємо файл `/etc/icinga2/zones.d/serv-22-50-5/services.conf` і додаємо такі перевірки:

```
apply Service "load" {  
    import "generic-service"  
    display_name = "CPU Load Average"  
    check_command = "load"  
    assign where host.name == "serv-22-50-5"  
}  
apply Service "disk" {  
    import "generic-service"  
    display_name = "Disk Usage /"
```



```
check_command = "disk"
vars.disk_partitions = [ "/" ]
vars.disk_w = 80
vars.disk_c = 90
assign where host.name == "serv-22-50-5"
}
```

cpu-load — показує середнє навантаження на процесор (1, 5, 15 хвилин).

disk — використання файлової системи.

Зберігаємо файл, перевіряємо синтаксис та перезавантажуємо службу:

```
sudo icinga2 daemon -C
sudo systemctl reload icinga2
```

Через кілька хвилин у веб-інтерфейсі Icinga Web 2 біля хоста serv-22-50-5 мають з'явитися нові сервіси. Якщо хост успішно з'явився у веб-інтерфейсі та його базові перевірки працюють, можна перейти до додавання розширених перевірок. На цьому етапі одразу додамо кілька базових сервісів моніторингу. Назви сервісів будуть зрозумілі з параметра display_name у кожній секції конфігурації. У файл `/etc/icinga2/zones.d/serv-22-50-5/services.conf` додаємо такі перевірки:

```
apply Service "swap" {
import "generic-service"
display_name = "Swap Usage"
check_command = "swap"
vars.swap_w = 20
vars.swap_c = 10
assign where host.name == "serv-22-50-5"
}
apply Service "procs" {
import "generic-service"
display_name = "Number of Processes"
check_command = "procs"
vars.procs_warn = 150
vars.procs_crit = 200
assign where host.name == "serv-22-50-5"
}
apply Service "users" {
import "generic-service"
display_name = "Logged-in Users"
check_command = "users"
assign where host.name == "serv-22-50-5"
}
```

Повторюємо дії, по відпрацьованому алгоритму. Зберігаємо файл, перевіряємо синтаксис та перезавантажуємо службу:

```
sudo icinga2 daemon -C
sudo systemctl reload icinga2
```

Через кілька хвилин у веб-інтерфейсі Icinga Web 2 біля хоста serv-22-50-5 мають з'явитися нові сервіси.

Алгоритм автоматичного додавання сервісів для хосту

Після створення хосту serv-22-50-5 і перезапуску Icinga 2 заходимо у веб-інтерфейс Icinga Web 2. Вибираємо новий хост і бачимо, що вже налаштований базовий моніторинг: ping, SSH, CPU load, Disk, Swap, Processes, Users та інші сервіси. Звертає увагу, що ми вручну додавали лише кілька сервісів, але два з'явилося автоматично. Це сервіси ssh та ping4. Якщо ми переглянемо сервіси хосту serv-22-50-3, то побачимо що там їх трошки більше ☺

Це сталося завдяки механізму шаблонів та правил Icinga 2. Master-сервер застосовує шаблон generic-service та правила apply Service для нового хосту, створюючи стандартні перевірки без ручного додавання. Таким чином, автоматично додаються базові сервіси, а специфічні перевірки можна додавати власноруч у файлі `/etc/icinga2/conf.d/services.conf`.

Додамо «зрозумілі» назви кільком сервісам, що створено через цей шаблон. Відкриваємо для редагування файл `/etc/icinga2/conf.d/services.conf` та додаємо у секції шаблонів сервісів рядки з параметром display_name для деяких сервісів:

```
apply Service "ping4" {
...
display_name = "Ping IPv4"
...
}
```



```
apply Service "ssh" {
...
  display_name = "SSH Connectivity"
...
apply Service "icinga" {
...
  display_name = "Icinga 2 Process"
...
apply Service "procs" {
...
  display_name = "Number of Processes"
...
apply Service "users" {
...
  display_name = "Logged-in Users"
...
apply Service "swap" {
...
  display_name = "Swap Usage"
...
apply Service "load" {
...
  display_name = "CPU Load Average"
...
apply Service for (disk => config in host.vars.disks) {
...
  display_name = "Disk Usage"
```

Повторюємо перевірку та перезавантаження конфігурації служби:

```
sudo icinga2 daemon -C
sudo systemctl reload icinga2
```

Через кілька хвилин у веб-інтерфейсі Icinga Web 2 переглядаємо сервіси хостів serv-22-50-3 та serv-22-50-5. Отримані результати (рис. 4.12) не забуваємо долучити до звіту.

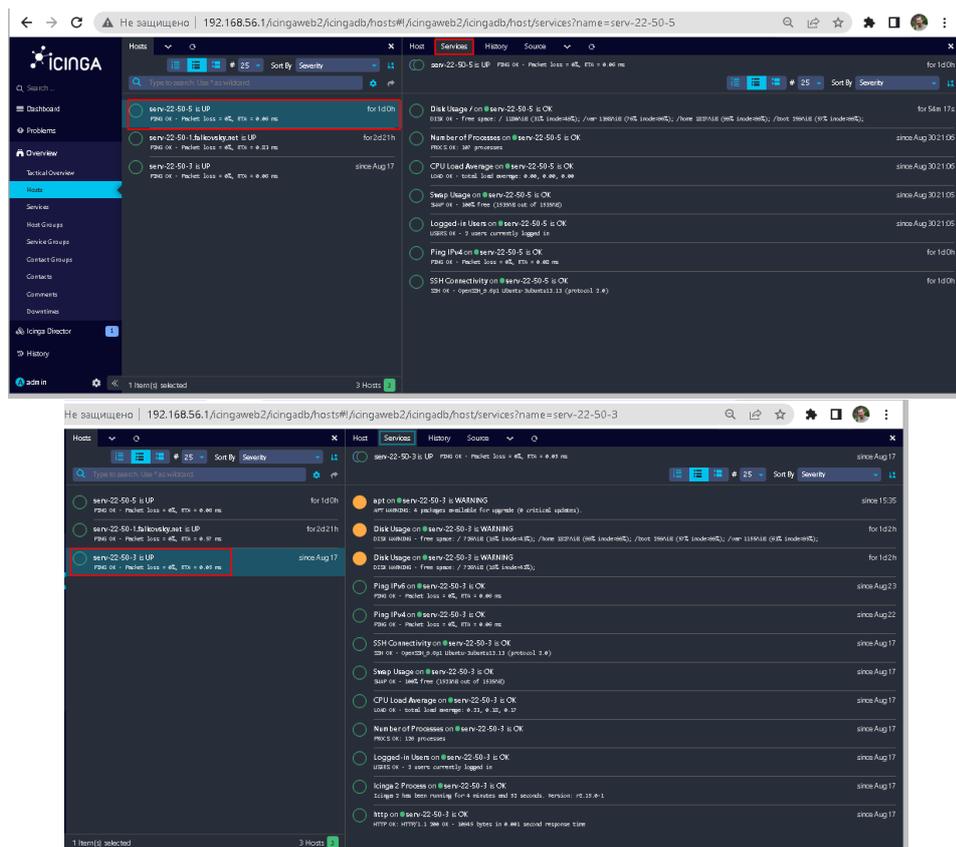


Рис. 4.12. WEB UI сервісу serv-22-50-5 та serv-22-50-3.

Завдання до лабораторної роботи



1. Встановити Icinga 2 Agent на Ubuntu (Serv-G-N-5) і підключити до Master (Serv-G-N-3). Перезапустити службу та перевірити статус (systemctl restart icinga2, systemctl status icinga2).
2. На Master створити конфігурацію хоста /etc/icinga2/zones.d/serv-22-50-5/host.conf із FQDN, IP та шаблоном generic-host. Перевірити (icinga2 daemon -C) і перезавантажити (systemctl reload icinga2).
3. Додати базові сервіси: Disk, CPU Load, Swap, Processes, Users (шаблон generic-service). Перевірити появу та статус у Icinga Web 2.
4. Протестувати сервіси через CLI (check_ping, check_load, check_disk, check_procs, check_swap, check_users) і виправити помилки у конфігурації.
5. Підготувати звіт із описом встановлення, конфігурацій хоста та сервісів, перевірки CLI та веб-інтерфейсу, коротким описом усунення помилок.

Додаток 1.

Набір CLI-команд для перевірки сервісів serv-22-50-5.

У цьому додатку наведено повний набір CLI-аналогів команд використаних у побудові файлу конфігурації /etc/icinga2/zones.d/serv-22-50-5/services.conf. Синім кольором позначено **рядки запрошення**, зеленим – **CLI-команди** а червоним – **відповіді на ці команди**.

```
student@serv-22-50-3:~$ ssh student@192.168.50.7 "/usr/lib/nagios/plugins/check_disk -w 80% -c 90% -p /"  
student@192.168.50.7's password:
```

```
DISK CRITICAL - free space: / 1180MiB (31% inode=49%); | /=2673868800B;828584755;414292377;0;4142923776
```

```
student@serv-22-50-3:~$ ssh student@192.168.50.7 "/usr/lib/nagios/plugins/check_load -w 5,4,3 -c 10,6,4"  
student@192.168.50.7's password:
```

```
LOAD OK - total load average: 0.08, 0.08, 0.04 | load1=0.080;5.000;10.000;0; load5=0.080;4.000;6.000;0;  
load15=0.040;3.000;4.000;0;
```

```
student@serv-22-50-3:~$ ssh student@192.168.50.7 "/usr/lib/nagios/plugins/check_swap -w 20% -c 10%"  
student@192.168.50.7's password:
```

```
SWAP OK - 100% free (1535MB out of 1535MB) | swap=1609564160B;321912820;160956410;0;1609564160
```

```
student@serv-22-50-3:~$ ssh student@192.168.50.7 "/usr/lib/nagios/plugins/check_procs -w 150 -c 200"  
student@192.168.50.7's password:
```

```
PROCS OK: 112 processes | procs=112;150;200;0;
```

```
student@serv-22-50-3:~$ ssh student@192.168.50.7 "/usr/lib/nagios/plugins/check_users -w 5 -c 10"  
student@192.168.50.7's password:
```

```
USERS OK - 2 users currently logged in | users=2;5;10;0
```



Методи визначення сервісів у Icinga 2

В Nagios ми робили купу `define service { ... host_name X }` або `hostgroup_name`, і це було досить жорстко. В Icinga 2 через `apply rules` і `vars` ми отримуємо дуже гнучку систему.

- ✓ Один і той самий сервіс для всіх хостів. Приклад - `ping` для всіх:

```
apply Service "ping4" {
    import "generic-service"
    check_command = "ping4"
    assign where host.address
}
```

Тепер будь-який хост із полем `address` автоматично отримає цей сервіс.

- ✓ Сервіси тільки для певних груп/OS. Наприклад, CPU-тести тільки для Windows:

```
apply Service "windows-cpu" {
    import "generic-service"
    check_command = "nscp-local-cpu"
    assign where host.vars.os == "Windows"
}
```

Або диски тільки для Linux:

```
apply Service "linux-disk" {
    import "generic-service"
    check_command = "disk"
    assign where host.vars.os == "Linux"
}
```

- ✓ Унікальні сервіси для окремих хостів (наприклад, додаткові диски). Використовуємо змінні (`vars`) прямо в описі хоста:

```
object Host "serv-22-50-1.falkovsky.net" {
    import "generic-host"
    address = "192.168.50.3"
    vars.os = "Windows"
    // оголошуємо, які диски моніторити
    vars.disks = [ "C:", "D:" ]
}
```

А в сервісах пишемо універсальне правило:

```
apply Service "disk" for (disk => config in host.vars.disks) {
    import "generic-service"
    check_command = "disk"
    vars.disk_partitions = [ disk ]
}
```

Результат - якщо у хоста є тільки "C:", то створиться лише перевірка диску C, якщо "C:" і "D:" — будуть два сервіси, якщо змінної `vars.disks` немає — сервіс взагалі не створиться.

- ✓ "Hardware" відмінності. Тут теж "рулить" `vars`. Наприклад, можна в хост додати:

```
vars.is_db_server = true
А правило:
apply Service "mssql" {
    check_command = "mssql"
    assign where host.vars.is_db_server
}
```

Отже, робиться одна універсальна конфігурація, а поведінка підлаштовується під хости завдяки `vars`.



Приклад структурованої конфігурації icinga 2

В Icinga2 дуже важливо правильно структурувати файли конфігурації, щоб потім не заплутатися. У цьому додатку приведено приклад логічного поділу (аналогічний Nagios — хости, сервіси, групи). Структура конфігурацій у каталозі /etc/icinga2/conf.d

```
/etc/icinga2/conf.d/
├── templates.conf      # шаблони хостів і сервісів
├── hosts/
│   ├── dc1.conf       # опис Domain Controller
│   ├── dns1.conf      # опис DNS сервера
│   └── dhcp1.conf      # опис DHCP сервера
├── services/
│   ├── base_services.conf # універсальні сервіси (ping, cpu, memory)
│   ├── role_services.conf # сервіси за ролями (AD, DNS, DHCP)
└── groups.conf        # групи хостів і сервісів (опціонально)
```

✓ templates.conf (шаблони хостів і сервісів)

```
template Host "windows-host" {
    import "generic-host"
    vars.os = "Windows"
}
```

```
template Service "generic-windows-service" {
    import "generic-service"
}
```

✓ sts/dc1.conf (опис контролера домену)

```
object Host "dc1.surname.net" {
    import "windows-host"
    address = "192.168.50.10"
    vars.is_domain_controller = true
    vars.is_dns_server = true
    vars.is_dhcp_server = false
}
```

✓ hosts/dhcp1.conf (опис DHCP сервера)

```
object Host "dhcp1.surname.net" {
    import "windows-host"
    address = "192.168.50.11"
    vars.is_domain_controller = false
    vars.is_dns_server = false
    vars.is_dhcp_server = true
}
```

✓ services/base_services.conf (універсальні сервіси для всіх Windows-хостів)

```
apply Service "ping4" {
    import "generic-service"
    check_command = "ping4"
    assign where host.address
}
```

```
apply Service "cpu" {
    import "generic-service"
    check_command = "nscp-local-cpu"
    assign where host.vars.os == "Windows"
}
```



```
apply Service "memory" {
  import "generic-service"
  check_command = "nscp-local-memory"
  assign where host.vars.os == "Windows"
}
```

- ✓ services/role_services.conf (сервіси за ролями)

```
apply Service "Active Directory Replication" {
  import "generic-service"
  check_command = "nscp-local-service"
  vars.nscp_service = "NTDS"
  assign where host.vars.is_domain_controller
}

DNS
apply Service "DNS Service" {
  import "generic-service"
  check_command = "nscp-local-service"
  vars.nscp_service = "DNS"
  assign where host.vars.is_dns_server
}

DHCP
apply Service "DHCP Service" {
  import "generic-service"
  check_command = "nscp-local-service"
  vars.nscp_service = "DHCPserver"
  assign where host.vars.is_dhcp_server
}
```

- ✓ (опціонально) groups.conf (групи хостів і сервісів)

```
object HostGroup "domain-controllers" {
  display_name = "Active Directory Controllers"
  assign where host.vars.is_domain_controller
}

object HostGroup "dns-servers" {
  display_name = "DNS Servers"
  assign where host.vars.is_dns_server
}

object HostGroup "dhcp-servers" {
  display_name = "DHCP Servers"
  assign where host.vars.is_dhcp_server
}
```

Така структура дозволяє чітко розділити хости, шаблони і сервіси, додавати новий сервер простим створенням файлу hosts/ім'я.conf та включати сервіси автоматично через прапорці ролей.

Корисні посилання

- How to set up Icinga2 agent monitoring.
<https://www.techrepublic.com/article/how-to-set-up-icinga2-agent-monitoring/>
- ICINGA Package Repository. Windows.
<https://packages.icinga.com/windows/>
- Icinga. Quickstart.
<https://icinga.com/docs/get-started/latest/>