**Linux Networking**

1. [На Server\_1 налаштувати статичні адреси на всіх інтерфейсах](#Перший_підтаск).
2. [На Server\_1 налаштувати DHCP сервіс, який буде конфігурувати адреси Int1 Client\_1 та Client\_2.](#Другий_підтаск)
3. [За допомогою команд ping та traceroute перевірити зв'язок між віртуальними машинами. Результат пояснити.](#Третій_підтаск)

4. [На віртуальному інтерфейсу lo Client\_1 призначити дві ІР адреси за таким](#Четвертий_підтаск)

[правилом: 172.17.D+10.1/24 та 172.17.D+20.1/24. Налаштувати маршрутизацію таким чином, щоб трафік з Client\_2 до 172.17.D+10.1 проходив через Server\_1, а до 172.17.D+20.1 через Net4. Для перевірки використати traceroute.](#Четвертий_підтаск)

5. [Розрахувати спільну адресу та маску (summarizing) адрес 172.17.D+10.1 та 172.17.D+20.1, при чому префікс має бути максимально можливим. Видалити маршрути, встановлені на попередньому кроці та замінити їх об’єднаним маршрутом, якій має проходити через Server\_1.](#Пятий_підтаск)

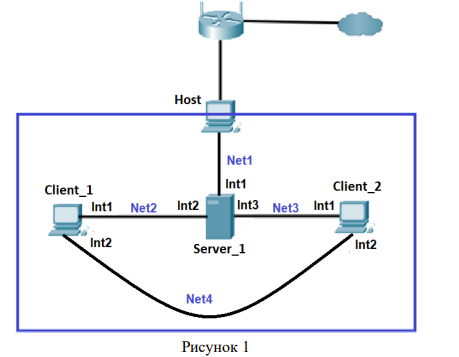
6. [Налаштувати SSH сервіс таким чином, щоб Client\_1 та Client\_2 могли підключатись до Server\_1 та один до одного.](#Шостий_підтаск)

7. [Налаштуйте на Server\_1 firewall таким чином:](#Сьомий_підтаск)

• Дозволено підключатись через SSH з Client\_1 та заборонено з Client\_2

• З Client\_2 на 172.17.D+10.1 ping проходив, а на 172.17.D+20.1 не проходив

Практична частина модуля Linux Networking передбачає створення засобами Virtual Box мережі, що показаний на рисунку 1.



**Host** – це комп’ютер, на якому запущений Virtual Box;

**Server\_1** – Віртуальна машина, на якій розгорнуто ОС Linux. **Int1** цієї машини в режимі «**Мережевий міст**» підключений до мережі **Net1**, тобто знаходиться в адресному просторі домашньої мережі. IP-адреса **Int1** встановлюється статично відповідно до адресного простору, наприклад 192.168.1.200/24. Інтерфейси **Int2** та **Int3** відповідно підключено в режимі «**Внутрішня мережа**» до мереж **Net2** та **Net3**.

**Client\_1** та **Client\_2** – Віртуальні машини, на яких розгорнуто ОС Linux (бажано різні дистрибутиви, наприклад Ubuntu та CentOS). Інтерфейси підключені в режимі «**Внутрішня мережа**» до мереж **Net2**, **Net3** та **Net4** як показано на рисунку 1.

Адреса мережі **Net2** – 10.Y.D.0/24, де **Y** – дві останні цифри з вашого року народження, **D** – дата народження.

Адреса мережі **Net3** – 10.M.Y.0/24, де **M** – номер місяця народження.

Адреса мережі **Net4** – 172.16.D.0/24.

**Увага!** Якщо, адресний простір **Net2**, **Net3** або **Net4** перетинається з адресним простором **Net1** – відповідну адресу можна змінити на власний розсуд.

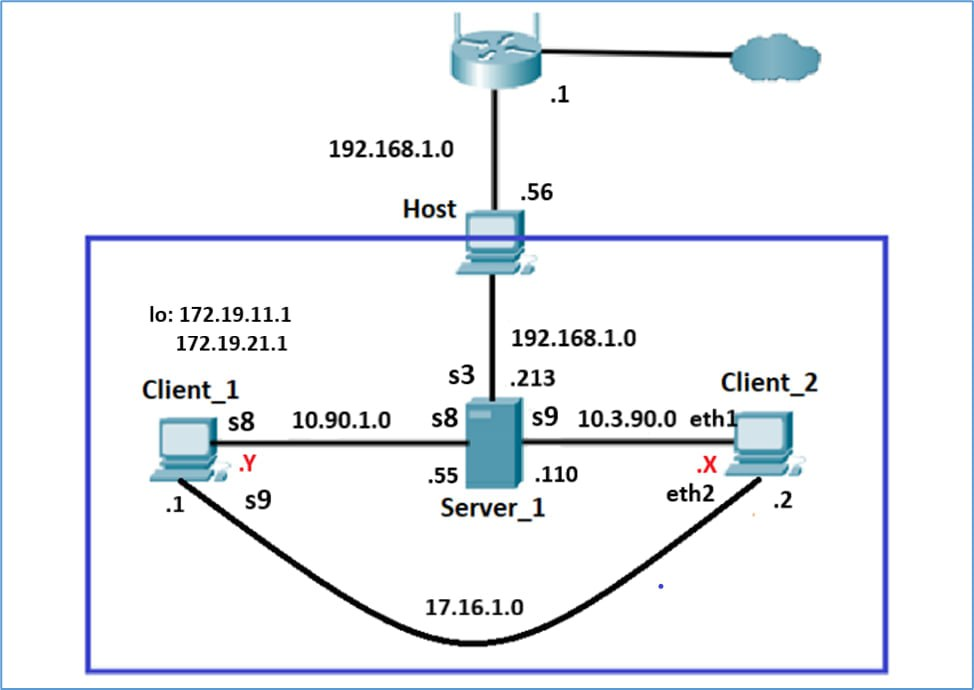
Адреса мережі **Net1** – 192.168.1.0/24

Адреса мережі **Net2** – 10.90.1.0/24

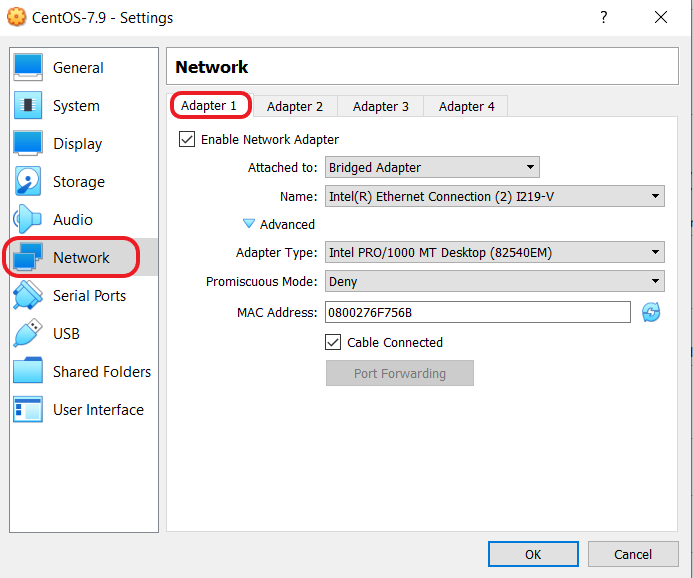
Адреса мережі **Net3** – 10.3.90.0/24

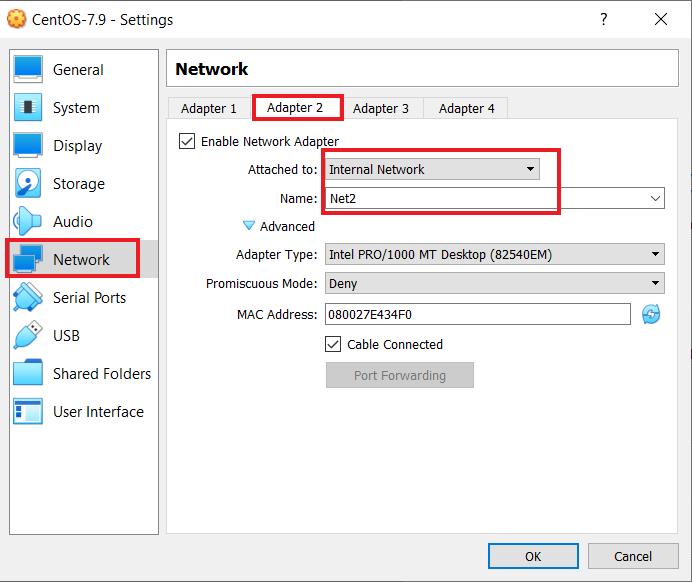
Адреса мережі **Net4** – 172.16.1.0/24

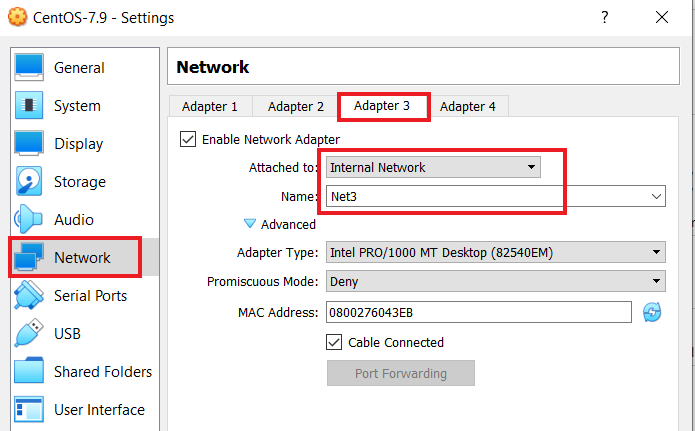
* **Network topology:**



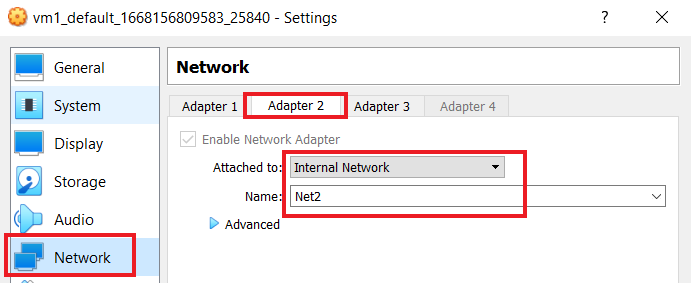
* **Setting network on Server1 (CentOS 7):**

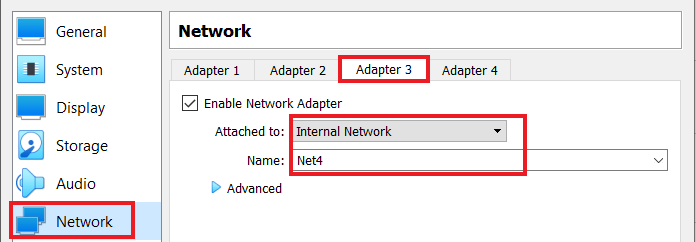




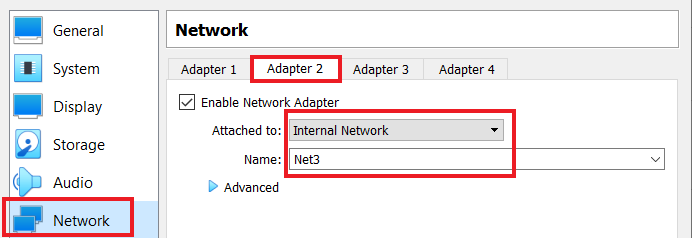


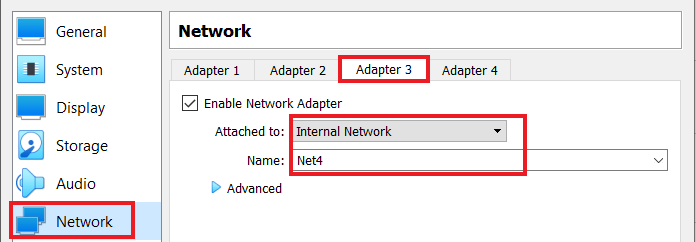
* **Setting network on Client1 (Ubuntu 22.04):**

****

****

* **Setting network on Client2 (CentOS 7):**

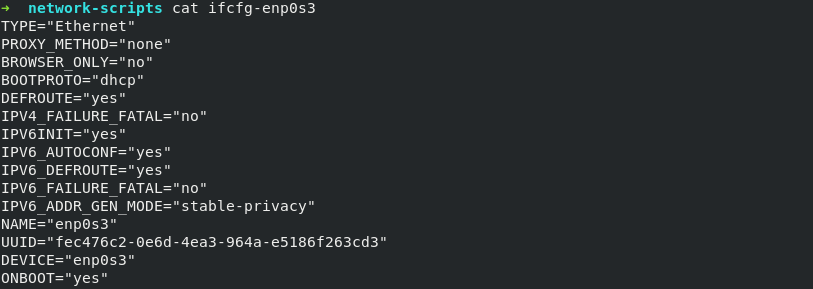
****

****

1. На Server1 налаштувати статичні адреси на всіх інтерфейсах.

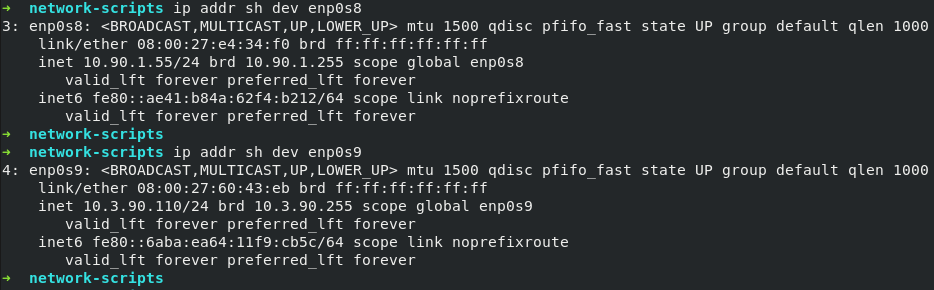
**Configuring static network addresses on Server1 (CentOS 7):**











For these changes to take effect, you must restart the network interfaces and service with the following commands:

$ sudo systemctl stop NetworkManager

$ sudo systemctl disable NetworkManager

$ sudo ifdown <<interface\_name>>

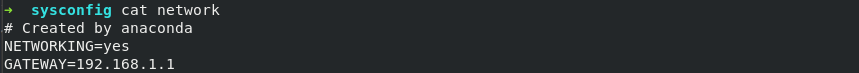
$ sudo ifup <<interface\_name>>

$ sudo systemctl restart network

**Configuring The Default Gateway (CentOS 7):**

The **default gateway** is determined by the network scripts which parse the **/etc/sysconfig/network** file first and then the network interface **ifcfg** files for interfaces that are “up”. The **ifcfg** files are parsed in numerically ascending order, and the **last GATEWAY** directive to be read is used to compose a **default route** in the **routing table**.

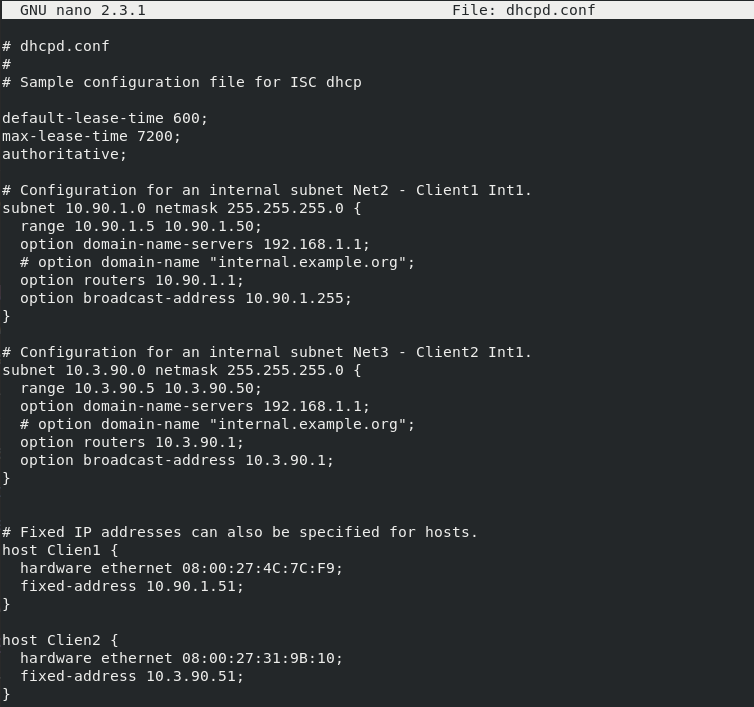
In special cases where it is necessary to influence NetworkManager's selection of the exit interface to be used to reach a gateway, make use of the DEFROUTE=”no” command in the **ifcfg** files for those interfaces which do not lead to the default gateway.



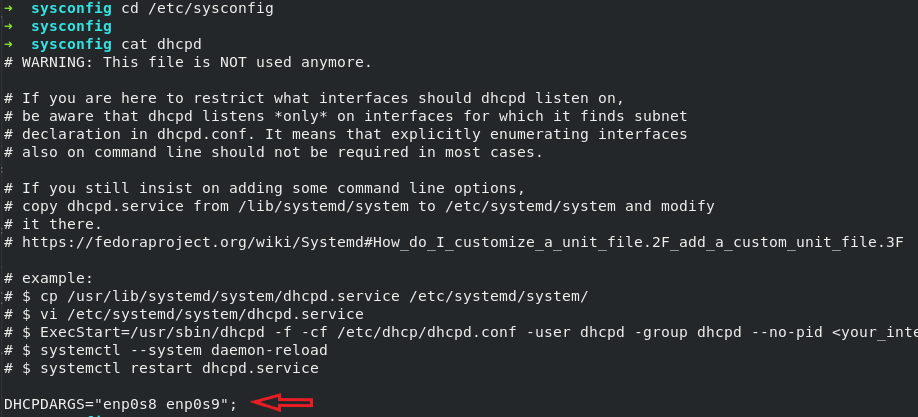
1. На Server1 налаштувати DHCP, який буде конфігурувати адреси Int1 Client1 та Client2.

**Configuring DHCP server on Server1 (CentOS 7):**

The following is a **/etc/dhcp/dhcpd.conf** file, for a server that has two network interfaces, **enp0s8** in a 10.90.1.0/24 network, and **enp0s9** in a 10.3.90.0/24 network. Multiple subnet declarations allow you to define different settings for multiple networks:



**The DHCP daemon** listens on all network interfaces unless otherwise specified. Use the **/etc/sysconfig/dhcpd** file to specify which network interfaces the **DHCP** daemon listens on. The following **/etc/sysconfig/dhcpd** example specifies that the **DHCP** daemon listens on the **enp0s8** and **enp0s9** interfaces:

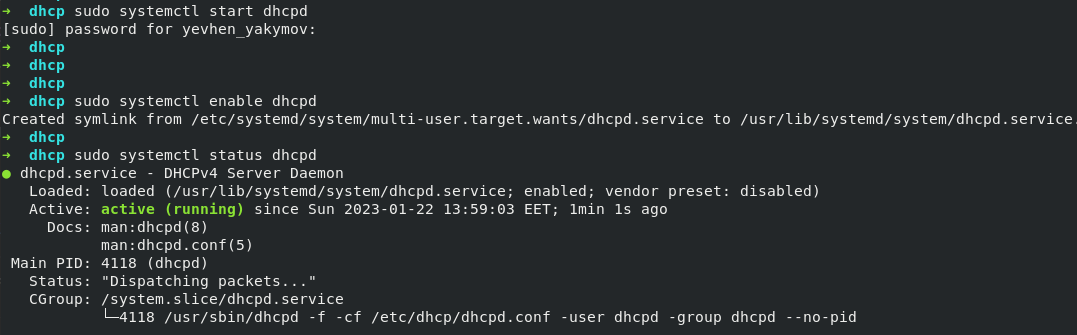


Now that you have configured your **DHCP** server daemon, you need to **start** the service for the mean time and **enable** it to start automatically from the next system boot, and **check** if its up and running using following commands:

$ sudo systemctl start dhcpd

$ sudo systemctl enable dhcpd

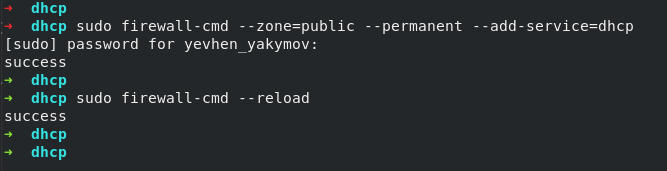
$ sudo systemctl status dhcpd



Next, permit requests to the **DHCP** daemon on **Firewall**, which listens on port **67/UDP**, by running:

$ firewall-cmd --zone=public --permanent --add-service=dhcp

$ firewall-cmd --reload



**Enable or disable IP forwarding:**

Switch on routing is needed only on transit devices. To check out routing enable use **sysctl net.ipv4.conf.all.forwarding** command. To switch “on” or “off” routing you must edit **/etc/sysctl.conf** file or use **sysctl -w net.ipv4.ip\_forward=1(0)** command.

$ sysctl net.ipv4.ip\_forward

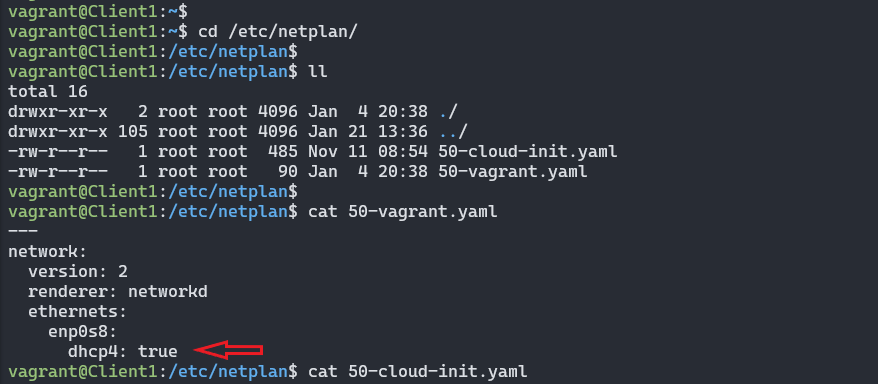
$ sysctl -w net.ipv4.ip\_forward=1

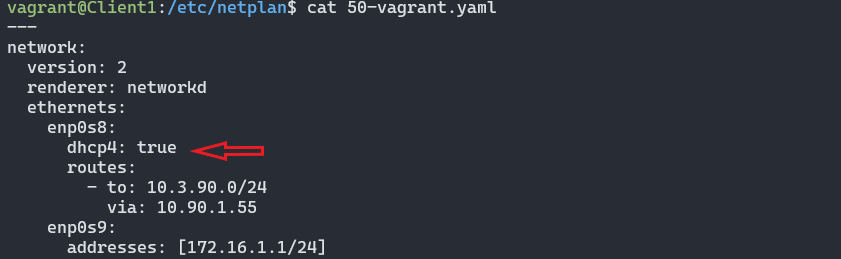
1. – Disable
2. - Enable

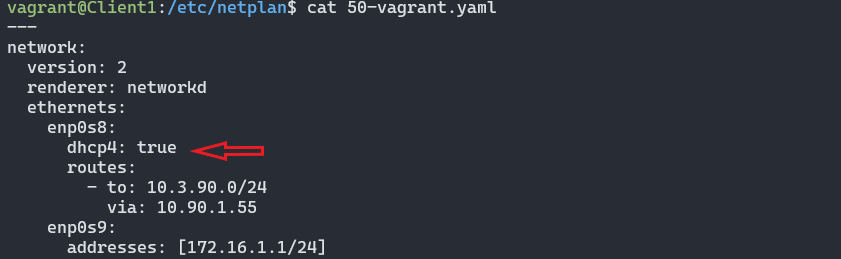
**Configuring DHCP clients.**

**Client1 – Ubuntu 22.04:**

On Ubuntu 18.04 and later, networking is controlled by the **Netplan** program. You need to edit the appropriate file under the directory **/etc/netplan/**. Then enable **dhcp4** under a specific interface for example under ethernets, **enp0s8**, and comment out static IP related configs:

****

****

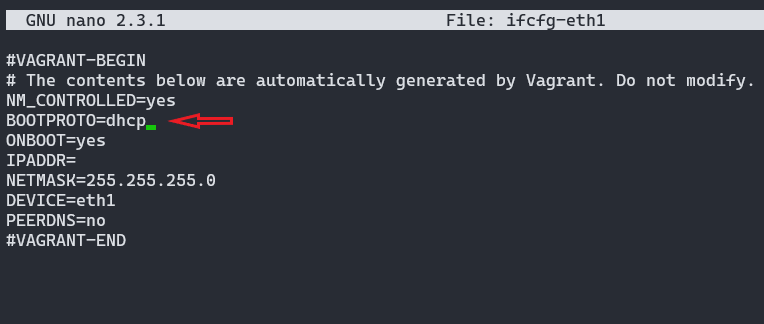
****

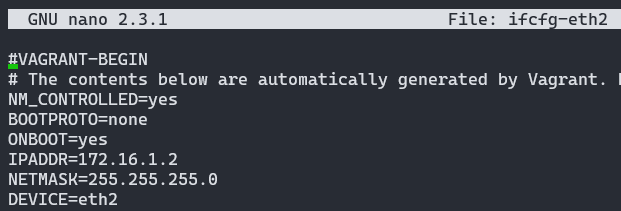
Save the changes and run the following command to effect the changes:

$ sudo netplan apply

**Client2 – CentOS 7:**

On CentOS, the interface config files ate located at **/etc/sysconfig/network-scripts/:**

****

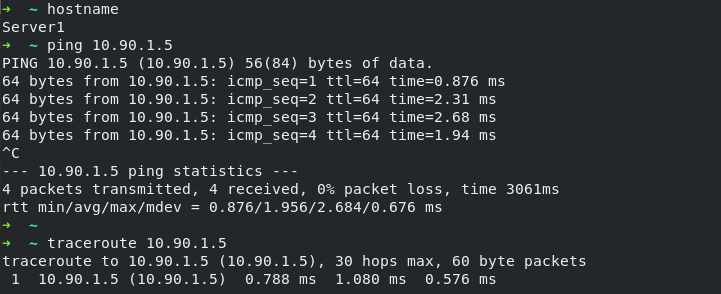


Save the file and restart network service (or reboot the system):

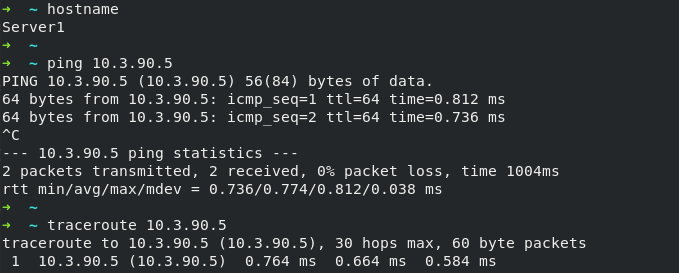
$ sudo systemctl restart network

1. За допомогою команд ping та traceroute перевірити зв'язок між віртуальними машинами. Результат пояснити.

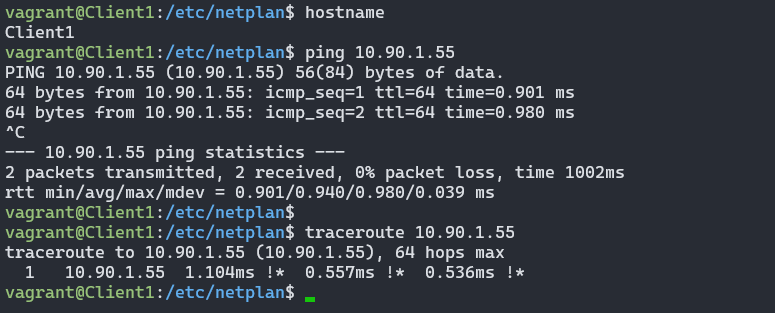
**Ping and traceroute from Server1 (Int2-10.90.1.55) to Client1 (Int1-10.90.1.5):**



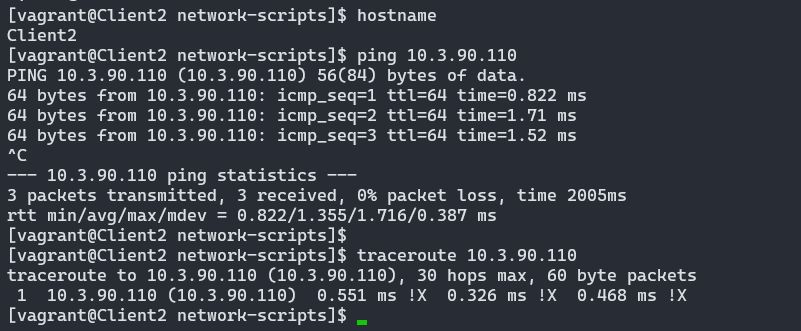
**Ping and traceroute from Server1 (Int3-10.3.90.110) to Client2 (Int1-10.3.90.5):**



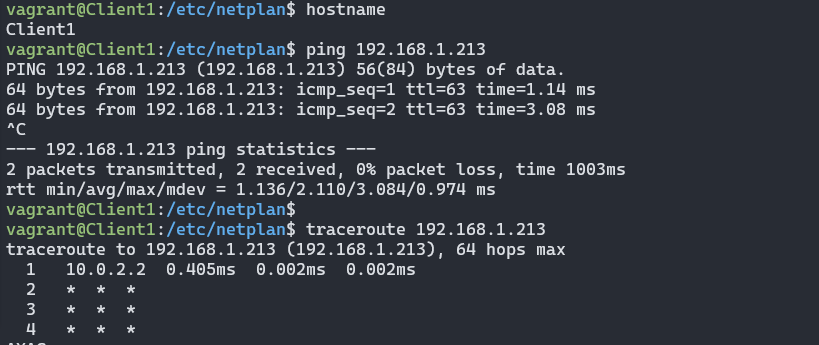
**Ping and traceroute from Client1 (Int1-10.90.1.5) to Server1 (Int2 - 10.90.1.55):**



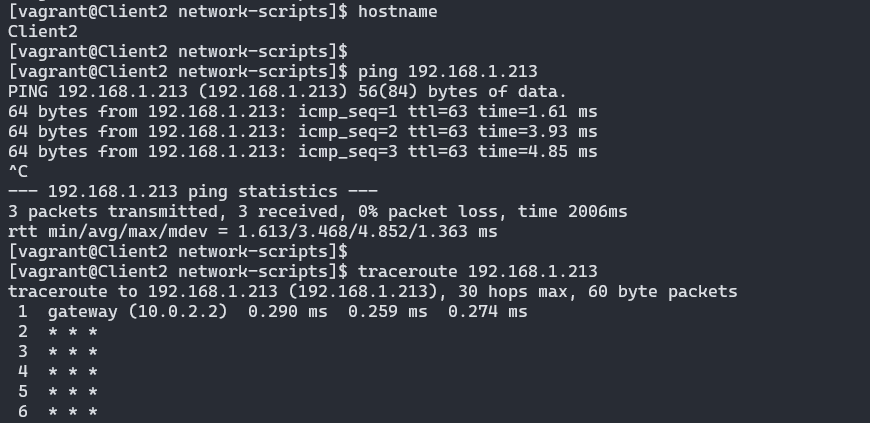
**Ping and traceroute from Client2 (Int1-10.3.90.5) to Server1 (Int3 - 10.3.90.110):**



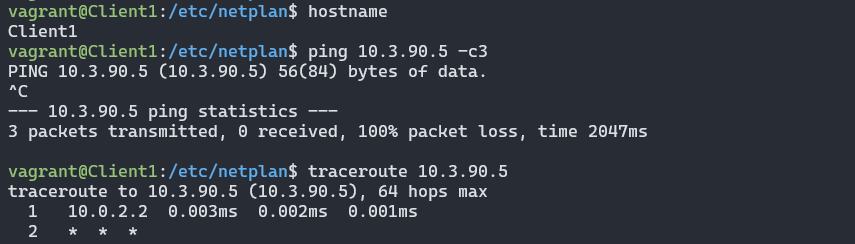
**Ping and traceroute from Client1 (Int1-10.90.1.5) to Server1 (Int1 - 192.168.1.213):**



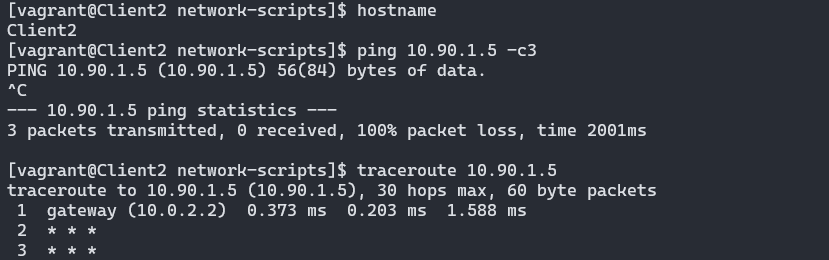
**Ping and traceroute from Client2 (Int1-10.3.90.5) to Server1 (Int1 - 192.168.1.213):**



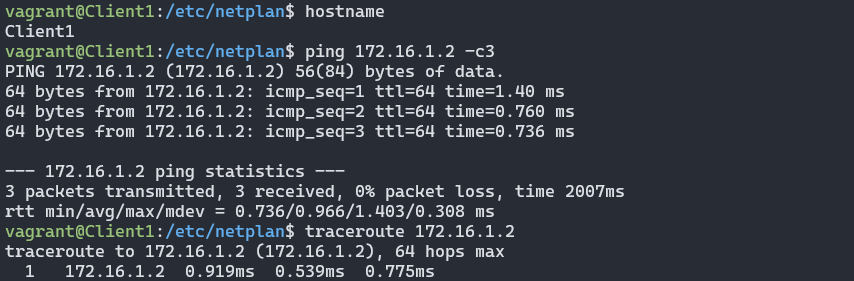
**Ping and traceroute from Client1 (Int1-10.90.1.5) to Client2 (Int1 – 10.3.90.5):**



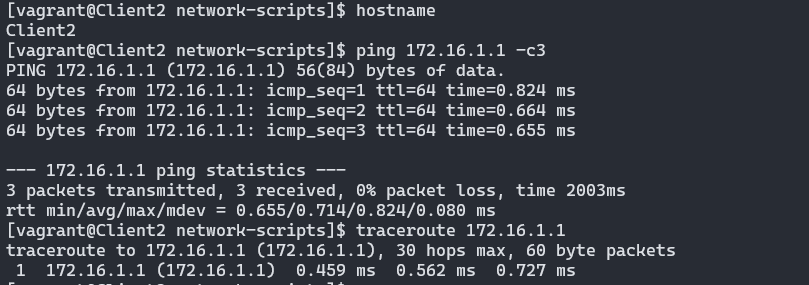
**Ping and traceroute from Client2 (Int1-10.3.90.5) to Client1 (Int1 - 10.90.1.5):**



**Ping and traceroute from Client1 (Int2-172.16.1.1) to Client2 (Int2 – 172.16.1.2):**



**Ping and traceroute from Client2 (Int2 – 172.16.1.2) to Client1 Int2-172.16.1.1):**

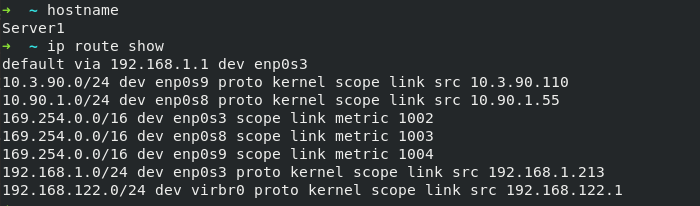


**Conclutions:**

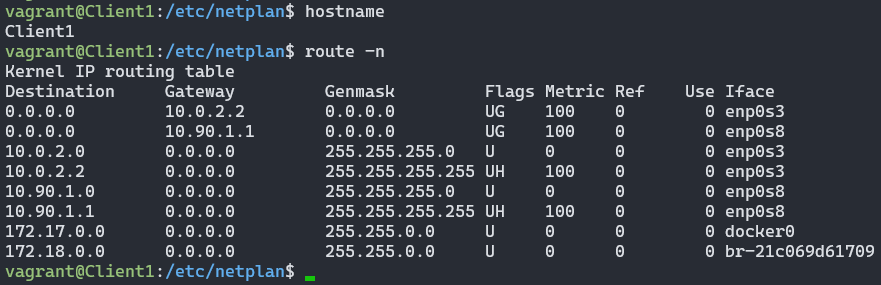
The **Client1** cannot ping **Client2** via **Server1**, because they are not in the same subnet: **10.90.1.0/24** for the **Client1(enp0s8)** network and **10.3.90.0/24** for the **Client2(eth1)** network. As the two hosts are not part of the same subnet, the ping command goes to the default gateway. So we should add static routes for **Client1** and **Client2.**

And the same time **Client1** can ping **Client2** from interface **enp0s9**, because they are in the same subnet: **172.16.1.0/24** for the **Client1(enp0s9)** network and **172.16.1.0/24** for the **Client2(eth2).**

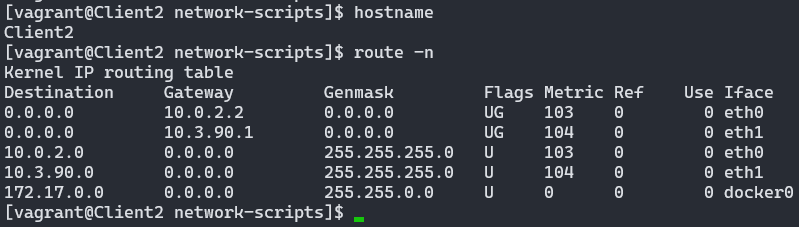
**Route table for Server1:**



**Route table for Client1:**



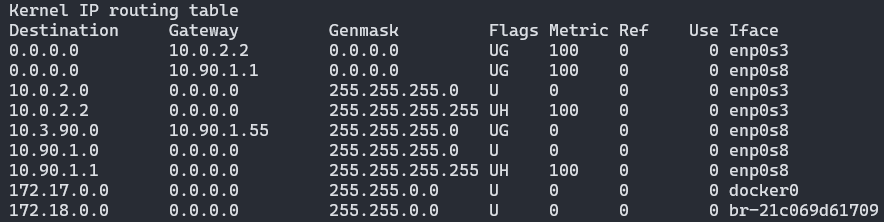
**Route table for Client2:**



**Temporary route adding**

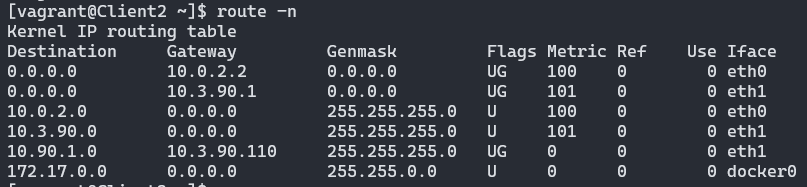
**Client1:**



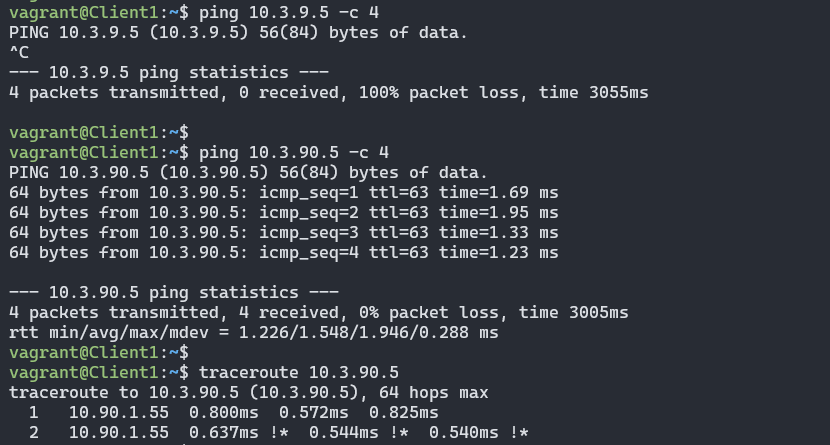


**Client2:**

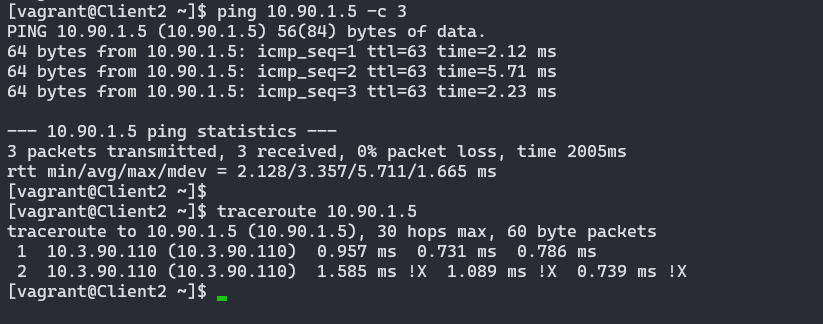




**Ping and traceroute from Client1 (Int1-10.90.1.5) to Client2 (Int1 - 10.3.90.5):**



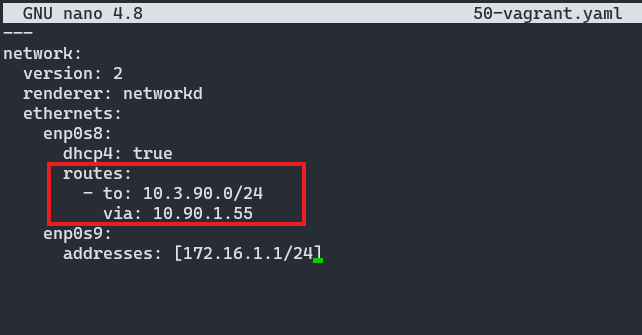
**Ping and traceroute from Client2 (Int1 - 10.3.90.5) to Client1 (Int1 - 10.90.1.5):**

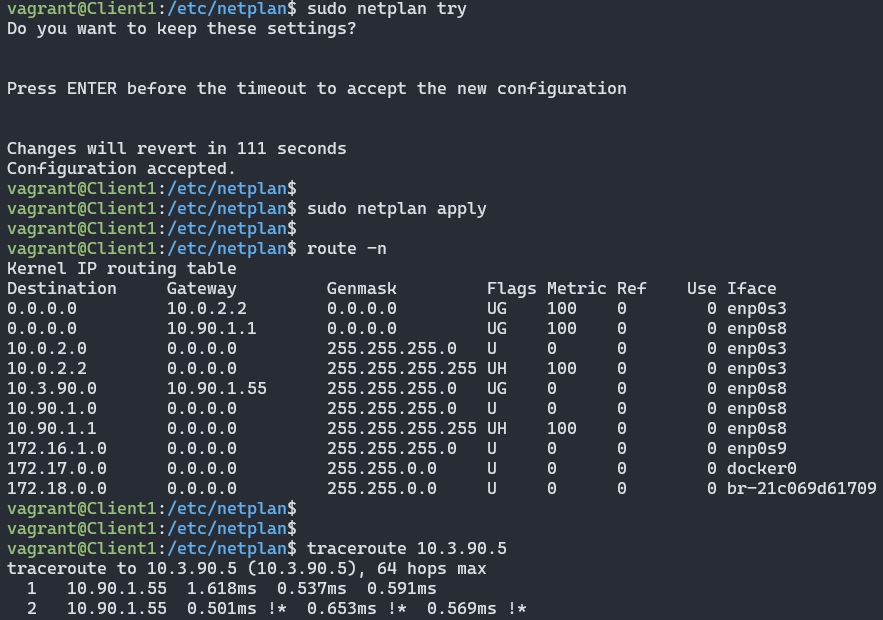


**Awesome, we have successfully added a route from one Linux computer to another!**

**Permanent route adding**

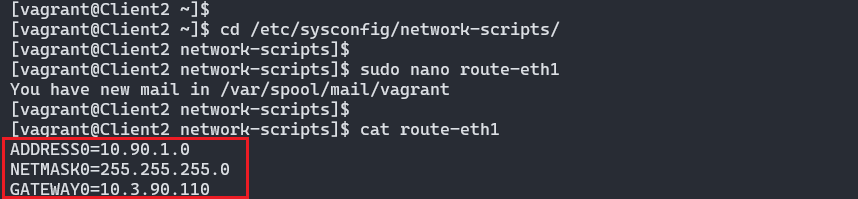
**For Ubuntu 18.04 and later using Netplan:**



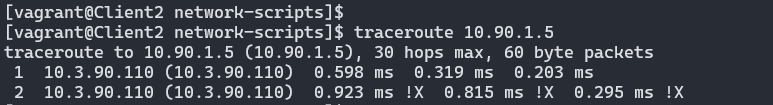


**For CentOS 7:**

On **RHEL** and **CentOS** distributions, you need to create a file named “**route-**” in the **“/etc/sysconfig/network-scripts”** folder:





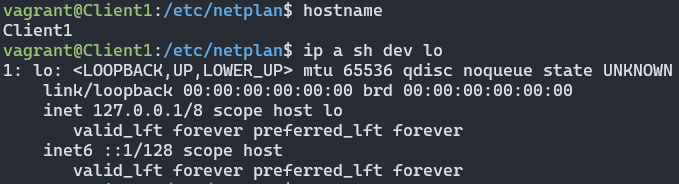


1. На віртуальному інтерфейсу lo Client1 призначити дві ІР адреси за таким правилом: 172.19.D+10.1/24 та 172.19.D+20.1/24. Налаштувати маршрутизацію таким чином, щоб трафік з Client2 до 172.19.D+10.1 проходив через Server1, а до 172.19.D+20.1 через Net4. Для перевірки використати traceroute.

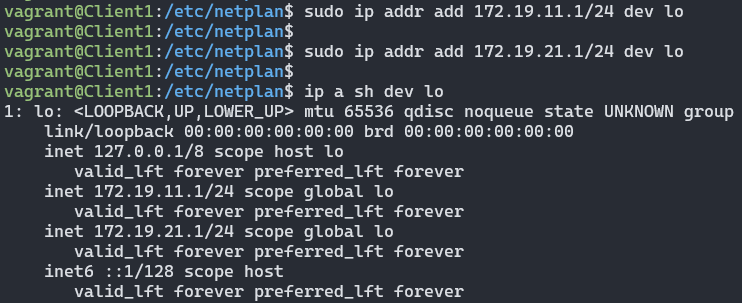
**Client1 (lo)** – 172.19.11.1/24

**Client1 (lo)** – 172.19.21.1/24

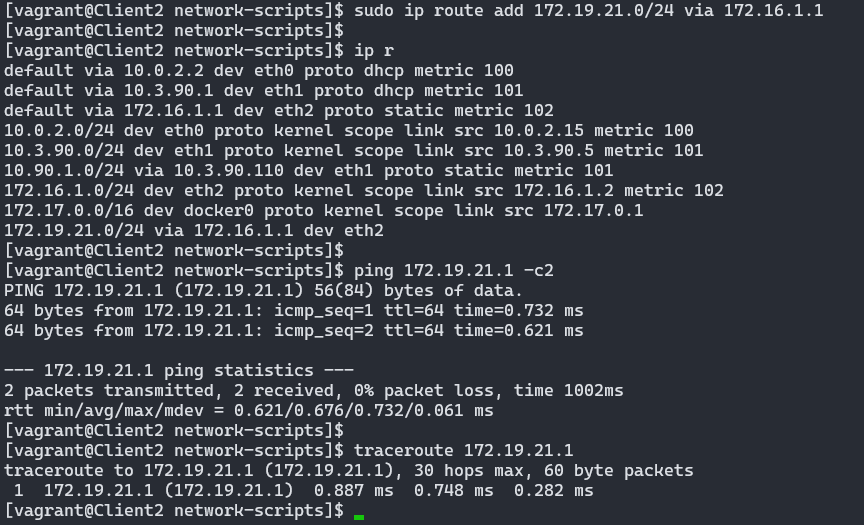
**Output addresses of the lo interface:**



**Add addresses to the lo interface:**



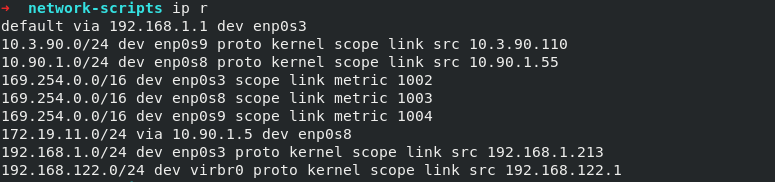
**Add route from Client2 to 172.19.21.1 via Net4(172.16.1.0/24) and check:**



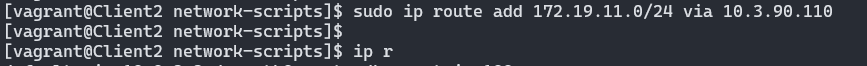
**Add routes from Client2 to 172.19.11.1 via Server1 and check:**

**On Server1**

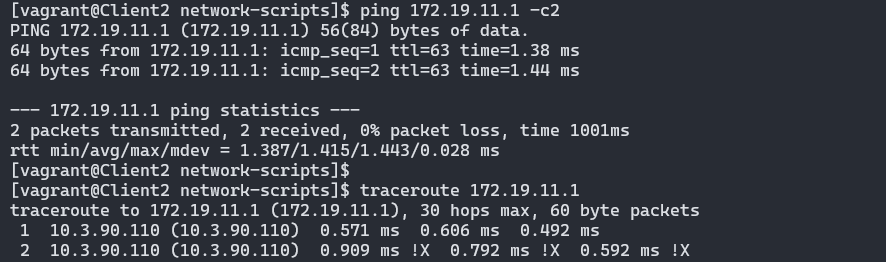




**On Client2**

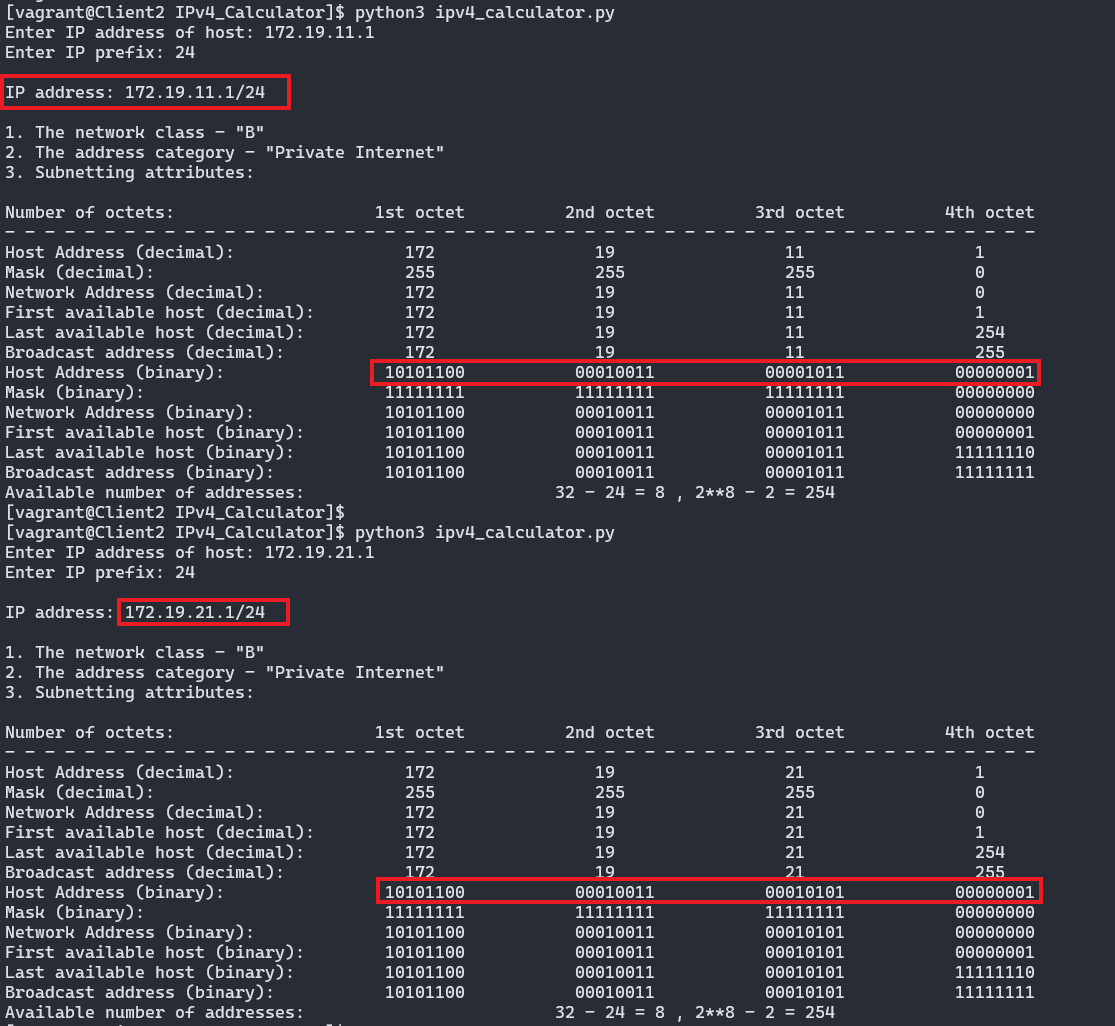






1. Розрахувати спільну адресу та маску (summarizing) адрес 172.19.D+10.1 та 172.19.D+20.1, при чому префікс має бути максимально можливим. Видалити маршрути, встановлені на попередньому кроці та замінити їх об’єднаним маршрутом, якій має проходити через Server1.

**Run script ipv4\_calculator.py of my development to known hosts addresses in binary:**



**Routes that can be summarized:**

**172.19.11.1** – 10101100.00010011.00001011.00000001 - first 19 bits are the same

**172.19.21.1** – 10101100.00010011.00010101.00000001

**Summarized route into one route:**

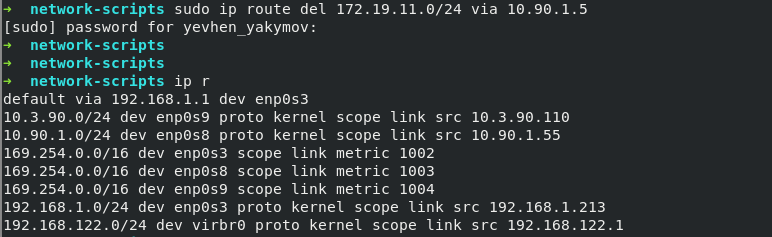
**172.19.0.0** – 10101100.00010011.00000000.00000000

**255.255.224.0** – 11111111.11111111.11100000.00000000

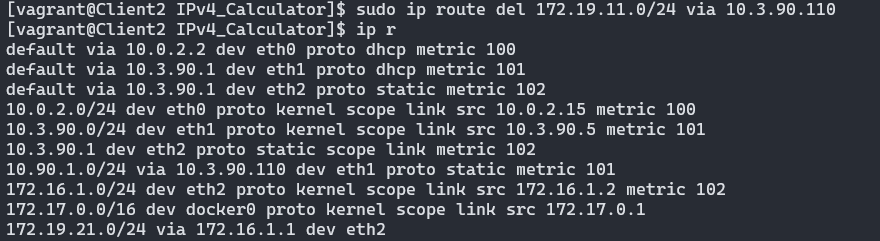
**/19**

**Delete routes dedicated in previous step:**

**On Server1**

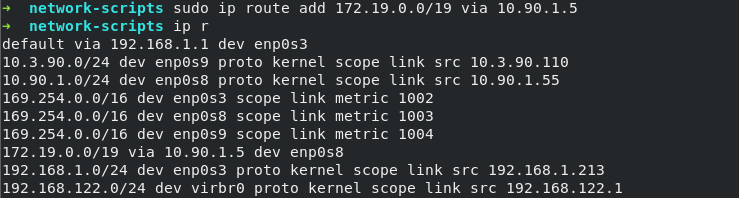


**On Client2**



**Add summary route from Client2 to 172.19.11.1 and .21.1 via Server1 and check:**

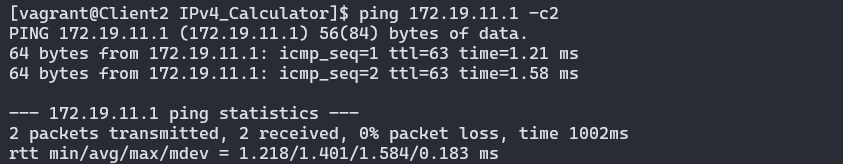
**On Server1**



**On Client2 and ping 172.19.11.1**



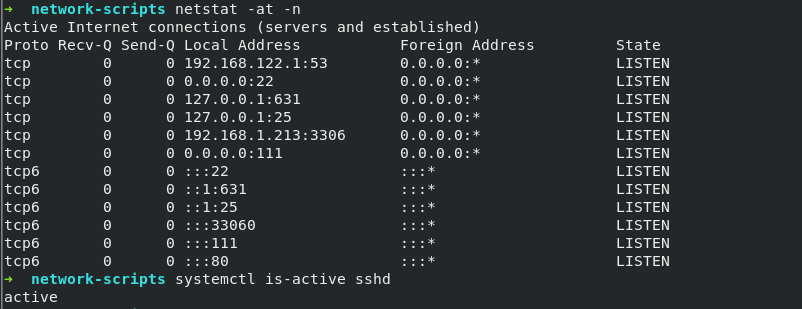




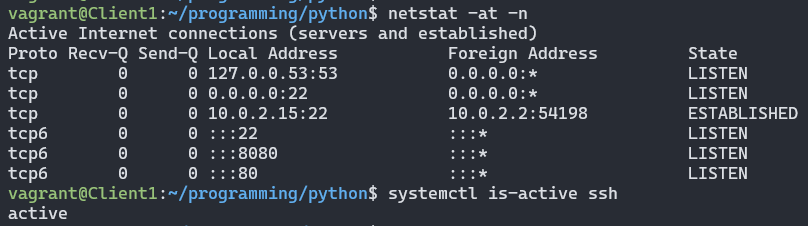
1. Налаштувати SSH сервіс таким чином, щоб Client1 та Client2 могли підключатись до Server1 та один до одного.

**Verify sshd service installation and active ports:**

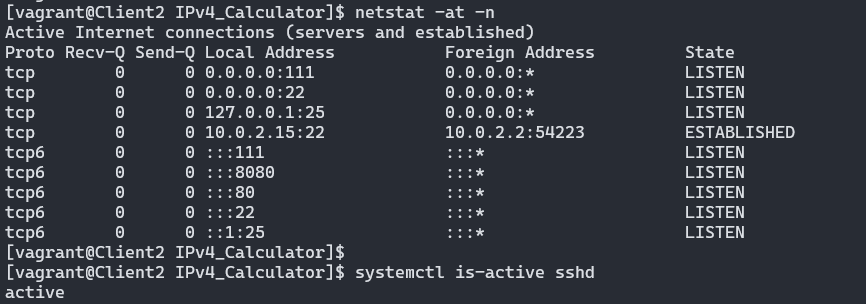
**On Server1**



**On Client1**

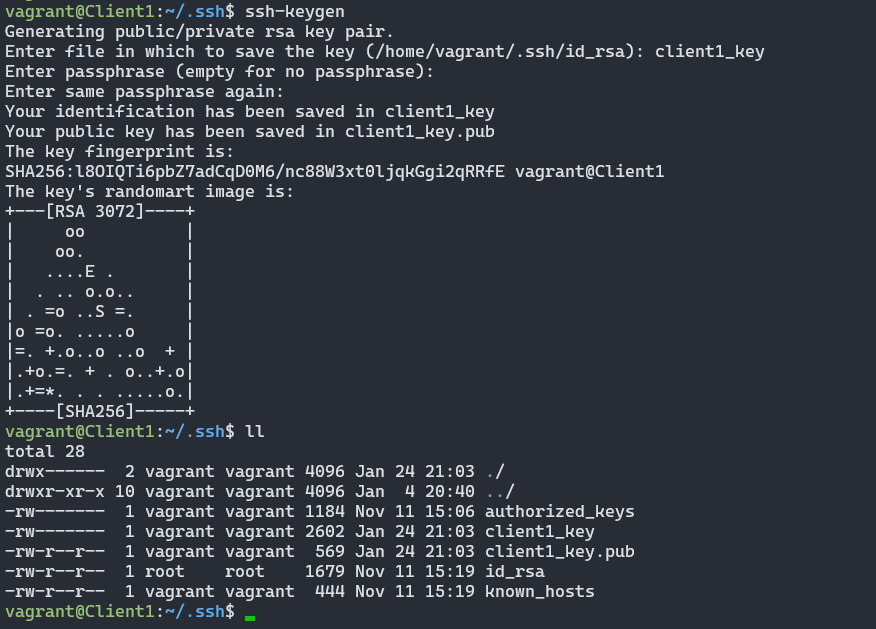


**On Client2**

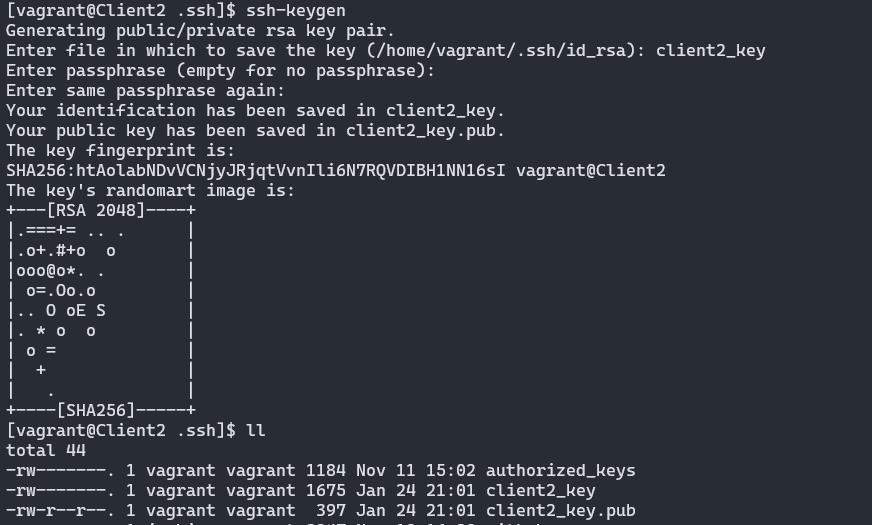


**Generates a pair of keys, of which one key is secret and the other public:**

**On Client1**

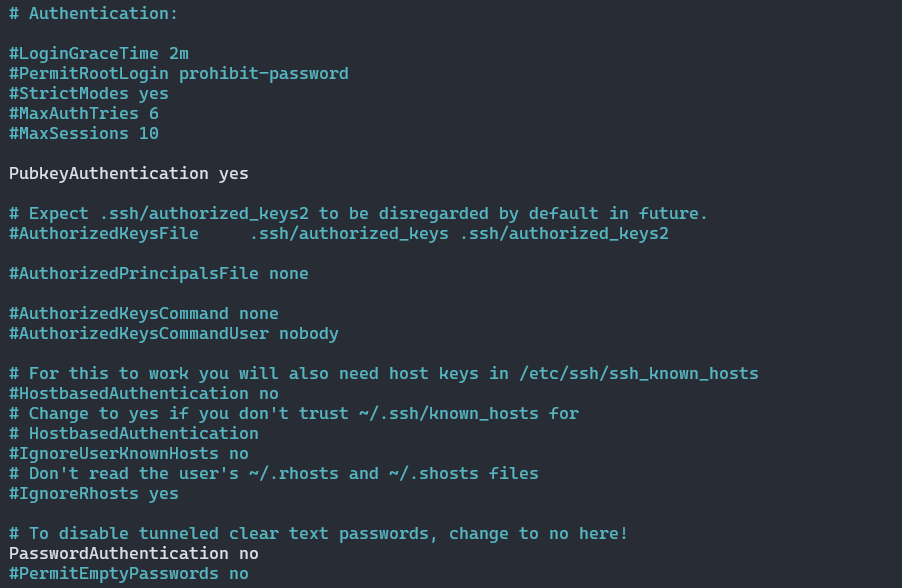


**On Client2**

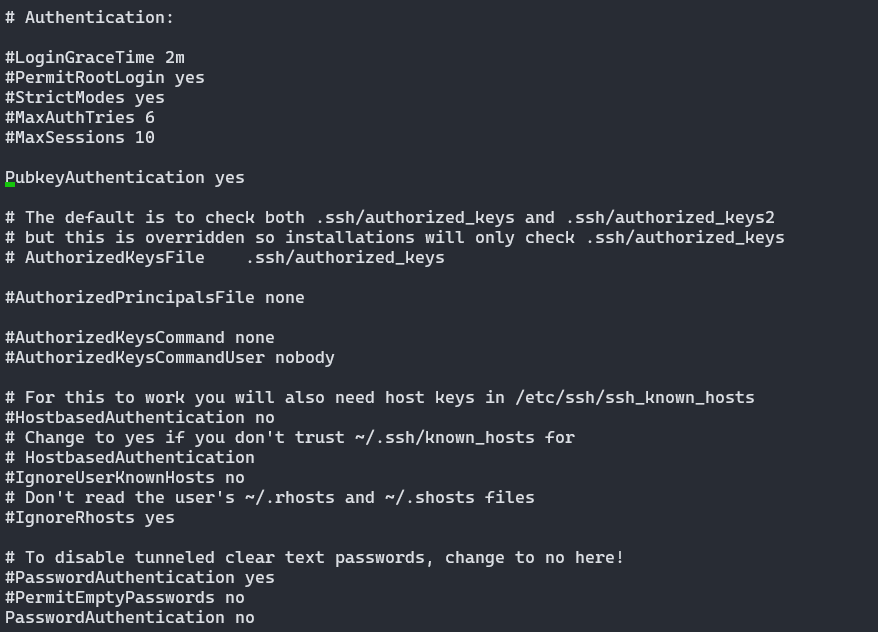


**Configure /etc/ssh/sshd\_config file:**

**Client1**

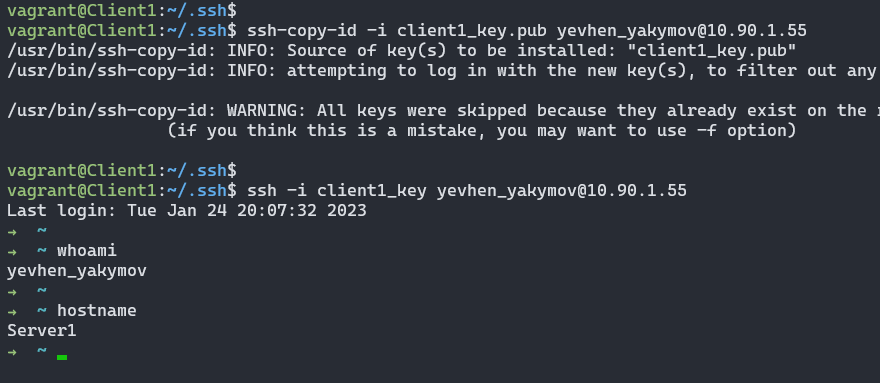


**Client2**

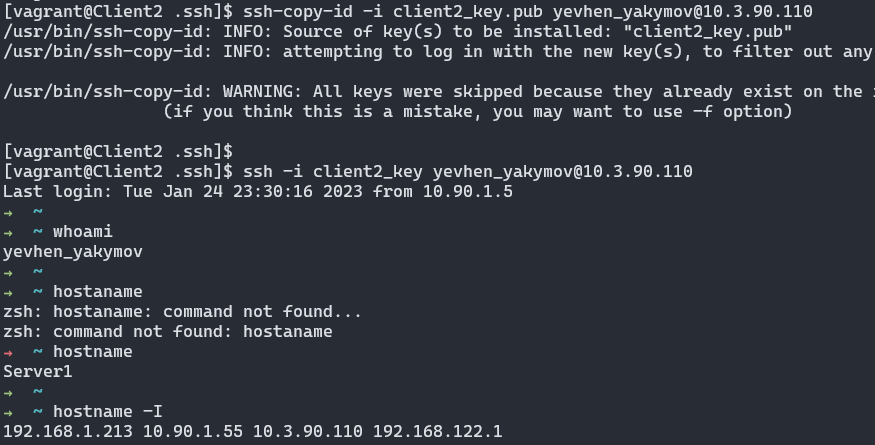


**Сopy public key to Server1 and establish ssh connection:**

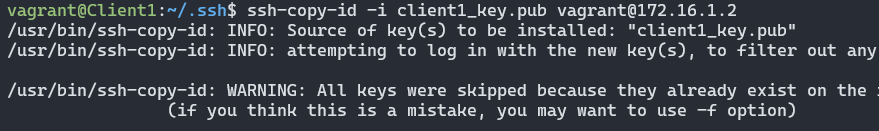
**From Client1 to Server1**

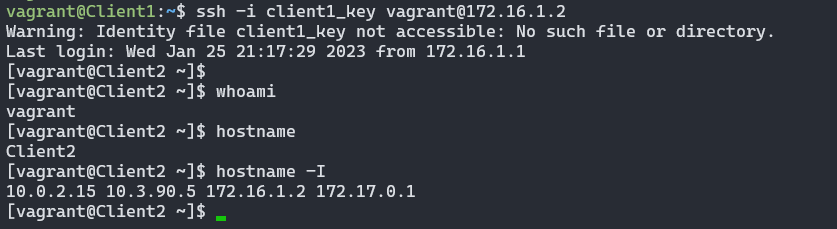


**From Client2 to Server1**



**From Client1 to Client2**

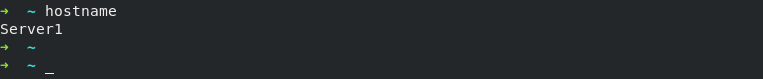


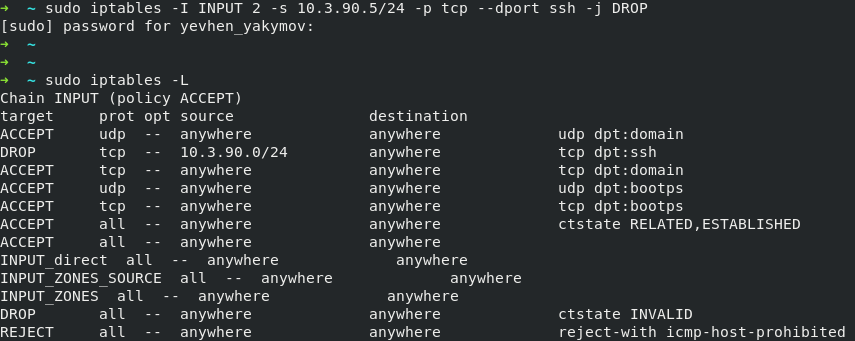


1. Налаштуйте на Server1 firewall таким чином:

**• Дозволено підключатись через SSH з Client\_1 та заборонено з Client\_2:**

**Deny ssh connection to Server1 from Client2**

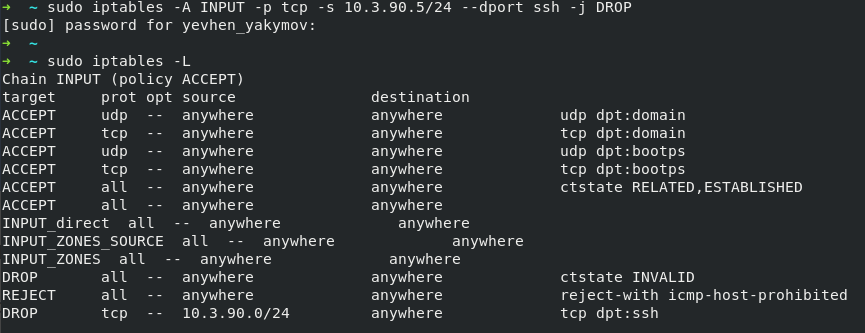




**Check ssh connection to Server1 from Client2**

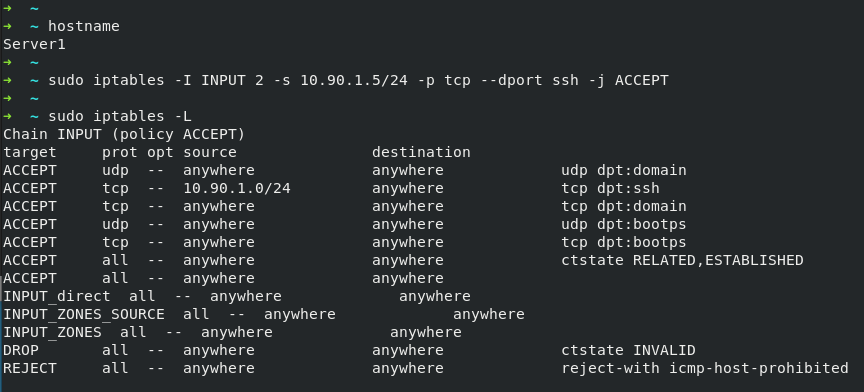


**(more different options)**

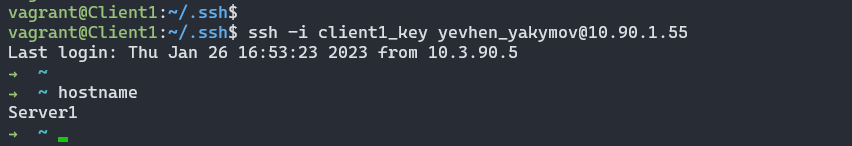




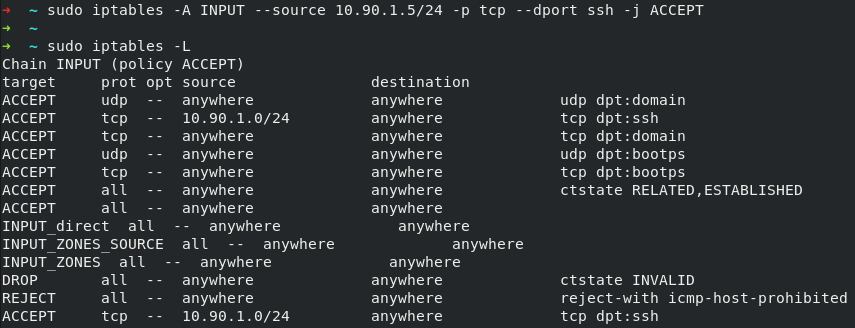
**Allow ssh connection to Server1 from Client1**

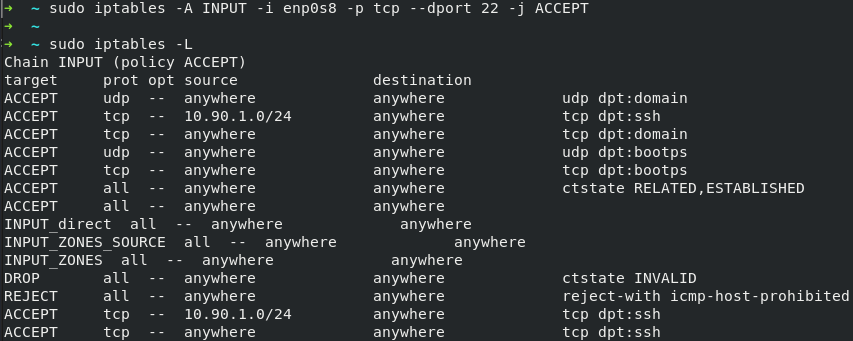


**Check ssh connection to Server1 from Client1**



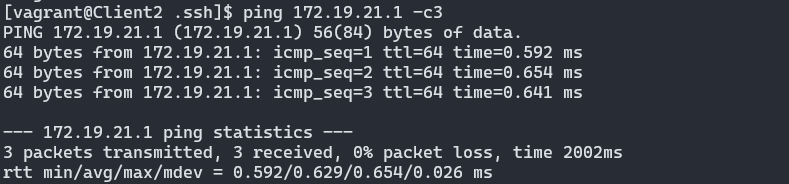
**(more different options)**

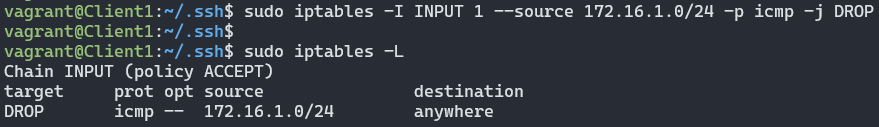


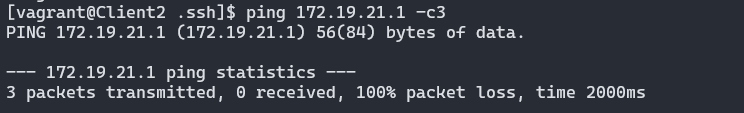


**• З Client\_2 на 172.19.D+10.1 ping проходив, а на 172.19.D+20.1 не проходив**

**Deny icmp to Client1(172.19.21.1) from Client2**







**Allow icmp to Client1(172.19.21.1) from Client2**

