

Bartosz Bieniek

gr. 7, st. 1, sem. 3, Informatyka RMS

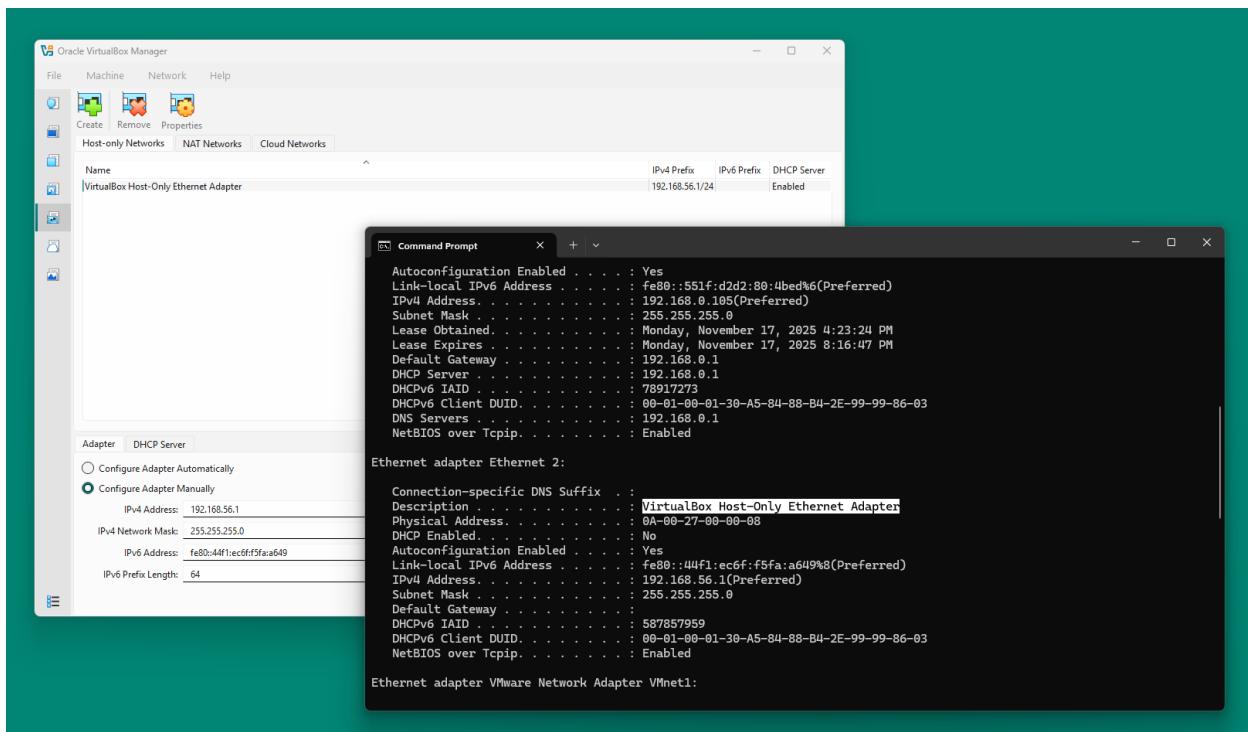
## Przygotowanie środowiska.

Do realizacji zadania wykorzystałem dwie maszyny wirtualne z systemami Debian GNU/Linux oraz Fedora GNU/Linux, z interfejsami sieciowymi skonfigurowanymi zgodnie z poniższą tabelą.

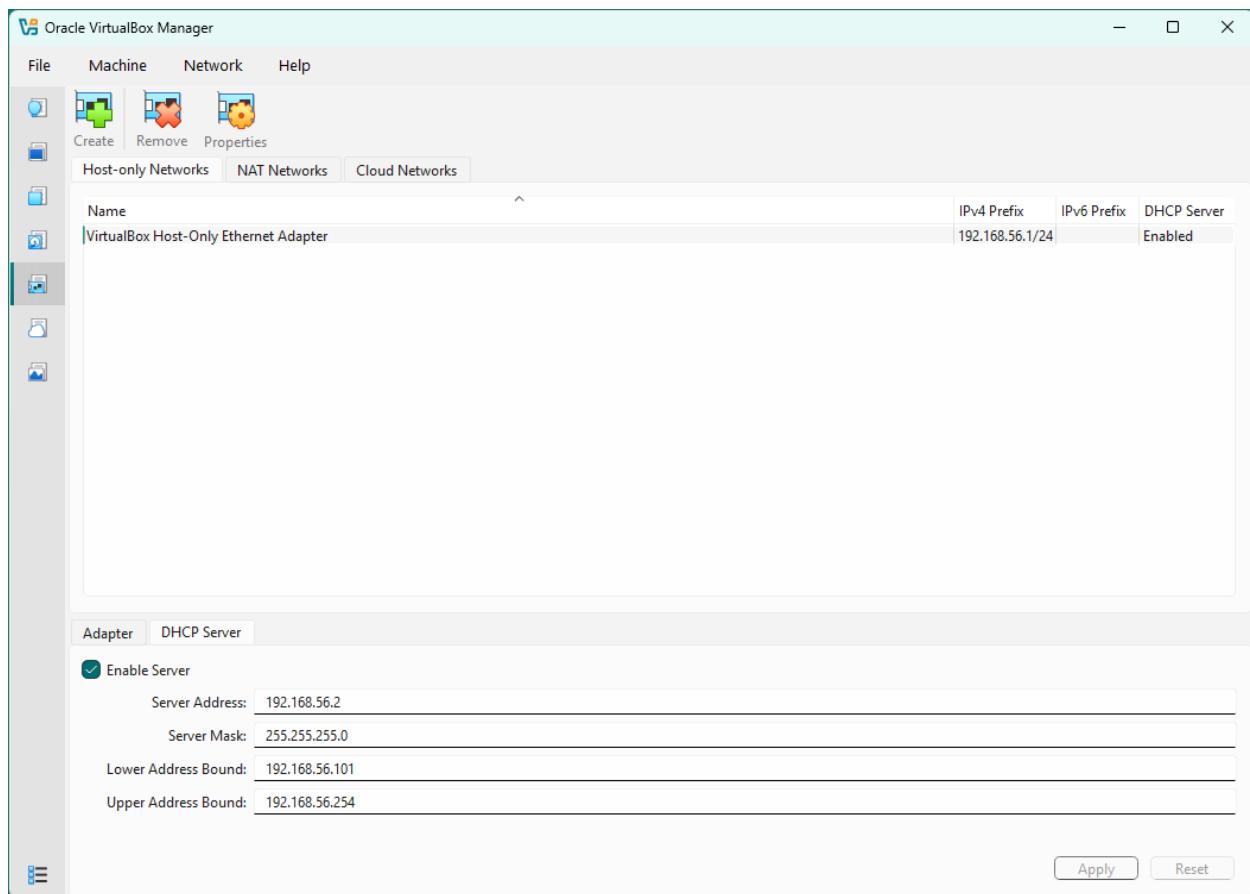
Interfejs	Debian	Fedora
1	NAT	NAT
2	Internal network (intnet)	Internal network (intnet)
3	Host-Only Adapter (VirtualBox Host-Only Ethernet Adapter)	

Tabela 1 Konfiguracja interfejsów sieciowych maszyn wirtualnych.

Na początku utworzyłem nową sieć typu host-only zgodnie z instrukcją laboratorium.

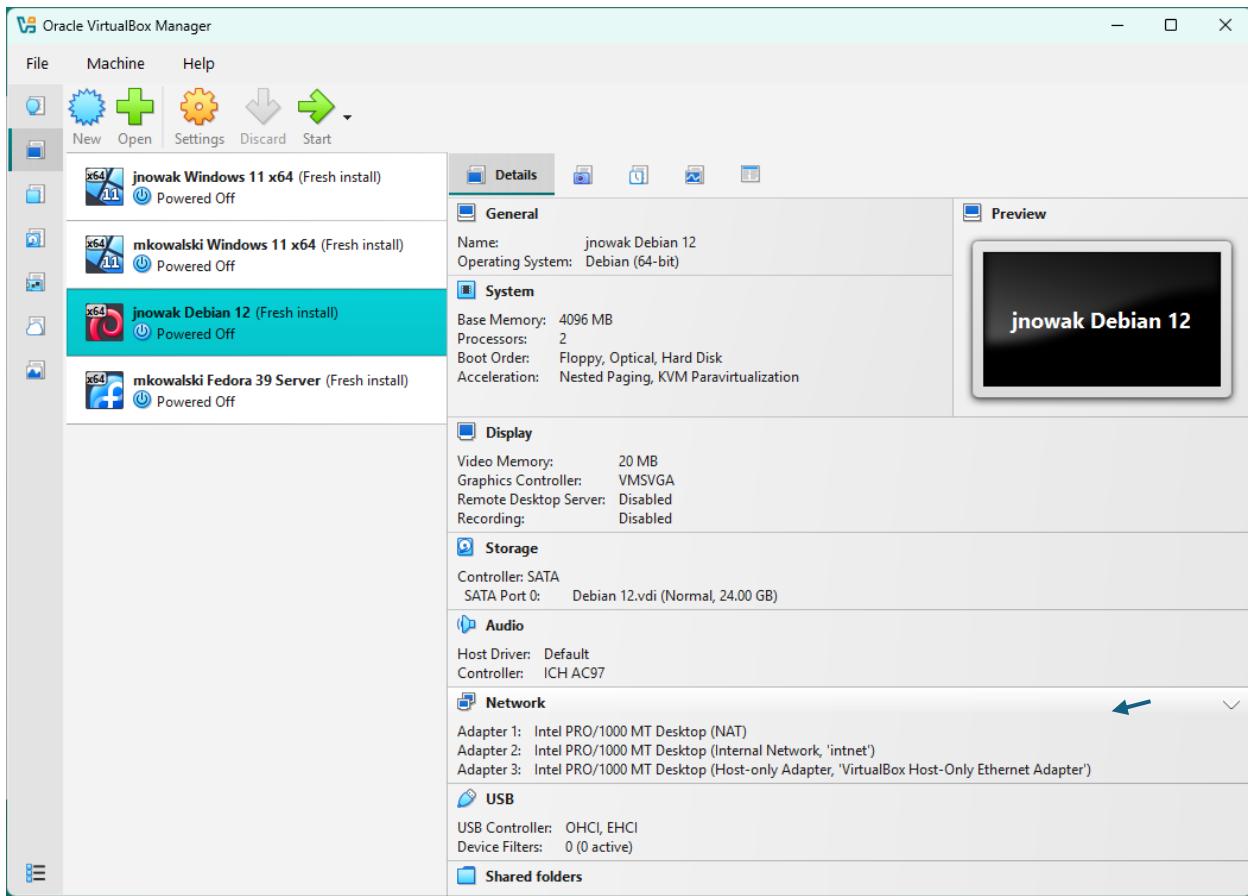


Zrzut ekranu 1 Konfiguracja sieci typu host-only. Wyświetlenie utworzonego na maszynie hosta adaptera sieciowego poleciением ipconfig /all.

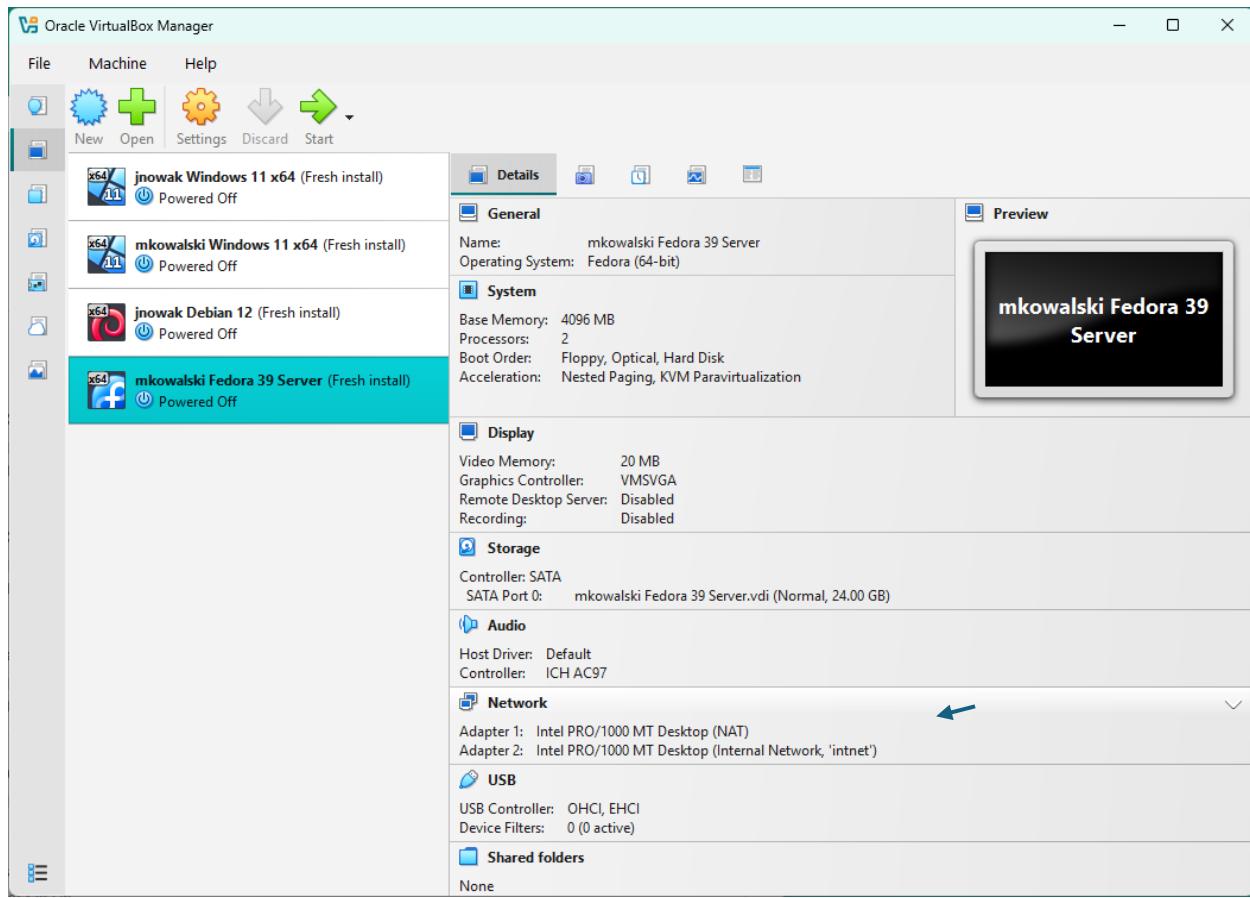


Zrzut ekranu 2 Konfiguracja DHCP w utworzonej sieci typu host-only.

Następnie, skonfigurowałem w maszynach wirtualnych adaptery sieciowe zgodnie z powyższą tabelą.



Zrzut ekranu 3 Podsumowanie konfiguracji maszyny wirtualnej z systemem Debian 12.

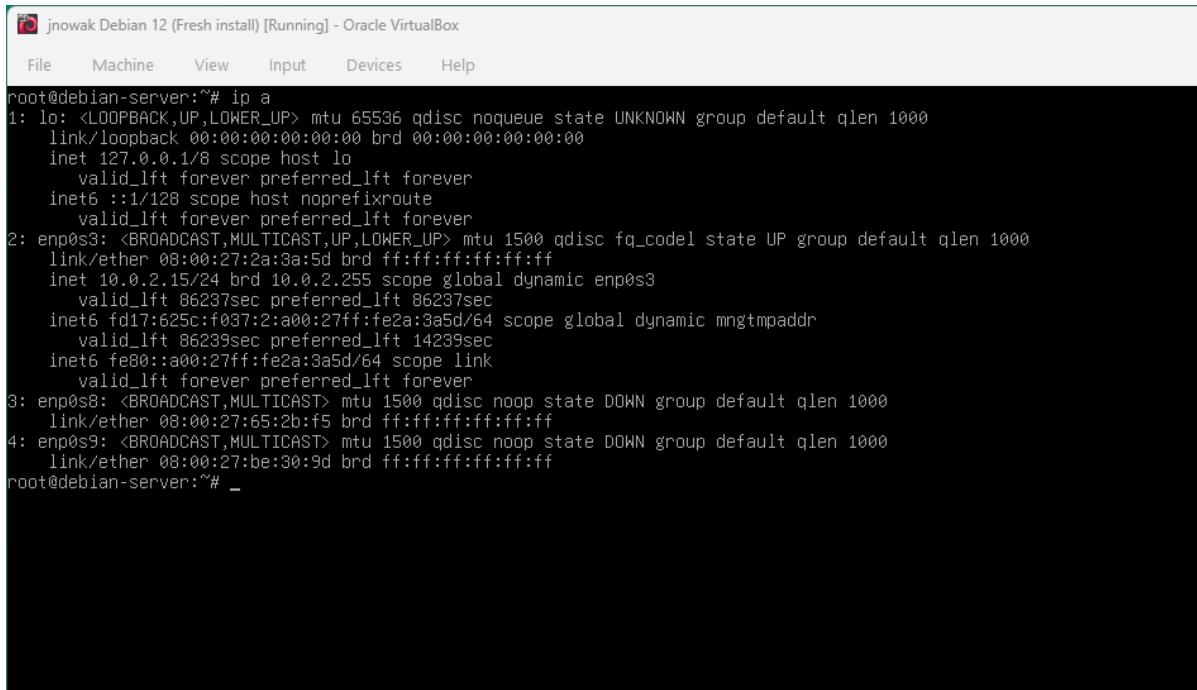


Zrzut ekranu 4 Podsumowanie konfiguracji maszyny wirtualnej z systemem Fedora 39 Server.

## Zadanie 1. Konfiguracja ustawień sieciowych w serwerze Debian 12.

Konfiguracji sieciowej pod systemem Debian 12 dokonuje się poprzez edycję pliku `/etc/network/interfaces`. Aby wprowadzić zmiany, wybrany interfejs wyłącza się i włącza poleceniami `ifdown <interfejs>` oraz `ifup <interfejs>` lub restartuje się całą usługę `networking` polecienniem `systemctl restart networking`.

Przed przystąpieniem do zadania, możemy sprawdzić stan adapterów sieciowych oraz przydzielone adresy polecienniem `ip a (od address)`.



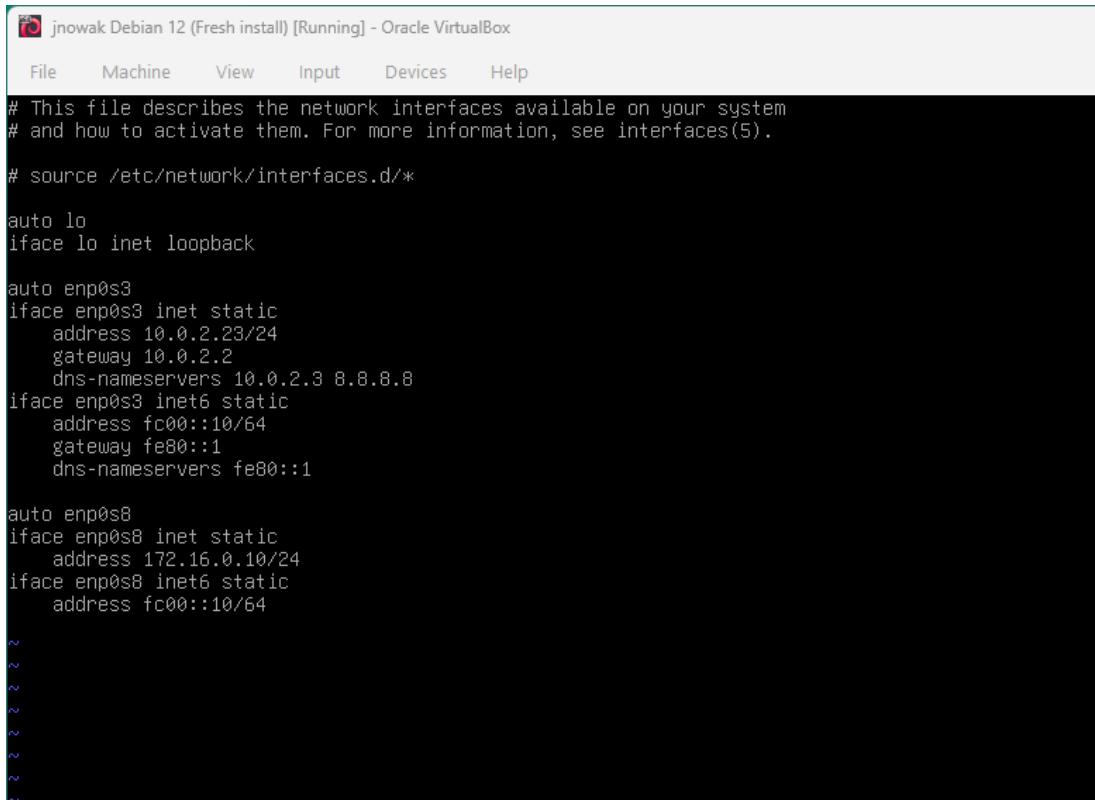
```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@debian-server:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 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 08:00:27:2a:3a:5d brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
            valid_lft 86237sec preferred_lft 14239sec
        inet6 fd17:625c:f037:2:a00:27ff:fe2a:3a5d/64 scope global dynamic mngtmpaddr
            valid_lft 86239sec preferred_lft 14239sec
        inet6 fe80::a00:27ff:fe2a:3a5d/64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:65:2b:f5 brd ff:ff:ff:ff:ff:ff
4: enp0s9: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:be:30:9d brd ff:ff:ff:ff:ff:ff
root@debian-server:~#
```

Zrzut ekranu 5 Początkowa konfiguracja sieciowa systemu Debian 12.

W ramach sieci NAT, komputerowi został już przydzielony adres na interfejsie drugim (`enp0s3`). Przekonfigurujmy teraz ten i kolejny adapter, zgodnie z poniższymi danymi.

enp0s3 (NAT)	enp0s8 (Internal network – intnet)
Adres IP: 10.0.2.23	Adres IP: 172.16.0.10
Maska podsieci: 255.255.255.0	Maska podsieci: 255.255.255.0
Brama: 10.0.2.2	
DNS: 10.0.2.3 i 8.8.8.8	
Adres IPv6: FC00::10/64	Adres IPv6: FD00::10/64
Brama IPv6: FE80::1	
DNS IPv6: FE80::1	

Tabela 2 Dane adresowe dla maszyny wirtualnej z systemem Debian.



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.0.2.23/24
    gateway 10.0.2.2
    dns-nameservers 10.0.2.3 8.8.8.8
iface enp0s3 inet6 static
    address fc00::10/64
    gateway fe80::1
    dns-nameservers fe80::1

auto enp0s8
iface enp0s8 inet static
    address 172.16.0.10/24
iface enp0s8 inet6 static
    address fc00::10/64

~
```

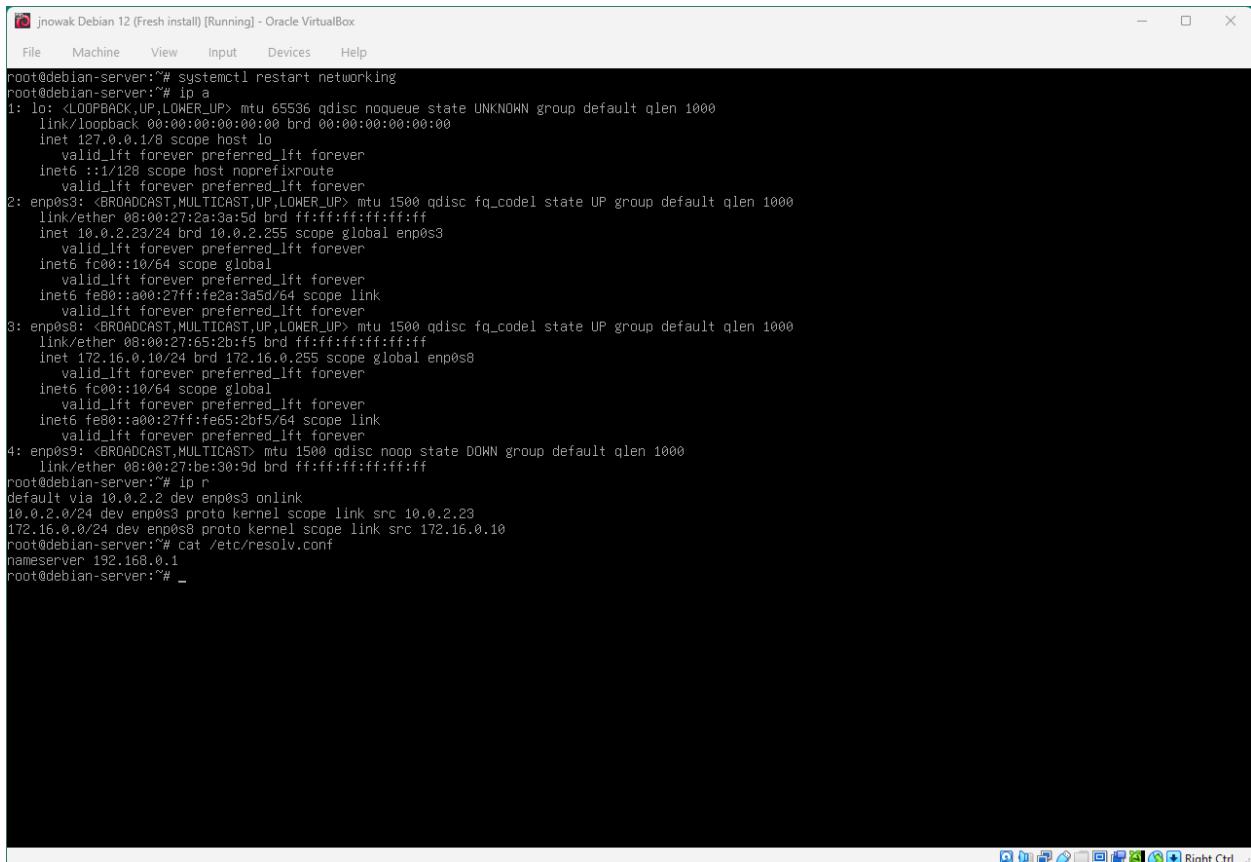
Zrzut ekranu 6 Konfiguracja interfejsów sieciowych maszyny wirtualnej z systemem Debian 12, zgodnie z danymi podanymi w Tabeli 2.

Realizując kolejne zadania zauważylem, że przypisałem na interfejsie `enp0s8` adres `fc00::10/64`, zamiast `fd::0/64`. Poprawkę wprowadziłem w ramach zadania 6 (bez zrzutu ekranu).

Do aktywacji interfejsu przy starcie systemu służy wpis `auto <interfejs>`. Można się także spotkać z dyrektywą `allow-hotplug <interfejs>`, która umożliwia konfigurację adapterów, które mogą być do komputera wpinane/wypinane w trakcie działania (na przykład takich na USB). Bloki zaczynające się od wyrażenia `iface <interfejs> <inet/inet6> <static/auto/dhcp/manual/loopback>` to bloki konfiguracyjne dla wskazanych interfejsów sieciowych, tak zwane „stanca” (ang. „stanzas”). Słowa `inet` i `inet6` odnoszą się do rodzaju

połączenia, którego ta konfiguracja się dotyczy – IPv4 i IPv6, natomiast *static*, *auto*, *dhcp*, *manual* i *loopback* umożliwiają na kolejno – skonfigurowanie danych adresowych statycznie, z wykorzystaniem mechanizmu SLAAC (tylko IPv6), z wykorzystaniem serwera DHCP, włączenie nieskonfigurowanego interfejsu, stworzenie pętli zwrotnej. Za pomocą poleceń *address*, *netmask*, *gateway* oraz *dns-nameservers* można skonfigurować, zgodnie z tym co wskazują nazwy – adres, maskę sieciową, bramę domyślną oraz serwery DNS. Dwie pierwsze z nich można zapisać w postaci skrótowej – na przykład *address 192.168.0.2/24*.

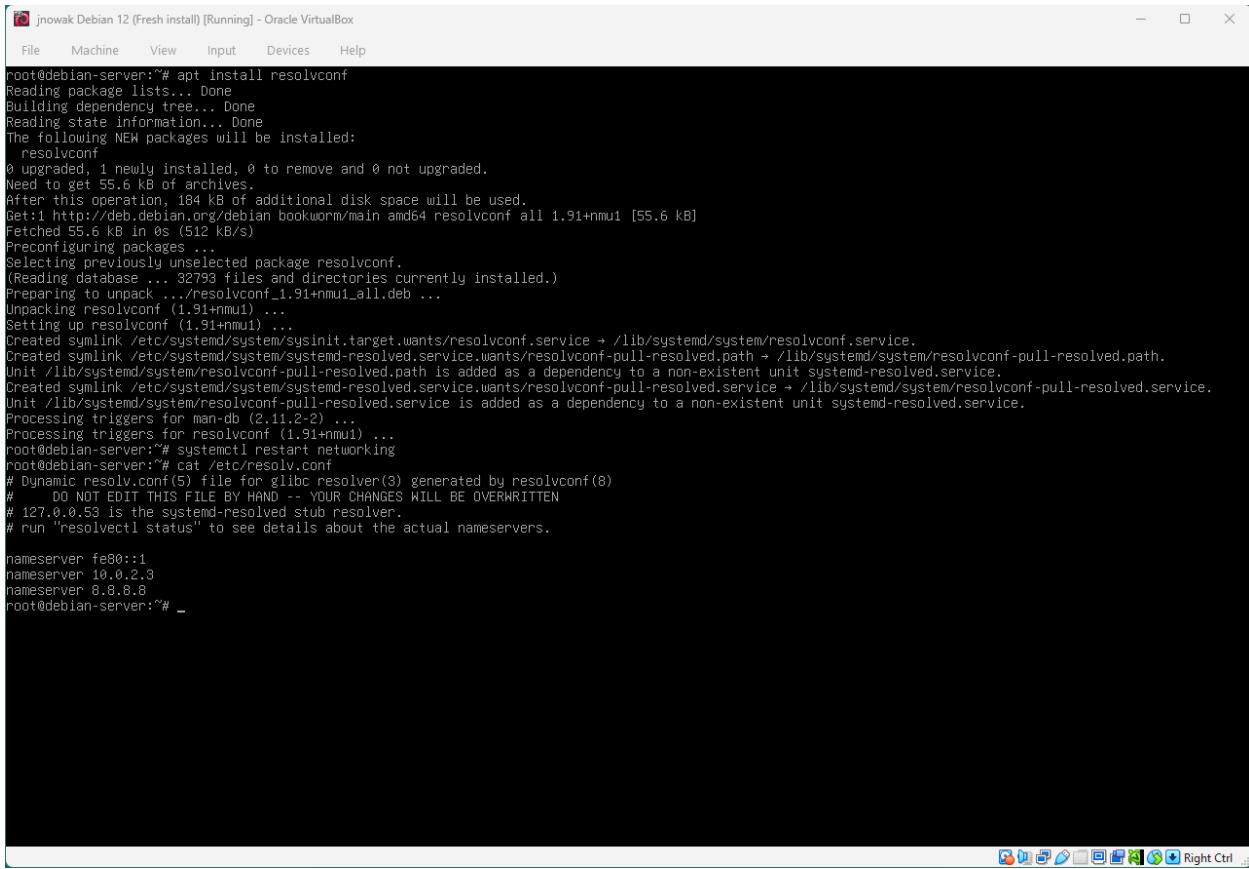
Aby aktywować nową konfigurację najłatwiej będzie zrestartować usługę *networking*. Dane adresowe sprawdzimy następnie polecienniem *ip a*, bramę domyślną przez *ip r*, a serwery DNS przez sprawdzenie zawartości pliku */etc/resolv.conf*.



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@debian-server:~# systemctl restart networking
root@debian-server:~# 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 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 08:00:27:2a:3a:5d brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.23/24 brd 10.0.2.255 scope global enp0s3
            valid_lft forever preferred_lft forever
            inet6 fe00::10/64 scope global
                valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe2a:3a5d/64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:65:2b:f5 brd ff:ff:ff:ff:ff:ff
        inet 172.16.0.10/24 brd 172.16.0.255 scope global enp0s8
            valid_lft forever preferred_lft forever
            inet6 fe00::10/64 scope global
                valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe65:2bf5/64 scope link
            valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:be:30:9d brd ff:ff:ff:ff:ff:ff
root@debian-server:~# ip r
default via 10.0.2.2 dev enp0s3 onlink
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.23
172.16.0.0/24 dev enp0s8 proto kernel scope link src 172.16.0.10
root@debian-server:~# cat /etc/resolv.conf
nameserver 192.168.0.1
root@debian-server:~# _
```

Zrzut ekranu 7 Restart usługi *networking*. Sprawdzenie poprawności konfiguracji sieciowej.

Możemy zauważyć, że zawartość pliku */etc/resolv.conf* nie pokrywa się z wprowadzoną konfiguracją. Wynika to z faktu, że w systemie nie ma usługi, która automatycznie przepisałaby dane z */etc/network/interfaces* do wskazanego pliku. Wobec tego możemy albo wprowadzić konfigurację ręcznie, albo zainstalować pakiet *resolvconf*, który został do tego celu stworzony.



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@debian-server:~# apt install resolvconf
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  resolvconf
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 55.6 kB of archives.
After this operation, 184 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 resolvconf all 1.91+nmu1 [55.6 kB]
Fetched 55.6 kB in 0s (512 kB/s)
Preconfiguring packages ...
Selecting previously unselected package resolvconf.
(Reading database ... 32793 files and directories currently installed.)
Preparing to unpack .../resolvconf_1.91+nmu1_all.deb ...
Unpacking resolvconf (1.91+nmu1) ...
Setting up resolvconf (1.91+nmu1) ...
Created symlink /etc/systemd/system/sysinit.target.wants/resolvconf.service → /lib/systemd/system/resolvconf.service.
Created symlink /etc/systemd/system/systemd-resolved.service.wants/resolvconf-pull-resolved.path → /lib/systemd/system/resolvconf-pull-resolved.path.
Unit /lib/systemd/system/resolvconf-pull-resolved.path is added as a dependency to a non-existent unit systemd-resolved.service.
Created symlink /etc/systemd/system/resolvconf-pull-resolved.service.wants/resolvconf-pull-resolved.service → /lib/systemd/system/resolvconf-pull-resolved.service.
Unit /lib/systemd/system/resolvconf-pull-resolved.service is added as a dependency to a non-existent unit systemd-resolved.service.
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for resolvconf (1.91+nmu1) ...
root@debian-server:~# systemctl restart networking
root@debian-server:~# cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "resolvectl status" to see details about the actual nameservers.

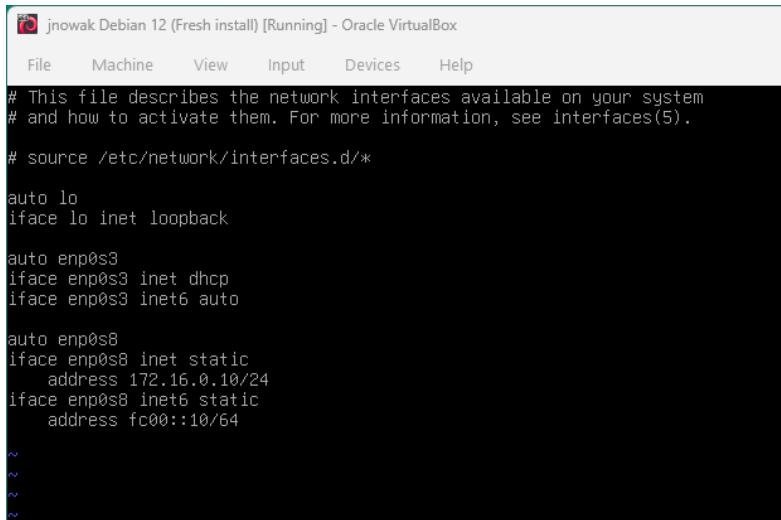
nameserver fe00::1
nameserver 10.0.2.3
nameserver 8.8.8
root@debian-server:~# _
```

Zrzut ekranu 8 Instalacja pakietu resolvconf i ponowne wypisanie zawartości pliku /etc/resolv.conf.

Jak widać, po zainstalowaniu wskazanego oprogramowania i ponownym uruchomieniu usługi *networking*, w pliku pojawiły się zgodnie z konfiguracją dane.

## Zadanie 2. Konfiguracja ustawień TCP/IP z wykorzystaniem serwera DHCP.

Przypuśćmy, że nastąpiła potrzeba przekonfigurowania drugiego interfejsu (enp0s3, 10.0.2.23), tak aby pobierał dane z serwera DHCP. W tym celu możemy wykasować zawartość bloków konfiguracyjnych pod niego podpiętych i przepisać dyrektywę *iface*, tak aby dane były pobierane z serwera DHCP – *iface enp0s3 inet dhcp* oraz *iface enp0s3 inet6 auto*. Warto zwrócić uwagę, że w przypadku trybu IPv6 została ustawiona wartość *auto*, ponieważ tryb adaptera *NAT* w VirtualBox nie umożliwia konfiguracji z wykorzystaniem *DHCP6* – zamiast niego wykorzystujemy mechanizm *SLAAC*.



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

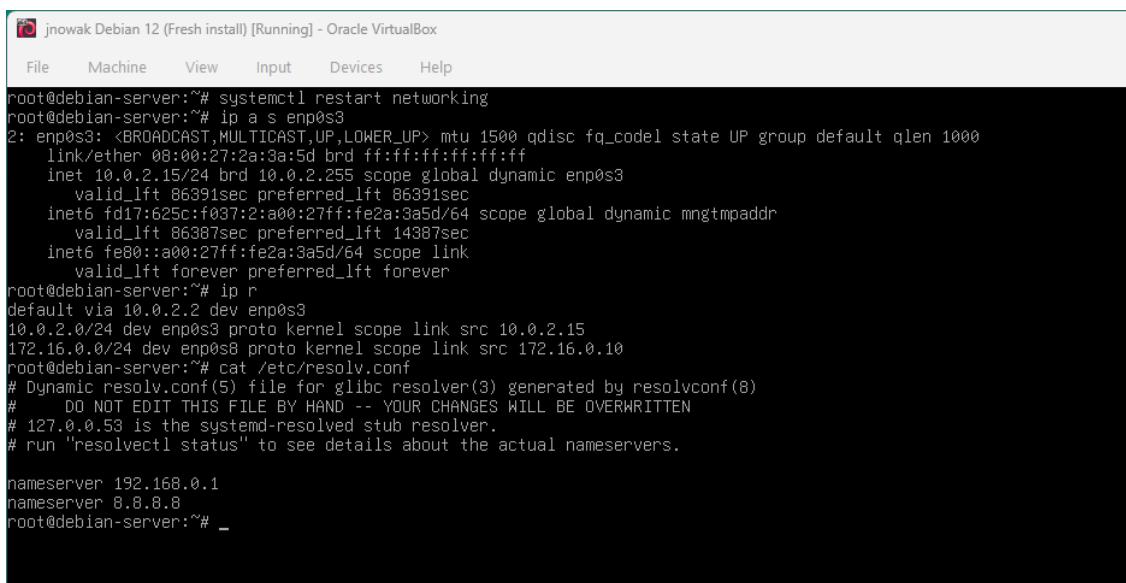
auto enp0s3
iface enp0s3 inet dhcp
iface enp0s3 inet6 auto

auto enp0s8
iface enp0s8 inet static
    address 172.16.0.10/24
iface enp0s8 inet6 static
    address fc00::10/64

~
~
~
```

Zrzut ekranu 9 Konfiguracja automatycznego pobierania danych adresowych w ramach DHCP na interfejsie enp0s3.

Po restarcie usługi networking możemy sprawdzić jakie dane zostały nadane.



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

root@debian-server:~# systemctl restart networking
root@debian-server:~# ip a s enp0s3
2: enp0s3: <BR0ADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:00:27:2a:3a:5d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86391sec preferred_lft 86391sec
    inet6 fd17:625c:037:2:a00:27ff:fe2a:3a5d/64 scope global dynamic mngtmpaddr
        valid_lft 14387sec preferred_lft 14387sec
    inet6 fe80::a00:27ff:fe2a:3a5d/64 scope link
        valid_lft forever preferred_lft forever
root@debian-server:~# ip r
default via 10.0.2.2 dev enp0s3
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15
172.16.0.0/24 dev enp0s8 proto kernel scope link src 172.16.0.10
root@debian-server:~# cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "resolvectl status" to see details about the actual nameservers.

nameserver 192.168.0.1
nameserver 8.8.8.8
root@debian-server:~# _
```

Zrzut ekranu 10 Dane adresowe uzyskane w ramach usługi DHCP.

W ramach usługi DHCP (dla IPv4) i mechanizmu SLAAC (dla IPv6) komputer uzyskał następujące dane adresowe.

**enp0s3**

Adres IP: 10.0.2.15 (wewnętrzny)

Maska podsieci: 255.255.255.0

Brama: 10.0.2.2

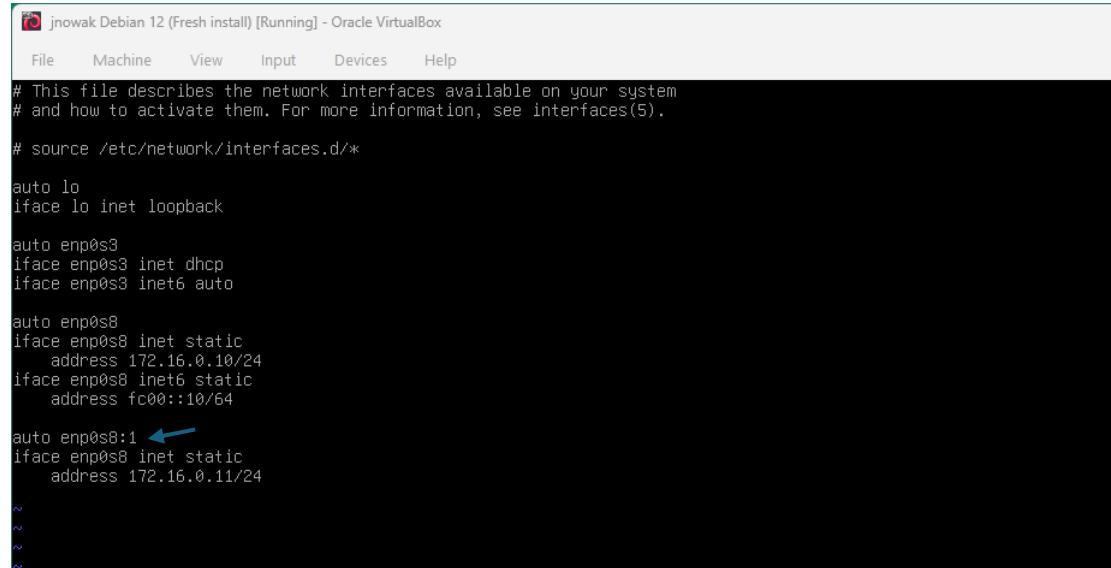
DNS: 192.168.0.1

Adres IPv6: FD17:625C:F037:2:A00:27FF:FE2A:3A5D/64 (unikalny lokalny)

*Tabela 3 Dane adresowe przydzielone na interfejsie enp0s3 serwera Debian 12 z wykorzystaniem DHCP i SLAAC.*

### Zadanie 3. Wirtualne interfejsy sieciowe (aliasy).

W systemie Debian 12 istnieją dwa główne sposoby na przypisanie wielu adresów do jednego interfejsu sieciowego – można w danej stancji dodać jednocześnie kilka wpisów *address* lub utworzyć kolejne bloki konfiguracyjne dla danego interfejsu z dopiskiem *:numer\_aliasu*, na przykład *enp0s8:1*. Ja wykorzystam tę drugą metodę.



```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet dhcp
iface enp0s3 inet6 auto

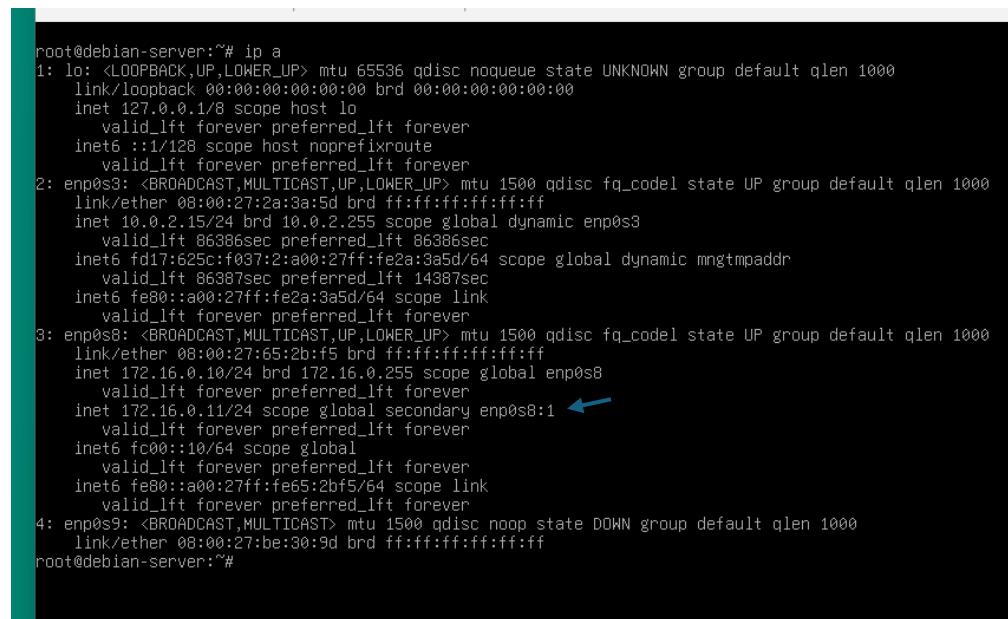
auto enp0s8
iface enp0s8 inet static
    address 172.16.0.10/24
iface enp0s8 inet6 static
    address fc00::10:64

auto enp0s8:1
iface enp0s8 inet static
    address 172.16.0.11/24

~
```

Zrzut ekranu 11 Utworzenie aliasu dla interfejsu *enp0s8* i skonfigurowanie danych adresowych

Wprowadzone zmiany możemy wyświetlić poleceniem *ip a*, a także przetestować wykonując polecenie *ping* na obydwa adresy z innej maszyny wirtualnej podłączonej do tej samej sieci.



```
root@debian-server:~# 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 08:00:27:2a:3a:5d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86386sec preferred_lft 86386sec
        inet6 fd17:625c:f037:2:a:3a5d/64 scope global dynamic mngtmpaddr
            valid_lft 86387sec preferred_lft 14387sec
            inet6 fe80::a00:27ff:fe2a:3a5d/64 scope link
                valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:65:2b:f5 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.10/24 brd 172.16.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
        inet 172.16.0.11/24 scope global secondary enp0s8:1
            valid_lft forever preferred_lft forever
            inet6 fc00::10:64 scope global
                valid_lft forever preferred_lft forever
                inet6 fe80::a00:27ff:fe65:2bf5/64 scope link
                    valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:be:30:9d brd ff:ff:ff:ff:ff:ff
root@debian-server:~#
```

Zrzut ekranu 12 Sprawdzenie konfiguracji sieciowej po utworzeniu aliasu na interfejsie *enp0s8*.

```
mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

[root@localhost ~]# nmcli connection modify Wired\ connection\ 1 ipv4.method manual ipv4.addresses 172.16.0.20/24
[root@localhost ~]# ip a s enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9f:1b:brd brd ff:ff:ff:ff:ff:ff
        inet 172.16.0.20/24 brd 172.16.0.255 scope global noprefixroute enp0s8
            valid_lft forever preferred_lft forever
        inet6 fe80::817c:2ce8:261c:fe61/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
[root@localhost ~]# ping 172.16.0.10
PING 172.16.0.10 (172.16.0.10) 56(84) bytes of data.
64 bytes from 172.16.0.10: icmp_seq=1 ttl=64 time=0.853 ms
64 bytes from 172.16.0.10: icmp_seq=2 ttl=64 time=0.642 ms
64 bytes from 172.16.0.10: icmp_seq=3 ttl=64 time=0.756 ms
64 bytes from 172.16.0.10: icmp_seq=4 ttl=64 time=1.08 ms
^C
--- 172.16.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.642/0.832/1.078/0.160 ms
[root@localhost ~]# ping 172.16.0.11
PING 172.16.0.11 (172.16.0.11) 56(84) bytes of data.
64 bytes from 172.16.0.11: icmp_seq=1 ttl=64 time=1.12 ms
64 bytes from 172.16.0.11: icmp_seq=2 ttl=64 time=0.589 ms
64 bytes from 172.16.0.11: icmp_seq=3 ttl=64 time=0.664 ms
64 bytes from 172.16.0.11: icmp_seq=4 ttl=64 time=0.779 ms
^C
--- 172.16.0.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.589/0.786/1.115/0.201 ms
[root@localhost ~]#
```

Zrzut ekranu 13 Efekt wykonania polecenia *ping* na obydwa adresy w ramach interfejsu enp0s8.

Zgodnie z oczekiwaniami, polecenie *ping* przeszło dla obydwu adresów IP w ramach tego samego interfejsu sieciowego maszyny wirtualnej z systemem Debian 12.

## Zadanie 4. Konfiguracja sieciowa w systemie Fedora 39 Server.

W odróżnieniu od systemu Debian 12, konfigurację sieciową w systemie Fedora 39 Server wykonuje się nie na plikach, a za pomocą polecień – konkretnie polecenia *nmcli* (NetworkManager CLI).

Konfigurując sieć w systemie Fedora będziemy wyróżniać dwa rodzaje obiektów – urządzenia i połączenia. W danej chwili dla danego urządzenia może być aktywne tylko jedno połączenie (chyba że będą to połączenia typu *bridge* lub *vlan*), ale dzięki takiemu podziałowi możliwe jest utworzenie kilku „profili”, na przykład jeden z adresacją statyczną, a drugi z dynamiczną z wykorzystaniem DHCP.

Istniejące połączenia można modyfikować poleceniem *nmcli connection modify <nazwa\_połączenia> [...opcje]*. Wszystkie skonfigurowane i dostępne opcje można sprawdzić poleceniem *nmcli connection show <nazwa\_połączenia>*.

Przykładowo, aby nadać komputerowi dane adresowe zgodnie z poniższą tabelą, wykonamy polecenia zgodnie z zamieszczonym zrzutem ekranu.

enp0s3 (NAT)	enp0s8 (Internal network – intnet)
Adres IP: 10.0.2.25 Maska podsieci: 255.255.255.0 Brama: 10.0.2.2 DNS: 10.0.2.3	Adres IP: 172.16.0.20 Maska podsieci: 255.255.255.0
Adres IPv6: FC00::20/64 Brama IPv6: FE80::1 DNS IPv6: FE80::1	Adres IPv6: FD00::20/64

Tabela 4 Dane adresowe dla maszyny wirtualnej z systemem Fedora Server.



```
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.method manual ipv4.addresses 10.0.2.25/24 ipv4.gateway 10.0.2.2 ipv4.dns 10.0.2.3 ipv6.method manual ipv6.addresses fc00::20/64 ipv6.gateway fe80::1 ipv6.dns fe80::1
[root@localhost ~]# nmcli connection modify Wired\ connection\ 1 connection.id enp0s8 ipv4.method manual ipv4.address 172.16.0.20/24 ipv6.method manual ipv6.address fd00::20/64
[root@localhost ~]# nmcli connection
NAME      UUID                                  TYPE      DEVICE
enp0s3    c168d1d2-1234-36be-8579-401c775e6b9d  ethernet  enp0s3
enp0s8    ade86491-a80f-38bc-a9c8-89630c400714  ethernet  enp0s8
lo        1a25e655-79db-4ebd-9616-7ce0f986db26  loopback  lo
[root@localhost ~]#
```

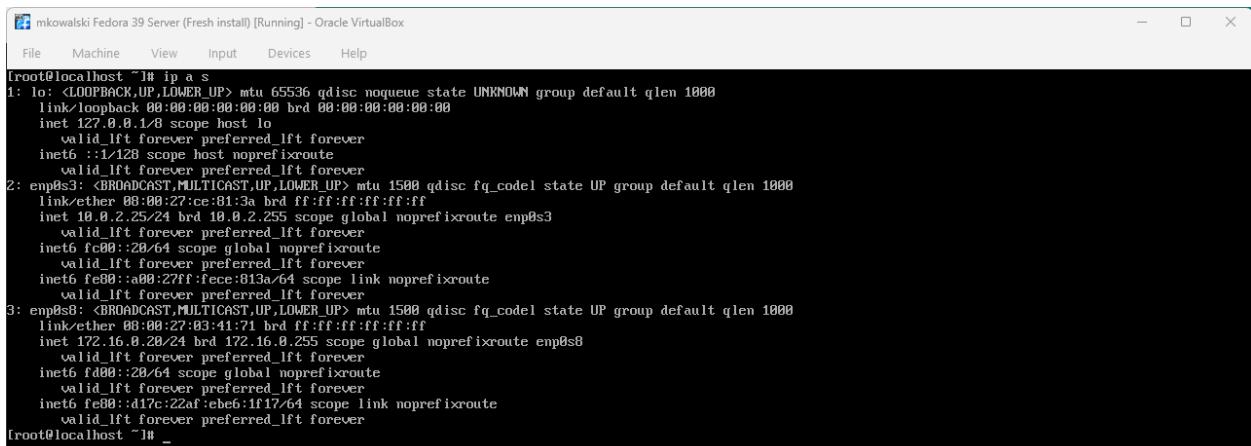
Zrzut ekranu 14 Konfiguracja sieciowa maszyny wirtualnej z systemem Fedora 39 Server.

```
nmcli connection modify enp0s3 ipv4.method manual ipv4.addresses 10.0.2.25/24
ipv4.gateway 10.0.2.2 ipv4.dns 10.0.2.3 ipv6.method manual ipv6.addresses fc00::20/64
ipv6.gateway fe80::1 ipv6.dns fe80::1
```

```
nmcli connection modify Wired\ connection\ 1 connection.id enp0s8 ipv4.method manual
ipv4.address 172.16.0.20/24 ipv6.method manual ipv6.address fd00::20/64
```

Dodatkowo, w ramach drugiego polecenia zmieniłem nazwę połączenia na `enp0s8`, tak, aby była zgodna z interfejsem, dla którego zostało ono zdefiniowane.

Wprowadzoną konfigurację możemy sprawdzić poleceniem `ip a s, ip r i cat /etc/resolv.conf`.

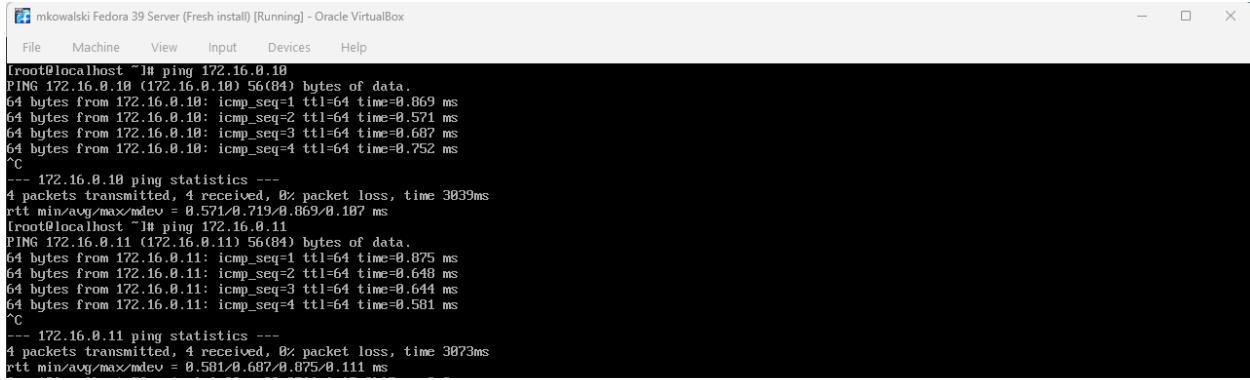


```
[root@localhost ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 0.0.0.0 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 brd 0.0.0.0 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 00:00:27:ce:81:3a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.25/24 brd 10.0.2.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fece:813a/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:00:27:03:41:71 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.20/24 brd 172.16.0.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe03:4171/64 brd ff:ff:ff:ff:ff:ff scope link noprefixroute
            valid_lft forever preferred_lft forever
[root@localhost ~]#
```

Zrzut ekranu 15 Sprawdzenie poprawności nadania danych adresowych.

Zwróćmy uwagę, że nie ma potrzeby restartowania żadnej usługi po wprowadzeniu zmian – zmiany wprowadzane są natychmiast po wykonaniu polecenia.

Za pomocą polecenia *ping* możemy też potwierdzić, czy możliwa jest komunikacja z maszyną wirtualną z systemem Debian.

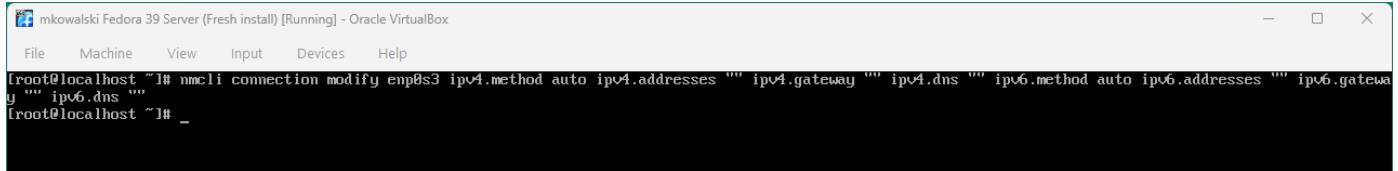


```
mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
lroot@localhost ~% ping 172.16.0.10
PING 172.16.0.10 (172.16.0.10) 56(84) bytes of data.
64 bytes from 172.16.0.10: icmp_seq=1 ttl=64 time=0.869 ms
64 bytes from 172.16.0.10: icmp_seq=2 ttl=64 time=0.571 ms
64 bytes from 172.16.0.10: icmp_seq=3 ttl=64 time=0.687 ms
64 bytes from 172.16.0.10: icmp_seq=4 ttl=64 time=0.752 ms
^C
--- 172.16.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3039ms
rtt min/avg/max/mdev = 0.571/0.719/0.869/0.107 ms
lroot@localhost ~% ping 172.16.0.11
PING 172.16.0.11 (172.16.0.11) 56(84) bytes of data.
64 bytes from 172.16.0.11: icmp_seq=1 ttl=64 time=0.875 ms
64 bytes from 172.16.0.11: icmp_seq=2 ttl=64 time=0.648 ms
64 bytes from 172.16.0.11: icmp_seq=3 ttl=64 time=0.644 ms
64 bytes from 172.16.0.11: icmp_seq=4 ttl=64 time=0.581 ms
^C
--- 172.16.0.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.581/0.687/0.875/0.111 ms
```

Zrzut ekranu 16 Weryfikacja komunikacji z serwerem Debian w ramach podsieci 172.16.0.0/24.

## Zadanie 5. Konfiguracja ustawień TCP/IP w systemie Fedora z wykorzystaniem serwera DHCP.

Do realizacji zadania wykorzystam interfejs `enp0s3` podłączony do adaptera w trybie `NAT`. Aby włączyć automatyczne uzyskiwanie danych adresowych z serwera DHCP wystarczy ustawić opcję `ipv4.method auto` i `ipv6.method auto`. NetworkManager zignoruje wszystkie wcześniej ustawione dane adresowe, ale można je wyczyścić przypisując pustą wartość do każdego ze skonfigurowanych wartości.

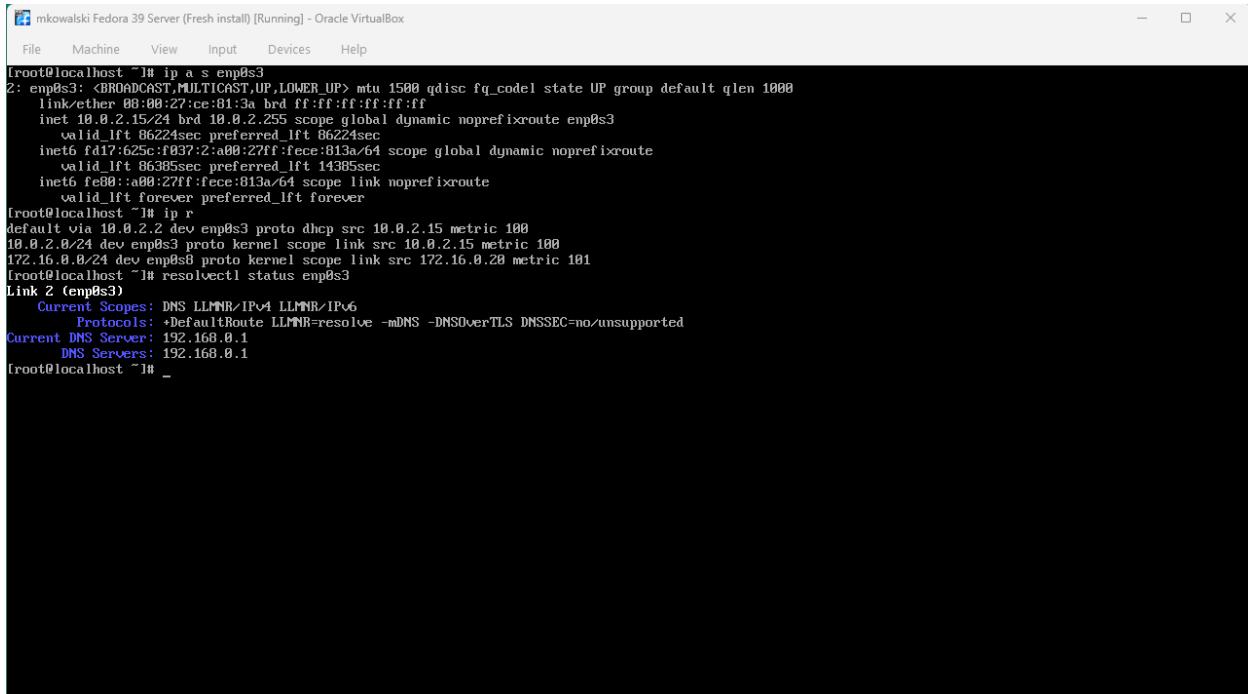


```
mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.method auto ipv4.addresses "" ipv4.gateway "" ipv4.dns "" ipv6.method auto ipv6.addresses "" ipv6.gateway "" ipv6.dns ""
[root@localhost ~]#
```

Zrzut ekranu 17 Włączenie pobierania danych adresowych z usługi DHCP.

`nmcli connection modify enp0s3 ipv4.method auto ipv4.addresses "" ipv4.gateway "" ipv4.dns "" ipv6.method auto ipv6.addresses "" ipv6.gateway "" ipv6.dns ""`

Po wykonaniu polecenia możemy sprawdzić jakie dane adresowe uzyskał komputer w ramach usługi DHCP (dla IPv4) i mechanizmu SLAAC (dla IPv6).



```
mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# ip a s enp0s3
2: enp0s3: BROADCAST,MULTICAST,UP,LOWER_UP mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0e:27:ce:81:3a brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
            valid_lft 86224sec preferred_lft 86224sec
            inet6 fd17:625c:0372:2:00:27ff:fece:813a/64 scope global dynamic noprefixroute
                valid_lft 14385sec preferred_lft 14385sec
                inet6 fe80::a00:27ff:fece:813a/64 scope link noprefixroute
                    valid_lft forever preferred_lft forever
[root@localhost ~]# ip r
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
172.16.0.0/24 dev enp0s8 proto kernel scope link src 172.16.0.20 metric 101
[root@localhost ~]# resolvelcti status enp0s3
Link 2 (enp0s3)
  Current Scopes: DNS LLNMR/IPv4 LLNMR/IPv6
  Protocols: +DefaultRoute LLNMR=resolve -mDNS -DNSOverTLS DNSSEC=no/unsupported
  Current DNS Server: 192.168.0.1
    DNS Servers: 192.168.0.1
[root@localhost ~]#
```

Zrzut ekranu 18 Sprawdzenie danych adresowych przydzielonych na interfejsie `enp0s3` maszynie wirtualnej z systemem Fedora.

**enp0s3**

Adres IP: 10.0.2.15 (wewnętrzny)

Maska podsieci: 255.255.255.0

Brama: 10.0.2.2

DNS: 192.168.0.1

Adres IPv6: FD17:625C:F037:2:A00:27FF:FECE:813A/64 (unikalny lokalny)

*Tabela 5 Dane adresowe przydzielone na interfejsie enp0s3 serwera Debian 12 z wykorzystaniem DHCP i SLAAC.*

Zwróćmy uwagę, że obydwa komputery uzyskały identyczne adresy IPv4. Nie jest to problem, ponieważ adaptery pracują w trybie sieci *NAT*, a to znaczy, że każdy z nich jest fizycznie w osobnej sieci.

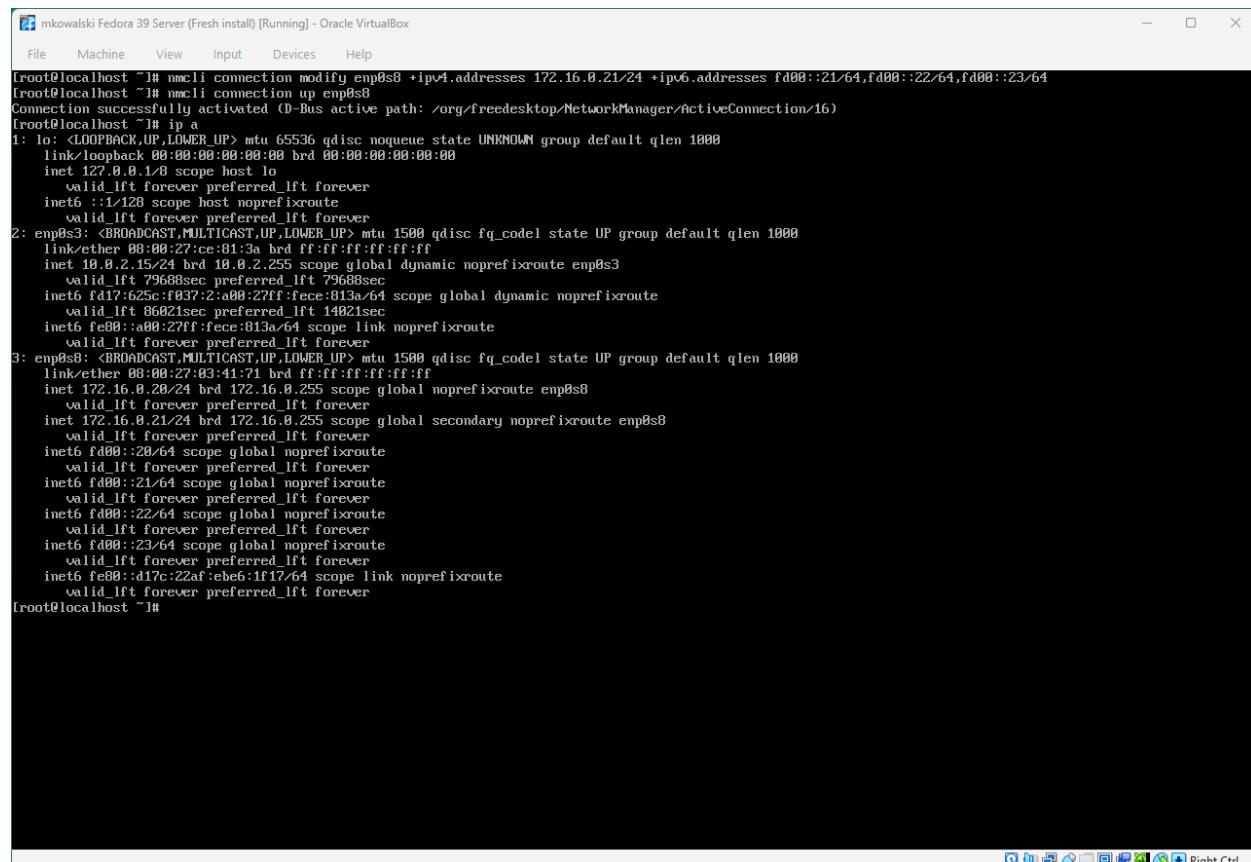
## Zadanie 6. Konfiguracja wielu adresów IP w ramach jednego połączenia.

Aby umożliwić komputerowi z systemem Fedora 39 Server podłączenie się do kilku sieci w ramach jednego interfejsu, wystarczy zmodyfikować połączenie, tak, aby miało skonfigurowane kilka adresów IP. Można wykorzystać do tego polecenie *nmcli connection modify*, wstawiając przed każdą opcją konfigurującą adres IP znak +, na przykład *+ipv4.addresses w.x.y.z/m*. Aby nie powtarzać wielokrotnie jednej opcji dla różnych adresów, można oddzielić je przecinkami.

Przykładowo, aby nadać na wybranym interfejsie dane adresowe zgodnie z poniższą tabelą, wykonamy polecenia zgodnie z zamieszczonym zrzutem ekranu.

enp0s8
Adres IP: 172.16.0.20/24
Adres IP: 172.16.0.21/24 (nowy)
Adres IPv6: FD00::20/64
Adres IPv6: FD00::21/64 (nowy)
Adres IPv6: FD00::22/64 (nowy)
Adres IPv6: FD00::23/64 (nowy)

Tabela 5 Dane adresowe dla maszyny wirtualnej z systemem Fedora Server.

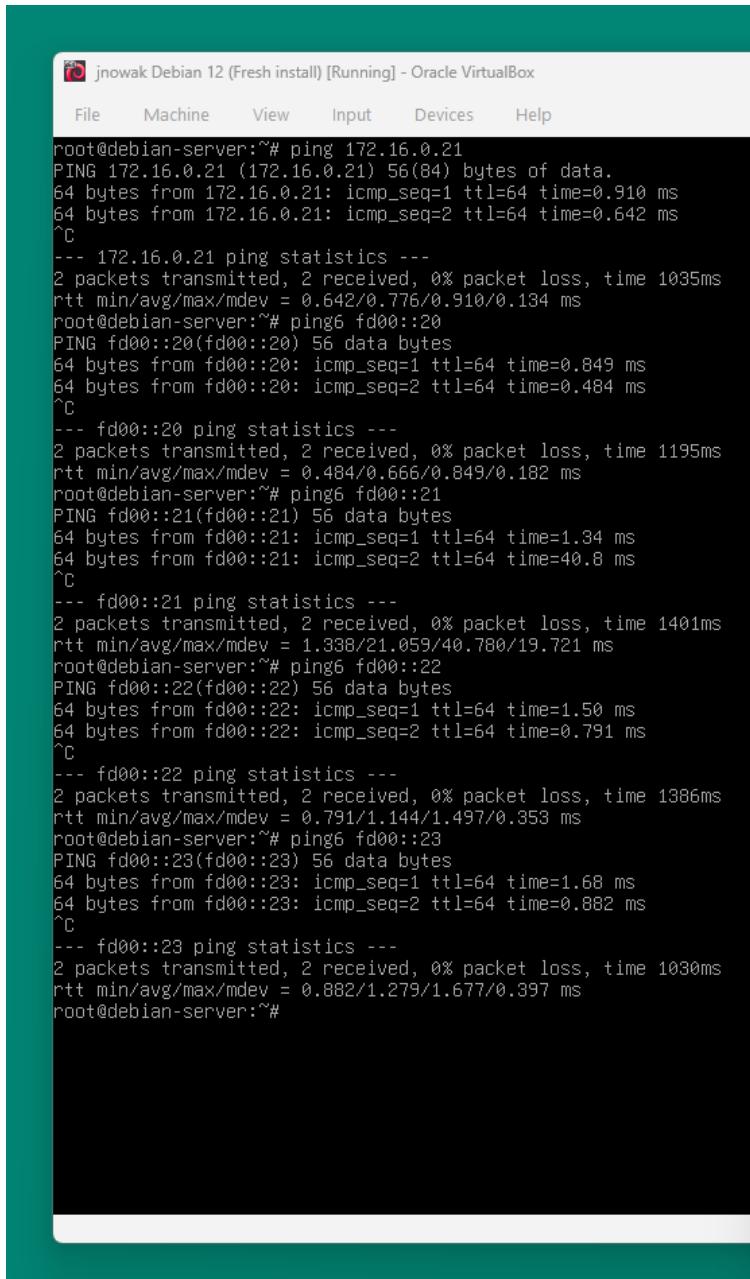


```
mikowalski Fedora 39 Server [Fresh install] [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# nmcli connection modify enp0s8 +ipv4.addresses 172.16.0.21/24 +ipv6.addresses fd00::21/64,fd00::22/64,fd00::23/64
[root@localhost ~]# nmcli connection up enp0s8
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/16)
[root@localhost ~]# 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 brd 127.0.0.1 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 00:00:27:ce:81:3a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 79688sec preferred_lft 79688sec
    inet6 fd17:625c:fd37:2:ae00:27ff:fece:813a/64 scope global dynamic noprefixroute
        valid_lft 14021sec
    inet6 fe80::ae00:27ff:fece:813a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:00:27:03:41:71 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.20/24 brd 172.16.0.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet 172.16.0.21/24 brd 172.16.0.255 scope global secondary noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fd00::20/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fd00::21/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fd00::22/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fd00::23/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::d17c:22af:febe:1f17/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost ~]#
```

Zrzut ekranu 19 Konfiguracja dodatkowych adresów IP na interfejsie sieciowym enp0s8 w systemie Fedora 39 Server.

```
nmcli connection modify enp0s8 +ipv4.addresses 172.16.0.21/24 +ipv6.addresses fd00::21/64,fd00::22/64,fd00::23/64
```

Na koniec warto zweryfikować poprawność dokonanej konfiguracji sieciowej. W tym celu na komputerze z systemem Debian 12 uruchomimy polecenia *ping* i *ping6*, nakierowane na kolejne adresy maszyny wirtualnej z systemem Fedora Server.



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

root@debian-server:~# ping 172.16.0.21
PING 172.16.0.21 (172.16.0.21) 56(84) bytes of data.
64 bytes from 172.16.0.21: icmp_seq=1 ttl=64 time=0.910 ms
64 bytes from 172.16.0.21: icmp_seq=2 ttl=64 time=0.642 ms
^C
--- 172.16.0.21 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1035ms
rtt min/avg/max/mdev = 0.642/0.776/0.910/0.134 ms
root@debian-server:~# ping6 fd00::20
PING fd00::20(fd00::20) 56 data bytes
64 bytes from fd00::20: icmp_seq=1 ttl=64 time=0.849 ms
64 bytes from fd00::20: icmp_seq=2 ttl=64 time=0.484 ms
^C
--- fd00::20 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1195ms
rtt min/avg/max/mdev = 0.484/0.666/0.849/0.182 ms
root@debian-server:~# ping6 fd00::21
PING fd00::21(fd00::21) 56 data bytes
64 bytes from fd00::21: icmp_seq=1 ttl=64 time=1.34 ms
64 bytes from fd00::21: icmp_seq=2 ttl=64 time=40.8 ms
^C
--- fd00::21 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1401ms
rtt min/avg/max/mdev = 1.338/21.059/40.780/19.721 ms
root@debian-server:~# ping6 fd00::22
PING fd00::22(fd00::22) 56 data bytes
64 bytes from fd00::22: icmp_seq=1 ttl=64 time=1.50 ms
64 bytes from fd00::22: icmp_seq=2 ttl=64 time=0.791 ms
^C
--- fd00::22 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1386ms
rtt min/avg/max/mdev = 0.791/1.144/1.497/0.353 ms
root@debian-server:~# ping6 fd00::23
PING fd00::23(fd00::23) 56 data bytes
64 bytes from fd00::23: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from fd00::23: icmp_seq=2 ttl=64 time=0.882 ms
^C
--- fd00::23 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1030ms
rtt min/avg/max/mdev = 0.882/1.279/1.677/0.397 ms
root@debian-server:~#
```

Zrzut ekranu 20 Weryfikacja poprawności konfiguracji sieciowej maszyny wirtualnej z systemem Fedora.

Jak widać, wszystkie polecenia *ping* „przechodzą”, a więc konfiguracja jest zrealizowana poprawnie.

## Zadanie 7. Plik /etc/hosts.

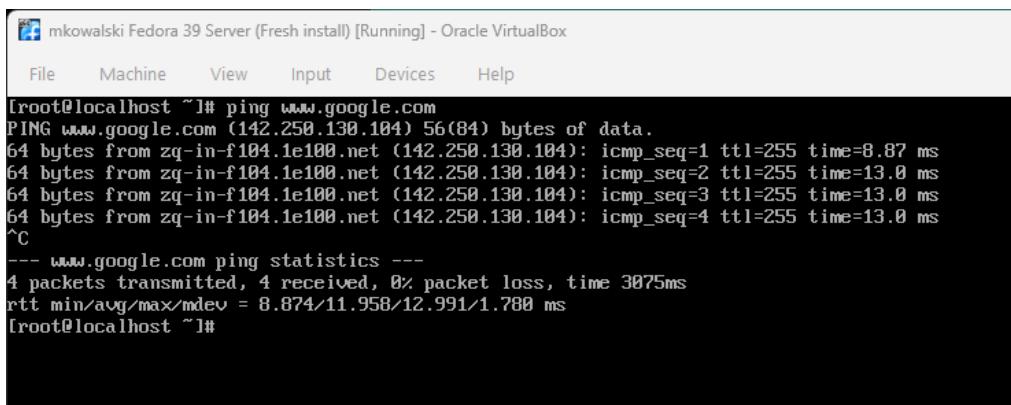
Plik `/etc/hosts` pozwala na ręczne zdefiniowanie mappingu dla wybranych nazw mnemonicznych w adresy IP. Przypisując danemu rekordowi adres `0.0.0.0` można uniemożliwić komunikację komputera z zadaną, kwalifikowaną nazwą domenową.

Dla przykładu, zrealizujmy poniższy scenariusz mapowania.

Maszyna wirtualna	Nazwa	Adres IPv4/IPv6
Fedora	debian	172.16.0.10
	debian6	FD00::10
	www.google.com	0.0.0.0 (blokada)
Debian	fedora	172.16.0.20
	fedora6	FD00::20

Tabela 6 Docelowa konfiguracja plików `/etc/hosts`.

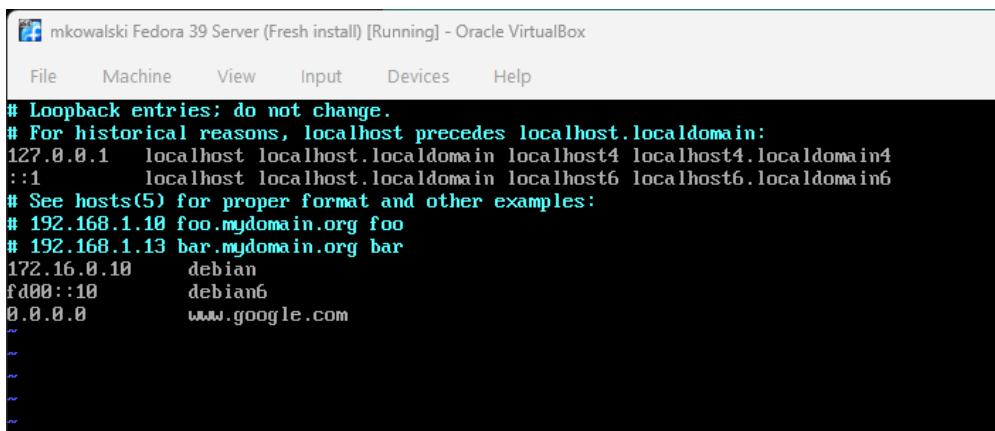
Na początku sprawdźmy na jaki adres nakierowana jest domena `www.google.com`, korzystając z polecenia `ping` na serwerze Fedora.



```
mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@localhost ~]# ping www.google.com
PING www.google.com (142.250.130.104) 56(84) bytes of data.
64 bytes from zq-in-f104.1e100.net (142.250.130.104): icmp_seq=1 ttl=255 time=8.87 ms
64 bytes from zq-in-f104.1e100.net (142.250.130.104): icmp_seq=2 ttl=255 time=13.0 ms
64 bytes from zq-in-f104.1e100.net (142.250.130.104): icmp_seq=3 ttl=255 time=13.0 ms
64 bytes from zq-in-f104.1e100.net (142.250.130.104): icmp_seq=4 ttl=255 time=13.0 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 8.874/11.958/12.991/1.780 ms
[root@localhost ~]#
```

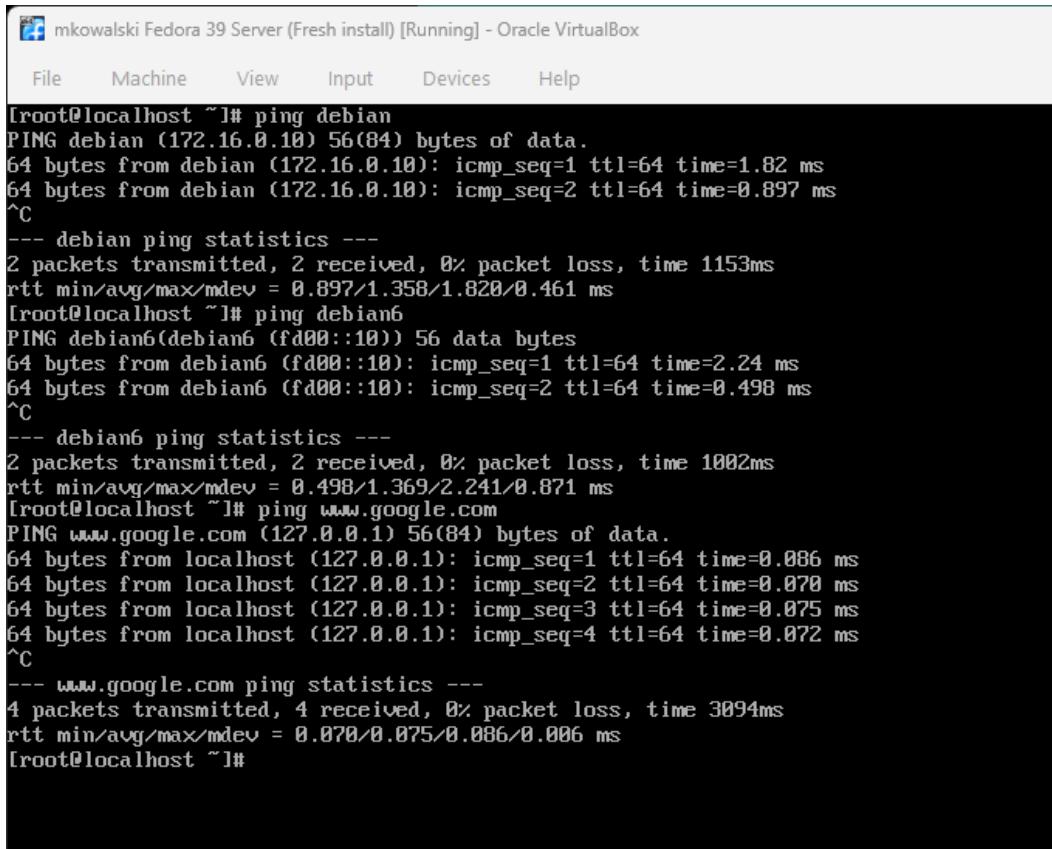
Zrzut ekranu 21 Sprawdzenie adresu IP na jaki nakierowana jest domena `www.google.com` przy pomocy polecenia `ping`.

Możemy teraz wprowadzić zmiany w pliku `/etc/hosts`, zgodnie z powyższą tabelką.



```
mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
# Loopback entries: do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.mydomain.org foo
# 192.168.1.13 bar.mydomain.org bar
172.16.0.10    debian
fd00::10      debian6
0.0.0.0        www.google.com
~
~
~
~
```

Zrzut ekranu 22 Konfiguracja pliku `/etc/hosts` na maszynie wirtualnej z systemem Fedora 39 Server.

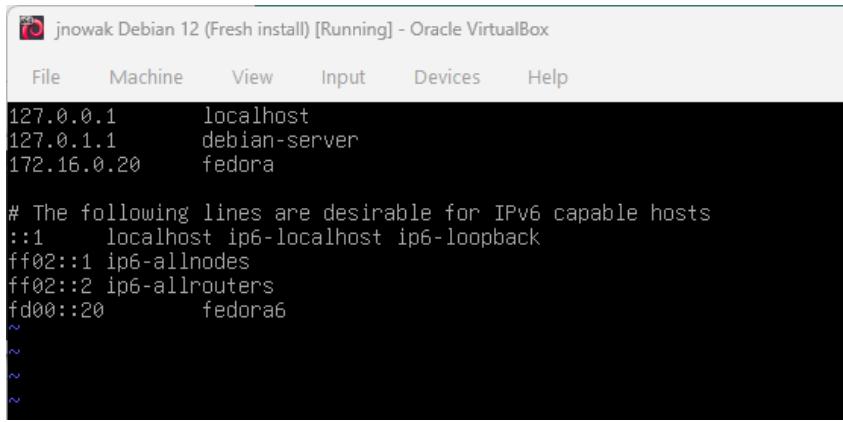


```
[root@localhost ~]# ping debian
PING debian (172.16.0.10) 56(84) bytes of data.
64 bytes from debian (172.16.0.10): icmp_seq=1 ttl=64 time=1.82 ms
64 bytes from debian (172.16.0.10): icmp_seq=2 ttl=64 time=0.897 ms
^C
--- debian ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1153ms
rtt min/avg/max/mdev = 0.897/1.358/1.820/0.461 ms
[root@localhost ~]# ping debian6
PING debian6(fd00::10) 56 data bytes
64 bytes from debian6 (fd00::10): icmp_seq=1 ttl=64 time=2.24 ms
64 bytes from debian6 (fd00::10): icmp_seq=2 ttl=64 time=0.498 ms
^C
--- debian6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.498/1.369/2.241/0.871 ms
[root@localhost ~]# ping www.google.com
PING www.google.com (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.072 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3094ms
rtt min/avg/max/mdev = 0.070/0.075/0.086/0.006 ms
[root@localhost ~]#
```

Zrzut ekranu 23 Weryfikacja konfiguracji pliku /etc/hosts pod systemem Fedora.

Jak widać, komunikacja z serwerem Debian możliwa jest z wykorzystaniem nazwy *debian* i *debian 6*, a komunikacja z domeną [www.google.com](http://www.google.com) jest blokowana – na żądanie odpowiada *localhost*.

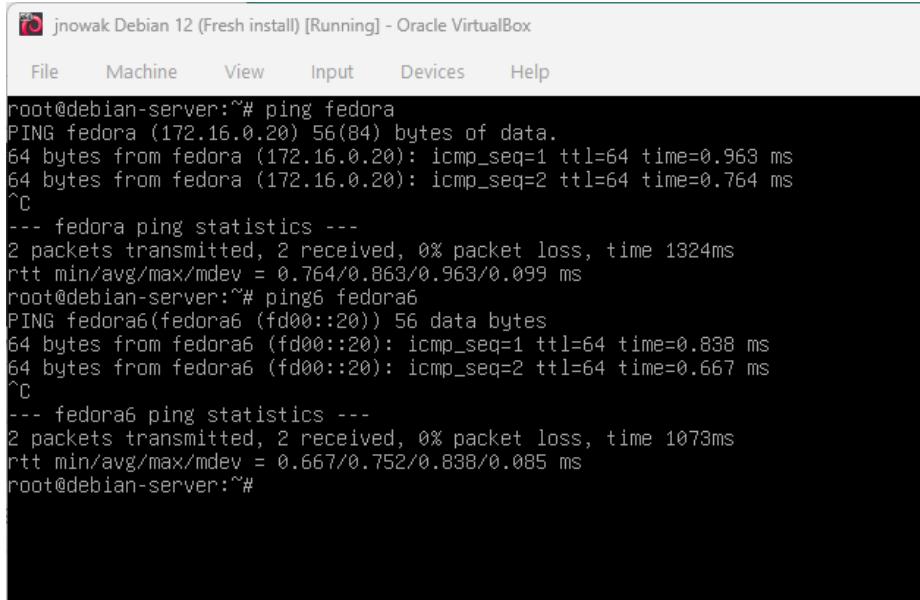
W analogiczny sposób przebiega konfiguracja serwera z systemem Debian 12.



```
127.0.0.1      localhost
127.0.1.1      debian-server
172.16.0.20    fedora

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
fd00::20    fedora6
~
~
~
```

Zrzut ekranu 24 Konfiguracja pliku /etc/hosts na maszynie wirtualnej z systemem Debian 12.



```
root@debian-server:~# ping fedora
PING fedora (172.16.0.20) 56(84) bytes of data.
64 bytes from fedora (172.16.0.20): icmp_seq=1 ttl=64 time=0.963 ms
64 bytes from fedora (172.16.0.20): icmp_seq=2 ttl=64 time=0.764 ms
^C
--- fedora ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1324ms
rtt min/avg/max/mdev = 0.764/0.863/0.963/0.099 ms
root@debian-server:~# ping6 fedora6
PING fedora6(fd00::20) 56 data bytes
64 bytes from fedora6 (fd00::20): icmp_seq=1 ttl=64 time=0.838 ms
64 bytes from fedora6 (fd00::20): icmp_seq=2 ttl=64 time=0.667 ms
^C
--- fedora6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1073ms
rtt min/avg/max/mdev = 0.667/0.752/0.838/0.085 ms
root@debian-server:~#
```

Zrzut ekranu 25 Weryfikacja konfiguracji pliku /etc/hosts pod systemem Debian.

Tutaj również możliwa jest komunikacja z wykorzystaniem utworzonych nazw mnemonicznych – **fedora** i **fedora6**.

## Zadanie 8. Zmiana kolejności rozwiązywania nazw na adresy IP. Plik */etc/nsswitch.conf*.

W poprzednim zadaniu uniemożliwiliśmy komunikację z serwerem www.google.com tworząc w pliku */etc/hosts* mapping na adres 0.0.0.0.

Zrzut ekranu 26 Zablokowana komunikacja z serwerem www.google.com

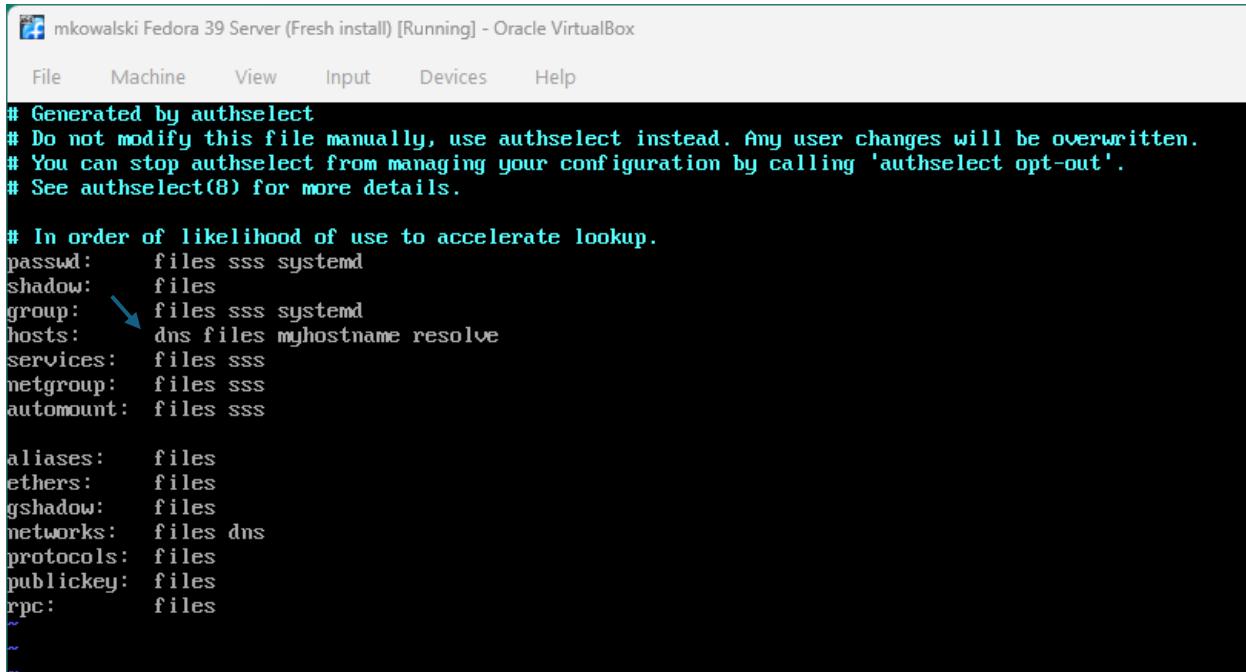
Istnieje jednak możliwość, aby to DNS (lub inne wskazane źródło) miało większy priorytet nad zdefiniowanymi w pliku mappingami. Do określenia tej kolejności służy plik */etc/nsswitch.conf*.

Zrzut ekranu 27 Zawartość pliku */etc/nsswitch.conf* przed modyfikacją.

Każdą linijkę możemy traktować jako definicję potoku, określającego kolejność odszukiwania danych. W momencie, gdy dane źródło odnajdzie informację, wynik jest natychmiast zwracany, a dalsza część sekwencji nie jest wykonywana. Sekcja *hosts* pozwala na zmianę kolejności rozwiązywania nazw na adresy IP. W tym momencie

największy priorytet mają dane zdefiniowane w plikach, a najmniejszy – te z DNS. Warunek `[!UNAVAIL=return]` spowoduje, że w miejscu jego występowania każdy stan inny niż `UNAVAIL` (źródło niedostępne), a więc między innymi „znalezione” lub „definitywnie niedostępne”, zakończy potok i następujące po nim źródła nie zostaną sprawdzone.

W celu zmiany kolejności tak, aby dane z DNS były pobierane przez danymi z pliku, należy przenieść wpis `dns` na początek linijki.

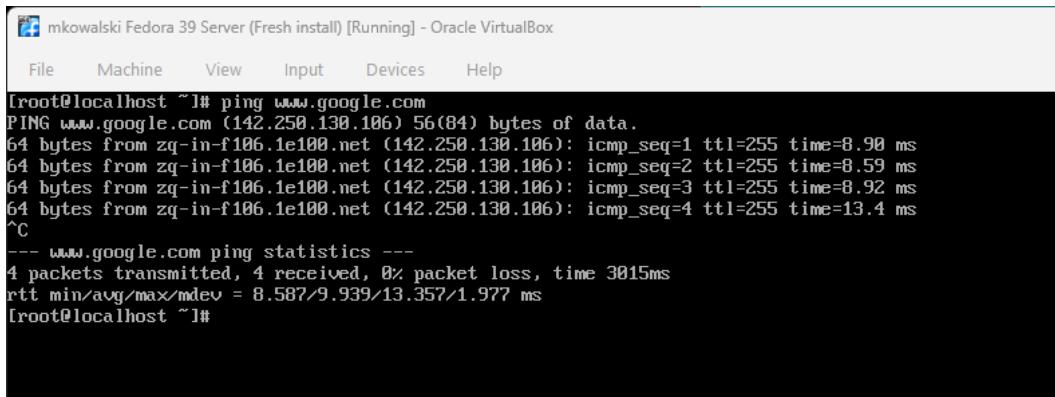


```
# Generated by authselect
# Do not modify this file manually, use authselect instead. Any user changes will be overwritten.
# You can stop authselect from managing your configuration by calling 'authselect opt-out'.
# See authselect(8) for more details.

# In order of likelihood of use to accelerate lookup.
passwd:      files sss systemd
shadow:      files
group:       files sss systemd
hosts:       dns files myhostname resolve
services:    files sss
netgroup:    files sss
automount:   files sss

aliases:     files
ethers:      files
gshadow:     files
networks:    files dns
protocols:   files
publickey:   files
rpc:         files
^
^
^
```

Zrzut ekranu 28 Zmiana kolejności potoku dla zapytań DNS.



```
[root@localhost ~]# ping www.google.com
PING www.google.com (142.250.130.106) 56(84) bytes of data.
64 bytes from zq-in-f106.1e100.net (142.250.130.106): icmp_seq=1 ttl=255 time=8.90 ms
64 bytes from zq-in-f106.1e100.net (142.250.130.106): icmp_seq=2 ttl=255 time=8.59 ms
64 bytes from zq-in-f106.1e100.net (142.250.130.106): icmp_seq=3 ttl=255 time=8.92 ms
64 bytes from zq-in-f106.1e100.net (142.250.130.106): icmp_seq=4 ttl=255 time=13.4 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3015ms
rtt min/avg/max/mdev = 8.587/9.939/13.357/1.977 ms
[root@localhost ~]#
```

Zrzut ekranu 29 Przetestowanie nowej konfiguracji w sekcji hosts w pliku /etc/nsswitch.conf.

Jak widać, po dokonaniu zmian, zablokowana w pliku `/etc/hosts` domena `www.google.com` została tym razem poprawnie rozwiązana na adres IP.

## Zadanie 9. Zarządzanie serwerem SSH.

Na maszynie wirtualnej z systemem Debian, serwer SSH można zainstalować poleceniem `apt install openssh-server`, które na koniec również go uruchomi.

Konfiguracji programu dokonamy z poziomu pliku `/etc/ssh/sshd_config` – można tu na przykład ustawić port (*Port*), na którym aplikacja nasłuchuje połączeń przychodzących i uniemożliwić bezpośrednie logowanie na konto *root* (*PermitRootLogin no*).



```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File    Machine    View    Input    Devices    Help

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 30 ←
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no ←
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile    .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

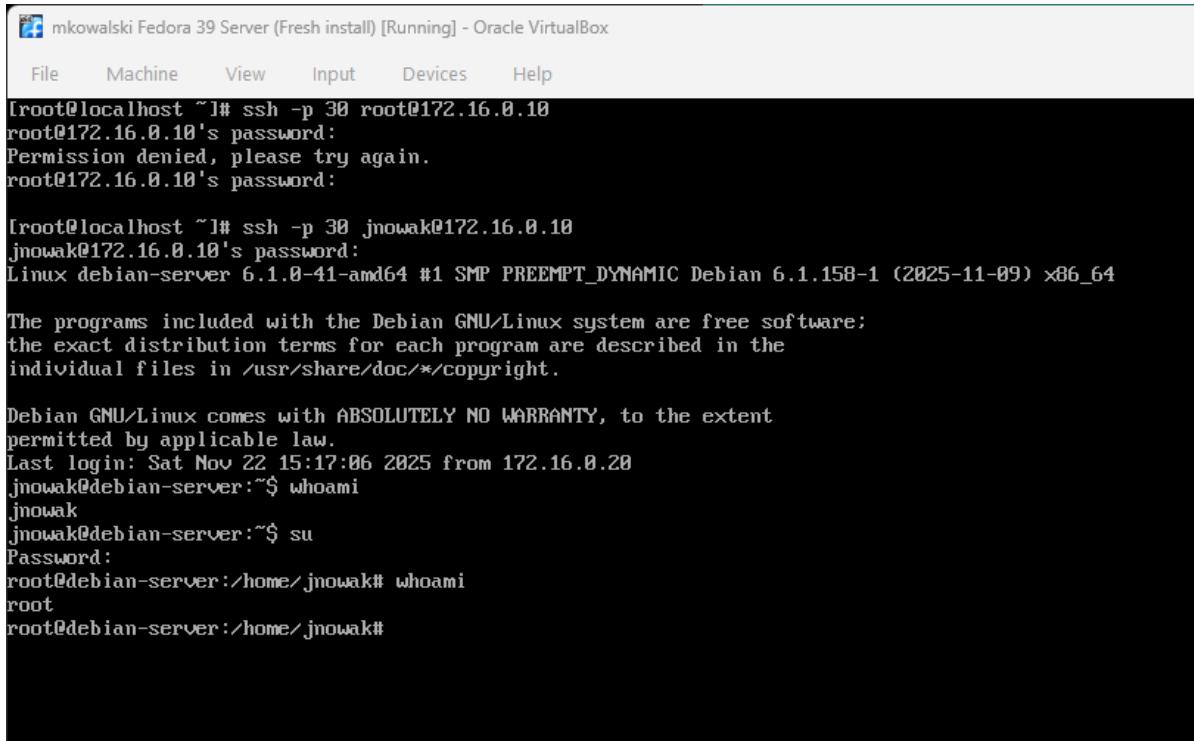
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
1 more line; before #10 1 second ago
```

Zrzut ekranu 30 Zmiana portu, na którym działa serwer SSH na 30 oraz uniemożliwienie bezpośredniego logowania użytkownikowi root.

Po wprowadzeniu zmian konieczne jest ponowne uruchomienie serwera SSH poleceniem `systemctl restart sshd`.

Aby sprawdzić poprawność dokonanej konfiguracji, spróbujemy zalogować się przez SSH na serwer Debian. Wykorzystamy w tym celu polecenie `ssh`, wskazując numer portu przy pomocy flagi `-p`.

```
ssh -p 30 <użytkownik>@172.16.0.10
```



The screenshot shows a terminal window within Oracle VM VirtualBox. The title bar reads "mkowalski Fedora 39 Server (Fresh install) [Running] - Oracle VirtualBox". The menu bar includes "File", "Machine", "View", "Input", "Devices", and "Help". The terminal content is as follows:

```
[root@localhost ~]# ssh -p 30 root@172.16.0.10
root@172.16.0.10's password:
Permission denied, please try again.
root@172.16.0.10's password:

[root@localhost ~]# ssh -p 30 jnowak@172.16.0.10
jnowak@172.16.0.10's password:
Linux debian-server 6.1.0-41-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.158-1 (2025-11-09) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Nov 22 15:17:06 2025 from 172.16.0.20
jnowak@debian-server:~$ whoami
jnowak
jnowak@debian-server:~$ su
Password:
root@debian-server:/home/jnowak$ whoami
root
root@debian-server:/home/jnowak$
```

Zrzut ekranu 31 Niedana próba zalogowania na konto root. Poprawne logowanie na konto jnowak i przelogowanie się na konto superużytkownika.

Jak widać, próba zalogowania się na konto administratora nie powiodła się – mimo wpisywania poprawnego hasła, logowanie nie było możliwe. Natomiast, w przypadku użytkownika `jnowak`, udało się poprawnie zalogować, a następnie polecieniem `su` przelogować na konto `root`.

## Zadanie 10. Plik `~/.ssh/known_hosts`.

W przypadku, gdy zmieni się klucz szyfrujący dla serwera SSH, użytkownikom, którzy wcześniej się do niego łączyli, przy następnej próbie logowania zostanie wyświetcone ostrzeżenie. Takie zabezpieczenie pozwala zapobiegać atakom typu *man-in-the-middle*, ale również blokuje logowanie w sytuacjach, w których mamy pewność, że rzeczywiście zmieniły się poświadczenia serwera.

Aby rozwiązać ten problem, należy z pliku `~/.ssh/known_hosts` usunąć linijkę odpowiadającą danemu kluczowi.

Zrzut ekranu 32 Zawartość pliku ~/.ssh/known\_hosts.

Przypuśćmy, że zmienił się klucz `ed25519`, dlatego usuwamy pierwszą linijkę, zapisujemy plik i próbujemy zalogować się ponownie przez `ssh`.

```
[mkowalski@localhost ~]$ ssh -p 30 jnowak@172.16.0.10
The authenticity of host '172.16.0.10:30 ([172.16.0.10]:30)' can't be established.
ED25519 key fingerprint is SHA256:bg5kCkpt0+stICzAnfSashMgFyaypyMKBP1I6I3m8Bs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.16.0.10]:30 (ED25519)' to the list of known hosts.
jnowak@172.16.0.10's password:
jnowak@172.16.0.10:~# lsb_release -a
LSB Version:    [lsb_release -a] 6.1.0-41-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.158-1 (2025-11-09) x86_64

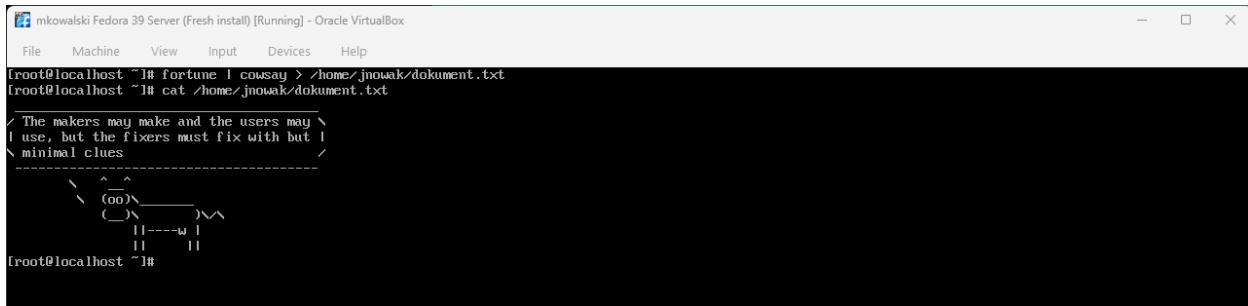
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Nov 22 15:17:46 2025 from 172.16.0.20
jnowak@debian-server:~$ su
Password:
root@debian-server:/home/jnowak# whoami
root
root@debian-server:/home/jnowak#
```

Zrzut ekranu 33 Potwierdzenie chęci połączenia się z serwerem o nieznanym kluczu ed25519.

## Zadanie 11. Przesyłanie plików z wykorzystaniem programu *scp* (OpenSSH secure file copy).

Możliwe jest także przesyłanie plików w ramach szyfrowanej transmisji *ssh*, wykorzystując aplikację *scp*. Aby zademonstrować działanie tego mechanizmu, utwórzmy najpierw w systemie Fedora nowy plik, który następnie prześlemy na serwer z systemem Debian.



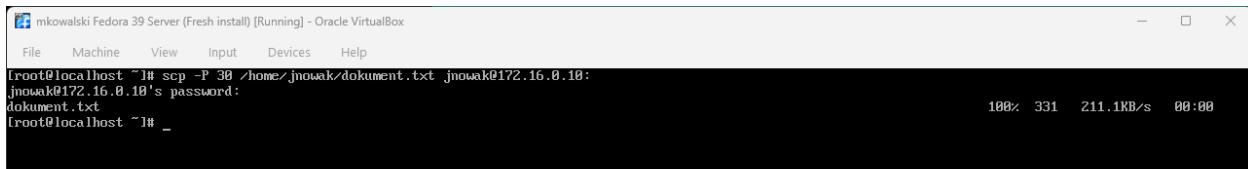
```
mkowalski Fedora 39 Server [Fresh install] [Running] - Oracle VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# fortune | cowsay > /home/jnowak/dokument.txt
[root@localhost ~]# cat /home/jnowak/dokument.txt
/ The makers may make and the users may \
| use, but the fixers must fix with but |
\ minimal clues
-----
      \  ^__^
       (oo)\_____
          (__)\       )\/\
             ||----w |
             ||     ||

[root@localhost ~]#
```

Zrzut ekranu 34 Utworzenie przykładowego pliku tekstowego w folderze użytkownika.

Mimo innego opisu maszyny, w systemie nie istnieje użytkownik *mkowalski*, a *jnowak*, stąd tak dobrana ścieżka.

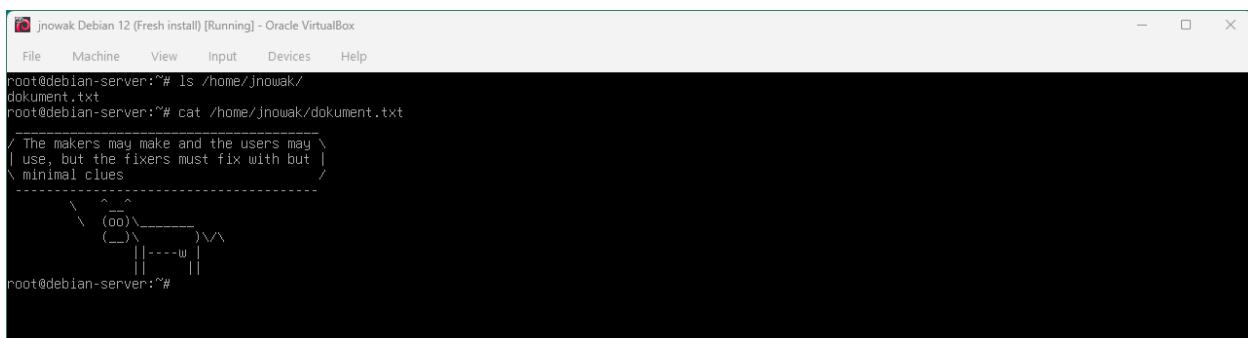
Utworzony plik możemy teraz przesłać na komputer zdalny polecienniem *scp -P x <lokalizacja\_źródłowa> <lokalizacja\_docelowa>*, określając niestandardowy port flagą *-P*. Lokalizacja zdalna zapisywana jest w formacie *użytkownik@host:ścieżka*, a pusta ścieżka do pliku (folderu) oznacza utworzenie go bezpośrednio w folderze użytkownika.



```
mkowalski Fedora 39 Server [Fresh install] [Running] - Oracle VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# scp -P 30 /home/jnowak/dokument.txt jnowak@172.16.0.10:
jnowak@172.16.0.10's password:
dokument.txt
[root@localhost ~]#
```

Zrzut ekranu 35 Przesłanie pliku na serwer Debian.

Przesłany plik możemy teraz odczytać na serwerze Debian, wykorzystując polecenie *cat*.



```
jnowak Debian 12 [Fresh install] [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@debian-server:~# ls /home/jnowak/
dokument.txt
root@debian-server:~# cat /home/jnowak/dokument.txt
/ The makers may make and the users may \
| use, but the fixers must fix with but |
\ minimal clues
-----
      \  ^__^
       (oo)\_____
          (__)\       )\/\
             ||----w |
             ||     ||

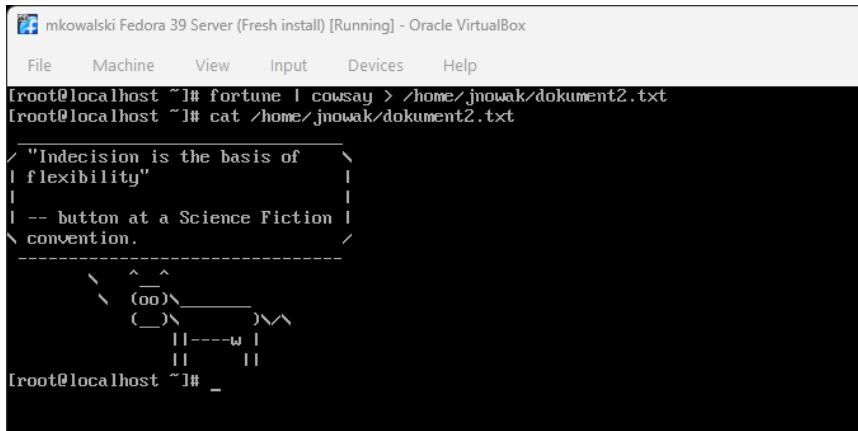
root@debian-server:~#
```

Zrzut ekranu 36 Odczytanie zawartości pliku przesłanego na serwer Debian.

Jak widać, plik został poprawnie skopiowany do folderu użytkownika *jnowak* na serwerze zdalnym.

## Zadanie 12. Kopiowanie plików z wykorzystaniem aplikacji Midnight Commander.

Przetestujmy także kopiowanie plików na serwer zdalny wykorzystując aplikację Midnight Commander. Na początku utwórzmy kolejny plik z przykładową zawartością w systemie Fedora.

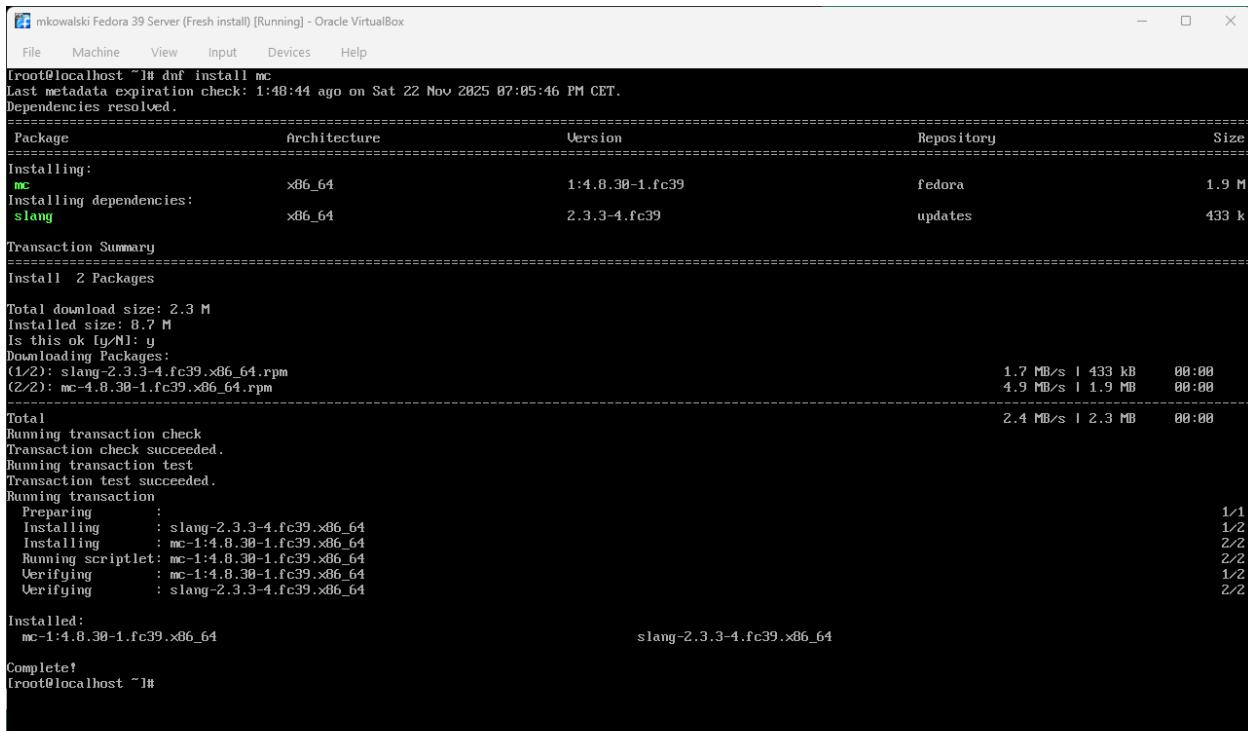


```
mkowalski Fedora 39 Server [Fresh install] [Running] - Oracle VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# fortune | cowsay > /home/jnowak/dokument2.txt
[root@localhost ~]# cat /home/jnowak/dokument2.txt
/ "Indecision is the basis of
| flexibility"
|
| -- button at a Science Fiction
\ convention.

      ^ ^
      (oo)\_____
      \  )\/\
         ||----w |
         ||     |
[root@localhost ~]# _
```

Zrzut ekranu 37 Utworzenie pliku z przykładową zawartością na serwerze Fedora.

Jeżeli na komputerze nie jest jeszcze zainstalowany jeszcze program *Midnight Commander*, należy doinstalować go poleceniem *dnf install mc*.



```
mkowalski Fedora 39 Server [Fresh install] [Running] - Oracle VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# dnf install mc
Last metadata expiration check: 1:48:44 ago on Sat 22 Nov 2025 07:05:46 PM CET.
Dependencies resolved.
=====
Transaction Summary
=====
Install 2 Packages

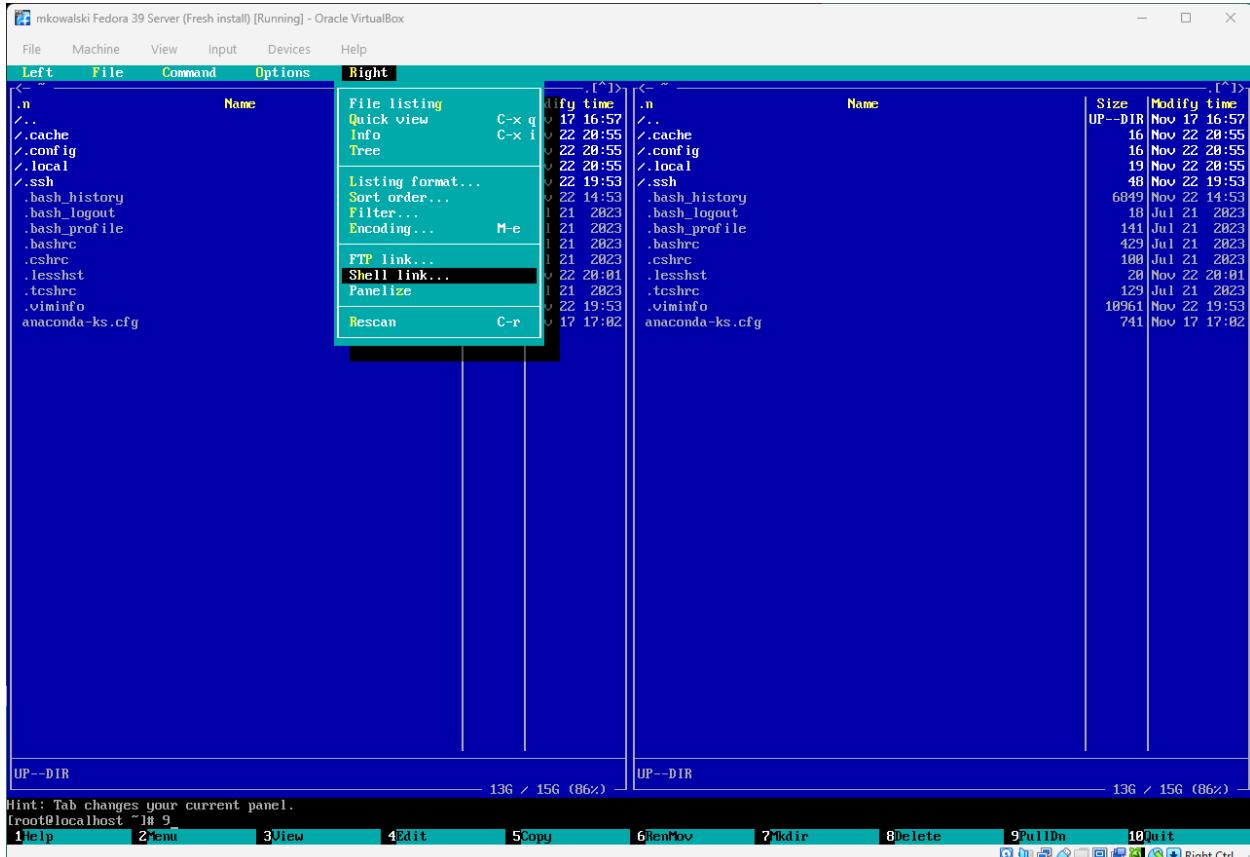
Total download size: 2.3 M
Installed size: 0.7 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): slang-2.3.3-4.fc39.x86_64.rpm 1.7 MB/s | 433 kB 00:00
(2/2): mc-4.8.30-1.fc39.x86_64.rpm 4.9 MB/s | 1.9 MB 00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Installing : slang-2.3.3-4.fc39.x86_64 1/2
  Installing : mc-4.8.30-1.fc39.x86_64 2/2
  Running scriptlet: mc-1:4.8.30-1.fc39.x86_64 2/2
  Verifying : mc-1:4.8.30-1.fc39.x86_64 1/2
  Verifying : slang-2.3.3-4.fc39.x86_64 2/2

Installed:
  mc-1:4.8.30-1.fc39.x86_64

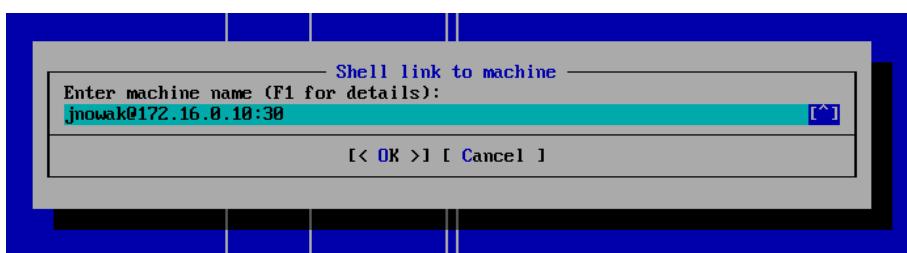
Complete!
[root@localhost ~]#
```

Zrzut ekranu 38 Instalacja pakietu *Midnight Commander*.

Po zakończonej instalacji otwieramy aplikację *mc* [*ścieżka\_do\_folderu*], a następnie z górnego menu (*F9*) dla dowolnego panelu wybieramy opcję *Shell link...*, wskazując nazwę maszyny w formacie *użytkownik@host:port*.



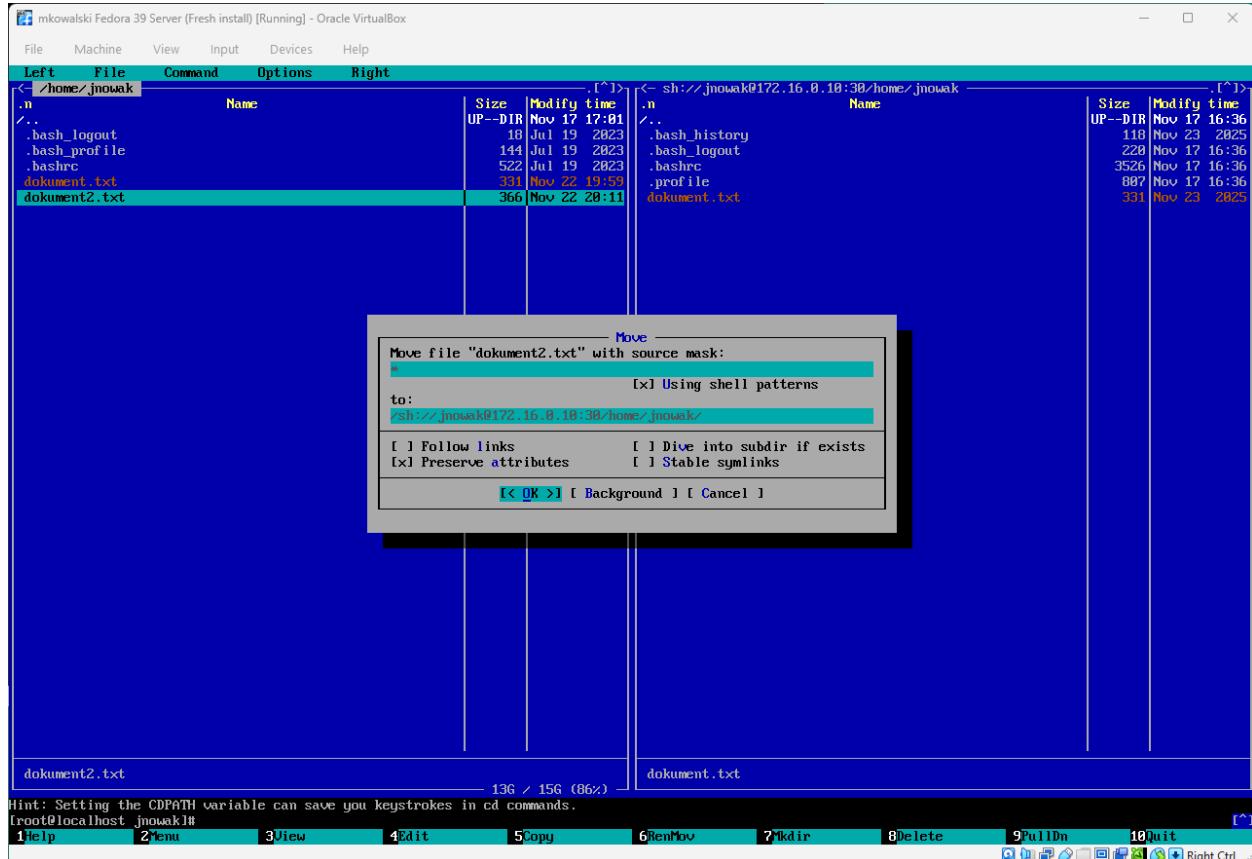
Zrzut ekranu 39 Opcja *Shell link...* otwierająca połączenie z maszyną zdalną w opcjach prawego panelu.



Zrzut ekranu 40 Wskazanie komputera zdalnego, z którym ma zostać ustanowione połączenie zdalne.

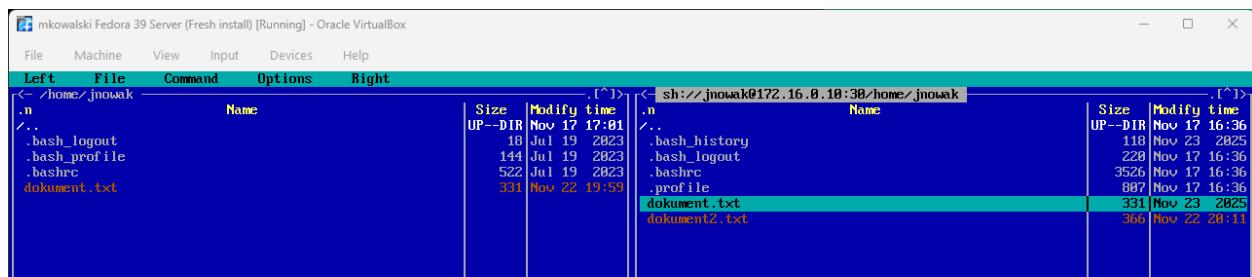
Po podaniu poprawnego hasła zobaczymy zawartość dysku komputera zdalnego (tylko te pliki, do których ma dostęp użytkownik *jnowak*).

Aby skopiować plik, odszukujemy go najpierw w panelu naszego komputera, po czym wciskamy na klawiaturze przycisk *F5*, otwierający okienko pozwalające na wskazanie lokalizacji docelowej.

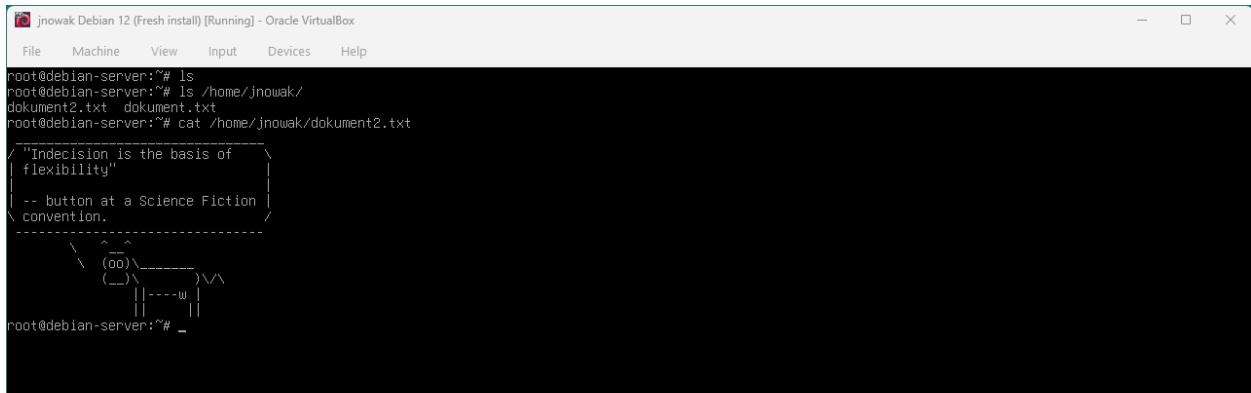


Zrzut ekranu 41 Kopiowanie pliku na maszynę zdalną.

Po zatwierdzeniu operacji, plik zostanie skopiowany na serwer Debian, co możemy potwierdzić wykonując na nim polecenie *cat*.



Zrzut ekranu 42 Skopiowany na serwer zdalny plik.



```
jnowak@debian-server:~# ls
jnowak@debian-server:~# ls /home/jnowak/
dokument2.txt dokument.txt
jnowak@debian-server:~# cat /home/jnowak/dokument2.txt
/ "Indecision is the basis of
| flexibility"
| ... button at a Science Fiction
\ convention.
-----
\ ^ ^
\ (oo)\_____
(_)\_____)\/\
||----w |
|| |
jnowak@debian-server:~#
```

Zrzut ekranu 43 Potwierdzenie poprawności skopiowania pliku na serwer zdalny z systemem Debian.

## Zadanie 13. Połączenie zdalne SSH z wykorzystaniem aplikacji PuTTY.

Połączenie zdalne z serwerem można także nawiązać z wykorzystaniem aplikacji *PuTTY*. Aby z systemu gospodarza połączyć się z maszyną Debian, możemy wykorzystać do tego trzeci interfejs sieciowy, podłączony do adaptera w trybie Host-Only. Trzeba go jednak najpierw skonfigurować, na przykład włączając uzyskiwanie danych adresowych przez DHCP.

```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet dhcp
iface enp0s3 inet6 auto

auto enp0s8
iface enp0s8 inet static
    address 172.16.0.10/24
iface enp0s8 inet6 static
    address fd00::10/64

auto enp0s8:1
iface enp0s8:1 inet static
    address 172.16.0.11/24

auto enp0s9
iface enp0s9 inet dhcp
~
~
~
```

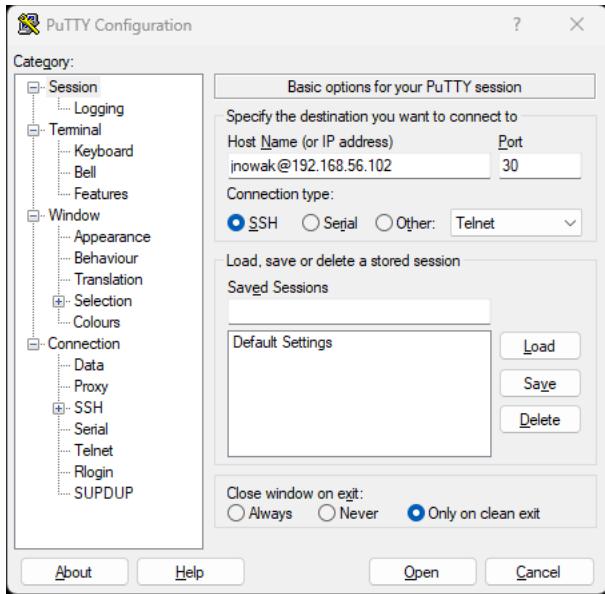
Zrzut ekranu 44 Konfiguracja interfejsu sieciowego adaptera w trybie host-only.

Po restarcie usługi możemy sprawdzić jakie dane zostały nadane.

```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@debian-server:~# systemctl restart networking
root@debian-server:~# ip a s enp0s9
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:be:30:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic enp0s9
        valid_lft 597sec preferred_lft 597sec
    inet6 fe80::a00:27ff:febe:309d%enp0s9 scope link
        valid_lft forever preferred_lft forever
root@debian-server:~# ~
```

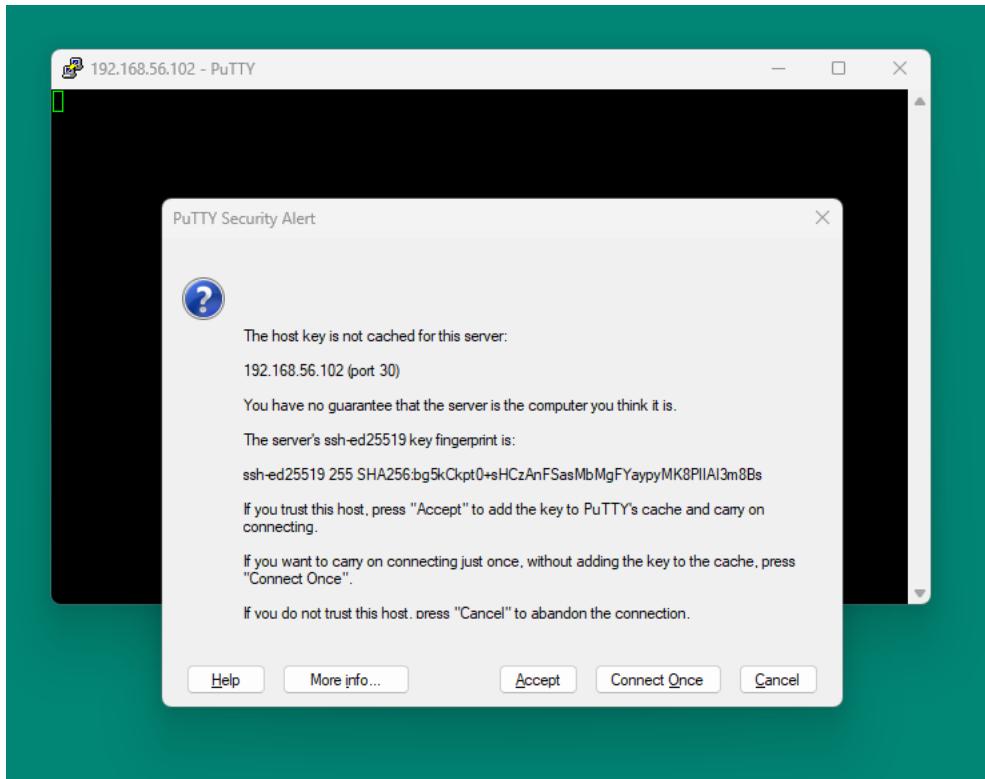
Zrzut ekranu 45 Sprawdzenie danych adresowych nadanych w ramach interfejsu enp0s9, podłączonego do sieci w trybie Host-Only.

Wykorzystując uzyskane dane adresowe, możemy podłączyć się do serwera z systemu gospodarza przez aplikację *PutTY*.



Zrzut ekranu 46 Interfejs aplikacji PutTY.

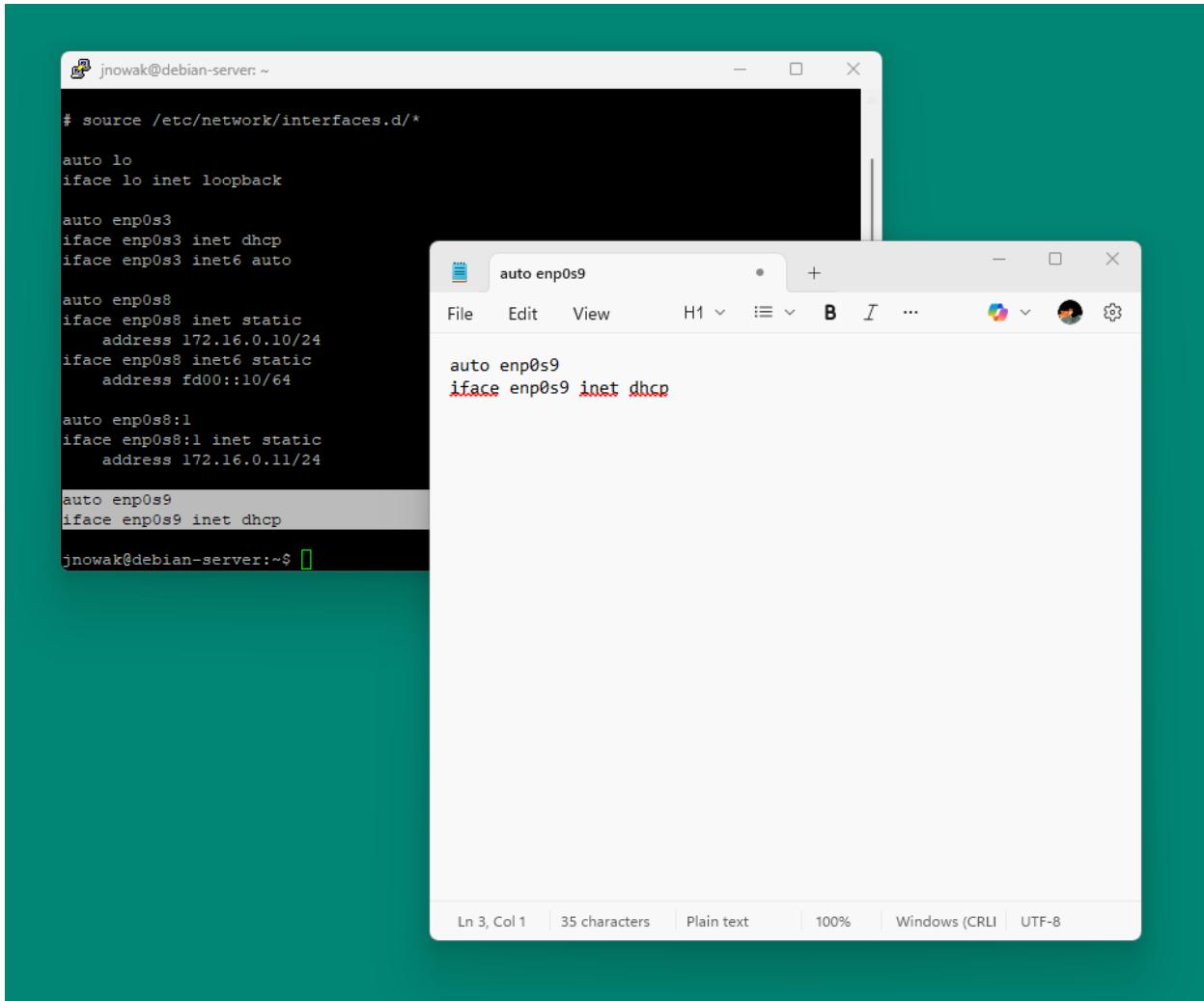
W znanym już formacie, *login@host*, przekazujemy nazwę hosta, w osobnym polu wskazujemy numer portu i otwieramy połaczenie przyciskiem „Open”.



Zrzut ekranu 47 Nawiązywanie połączenia z serwerem Debian.

Ponieważ łączymy się do serwera po raz pierwszy, jesteśmy pytani o chęć dodania go do listy zaufanych urządzeń. Jeżeli jest to połączenie jednorazowe, można wybrać opcję *Connect Once*, która nie zapisze certyfikatu.

Korzystanie z aplikacji *PutTY* ma ważną zaletę – możemy łatwo kopiować tekst między konsolą podłączoną do serwera zdalnego, a systemem gospodarza. Przykładowo, możemy skopiować zawartość pliku */etc/network/interfaces* do notatnika. Zamiast standardowego *CTRL+C*, musimy jednak skorzystać z kombinacji *CTRL+SHIFT+C*.

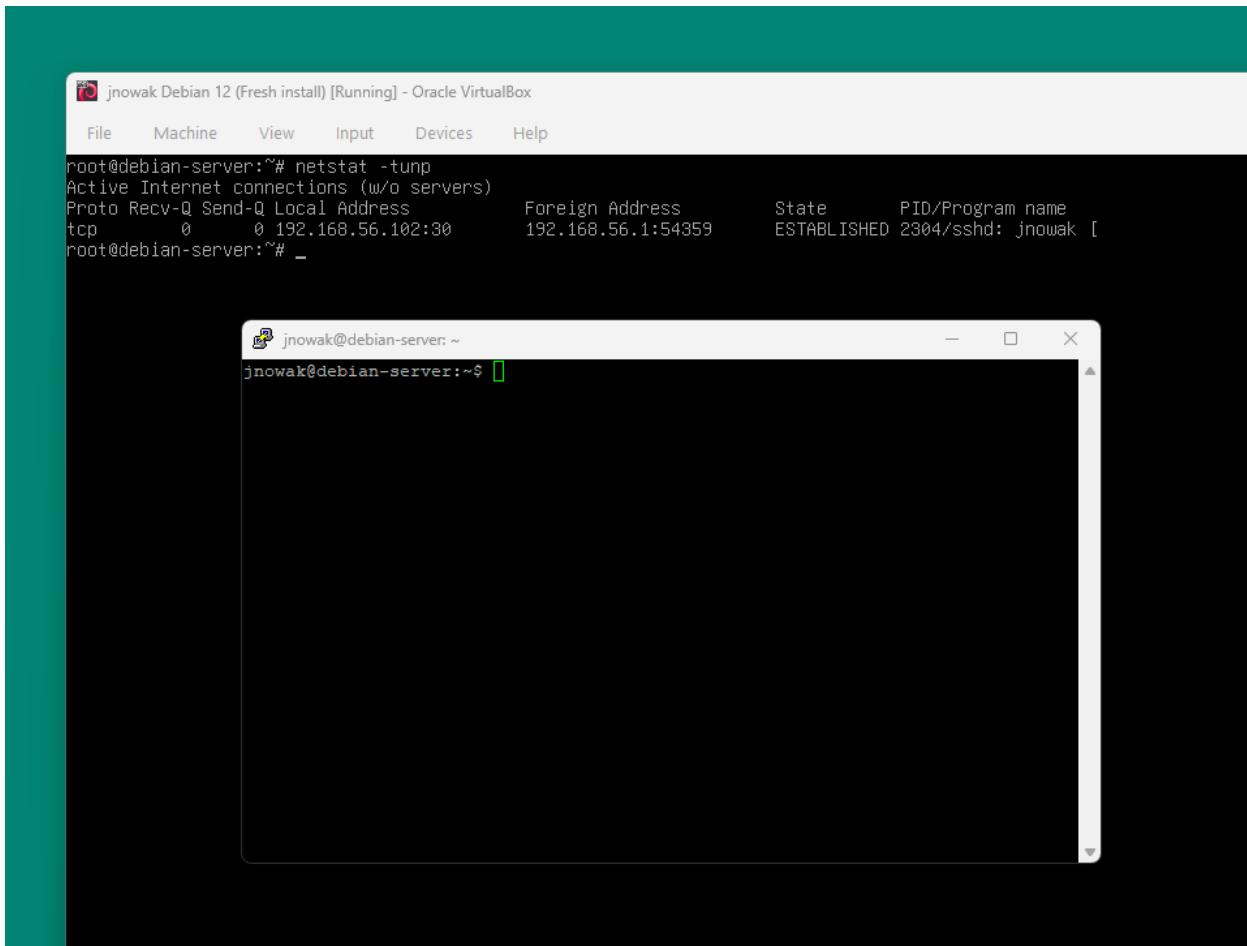


Zrzut ekranu 48 Przykład kopiowania tekstu między konsolą PutTY (SSH), a notatnikiem w systemie gospodarza.

## Zadanie 14. Polecenie *netstat*.

Polecenie *netstat* pozwala na wyświetlanie operacji i statystyk sieciowych.

Aby wyświetlić listę wykonywanych z aplikacji połączeń, skorzystamy z polecenia *netstat -tunp* (*TCP, UDP, numerycznie, PID programu*).



```
jnowak@debian-server:~# netstat -tunp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 192.168.56.102:30       192.168.56.1:54359  ESTABLISHED 2304/sshd: jnowak [
```

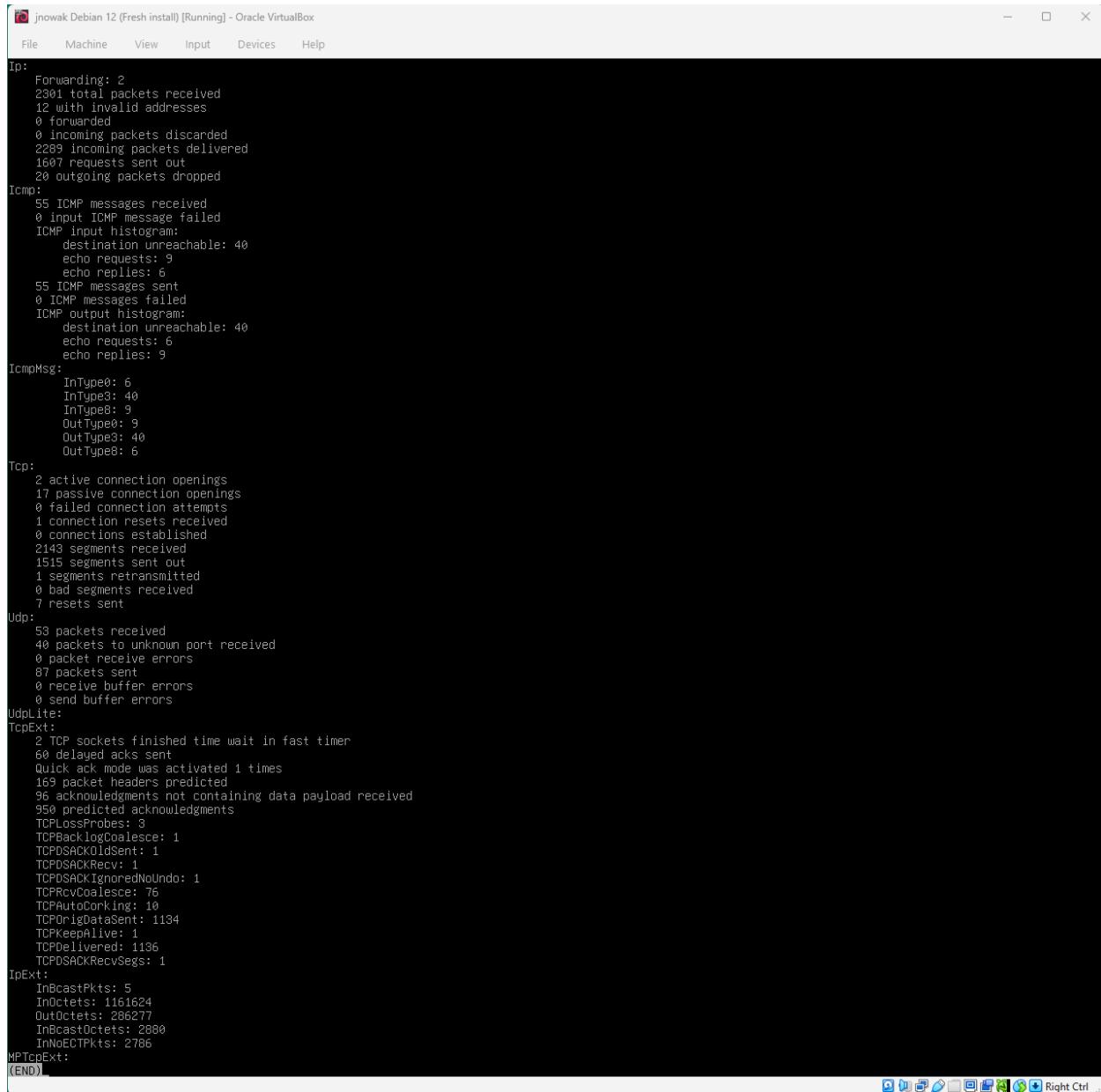
Zrzut ekranu 49 Widniejące otwarte połączenie w trakcie korzystania z dostępu zdalnego przez SSH.

Do wyświetlenia uruchomionych aplikacji serwerowych, które nasłuchują połączeń przychodzących wykorzystamy polecenie  
*netstat -tunlp (TCP, UDP, numerycznie, listening (nasłuchujące), PID programu).*

```
jnowak Debian 12 (Fresh install) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@debian-server:~# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp        0      0 0.0.0.0:30              0.0.0.0:*              LISTEN    1125/sshd: /usr/sbi
tcp6       0      0 ::1:30                 ::*:*                  LISTEN    1125/sshd: /usr/sbi
udp        0      0 0.0.0.0:68              0.0.0.0:*              2135/dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*              1947/dhclient
root@debian-server:~#
```

Zrzut ekranu 50 Lista uruchomionych aplikacji serwerowych, które nasłuchują połączeń przychodzących z sieci.

Statystyki odnośnie działania protokołów TCP/UDP wyświetlimy poleceniem `netstat -s` (statystyki).



```
File Machine View Input Devices Help
Ip:
  Forwarding: 2
  2301 total packets received
  12 with invalid addresses
  0 forwarded
  0 incoming packets discarded
  2289 incoming packets delivered
  1607 requests sent out
  20 outgoing packets dropped
Icmp:
  55 ICMP messages received
  0 input ICMP message failed
  ICMP input histogram:
    destination unreachable: 40
    echo requests: 9
    echo replies: 6
  55 ICMP messages sent
  0 ICMP messages failed
  ICMP output histogram:
    destination unreachable: 40
    echo requests: 6
    echo replies: 9
IcmpMsg:
  InType0: 6
  InType3: 40
  InType8: 9
  OutType0: 9
  OutType3: 40
  OutType8: 6
Tcp:
  2 active connection openings
  17 passive connection openings
  0 failed connection attempts
  1 connection resets received
  0 connections established
  2143 segments received
  1515 segments sent out
  1 segments retransmitted
  0 bad segments received
  7 resets sent
Udp:
  55 packets received
  40 packets to unknown port received
  0 packet receive errors
  87 packets sent
  0 receive buffer errors
  0 send buffer errors
Udplite:
TcpExt:
  2 TCP sockets finished time wait in fast timer
  60 delayed acks sent
  Quick ack mode was activated 1 times
  169 packet headers predicted
  96 acknowledgments not containing data payload received
  950 predicted acknowledgments
  TCPLOSSProbes: 3
  TCPBacklogCoalesce: 1
  TCPDSACK0OldSent: 1
  TCPDSACKRecv: 1
  TCPDSACKIgnoredNoUndo: 1
  TCPRcvCoalesce: 76
  TCPAutoCorking: 10
  TCPOrigDataSent: 1134
  TCPKeepAlive: 1
  TCPDelivered: 1136
  TCPDSACKRecvSegs: 1
IpExt:
  InBroadcastPkts: 5
  InOctets: 1161624
  OutOctets: 286277
  InBroadcastOctets: 2880
  InNoECTPkts: 2786
MPTcpExt:
(END)
```

Zrzut ekranu 51 Wyświetlenie statystyk działania protokołów TCP/UDP.

Informacje statystykach (w tym błędach) transmisji danych w ramach kanału RX/TX wyświetlimy poleceniem *ip -s link*.

```
root@localhost:~# ip -s link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX: bytes packets errors dropped missed mcast
        509128    5048      0      0      0
    TX: bytes packets errors dropped carrier collsns
        509128    5048      0      0      0
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:fa:1b:58 brd ff:ff:ff:ff:ff:ff
    RX: bytes packets errors dropped missed mcast
        190486    2430      0      0      0
    TX: bytes packets errors dropped carrier collsns
        375762    8147      0      0      0
    altname enp2s0
root@localhost:~# _
```

Zrzut ekranu 52 Wyświetlenie statystyk w ramach kanału RX/TX.