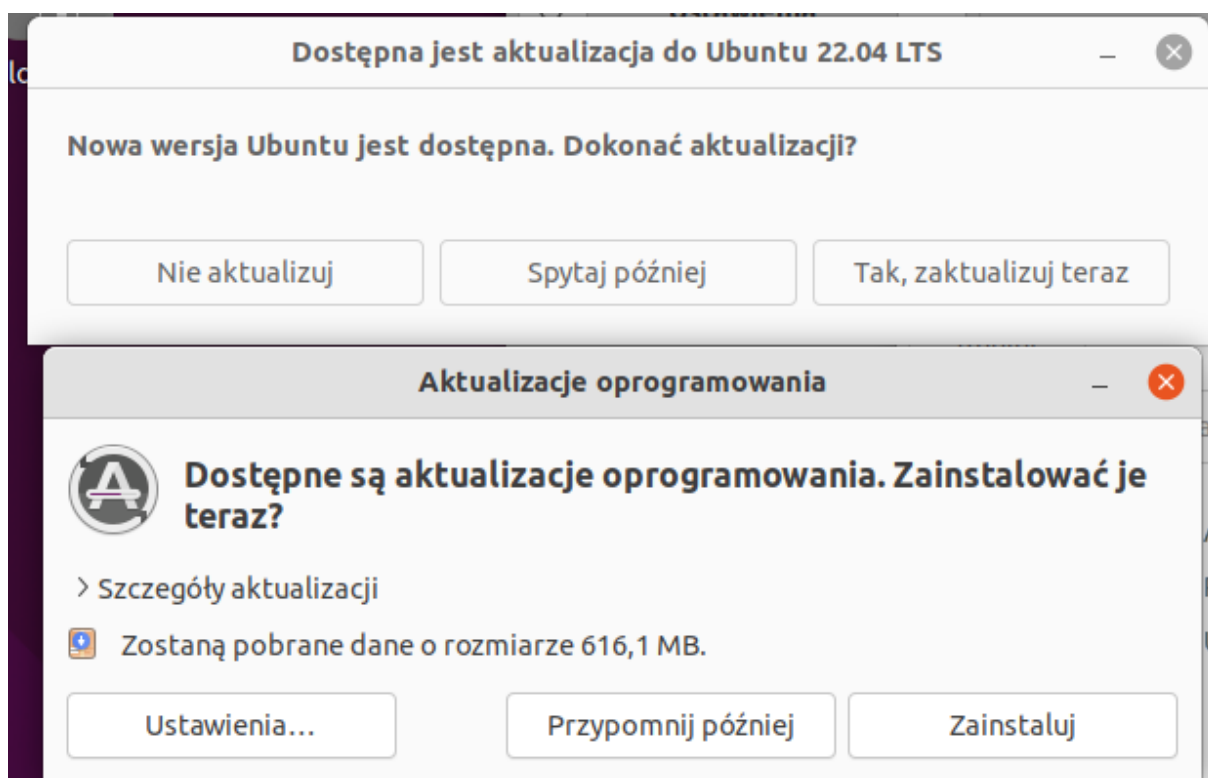


## Temat: Ćwiczenie 1 Interfejsy sieciowe klienta i serwera

### W jakim celu?

Rozpoczynając pracę zastajemy system w określonym momencie – należy go zaktualizować przynajmniej niezbędnymi pakietami w bieżącej wersji. Zaleca się aktualizację pełną do bieżącej dystrybucji( jak na rys górne SPYTAJ PÓŹNIEJ dolne ZAINSTALUJ).



### Co się nauczymy?

Panować nad tym jakiego typu interfejs sieciowy pracuje w jakim trybie. Eliminować błędy ustawień interfejsów. Jest nam to potrzebne do kolejnych ćwiczeń w których zarządzanie interfejsami sieciowymi jest elementarnie konieczne do sprawnego zarządzania serwerem.

### Co mamy?

Na wstępie mamy :

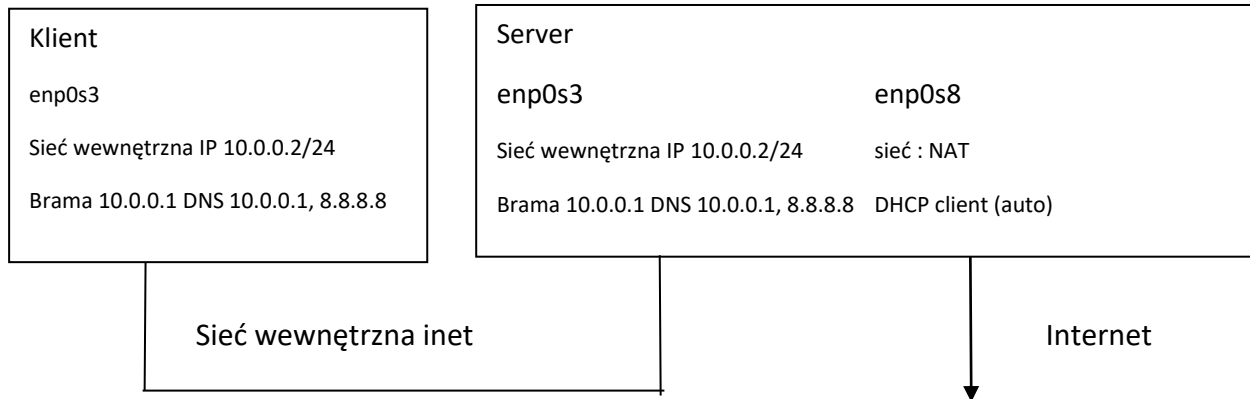
- maszynę wirtualną klienta Ubuntu 21.10 w wersji z domyślnym GUI czyli GNOME zainstalowanym na Virtualboxie jako legacy BIOS z kontem administracyjnym ad mającym hasło 123.Dysk twardy maszyny jest w formacie vdi i ma 50GB.
- maszynę wirtualną serwera Ubuntu 21.10 w wersji bez GUI zainstalowanym na Virtualboxie jako legacy BIOS z kontem administracyjnym ad mającym hasło 123.Dysk twardy maszyny jest w formacie vdi i ma 40GB.

Maszyny startowe (początek ćwiczenia): [klient Linux](#) [serwer Linux](#)

Wynikowe maszyny (koniec ćwiczenia): [klient Linux](#) [serwer Linux](#)

## Polecenia ćwiczenia:

- Skonfiguruj interfejsy serwera w wirtualboxie i systemie operacyjnym wg poniższego diagramu:

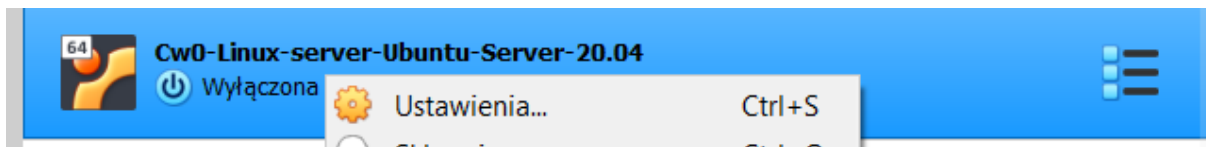


- Zaktualizuj system i sprawdź łączność pomiędzy klientem a serwerem
- Nazwij komputer ( maszynę wirtualną ) klienta – klient a serwera : server

## Jak zrobić

Przed włączeniem maszyny serwera

Prawy klawisz myszy na nazwie maszyny ustawienia



Przechodzimy do zakładki sieć -> karta 1



Zmieniamy na sieć wewnętrzna – w systemie będzie taka karta nosiła nazwę `enp0s3` i będzie służyła do połączenia z klientem

Przechodzimy na Zakładkę karta 2 - włącz karta sieciowa – podłączona do NAT

Karta będzie służyć serwerowi do połączeń do Internetu a nosiła nazwę enp0s8

Zatwierdzamy i włączamy maszynę. Po uruchomieniu logujemy się na użytkownika ad z hasłem 123.

Korzystamy z poradnika np. :

<https://www.2daygeek.com/enable-disable-up-down-nic-network-interface-port-linux/>

### *Włączanie i wyłączenie interfejsów serwera*

Na serwerze w konsoli po zalogowaniu:

`sudo -i` (opcjonalne zalogowanie się na super użytkownika konsola zmieni początek linii z \$ na # - niezalecane)

`sudo hostname server` (nazwanie komputera serwera „server” – analogicznie na kliencie)

### *Nazwanie komputera (maszyny wirtualnej) klienta -klient i serwera - server*

`ip a` (karty sieciowe enp0s3 i enp0s8 umownie LAN WAN )

`sudo ip link set enp0s3 down` (wyłączenie karty LAN)

`sudo ip link set enp0s8 up` (wyłączenie karty WAN)

`ping 10.0.0.1` (sprawdzamy czy działa połączenie do klienta )

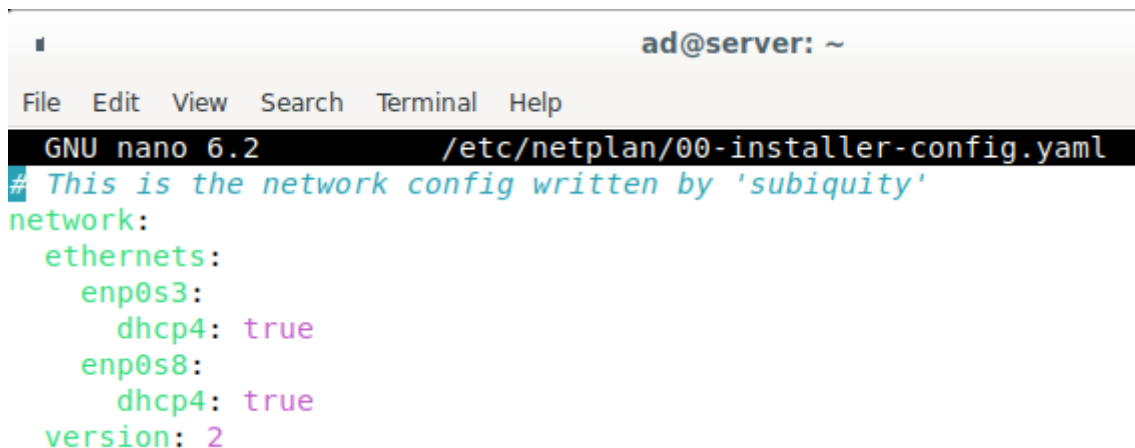
`ping 8.8.8.8` (sprawdzamy czy działa połączenie internetu)

`Network unreachable` (uzyskujemy komunikat )

Tak powinno być – żeby to naprawić musimy wykonać kolejny punkt

### *Edycja netplan*

`sudo nano /etc/netplan/00-installer-config.yaml`



```
ad@server: ~
File Edit View Search Terminal Help
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: true
  version: 2
```

(będziesz miał inny interfejs graficzny pliku lecz interesuje nas zawartość pliku)

dopisać po `dhcp4:true`

`enp0s8:`

`dhcp4:true`

CRTL+X zapisać i wyjść z edycji z pliku

`sudo netplan try`

jeśli nie ma błędów naciśnij ENTER –zmiany zostaną wdrożone. Jeśli będą błędy to należy ponownie do skutku edytować powyższy plik nie używając tabulatora i zwracając uwagę na to by był identycznej zawartości.

`sudo ip link set enp0s8 down` (opcjonalnie)

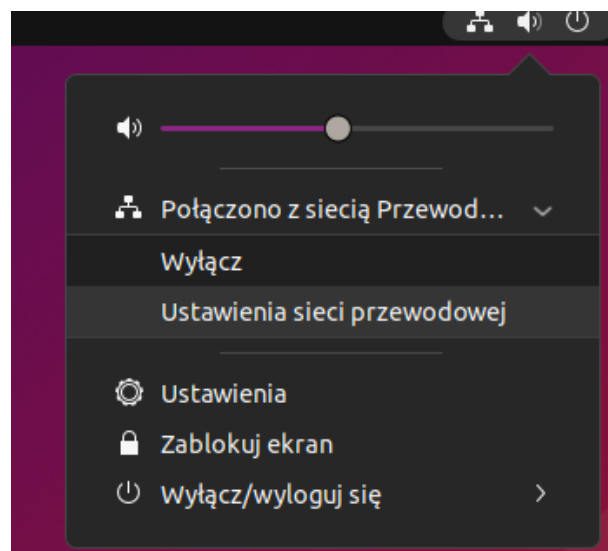
`sudo ip link set enp0s8 up` (opcjonalnie)

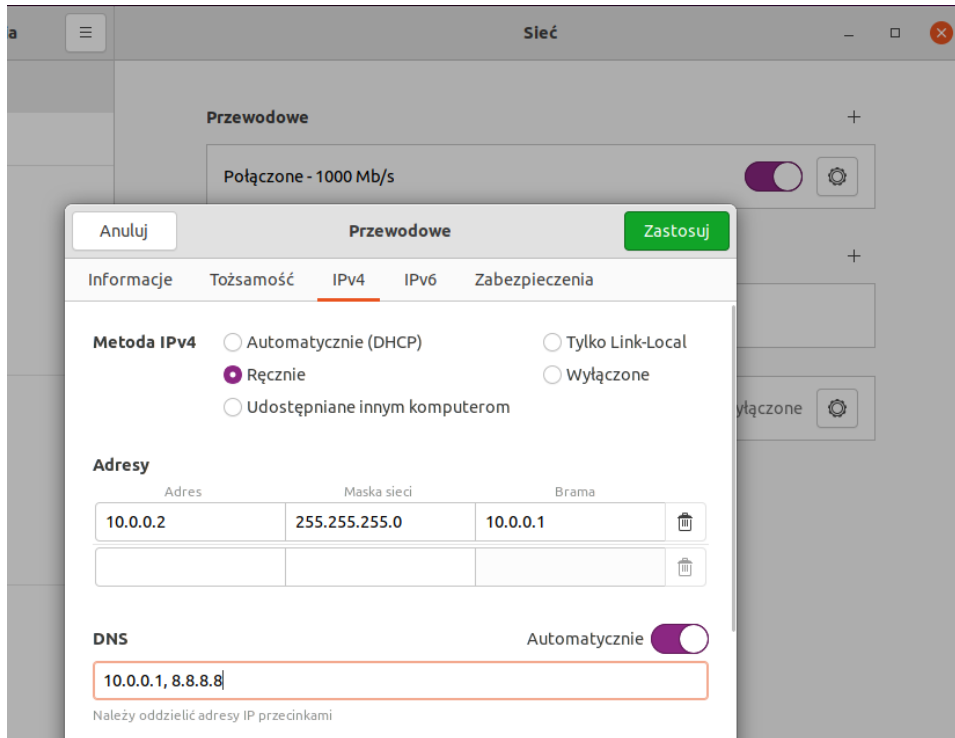
`ping 8.8.8.8`

`ping 10.0.0.2`

działa nam już połączenie do Internetu ale nie działa połączenie do klienta. Teraz to naprawimy

na kliencie w GUI ustaw stały adres IP 10.0.0.2 z maską 24bitowa





Wracamy do konfiguracji servera

```
sudo nano /etc/netplan/00-installer-config.yaml
```

(konfiguracja karty LAN jako 10.0.0.1)

zmienić z :

```
# This is the network config written by 'subiquity'
```

```
network:
```

```
  ethernets:
```

```
    enp0s3:
```

```
      dhcp4: true
```

```
    enp0s8:
```

```
      dhcp4: true
```

```
version: 2
```

```
na
```

```
network:
```

```
  ethernets:
```

```
    enp0s3:
```

```
      dhcp4: true
```

```
    enp0s8:
```

```
      dhcp4: no
```

```
      addresses: [10.0.0.1/24]
```

```
# gateway4: 10.0.0.1
nameservers:
addresses: [10.0.0.1,8.8.8.8]
version: 2
```

## CRTL+X zapisać

```
sudo netplan try
```

zatwierdzamy klawiszem ENTER

```
ip link set enp0s3 down (opcjonalnie)
```

```
ip link set enp0s3 up (opcjonalnie)
```

```
ip a
```

powinno dać rezultat:

```
%2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 08:00:27:05:76:3c brd ff:ff:ff:ff:ff:ff
inet 10.0.0.1/24 brd 10.0.0.255 scope global dynamic enp0s3
    valid_lft 86382sec preferred_lft 86382sec
inet6 fe80::a00:27ff:fe05:763c/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 08:00:27:09:38:e7 brd ff:ff:ff:ff:ff:ff
inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s8
    valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe09:38e7/64 scope link
    valid_lft forever preferred_lft forever
```

wówczas

ping 10.0.0.2 działa podobnie jak ping 8.8.8.8

## Podsumowanie

Skonfigurowaliśmy solidne podłoże do kolejnych ćwiczeń i możemy już panować nad siecią w systemie linux.

Niestety nie możemy pingować komputerów po nazwie co naprawimy w kolejnym ćwiczeniu. Nie możemy też korzystać na serwerze z graficznego interfejsu użytkownika. To z punktu widzenia stabilności i zużycia zasobów dobrze ale mało wygodnie. Dodatkowo klientowi ręcznie należy przypisywać parametry sieciowe postaramy się to zautomatyzować w kolejnym ćwiczeniu.