

Akademia Nauk Stosowanych w Nowym Sączu

Wydział Nauk Inżynierjnych

Systemy operacyjne – projekt

studia stacjonarne

semestr letni 2023/2024

Temat projektu:

1. Zaprojektować infrastrukturę informatyczną na potrzeby firmy Binary-Builders. Realizacja serwerowa w oparciu o system operacyjny Linux, np. Fedora Server 39, stacje klienckie np. Linux MINT.
2. Wdrożyć niezbędne usługi wynikające z założeń takie jak: SSH, DHCP, DNS, HTTP/S, motor bazodanowy (MySql)+PHP+phpMyAdmin, CMS WordPress, RAID, SAMBA, SQUID, Postfix(SMTP) + Dovecot(POP/IMAP), oraz wybraną usługę. Wdrożyć automatyzację przy użyciu skryptu np. Bash, oraz usługi cron.
3. Cele projektu zweryfikować z założeniami zapisanymi w dokumencie „Szczegółowy zarys projektu”.

Imię i nazwisko:

Maciej Wójs

Data oddania:

2 czerwca 2024

Nr grupy:

L3

Ocena:

Spis treści

1 Założenia projektowe – wymagania	4
2 Opis użytych technologii	5
2.1 SSH (Secure Shell)	5
2.2 DHCP (Dynamic Host Configuration Protocol)	5
2.3 DNS (Domain Name System)	5
2.4 HTTP/S (Hypertext Transfer Protocol/Secure)	5
2.5 MySQL	5
2.6 PHP	5
2.7 phpMyAdmin	5
2.8 CMS WordPress	5
2.9 RAID (Redundant Array of Independent Disks)	6
2.10 SAMBA	6
2.11 SQUID	6
2.12 Postfix (SMTP) + Dovecot (POP/IMAP)	6
2.12.1 Postfix	6
2.12.2 Dovecot	6
2.13 Automatyzacja za pomocą skryptów Bash i usług cron	6
2.13.1 Skrypty Bash	6
2.13.2 cron	6
3 Schemat logiczny projektowanej infrastruktury sieciowej	7
4 Procedury instalacyjne poszczególnych usług	8
4.1 Instalacja systemu klienta – Linux Mint	8
4.1.1 Proces instalacji	8
4.1.2 Wstępna konfiguracja systemu	12
4.2 Instalacja serwera – Fedora 40	13
4.2.1 Proces instalacji	13
4.2.2 Wstępna konfiguracja	19
4.3 Konfiguracja SSH	23
4.4 Nazwa serwera – hostname	25
4.5 DNS – instalacja i konfiguracja	25
4.6 DHCP – instalacja i konfiguracja	32
4.7 RAID 5 – konfiguracja	35
4.8 Samba – instalacja i konfiguracja	39
4.9 HTTP – instalacja i konfiguracja	43
4.10 PHP – instalacja i konfiguracja	46
4.11 mariadb – instalacja i konfiguracja	47
5 Testy działania wdrożonych usług	49
5.1 DNS	49
5.2 DHCP	49
5.3 Raid 5	50
5.4 Samba	50
5.5 HTTP	51
5.6 PHP	52
5.7 MySQL	52

6 Kod skryptu BASH, oraz tablica crontab	53
7 Wnioski	53
8 Literatura	54

Spis rysunków

1	Schemat logiczny sieci	7
2	Tworzenie nowej maszyny wirtualnej	8
3	Przydzielanie zasobów maszynie wirtualnej	8
4	Określenie rozmiaru dysku wirtualnego.	9
5	Podsumowanie konfiguracji maszyny wirtualnej	9
6	Rozpoczęcie instalacji Linux Mint	10
7	Wybór trybu instalacji na dysku twardym.	10
8	Tworzenie konta użytkownika	11
9	Zakończenie instalacji systemu Linux Mint.	11
10	Instalacja dodatków gościa	12
11	Aktualizacja pakietów	12
12	Podsumowanie maszyny wirtualnej Fedora 40	13
13	Dodanie pierwszej karty sieciowej	13
14	Dodanie drugiej karty sieciowej	14
15	Dodanie trzeciej karty sieciowej	14
16	Uruchomienie instalatora Fedory.	15
17	Rozpoczęcie instalacji Fedora	15
18	Wybór dysku instalacji	16
19	Ustawienie konta root	16
20	Stworzenie użytkownika	17
21	Ekran postępującej instalacji	17
22	Ekran przed restartem do systemu.	18
23	Zainstalowany system Fedora 40	18
24	Konfiguracja dnf	19
25	Aktualizacja pakietów	20
26	plik /etc/default/grub przed zmianą	20
27	plik /etc/default/grub po zmianie	21
28	Zastosowanie zmian po edycji grub	21
29	Zwiększenie wygody wpisywania haseł	22
30	Efekt działania zmiany ustawień	22
31	konfiguracja karty sieciowej	23
32	Konfiguracja PuTTY	23
33	Próba podłączenia poprzez PuTTY	24
34	Wynik połączenia poprzez PuTTY	24
35	Zmiana nazwy serwera	25
36	Edycja /etc/hosts	25
37	Instalacja DNS	26
38	Kopia zapasowa pliku konfiguracyjnego DNS	27
39	zawartość named.conf	29
40	zawartość pliku strefy podstawowej	30
41	zawartość pliku strefy dla przeszukiwania wstępznego	31
42	Uruchomienie usługi DNS	32
43	Instalacja DHCP	32
44	Konfiguracja DHCP	33
45	Instalacja DHCP	34
46	Dodanie dysków w VirtualBox	35
47	Stworzenie macierzy raid 5	36
48	Partycjonowanie macierzy narzędziem cfdisk	36

49	Stworzenie dwóch partycji	37
50	Wynik partycjonowania	37
51	Przygotowanie ścieżek do montowania	38
52	Edycja /etc/fstab	39
53	Samba – instalacja	39
54	Edycja pliku /etc/samba/smb.conf	40
55	Samba – ustawienia SELinux oraz firewall	42
56	Edycja konfiguracji DNS	42
57	Instalacja serwera HTTP	43
58	Edycja /etc/httpd/conf/httpd.conf – 1	43
59	Edycja /etc/httpd/conf/httpd.conf – 2	44
60	Edycja /etc/httpd/conf/httpd.conf – 3	44
61	Edycja /etc/httpd/conf/httpd.conf – 4	45
62	Strona html – domyślna strona serwera	45
63	PHP – instalacja	46
64	PHP – stworzenie strony internetowej	46
65	mariadb – instalacja usługi	47
66	mariadb – edycja pliku konfiguracyjnego	47
67	MySQL – instalacja część pierwsza	48
68	MySQL – instalacja część druga	48
69	Test DNS	49
70	Instalacja DHCP	49
71	Test automatycznego montowania	50
72	Samba – próba podłączenia się do udziału na serwerze	50
73	Samba – wynik poprzedniego kroku	51
74	Test działania serwera WWW	51
75	Strona html + PHP	52
76	Test usługi mariadb (MySQL)	52

1 Założenia projektowe – wymagania

- a) Systemy operacyjne: Fedora Server 39 lub inny serwer z rodzinie Linux, oraz system kliencki np. Linux MINT.
- b) zarządzanie serwerem poprzez SSH, oraz emulator putty.exe
- c) nazwa serwera ma być zgodna z nazewnictwem: svrXX-firma, gdzie XX oznaczają dwie ostatnie cyfry numeru albumu wykonawcy, a firma to skrót nazwy swojej firmy (niepowtarzalny) – wymyślonej,
- d) na podstawie nazwy firmy należy założyć lokalną domenę o nazwie np. firma.ns i skonfigurować usługę DNS Server,
- e) adres IP serwera, zakres adresacji IP, oraz brama domyślna od strony sieci wewnętrznej VirtualBOXa (sieć LAN firmy) w której ma działać serwer DHCP ma mieć następujące wartości:

adres IP:	192.168.230.1/24,
zakres:	192.168.230.10–60
brama domyślna:	192.168.230.1
- f) należy utworzyć macierz dyskową programową na poziomie RAID 5 z dyskiem zapasowym. Uzyskać wypadkową pojemności macierzy 10GB. Przestrzeń macierzy podzielić na dwie równe partycje,
- g) Pierwszą partycję zamontować do punktu **/dysksieciowy**, a drugą do punktu **/kopie**. Zapewnić ich automatyczne montowanie podczas startu systemu,
- h) serwer ma udostępniać zasób sieciowy o adresie UNC **\sfs.firma.ns\dysk** odnoszący się do systemu plików **/dysksieciowy** (ppkt. g),
- i) należy wdrożyć usługę WEB Server z obsługą PHP, oraz serwer bazodanowy zarządzany przez phpMyAdmin, oraz CMS WordPress, skonfigurować UserDir dla WEB Serwer'a,
- j) dostęp do sieci Internet z sieci wewnętrznej ma się odbywać za pośrednictwem serwera PROXY(squid), a aktywność pracowników firmy ma być monitorowana,
- k) w firmie należy wdrożyć serwer pocztowy, oraz klienta mail,
- l) zapewnić aby popularne usługi były dostępne jako oddzielne nazwy hostów, jak np.:
 - **www.firma.ns** (serwer www),
 - **poczta.firma.ns** (serwer poczty),
 - **sfs.firma.ns** (serwer samby),
- m) wdrożyć automatyczną archiwizację systemu plików /home zawierającego katalogi użytkowników. Archiwizacja ma rozpoczynać się automatycznie codziennie o 21:00. W wyniku archiwizacji ma powstać plik **home_20240510.tar.gz** zapisany w **/kopie** (ppkt. g)
- n) Dodatkowo wdrożyć dowolną usługę, ale taką która nie była wdrażana podczas zajęć.

2 Opis użytych technologii

2.1 SSH (Secure Shell)

SSH to protokół sieciowy, który umożliwia bezpieczne zdalne logowanie oraz wykonywanie poleceń na odległym serwerze. Zapewnia szyfrowanie komunikacji, co chroni przed podsłuchiwaniem oraz atakami typu man-in-the-middle.

2.2 DHCP (Dynamic Host Configuration Protocol)

DHCP to protokół używany do automatycznego przydzielania adresów IP i innych parametrów konfiguracyjnych urządzeniom w sieci. Ułatwia zarządzanie siecią poprzez automatyczne przypisywanie ustawień.

2.3 DNS (Domain Name System)

DNS to system, który przekształca łatwe do zapamiętania nazwy domen (np. www.example.com) na adresy IP, które są wykorzystywane przez urządzenia sieciowe do komunikacji. DNS działa jak książka telefoniczna internetu.

2.4 HTTP/S (Hypertext Transfer Protocol/Secure)

HTTP to protokół komunikacyjny używany do przesyłania stron internetowych. HTTPS to jego bezpieczna wersja, która wykorzystuje TLS/SSL do szyfrowania danych, zapewniając poufność i integralność komunikacji między przeglądarką a serwerem.

2.5 MySQL

Popularny system zarządzania relacyjnymi bazami danych. Umożliwia przechowywanie i zarządzanie dużą ilością danych w strukturach tabelarycznych.

2.6 PHP

Skryptowy język programowania, często używany do tworzenia dynamicznych stron internetowych. PHP może komunikować się z bazami danych, takimi jak MySQL.

2.7 phpMyAdmin

Narzędzie webowe do zarządzania bazami danych MySQL. Umożliwia wykonywanie operacji na bazach danych za pomocą interfejsu graficznego.

2.8 CMS WordPress

WordPress to system zarządzania treścią (CMS), który pozwala na łatwe tworzenie i zarządzanie stronami internetowymi. Jest bardzo popularny ze względu na swoją elastyczność, prostotę obsługi oraz bogaty ekosystem wtyczek i motywów.

2.9 RAID (Redundant Array of Independent Disks)

RAID to technologia, która łączy kilka dysków twardych w jedną jednostkę logiczną w celu poprawy wydajności i/lub redundancji danych. Istnieje kilka poziomów RAID, z których każdy oferuje różne kombinacje wydajności i bezpieczeństwa danych.

2.10 SAMBA

SAMBA to pakiet oprogramowania, który umożliwia integrację systemów operacyjnych Linux/Unix z sieciami Windows. Pozwala na udostępnianie plików i drukarek w sieci oraz współpracę z domenami Windows (Active Directory).

2.11 SQUID

SQUID to serwer proxy i buforujący, który może przyspieszyć dostęp do zasobów internetowych poprzez przechowywanie często używanych danych w lokalnej pamięci podręcznej. Może również służyć jako filtr treści i narzędzie do monitorowania ruchu sieciowego.

2.12 Postfix (SMTP) + Dovecot (POP/IMAP)

2.12.1 Postfix

Serwer pocztowy obsługujący protokół SMTP, używany do wysyłania i odbierania wiadomości e-mail. Jest znany z wydajności i bezpieczeństwa.

2.12.2 Dovecot

Serwer IMAP i POP3 używany do odbierania i przechowywania wiadomości e-mail. Jest zoptymalizowany pod kątem wydajności i bezpieczeństwa, oferując wsparcie dla nowoczesnych standardów pocztowych.

2.13 Automatyzacja za pomocą skryptów Bash i usług cron

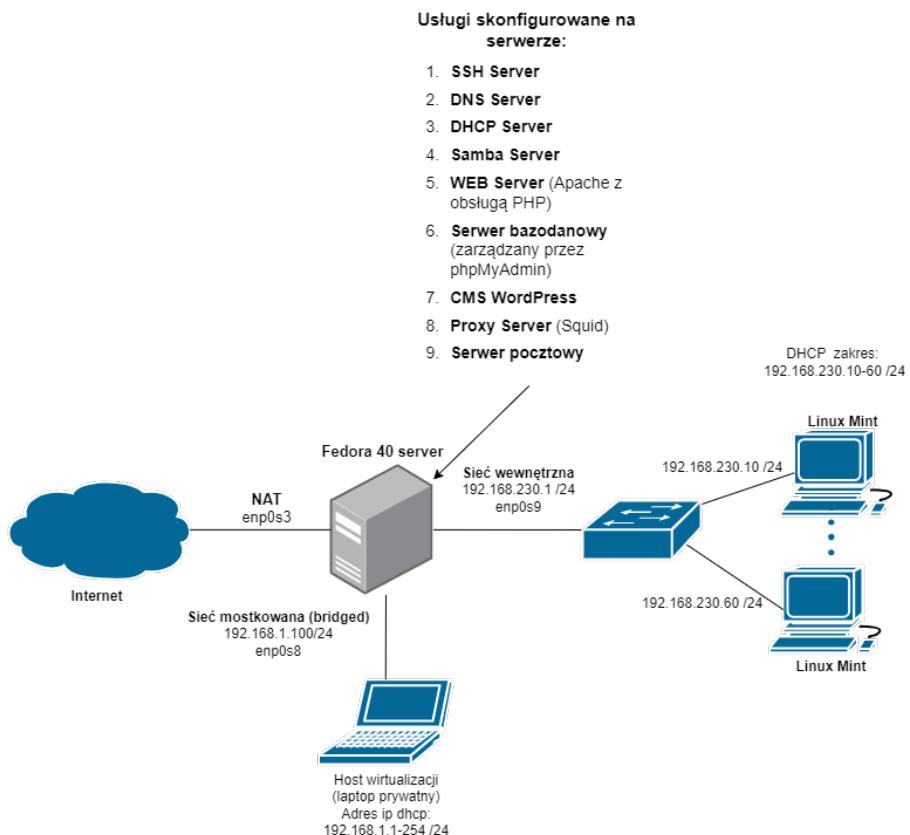
2.13.1 Skrypty Bash

Skrypty napisane w Bash (Bourne Again Shell) służą do automatyzacji zadań w systemach Unix/Linux. Mogą być używane do instalacji oprogramowania, konfiguracji systemu, zarządzania plikami i wielu innych zadań.

2.13.2 cron

Usługa systemowa w Unix/Linux, która pozwala na planowanie zadań do wykonania w określonym czasie lub regularnych odstępach czasu. Jest używana do automatyzacji zadań takich jak backup, aktualizacje systemu czy uruchamianie skryptów.

3 Schemat logiczny projektowanej infrastruktury sieciowej

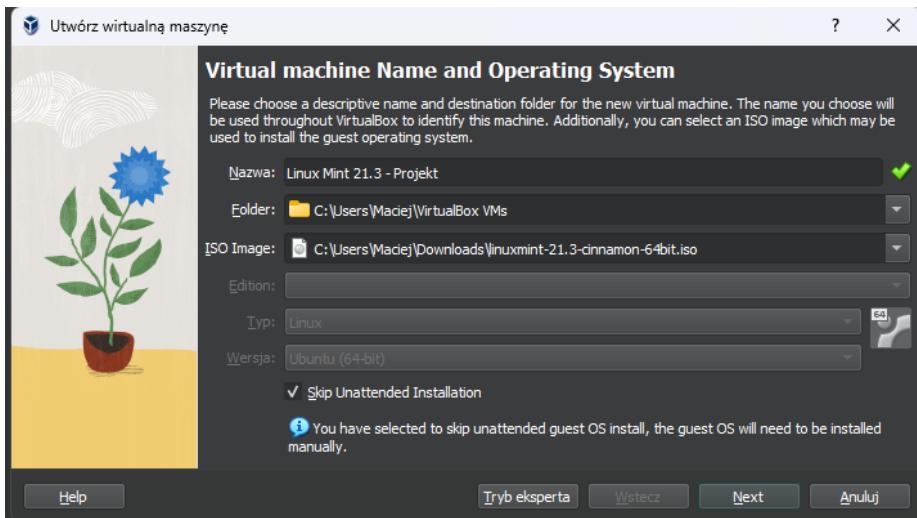


Rysunek 1: Schemat logiczny sieci

4 Procedury instalacyjne poszczególnych usług

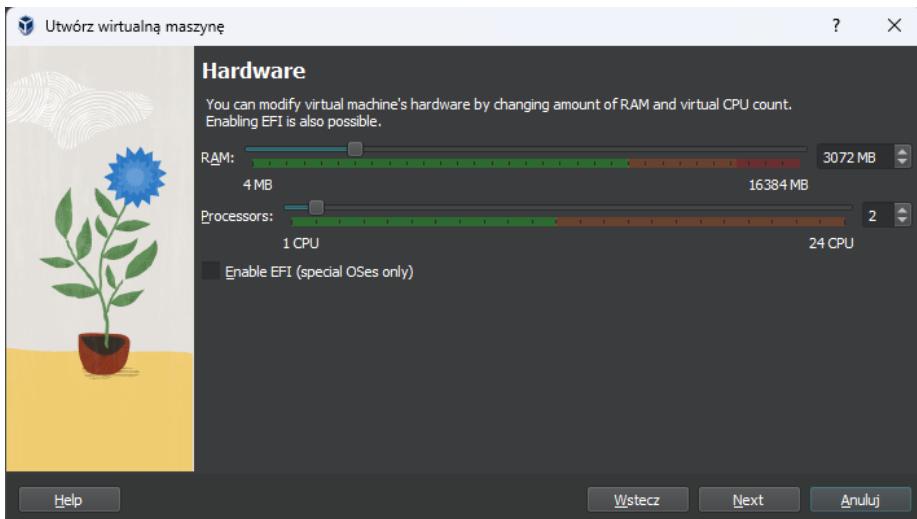
4.1 Instalacja systemu klienta – Linux Mint

4.1.1 Proces instalacji



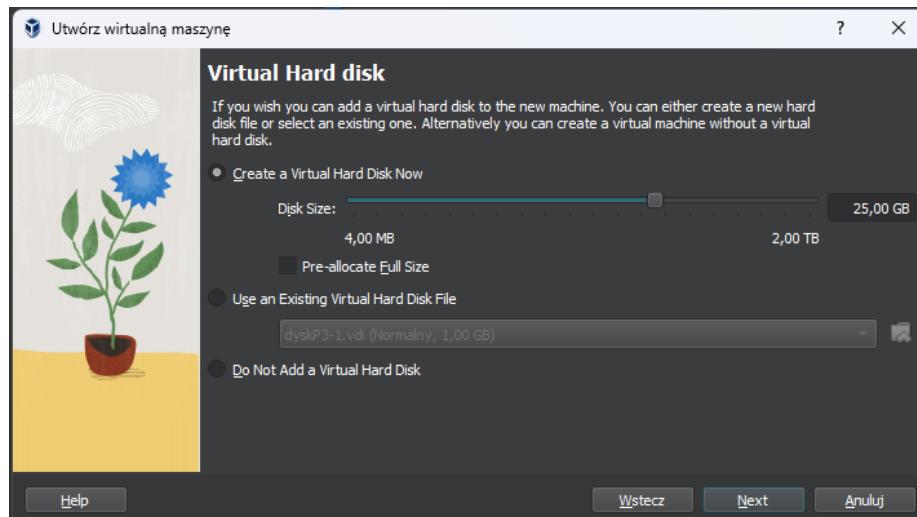
Rysunek 2: Tworzenie nowej maszyny wirtualnej. Ustawienia nazwy, lokalizacji dysku oraz wybór pliku ISO systemu operacyjnego.

Pierwszym krokiem jest utworzenie nowej maszyny wirtualnej (VM). W tym etapie określa się nazwę maszyny, lokalizację dysku, gdzie będzie przechowywana, oraz wybiera odpowiedni plik ISO z systemem Linux Mint.



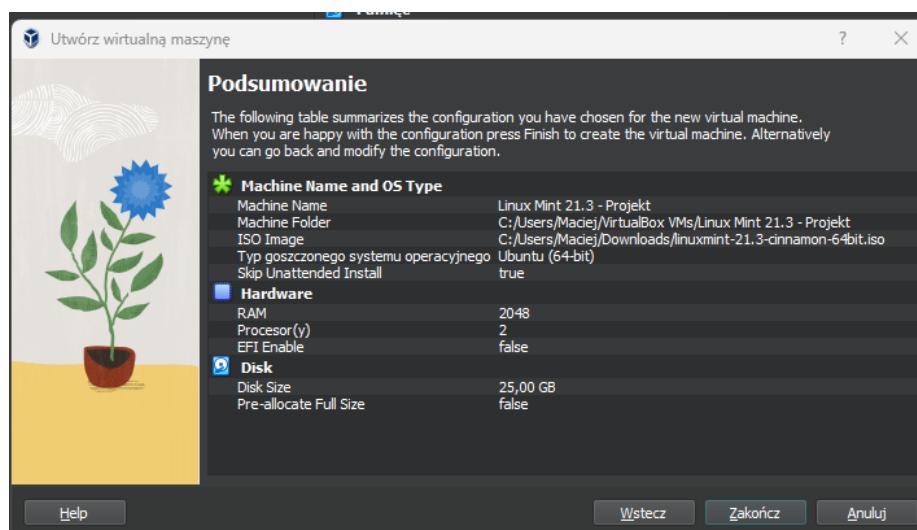
Rysunek 3: Przydzielanie zasobów maszynie wirtualnej, takich jak pamięć RAM i procesor.

W kolejnym kroku przydzielane są zasoby dla maszyny wirtualnej, w tym ilość pamięci RAM oraz liczba rdzeni procesora.



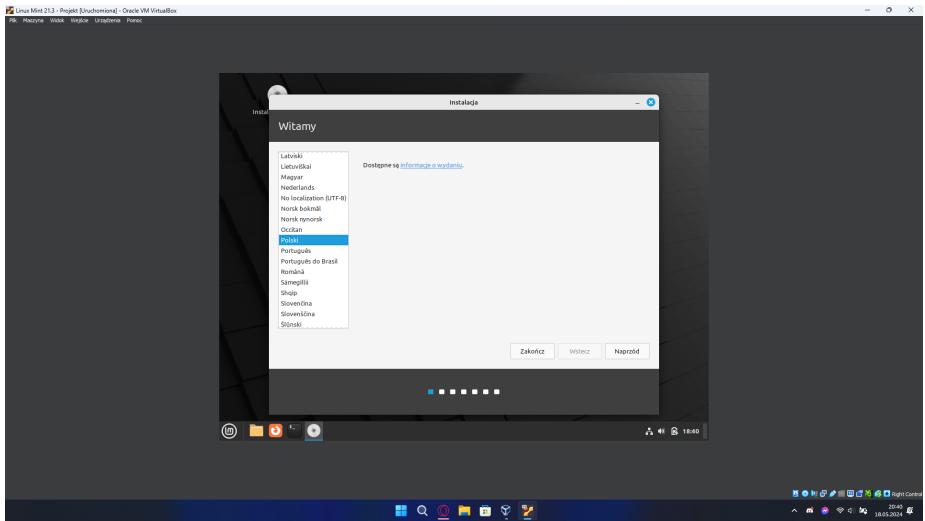
Rysunek 4: Określenie rozmiaru dysku wirtualnego.

Następnie należy zdefiniować rozmiar wirtualnego dysku twardego, który będzie używany przez maszynę wirtualną.



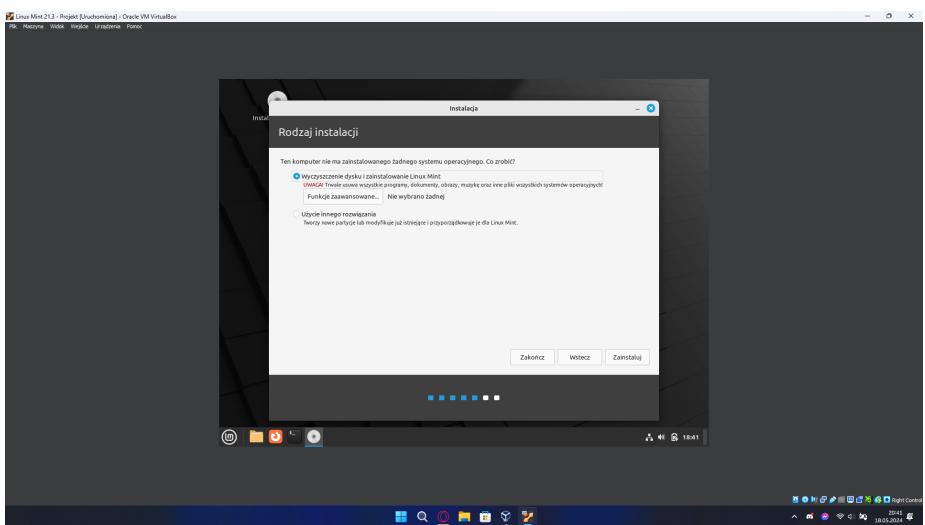
Rysunek 5: Podsumowanie konfiguracji maszyny wirtualnej przed rozpoczęciem instalacji systemu.

Po skonfigurowaniu wszystkich ustawień, wyświetlane jest podsumowanie zawierające wszystkie wybrane opcje dla nowo utworzonej maszyny wirtualnej.



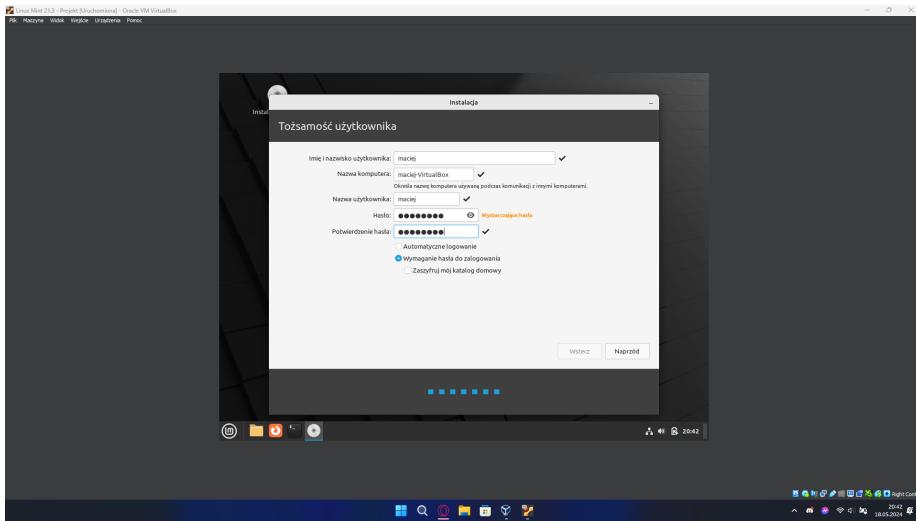
Rysunek 6: Rozpoczęcie instalacji Linux Mint – wybór języka instalacji.

Rozpoczyna się proces instalacji Linux Mint. Pierwszym krokiem jest wybór języka, który będzie używany podczas instalacji i w systemie.



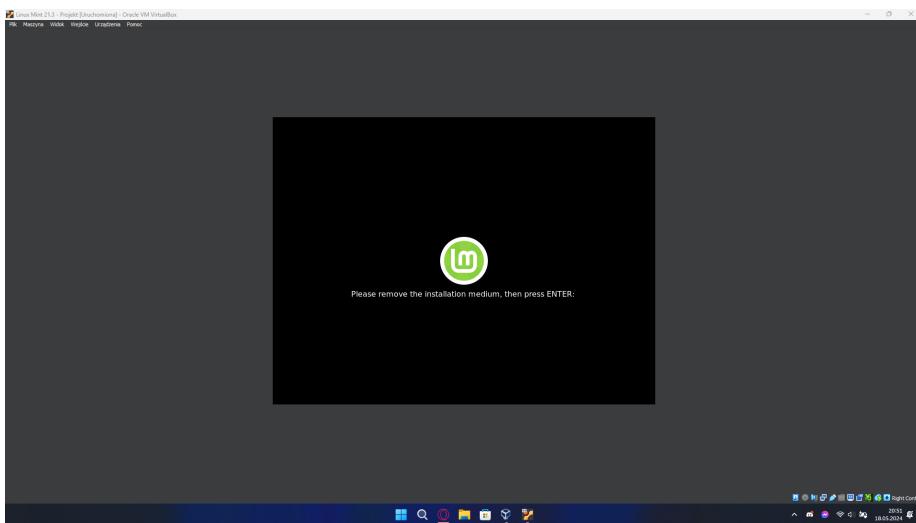
Rysunek 7: Wybór trybu instalacji na dysku twardym.

Następnie użytkownik wybiera sposób instalacji systemu na dysku twardym, na przykład automatyczne partycjonowanie lub ręczne tworzenie partycji.



Rysunek 8: Tworzenie konta użytkownika i konfiguracja podstawowych ustawień.

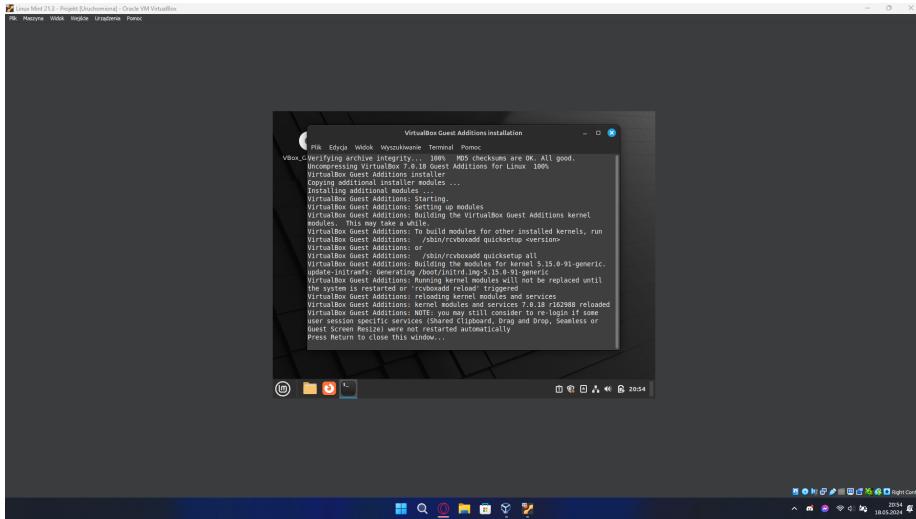
Kolejnym krokiem jest utworzenie konta użytkownika, wprowadzenie nazwy użytkownika, hasła oraz nazwy komputera.



Rysunek 9: Zakończenie instalacji systemu Linux Mint.

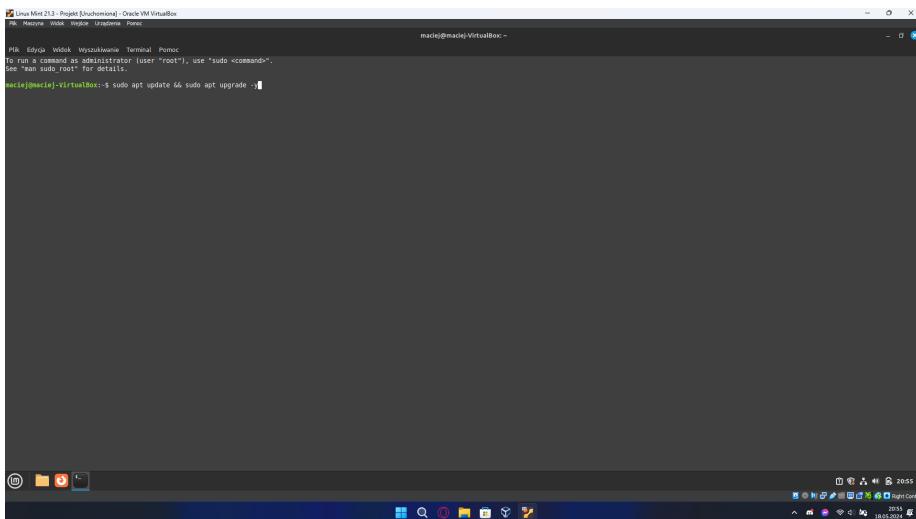
Wyświetlony zostaje monit z prośbą o usunięcie nośnika instalacyjnego. Po zakończeniu instalacji system wyświetla ekran informujący o pomyślnym zakończeniu procesu.

4.1.2 Wstępna konfiguracja systemu



Rysunek 10: Instalacja dodatków gościa dla poprawy wydajności i integracji z systemem hosta.

Po zainstalowaniu systemu operacyjnego warto zainstalować dodatki gościa, które poprawiają integrację maszyny wirtualnej z systemem hosta, co zwiększa komfort pracy.

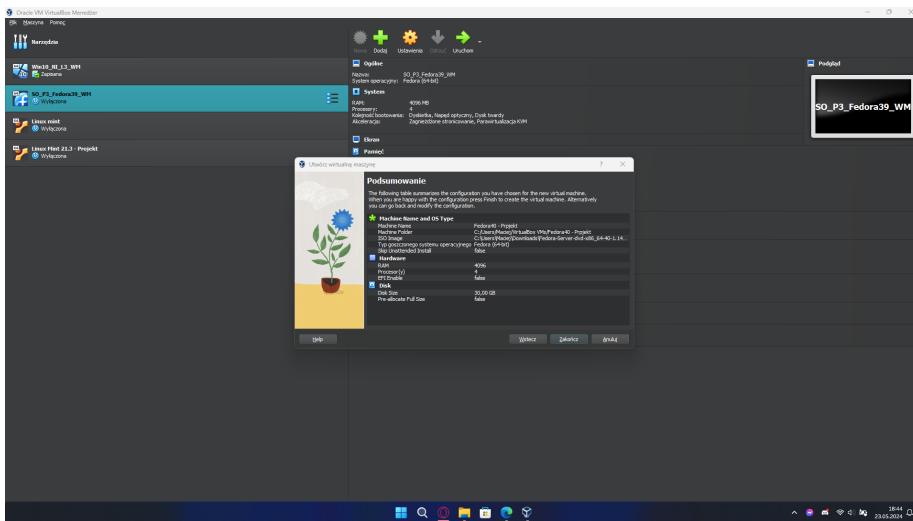


Rysunek 11: Aktualizacja pakietów systemowych.

Ostatnim krokiem wstępnej konfiguracji jest aktualizacja pakietów systemowych, aby zapewnić, że system operacyjny ma najnowsze poprawki i funkcje.

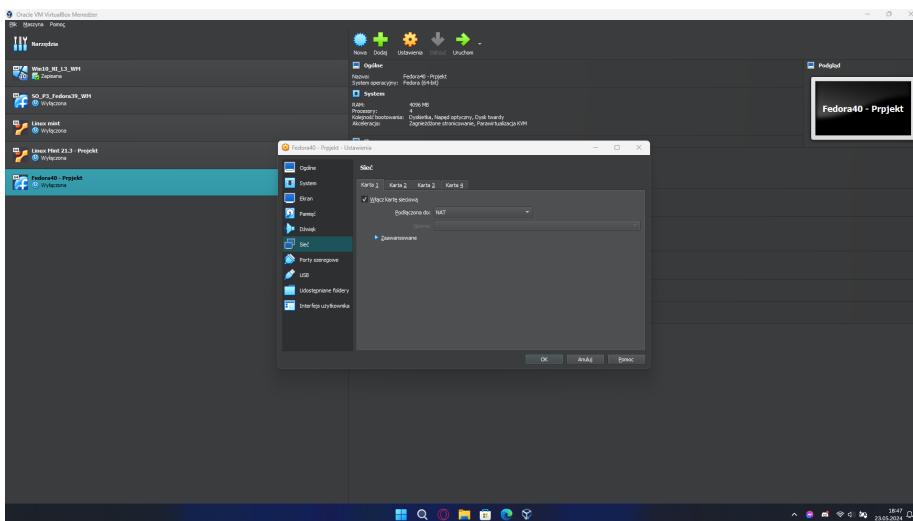
4.2 Instalacja serwera – Fedora 40

4.2.1 Proces instalacji



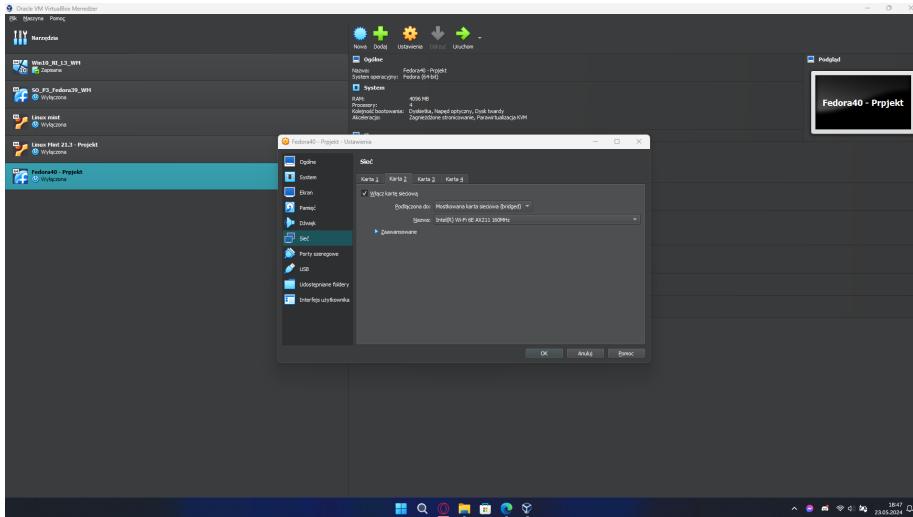
Rysunek 12: Analogicznie jak w przypadku instalacji Linux Mint – wymagane jest ustawienie nazwy maszyny wirtualnej, przydzielenie jej zasobów, ustalenie rozmiaru dysku. Powyższe zdjęcie ukazuje ekran z podsumowaniem wybranych opcji

Aby maszyna wirtualna miała dostęp do internetu wymagane jest dodanie karty sieciowej NAT, co widać na poniższym zdjęciu.



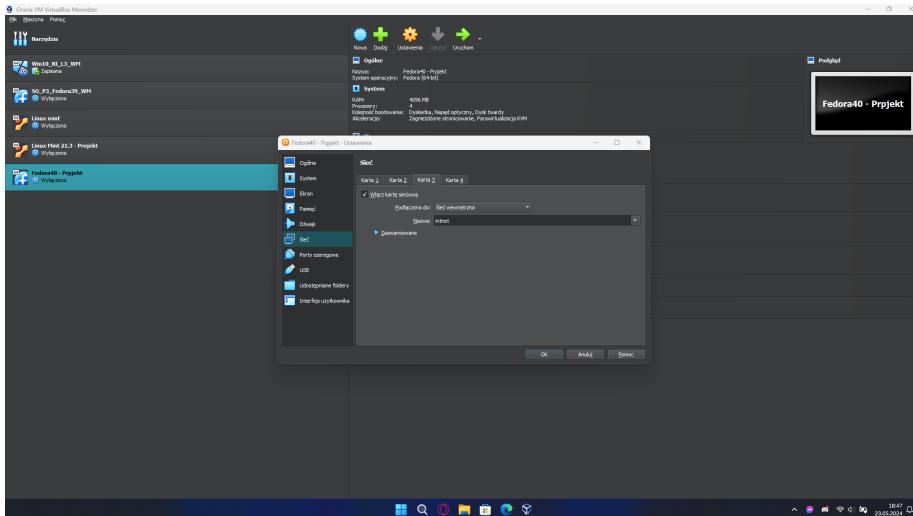
Rysunek 13: Dodanie pierszej karty sieciowej – NAT

Druga karta sieciowa jest dodana w celu połączenia się hosta z maszyną wirtualną poprzez protokół SSH oraz udostępnienia usług takich jak http czy samba. Połączenie poprzez SSH umożliwia łatwiejszą konfigurację maszyny wirtualnej.



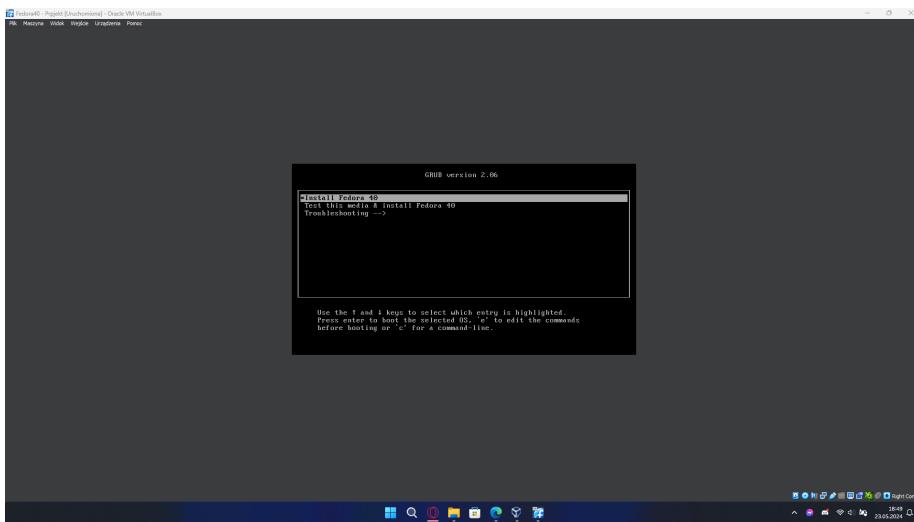
Rysunek 14: Dodanie pierszej drugiej karty sieciowej – sieć mostkowana (bridged)

Trzecia karta sieciowa posłuży do stworzenia sieci wewnętrznej dla maszyn wirtualnych w sposób taki aby się one wzajemnie widziały (tzn. były dostępne), a nie były dostępne z poziomu hosta.



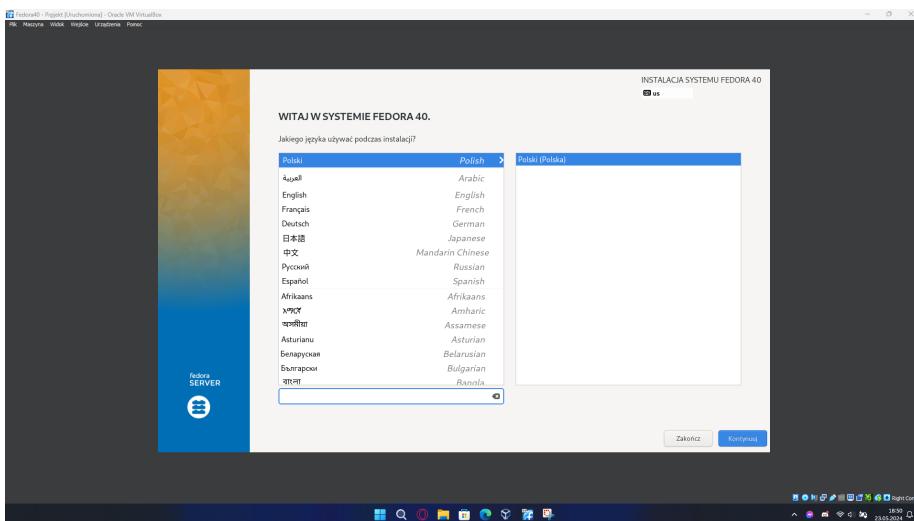
Rysunek 15: Dodanie pierszej trzeciej karty sieciowej – sieć wewnętrzna

Po dodaniu kart sieciowych można uruchomić maszynę wirtualną. Po chwili ukazuje się menu grub z opcją instalacji Fedory 40. Tą opcję należy wybrać w celu dalszej instalacji.



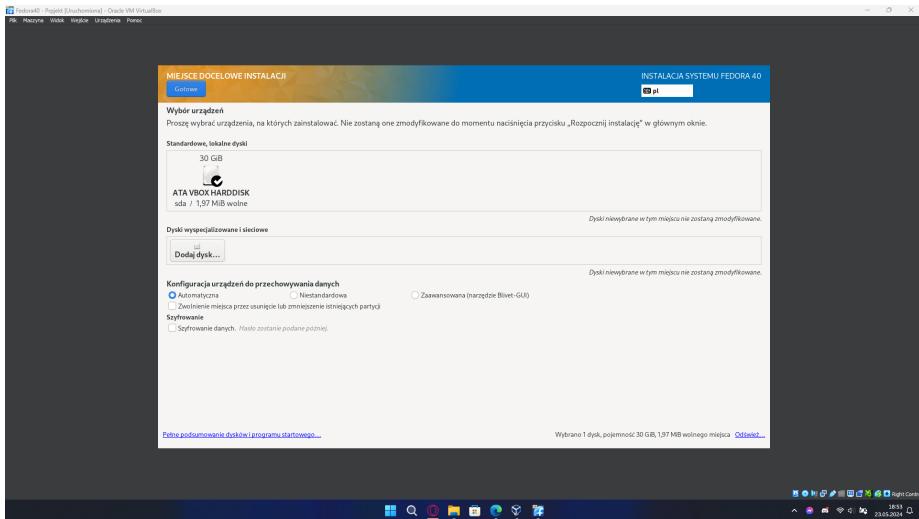
Rysunek 16: Uruchomienie instalatora Fedory.

W następnym kroku wybiera się język instalatora oraz układ klawiatury.



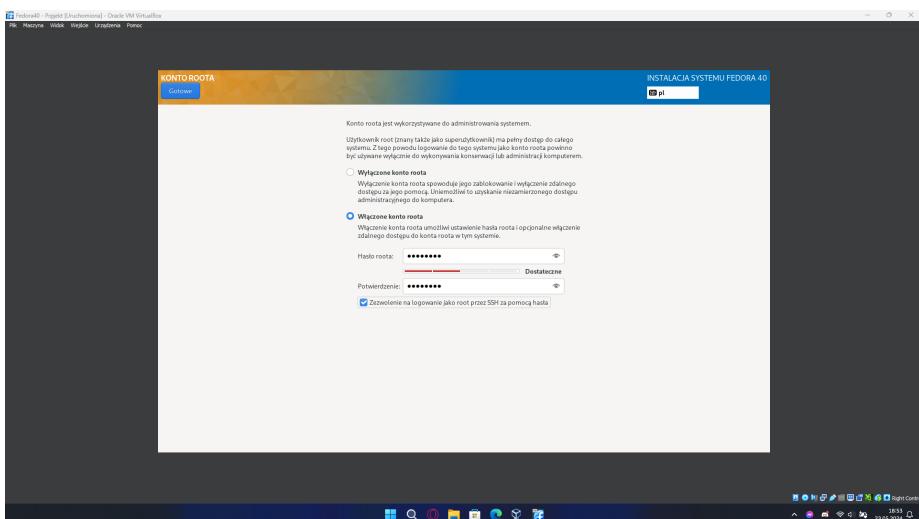
Rysunek 17: Rozpoczęcie instalacji Fedora 40 – wybór języka instalacji.

W kolejnym kroku wybieram dysk na którym ma zostać zainstalowany system. W tym miejscu można podzielić dysk na partie (podzielić na części które w systemie będą widoczne jako samodzielne dyski), sformatować go, zaszyfrować, wybrać system plików (np. ext3, ext4, zfs).



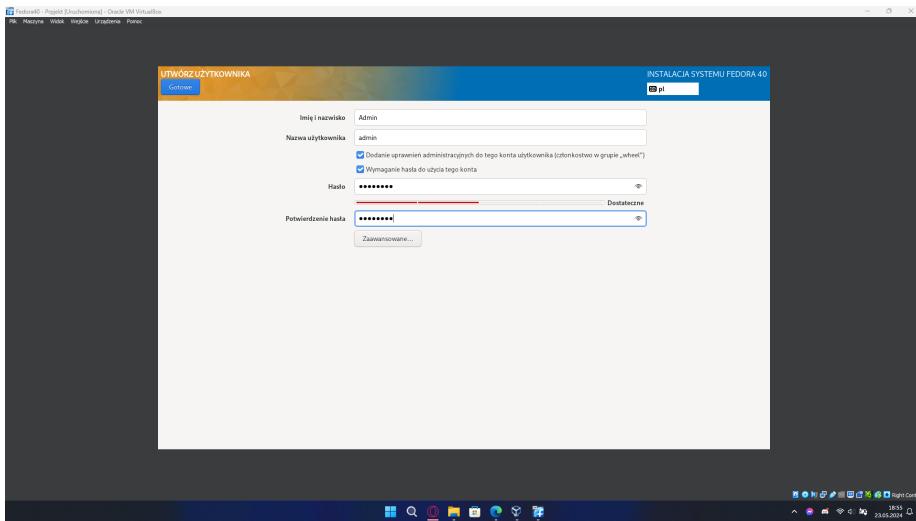
Rysunek 18: Wybór dysku na którym zostanie zainstalowany serwer

Następnie przechodzę do zakładki z ustawieniami dotyczącymi konta root. W tej zakładce ustawiam hasło do konta oraz zezwalam na połączenia SSH tym kontem. Na serwerze produkcyjnym połączenie poprzez konto root nie jest zalecanym rozwiązaniem, gdyż stanowi zagrożenie bezpieczeństwa sieci firmowej.



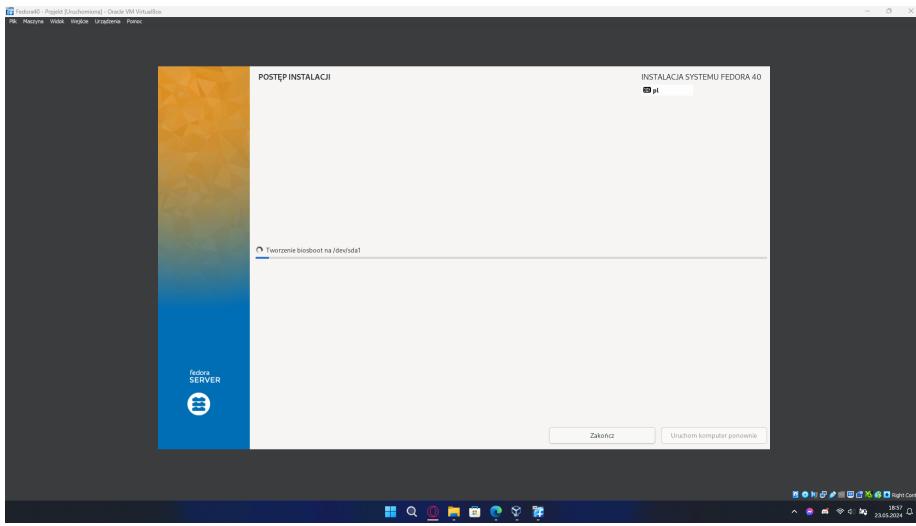
Rysunek 19: Ustawienie konta root – włączenie konta, ustawienie hasła i zezwolenie na połączenie ssh jako root

Po ustawieniu konta root'a zabieram się za stworzenie konta użytkownika. W tej części konfiguracji zaznaczam checkbox'a dotyczącego dodania konta admin do grupy wheel. Umożliwi mi to wykonywanie komendy sudo (Super User DO).



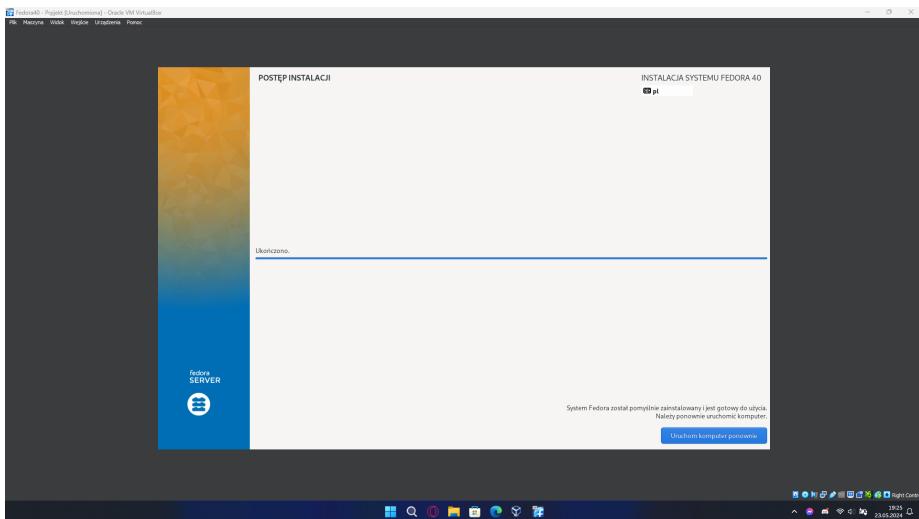
Rysunek 20: Stworzenie użytkownika – admin

Po wykonaniu powyższych kroków nie pozostaje nic innego jak rozpoczęcie instalacji.



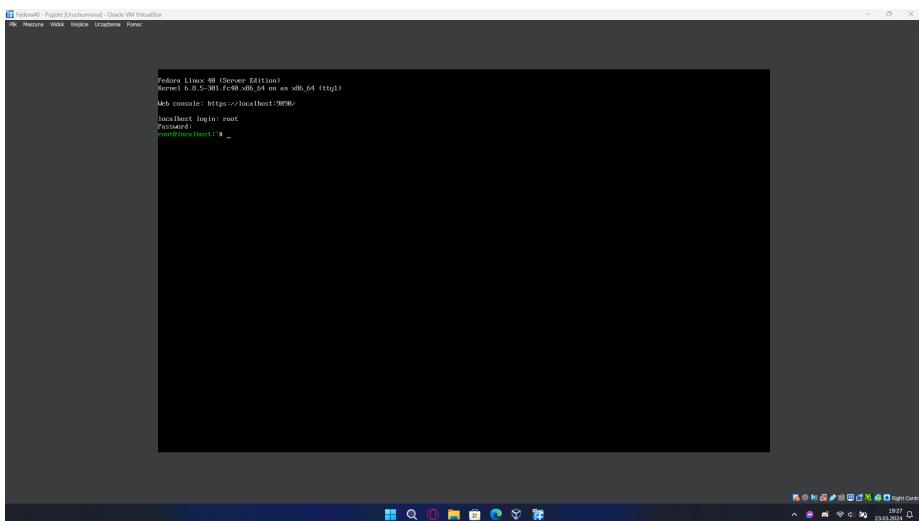
Rysunek 21: Ekran postępującej instalacji

Po jakimś czasie mogę uruchomić ponownie serwer kończąc tym samym instalację systemu.



Rysunek 22: Ekran postępującej instalacji – koniec instalacji

Po Uruchomieniu ponownym mogę zalogować się na konto root'a i zacząć konfigurację wstępna serwera.



Rysunek 23: Zainstalowany system – przed wstępnią konfiguracją

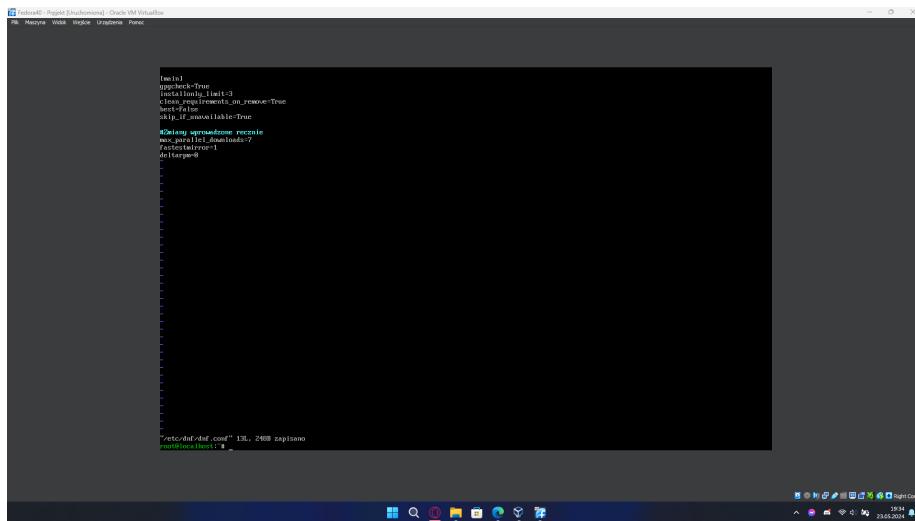
4.2.2 Wstępna konfiguracja

Po zainstalowaniu systemu, następnym krokiem powinno być zaktualizowanie pakietów aby zapewnić najnowszą funkcjonalność oraz poprawki bezpieczeństwa. Jednakże przed tym krokiem decyduję się na konfigurację menadżera pakietów dnf, aby przyśpieszyć pobieranie pakietów. Do pliku /etc/dnf/dnf.conf dodaje następujące wpisy:

```
#Zmiany wprowadzone ręcznie
max_parallel_downloads=7
fastestmirror=1
deltarpm=0
```

Wytłumaczenie opcji:

- max_parallel_downloads=7 Opcja ta pozwala menadżerowi pakietów na pobieranie do 7 pakietów na raz.
- fastestmirror=1 Opcja ta wymusza wyszukiwanie najszybszego serwera zwierciadlanego.
- deltarpm=0



```
[main]
http_ca_trust=True
installonly_limit=3
strict_requirements_on_remove=True
best=False
skip_if_unavailable=True
fastestmirror=1
max_parallel_downloads=7
deltarpm=0

cat > /etc/dnf/dnf.conf
[main]
http_ca_trust=True
installonly_limit=3
strict_requirements_on_remove=True
best=False
skip_if_unavailable=True
fastestmirror=1
max_parallel_downloads=7
deltarpm=0

cat > /etc/dnf/dnf.conf
```

Rysunek 24: Dodanie wpisów do /etc/dnf/dnf.conf aby przyśpieszyć działanie menadżera pakietów dnf

Teraz po skonfigurowaniu menadżera pakietów można wykonać aktualizację pakietów.

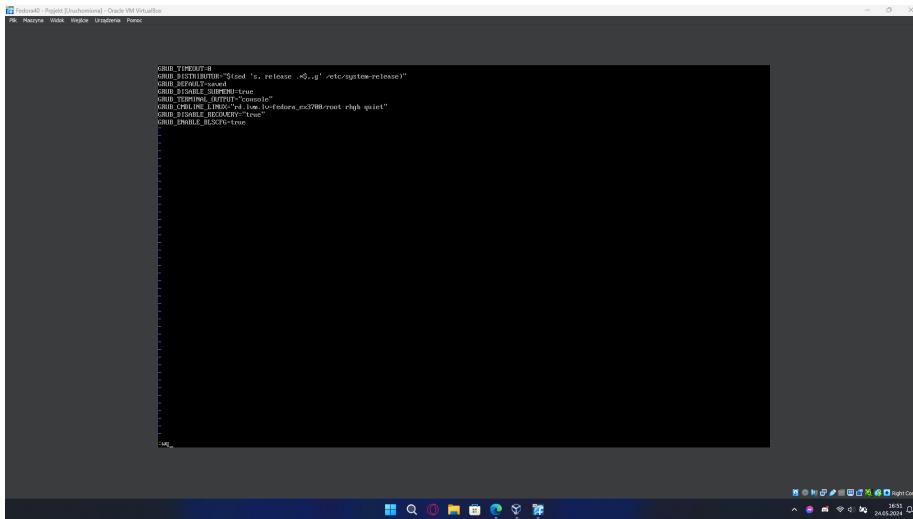
Rysunek 25: Aktualizacja pakietów systemowych – test konfiguracji dnf

Po aktualizacji pakietów postanowiłem edytować irytującą mnie rzeczą tj. uruchamianie się grubą przy jednym systemie operacyjnym. Na poniższym zdjęciu jest plik /etc/default/grub oryginalny (przed modyfikacją)

```
Fedora40 - Projekt [Unuchoming] - Oracle VM VirtualBox  
File Mausika Webk Inspector Uradbenik Ponac  
  
GRUB_TIMEOUT=5  
GRUB_BIOSINITOR="Sbcd 's, release .w2,g' /etc/system-release"  
GRUB_DISABLE_LINUXRC_WARNING=true  
GRUB_TERMINAL_OUTPUT="text"  
GRUB_GFXMODE="auto"  
GRUB_GFXPAYLOAD_LINUX="text b-fedora.xz:2560xroot rfbq quiet"  
GRUB_STABILE_RECOVERY="true"  
GRUB_ENABLE_HIDDEN=true  
  
[...]  
  
"/etc/default/grub" 8L, 271B  
1.14 10:57:00 24.05.2024
```

Rysunek 26: plik /etc/default/grub przed zmianą

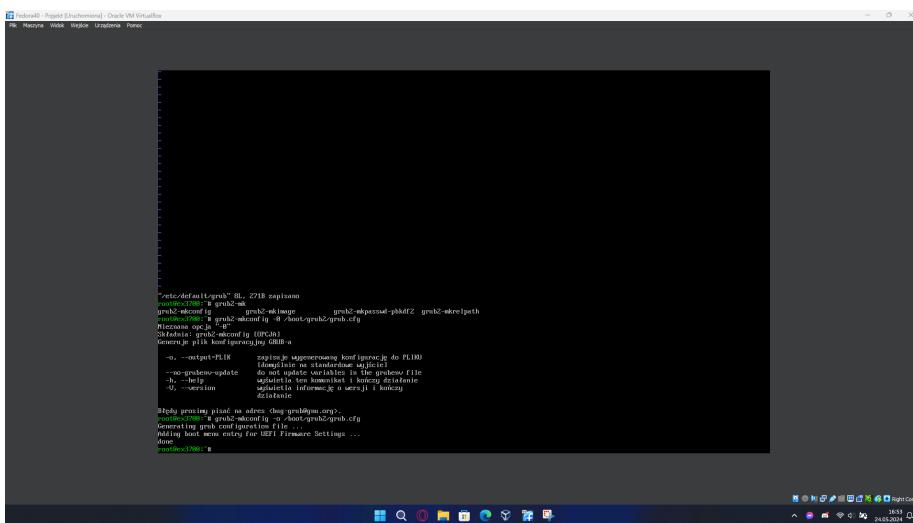
W kolejnym kroku zmieniłem GRUB_TIMEOUT=5 na GRUB_TIMEOUT=0
Co można zauważyc poniższym zdjeciu.



Rysunek 27: plik /etc/default/grub po zmianie.

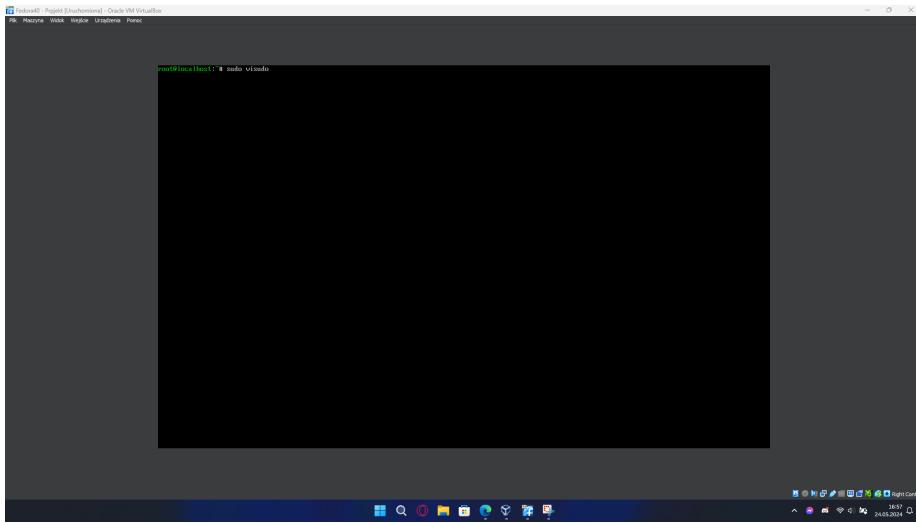
Aby zatwierdzić zmiany należy użyć komendy:

```
grub2-mkconfig -o /boot/grub2/grub2.cfg
```



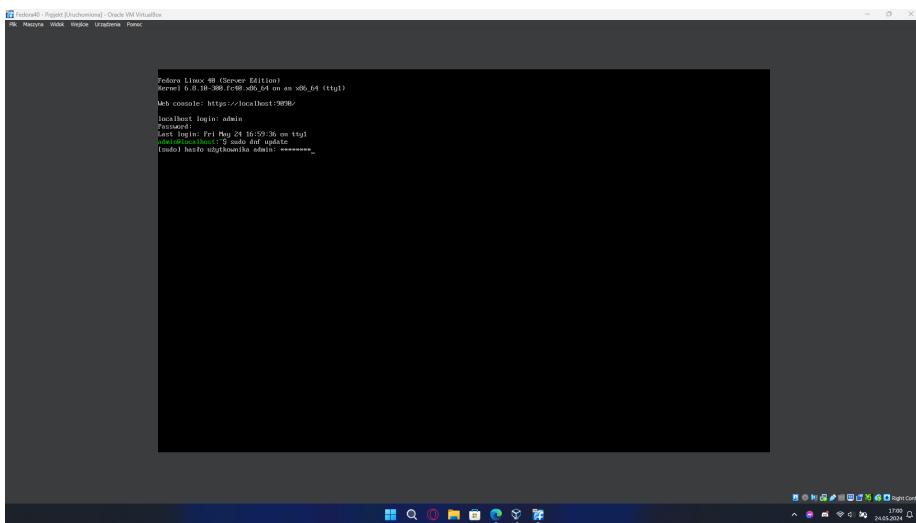
Rysunek 28: Zastosowanie zmian po edycji grub

W kolejnym kroku postanowiłem ułatwić wpisywanie hasła, gdy korzystam z sudo.



Rysunek 29: Zwiększenie wygody wpisywania haseł – edycja pliku komendą sudo visudo

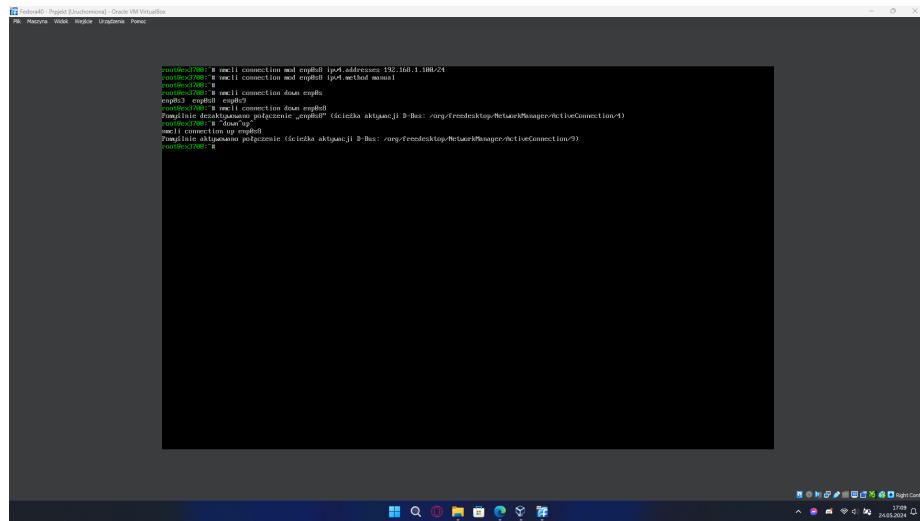
Efekt powyższego kroku:



Rysunek 30: Zwiększenie wygody wpisywania haseł – efekt działania po zmianach

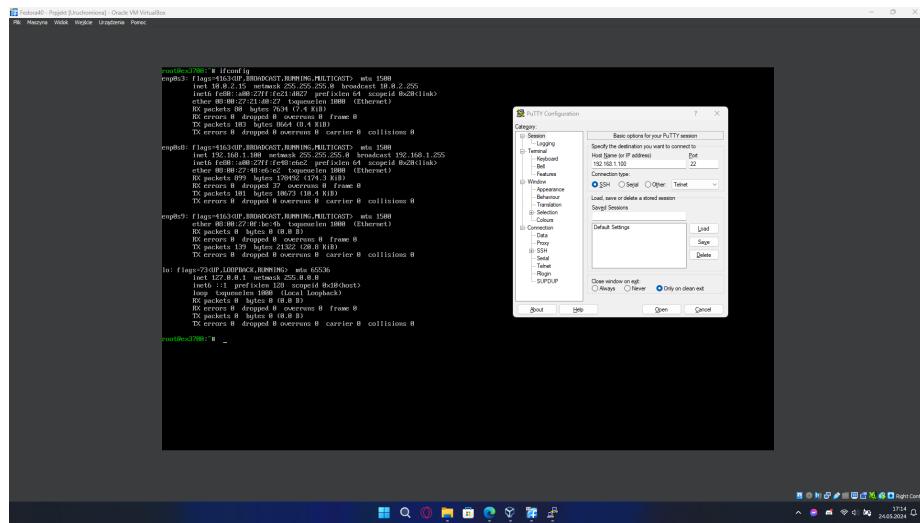
4.3 Konfiguracja SSH

Aby umożliwić połączenie z SSH na serwerze (VirtualBox) w pierszej kolejności potrzeba jest ustawić poprawnego adresu IP z sieci lokalnej dla karty ustawionej na sieć mostkowaną (w moim przypadku jest to enp0s8)



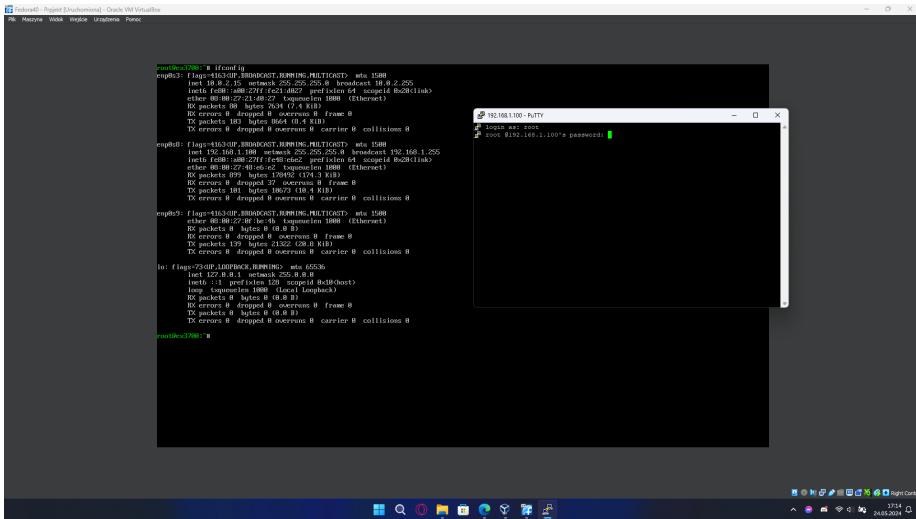
Rysunek 31: konfiguracja karty sieciowej

W serwerze Fedora 40 SSH jest domyślnie włączone i skonfigurowane. Wystarczy tylko się połączyć



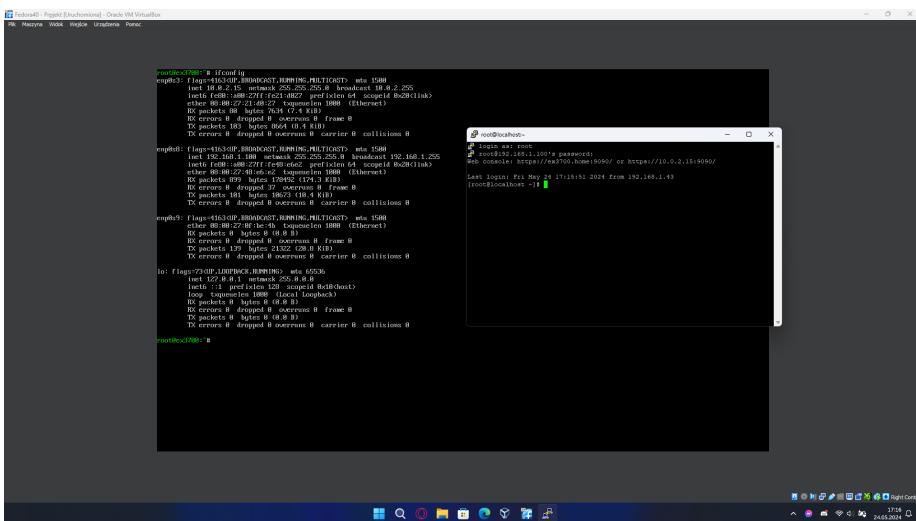
Rysunek 32: Konfiguracja aplikacji PuTTY

Próba zalogowania na konto root'a:



Rysunek 33: Podlaczenie poprzez PuTTY na konto root'a

Wynik powyższego kroku:

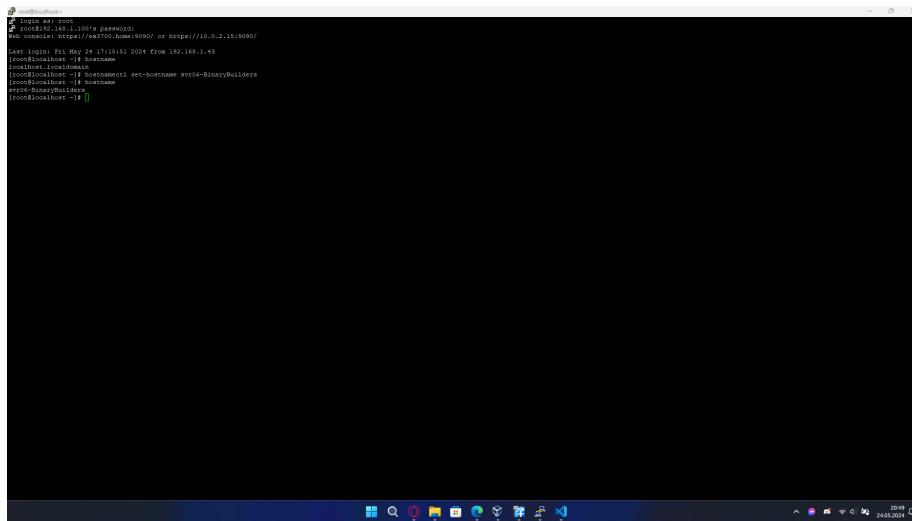


Rysunek 34: Wynik połączenia poprzez PuTTY

4.4 Nazwa serwera – hostname

Aby zmienić nazwę serwera (hostname) można użyć komendy:

```
hostnamectl set-hostname nazwa-komputera
```



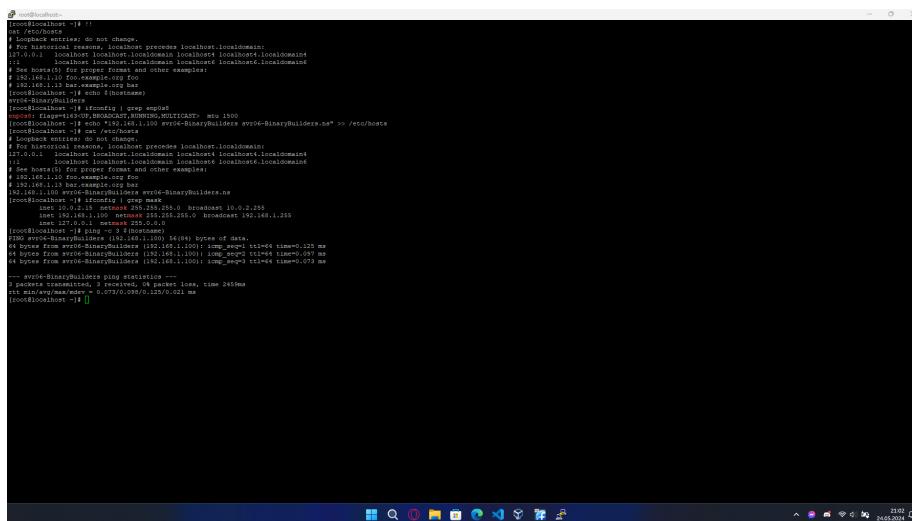
```
[root@svr04 ~]# hostnamectl set-hostname svr04-BinaryBuilders
[root@svr04 ~]#
```

Rysunek 35: Zmiana nazwy serwera

4.5 DNS – instalacja i konfiguracja

Pierwszym krokiem w konfiguracji DNS jest dodanie odpowiedniego wpisu do /etc/hosts. W moim przypadku jest to:

```
192.168.230.1 svr06-BinaryBuilders svr06-BinaryBuilders.ns
```



```
[root@svr04 ~]# cat >/etc/hosts <> 192.168.230.1 svr06-BinaryBuilders svr06-BinaryBuilders.ns
[root@svr04 ~]# ping -c 1 svr06-BinaryBuilders.ns
PING svr06-BinaryBuilders.ns (192.168.230.1) 56(84) bytes of data:
64 bytes from svr06-BinaryBuilders.ns(192.168.230.1): icmp_seq=1 ttl=64 time=0.125 ms
64 bytes from svr06-BinaryBuilders.ns(192.168.230.1): icmp_seq=2 ttl=64 time=0.097 ms
64 bytes from svr06-BinaryBuilders.ns(192.168.230.1): icmp_seq=3 ttl=64 time=0.073 ms
3 packets transmitted, 3 received, 0% packet loss, time 246ms
rtt min/avg/max/mdev = 0.073/0.096/0.125/0.021 ms
[root@svr04 ~]#
```

Rysunek 36: Edycja /etc/hosts

Aby zainstalować oprogramowanie do stworzenia serwera DNS należy wydać polecenie:

- Jeśli jesteś na koncie root:

```
dnf install bind bind-utils -y
```

- jeżeli jesteś na innym koncie ale jesteś w grupie sudoers:

```
sudo dnf install bind bind-utils -y
```

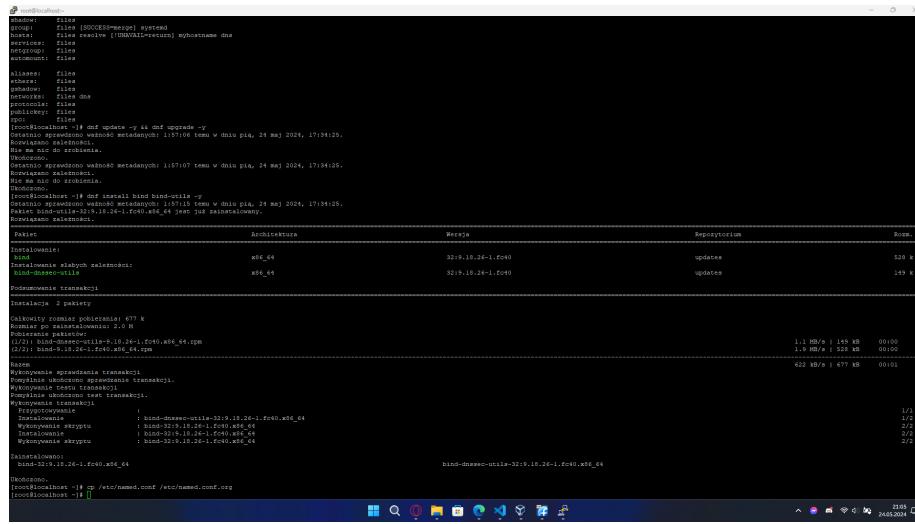
Po zainstalowaniu wymaganych pakietów należy wykonać kopię zapasową plików konfiguracyjnych. Można to zrobić komendą:

- Jeśli jesteś na koncie root:

```
cp /etc/named.conf /etc/named.conf.org
```

- jeżeli jesteś na innym koncie ale jesteś w grupie sudoers:

```
sudo cp /etc/named.conf /etc/named.conf.org
```



Rysunek 37: Instalacja DNS

Następnie trzeba skonfigurować plik /etc/named.conf. Można zrobić to komendą:

```
sudo nano /etc/named.conf
```

```

root@cockpit-192-168-230-1: ~ % cp /etc/named.conf /etc/named.conf.org
[1] 11865 cp: /etc/named.conf.org: Success

```

Rysunek 38: Stworzenie kopii zapasowej pilku konfiguracyjnego DNS

```

options {
    listen-on port 53 { 127.0.0.1; 192.168.230.1; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    secroots-file "/var/named/data/named.secroots";
    recursing-file "/var/named/data/named.recursing";
    allow-query { 127.0.0.1; 192.168.230.0/24;};

    /*
     - If you are building an AUTHORITATIVE DNS server, do NOT enable
     ↳ recursion.
     - If you are building a RECURSIVE (caching) DNS server, you need to
     ↳ enable
     recursion.
     - If your recursive DNS server has a public IP address, you MUST
     ↳ enable access
     control to limit queries to your legitimate users. Failing to do so
     ↳ will
     cause your server to become part of large scale DNS amplification
     attacks. Implementing BCP38 within your network would greatly
     reduce such attack surface
    */
    recursion yes;
    /*dnssec-enable yes;*/
    dnssec-validation yes;
    managed-keys-directory "/var/named/dynamic";
    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
}

```

```

/* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
include "/etc/crypto-policies/back-ends/bind.config";
};

logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };

    channel queries_log {
        file "/var/named/queries.log" versions 600 size 20m;
        print-time yes;
        print-category yes;
        print-severity yes;
        severity info;
    };
    category queries { queries_log; };
};

view "internal" {
    match-clients {
        localhost;
        192.168.230.0/24;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};

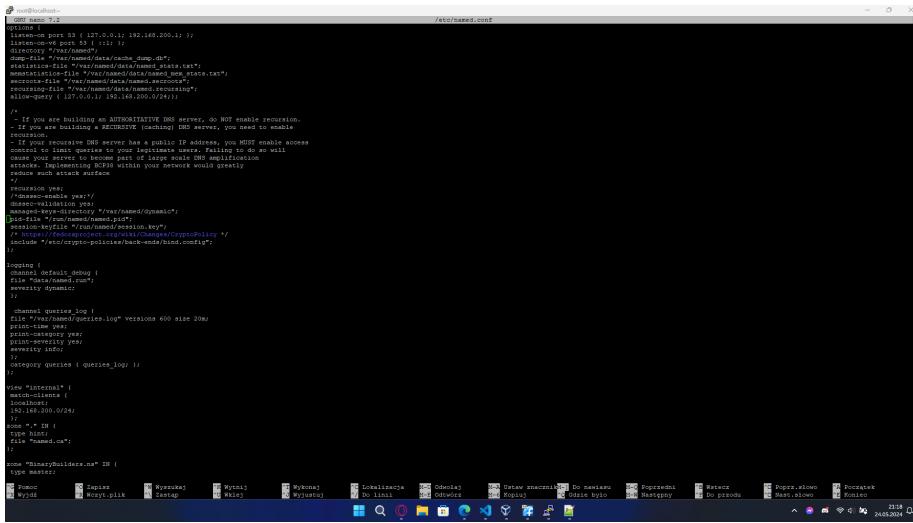
zone "BinaryBuilders.ns" IN {
    type master;
    file "BinaryBuilders.ns.lan_in";
    allow-update { none; };
};

zone "230.168.192.in-addr.arpa" IN {
    type master;
    file "230.168.192.lan_in";
    allow-update { none; };
    include "/etc/named.rfc1912.zones";
    include "/etc/named.root.key";
};

```

Powyżej znajduje się zawartość pliku /etc/named.conf, którą należy wprowadzić.

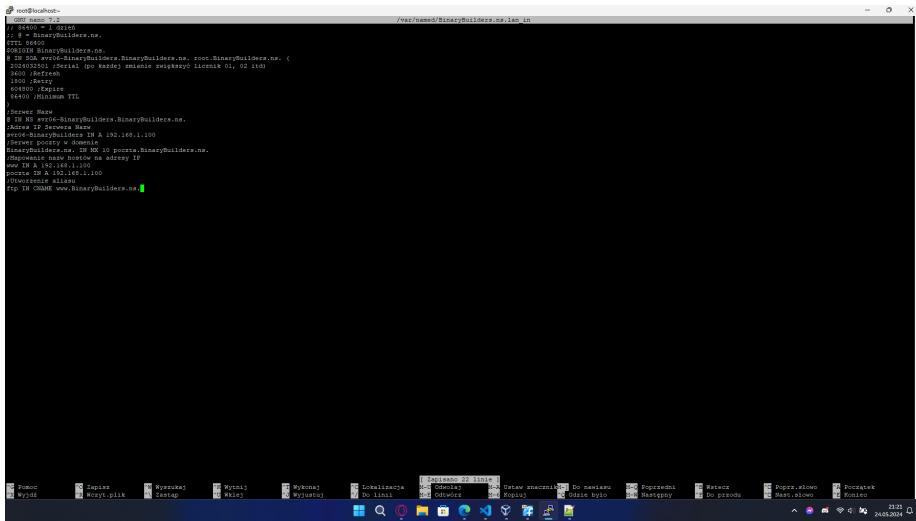
W kolejnym kroku trzeba utworzyć plik strefy podstawowej. W moim przypadku jest to plik /var/BinaryBuilders.ns.lan.in. Zawartość tego pliku:



Rysunek 39: zawartość named.conf

```
;; 86400 = 1 dzień
;; @ = BinaryBuilders.ns.
$TTL 86400
$ORIGIN BinaryBuilders.ns.
@ IN SOA svr06-BinaryBuilders.BinaryBuilders.ns.
    → root.BinaryBuilders.ns. (
        2024032502 ;Serial (po każdej zmianie zwiększyć licznik 01,
        → 02 itd)
        3600 ;Refresh
        1800 ;Retry
        604800 ;Expire
        86400 ;Minimum TTL
    )
;Serwer Nazw
@ IN NS svr06-BinaryBuilders.BinaryBuilders.ns.
;Adres IP Serwera Nazw
svr06-BinaryBuilders IN A 192.168.230.1
;Serwer poczty w domenie
BinaryBuilders.ns. IN MX 10 poczta.BinaryBuilders.ns.
;Mapowanie nazw hostów na adresy IP
www IN A 192.168.230.1
poczta IN A 192.168.230.1
sfs IN A 192.168.230.1
;Utworzenie aliasu
ftp IN CNAME www.BinaryBuilders.ns.
```

W kolejnym kroku trzeba utworzyć plik strefy dla przeszukiwania wstecznego. W moim przypadku jest to plik /var/230.168.192.lan.in. Zawartość tego pliku:



Rysunek 40: zawartość pliku strefy podstawowej

```

$TTL 86400
@ IN SOA svr06-BinaryBuilder.BinaryBuilders.ns.
    root.BinaryBuilders.ns. (
        2023032902 ;Serial (po każdej zmianie zwiększyć licznik
        01,02 itd.)
        3600 ;Refresh
        1800 ;Retry
        604800 ;Expire
        86400 ;Minimum TTL
)
;Serwer Nazw
@ IN NS svr06-BinaryBuilder.BinaryBuilders.ns.
svr06-BinaryBuilder.BinaryBuilders.ns. IN A 192.168.230.1
;Odwrotny wpis dla Serwera Nazw
1 IN PTR svr06-BinaryBuilder.BinaryBuilders.ns.
;PTR adresów IP danych hostów
1 IN PTR poczta.BinaryBuilders.ns.
1 IN PTR www.BinaryBuilders.ns.

```

Następnym krokiem jest uruchomienie kilku komend:

```

systemctl start named
systemctl enable named
firewall-cmd --add-service=dns --permanent
firewall-cmd --reload
nmcli con mod enp0s9 ipv4.dns 192.168.230.1
nmcli con down enp0s9 && nmcli con up enp0s9
rndc reload
rndc status

```

```

root@localhost:~#
root@localhost:~# cat /var/named/1.192.168.lan.in
;TTL 86400
$TTL 86400
;@ IN NS svr01-BinaryBuilder.BINARYBUILDERS.NS. root.BINARYBUILDERS.NS. (
;202.30.230.191. ;Serial (po każdej zmianie zwiększyć licznik o 1,02 itd.)
;1000 ;Refresh
;100 ;Retry
;604800 ;Expire
;3600 ;TTL
;
;Server Name
;IN NS svr01-BinaryBuilder.BINARYBUILDERS.NS.
;IN A 192.168.1.100
;Odbierac wisi dla Servera Name
;adresów IP dla danego hosta
;PTR adresów IP dla danego hosta
;IN PTR poista.BINARYBUILDERS.NS.
;IN PTR www.BINARYBUILDERS.NS.
;

```

Rysunek 41: zawartość pliku strefy dla przeszukiwania wstecznego

Wytlumaczenie powyższych poleceń:

- systemctl start named

Komenda ta uruchomi usługę

- systemctl enable named

Polecenie to spowoduje że usługa będzie uruchamiana automatycznie przy włączeniu serwera

- firewall-cmd –add-service=dns –permanent

Dodaje regułę zapory sieciowej, aby na stałe zezwalać na ruch DNS.

- firewall-cmd –reload

Przeładowuje ustawienia zapory sieciowej, aby zastosować wprowadzone zmiany.

- nmcli con mod enp0s9 ipv4.dns 192.168.230.1

Modyfikuje połaczenie enp0s9, aby używało serwera DNS o adresie 192.168.230.1.

- nmcli con down enp0s9 && nmcli con up enp0s9

Dezaktywuje i ponownie aktywuje połaczenie sieciowe enp0s9.

- rndc reload

Przeładowuje konfigurację serwera serwera DNS.

- rndc status

wyświetla status serwera serwera DNS.

Jak widać na zrzucie ekranu powyżej miałem problemy z błędna konfiguracją jednego z pliku, jednakże udało mi się naprawić problem i uruchomić usługę DNS. Kolejnym i ostatnim krokiem jest test usługi DNS. Wykonać go można korzystając z drugiej maszyny wirtualnej. Przykładowy test DNS możesz zobaczyć tutaj.

Rysunek 42: Uruchomienie usługi DNS

4.6 DHCP – instalacja i konfiguracja

Aby zainstalować oprogramowanie do stworzenia serwera DHCP należy wydać polecenie:

```
sudo dnf install -y dhcp-server
```

Rysunek 43: Instalacja DHCP

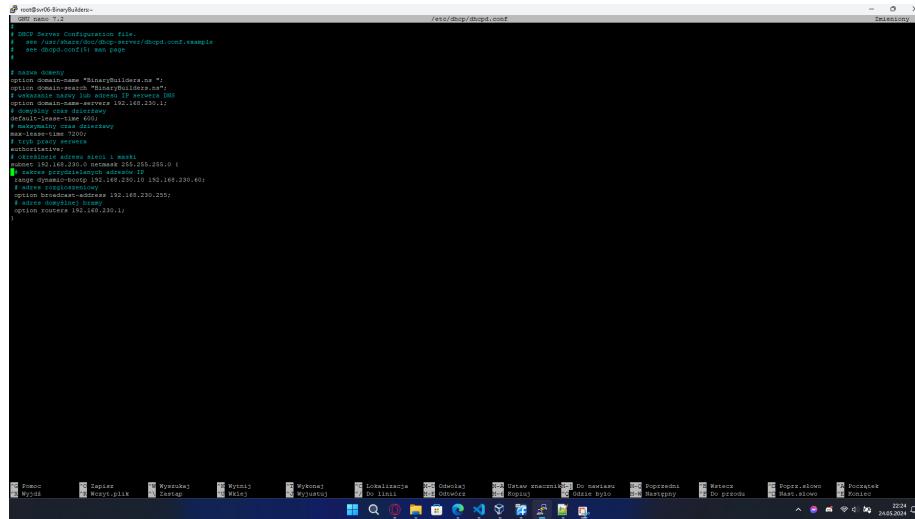
Po zainstalowaniu odpowiednich pakietów dobrze jest wykonać kopię zapasową oryginalnego pliku konfiguracji, co widać na zrucie powyżej. Można to zrobić następującą komendą:

```
cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.org
```

W kolejnym kroku należy wprowadzić zmiany w pliku konfiguracyjnym DHCP tj. /etc/dhcp/dhcpd.conf. W moim przypadku:

```
# nazwa domeny
option domain-name "BinaryBuilders.ns ";
option domain-search "BinaryBuilders.ns";
# wskazanie nazwy lub adresu IP serwera DNS
option domain-name-servers 192.168.230.1;
# domyślny czas dzierżawy
default-lease-time 600;
# maksymalny czas dzierżawy
max-lease-time 7200;
# tryb pracy serwera
authoritative;
# określneie adresu sieci i maski
subnet 192.168.230.0 netmask 255.255.255.0 {
    # zakres przydzielanych adresów IP
    range dynamic-bootp 192.168.230.10 192.168.230.60;
    # adres rozgłoszeniowy
    option broadcast-address 192.168.230.255;
    # adres domyślnej bramy
    option routers 192.168.230.1;
}
```

Wprowadzoną treść widać na zrzucie poniżej.



Rysunek 44: Konfiguracja DHCP – edycja pliku /etc/dhcp/dhcpd.conf

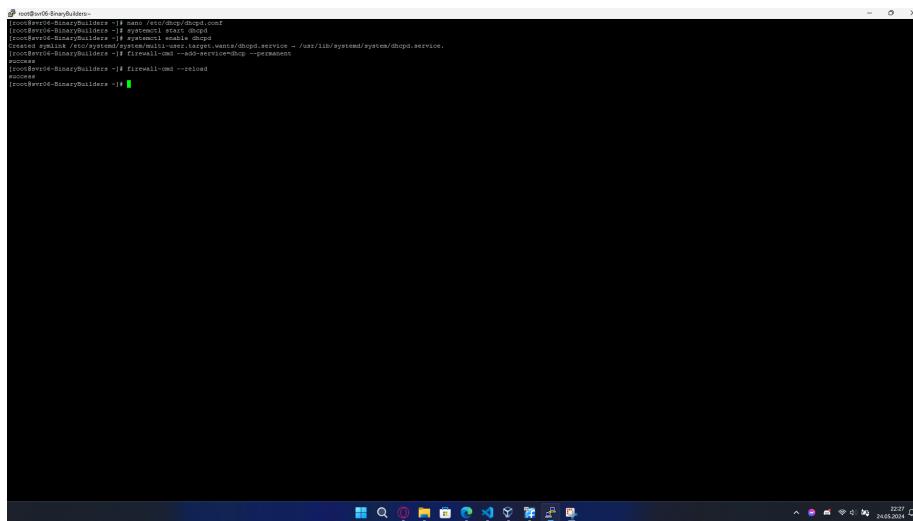
Następnym krokiem jest wprowadzenie kilku poleceń:

```
systemctl start dhcpcd  
systemctl enable dhcpcd  
firewall-cmd --add-service=dhcp --permanent  
firewall-cmd --reload
```

Wytłumaczenie powyższych poleceń:

- systemctl start dhcpcd
Komenda ta uruchomi usługę DHCP
- systemctl enable dhcpcd
Polecenie to spowoduje że usługa DHCP będzie uruchamiana automatycznie przy włączeniu serwera
- irewall-cmd --add-service=dhcp --permanent
Dodaje regułę zapory sieciowej, aby na stałe zezwalać na ruch DHCP.
- firewall-cmd --reload
Przeładowuje ustawienia zapory sieciowej, aby zastosować wprowadzone zmiany.

Zrzut ekranu poniżej przedstawia wykonanie tych komend.



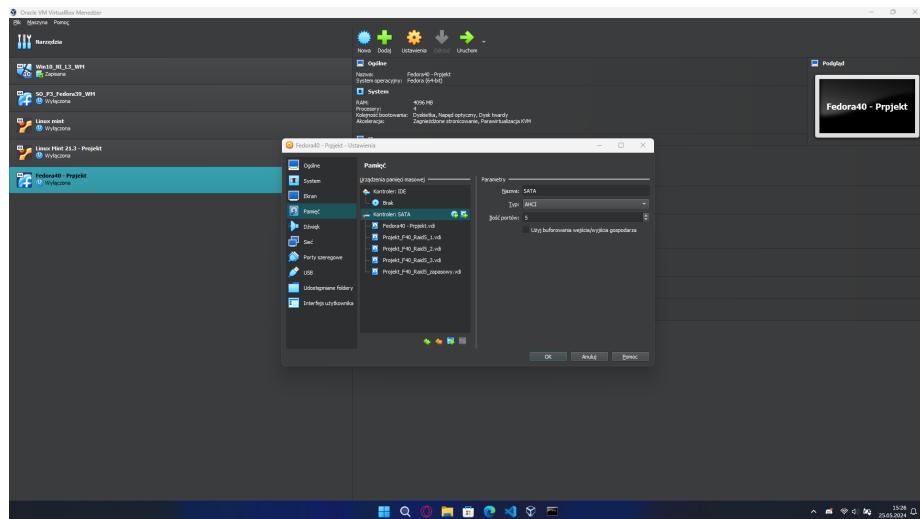
```
[root@srv06-BinaryBuilders ~]# nano /etc/dhcp/dhcpd.conf
[root@srv06-BinaryBuilders ~]# systemctl start dhcpcd
[root@srv06-BinaryBuilders ~]# systemctl enable dhcpcd
[root@srv06-BinaryBuilders ~]# useradd -r -u 1000 -g 1000 -s /usr/bin/false -c 'dhcp' -m -d /var/lib/centos/dhcp - /usr/lib/systemd/system/dhcpcd.service
[root@srv06-BinaryBuilders ~]# firewall-cmd --add-service=dhcp --permanent
[root@srv06-BinaryBuilders ~]# firewall-cmd --reload
[root@srv06-BinaryBuilders ~]#
```

Rysunek 45: Instalacja DHCP

Po powyższym kroku nie pozostaje nic innego jak przetestować działanie DHCP. Wyniki testu dostępne są [tutaj](#).

4.7 RAID 5 – konfiguracja

Aby skonfigurować RAID 5 z 3 dysków głównych i jednym dyskiem zapasowym o wypadkowej pojemności 10GB, trzeba dodać 4 dyski o pojemności 5GB.



Rysunek 46: Dodanie dysków w VirtualBox

Po dodaniu dysków w VirtualBox należy uruchomić serwer. Po uruchomieniu serwera sprawdzam czy dyski są widoczne przez system operacyjny. Można to sprawdzić wykonując komendę:

```
lsblk
```

Następnie tworzę macierz następującymi komendami:

```
mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3  
  /dev/sd[b-d] --spare-devices=1 /dev/sde  
mdadm -D /dev/md0
```

Wytłumaczenie komend powyżej:

- mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sd[b-d]
--spare-devices=1 /dev/sde

Komenda ta tworzy macierz RAID 5 o nazwie /dev/md0 z trzech urządzeń (tutaj /dev/sdb, /dev/sdc, i /dev/sdd) i jednym urządzeniem zapasowym (/dev/sde). Parametr --verbose sprawia, że proces tworzenia macierzy będzie wyświetlał szczegółowe informacje na temat wykonywanych operacji.

- mdadm -D /dev/md0

Polecenie to wyświetla szczegółowe informacje o istniejącej macierzy RAID /dev/md0. Pokazuje takie dane jak status macierzy, urządzenia składowe, poziom RAID i wiele innych.

Na następnej stronie znajduje się zrzut ekranu prezentujący działanie tych poleceń.

```

root@avrd0-BinaryBuilders: ~ % mdadm --create /dev/md0 --level=5 --raid-devices=2 /dev/sda /dev/sdb
mdadm: layout defaults to left-symmetric
mdadm: array /dev/md0 started.
mdadm: chunk size defaults to 64K
mdadm: array set to 532368K (5.16 GB)
mdadm: array /dev/md0 started
mdadm: cannot open /dev/sda[0]: No such file or directory
mdadm: cannot open /dev/sdb[1]: No such file or directory
mdadm: array /dev/md0 started
mdadm: BinaryBuilders -> /dev/md0
[mdadm]
[mdadm]
Creating raid 5 at May 25 18:54:23 2024
  Array Size : 10.73520 (9.99 GiB 10.73 GB)
  Used Dev Size : 5.37740 (4.99 GiB 5.16 GB)
  Raid Level   : raid5
  Total Devices : 2
    Persistence : superblock is persistent
  Update Time : Sat May 25 18:53:07 2024
    State : clean
    Active Devices: 2
    Failed Devices: 0
    SpARE Devices: 0
      Layout : left-symmetric
      Chunk Size : 512K

Consistency Policy : syncbybyte

Name : avrd0-BinaryBuilders:0 (local to host avrd0-BinaryBuilders)
UUID : 00974200-0097-3a1e-9a01-0f0d5bc0
Events : 1

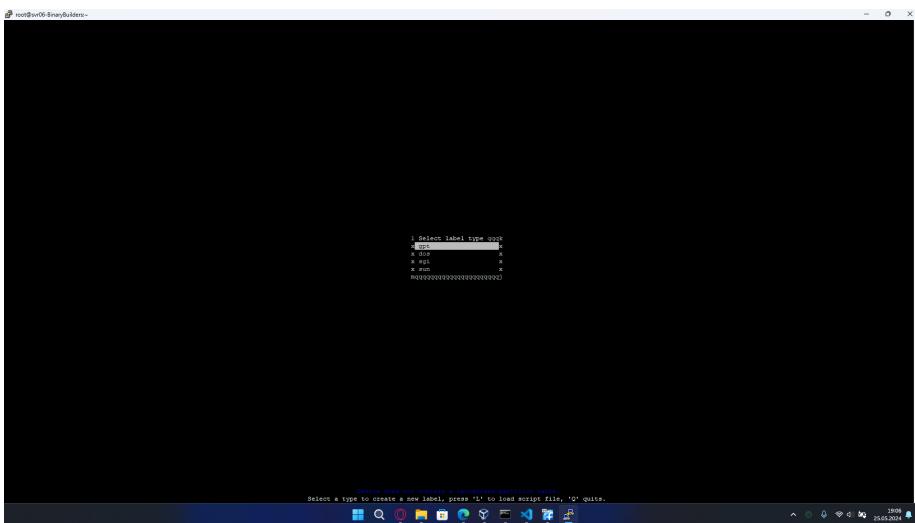
Number Major Minor RaidDevice State
0      8      16        0     active sync  /dev/sda
1      8      48        1     active sync  /dev/sdb
2      8      64        -     spare      /dev/sde

[mdadm@avrd0-BinaryBuilders ~]%

```

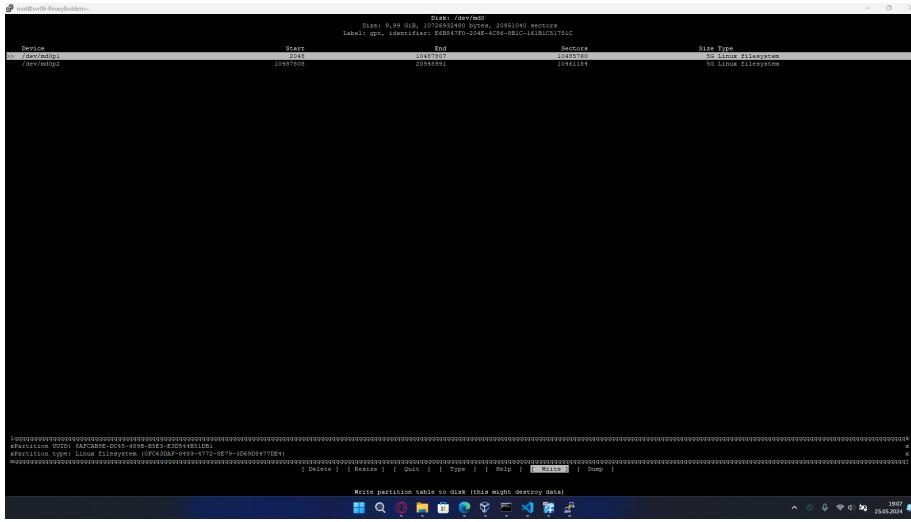
Rysunek 47: Stworzenie macierzy raid 5

Następnym krokiem jest wybranie schematu partycjonowania. Ja zostawiłem domyślny wybór – GPT



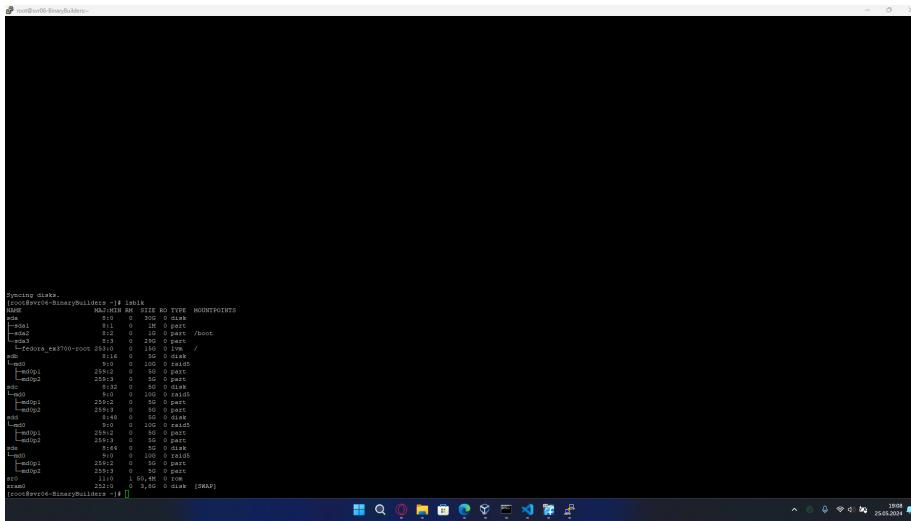
Rysunek 48: Partycjonowanie macierzy narzędziem cfdisk

Kolejnym krokiem jest utworzenie dwóch partycji na macierzy, którą wcześniej stworzyłem.



Rysunek 49: Stworzenie dwóch partycji – każda 5GB

Potwierdzenie działania poprzedniej komendy:



Rysunek 50: Wynik partycjonowania

Formatowanie przed chwilą stworzonych partycji (w systemie plików ext4), stworzenie katalogów /dysksieciowy, /kopie, zamontowanie partycji do tych katalogów oraz wyświetlenie id dysków i partycji w systemie.

```

[root@Fedor 0]# ./BinaryBuilder -J mfs.ext4 /dev/mq0
[...]
[root@Fedor 0]# cd /mnt/mq0
[root@Fedor mq0]# mkdir -p /dysksieciowy /kopie
[root@Fedor mq0]# mount /dev/mq0 /dysksieciowy
[root@Fedor mq0]# mount /dev/mq0 /kopie
[root@Fedor mq0]# ls -l /dysksieciowy /kopie
total 0
total 0
[root@Fedor mq0]# ./BinaryBuilder -J mfs.ext4 /dev/mq1
[...]
[root@Fedor mq1]# cd /mnt/mq1
[root@Fedor mq1]# mkdir -p /dysksieciowy /kopie
[root@Fedor mq1]# mount /dev/mq1 /dysksieciowy
[root@Fedor mq1]# mount /dev/mq1 /kopie
[root@Fedor mq1]# ls -l /dysksieciowy /kopie
total 0
total 0

```

Rysunek 51: Przygotowanie ścieżek do montowania

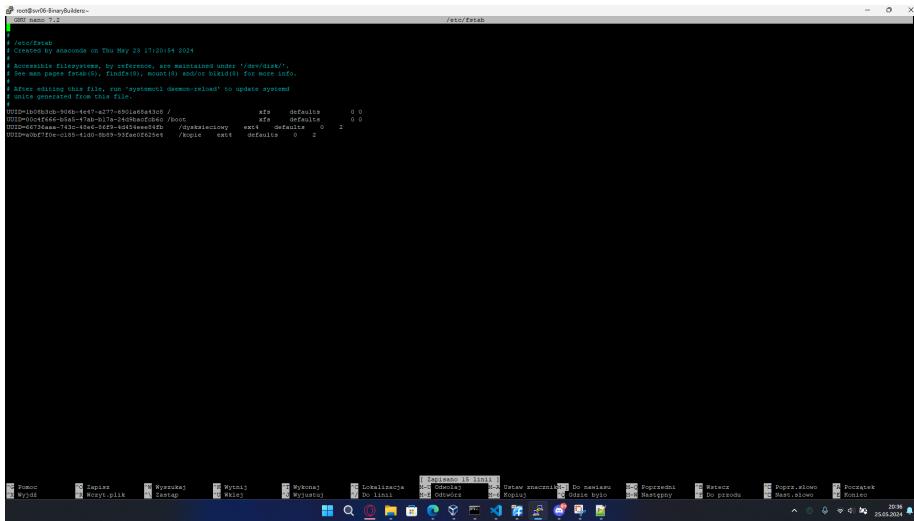
Następnym krokiem jest zapewnienie automatycznego montowania utworzonych partycji. Aby to osiągnąć należy zmodyfikować /etc/fstab, ale najpierw warto wykonać kopię, gdyż jest to kluczowy składnik systemu. W razie awarii tego pliku nawet cały system może się nie uruchomić. Postanowilem użyć id ponieważ jest niezmienne w przeciwieństwie do nazwy (np. /dev/md0p1 można zmienić). Tak wygląda moja tablica fstab:

```

# /etc/fstab
# Created by anaconda on Thu May 23 17:20:54 2024
# Accessible filesystems, by reference, are maintained under
#   '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8)
#   for more info.
#
# After editing this file, run 'systemctl daemon-reload' to
#   update systemd
# units generated from this file.

UUID=1b08b3cb-906b-4e47-a277-6901a68a43c8 /
    ↳ xfs      defaults          0 0
UUID=00c4f666-b5a5-47ab-b17a-24d9bacfc6c /boot
    ↳ xfs      defaults          0 0
UUID=66736aaa-743c-48e6-86f9-4d454eee84fb    /dysksieciowy
    ↳ ext4    defaults          0 2
UUID=a0bf7f0e-c185-41d0-8b89-93fae0f625e4    /kopie    ext4
    ↳ defaults        0 2

```



Rysunek 52: Edycja /etc/fstab

Test po ponownym uruchominiu jest dostępny [tutaj](#).

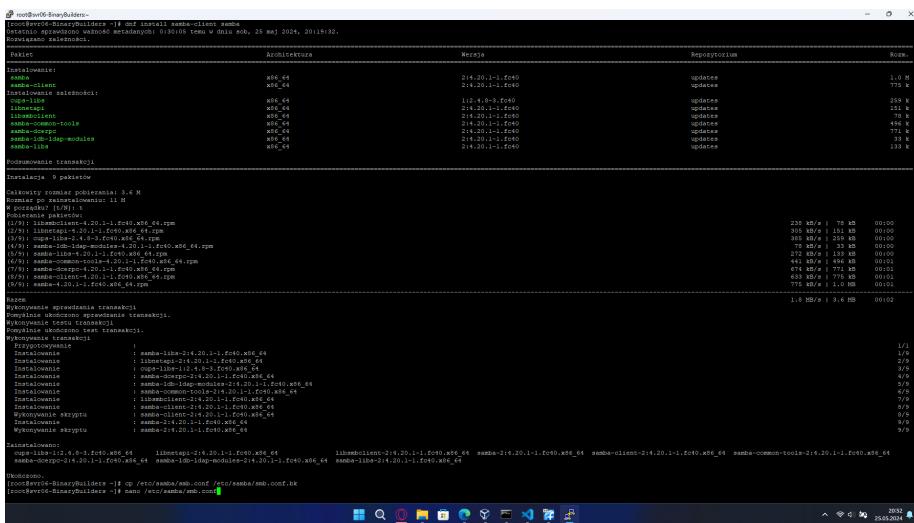
4.8 Samba – instalacja i konfiguracja

Aby zainstalować oprogramowanie do stworzenia serwera samba należy wydać polecenie:

```
sudo apt install samba-client samba -y
```

Po zainstalowaniu wymaganego oprogramowania wykonuję kopię zapasową pliku konfiguracyjnego samby. Można to zrobić komendą:

```
cp /etc/samba/smb.conf /etc/samba/smb.conf.bk
```



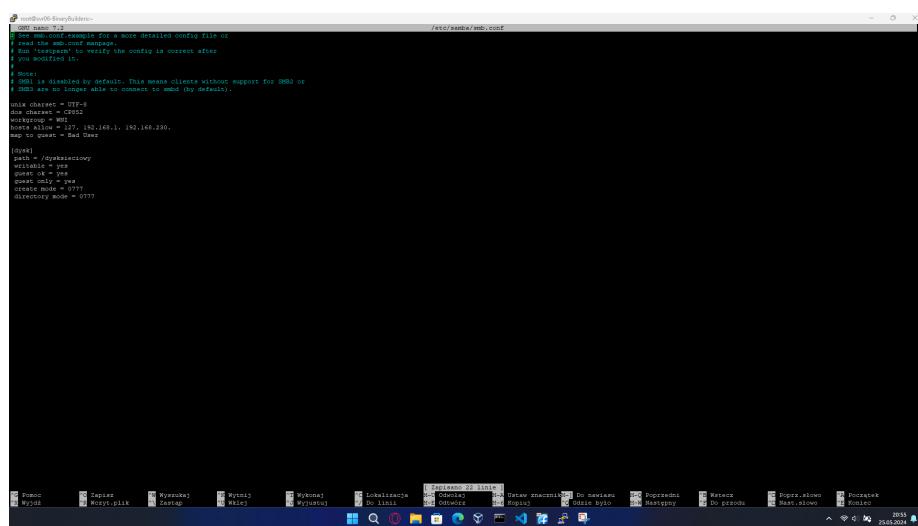
Rysunek 53: Samba – instalacja

Po wykonaniu kopii zapasowej można zabrać się za edycję /etc/samba/smb.conf. Tak wygląda ten plik u mnie:

```
# See smb.conf.example for a more detailed config file or
# read the smb.conf manpage.
# Run 'testparm' to verify the config is correct after
# you modified it.
#
# Note:
# SMB1 is disabled by default. This means clients without
# support for SMB2 or
# SMB3 are no longer able to connect to smbd (by default).

unix charset = UTF-8
dos charset = CP852
workgroup = BinaryBuilders
hosts allow = 127. 192.168.1. 192.168.230.
map to guest = Bad User
netbios name = sfs

[dysk]
path = /dysksieciowy
writable = yes
guest ok = yes
guest only = yes
create mode = 0777
directory mode = 0777
```



Rysunek 54: Edycja pliku /etc/samba/smb.conf

Następnym krokiem jest wprowadzenie kilku polecień:

```
testparam
systemctl start smb nmb
systemctl enable smb nmb
firewall-cmd --add-service=samba --permanent
firewall-cmd --reload
setsebool -P samba_export_all_rw on
```

Wytłumaczenie powyższych polecień:

- testparm

Testuje i wyświetla aktualne ustawienia konfiguracji Samby. Używane jest do sprawdzenia pliku konfiguracyjnego Samba (smb.conf) pod kątem błędów i wyświetlenia aktywnych ustawień.

- systemctl start smb nmb

Uruchamia usługi smb (serwer SMB) i nmb (serwer NetBIOS). Jest to wymagane, aby Samba mogła działać poprawnie i udostępniać zasoby w sieci.

- systemctl enable smb nmb

Ustawia usługi smb i nmb do automatycznego uruchamiania przy starcie systemu. Dzięki temu nie trzeba ich ręcznie uruchamiać po każdym restarcie serwera.

- firewall-cmd --add-service=samba --permanent

Dodaje regułę zapory sieciowej, aby na stałe zezwalać na ruch Samba. Umożliwia to komunikację Samby przez zaporę sieciową.

- firewall-cmd --reload

Przeładowuje ustawienia zapory sieciowej, aby zastosować wprowadzone zmiany. Jest to konieczne po dodaniu nowych reguł do zapory.

- setsebool -P samba_export_all_rw on

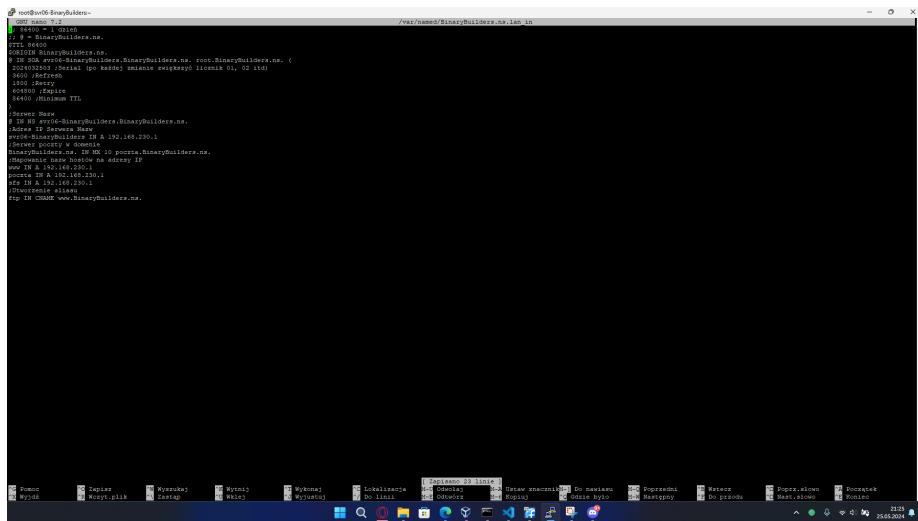
Ustawia w SELinux politykę, która pozwala Sambie na eksportowanie wszystkich udziałów z prawami do odczytu i zapisu. Dzięki temu Samba może zarządzać plikami z pełnym dostępem zgodnie z ustawieniami SELinux.

Na następnej stronie znajduje się zrzut ekranu z wykonaniem tych komend.

Rysunek 55: Samba – ustawienia SELinux oraz firewall

Aby udział był dostępny pod adresem \\sfs.firma.ns\dysk porzebna była zmiana konfiguracji DNS. Wymagane było dodanie linijki w pliku /var/named/BinaryBuilders.ns.lan_in:

sfs IN A 192.168.230.1



Rysunek 56: Edycja konfiguracji DNS

Po restartie usługi named nie pozostaje nic innego jak sprawdzenie czy podłączenie do udziału działa. Test jest dostępny [tutaj](#).

4.9 HTTP – instalacja i konfiguracja

Aby zainstalować oprogramowanie do stworzenia serwera HTTP należy wydać polecenie:

```
sudo dnf install httpd -y
```

Rysunek 57: Instalacja serwera HTTP

W kolejnym kroku wykonuje kopie oryginalnego pliku konfiguracyjnego serwera http. Robie to nastepujaca komenda:

```
cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.org
```

w kolejności zabieram się za edycję wcześniej wspomnianego pliku. Edycja tego pliku obejmuje aż cztery zdjęcia.

Rysunek 58: Edycja /etc/httpd/conf/httpd.conf – część pierwsza

```

root@sw06-dvayubalen-:
[ 0 ~ ]# cat /etc/httpd/conf/httpd.conf
1 # Allow access to content within /var/www.
2
3 <Directory "/var/www">
4     AllowOverride All
5     Options +Indexes FollowSymLinks +SymLinksIfOwnerMatch +ExecCGI MultiViews
6
7     # Possible values for the Options directive are "None", "All",
8     # or any combination of the keywords:
9     #   +FollowSymLinks:      Require all granted
10    #   -FollowSymLinks:     Require all denied
11
12    # Further relax access to the default document root:
13 <Directory "/var/www/html">
14     # Possible values for the Options directive are "None", "All",
15     # or any combination of the keywords:
16     #   +FollowSymLinks:      Require all granted
17     #   -FollowSymLinks:     Require all denied
18     # Note that "MultiViews" must be named *explicitly* --- "Options All"
19     # doesn't give it to you.
20
21     # The Options directive is both complicated and important. Please see
22     # http://httpd.apache.org/docs/2.4/sections/mods/section-options.html#options
23     # for more information.
24
25     Options Indexes FollowSymLinks
26
27     # AllowOverride controls what directives may be placed in .htaccess files.
28     # It can be "All", "None", or any combination of the keywords:
29     #   +Options FileInfo AuthConfig Limit
30     #   -AllowOverride All
31
32     # Controls who can get stuff from this server.
33     Options +Indexes All
34
35     # DirectoryIndex sets the file that Apache will serve if a directory
36     # is requested.
37
38 <DirectoryIndex index.html>
39 </DirectoryIndex>
40
41 # The following lines prevent .htaccess and .htpasswd files from being
42 # viewed by Web clients.
43 <Files ".ht*">
44     Require all denied
45 </Files>
46
47 # ErrorLog: The location of the error log file.
48 # If you do not specify an ErrorLog directive within a <VirtualHost>
49 # container, then host's errors will be logged there and not here.
50 # LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\""
51 # combinedLog
52
53 <ErrorLog "/var/log/error_log">
54
55 # LogLevel: Control the number of messages logged to the error log.
56 # Possible values include: debug, info, notice, warn, error, crit,
57 # emergency.
58
59 LogLevel warn
60
61 <IfModule log_config_module>
62     # The following directives define some format nickname for use with
63     # the LogFormat and CustomLog directives below.
64     LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
65     LogFormat "%h %l %u %t \"%r\" %s %b" common
66
67 </IfModule log_config_module>

```

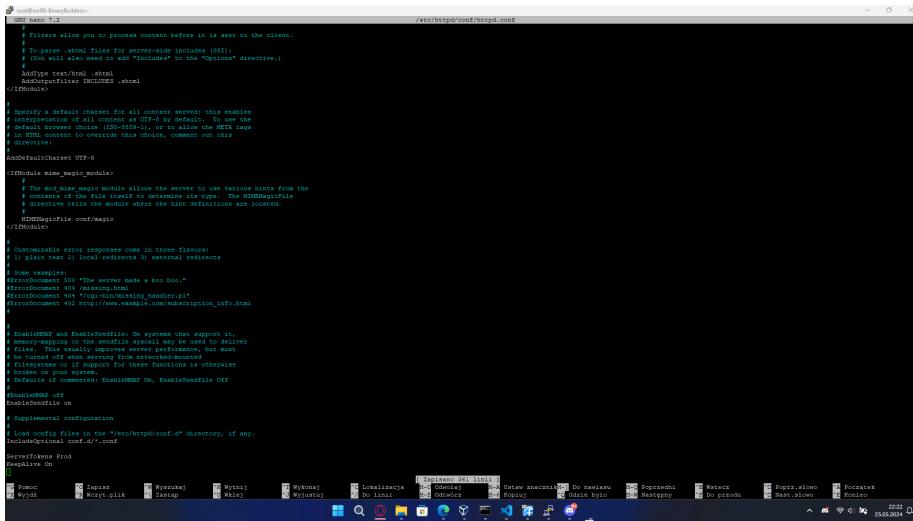
Rysunek 59: Edycja /etc/httpd/conf/httpd.conf – część druga

```

root@sw06-dvayubalen-:
[ 0 ~ ]# cat /etc/httpd/conf/httpd.conf
1 <ErrorLog "/var/log/error_log">
2
3 # If you do not specify an ErrorLog directive within a <VirtualHost>
4 # container, then host's errors will be logged there and not here.
5 # LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\""
6 # combinedLog
7
8 <IfModule log_config_module>
9     # The following directives define some format nickname for use with
10    # the LogFormat and CustomLog directives below.
11    LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
12    LogFormat "%h %l %u %t \"%r\" %s %b" common
13
14 </IfModule log_config_module>

```

Rysunek 60: Edycja /etc/httpd/conf/httpd.conf – część trzecia



```

[root@www05 ~]# cat /etc/httpd/conf/httpd.conf
# This file contains the configuration for the Apache web server.  See
# /usr/share/doc/httpd-2.4.10/html/manual/mod/mod_config.html
# for details on configuration.  Several sections of this file are
# commented out--see the comments in the file for information on how to
# enable these features.

# DocumentRoot: The directory where your documents are stored.
DocumentRoot "/var/www/html"

# The port on which to accept incoming connections.
Port 80

# AddType: tells the server what type of file it is.
# AddOutputFilter: tells the server what filter to run on a file.
# AddEncoding: tells the server what encoding to use for a file.
# AddLanguage: tells the server what language to use for a file.
# AddHandler: tells the server what handler to use for a file.
# AddType text/html .html
# AddOutputFilter INCLUDES .html
# AddLanguage pl .pl

# Specify a default charset for all content served; this enables
# the server to use various tricks to determine the encoding of files
# without having to guess.  The X-MIMEMIME-Charset header can be used
# to force a different choice (ISO-8859-1), or to allow the META tag
# (Content-Type) to override this choice, comment out this
# directive.
AddDefaultCharset UTF-8

#(module name mod_cgi_module)
# The mod_cgi module allows the server to use various blocks from the
# contents of the file itself to determine its type. The X-MIMEMIME-Charset
# directive tells the module where the main definitions are located.
# AddHandler cgi-script .cgi
# AddType application/x-gzip .gz .tgz

#(module)
#(directive error responses come in three flavors)
# (1) plain text (2) local redirects (3) external redirects
# See examples:
ErrorDocument 500 "The server made a boo boo."
ErrorDocument 404 "/wp-content/themes/handler.php"
ErrorDocument 403 http://www.example.com/subscription_info.html

#(directive)
#(directive) and #(directive) are systems that support it.
#(directive) of the mod_cgi module may not be used to deliver
# files. This usually improves server performance, but must
# be used if you want to use filters or if support for these functions is otherwise
# required.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#EnableMMAP off
#EnableSendfile on
#(directive) configuration
# Load config files in the "/etc/httpd/conf.d/" directory, if any.
IncludeOptional conf.d/*
#(directive) Prior
Keepalive On
#(directive)

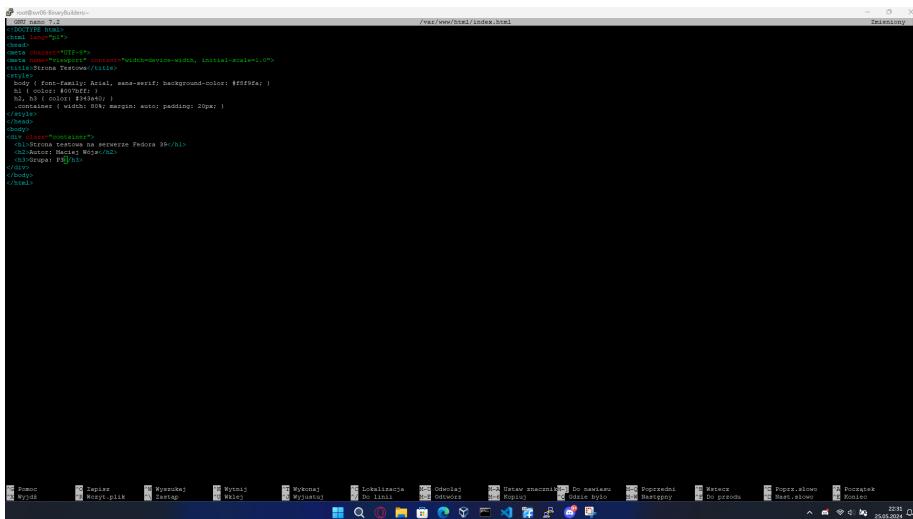
```

Rysunek 61: Edycja /etc/httpd/conf/httpd.conf – część czwarta

W następnym kroku wykonuje następujące polecenia:

```
firewall-cmd --add-server=http --permanent
firewall-cmd --reload
```

Stworzenie poglądowej strony html w lokalizacji /var/www/html/



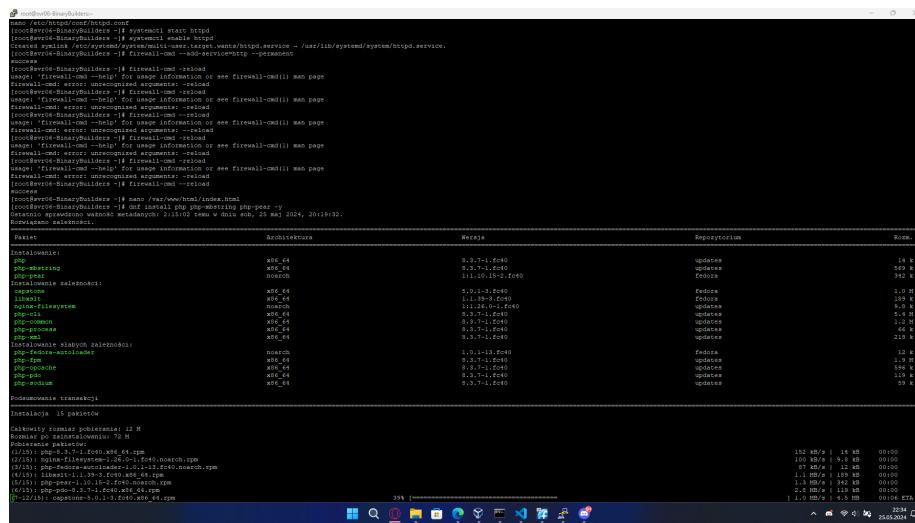
Rysunek 62: Strona html – domyślna strona serwera

Test działania web serwera dostępny jest **tutaj**.

4.10 PHP – instalacja i konfiguracja

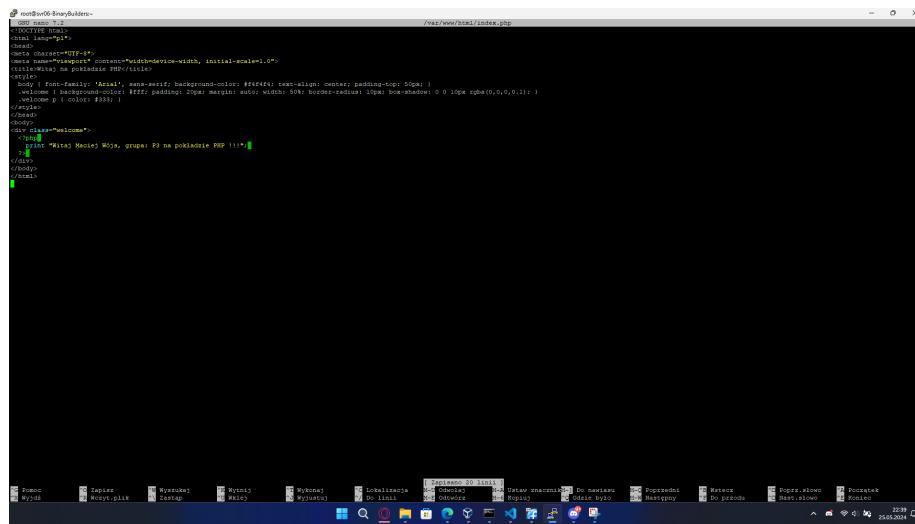
Aby zainstalować PHP należy wydać polecenie:

```
sudo dnf install php php-mbstring php-pear -y
```



Rysunek 63: PHP – instalacja

Po zainstalowaniu wymaganych pakietów tworzę prostą stronę internetową wykorzystującą PHP.



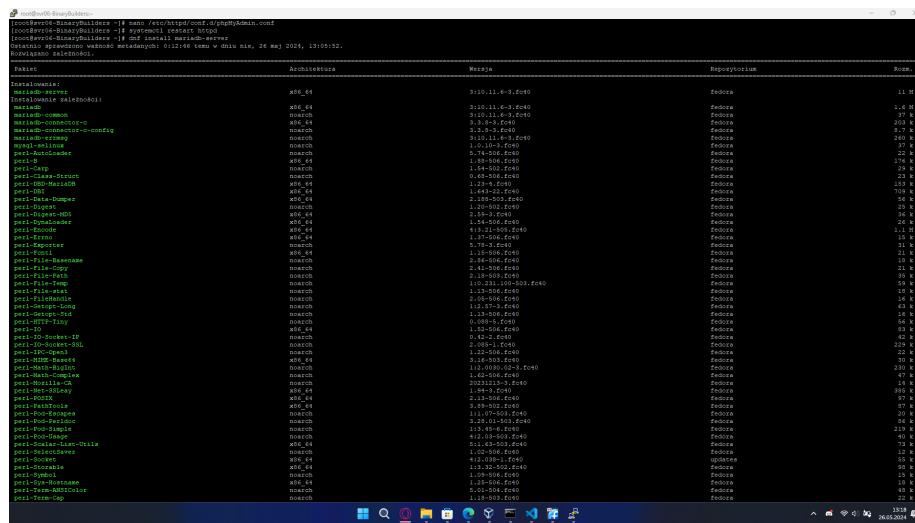
Rysunek 64: PHP – stworzenie strony internetowej

Test działania strony dostępny jest [tutaj](#).

4.11 mariadb – instalacja i konfiguracja

Aby zainstalować silnik bazydanych mariadb należy wydać polecenie:

```
sudo dnf install mariadb-server -y
```



Rysunek 65: mariadb – instalacja usługi

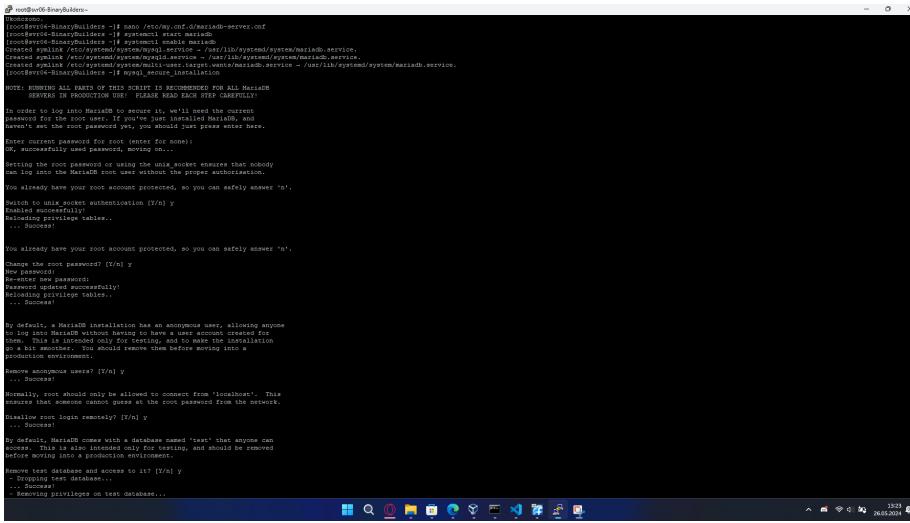
Do pliku /etc/my.cnf.d/mariadb-server.cnf w sekcji [mysqld] dodałem linię:

```
character-set-server=utf8
```



Rysunek 66: mariadb – edycja pliku konfiguracyjnego

Restart usługi mariadb oraz instalacja serwera MySQL.



```
root@DevW8-Server:~# ./mysql_secure_installation
-- MySQL binary builder -- 1.0 systemct start mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
[root@DevW8-Server:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
CONNECTIONS AS IT WILL SECURE YOUR SERVER.

It is recommended to run MariaDB or MySQL as the "root" user.
This means you log into MariaDB or MySQL as "root" and enter the
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
+-----+
|          |
|          |
|          |
|          |
|          |
+-----+
Setting the root password or using the unix socket ensures that nobody
can log into the MariaDB root user without the proper authorization.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables...
... Success!

You already have your root account protected, so you can safely answer 'n'.
Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables...
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!
Normally, root should only be allowed to connect from 'localhost'. This
means that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
- Removing privileges on test database...
... Success!

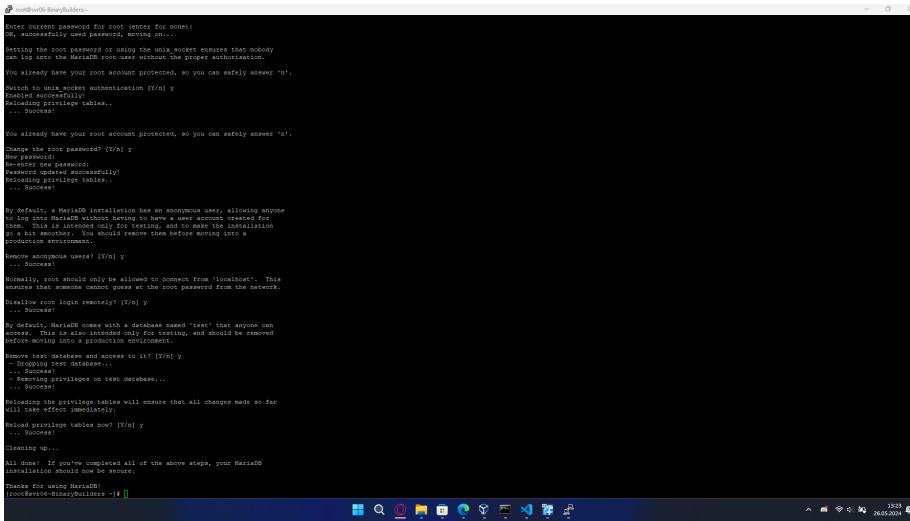
All privilege changes will ensure that all changes made so far
will take effect immediately.

Reloading privilege tables now? [Y/n] Y
... Success!
Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
prosderice@DevW8-Server:~#
```

Rysunek 67: MySQL – instalacja część pierwsza



```
root@DevW8-Server:~# ./mysql_secure_installation

Enter current password for root (enter for none):
+-----+
|          |
|          |
|          |
|          |
|          |
+-----+
You already have your root account protected, so you can safely answer 'n'.

Switch to unix socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables...
... Success!

You already have your root account protected, so you can safely answer 'n'.
Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables...
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!
Normally, root should only be allowed to connect from 'localhost'. This
means that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
- Removing privileges on test database...
... Success!

All privilege changes will ensure that all changes made so far
will take effect immediately.

Reloading privilege tables now? [Y/n] Y
... Success!
Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

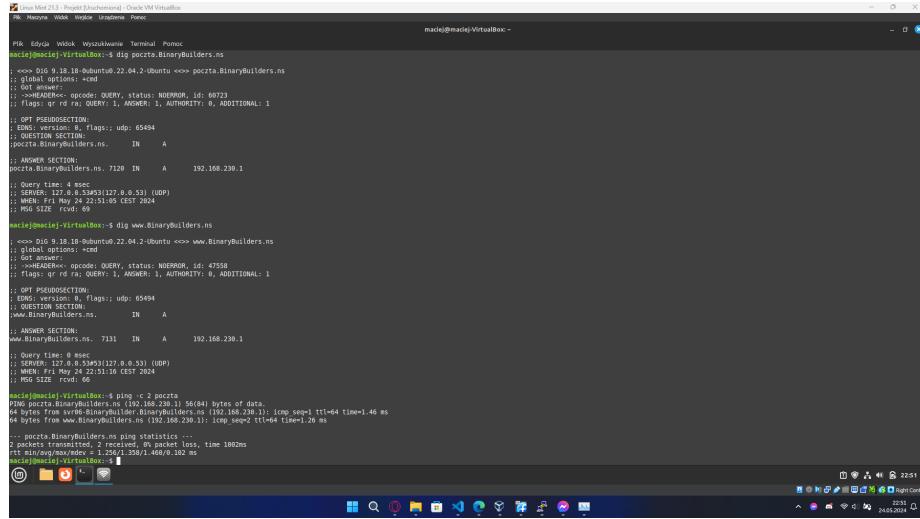
Thanks for using MariaDB!
prosderice@DevW8-Server:~#
```

Rysunek 68: MySQL – instalacja część druga

Test podłączenia do serwera MySQL dostępny jest **tutaj**.

5 Testy działania wdrożonych usług

5.1 DNS

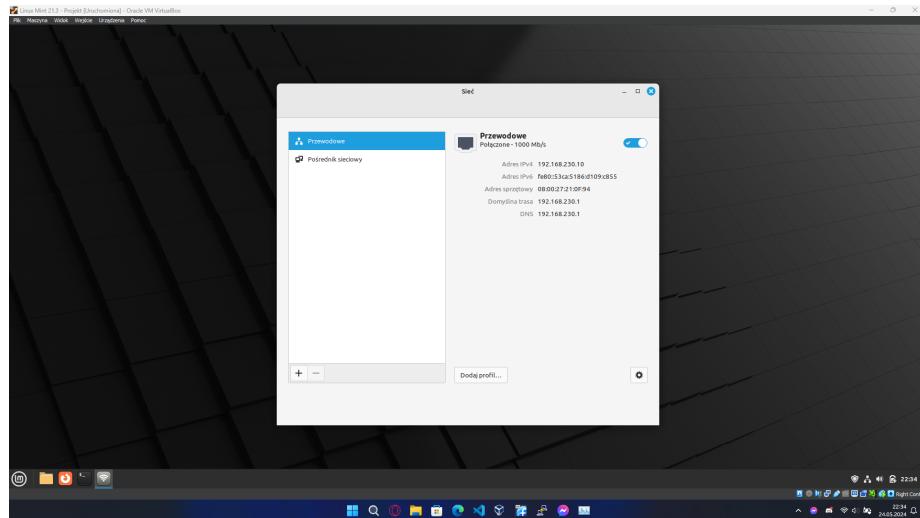


```
maciej@maciej-VirtualBox:~$ dig pocita.BinaryBuilders.ns
;; global options: +cmd
maciej@maciej-VirtualBox:~$ dig www.BinaryBuilders.ns
;; global options: +cmd
maciej@maciej-VirtualBox:~$ ping -c 2 pocita
PING pocita.BinaryBuilders.ns (192.168.230.1) 56(84) bytes of data.
64 bytes from www.Bin... (192.168.230.1): icmp_seq=1 ttl=64 time=1.46 ms
64 bytes from www.Bin... (192.168.230.1): icmp_seq=2 ttl=64 time=1.46 ms
--- pocita.Bin... ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 100ms
maciej@maciej-VirtualBox:~$
```

Rysunek 69: Test DNS

Jak widać na powyższym zdjęciu system w sieci wewnętrznej dostaje odpowiedzi od serwera na zapytania, co sugeruje że usługa DNS została skonfigurowana poprawnie.

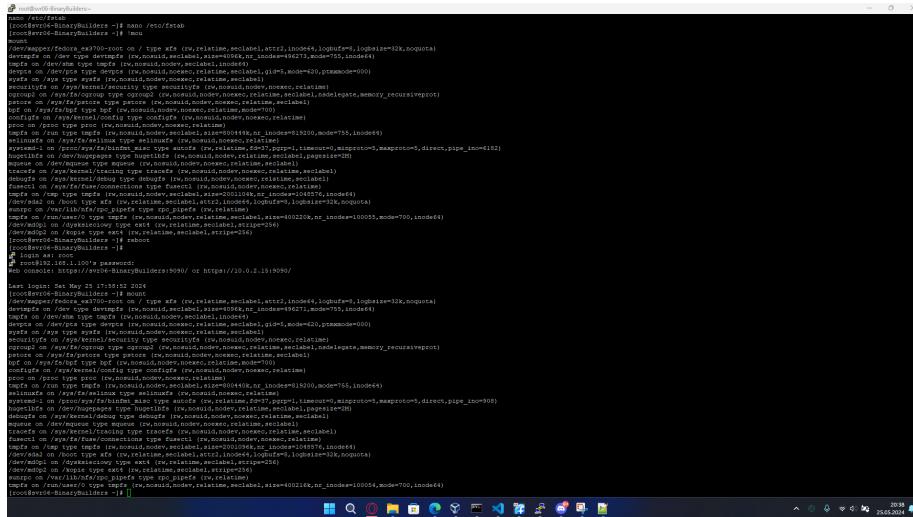
5.2 DHCP



Rysunek 70: Instalacja DHCP

Jak widać na powyższym zdjęciu karta w systemie klienta ustawiona na sieć wewnętrzną dostała poprawny adres IP, adres bramy domyślnej i DNS. Na zdjęciu również widać że pula DHCP działa poprawnie.

5.3 Raid 5

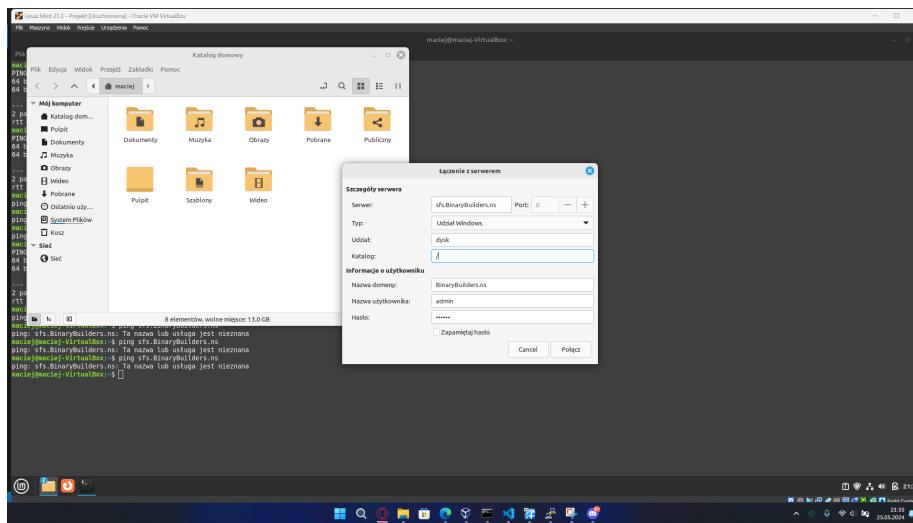


```
root@evr06-BinaryBuilders:~# mdadm --create /dev/md0 --level=5 --raid-devices=2 /dev/sda1 /dev/sdb1
root@evr06-BinaryBuilders:~# mdadm -D /dev/md0
root@evr06-BinaryBuilders:~# cat /proc/mdstat
root@evr06-BinaryBuilders:~#
```

Rysunek 71: Test automatycznego montowania partycji po ponownym uruchomieniu serwera

Jak widać na powyższym zrzucie ekranu po restarcie serwera partycje są montowane poprawnie.

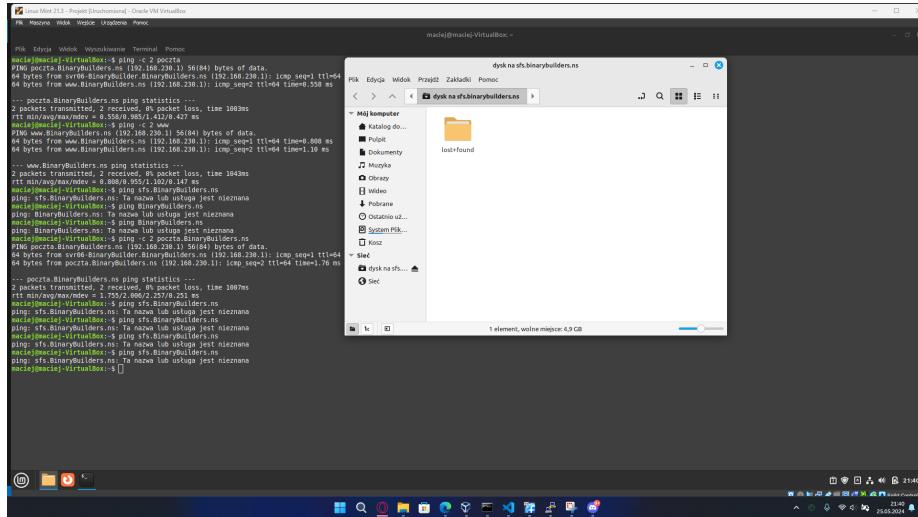
5.4 Samba



Rysunek 72: Samba – próba podłączenia się do udziału na serwerze

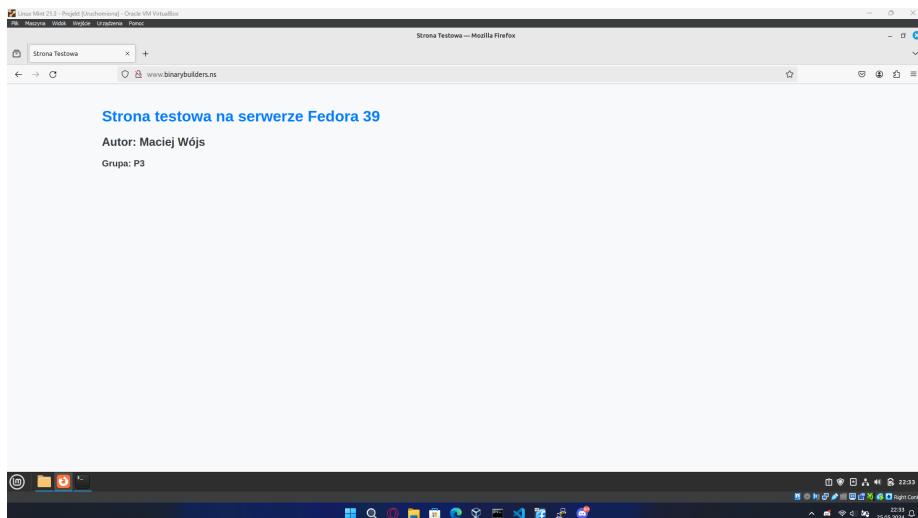
Aby podłączyć się z serwerem Samby należy otworzyć menadżer plików następnie otworzyć zakładkę Plik w lewym górnym rogu, następnie połącz z

serwerem, kolejno w typie należy wybrać Udział Windows, ostatecznie należy wypełnić wymagane dane. Przykładowa próba podłączenia powyżej, a efekt tego działania poniżej.



Rysunek 73: Samba – wynik poprzedniego kroku

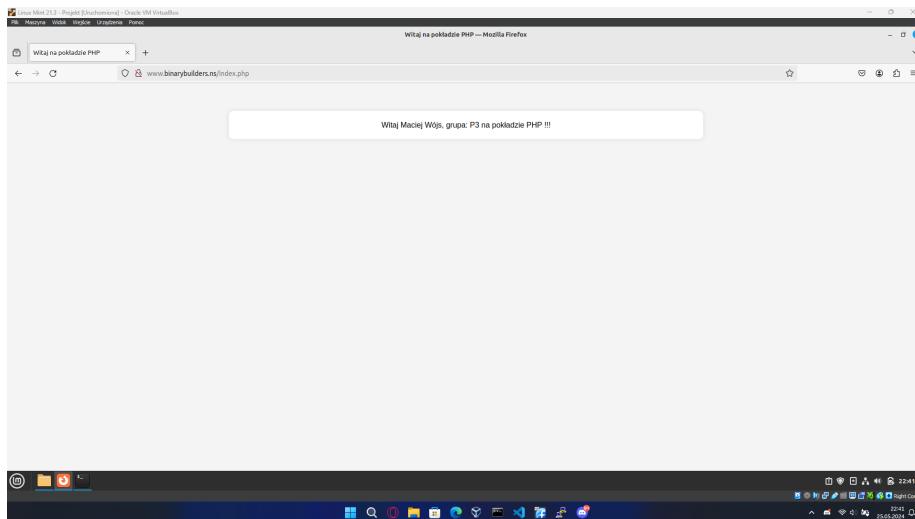
5.5 HTTP



Rysunek 74: Test działania serwera WWW

Jak widać na zdj&eju powyżej web serwer działa poprawnie.

5.6 PHP



Rysunek 75: Strona html + PHP

Jak można zauważyc strona wykorzystujaca PHP dziala poprawnie, nie wyskoczył żaden błąd dotyczący błędnej konfiguracji PHP, czy błędneego użycia go na stronie.

5.7 MySQL

```
[root@Server04-BinaryBuilders ~]# mysql -u root -p
Welcome to the MariaDB monitor.  Commands end with ; or \q.
Your MariaDB connection id is 1,
    thread_id=1033  query_time=0.000000  slow_query_time=0.000000  connect_time=0
Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> select user,host,password from mysql.user;
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| root | %    |          |
|      | localhost |          |
|      | localhost | invalid |
| mysql | %    |          |
+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| sys                |
+--------------------+
3 rows in set (0.001 sec)

MariaDB [(none)]> exit
[root@Server04-BinaryBuilders ~]# firewall-cmd --add-service=mysql --permanent
[root@Server04-BinaryBuilders ~]# firewall-cmd --reload
[root@Server04-BinaryBuilders ~]# [ ]
```

Rysunek 76: Test usługi mariadb (MySQL)

Jak widać połaczenie zz bazą danych działa. Następnym i ostatnim krokiem jest dodanie usługi MySQL do dozwolonych usług w zaporze ogniwowej.

6 Kod skryptu BASH, oraz tablica crontab

FFFFFFFFFFFFn

7 Wnioski

EEEEEEEEE

8 Literatura

- [1] *Kubernetes Blog*. Dostęp: 2024-01-19. URL: <https://kubernetes.io/blog/>.
- [2] *Kubernetes Documentation*. Dostęp: 2024-01-19. URL: <https://kubernetes.io/docs/>.
- [3] *Kubernetes GitHub Repository*. Dostęp: 2024-01-19. URL: <https://github.com/kubernetes/kubernetes>.