

# **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

<b>1 Założenia projektowe – wymagania</b>	<b>4</b>
<b>2 Opis użytych technologii</b>	<b>5</b>
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
<b>3 Schemat logiczny projektowanej infrastruktury sieciowej</b>	<b>7</b>
<b>4 Procedury instalacyjne poszczególnych usług</b>	<b>8</b>
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
4.12 phpMyAdmin – instalacja i konfiguracja . . . . .	49
4.13 UserDir na serwerze HTTP – konfiguracja . . . . .	50
4.14 WordPress – instalacja i konfiguracja . . . . .	51
<b>5 Testy działania wdrożonych usług</b>	<b>55</b>
5.1 DNS . . . . .	55
5.2 DHCP . . . . .	55
5.3 Raid 5 . . . . .	56
5.4 Samba . . . . .	56

5.5	HTTP . . . . .	57
5.6	PHP . . . . .	58
5.7	MySQL . . . . .	58
5.8	phpMyAdmin . . . . .	59
5.9	UserDir – serwer http . . . . .	60
5.10	WordPress . . . . .	60
<b>6</b>	<b>Kod skryptu BASH, oraz tablica crontab</b>	<b>61</b>
<b>7</b>	<b>Wnioski</b>	<b>61</b>
<b>8</b>	<b>Literatura</b>	<b>62</b>

## 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	phpMyAdmin – instalacja . . . . .	49
70	Konfiguracja pliku /etc/httpd/conf.d/phpMyAdmin.conf . . . . .	49
71	Konfiguracja pliku /etc/httpd/conf.d/userdir.conf . . . . .	50
72	Dodanie użytkownika maciej . . . . .	50
73	Stworzenie strony użytkownia maciej . . . . .	51
74	WordPress – instalacja . . . . .	51
75	Stworzenie bazy danych dla WordPress'a . . . . .	52
76	Edycja pliku /etc/httpd/conf.d/wordpress.conf . . . . .	53
77	Edycja pliku /etc/wordpress/wp-config.php . . . . .	53
78	Instalacja WordPress . . . . .	54
79	Instalacja WordPress – sukces . . . . .	54
80	Test DNS . . . . .	55
81	Instalacja DHCP . . . . .	55
82	Test automatycznego montowania . . . . .	56
83	Samba – próba podłączenia się do udziału na serwerze . . . . .	56
84	Samba – wynik poprzedniego kroku . . . . .	57
85	Test działania serwera WWW . . . . .	57
86	Strona html + PHP . . . . .	58
87	Test usługi mariadb (MySQL) . . . . .	58
88	Test usługi phpMyAdmin – część pierwsza . . . . .	59
89	Test usługi phpMyAdmin – część druga . . . . .	59
90	MySQL – instalacja część druga . . . . .	60
91	Dashboard WordPress'a . . . . .	60

# 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**

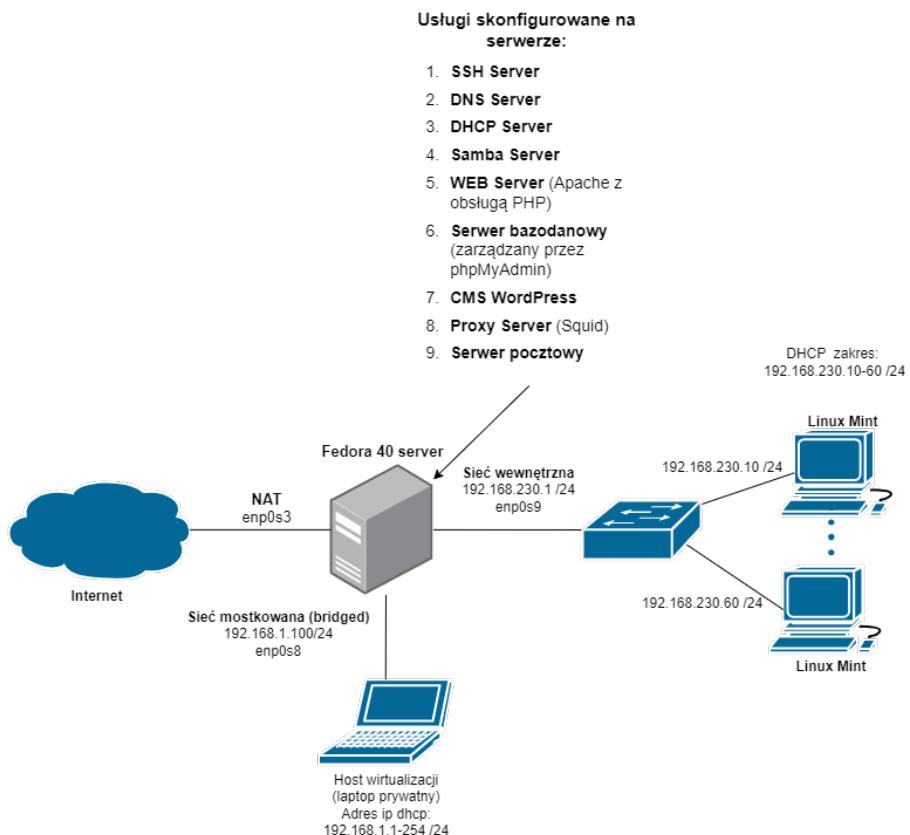
### **2.13.1 Skrypty Bash**

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

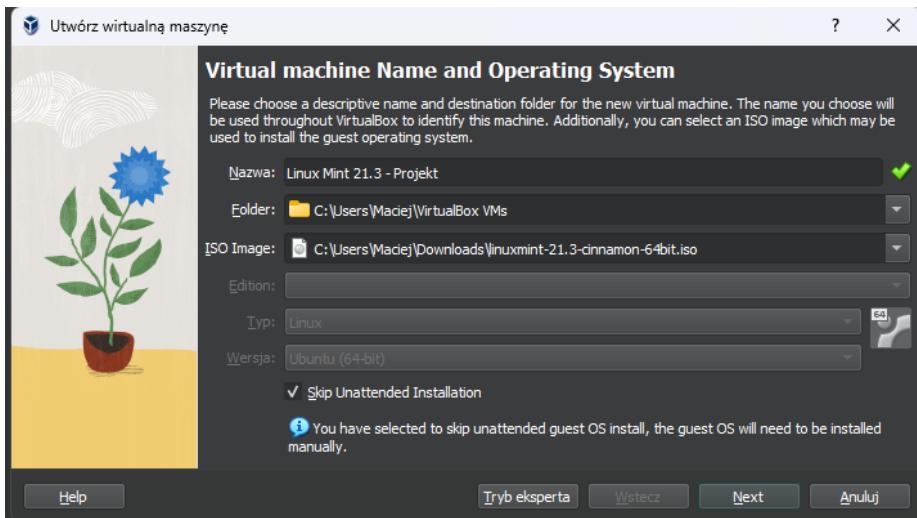


Rysunek 1: Schemat logiczny sieci

## 4 Procedury instalacyjne poszczególnych usług

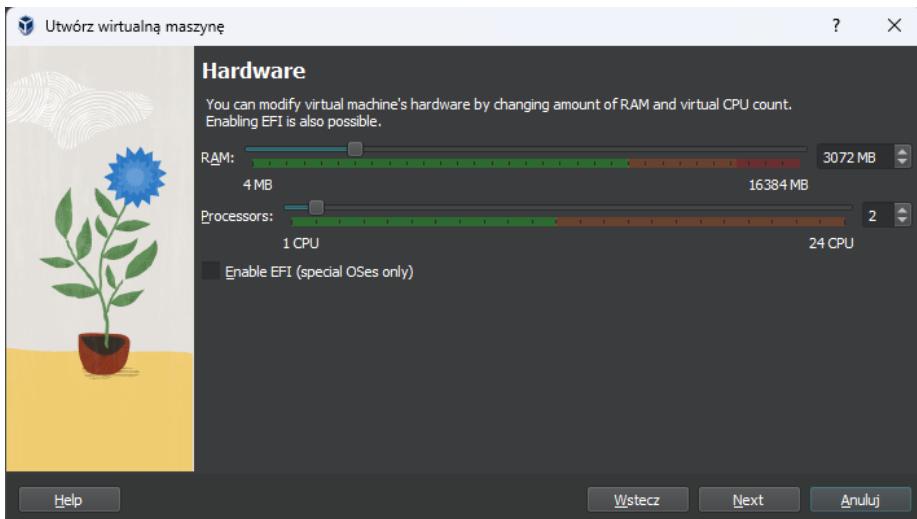
### 4.1 Instalacja systemu klienta – Linux Mint

#### 4.1.1 Proces instalacji



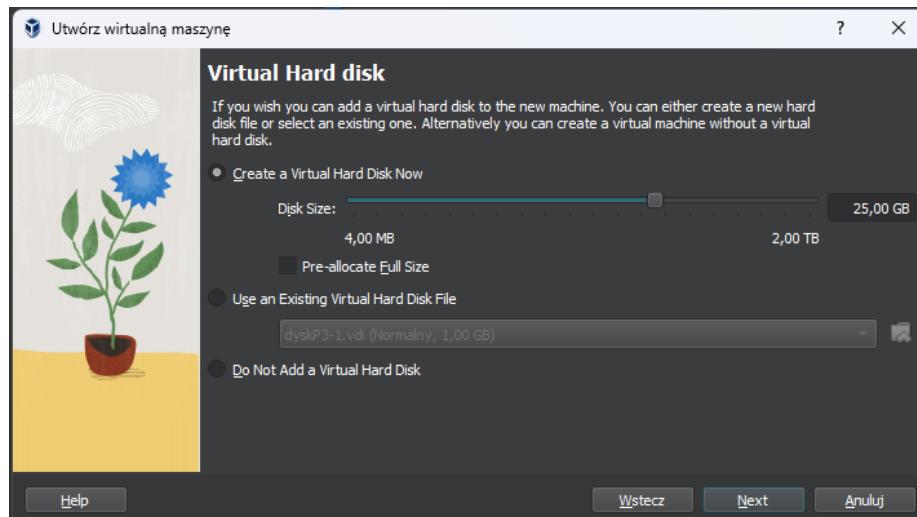
Rysunek 2: Tworzenie nowej maszyny wirtualnej. Ustawienia nazwy, lokalizacji dysku oraz wybór pliku ISO systemu operacyjnego.

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.



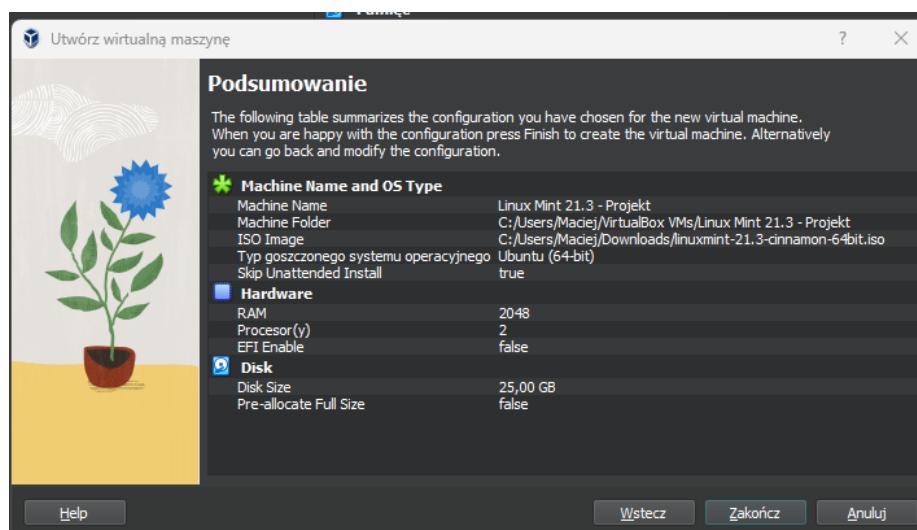
Rysunek 3: Przydzielanie zasobów maszynie wirtualnej, takich jak pamięć RAM i procesor.

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



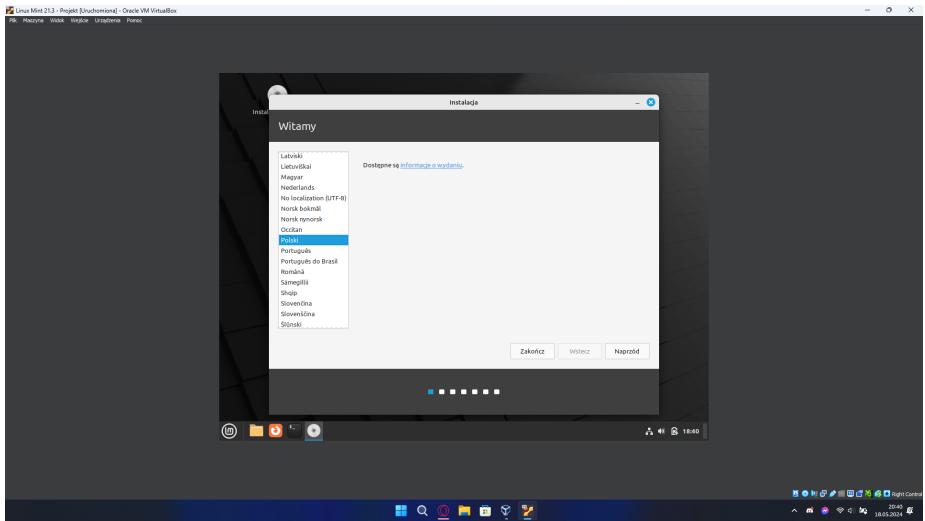
Rysunek 4: Określenie rozmiaru dysku wirtualnego.

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



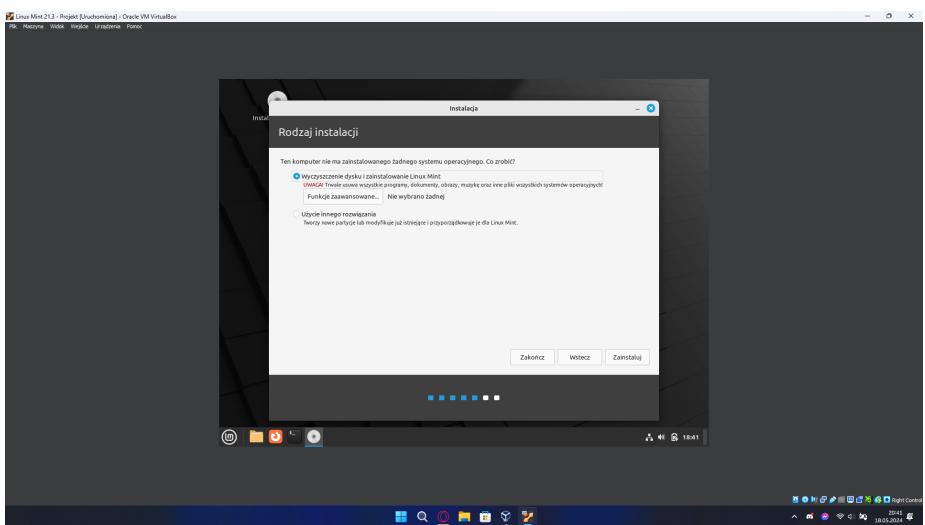
Rysunek 5: Podsumowanie konfiguracji maszyny wirtualnej przed rozpoczęciem instalacji systemu.

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



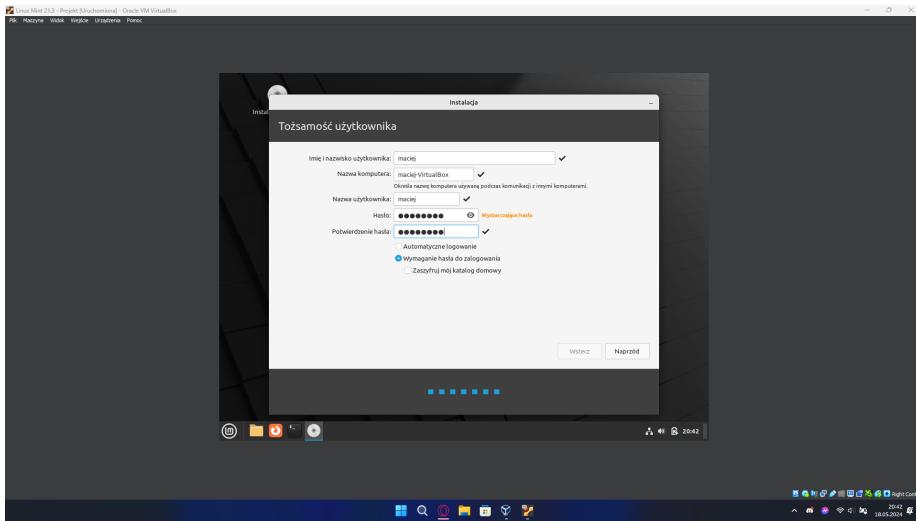
Rysunek 6: Rozpoczęcie instalacji Linux Mint – wybór języka instalacji.

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



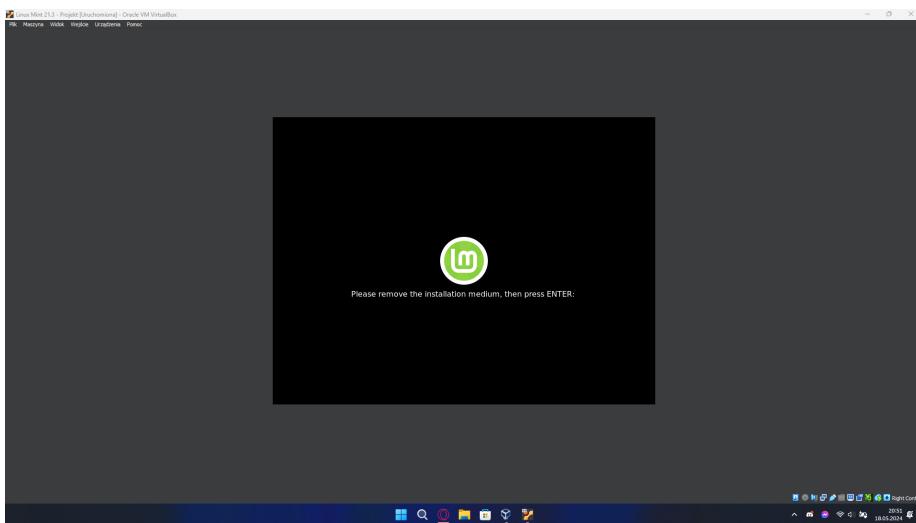
Rysunek 7: Wybór trybu instalacji na dysku twardym.

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.

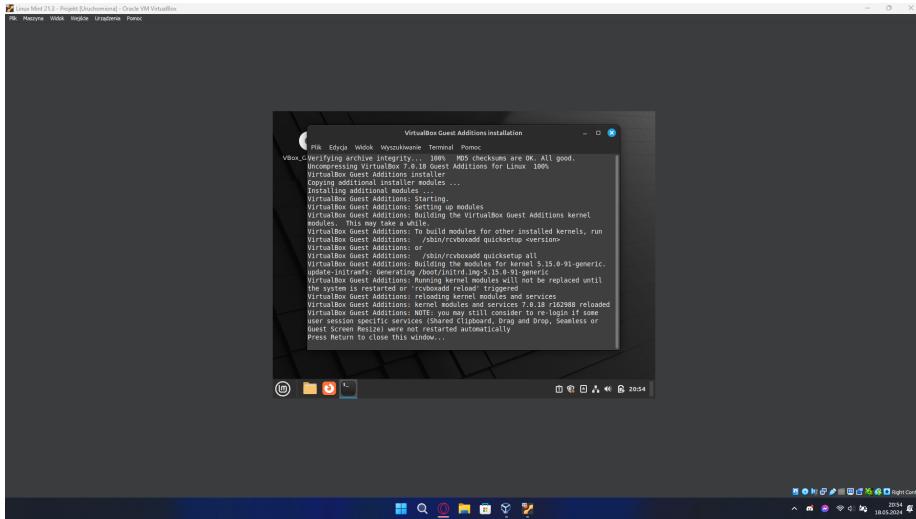
Kolejnym krokiem jest utworzenie konta użytkownika, wprowadzenie nazwy użytkownika, hasła oraz nazwy komputera.



Rysunek 9: Zakończenie instalacji systemu Linux Mint.

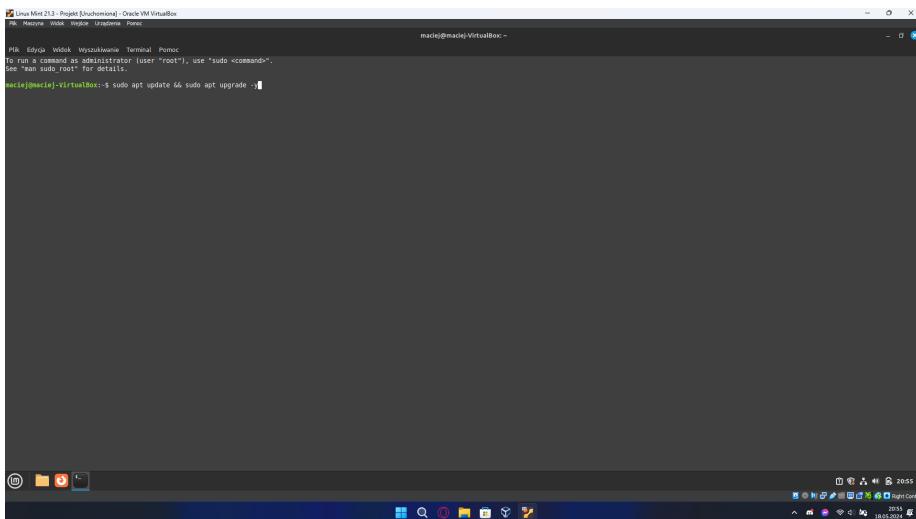
Wyświetlony zostaje monit z prośbą o usunięcie nośnika instalacyjnego. Po zakończeniu instalacji system wyświetla ekran informujący o pomyślnym zakończeniu procesu.

#### 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.

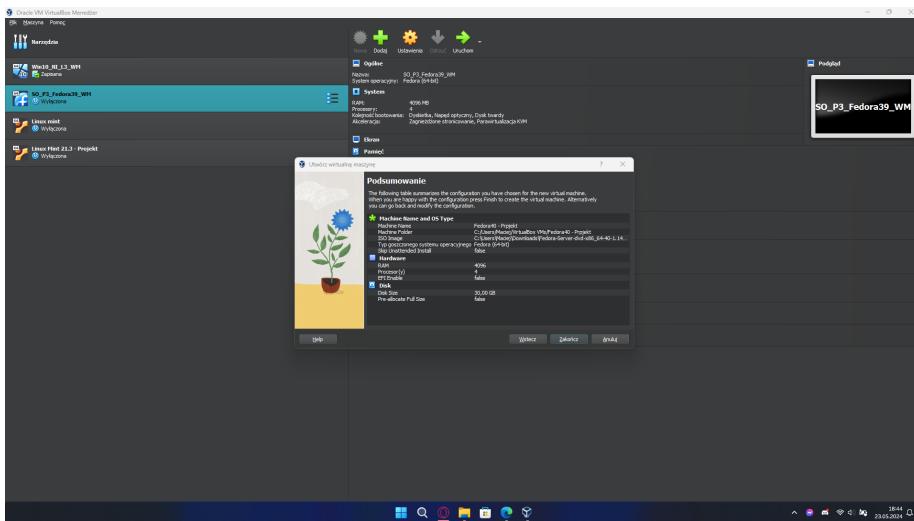


Rysunek 11: Aktualizacja pakietów systemowych.

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

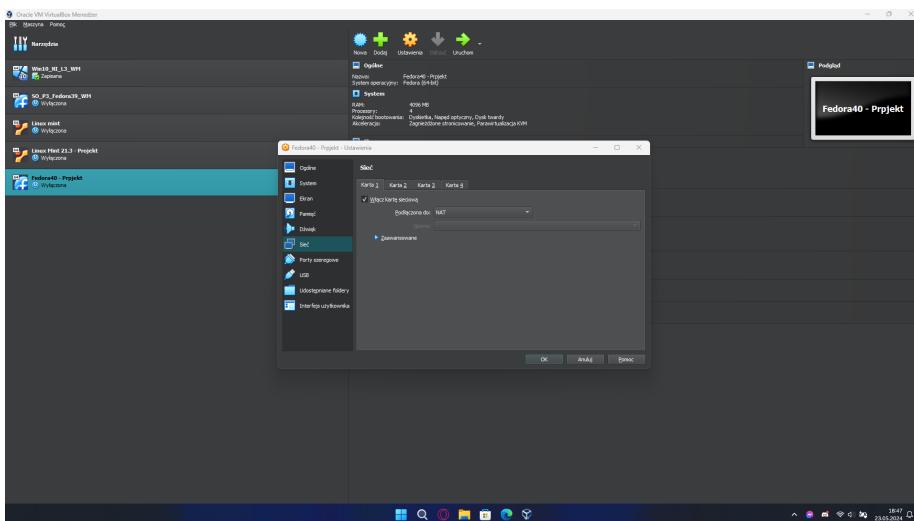
## 4.2 Instalacja serwera – Fedora 40

### 4.2.1 Proces instalacji



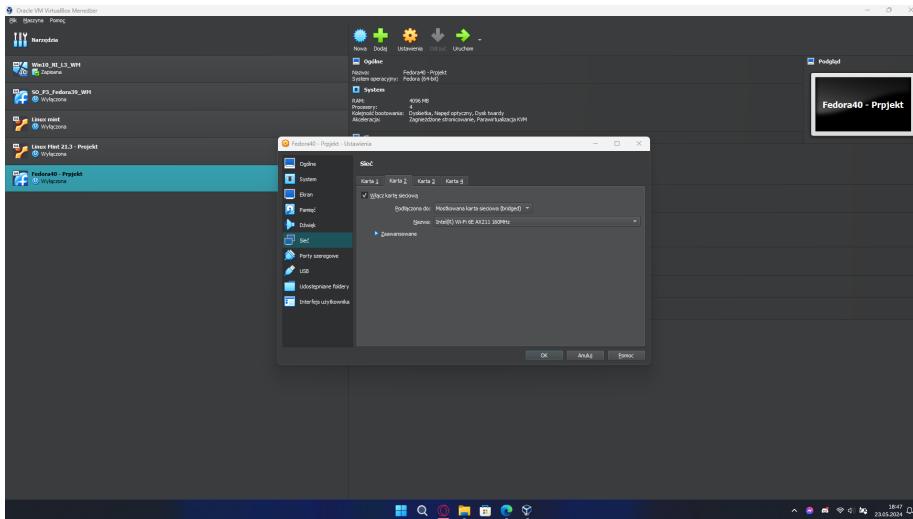
Rysunek 12: Analogicznie jak w przypadku instalacji Linux Mint – wymagane jest ustawienie nazwy maszyny wirtualnej, przydzielenie jej zasobów, ustalenie rozmiaru dysku. Powyższe zdjęcie ukazuje ekran z podsumowaniem wybranych opcji

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



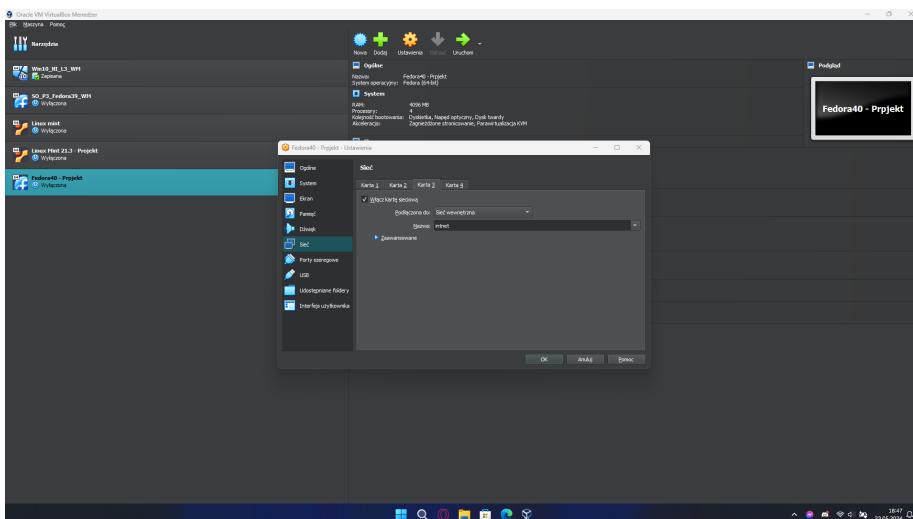
Rysunek 13: Dodanie pierwszej karty sieciowej – NAT

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.



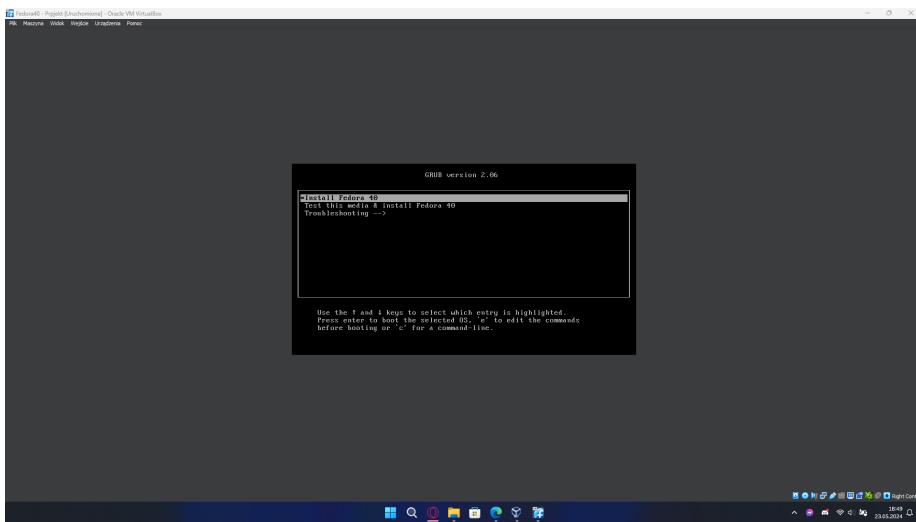
Rysunek 14: Dodanie pierszej drugiej karty sieciowej – sieć mostkowana (bridged)

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.



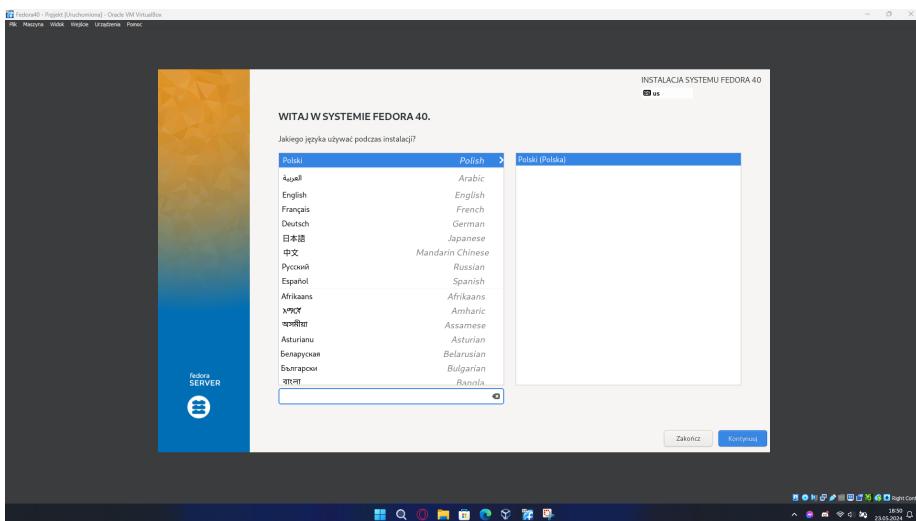
Rysunek 15: Dodanie pierszej trzeciej karty sieciowej – sieć wewnętrzna

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.



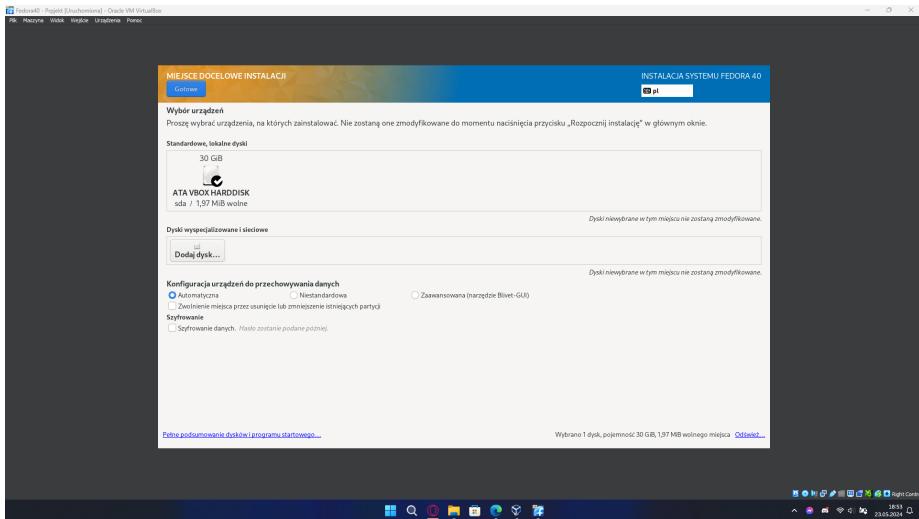
Rysunek 16: Uruchomienie instalatora Fedory.

W następnym kroku wybiera się język instalatora oraz układ klawiatury.



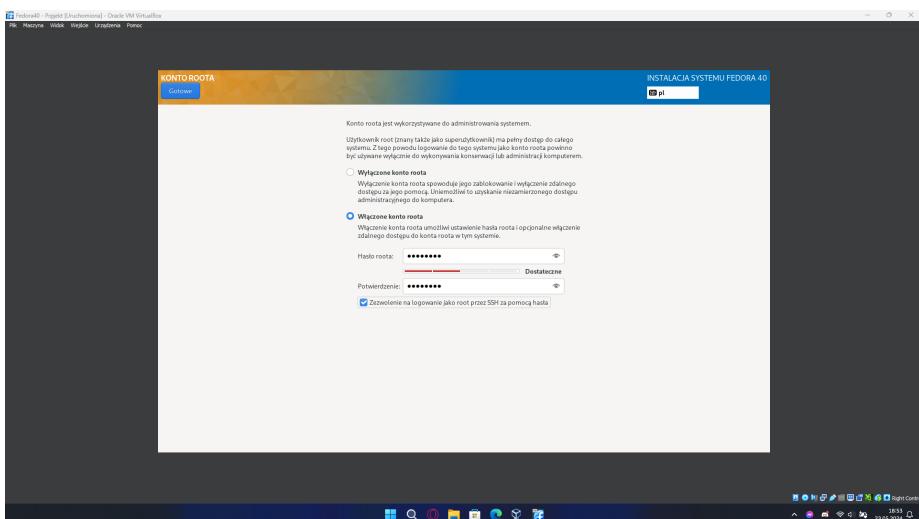
Rysunek 17: Rozpoczęcie instalacji Fedora 40 – wybór języka instalacji.

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).



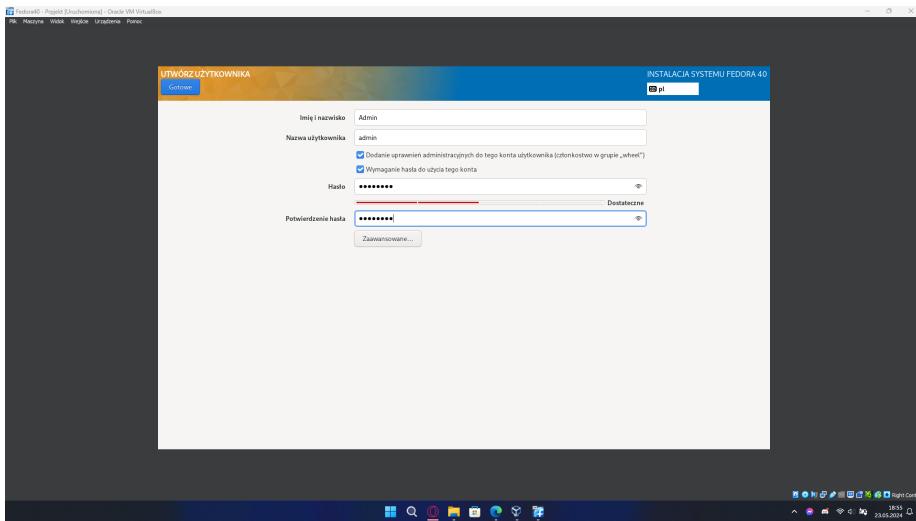
Rysunek 18: Wybór dysku na którym zostanie zainstalowany serwer

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.



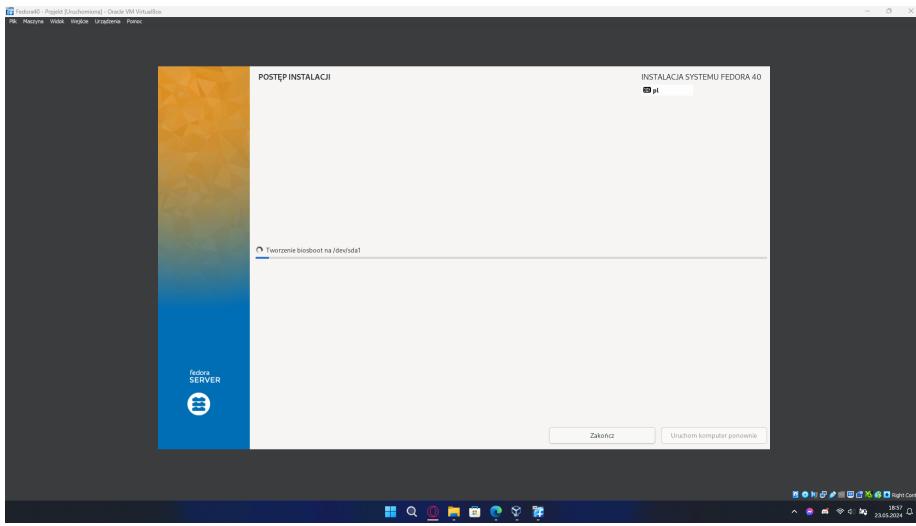
Rysunek 19: Ustawienie konta roota – włączenie konta, ustawienie hasła i zezwolenie na połączenie ssh jako root

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).



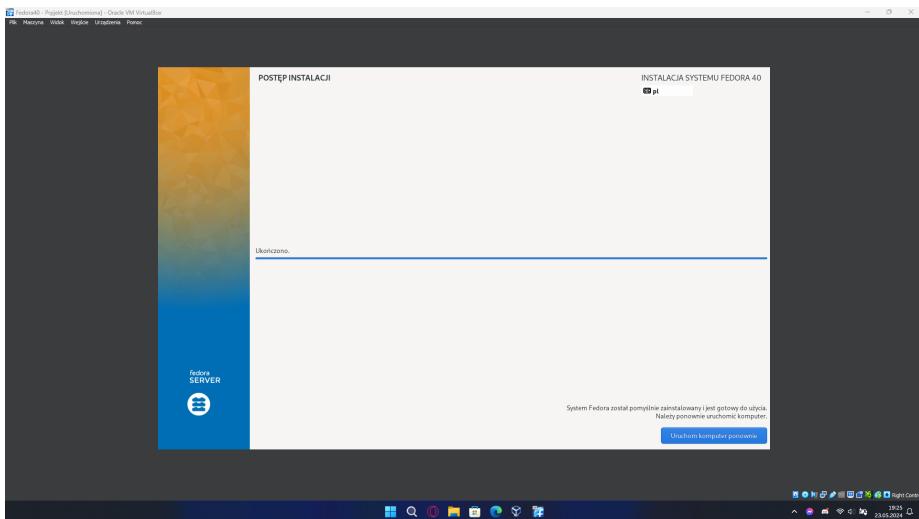
Rysunek 20: Stworzenie użytkownika – admin

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



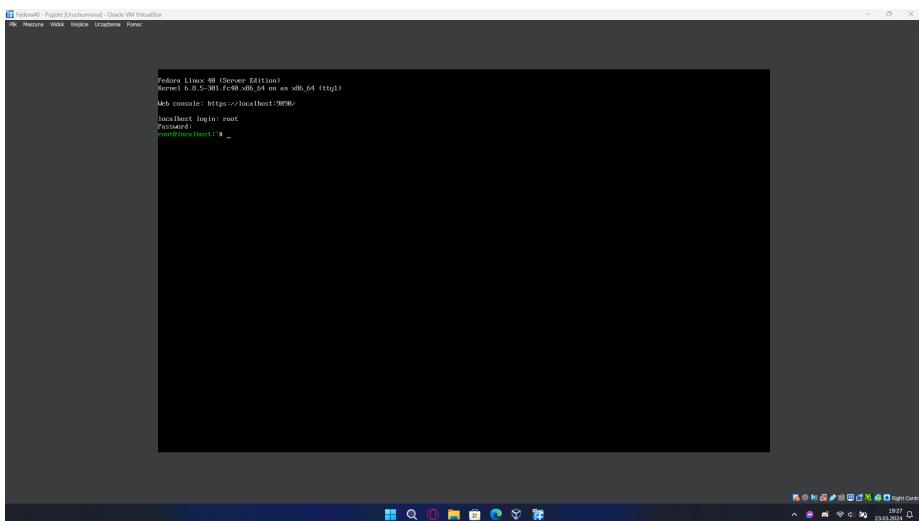
Rysunek 21: Ekran postępującej instalacji

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

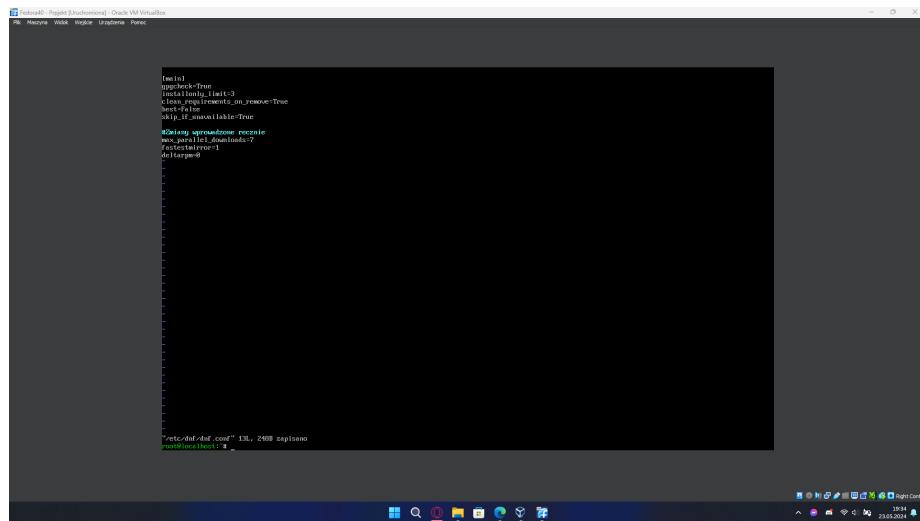
#### 4.2.2 Wstępna konfiguracja

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

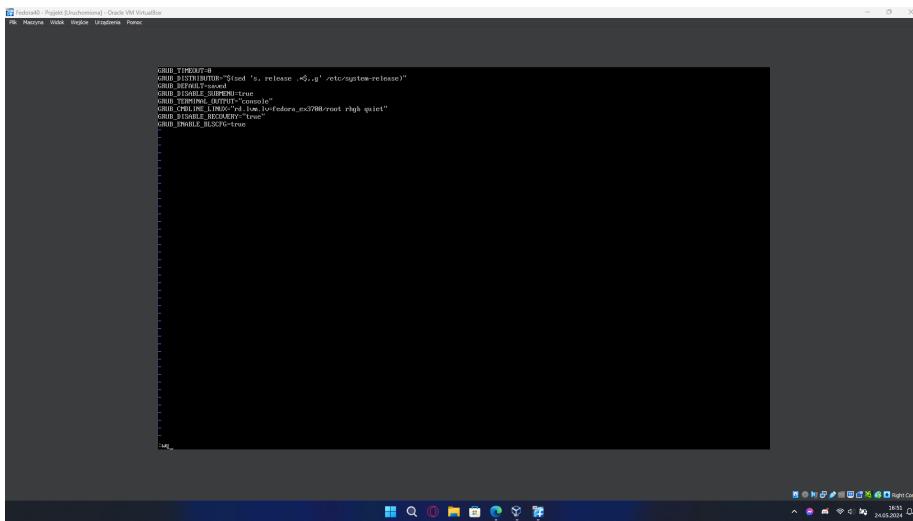
Teraz po skonfigurowaniu menadżera pakietów można wykonać aktualizację pakietów.

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)

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

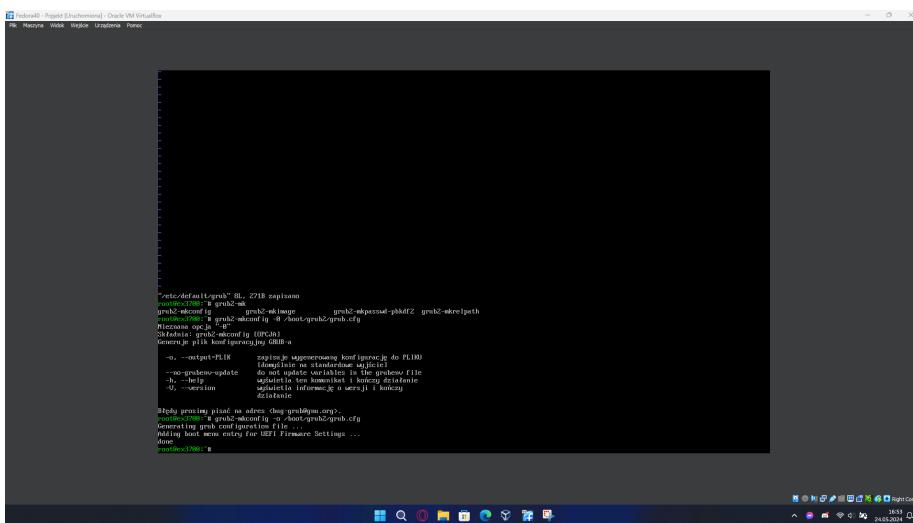
W kolejnym kroku zmieniłem GRUB\_TIMEOUT=5 na GRUB\_TIMEOUT=0  
Co można zauważyc poniższym zdjeciu.



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

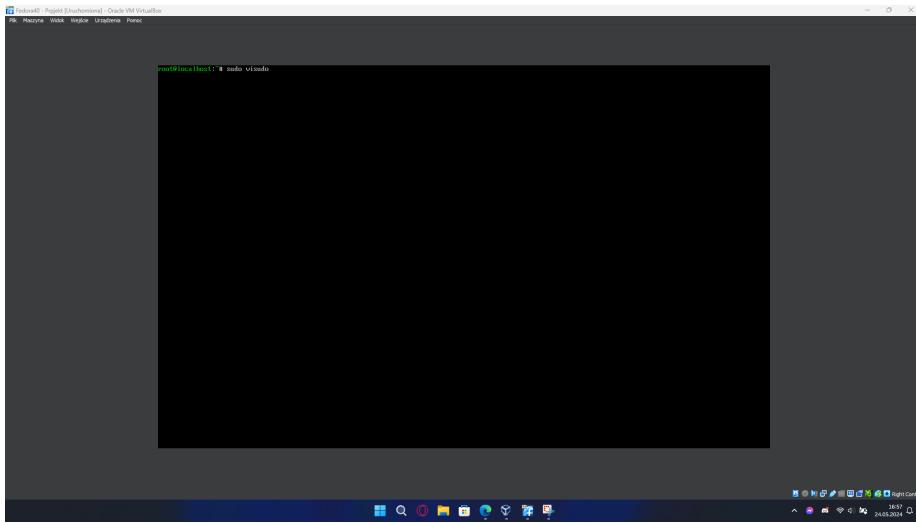
Aby zatwierdzić zmiany należy użyć komendy:

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



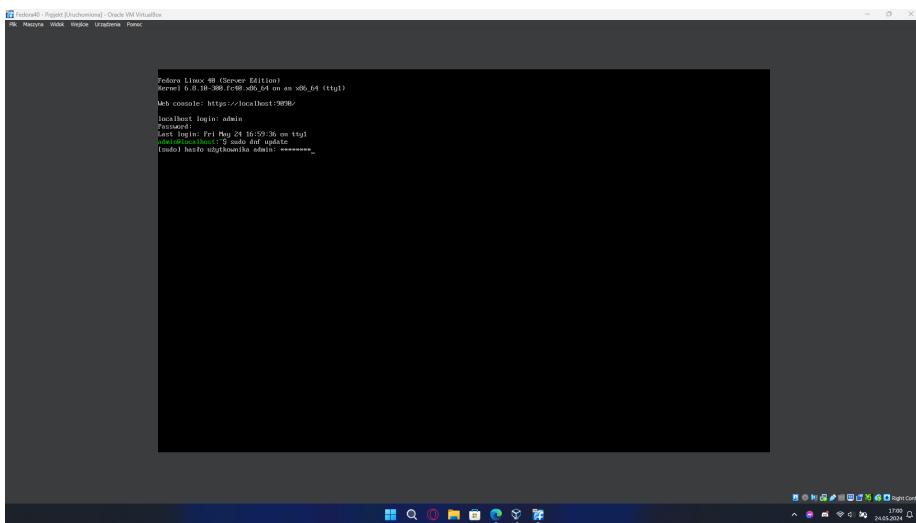
Rysunek 28: Zastosowanie zmian po edycji grub

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



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

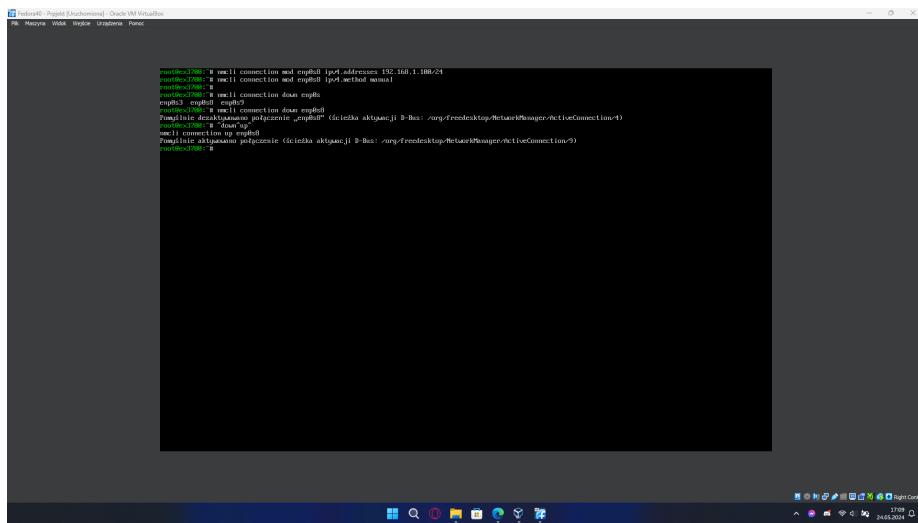
Efekt powyższego kroku:



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

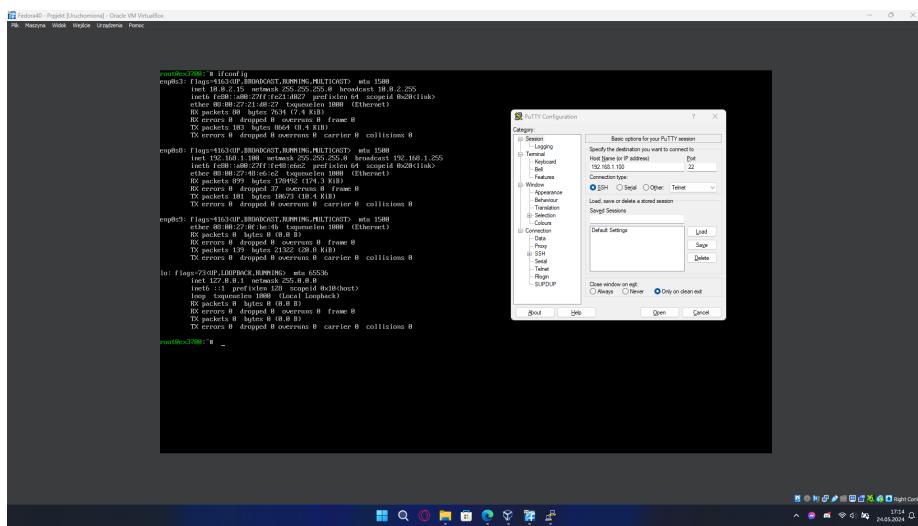
## 4.3 Konfiguracja SSH

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



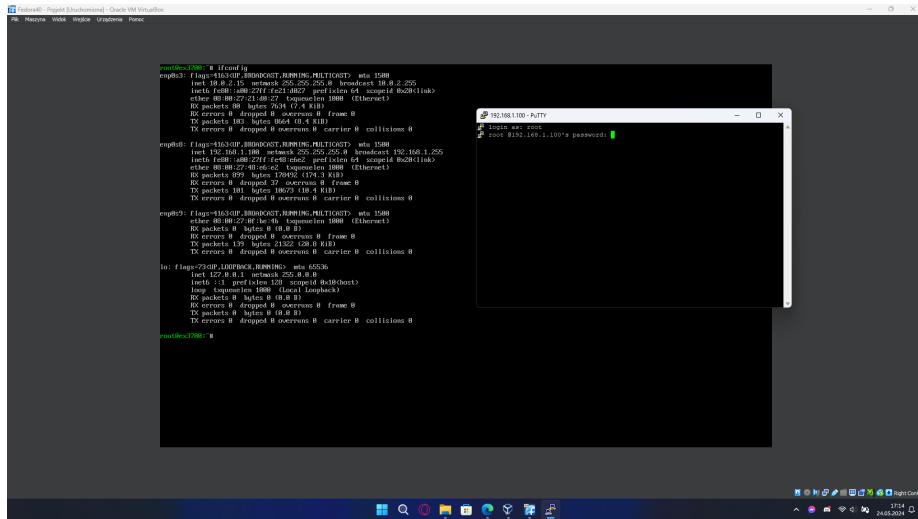
Rysunek 31: konfiguracja karty sieciowej

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



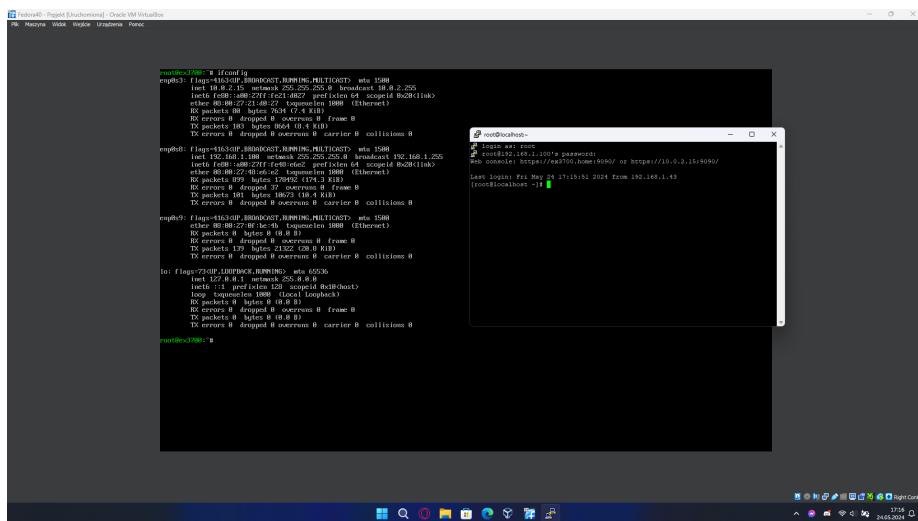
Rysunek 32: Konfiguracja aplikacji PuTTY

Próba zalogowania na konto root'a:



Rysunek 33: Podlaczenie poprzez PuTTY na konto root'a

Wynik powyższego kroku:

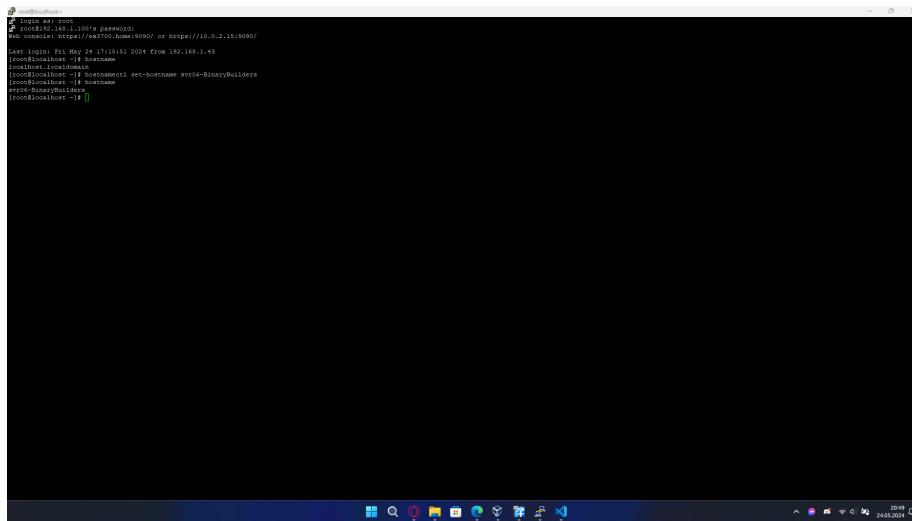


Rysunek 34: Wynik połączenia poprzez PuTTY

## 4.4 Nazwa serwera – hostname

Aby zmienić nazwę serwera (hostname) można użyć komendy:

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



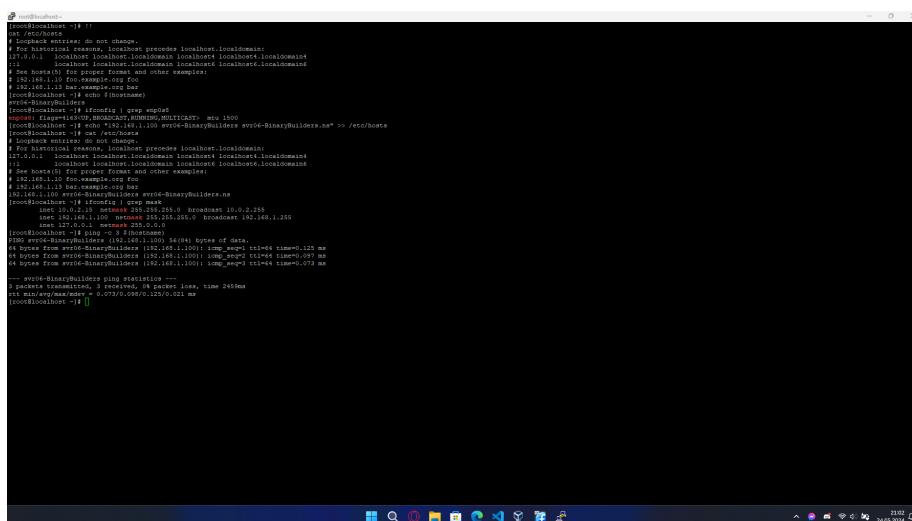
```
[root@svr04 ~]# hostnamectl set-hostname svr04-BinaryBuilders
[root@svr04 ~]#
```

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
```

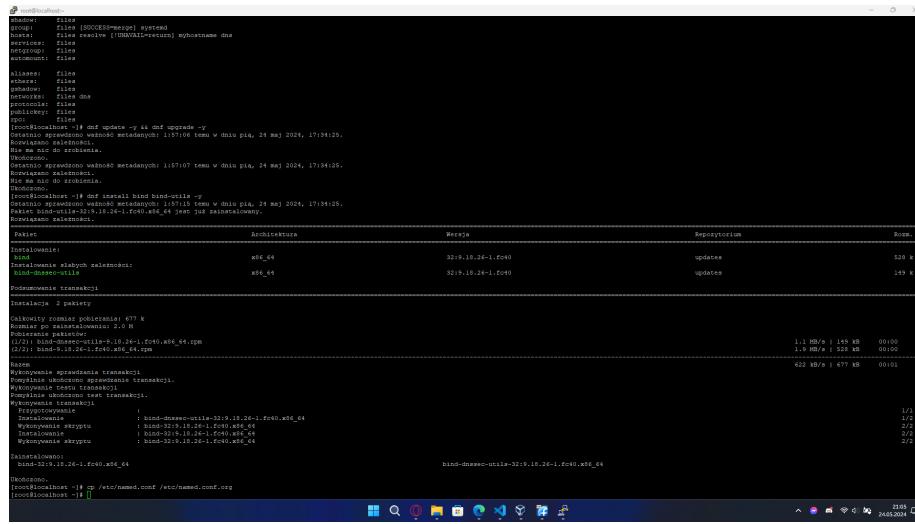
Po zainstalowaniu wymaganych pakietów należy wykonać kopię zapasową plików konfiguracyjnych. Można to zrobić komendą:

- 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
```

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.recurse";
    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";
};

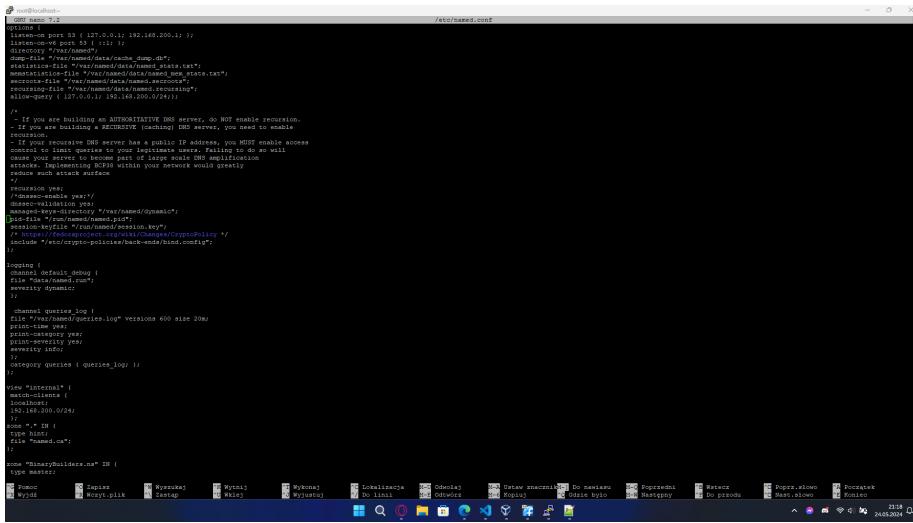
zone "BinaryBuilders.ns" IN {
    type master;
    file "BinaryBuilders.ns.lan_in";
    allow-update { none; };
};

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.

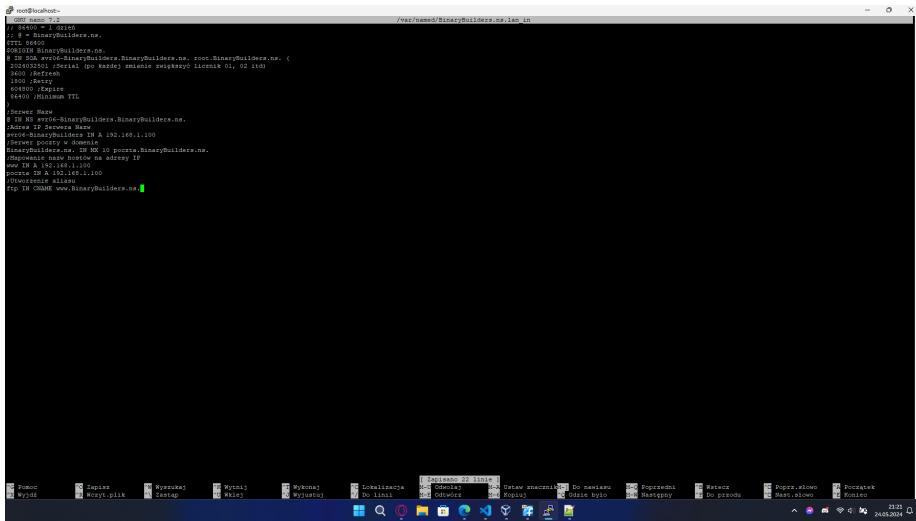
W kolejnym kroku trzeba utworzyć plik strefy podstawowej. W moim przypadku jest to plik /var/BinaryBuilders.ns.lan.in. Zawartość tego pliku:



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.

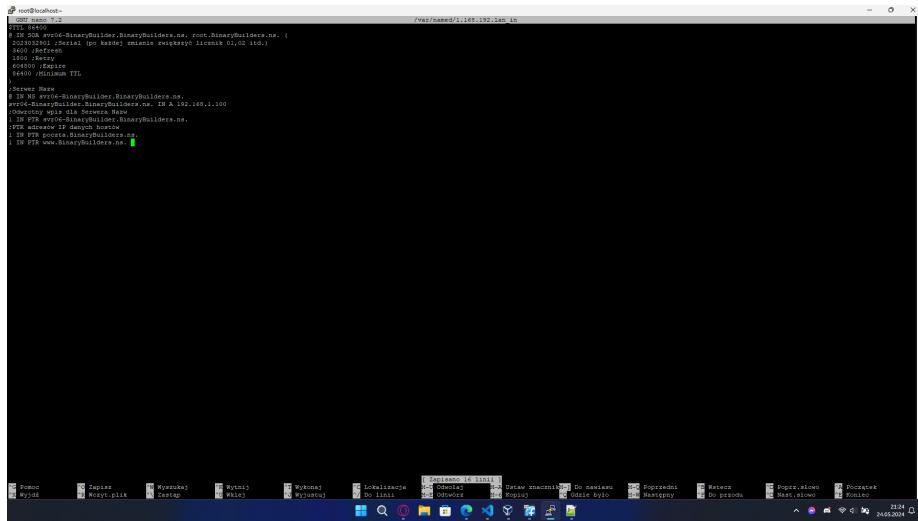
```

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

```



Rysunek 41: zawartość pliku strefy dla przeszukiwania wstecznego

Wytłumaczenie powyższych poleceń:

- `systemctl start named`

Komenda ta uruchomi usługe

- `systemctl enable named`

Polecenie to spowoduje że usługa będzie uruchamiana automatycznie przy włączeniu serwera.

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

Dodaje reguły 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

Modyfikuj połaczenie `enp0s9`, aby używało serwera DNS o adresie 192.168.230.1

- nmcli con down enp0s9 & & nmcli con up enp0s9

Dezaktywuje i ponownie aktywuje połączenie sieciowe enp0s9

- ## • `rndc reload`

Przeładowuj konfigurację serwera serwera DNS

- www.dz-statoen

Hide status

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

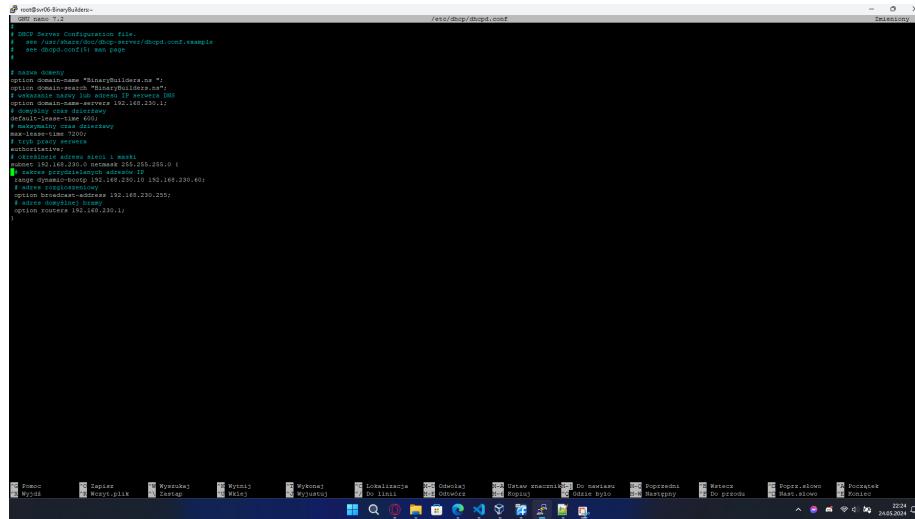
Po zainstalowaniu odpowiednich pakietów dobrze jest wykonać kopię zapasową oryginalnego pliku konfiguracji, co widać na zrucie powyżej. Można to zrobić następującą komendą:

```
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;
# określne 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

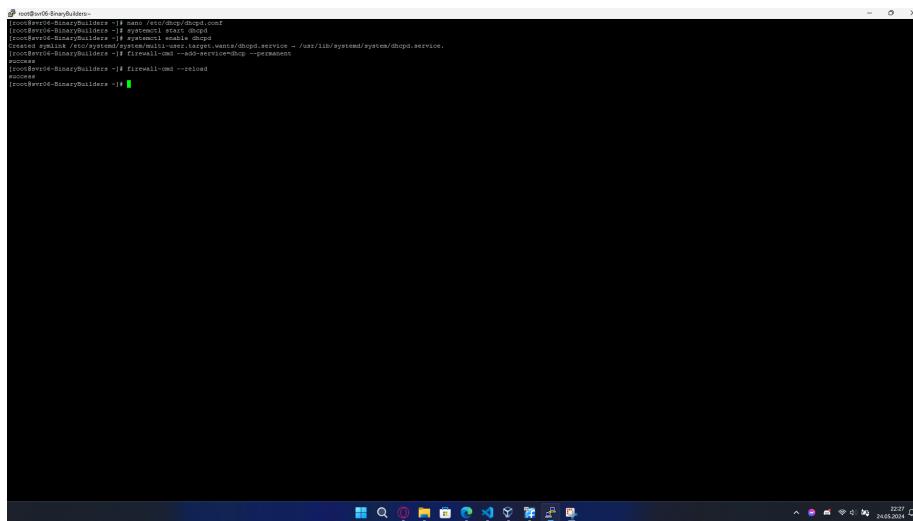
Następnym krokiem jest wprowadzenie kilku poleceń:

```
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.



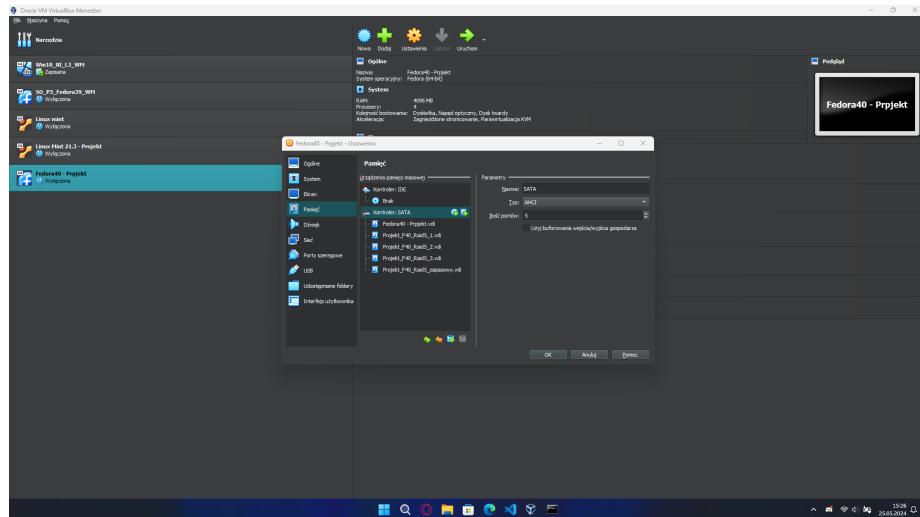
```
[root@srv06-BinaryBuilders ~]# nano /etc/dhcp/dhcpd.conf
[root@srv06-BinaryBuilders ~]# systemctl start dhcpcd
[root@srv06-BinaryBuilders ~]# systemctl enable dhcpcd
[root@srv06-BinaryBuilders ~]# useradd -r -u 1000 -g 1000 -s /usr/bin/false -c 'dhcp' -m -d /var/lib/centos/dhcp - /usr/lib/systemd/system/dhcpcd.service
[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@centos-OptiPlex-5090 ~]# mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sd[b-d] --spare-devices=1 /dev/sde
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: Defaulting to use metadata 1.2
mdadm: array /dev/md0 starting...
mdadm: /dev/sdb[0] added -> /dev/md0[0]
mdadm: cannot open /dev/sd[c]: No such file or directory
mdadm: cannot open /dev/sd[d]: No such file or directory
mdadm: cannot open /dev/sd[e]: No such file or directory
mdadm: /dev/sde[0] added -> /dev/md0[1]
mdadm: /dev/sdc[0] added -> /dev/md0[2]

      Version : 1.2
Creation Time : Sat May 25 19:54:12 2008
          Raid Level : raid5
        Array Size : 1067520 (10.99 GiB 10.73 GB)
    Used Dev Size : 3225070 (5.00 GiB 5.34 GB)
    Raid Devices : 3
      Total Dev Size : 975000 (9.33 GiB 9.09 GB)
 Persistence: Superblock is persistent

  Update Time : Sat May 25 19:55:07 2008
        State : clean
    Active Devices : 3
Working Devices : 3
Standby Devices : 0
     Spare Devices : 1

      Layout : left-symmetric
        Chunk Size : 512K

Confining Policy : 1+0+0

[root@centos-OptiPlex-5090 ~]# mdadm --status /dev/sd[b-d]
mdadm: /dev/sd[b] is part of /dev/sd[b-d] (md0) as /dev/sd[b]
      UUID: 7e4f7d2d:00705977:2a11:e1bd1fbc
  Events: 18

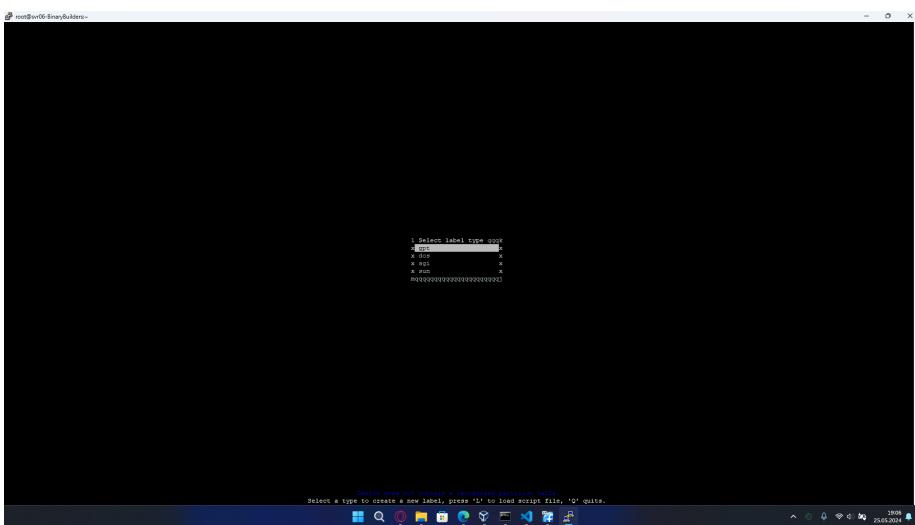
Number Major Minor RaidDevice State
  0   8      32       0      active sync  /dev/sdb
  1   8      32       1      active sync  /dev/sdc
  2   8      32       2      active sync  /dev/sdd
  3   8      44       0      spare   /dev/sde

[root@centos-OptiPlex-5090 ~]# 0

```

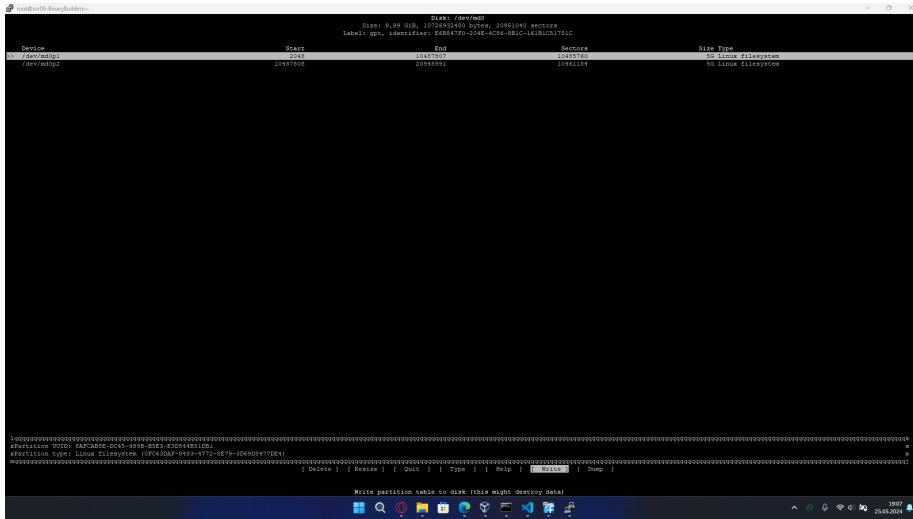
Rysunek 47: Stworzenie macierzy raid 5

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



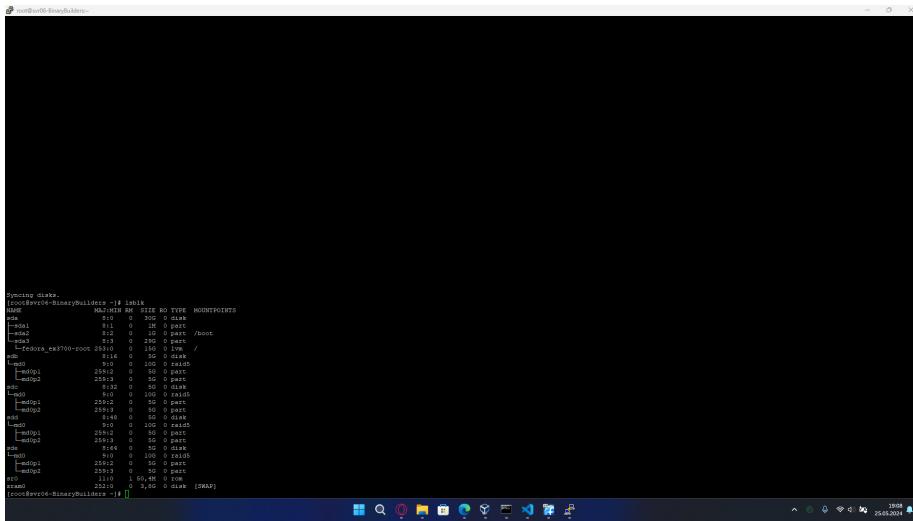
Rysunek 48: Partycjonowanie macierzy narzędziem cfdisk

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

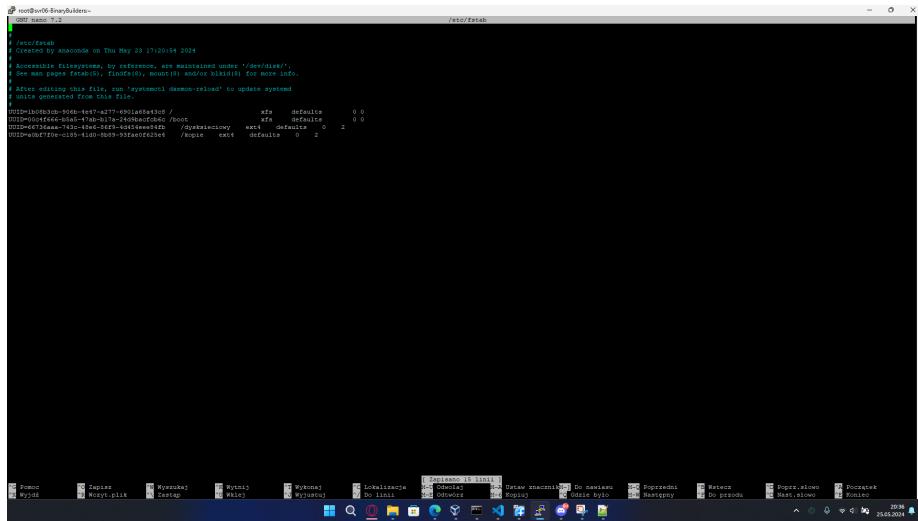
Formatowanie przed chwilą stworzonych partycji (w systemie plików ext4), stworzenie katalogów /dysksieciowy, /kopie, zamontowanie partycji do tych katalogów oraz wyświetlenie id dysków i partycji w systemie.

Rysunek 51: Przygotowanie ścieżek do montowania

Następnym krokiem jest zapewnienie automatycznego montowania utworzonych partycji. Aby to osiągnąć należy zmodyfikować /etc/fstab, ale najpierw wato wykonać kopię, gdyż jest to kluczowy składnik sysyemu. 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-24d9bacfcbb6c /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](#).

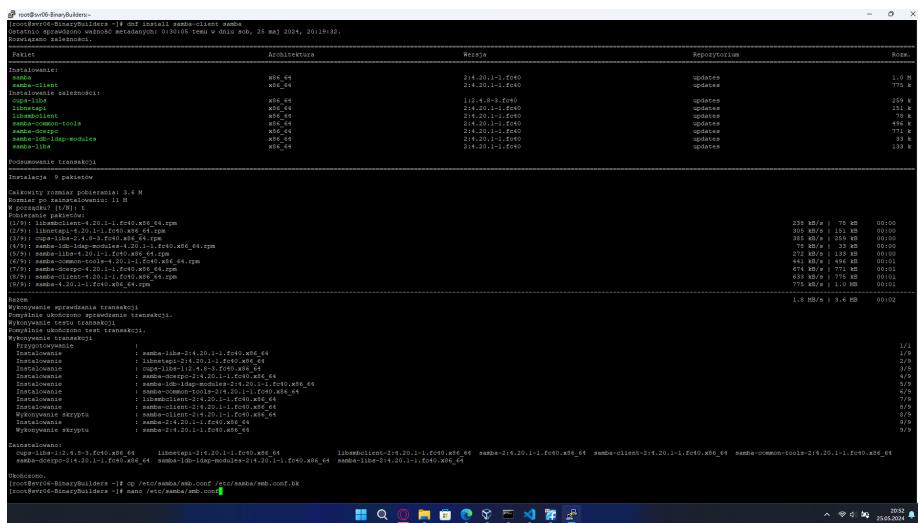
## 4.8 Samba – instalacja i konfiguracja

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
```



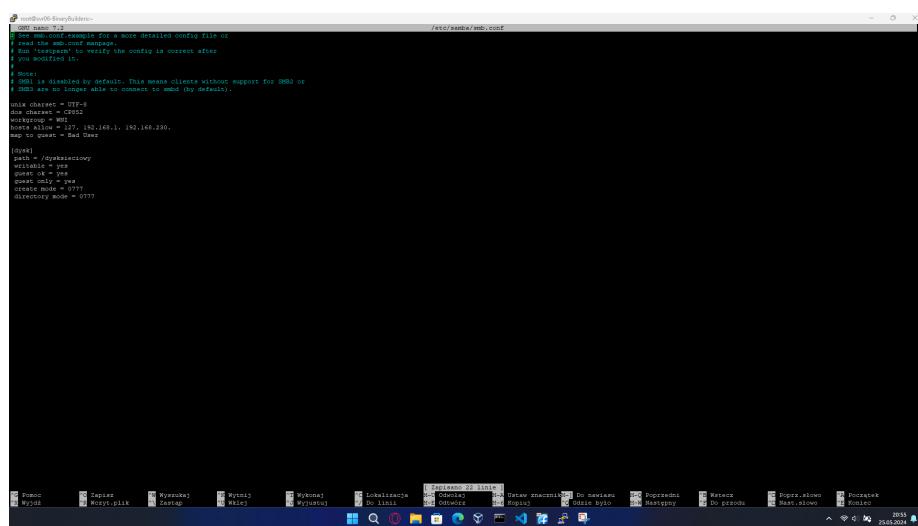
Rysunek 53: Samba – instalacja

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

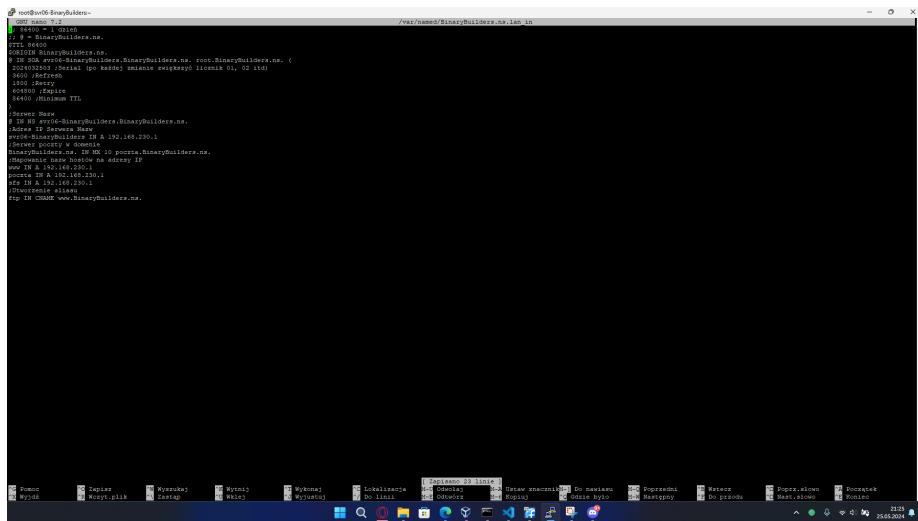
Ustawia w SELinux politykę, która pozwala Sambie na eksportowanie wszystkich udziałów z prawami do odczytu i zapisu. Dzięki temu Samba może zarządzać plikami z pełnym dostępem zgodnie z ustawieniami SELinux.

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
```

```
[root@localhost ~]# curl -s https://raw.githubusercontent.com/luke-labs/luke-labs/main/luke-labs.sh | sh
[luke-labs] LuKE-Labs v1.0.0 - A multi-user exploit development environment for Linux
[luke-labs] Copyright (c) 2024 by Luke Labs LLC
[luke-labs] https://luke-labs.com
[luke-labs] https://github.com/luke-labs/luke-labs

[luke-labs] LuKE-Labs is a multi-user exploit development environment for Linux. It provides a
[luke-labs] secure, isolated, and controlled environment for developing and testing exploits.
[luke-labs] It includes a web-based interface for managing users, a debugger, and various tools
[luke-labs] for exploit development.

[luke-labs] This environment is designed for ethical hacking, penetration testing, and
[luke-labs] research purposes only. It is not intended for illegal or malicious use.

[luke-labs] LuKE-Labs is open source and available under the MIT License. You can find the
[luke-labs] source code on GitHub at https://github.com/luke-labs/luke-labs.

[luke-labs] LuKE-Labs is a trademark of Luke Labs LLC.
```

Rysunek 57: Instalacja serwera HTTP

W kolejnym kroku wykonuje kopie oryginalnego pliku kongiguracyjnego serwera http. Robię to następującą komendą:

```
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

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

```
# /etc/httpd/conf.d/ssl.conf
#
# GDN name: 7.2
# http://sthttpd-apache.org/docs/7.2/mod/core/sslconf.html#sslconf
#
# See more information:
#   http://sthttpd-apache.org/docs/7.2/mod/core/sslconf.html#sslconf
#
# Options Indexes FollowSymLinks
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   All, None, Options, FileInfo, AuthName, AuthType, AuthBasicAuth
#
# AllowOverride All
#
# Controls who can get stuff from this server.
#
# Require all granted
</Directory>
</VirtualHost>
<DirectoryIndex> sets the file that Apache will serve if a directory
is requested.
<FilesModule dir_module>
  DirectoryIndex index.html index.cgi index.php
</FilesModule>

# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
<Files ~\.(htaccess|htpasswd)$>
  Require all denied
</Files>

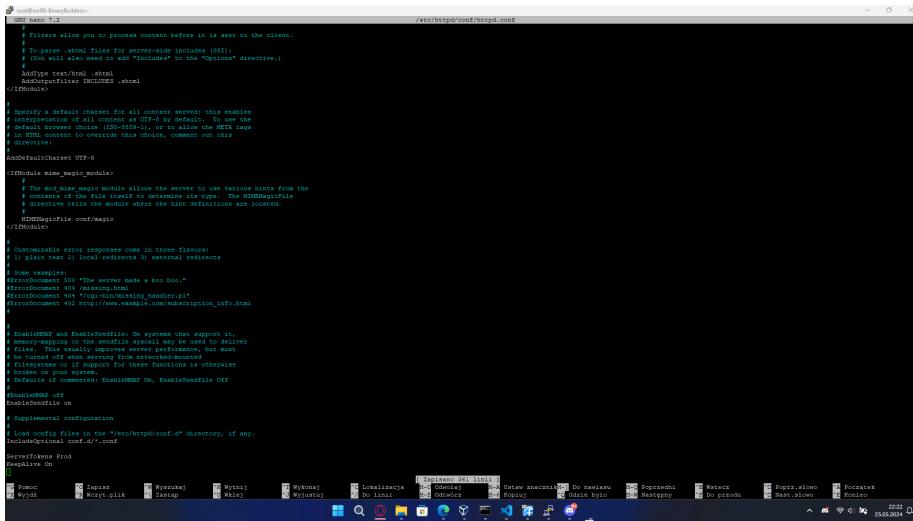
<ErrorLog> The location of the error log file
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, then Apache will use the ErrorLog directive from the
# configuration file. If you do not define an error logfile for a <VirtualHost>
# container, then host's errors will be logged there and not here.
<ErrorLog> logs/error_log

<LogLevel> Control the number of messages logged to the error log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
<LogLevel> warn

<FilesModule log_config_module>
  # The following directives define some format nickname for use with
  # the LogFormat directive.
  # 
  # LogFormat "%{$_GET}i %$_GET{s}i %$_GET{a}i %$_GET{U}i %$_GET{A}i" combined
  # LogFormat "%{$_GET}i %$_GET{s}i %$_GET{a}i %$_GET{U}i %$_GET{A}i" common
</FilesModule>

<FilesModule logio_module>
  # 
</FilesModule>
```

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



```

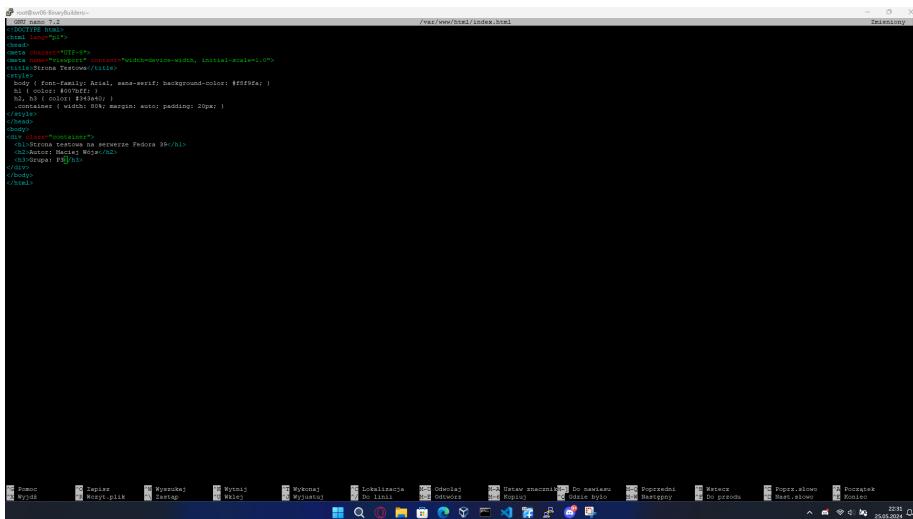
[root@www05 ~]# cat /etc/httpd/conf/httpd.conf
# This file contains the configuration for the Apache web server.  See
# /usr/share/doc/httpd-2.4.10/html/manual/mod/mod_config.html
# for details on configuration.  Several sections of this file are
# commented out--see the comments in those sections for more information.
#
# Configuration parameters, defined here, apply to the whole server.
#
# User (or group) and Group you run as.
#ServerName www05.silesia.pl
#User www-data
#Group www-data
#
# Document root of your site.  Change to "/var/www/html" or point to
# a subdirectory below it.
#
#DocumentRoot /var/www/html
#
# The path to the .htaccess files directory.
#
#    Options Indexes FollowSymLinks
#    AllowOverride All
#    Require all granted
#
#
# DirectoryIndex: sets the file that Apache will serve if no file
# is specified in the URL.  Set DirectoryIndex to "index.html"
# to make Apache serve index.html files by default.
#
#DirectoryIndex index.html
#
# Access log format:
#
#    LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\""
#    CustomLog "/var/log/httpd/error_log" combined
#
#
# ErrorLog: where to log error messages.
#ErrorLog "/var/log/httpd/error_log"
#
# LogLevel: how many messages from each module to log.
#LogLevel warn
#
# If you want to override the global error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# ErrorDocument 400 "The server made a boo boo."
# ErrorDocument 404 "/wp-content/themes/headers.php"
# ErrorDocument 404 "http://www.example.com/subscription_info.html"
#
# EnableEMSG and EnableSendfile On systems that support it,
# EnableSendfile off will not be used to deliver
# files. This usually improves server performance, but must
# be disabled if support for these functions is otherwise
# disabled or commented; EnableEMSG On, EnableSendfile Off
#
#EnableEMSG off
#EnableSendfile on
#
# Supplemental configuration
# Load config files in the "/etc/httpd/conf.d/" directory, if any.
#Include conf.d/*
#
# Virtual hosts
# Keepalive On
#
#
```

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.

```
root@nuc5:~# curl http://127.0.0.1:8080/index.php
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 103
Date: Mon, 10 Jul 2023 10:45:14 GMT
Server: Apache/2.4.42 (Ubuntu)
X-Powered-By: PHP/8.1.12-1+ubuntu22.04.1+deb.sury.org+1

<!DOCTYPE html>
<html name="index">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            body { background-color: #f0f0f0; font-family: Arial, sans-serif; background-color: #f0f0f0; text-align: center; padding-top: 50px; }
            .welcome { background-color: #ffccbc; padding: 20px; margin: auto; width: 50%; border-radius: 10px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }
            .welcome p { color: #333; }
        </style>
    </head>
    <body>
        <div class="welcome">
            <p>Witaj Macie! Wpisz grupę: P9 na podkładzie PHP !!!</p>
        </div>
    </body>
</html>
```

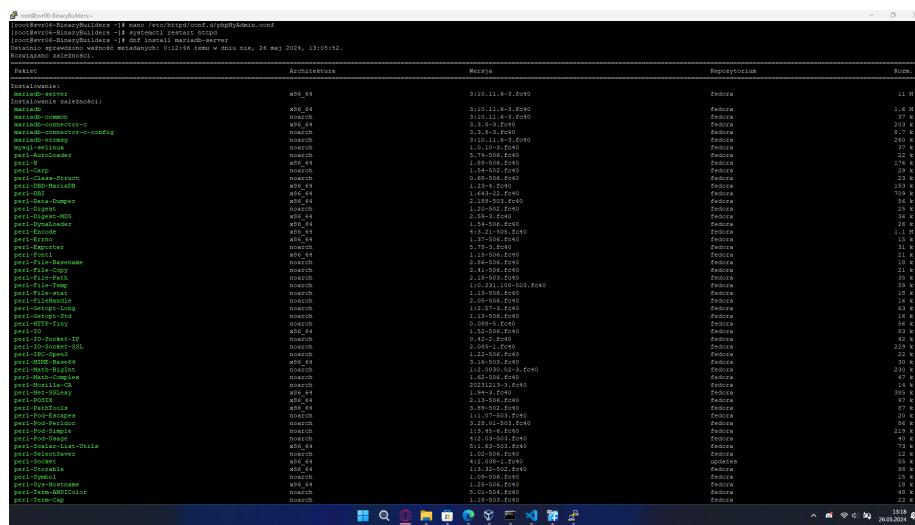
Rysunek 64: PHP – stworzenie strony internetowej

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

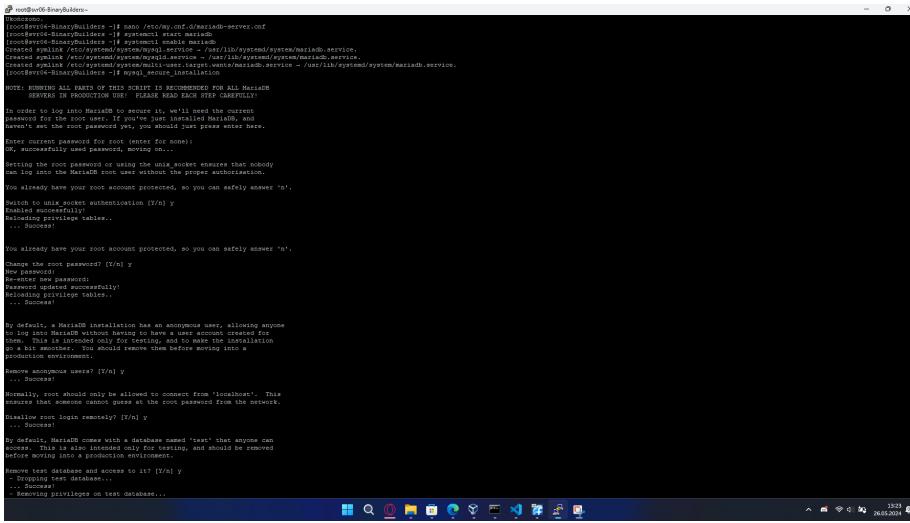
Do pliku /etc/my.cnf.d/mariadb-server.cnf w sekcji [mysqld] dodałem linię:

```
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 set a root password for security reasons.

To enter root log type 'root' or provide the root password.

Enter current password for root (enter for none):
Not currently used by 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 should be removed
before moving into a production environment.

Remove anonymous users? [Y/n] Y
... Success!
Normally, root should only be allowed to connect from 'localhost'. This
server that someone could 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!

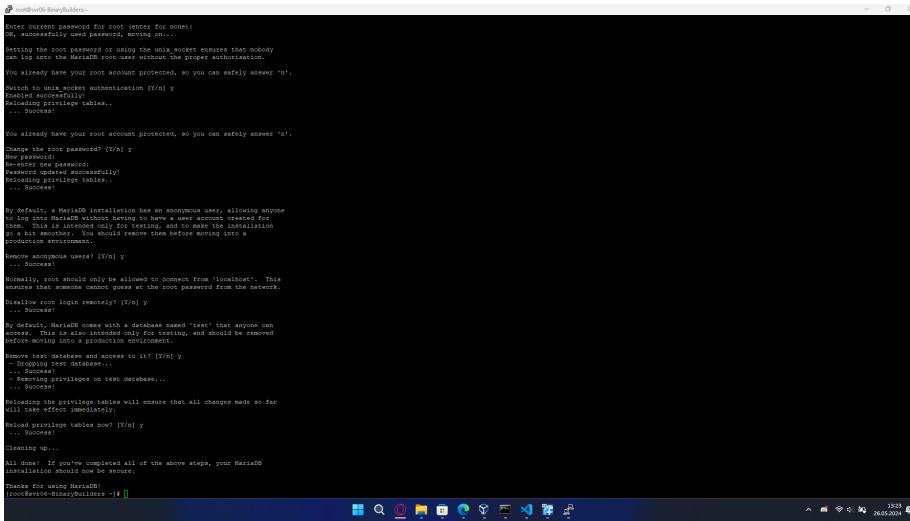
All privilege changes will ensure that all changes made so far
will take effect immediately.

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!
[root@DevW8-Server:~# ]
```

Rysunek 67: MySQL – instalacja część pierwsza



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

Enter current password for root (enter for none):
[REDACTED] successfully used password, moving on...
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 should be removed
before moving into a production environment.

Remove anonymous users? [Y/n] Y
... Success!
Normally, root should only be allowed to connect from 'localhost'. This
server that someone could 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!
[root@DevW8-Server:~# ]
```

Rysunek 68: MySQL – instalacja część druga

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

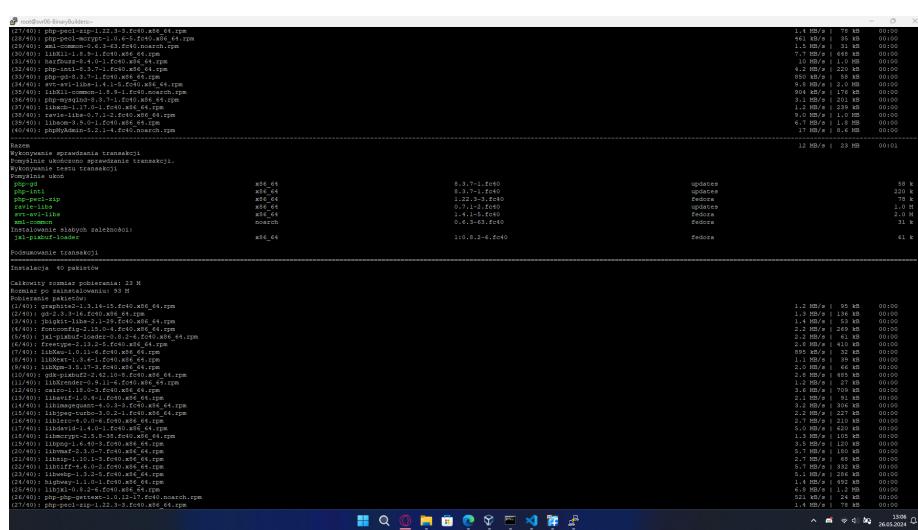
#### 4.12 phpMyAdmin – instalacja i konfiguracja

Aby zainstalować phpMyAdmin należy użyć komendy:

```
sudo dnf install phpMyAdmin php-mysqlnd php-mcrypt php-php-gettext -y
```

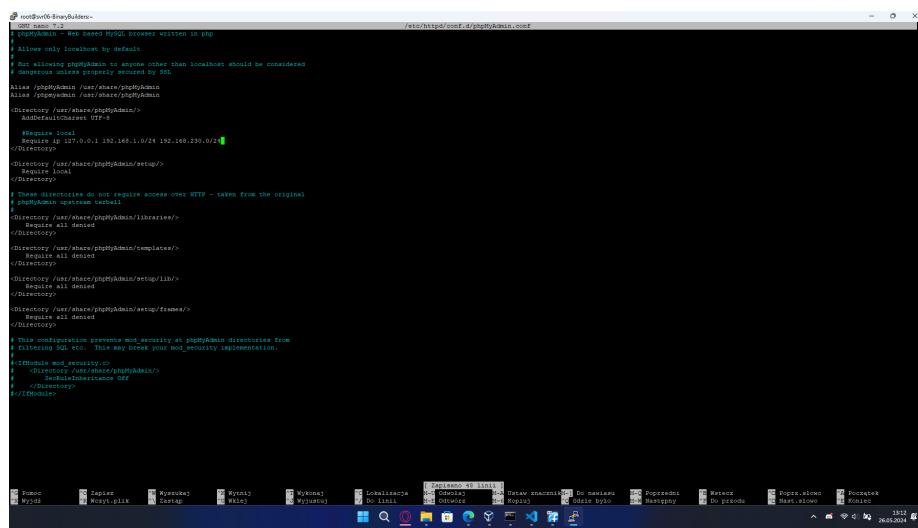
Po instalacji wykonuje kopię oryginalnego pliku konfiguracyjnego. Można to zrobić komendą:

```
sudo cp /etc/httpd/conf.d/phpMyAdmin.conf /etc/httpd/conf.d/phpMyAdmin.conf.org
```



Rysunek 69: phpMyAdmin – instalacja

Konfiguracja pliku /etc/httpd/conf.d/phpMyAdmin.conf. W tym pliku należy zmienić sieci tak aby odzwierciedlały potrzeby firmy.



Rysunek 70: Konfiguracja pliku /etc/httpd/conf.d/phpMyAdmin.conf

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

#### 4.13 UserDir na serwerze HTTP – konfiguracja

Gdy mamy już zainstalowany serwer http to pierwszym krokiem do skonfigurowania UserDir jest modyfikacja pliku konfiguracyjnego /etc/httpd/conf.d/userdir.conf, ale przed tym dobrze jest zrobić kopię zapasową tego pliku. Można to wykonać następującą komendą:

```
sudo cp /etc/httpd/conf.d/userdir.conf
```

Zrzut ekranu mojej konfiguracji pliku /etc/httpd/conf.d/userdir.conf

```
root@vps:~# nano -w .htaccess

# UserDir The name of the directory that is appended onto a user's home
# directory if a "/user" request is received.
#
# The path to the end user account "public_html" directory must be
# specified. This directory must be owned by the user account, and the user
# must have permissions to it. "script/public_html" must have permissions
# to execute, and any files contained therein must be world-readable.
# Otherwise, the client will only receive a 403 Forbidden message.
#
# Otherwise, the client will only receive a 403 Forbidden message.

#Includes and userdir.c0
#
# UserDir is disabled by default since it can confuse the presence
# of a username on the system (depending on home directory
# permissions).
#
#UserDir disabled

#
# To enable requests to /~user/ to serve the user's public_html
# directory, remove the "#UserDir disabled" line above, and uncomment
# the following line instead:
#
#Includes public_html
#/Includes

#Control access to UserDir directories. The following is an example
#for sites where these directories are examined as read-only.

Directory "/home/~public_html"
  AllowOverride FileInfo AuthConfig Limit Includes
  Order Allow,Deny Options FollowSymlinks IncludesNoExec
  AllowOverride All
  Options None
  MultiViews Off
  RequestHeader set method GET POST OPTIONS
/Directory
```

Rysunek 71: Konfiguracja pliku /etc/httpd/conf.d/userdir.conf

Teraz dodaję reguły SELinux, tworzę nowego użytkownika, oraz nadaję uprawnienia odpowiednim katalogom, następnie tworzę stronę użytkownika maciej.

Rysunek 72: Dodanie użytkownika maciej

## Kod html tej strony

```
GNU nano 3.2
<!DOCTYPE html>
<html lang="pl-PL">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            body {
                margin: 0;
                background-color: #f0f0f0;
                font-family: Arial, sans-serif;
            }
            .header {
                background-color: #333333;
                color: white;
                padding: 10px;
            }
            .header h1 {
                color: #ffff;
                margin: 0;
            }
            .content {
                padding: 20px;
            }
        </style>
    </head>
    <body>
        <div class="header">
            <h1>Witaj na stronie użytkownika Macieja!</h1>
        </div>
        <div class="content">
            <p>Witaj na stronie użytkownika Macieja!</p>
        </div>
    </body>
</html>
```

Rysunek 73: Stworzenie strony użytkownika maciej

Test działania dostępny jest [tutaj](#).

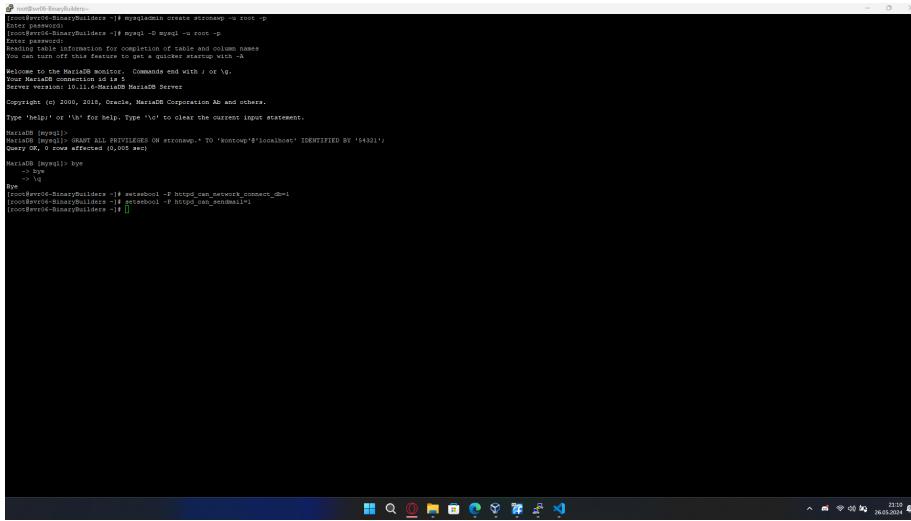
## 4.14 WordPress – instalacja i konfiguracja

Aby zainstalować WordPress'a należy wykonać polecenie:

```
sudo dnf install wordpress -y
```

Rysunek 74: WordPress – instalacja

Stworzenie bazy danych dla WordPress'a oraz stworzenie użytkownika konta administratora bazy danych o nazwie kontowp.



```
[root@CentOS-Server ~]# mysqladmin create strongwp -u root -p
[root@CentOS-BinaryBuilder ~]# mysql -D strong -u root -p
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with '--skip-table-names'.
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Server version: 10.1.14-MariaDB MariaDB Server
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 [strongwp]>
MariaDB [strongwp]> CREATE USER 'kontowp'@'localhost' IDENTIFIED BY 'kontowp';
Query OK, 0 rows affected (0.00 sec)
MariaDB [strongwp]> GRANT ALL PRIVILEGES ON strongwp.* TO 'kontowp'@'localhost' IDENTIFIED BY 'kontowp';
Query OK, 0 rows affected (0.00 sec)
[root@CentOS-BinaryBuilder ~]# exit
[root@CentOS-BinaryBuilder ~]#
```

Rysunek 75: Stworzenie bazy danych dla WordPress'a

Do pliku etc/httpd/conf.d/wordpress.conf należy wprowadzić następującą zawartość

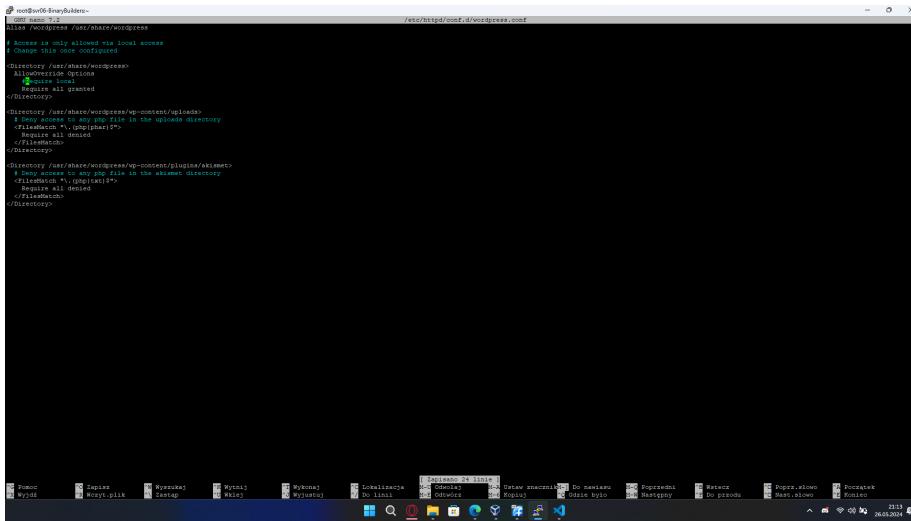
```
Alias /wordpress /usr/share/wordpress

# Access is only allowed via local access
# Change this once configured

<Directory /usr/share/wordpress>
    AllowOverride Options
        #Require local
        Require all granted
</Directory>

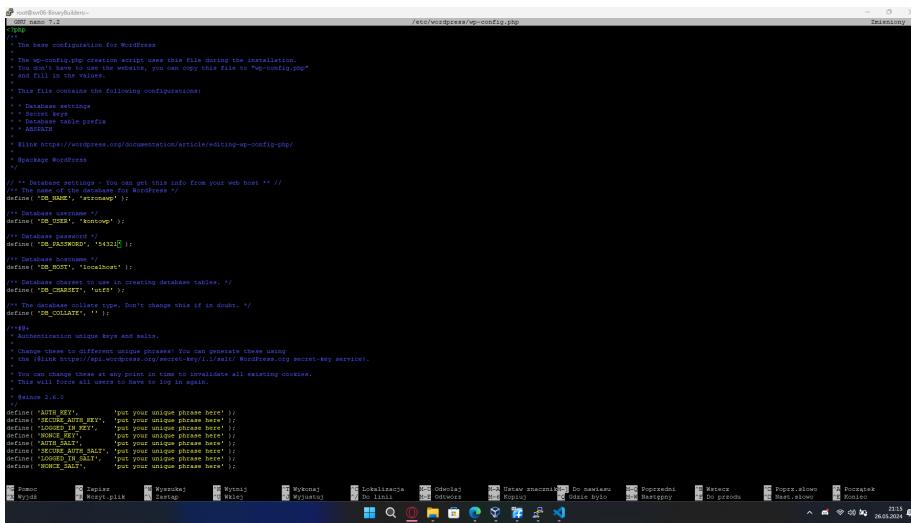
<Directory /usr/share/wordpress/wp-content/uploads>
    # Deny access to any php file in the uploads directory
    <FilesMatch "\.(php|phar)$">
        Require all denied
    </FilesMatch>
</Directory>

<Directory /usr/share/wordpress/wp-content/plugins/akismet>
    # Deny access to any php file in the akismet directory
    <FilesMatch "\.(php|txt)$">
        Require all denied
    </FilesMatch>
</Directory>
```



Rysunek 76: Edycja pliku /etc/httpd/conf.d/wordpress.conf

W pliku /etc/wordpress/wp-config.php należy zmienić nazwę bazy danych na taką jak ustalono w kroku 2, tak samo w przypadku użytkownika kontop.

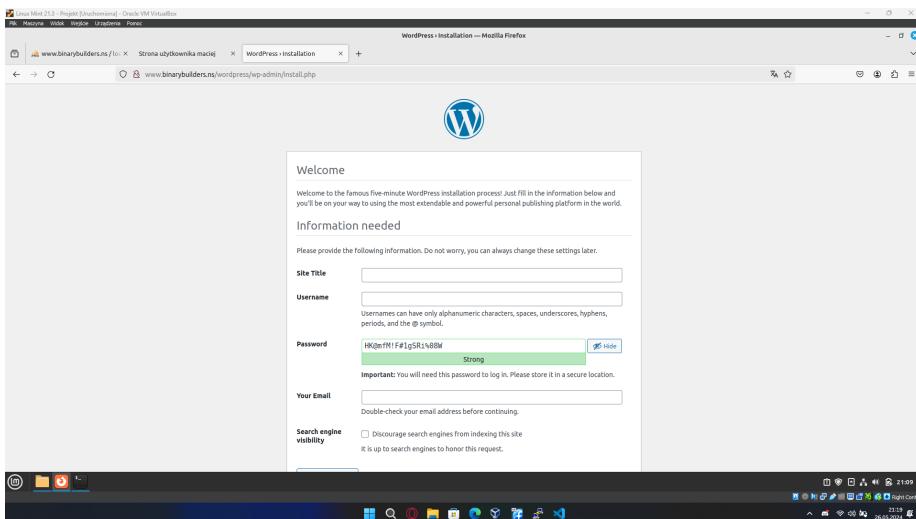


Rysunek 77: Edycja pliku /etc/wordpress/wp-config.php

W kolejnym kroku należy wykonać następujące polecenia:

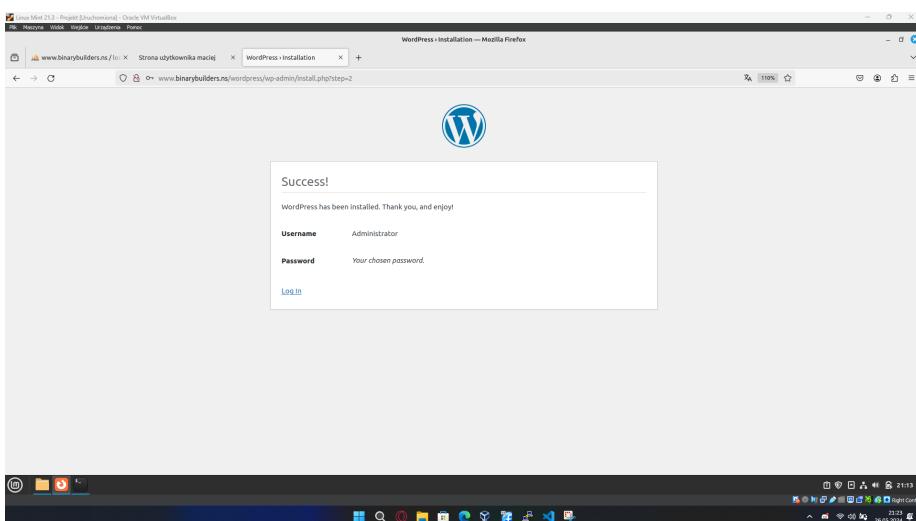
```
sudo setsebool -P httpd_can_network_connect_db=1
sudo setsebool -P httpd_can_sendmail=1
sudo systemctl restart httpd
```

W kolejnym kroku należy z poziomu klienta w przeglądarce wejść na stronę [www.BinaryBuilders.ns/wordpress](http://www.BinaryBuilders.ns/wordpress) oraz zainstalować stronę WordPress.



Rysunek 78: Instalacja WordPress

#### Efekt działania powyższego kroku

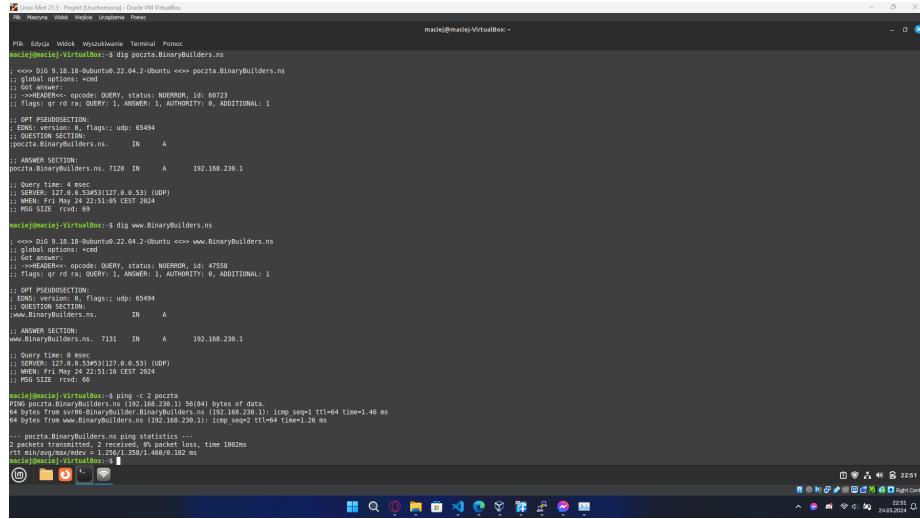


Rysunek 79: Instalacja WordPress – sukces

Test działa WordPress'a można znaleźć się [tutaj](#).

## 5 Testy działania wdrożonych usług

### 5.1 DNS

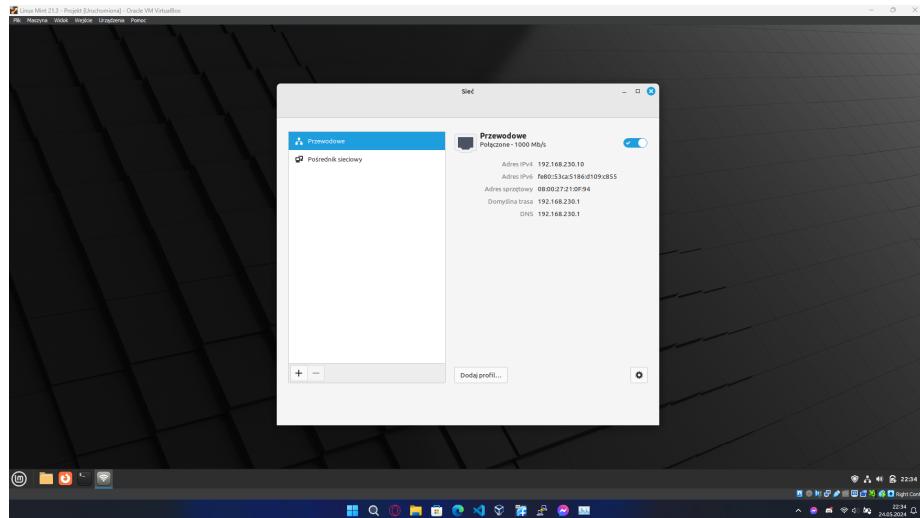


```
maciej@maciej-VirtualBox:~$ dig pocita.BinaryBuilders.ns
;; global options: +cmd
maciej@maciej-VirtualBox:~$ dig www.BinaryBuilders.ns
;; global options: +cmd
maciej@maciej-VirtualBox:~$ ping -c 2 pocita
PING pocita.BinaryBuilders.ns (192.168.230.1) 56(84) bytes of data.
64 bytes from www.BinaryBuilders.ns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

Rysunek 80: 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 81: Instalacja DHCP

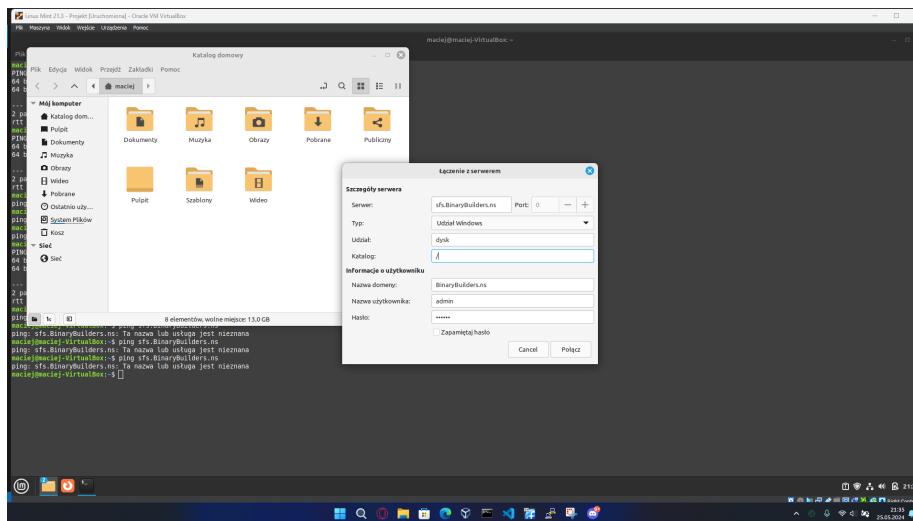
Jak widać na powyższym zdjęciu karta w systemie klienta ustawiona na sieć wewnętrzną dostała poprawny adres IP, adres bramy domyślnej i DNS. Na zdjęciu również widać że pula DHCP działa poprawnie.

### 5.3 Raid 5

Rysunek 82: Test automatycznego montowania partycji po ponownym uruchomieniu serwera

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

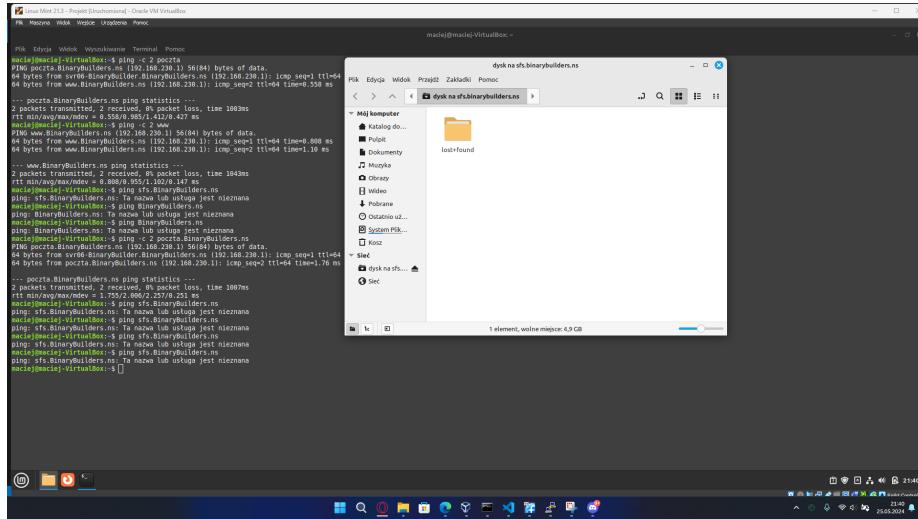
## 5.4 Samba



Rysunek 83: Samba – próba podłączenia się do udziału na serwerze

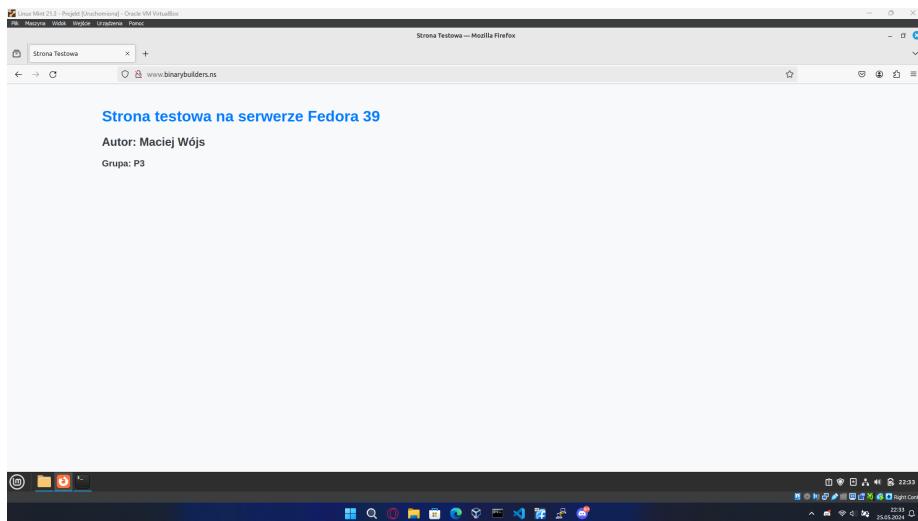
Aby podłączyć się z serwerem Samby należy otworzyć menadżer plików następnie otworzyć zakładkę Plik w lewym górnym rogu, następnie połącz z

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



Rysunek 84: Samba – wynik poprzedniego kroku

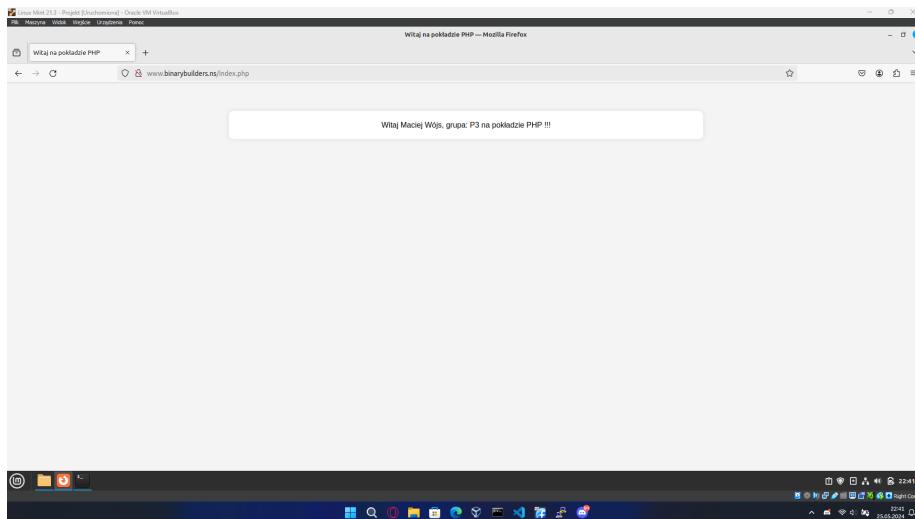
## 5.5 HTTP



Rysunek 85: Test działania serwera WWW

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

## 5.6 PHP



Rysunek 86: Strona html + PHP

Jak można zauważyc strona wykorzystujacą PHP działa 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=10333  processlist_id=208 MariaDB Server
Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.
Type 'help' or '\h' for help. Type '\u' to clear the current input statement.
MariaDB [(none)]> select user,host,password from mysql.user;
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| root | %    |          |
| root | localhost |          |
| mysql | localhost | invalid |
+-----+-----+-----+
3 rows in set (0,001 sec)

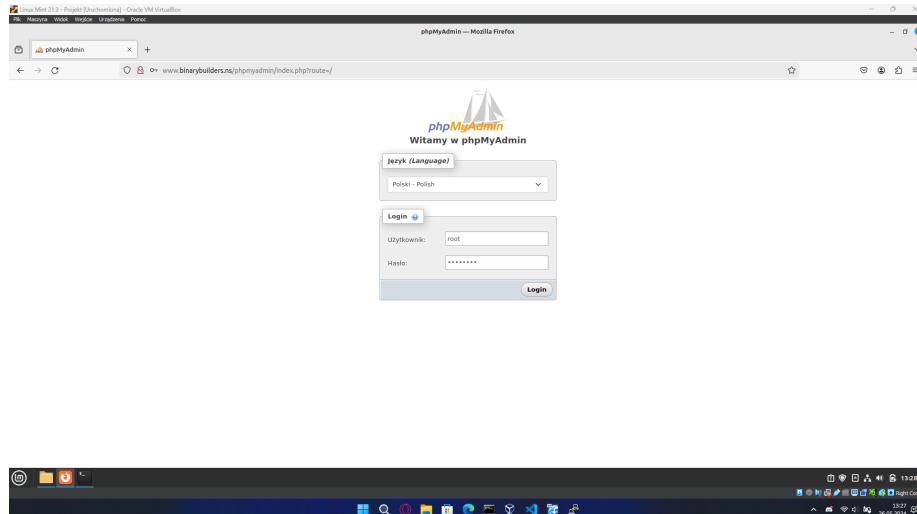
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| sys                |
+--------------------+
3 rows in set (0,001 sec)

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

Rysunek 87: Test usługi mariadb (MySQL)

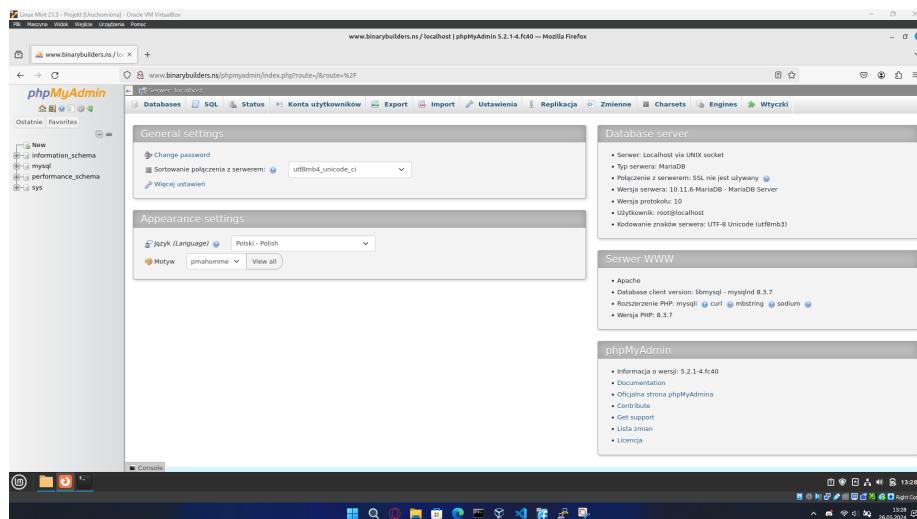
Jak widać połączenie z bazą danych działa. Następnym i ostatnim krokiem jest dodanie usługi MySQL do dozwolonych usług w zaporze ogniwowej.

## 5.8 phpMyAdmin



Rysunek 88: Test usługi phpMyAdmin – część pierwsza

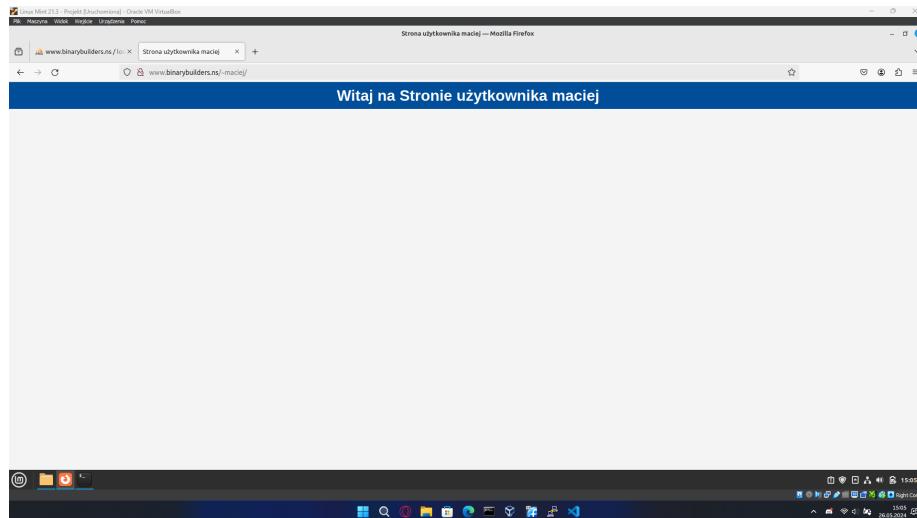
Pierwszym krokiem w testowaniu phpMyAdmin jest otworzenie strony serwera tej usługi w przeglądarce a następnie zalogowanie się na konto.



Rysunek 89: Test usługi phpMyAdmin – część druga

Jak widać po zalogowaniu mamy dostęp do administrowania bazami danych.

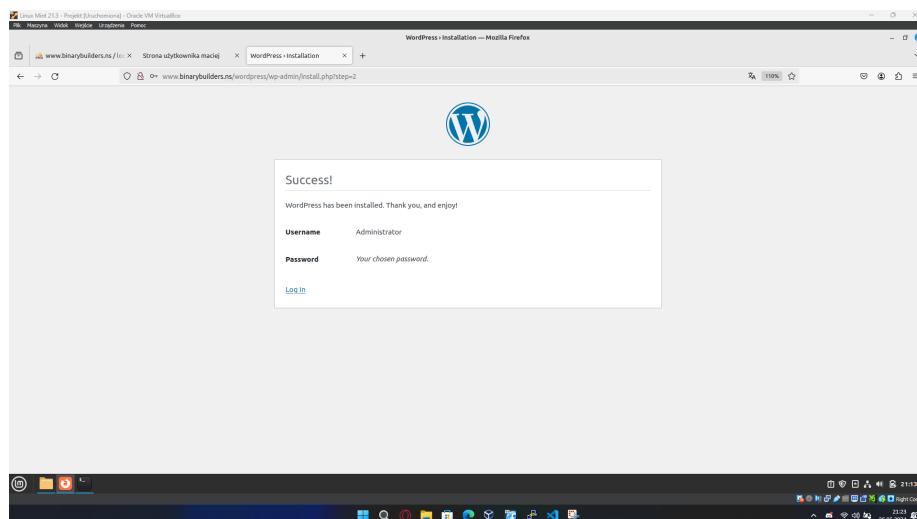
## 5.9 UserDir – serwer http



Rysunek 90: MySQL – instalacja część druga

W przeglądarce po wpisaniu adresu strony i ścieżki do profilu użytkownika maciej (tj. maciej) widać stronę użytkownika.

## 5.10 WordPress



Rysunek 91: Dashboard WordPress'a

Jak widać powyżej po zalogowaniu się na konto widać dashboard WordPress'a, skąd można skonfigurować stronę internetową.

## **6 Kod skryptu BASH, oraz tablica crontab**

FFFFFFFFFFFFFFn

## **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>.